KBO^3

Harald Zankl Aart Middeldorp

Institute of Computer Science
University of Innsbruck

21 August 2007

- Motivation
- Knuth-Bendix Order
- Encodings
- Experimental Results
- Concluding Remarks

Why Encode Termination Problems as Satisfiability Problems?

Why Encode Termination Problems as Satisfiability Problems?

- execution speed

Why Encode Termination Problems as Satisfiability Problems?

- execution speed
- ease of implementation

Why Encode Termination Problems as Satisfiability Problems?

- execution speed
- ease of implementation
- developments in SAT community are directly available

Why Encode Termination Problems as Satisfiability Problems?

- execution speed
- ease of implementation
- developments in SAT community are directly available

Why KBO?

Why Encode Termination Problems as Satisfiability Problems?

- execution speed
- ease of implementation
- developments in SAT community are directly available

Why KBO?

- more challenging than LPO

Kurihara & Kondo *1997 2004* *LPO*

Why Encode Termination Problems as Satisfiability Problems?

- execution speed
- ease of implementation
- developments in SAT community are directly available

Why KBO?

- more challenging than LPO

  Kurihara & Kondo *1997 2004* *LPO*

- existing implementations (in T$_\mathsf{T}$T and AProVE) based on polynomial time algorithm

  Korovin & Voronkov *2003*

  are slow

- Motivation
- Knuth-Bendix Order
- Encodings
- Experimental Results
- Concluding Remarks

Definition

- quasi-precedence $\gtrsim$ is quasi-order on signature $\mathcal{F}$

Definition

- quasi-precedence $\gtrsim$ is quasi-order on signature $\mathcal{F}$
- weight function (w, w_0) consists of mapping $w\colon \mathcal{F} \to \mathbb{N}$ and constant $w_0 > 0$ such that $w(c) \geqslant w_0$ for all constants $c \in \mathcal{F}$

Definition

- quasi-precedence $\gtrsim$ is quasi-order on signature $\mathcal{F}$
- weight function (w, w_0) consists of mapping $w\colon \mathcal{F} \to \mathbb{N}$ and constant $w_0 > 0$ such that $w(c) \geqslant w_0$ for all constants $c \in \mathcal{F}$
- weight of term t is

$$w(t) = \begin{cases} w_0 & \text{if } t \in \mathcal{V} \\ w(f) + \sum_{i=1}^{n} w(t_i) & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

Definition

- quasi-precedence $\gtrsim$ is quasi-order on signature $\mathcal{F}$
- weight function (w, w_0) consists of mapping $w\colon \mathcal{F} \to \mathbb{N}$ and constant $w_0 > 0$ such that $w(c) \geqslant w_0$ for all constants $c \in \mathcal{F}$
- weight of term t is

$$w(t) = \begin{cases} w_0 & \text{if } t \in \mathcal{V} \\ w(f) + \sum_{i=1}^{n} w(t_i) & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

- weight function (w, w_0) is admissible for quasi-precedence $\gtrsim$ if

$$f \gtrsim g \quad \forall\, g \in \mathcal{F}$$

whenever f is unary function symbol in $\mathcal{F}$ with $w(f) = 0$

Definition

Knuth-Bendix order $>_{\text{kbo}}$ on terms:
$s >_{\text{kbo}} t$ if $|s|_x \geqslant |t|_x$ for all $x \in \mathcal{V}$ and either

Definition

Knuth-Bendix order $>_{\text{kbo}}$ on terms:
$s >_{\text{kbo}} t$ if $|s|_x \geqslant |t|_x$ for all $x \in \mathcal{V}$ and either

1. $w(s) > w(t)$

Definition

Knuth-Bendix order $>_{\text{kbo}}$ on terms:
$s >_{\text{kbo}} t$ if $|s|_x \geqslant |t|_x$ for all $x \in \mathcal{V}$ and either

1. $w(s) > w(t)$
2. $w(s) = w(t)$ and either

Definition

Knuth-Bendix order $>_{\text{kbo}}$ on terms:
$s >_{\text{kbo}} t$ if $|s|_x \geqslant |t|_x$ for all $x \in \mathcal{V}$ and either

1. $w(s) > w(t)$
2. $w(s) = w(t)$ and either
   1. $\exists\, n > 0\ \exists\, x \in \mathcal{V}$ such that $s = f_n(\ldots(f_1(t))\ldots)$ and $t \in \mathcal{V}$

Definition

Knuth-Bendix order $>_{\text{kbo}}$ on terms:
$s >_{\text{kbo}} t$ if $|s|_x \geqslant |t|_x$ for all $x \in \mathcal{V}$ and either

1. $w(s) > w(t)$
2. $w(s) = w(t)$ and either
   1. $\exists\, n > 0\ \exists\, x \in \mathcal{V}$ such that $s = f_n(\ldots(f_1(t))\ldots)$ and $t \in \mathcal{V}$
   2. $s = f(s_1, \ldots, s_n)$ and $t = g(t_1, \ldots, t_m)$ and $f \sim g$ and $\exists\, i$
      $\forall\, j < i \quad s_j = t_j \qquad s_i >_{\text{kbo}} t_i$

Definition

Knuth-Bendix order $>_{\text{kbo}}$ on terms:
$s >_{\text{kbo}} t$ if $|s|_x \geqslant |t|_x$ for all $x \in \mathcal{V}$ and either

1. $w(s) > w(t)$
2. $w(s) = w(t)$ and either
   1. $\exists\, n > 0\ \exists\, x \in \mathcal{V}$ such that $s = f_n(\ldots(f_1(t))\ldots)$ and $t \in \mathcal{V}$
   2. $s = f(s_1, \ldots, s_n)$ and $t = g(t_1, \ldots, t_m)$ and $f \sim g$ and $\exists\, i$
      $\forall\, j < i \quad s_j = t_j \qquad s_i >_{\text{kbo}} t_i$
   3. $s = f(s_1, \ldots, s_n)$ and $t = g(t_1, \ldots, t_m)$ and $f > g$

Definition

Knuth-Bendix order $>_{\text{kbo}}$ on terms:
$s >_{\text{kbo}} t$ if $|s|_x \geqslant |t|_x$ for all $x \in \mathcal{V}$ and either

1. $w(s) > w(t)$
2. $w(s) = w(t)$ and either
   1. $\exists\, n > 0\ \exists\, x \in \mathcal{V}$ such that $s = f_n(\ldots(f_1(t))\ldots)$ and $t \in \mathcal{V}$
   2. $s = f(s_1, \ldots, s_n)$ and $t = g(t_1, \ldots, t_m)$ and $f \sim g$ and $\exists\, i$
      $\forall\, j < i \quad s_j = t_j \qquad s_i >_{\text{kbo}} t_i$
   3. $s = f(s_1, \ldots, s_n)$ and $t = g(t_1, \ldots, t_m)$ and $f > g$

Theorem

TRS $\mathcal{R}$ is KBO terminating if

$\exists$ *quasi-precedence* $\gtrsim$ $\exists$ *admissible weight function* (w, w_0)

such that $l >_{\text{kbo}} r$ *for all* $l \to r \in \mathcal{R}$

Example TRS/SK90_2.42

$$\begin{aligned}
\mathsf{flat}(\mathsf{nil}) &\to \mathsf{nil} \\
\mathsf{flat}(\mathsf{unit}(x)) &\to \mathsf{flat}(x) \\
\mathsf{flat}(x +\!\!+ y) &\to \mathsf{flat}(x) +\!\!+ \mathsf{flat}(y) \\
\mathsf{flat}(\mathsf{unit}(x) +\!\!+ y) &\to \mathsf{flat}(x) +\!\!+ \mathsf{flat}(y) \\
\mathsf{flat}(\mathsf{flat}(x)) &\to \mathsf{flat}(x) \\
x +\!\!+ \mathsf{nil} &\to x
\end{aligned}$$

$$\begin{aligned}
\mathsf{rev}(\mathsf{nil}) &\to \mathsf{nil} \\
\mathsf{rev}(\mathsf{unit}(x)) &\to \mathsf{unit}(x) \\
\mathsf{rev}(x +\!\!+ y) &\to \mathsf{rev}(y) +\!\!+ \mathsf{rev}(x) \\
\mathsf{rev}(\mathsf{rev}(x)) &\to x \\
(x +\!\!+ y) +\!\!+ z &\to x +\!\!+ (y +\!\!+ z) \\
\mathsf{nil} +\!\!+ y &\to y
\end{aligned}$$

Example TRS/SK90_2.42

$$\begin{aligned}
\mathsf{flat}(\mathsf{nil}) &\to \mathsf{nil} \\
\mathsf{flat}(\mathsf{unit}(x)) &\to \mathsf{flat}(x) \\
\mathsf{flat}(x +\!\!+ y) &\to \mathsf{flat}(x) +\!\!+ \mathsf{flat}(y) \\
\mathsf{flat}(\mathsf{unit}(x) +\!\!+ y) &\to \mathsf{flat}(x) +\!\!+ \mathsf{flat}(y) \\
\mathsf{flat}(\mathsf{flat}(x)) &\to \mathsf{flat}(x) \\
x +\!\!+ \mathsf{nil} &\to x
\end{aligned}$$

$$\begin{aligned}
\mathsf{rev}(\mathsf{nil}) &\to \mathsf{nil} \\
\mathsf{rev}(\mathsf{unit}(x)) &\to \mathsf{unit}(x) \\
\mathsf{rev}(x +\!\!+ y) &\to \mathsf{rev}(y) +\!\!+ \mathsf{rev}(x) \\
\mathsf{rev}(\mathsf{rev}(x)) &\to x \\
(x +\!\!+ y) +\!\!+ z &\to x +\!\!+ (y +\!\!+ z) \\
\mathsf{nil} +\!\!+ y &\to y
\end{aligned}$$

$$w(\mathsf{flat}) = w(\mathsf{rev}) = w(+\!\!+) = 0$$

$$w(\mathsf{unit}) = w(\mathsf{nil}) = w_0 = 1$$

$$\mathsf{flat} \sim \mathsf{rev} > \mathsf{unit} > +\!\!+ > \mathsf{nil}$$

- Motivation
- Knuth-Bendix Order
- Encodings
- Experimental Results
- Concluding Remarks

