

# KBO<sup>3</sup>

Harald Zankl    Aart Middeldorp

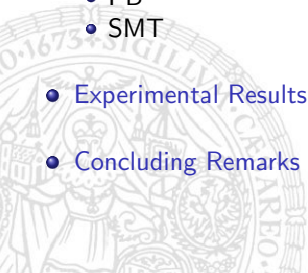
Institute of Computer Science  
University of Innsbruck

2 October 2007



# Overview

- History
- Motivation
- Knuth-Bendix Order
- Three Encodings
  - SAT
  - PB
  - SMT
- Experimental Results
- Concluding Remarks



1970 Knuth & Bendix

KBO introduced in landmark paper



1970 Knuth & Bendix

KBO introduced in landmark paper

1987 Martin

1990 Dick, Kalmus & Martin

algorithm for deciding KBO termination based on solving systems of linear inequalities using method of complete description



1970 Knuth & Bendix

KBO introduced in landmark paper

1987 Martin

1990 Dick, Kalmus & Martin

algorithm for deciding KBO termination based on solving systems of linear inequalities using method of complete description

2001 Korovin & Voronkov

polynomial time algorithm for deciding KBO termination



1970 Knuth & Bendix

KBO introduced in landmark paper

1987 Martin

1990 Dick, Kalmus & Martin

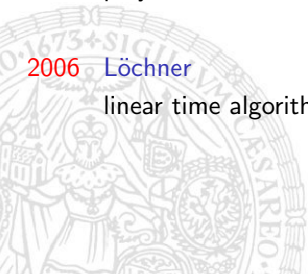
algorithm for deciding KBO termination based on solving systems of linear inequalities using method of complete description

2001 Korovin & Voronkov

polynomial time algorithm for deciding KBO termination

2