Aim

define propositional formula $\mathsf{SAT}(\mathcal{R})$ depending on TRS $\mathcal{R}$ such that

$$\models \mathsf{SAT}(\mathcal{R}) \implies \mathcal{R} \text{ is KBO terminating}$$
$$\not\models \mathsf{SAT}(\mathcal{R}) \implies \mathcal{R} \text{ is not KBO terminating}$$

Aim

define propositional formula $\mathsf{SAT}(\mathcal{R})$ depending on TRS $\mathcal{R}$ such that

$$\begin{aligned} \models \mathsf{SAT}(\mathcal{R}) &\implies \mathcal{R} \text{ is KBO terminating} \\ \not\models \mathsf{SAT}(\mathcal{R}) &\implies \mathcal{R} \text{ is not KBO terminating} \end{aligned}$$

Problem

weight function $w \colon \mathcal{F} \to \mathbb{N}$

Aim

define propositional formula $\mathrm{SAT}(\mathcal{R})$ depending on TRS $\mathcal{R}$ such that

$$\begin{aligned} \models \mathrm{SAT}(\mathcal{R}) &\implies \mathcal{R} \text{ is KBO terminating} \\ \not\models \mathrm{SAT}(\mathcal{R}) &\implies \mathcal{R} \text{ is not KBO terminating} \end{aligned}$$

Problem

weight function $w \colon \mathcal{F} \to \mathbb{N}$

Solution

restrict range of weight function to $\{0, \ldots, 2^k - 1\}$ (k bits)

Aim

define propositional formula $\mathrm{SAT}(\mathcal{R})$ depending on TRS $\mathcal{R}$ such that

$$\begin{aligned} \models \mathrm{SAT}(\mathcal{R}) &\implies \mathcal{R} \text{ is KBO terminating} \\ \not\models \mathrm{SAT}(\mathcal{R}) &\implies \mathcal{R} \text{ is not KBO terminating} \end{aligned}$$

Problem

weight function $w \colon \mathcal{F} \to \mathbb{N}$

Solution

restrict range of weight function to $\{0, \ldots, 2^k - 1\}$ (k bits)

Revised Aim

define propositional formula $\mathrm{SAT}_k(\mathcal{R})$ such that

$$\begin{aligned} \models \mathrm{SAT}_k(\mathcal{R}) &\implies \mathcal{R} \text{ is KBO terminating} \\ \not\models \mathrm{SAT}_k(\mathcal{R}) &\implies \mathcal{R} \text{ is not KBO terminating in } k \text{ bits} \end{aligned}$$

Definition (Codish, Lagoon & Stuckey 2006)

- $\mathbf{a} = \langle a_k, \ldots, a_1 \rangle$

Definition (Codish, Lagoon & Stuckey 2006)

- $\mathbf{a} = \langle a_k, \ldots, a_1 \rangle$
- $\ulcorner \mathbf{a} >_j \mathbf{b} \urcorner = \begin{cases} a_1 \wedge \neg b_1 & \text{if } j = 1 \\ a_j \wedge \neg b_j \vee (a_j \leftrightarrow b_j) \wedge \ulcorner \mathbf{a} >_{j-1} \mathbf{b} \urcorner & \text{if } j > 1 \end{cases}$

Definition (Codish, Lagoon & Stuckey 2006)

- $\mathbf{a} = \langle a_k, \ldots, a_1 \rangle$
- $\ulcorner \mathbf{a} >_j \mathbf{b} \urcorner = \begin{cases} a_1 \wedge \neg b_1 & \text{if } j = 1 \\ a_j \wedge \neg b_j \vee (a_j \leftrightarrow b_j) \wedge \ulcorner \mathbf{a} >_{j-1} \mathbf{b} \urcorner & \text{if } j > 1 \end{cases}$
- $\ulcorner \mathbf{a} > \mathbf{b} \urcorner = \ulcorner \mathbf{a} >_k \mathbf{b} \urcorner$

Definition (Codish, Lagoon & Stuckey 2006)

- $\mathbf{a} = \langle a_k, \ldots, a_1 \rangle$
- $\ulcorner \mathbf{a} >_j \mathbf{b} \urcorner = \begin{cases} a_1 \wedge \neg b_1 & \text{if } j = 1 \\ a_j \wedge \neg b_j \vee (a_j \leftrightarrow b_j) \wedge \ulcorner \mathbf{a} >_{j-1} \mathbf{b} \urcorner & \text{if } j > 1 \end{cases}$
- $\ulcorner \mathbf{a} > \mathbf{b} \urcorner = \ulcorner \mathbf{a} >_k \mathbf{b} \urcorner$
- $\ulcorner \mathbf{a} = \mathbf{b} \urcorner = \bigwedge_{i=1}^{k} (a_i \leftrightarrow b_i)$

Definition (Codish, Lagoon & Stuckey 2006)

- $\mathbf{a} = \langle a_k, \ldots, a_1 \rangle$
- $\ulcorner \mathbf{a} >_j \mathbf{b} \urcorner = \begin{cases} a_1 \wedge \neg b_1 & \text{if } j = 1 \\ a_j \wedge \neg b_j \vee (a_j \leftrightarrow b_j) \wedge \ulcorner \mathbf{a} >_{j-1} \mathbf{b} \urcorner & \text{if } j > 1 \end{cases}$
- $\ulcorner \mathbf{a} > \mathbf{b} \urcorner = \ulcorner \mathbf{a} >_k \mathbf{b} \urcorner$
- $\ulcorner \mathbf{a} = \mathbf{b} \urcorner = \bigwedge_{i=1}^{k} (a_i \leftrightarrow b_i)$
- $\ulcorner \mathbf{a} \geqslant \mathbf{b} \urcorner = \ulcorner \mathbf{a} > \mathbf{b} \urcorner \vee \ulcorner \mathbf{a} = \mathbf{b} \urcorner$

Definition

$$\ulcorner (\mathbf{a}, \varphi) + (\mathbf{b}, \psi) \urcorner \;=\; (\mathbf{s}, \varphi \wedge \psi \wedge \gamma \wedge \sigma)$$

with

$$\gamma \;=\; \neg c_k \wedge \neg c_0 \wedge \bigwedge_{i=1}^{k} \big(c_i \leftrightarrow ((a_i \wedge b_i) \vee (a_i \wedge c_{i-1}) \vee (b_i \wedge c_{i-1}))\big)$$

and

$$\sigma \;=\; \bigwedge_{i=1}^{k} \big(s_i \leftrightarrow (a_i \oplus b_i \oplus c_{i-1})\big)$$

Definition

$$\ulcorner (\mathbf{a}, \varphi) + (\mathbf{b}, \psi) \urcorner \;=\; (\mathbf{s}, \varphi \wedge \psi \wedge \gamma \wedge \sigma)$$

with

$$\gamma \;=\; \neg c_k \wedge \neg c_0 \wedge \bigwedge_{i=1}^{k} \big(c_i \leftrightarrow ((a_i \wedge b_i) \vee (a_i \wedge c_{i-1}) \vee (b_i \wedge c_{i-1}))\big)$$

and

$$\sigma \;=\; \bigwedge_{i=1}^{k} \big(s_i \leftrightarrow (a_i \oplus b_i \oplus c_{i-1})\big)$$

- fresh variables c_i $(0 \leqslant i \leqslant k)$ for carry and s_i $(1 \leqslant i \leqslant k)$ for sum

Definition

$$\ulcorner (\mathbf{a}, \varphi) + (\mathbf{b}, \psi) \urcorner \;=\; (\mathbf{s}, \varphi \wedge \psi \wedge \gamma \wedge \sigma)$$

with

$$\gamma \;=\; \neg c_k \wedge \neg c_0 \wedge \bigwedge_{i=1}^{k} \big(c_i \leftrightarrow ((a_i \wedge b_i) \vee (a_i \wedge c_{i-1}) \vee (b_i \wedge c_{i-1}))\big)$$

and

$$\sigma \;=\; \bigwedge_{i=1}^{k} \big(s_i \leftrightarrow (a_i \oplus b_i \oplus c_{i-1})\big)$$

- fresh variables c_i $(0 \leqslant i \leqslant k)$ for carry and s_i $(1 \leqslant i \leqslant k)$ for sum
- $\neg c_k$ prevents overflow

Definition

$$\ulcorner (\mathbf{a}, \varphi) + (\mathbf{b}, \psi) \urcorner \;=\; (\mathbf{s}, \varphi \wedge \psi \wedge \gamma \wedge \sigma)$$

with

$$\gamma \;=\; \neg c_k \wedge \neg c_0 \wedge \bigwedge_{i=1}^{k} \big(c_i \leftrightarrow ((a_i \wedge b_i) \vee (a_i \wedge c_{i-1}) \vee (b_i \wedge c_{i-1}))\big)$$

and

$$\sigma \;=\; \bigwedge_{i=1}^{k} \big(s_i \leftrightarrow (a_i \oplus b_i \oplus c_{i-1})\big)$$

- fresh variables c_i $(0 \leqslant i \leqslant k)$ for carry and s_i $(1 \leqslant i \leqslant k)$ for sum
- $\neg c_k$ prevents overflow
- $\oplus$ denotes exclusive or

Definition

- weight of term t

$$\mathsf{w}(t) = \begin{cases} (\mathbf{w_0}, \top) & \text{if } t \in \mathcal{V} \\ \ulcorner (\mathbf{f}, \top) + \sum_{i=1}^{n} \mathsf{w}(t_i) \urcorner & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

Definition

- weight of term t

$$\mathsf{w}(t) = \begin{cases} (\mathbf{w_0}, \top) & \text{if } t \in \mathcal{V} \\ \ulcorner\, (\mathbf{f}, \top) + \sum_{i=1}^{n} \mathsf{w}(t_i) \,\urcorner & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

- comparing weights

$$\ulcorner\, (\mathbf{f}, \varphi) > (\mathbf{g}, \psi) \,\urcorner = \ulcorner\, \mathbf{f} > \mathbf{g} \,\urcorner \wedge \varphi \wedge \psi$$

Definition

- weight of term t

$$\mathsf{w}(t) \;=\; \begin{cases} (\mathbf{w_0}, \top) & \text{if } t \in \mathcal{V} \\ \ulcorner\, (\mathbf{f}, \top) + \sum_{i=1}^{n} \mathsf{w}(t_i) \,\urcorner & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

- comparing weights

$$\ulcorner\, (\mathbf{f}, \varphi) > (\mathbf{g}, \psi) \,\urcorner \;=\; \ulcorner\, \mathbf{f} > \mathbf{g} \,\urcorner \wedge \varphi \wedge \psi$$

- admissibility condition

$$\mathsf{ADM}(\mathcal{F}) \;=\; \ulcorner\, \mathbf{w_0} > \mathbf{0} \,\urcorner \;\wedge \bigwedge_{c \in \mathcal{F}^{(0)}} \ulcorner\, \mathbf{c} \geqslant \mathbf{w_0} \,\urcorner \;\wedge$$

$$\bigwedge_{f \in \mathcal{F}^{(1)}} \big(\ulcorner\, \mathbf{f} = \mathbf{0} \,\urcorner \;\rightarrow \bigwedge_{g \in \mathcal{F}} (\mathsf{X}_{fg} \vee \mathsf{Y}_{fg}) \big)$$

Quasi-precedence

symbol-based approach Codish, Lagoon & Stuckey 2006

interpret function symbols in $\{0, \ldots, |\mathcal{F}| - 1\}$

Quasi-precedence

symbol-based approach Codish, Lagoon & Stuckey 2006

interpret function symbols in $\{0, \ldots, |\mathcal{F}| - 1\}$

Propositionally ($l := \lceil log_2(|\mathcal{F}|) \rceil$)

Quasi-precedence

symbol-based approach Codish, Lagoon & Stuckey 2006

interpret function symbols in $\{0, \ldots, |\mathcal{F}| - 1\}$

Propositionally ($l := \lceil log_2(|\mathcal{F}|) \rceil$)

interpret function symbols in $\{\langle \overbrace{0, \ldots, 0}^{l} \rangle, \ldots, \langle \overbrace{1, \ldots, 1}^{l} \rangle\}$

Quasi-precedence

symbol-based approach Codish, Lagoon & Stuckey 2006

interpret function symbols in $\{ 0, \ldots, |\mathcal{F}| - 1 \}$

Propositionally ($l := \lceil log_2(|\mathcal{F}|) \rceil$)

interpret function symbols in $\{\overbrace{\langle 0, \ldots, 0 \rangle}^{l}, \ldots, \overbrace{\langle 1, \ldots, 1 \rangle}^{l}\}$

$$\mathrm{PREC}(\mathcal{F}) = \bigwedge_{f,\, g \in \mathcal{F}} ((X_{fg} \to \ulcorner \mathbf{f}' >_l \mathbf{g}' \urcorner) \wedge (Y_{fg} \to \ulcorner \mathbf{f}' =_l \mathbf{g}' \urcorner))$$

Definition

$$\mathsf{SAT}(s >_{\mathsf{kbo}} t) \;=\; \begin{cases} \bot & \text{if } s \in \mathcal{V} \text{ or } s = t \text{ or } \exists\, x\ |s|_x < |t|_x \\ \ulcorner\, \mathsf{w}(s) > \mathsf{w}(t) \,\urcorner \;\vee\; \ulcorner\, \mathsf{w}(s) = \mathsf{w}(t) \,\urcorner \wedge \mathsf{SAT}(s >'_{\mathsf{kbo}} t) & \\ & \text{otherwise} \end{cases}$$

Definition

$$\mathsf{SAT}(s >_{\mathsf{kbo}} t) = \begin{cases} \bot & \text{if } s \in \mathcal{V} \text{ or } s = t \text{ or } \exists\, x\ |s|_x < |t|_x \\ \ulcorner\, \mathsf{w}(s) > \mathsf{w}(t) \,\urcorner \ \vee\ \ulcorner\, \mathsf{w}(s) = \mathsf{w}(t) \,\urcorner \wedge \mathsf{SAT}(s >'_{\mathsf{kbo}} t) & \\ & \text{otherwise} \end{cases}$$

with

$$\mathsf{SAT}(s >'_{\mathsf{kbo}} t) = \begin{cases} \top & \text{if } s = f_n(\ldots(f_1(t))\ldots),\ t \in \mathcal{V},\ n > 0 \\ \mathsf{SAT}(s_i >_{\mathsf{kbo}} t_i) & \text{if } s = f(s_1, \ldots, s_n),\ t = f(t_1, \ldots, t_n) \\ \mathsf{X}_{fg} \ \vee\ \mathsf{Y}_{fg} \wedge \mathsf{SAT}(s_i >_{\mathsf{kbo}} t_i) & \\ & \text{if } s = f(s_1, \ldots, s_n),\ t = g(t_1, \ldots, t_m) \end{cases}$$

where i is least $1 \leqslant j \leqslant \min\{m, n\}$ with $s_j \neq t_j$

Theorem

TRS $\mathcal{R}$ is KBO terminating if

$$\bigwedge_{l \to r \in \mathcal{R}} \mathsf{SAT}(\, l >_{\mathsf{kbo}} r \,)$$

is satisfiable

Theorem

TRS $\mathcal{R}$ is KBO terminating if

$$\mathsf{PREC}(\mathcal{F}) \ \wedge \bigwedge_{l \to r \in \mathcal{R}} \mathsf{SAT}(\, l >_{\mathsf{kbo}} r \,)$$

is satisfiable

Theorem

TRS $\mathcal{R}$ is KBO terminating if

$$\mathsf{ADM}(\mathcal{F}) \ \wedge \ \mathsf{PREC}(\mathcal{F}) \ \wedge \ \bigwedge_{l \to r \in \mathcal{R}} \mathsf{SAT}(\, l >_{\mathsf{kbo}} r \,)$$

is satisfiable

Theorem

TRS $\mathcal{R}$ is KBO terminating if

$$\mathsf{SAT}_k(\mathcal{R}) \;=\; \mathsf{ADM}(\mathcal{F}) \;\wedge\; \mathsf{PREC}(\mathcal{F}) \;\wedge \bigwedge_{l \to r \in \mathcal{R}} \mathsf{SAT}(\, l >_{\mathsf{kbo}} r \,)$$

is satisfiable

Theorem

TRS $\mathcal{R}$ is KBO terminating if

$$\mathsf{SAT}_k(\mathcal{R}) \;=\; \mathsf{ADM}(\mathcal{F}) \;\wedge\; \mathsf{PREC}(\mathcal{F}) \;\wedge \bigwedge_{l \to r \in \mathcal{R}} \mathsf{SAT}(\, l >_{\mathsf{kbo}} r \,)$$

is satisfiable

Remark

satisfying assignment encodes quasi-precedence and weight function

Pseudo Boolean Constraints (PBC)

What are PBC?

Pseudo Boolean Constraints (PBC)

What are PBC?

- $-1 * x_1 + 4 * x_2 - 2 * x_5 \geqslant 3;$ (constraint)

Pseudo Boolean Constraints (PBC)

What are PBC?

- $-1 * x_1 + 4 * x_2 - 2 * x_5 \geqslant 3;$ (constraint)
- $123456789012345678 90 * x_4 + 4 * x_3 \geqslant 10;$ (constraint)

Pseudo Boolean Constraints (PBC)

What are PBC?

- $-1 * x_1 + 4 * x_2 - 2 * x_5 \geqslant 3;$ (constraint)
- $12345678901234567890 * x_4 + 4 * x_3 \geqslant 10;$ (constraint)
- $2 * x_2 + 3 * x_4 = 5;$ (constraint)

Pseudo Boolean Constraints (PBC)

What are PBC?

- $min : 1 * x_2 - 1 * x_3;$ (goal function)
- $-1 * x_1 + 4 * x_2 - 2 * x_5 \geqslant 3;$ (constraint)
- $12345678901234567890 * x_4 + 4 * x_3 \geqslant 10;$ (constraint)
- $2 * x_2 + 3 * x_4 = 5;$ (constraint)

Pseudo Boolean Constraints (PBC)

What are PBC?

- $min : 1 * x_2 - 1 * x_3;$ (goal function)
- $-1 * x_1 + 4 * x_2 - 2 * x_5 \geqslant 3;$ (constraint)
- $12345678901234567890 * x_4 + 4 * x_3 \geqslant 10;$ (constraint)
- $2 * x_2 + 3 * x_4 = 5;$ (constraint)

Why PBC?

Pseudo Boolean Constraints (PBC)

What are PBC?

- $min : 1 * x_2 - 1 * x_3;$ (goal function)
- $-1 * x_1 + 4 * x_2 - 2 * x_5 \geqslant 3;$ (constraint)
- $12345678901234567890 * x_4 + 4 * x_3 \geqslant 10;$ (constraint)
- $2 * x_2 + 3 * x_4 = 5;$ (constraint)

Why PBC?

- addition and comparisons for free

Pseudo Boolean Constraints (PBC)

What are PBC?

- $min : 1 * x_2 - 1 * x_3;$ (goal function)
- $-1 * x_1 + 4 * x_2 - 2 * x_5 \geqslant 3;$ (constraint)
- $12345678901234567890 * x_4 + 4 * x_3 \geqslant 10;$ (constraint)
- $2 * x_2 + 3 * x_4 = 5;$ (constraint)

Why PBC?

- addition and comparisons for free
- less implementation work

Pseudo Boolean Constraints (PBC)

What are PBC?

- $min : 1 * x_2 - 1 * x_3;$ (goal function)
- $-1 * x_1 + 4 * x_2 - 2 * x_5 \geqslant 3;$ (constraint)
- $12345678901234567890 * x_4 + 4 * x_3 \geqslant 10;$ (constraint)
- $2 * x_2 + 3 * x_4 = 5;$ (constraint)

Why PBC?

- addition and comparisons for free
- less implementation work
- deals with arbitrary large natural numbers

Pseudo Boolean Constraints (PBC)

What are PBC?

- $min : 1 * x_2 - 1 * x_3;$ (goal function)
- $-1 * x_1 + 4 * x_2 - 2 * x_5 \geqslant 3;$ (constraint)
- $12345678901234567890 * x_4 + 4 * x_3 \geqslant 10;$ (constraint)
- $2 * x_2 + 3 * x_4 = 5;$ (constraint)

Why PBC?

- addition and comparisons for free
- less implementation work
- deals with arbitrary large natural numbers
- faster

Pseudo Boolean Constraints (PBC)

What are PBC?

- $min : 1 * x_2 - 1 * x_3;$ (goal function)
- $-1 * x_1 + 4 * x_2 - 2 * x_5 \geqslant 3;$ (constraint)
- $12345678901234567890 * x_4 + 4 * x_3 \geqslant 10;$ (constraint)
- $2 * x_2 + 3 * x_4 = 5;$ (constraint)

Why PBC?

- addition and comparisons for free
- less implementation work
- deals with arbitrary large natural numbers
- faster ?

Encoding KBO

Admissibility

$$\begin{aligned} \overline{w}_0 &\geqslant 1; \\ \overline{w}(c) - \overline{w}_0 &\geqslant 0; && \forall \text{ constants } c \in \mathcal{F} \\ n * \overline{w}(f) + \sum_{g \in \mathcal{F}} (X_{fg} + Y_{fg}) &\geqslant n; && \forall \text{ unary } f \in \mathcal{F}, n := |\mathcal{F}| \end{aligned}$$

where $\overline{w}(f) = 2^{k-1} * f_k + \cdots + 2^0 * f_1$ and $\overline{w}_0 = 2^{k-1} * w_{0k} + \cdots + 2^0 * w_{01}$

Encoding KBO

Admissibility

$$\begin{aligned} \overline{w}_0 &\geqslant 1; \\ \overline{w}(c) - \overline{w}_0 &\geqslant 0; && \forall \text{ constants } c \in \mathcal{F} \\ n * \overline{w}(f) + \sum_{g \in \mathcal{F}} (X_{fg} + Y_{fg}) &\geqslant n; && \forall \text{ unary } f \in \mathcal{F}, n := |\mathcal{F}| \end{aligned}$$

where $\overline{w}(f) = 2^{k-1} * f_k + \cdots + 2^0 * f_1$ and $\overline{w}_0 = 2^{k-1} * w_{0k} + \cdots + 2^0 * w_{01}$

Precedence

$$2 * X_{fg} + Y_{fg} + Y_{gf} + 2 * Z_{fg} = 2$$

Encoding KBO

Admissibility

$$\begin{aligned} \overline{w}_0 &\geqslant 1; \\ \overline{w}(c) - \overline{w}_0 &\geqslant 0; && \forall \text{ constants } c \in \mathcal{F} \\ n * \overline{w}(f) + \sum_{g \in \mathcal{F}} (X_{fg} + Y_{fg}) &\geqslant n; && \forall \text{ unary } f \in \mathcal{F}, n := |\mathcal{F}| \end{aligned}$$

where $\overline{w}(f) = 2^{k-1} * f_k + \cdots + 2^0 * f_1$ and $\overline{w}_0 = 2^{k-1} * w_{0k} + \cdots + 2^0 * w_{01}$

Precedence

$$\begin{aligned} 2 * X_{fg} + Y_{fg} + Y_{gf} + 2 * Z_{fg} &= 2 \\ -X_{fg} + 2^l * Y_{fg} + 2^l * Z_{fg} + i(f) - i(g) &\geqslant 0 \end{aligned}$$

where $l := \lceil log_2(|\mathcal{F}|) \rceil$ and $i(f) = 2^{l-1} * f'_l + \cdots + 2^0 * f'_1$

Encoding KBO

Admissibility

$$\begin{aligned} \overline{w}_0 &\geqslant 1; \\ \overline{w}(c) - \overline{w}_0 &\geqslant 0; && \forall \text{ constants } c \in \mathcal{F} \\ n * \overline{w}(f) + \sum_{g \in \mathcal{F}} (X_{fg} + Y_{fg}) &\geqslant n; && \forall \text{ unary } f \in \mathcal{F}, n := |\mathcal{F}| \end{aligned}$$

where $\overline{w}(f) = 2^{k-1} * f_k + \cdots + 2^0 * f_1$ and $\overline{w}_0 = 2^{k-1} * w_{0k} + \cdots + 2^0 * w_{01}$

Precedence

$$\begin{aligned} 2 * X_{fg} + Y_{fg} + Y_{gf} + 2 * Z_{fg} &= 2 \\ -X_{fg} + i(f) - i(g) &\geqslant 0 \end{aligned}$$

where $l := \lceil log_2(|\mathcal{F}|) \rceil$ and $i(f) = 2^{l-1} * f'_l + \cdots + 2^0 * f'_1$

Encoding KBO

Admissibility

$$\begin{aligned} \overline{w}_0 &\geqslant 1; \\ \overline{w}(c) - \overline{w}_0 &\geqslant 0; && \forall \text{ constants } c \in \mathcal{F} \\ n * \overline{w}(f) + \sum_{g \in \mathcal{F}} (X_{fg} + Y_{fg}) &\geqslant n; && \forall \text{ unary } f \in \mathcal{F}, n := |\mathcal{F}| \end{aligned}$$

where $\overline{w}(f) = 2^{k-1} * f_k + \cdots + 2^0 * f_1$ and $\overline{w}_0 = 2^{k-1} * w_{0k} + \cdots + 2^0 * w_{01}$

Precedence

$$\begin{aligned} 2 * X_{fg} + Y_{fg} + Y_{gf} + 2 * Z_{fg} &= 2 \\ -X_{fg} + 2^l * Y_{fg} + 2^l * Z_{fg} + i(f) - i(g) &\geqslant 0 \end{aligned}$$

where $l := \lceil log_2(|\mathcal{F}|) \rceil$ and $i(f) = 2^{l-1} * f'_l + \cdots + 2^0 * f'_1$

Encoding KBO

Admissibility

$$\begin{aligned} \overline{w}_0 &\geqslant 1; \\ \overline{w}(c) - \overline{w}_0 &\geqslant 0; && \forall \text{ constants } c \in \mathcal{F} \\ n * \overline{w}(f) + \sum_{g \in \mathcal{F}} (X_{fg} + Y_{fg}) &\geqslant n; && \forall \text{ unary } f \in \mathcal{F}, n := |\mathcal{F}| \end{aligned}$$

where $\overline{w}(f) = 2^{k-1} * f_k + \cdots + 2^0 * f_1$ and $\overline{w}_0 = 2^{k-1} * w_{0k} + \cdots + 2^0 * w_{01}$

Precedence

$$\begin{aligned} 2 * X_{fg} + Y_{fg} + Y_{gf} + 2 * Z_{fg} &= 2 \\ -X_{fg} + 2^l * Y_{fg} + 2^l * Z_{fg} + i(f) - i(g) &\geqslant 0 \\ 2^l * X_{fg} + Y_{fg} + 2^l * Z_{fg} + i(f) - i(g) &\geqslant 1 \end{aligned}$$

where $l := \lceil log_2(|\mathcal{F}|) \rceil$ and $i(f) = 2^{l-1} * f'_l + \cdots + 2^0 * f'_1$

Encoding KBO (cont'd)

$$m := 2^k * |t|$$

$$\mathrm{PBC}(s >_{\mathrm{kbo}} t) = \begin{cases} KBO_{st} = 0 & \text{if } \exists x : |s|_x < |t|_x \\ -(m+1) * KBO_{st} + \overline{w}(s) - \overline{w}(t) + KBO'_{st} \geqslant -m; & \\ \mathrm{PBC}(s >'_{kbo} t) & \text{otherwise} \end{cases}$$

with

Encoding KBO (cont'd)

$m := 2^k * |t|$

$$\mathrm{PBC}(s >_{\mathrm{kbo}} t) = \begin{cases} KBO_{st} = 0 & \text{if } \exists x : |s|_x < |t|_x \\ -(m+1) * KBO_{st} + \overline{w}(s) - \overline{w}(t) + KBO'_{st} \geqslant -m; & \\ \mathrm{PBC}(s >'_{kbo} t) & \text{otherwise} \end{cases}$$

with $\mathrm{PBC}(s >'_{\mathrm{kbo}} t) = \mathrm{PBC}(s_i >_{\mathrm{kbo}} t_i)$;

$$\begin{cases} KBO_{st} = 1 & \text{if } s = f_n(\ldots(f_1(t))\ldots) \\ -KBO'_{st} + KBO_{s_i t_i} \geqslant 0 & \text{if } f = g \\ -2 * KBO'_{st} + 2 * X_{fg} + Y_{fg} + KBO_{s_i t_i} \geqslant 0 & \text{if } f \neq g \\ KBO_{st} = 0 & \text{otherwise} \end{cases}$$

where i is least $1 \leqslant j \leqslant \min\{m, n\}$ with $s_j \neq t_j$

Satisfiability Modulo Theories (SMT)

Satisfiability Modulo Theories (SMT)

What is SMT?

Satisfiability Modulo Theories (SMT)

What is SMT?

- satisfiability modulo theories (1^{st} order)

Satisfiability Modulo Theories (SMT)

What is SMT?

- satisfiability modulo theories (1^{st} order)
- real numbers, integers, bitvectors, lists, arrays, ...

Satisfiability Modulo Theories (SMT)

What is SMT?

- satisfiability modulo theories (1^{st} order)
- real numbers, integers, bitvectors, lists, arrays, ...

What is YICES?

Satisfiability Modulo Theories (SMT)

What is SMT?

- satisfiability modulo theories (1^{st} order)
- real numbers, integers, bitvectors, lists, arrays, ...

What is YICES?

- an SMT-solver

Satisfiability Modulo Theories (SMT)

What is SMT?

- satisfiability modulo theories (1^{st} order)
- real numbers, integers, bitvectors, lists, arrays, ...

What is YICES?

- an SMT-solver
- supports SMT-LIB format (cryptic)

Satisfiability Modulo Theories (SMT)

What is SMT?

- satisfiability modulo theories (1^{st} order)
- real numbers, integers, bitvectors, lists, arrays, ...

What is YICES?

- an SMT-solver
- supports SMT-LIB format (cryptic)
- supports own format (easy, user need not specify theory)

Example

```
;; part of f(x) -> g(x)
(define wf::nat)
(define wg::nat)
(define pf::nat)
(define pg::nat)

(assert
 (or
  (> wf wg)
  (and
   (= wf wg)
   (or (> pf pg) (and (= pf pg) false)))))
(check)
```

Example

```
;; part of f(x) -> g(x)
(define wf::nat)
(define wg::nat)
(define pf::nat)
(define pg::nat)

(assert
 (or
  (> wf wg)
  (and
   (= wf wg)
   (or (> pf pg) (and (= pf pg) false)))))
(check)
```

Encoding

There is no encoding!

- Motivation
- Knuth-Bendix Order
- Encodings
- Experimental Results
- Concluding Remarks

KBO Experiments

1358 TRSs in version 4.0 of TPDB

† strict precedence

method(# bits)	total time	# successes	# timeouts (60s)
$\mathsf{T}_{\mathsf{T}}\mathsf{T}^{\dagger}$	489	101	1

KBO Experiments

1358 TRSs in version 4.0 of TPDB

† strict precedence

method(# bits)	total time	# successes	# timeouts (60s)
$T_T T^\dagger$	489	101	1
sat/pbc(2)	449/285	92/102	2/3

KBO Experiments

1358 TRSs in version 4.0 of TPDB

† strict precedence

method(# bits)	total time	# successes	# timeouts (60s)
$\mathsf{T}_{\mathsf{T}}\mathsf{T}^{\dagger}$	489	101	1
sat/pbc(2)	449/285	92/102	2/3
sat/pbc(3)	471/160	107/107	2/0

KBO Experiments

1358 TRSs in version 4.0 of TPDB

† strict precedence

method(# bits)	total time	# successes	# timeouts (60s)
TTT†	489	101	1
sat/pbc(2)	449/285	92/102	2/3
sat/pbc(3)	471/160	107/107	2/0
sat/pbc(4)	467/201	108/108	2/1

KBO Experiments

1358 TRSs in version 4.0 of TPDB

† strict precedence

method(# bits)	total time	# successes	# timeouts (60s)
$\mathsf{T}_\mathsf{T}\mathsf{T}^\dagger$	489	101	1
sat/pbc(2)	449/285	92/102	2/3
sat/pbc(3)	471/160	107/107	2/0
sat/pbc(4)	467/201	108/108	2/1
smt	33	108	0

Example TRS/various_21

$$
\begin{aligned}
\mathsf{p1} + \mathsf{p1} &\to \mathsf{p2} \\
\mathsf{p2} + \mathsf{p1} &\to \mathsf{p1} + \mathsf{p2} \\
\mathsf{p5} + \mathsf{p1} &\to \mathsf{p1} + \mathsf{p5} \\
\mathsf{p5} + \mathsf{p2} &\to \mathsf{p2} + \mathsf{p5} \\
\mathsf{p5} + \mathsf{p5} &\to \mathsf{p10} \\
\mathsf{p10} + \mathsf{p1} &\to \mathsf{p1} + \mathsf{p10} \\
\mathsf{p10} + \mathsf{p2} &\to \mathsf{p2} + \mathsf{p10} \\
\mathsf{p10} + \mathsf{p5} &\to \mathsf{p5} + \mathsf{p10} \\
\mathsf{p1} + (\mathsf{p2} + \mathsf{p2}) &\to \mathsf{p5} \\
\mathsf{p2} + (\mathsf{p2} + \mathsf{p2}) &\to \mathsf{p1} + \mathsf{p5}
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{p1} + (\mathsf{p1} + x) &\to \mathsf{p2} + x \\
\mathsf{p2} + (\mathsf{p1} + x) &\to \mathsf{p1} + (\mathsf{p2} + x) \\
\mathsf{p5} + (\mathsf{p1} + x) &\to \mathsf{p1} + (\mathsf{p5} + x) \\
\mathsf{p5} + (\mathsf{p2} + x) &\to \mathsf{p2} + (\mathsf{p5} + x) \\
\mathsf{p5} + (\mathsf{p5} + x) &\to \mathsf{p10} + x \\
\mathsf{p10} + (\mathsf{p1} + x) &\to \mathsf{p1} + (\mathsf{p10} + x) \\
\mathsf{p10} + (\mathsf{p2} + x) &\to \mathsf{p2} + (\mathsf{p10} + x) \\
\mathsf{p10} + (\mathsf{p5} + x) &\to \mathsf{p5} + (\mathsf{p10} + x) \\
\mathsf{p1} + (\mathsf{p2} + (\mathsf{p2} + x)) &\to \mathsf{p5} + x \\
\mathsf{p2} + (\mathsf{p2} + (\mathsf{p2} + x)) &\to \mathsf{p1} + (\mathsf{p5} + x) \\
(x + y) + z &\to x + (y + z)
\end{aligned}
$$

Example TRS/various_21

$$\begin{aligned}
\mathrm{p1} + \mathrm{p1} &\to \mathrm{p2} & \mathrm{p1} + (\mathrm{p1} + x) &\to \mathrm{p2} + x\\
\mathrm{p2} + \mathrm{p1} &\to \mathrm{p1} + \mathrm{p2} & \mathrm{p2} + (\mathrm{p1} + x) &\to \mathrm{p1} + (\mathrm{p2} + x)\\
\mathrm{p5} + \mathrm{p1} &\to \mathrm{p1} + \mathrm{p5} & \mathrm{p5} + (\mathrm{p1} + x) &\to \mathrm{p1} + (\mathrm{p5} + x)\\
\mathrm{p5} + \mathrm{p2} &\to \mathrm{p2} + \mathrm{p5} & \mathrm{p5} + (\mathrm{p2} + x) &\to \mathrm{p2} + (\mathrm{p5} + x)\\
\mathrm{p5} + \mathrm{p5} &\to \mathrm{p10} & \mathrm{p5} + (\mathrm{p5} + x) &\to \mathrm{p10} + x\\
\mathrm{p10} + \mathrm{p1} &\to \mathrm{p1} + \mathrm{p10} & \mathrm{p10} + (\mathrm{p1} + x) &\to \mathrm{p1} + (\mathrm{p10} + x)\\
\mathrm{p10} + \mathrm{p2} &\to \mathrm{p2} + \mathrm{p10} & \mathrm{p10} + (\mathrm{p2} + x) &\to \mathrm{p2} + (\mathrm{p10} + x)\\
\mathrm{p10} + \mathrm{p5} &\to \mathrm{p5} + \mathrm{p10} & \mathrm{p10} + (\mathrm{p5} + x) &\to \mathrm{p5} + (\mathrm{p10} + x)\\
\mathrm{p1} + (\mathrm{p2} + \mathrm{p2}) &\to \mathrm{p5} & \mathrm{p1} + (\mathrm{p2} + (\mathrm{p2} + x)) &\to \mathrm{p5} + x\\
\mathrm{p2} + (\mathrm{p2} + \mathrm{p2}) &\to \mathrm{p1} + \mathrm{p5} & \mathrm{p2} + (\mathrm{p2} + (\mathrm{p2} + x)) &\to \mathrm{p1} + (\mathrm{p5} + x)\\
& & (x + y) + z &\to x + (y + z)
\end{aligned}$$

$$w(\mathrm{p1}) = w(\mathrm{p2}) = 4 \qquad w(\mathrm{p5}) = 6 \qquad w(\mathrm{p10}) = 11 \qquad w(+) = 0$$

$$\mathrm{p2} > \mathrm{p1} \qquad \mathrm{p5} > \mathrm{p1} \qquad \mathrm{p10} > \mathrm{p1}$$

Example TRS/various_21

$$\begin{aligned}
\mathsf{p1} + \mathsf{p1} &\to \mathsf{p2} \\
\mathsf{p2} + \mathsf{p1} &\to \mathsf{p1} + \mathsf{p2} \\
\mathsf{p5} + \mathsf{p1} &\to \mathsf{p1} + \mathsf{p5} \\
\mathsf{p5} + \mathsf{p2} &\to \mathsf{p2} + \mathsf{p5} \\
\mathsf{p5} + \mathsf{p5} &\to \mathsf{p10} \\
\mathsf{p10} + \mathsf{p1} &\to \mathsf{p1} + \mathsf{p10} \\
\mathsf{p10} + \mathsf{p2} &\to \mathsf{p2} + \mathsf{p10} \\
\mathsf{p10} + \mathsf{p5} &\to \mathsf{p5} + \mathsf{p10} \\
\mathsf{p1} + (\mathsf{p2} + \mathsf{p2}) &\to \mathsf{p5} \\
\mathsf{p2} + (\mathsf{p2} + \mathsf{p2}) &\to \mathsf{p1} + \mathsf{p5}
\end{aligned}$$

$$\begin{aligned}
\mathsf{p1} + (\mathsf{p1} + x) &\to \mathsf{p2} + x \\
\mathsf{p2} + (\mathsf{p1} + x) &\to \mathsf{p1} + (\mathsf{p2} + x) \\
\mathsf{p5} + (\mathsf{p1} + x) &\to \mathsf{p1} + (\mathsf{p5} + x) \\
\mathsf{p5} + (\mathsf{p2} + x) &\to \mathsf{p2} + (\mathsf{p5} + x) \\
\mathsf{p5} + (\mathsf{p5} + x) &\to \mathsf{p10} + x \\
\mathsf{p10} + (\mathsf{p1} + x) &\to \mathsf{p1} + (\mathsf{p10} + x) \\
\mathsf{p10} + (\mathsf{p2} + x) &\to \mathsf{p2} + (\mathsf{p10} + x) \\
\mathsf{p10} + (\mathsf{p5} + x) &\to \mathsf{p5} + (\mathsf{p10} + x) \\
\mathsf{p1} + (\mathsf{p2} + (\mathsf{p2} + x)) &\to \mathsf{p5} + x \\
\mathsf{p2} + (\mathsf{p2} + (\mathsf{p2} + x)) &\to \mathsf{p1} + (\mathsf{p5} + x) \\
(x + y) + z &\to x + (y + z)
\end{aligned}$$

$$w(\mathsf{p1}) = w(\mathsf{p2}) = 4 \qquad w(\mathsf{p5}) = 6 \qquad w(\mathsf{p10}) = 11 \qquad w(+) = 0$$

$$\mathsf{p2} > \mathsf{p1} \qquad \mathsf{p5} > \mathsf{p1} \qquad \mathsf{p10} > \mathsf{p1}$$

sat(4)	0.19
smt	0.33

Example TRS/various_21

$$
\begin{aligned}
\mathsf{p1} + \mathsf{p1} &\to \mathsf{p2} \\
\mathsf{p2} + \mathsf{p1} &\to \mathsf{p1} + \mathsf{p2} \\
\mathsf{p5} + \mathsf{p1} &\to \mathsf{p1} + \mathsf{p5} \\
\mathsf{p5} + \mathsf{p2} &\to \mathsf{p2} + \mathsf{p5} \\
\mathsf{p5} + \mathsf{p5} &\to \mathsf{p10} \\
\mathsf{p10} + \mathsf{p1} &\to \mathsf{p1} + \mathsf{p10} \\
\mathsf{p10} + \mathsf{p2} &\to \mathsf{p2} + \mathsf{p10} \\
\mathsf{p10} + \mathsf{p5} &\to \mathsf{p5} + \mathsf{p10} \\
\mathsf{p1} + (\mathsf{p2} + \mathsf{p2}) &\to \mathsf{p5} \\
\mathsf{p2} + (\mathsf{p2} + \mathsf{p2}) &\to \mathsf{p1} + \mathsf{p5}
\end{aligned}
\qquad
\begin{aligned}
\mathsf{p1} + (\mathsf{p1} + x) &\to \mathsf{p2} + x \\
\mathsf{p2} + (\mathsf{p1} + x) &\to \mathsf{p1} + (\mathsf{p2} + x) \\
\mathsf{p5} + (\mathsf{p1} + x) &\to \mathsf{p1} + (\mathsf{p5} + x) \\
\mathsf{p5} + (\mathsf{p2} + x) &\to \mathsf{p2} + (\mathsf{p5} + x) \\
\mathsf{p5} + (\mathsf{p5} + x) &\to \mathsf{p10} + x \\
\mathsf{p10} + (\mathsf{p1} + x) &\to \mathsf{p1} + (\mathsf{p10} + x) \\
\mathsf{p10} + (\mathsf{p2} + x) &\to \mathsf{p2} + (\mathsf{p10} + x) \\
\mathsf{p10} + (\mathsf{p5} + x) &\to \mathsf{p5} + (\mathsf{p10} + x) \\
\mathsf{p1} + (\mathsf{p2} + (\mathsf{p2} + x)) &\to \mathsf{p5} + x \\
\mathsf{p2} + (\mathsf{p2} + (\mathsf{p2} + x)) &\to \mathsf{p1} + (\mathsf{p5} + x) \\
(x + y) + z &\to x + (y + z)
\end{aligned}
$$

$$w(\mathsf{p1}) = w(\mathsf{p2}) = 4 \qquad w(\mathsf{p5}) = 6 \qquad w(\mathsf{p10}) = 11 \qquad w(+) = 0$$

$$\mathsf{p2} > \mathsf{p1} \qquad \mathsf{p5} > \mathsf{p1} \qquad \mathsf{p10} > \mathsf{p1}$$

sat(4)	0.19
smt	0.33
TTT	4016.23

Example TRS/HM_t000

$$
\begin{array}{llll}
0+0 \to 0 & 1+0 \to 1 & \cdots & 9+0 \to 9 \\
0+1 \to 1 & 1+1 \to 2 & \cdots & 9+1 \to 1:0 \\
0+2 \to 2 & 1+2 \to 3 & \cdots & 9+2 \to 1:1 \\
\vdots & \vdots & & \vdots \\
0+8 \to 8 & 1+8 \to 9 & \cdots & 9+8 \to 1:7 \\
0+9 \to 9 & 1+9 \to 1:0 & \cdots & 9+9 \to 1:8 \\
x+(y:z) \to y:(x+z) & & & 0:x \to x \\
(x:y)+z \to x:(y+z) & & & x:(y:z) \to (x+y):z
\end{array}
$$

Example TRS/HM_t000

$$
\begin{array}{llll}
0+0 \to 0 & 1+0 \to 1 & \cdots & 9+0 \to 9 \\
0+1 \to 1 & 1+1 \to 2 & \cdots & 9+1 \to 1:0 \\
0+2 \to 2 & 1+2 \to 3 & \cdots & 9+2 \to 1:1 \\
\vdots & \vdots & & \vdots \\
0+8 \to 8 & 1+8 \to 9 & \cdots & 9+8 \to 1:7 \\
0+9 \to 9 & 1+9 \to 1:0 & \cdots & 9+9 \to 1:8 \\
x+(y:z) \to y:(x+z) & & & 0:x \to x \\
(x:y)+z \to x:(y+z) & & & x:(y:z) \to (x+y):z
\end{array}
$$

$$w(0) = w(1) = w(2) = w(3) = w(4) = 1 \qquad w(+) = 7$$

$$w(5) = w(6) = w(7) = w(8) = w(9) = 2 \qquad w(:) = 8 \qquad + > :$$

Example TRS/HM_t000

$$
\begin{array}{llll}
0+0\to 0 & 1+0\to 1 & \cdots & 9+0\to 9\\
0+1\to 1 & 1+1\to 2 & \cdots & 9+1\to 1:0\\
0+2\to 2 & 1+2\to 3 & \cdots & 9+2\to 1:1\\
\vdots & \vdots & & \vdots\\
0+8\to 8 & 1+8\to 9 & \cdots & 9+8\to 1:7\\
0+9\to 9 & 1+9\to 1:0 & \cdots & 9+9\to 1:8\\
x+(y:z)\to y:(x+z) & & & 0:x\to x\\
(x:y)+z\to x:(y+z) & & & x:(y:z)\to (x+y):z
\end{array}
$$

$$w(0)=w(1)=w(2)=w(3)=w(4)=1 \qquad w(+)=7$$

$$w(5)=w(6)=w(7)=w(8)=w(9)=2 \qquad w(:)=8 \qquad + > :$$

sat(4)	1.77
smt	0.67

Example TRS/HM_t000

$$
\begin{array}{lll}
0+0\to 0 & 1+0\to 1 & \cdots \quad 9+0\to 9 \\
0+1\to 1 & 1+1\to 2 & \cdots \quad 9+1\to 1:0 \\
0+2\to 2 & 1+2\to 3 & \cdots \quad 9+2\to 1:1 \\
\vdots & \vdots & \vdots \\
0+8\to 8 & 1+8\to 9 & \cdots \quad 9+8\to 1:7 \\
0+9\to 9 & 1+9\to 1:0 & \cdots \quad 9+9\to 1:8 \\
x+(y:z)\to y:(x+z) & & 0:x\to x \\
(x:y)+z\to x:(y+z) & & x:(y:z)\to (x+y):z
\end{array}
$$

$$w(0) = w(1) = w(2) = w(3) = w(4) = 1 \qquad w(+) = 7$$

$$w(5) = w(6) = w(7) = w(8) = w(9) = 2 \qquad w(:) = 8 \qquad + > :$$

sat(4)	1.77
smt	0.67
TTT	??

Example SRS/Zantema_z113

$11 \to 43$ $33 \to 56$ $55 \to 62$

$12 \to 21$ $34 \to 11$ $56 \to 12$

$22 \to 111$ $44 \to 3$ $66 \to 21$

Example SRS/Zantema_z113

$$11 \to 43 \qquad 33 \to 56 \qquad 55 \to 62$$
$$12 \to 21 \qquad 34 \to 11 \qquad 56 \to 12$$
$$22 \to 111 \qquad 44 \to 3 \qquad 66 \to 21$$

$\mathsf{T}_{\mathsf{T}}\mathsf{T}$

$$w(1) = 32471712256 \qquad w(4) = 21696293888$$
$$w(2) = 48725750528 \qquad w(5) = 44731872512$$
$$w(3) = 43247130624 \qquad w(6) = 40598731520$$
$$3 > 1 > 2 \qquad 1 > 4$$

Example SRS/Zantema_z113

$$11 \to 43 \qquad 33 \to 56 \qquad 55 \to 62$$
$$12 \to 21 \qquad 34 \to 11 \qquad 56 \to 12$$
$$22 \to 111 \qquad 44 \to 3 \qquad 66 \to 21$$

$\mathsf{T}_{\mathsf{T}}\mathsf{T}$

$$w(1) = 32471712256 \qquad w(4) = 21696293888$$
$$w(2) = 48725750528 \qquad w(5) = 44731872512$$
$$w(3) = 43247130624 \qquad w(6) = 40598731520$$
$$3 > 1 > 2 \qquad 1 > 4$$

$\mathsf{sat}(7)$

$$w(1) = 31 \qquad w(2) = 47 \qquad w(3) = 41$$
$$w(4) = 21 \qquad w(5) = 43 \qquad w(6) = 39$$
$$3 > 5 > 6 > 1 > 4 \qquad 1 > 2$$

Example SRS/Zantema_z113

$11 \to 43$ $33 \to 56$ $55 \to 62$

$12 \to 21$ $34 \to 11$ $56 \to 12$

$22 \to 111$ $44 \to 3$ $66 \to 21$

$\mathsf{T}_{\mathsf{T}}\mathsf{T}$

$w(1) = 32471712256$ $w(4) = 21696293888$

$w(2) = 48725750528$ $w(5) = 44731872512$

$w(3) = 43247130624$ $w(6) = 40598731520$

$3 > 1 > 2$ $1 > 4$

sat(7)

$w(1) = 31$ $w(2) = 47$ $w(3) = 41$

$w(4) = 21$ $w(5) = 43$ $w(6) = 39$

$3 > 5 > 6 > 1 > 4$ $1 > 2$

smt

$w(1) = 61$ $w(2) = 92$ $w(3) = 81$

$w(4) = 41$ $w(5) = 85$ $w(6) = 77$

$6 > 3 > 1 > 2 > 5 > 4$

DP with Strict Precedence

1358 TRSs in version 4.0 of TPDB

‡ weaker EDG, UR

method(# bits)	total time	# successes	# timeouts (60s)
$T_T T^{\ddagger}$	36352	397	564

DP with Strict Precedence

1358 TRSs in version 4.0 of TPDB

‡ weaker EDG, UR

method(# bits)	total time	# successes	# timeouts (60s)
$T_T T^{\ddagger}$	36352	397	564
sat(2)	2198	465	19

DP with Strict Precedence

1358 TRSs in version 4.0 of TPDB

‡ weaker EDG, UR

method(# bits)	total time	# successes	# timeouts (60s)
$T_TT^{\ddagger}$	36352	397	564
sat(2)	2198	465	19
sat(3)	2634	509	20

DP with Strict Precedence

1358 TRSs in version 4.0 of TPDB

‡ weaker EDG, UR

method(# bits)	total time	# successes	# timeouts (60s)
$T_T T^{\ddagger}$	36352	397	564
sat(2)	2198	465	19
sat(3)	2634	509	20
sat(4)	4313	508	33

DP with Strict Precedence

1358 TRSs in version 4.0 of TPDB

‡ weaker EDG, UR

method(# bits)	total time	# successes	# timeouts (60s)
$T_T T^{\ddagger}$	36352	397	564
sat(2)	2198	465	19
sat(3)	2634	509	20
sat(4)	4313	508	33
sat(5)	8154	506	71

DP with Strict Precedence

1358 TRSs in version 4.0 of TPDB

‡ weaker EDG, UR

method(# bits)	total time	# successes	# timeouts (60s)
$T_T T^{\ddagger}$	36352	397	564
sat(2)	2198	465	19
sat(3)	2634	509	20
sat(4)	4313	508	33
sat(5)	8154	506	71
smt	2779	508	20

- Motivation
- Knuth-Bendix Order
- Encodings
- Experimental Results
- Concluding Remarks

- $\forall\, k > 0\ \exists$ TRS $\mathcal{R}_k$ s.t. $\mathcal{R}_k$ is terminating in k bits but not in $k-1$ bits

- $\forall\, k > 0\ \exists$ TRS $\mathcal{R}_k$ s.t. $\mathcal{R}_k$ is terminating in k bits but not in $k-1$ bits

$$\mathsf{f}(\mathsf{g}(x,y)) \to \mathsf{g}(\mathsf{f}(x),\mathsf{f}(y)) \qquad \mathsf{h}(x) \to \mathsf{f}(\mathsf{f}(x)) \qquad \mathsf{i}(x) \to \mathsf{h}^{2^k}(x)$$

- $\forall\, k > 0\ \exists$ TRS $\mathcal{R}_k$ s.t. $\mathcal{R}_k$ is terminating in k bits but not in $k-1$ bits

$$\mathsf{f}(\mathsf{g}(x,y)) \to \mathsf{g}(\mathsf{f}(x),\mathsf{f}(y)) \qquad \mathsf{h}(x) \to \mathsf{f}(\mathsf{f}(x)) \qquad \mathsf{i}(x) \to \mathsf{h}^{2^k}(x)$$

- why is PBC better?

- $\forall\, k > 0\ \exists$ TRS $\mathcal{R}_k$ s.t. $\mathcal{R}_k$ is terminating in k bits but not in $k-1$ bits

$$\mathsf{f}(\mathsf{g}(x,y)) \to \mathsf{g}(\mathsf{f}(x),\mathsf{f}(y)) \qquad \mathsf{h}(x) \to \mathsf{f}(\mathsf{f}(x)) \qquad \mathsf{i}(x) \to \mathsf{h}^{2^k}(x)$$

- why is PBC better?
  - encoding $\longrightarrow$ less implementation work

- $\forall\, k > 0\ \exists$ TRS $\mathcal{R}_k$ s.t. $\mathcal{R}_k$ is terminating in k bits but not in $k-1$ bits

$$\mathsf{f}(\mathsf{g}(x,y)) \to \mathsf{g}(\mathsf{f}(x),\mathsf{f}(y)) \qquad \mathsf{h}(x) \to \mathsf{f}(\mathsf{f}(x)) \qquad \mathsf{i}(x) \to \mathsf{h}^{2^k}(x)$$

- why is PBC better?
  - encoding $\longrightarrow$ less implementation work
  - performance $\longrightarrow$ simulating adders in SAT destroys structure

- $\forall\, k > 0\ \exists$ TRS $\mathcal{R}_k$ s.t. $\mathcal{R}_k$ is terminating in k bits but not in $k-1$ bits

$$\mathsf{f}(\mathsf{g}(x,y)) \to \mathsf{g}(\mathsf{f}(x),\mathsf{f}(y)) \qquad \mathsf{h}(x) \to \mathsf{f}(\mathsf{f}(x)) \qquad \mathsf{i}(x) \to \mathsf{h}^{2^k}(x)$$

- why is PBC better?
  - encoding $\longrightarrow$ less implementation work
  - performance $\longrightarrow$ simulating adders in SAT destroys structure
- why is SMT best?

- $\forall\, k > 0\ \exists$ TRS $\mathcal{R}_k$ s.t. $\mathcal{R}_k$ is terminating in k bits but not in $k-1$ bits

$$\mathsf{f}(\mathsf{g}(x,y)) \to \mathsf{g}(\mathsf{f}(x),\mathsf{f}(y)) \qquad \mathsf{h}(x) \to \mathsf{f}(\mathsf{f}(x)) \qquad \mathsf{i}(x) \to \mathsf{h}^{2^k}(x)$$

- why is PBC better?
  - encoding $\longrightarrow$ less implementation work
  - performance $\longrightarrow$ simulating adders in SAT destroys structure
- why is SMT best?
  - encoding $\longrightarrow$ no work

- $\forall\, k > 0\ \exists$ TRS $\mathcal{R}_k$ s.t. $\mathcal{R}_k$ is terminating in k bits but not in $k-1$ bits

$$\mathsf{f}(\mathsf{g}(x,y)) \to \mathsf{g}(\mathsf{f}(x),\mathsf{f}(y)) \qquad \mathsf{h}(x) \to \mathsf{f}(\mathsf{f}(x)) \qquad \mathsf{i}(x) \to \mathsf{h}^{2^k}(x)$$

- why is PBC better?
  - encoding $\longrightarrow$ less implementation work
  - performance $\longrightarrow$ simulating adders in SAT destroys structure
- why is SMT best?
  - encoding $\longrightarrow$ no work
  - performance $\longrightarrow$ fastest for KBO, almost fastest for KBO+AF

- $\forall\, k > 0\ \exists$ TRS $\mathcal{R}_k$ s.t. $\mathcal{R}_k$ is terminating in k bits but not in $k-1$ bits

$$\mathsf{f}(\mathsf{g}(x,y)) \to \mathsf{g}(\mathsf{f}(x),\mathsf{f}(y)) \qquad \mathsf{h}(x) \to \mathsf{f}(\mathsf{f}(x)) \qquad \mathsf{i}(x) \to \mathsf{h}^{2^k}(x)$$

- why is PBC better?
  - encoding $\longrightarrow$ less implementation work
  - performance $\longrightarrow$ simulating adders in SAT destroys structure
- why is SMT best?
  - encoding $\longrightarrow$ no work
  - performance $\longrightarrow$ fastest for KBO, almost fastest for KBO+AF
  - no restriction in bits

- $\forall\, k > 0\ \exists$ TRS $\mathcal{R}_k$ s.t. $\mathcal{R}_k$ is terminating in k bits but not in $k-1$ bits

$$\mathsf{f}(\mathsf{g}(x,y)) \to \mathsf{g}(\mathsf{f}(x),\mathsf{f}(y)) \qquad \mathsf{h}(x) \to \mathsf{f}(\mathsf{f}(x)) \qquad \mathsf{i}(x) \to \mathsf{h}^{2^k}(x)$$

- why is PBC better?
  - encoding $\longrightarrow$ less implementation work
  - performance $\longrightarrow$ simulating adders in SAT destroys structure
- why is SMT best?
  - encoding $\longrightarrow$ no work
  - performance $\longrightarrow$ fastest for KBO, almost fastest for KBO+AF
  - no restriction in bits
- much related work

- $\forall\, k > 0\ \exists$ TRS $\mathcal{R}_k$ s.t. $\mathcal{R}_k$ is terminating in k bits but not in $k-1$ bits

$$\mathsf{f}(\mathsf{g}(x,y)) \to \mathsf{g}(\mathsf{f}(x),\mathsf{f}(y)) \qquad \mathsf{h}(x) \to \mathsf{f}(\mathsf{f}(x)) \qquad \mathsf{i}(x) \to \mathsf{h}^{2^k}(x)$$

- why is PBC better?
  - encoding $\longrightarrow$ less implementation work
  - performance $\longrightarrow$ simulating adders in SAT destroys structure
- why is SMT best?
  - encoding $\longrightarrow$ no work
  - performance $\longrightarrow$ fastest for KBO, almost fastest for KBO+AF
  - no restriction in bits
- much related work
  - LPO+AF

- $\forall\, k > 0\ \exists$ TRS $\mathcal{R}_k$ s.t. $\mathcal{R}_k$ is terminating in k bits but not in $k-1$ bits

$$\mathsf{f}(\mathsf{g}(x,y)) \to \mathsf{g}(\mathsf{f}(x),\mathsf{f}(y)) \qquad \mathsf{h}(x) \to \mathsf{f}(\mathsf{f}(x)) \qquad \mathsf{i}(x) \to \mathsf{h}^{2^k}(x)$$

- why is PBC better?
  - encoding $\longrightarrow$ less implementation work
  - performance $\longrightarrow$ simulating adders in SAT destroys structure
- why is SMT best?
  - encoding $\longrightarrow$ no work
  - performance $\longrightarrow$ fastest for KBO, almost fastest for KBO+AF
  - no restriction in bits
- much related work
  - LPO+AF
  - RPO+AF

- $\forall\, k > 0\ \exists$ TRS $\mathcal{R}_k$ s.t. $\mathcal{R}_k$ is terminating in k bits but not in $k-1$ bits

$$\mathsf{f}(\mathsf{g}(x,y)) \to \mathsf{g}(\mathsf{f}(x),\mathsf{f}(y)) \qquad \mathsf{h}(x) \to \mathsf{f}(\mathsf{f}(x)) \qquad \mathsf{i}(x) \to \mathsf{h}^{2^k}(x)$$

- why is PBC better?
  - encoding $\longrightarrow$ less implementation work
  - performance $\longrightarrow$ simulating adders in SAT destroys structure
- why is SMT best?
  - encoding $\longrightarrow$ no work
  - performance $\longrightarrow$ fastest for KBO, almost fastest for KBO+AF
  - no restriction in bits
- much related work
  - LPO+AF
  - RPO+AF
  - matrix interpretation

- $\forall\, k > 0\ \exists$ TRS $\mathcal{R}_k$ s.t. $\mathcal{R}_k$ is terminating in k bits but not in $k-1$ bits

$$\mathsf{f}(\mathsf{g}(x,y)) \to \mathsf{g}(\mathsf{f}(x),\mathsf{f}(y)) \qquad \mathsf{h}(x) \to \mathsf{f}(\mathsf{f}(x)) \qquad \mathsf{i}(x) \to \mathsf{h}^{2^k}(x)$$

- why is PBC better?
  - encoding $\longrightarrow$ less implementation work
  - performance $\longrightarrow$ simulating adders in SAT destroys structure
- why is SMT best?
  - encoding $\longrightarrow$ no work
  - performance $\longrightarrow$ fastest for KBO, almost fastest for KBO+AF
  - no restriction in bits
- much related work
  - LPO+AF
  - RPO+AF
  - matrix interpretation
  - (linear) polynomial interpretations

- $\forall\, k > 0\ \exists$ TRS $\mathcal{R}_k$ s.t. $\mathcal{R}_k$ is terminating in k bits but not in $k-1$ bits

$$\mathsf{f}(\mathsf{g}(x,y)) \to \mathsf{g}(\mathsf{f}(x),\mathsf{f}(y)) \qquad \mathsf{h}(x) \to \mathsf{f}(\mathsf{f}(x)) \qquad \mathsf{i}(x) \to \mathsf{h}^{2^k}(x)$$

- why is PBC better?
  - encoding $\longrightarrow$ less implementation work
  - performance $\longrightarrow$ simulating adders in SAT destroys structure
- why is SMT best?
  - encoding $\longrightarrow$ no work
  - performance $\longrightarrow$ fastest for KBO, almost fastest for KBO+AF
  - no restriction in bits
- much related work
  - LPO+AF
  - RPO+AF
  - matrix interpretation
  - (linear) polynomial interpretations
  - semantic labeling

- $\forall\, k > 0\ \exists$ TRS $\mathcal{R}_k$ s.t. $\mathcal{R}_k$ is terminating in k bits but not in $k-1$ bits

$$\mathsf{f}(\mathsf{g}(x,y)) \to \mathsf{g}(\mathsf{f}(x),\mathsf{f}(y)) \qquad \mathsf{h}(x) \to \mathsf{f}(\mathsf{f}(x)) \qquad \mathsf{i}(x) \to \mathsf{h}^{2^k}(x)$$

- why is PBC better?
  - encoding $\longrightarrow$ less implementation work
  - performance $\longrightarrow$ simulating adders in SAT destroys structure
- why is SMT best?
  - encoding $\longrightarrow$ no work
  - performance $\longrightarrow$ fastest for KBO, almost fastest for KBO+AF
  - no restriction in bits
- much related work
  - LPO+AF
  - RPO+AF
  - matrix interpretation
  - (linear) polynomial interpretations
  - semantic labeling
  - nontermination

- $\forall\, k > 0\ \exists$ TRS $\mathcal{R}_k$ s.t. $\mathcal{R}_k$ is terminating in k bits but not in $k-1$ bits

$$\mathsf{f}(\mathsf{g}(x,y)) \to \mathsf{g}(\mathsf{f}(x),\mathsf{f}(y)) \qquad \mathsf{h}(x) \to \mathsf{f}(\mathsf{f}(x)) \qquad \mathsf{i}(x) \to \mathsf{h}^{2^k}(x)$$

- why is PBC better?
  - encoding $\longrightarrow$ less implementation work
  - performance $\longrightarrow$ simulating adders in SAT destroys structure
- why is SMT best?
  - encoding $\longrightarrow$ no work
  - performance $\longrightarrow$ fastest for KBO, almost fastest for KBO+AF
  - no restriction in bits
- much related work
  - LPO+AF
  - RPO+AF
  - matrix interpretation
  - (linear) polynomial interpretations
  - semantic labeling
  - nontermination
  - ...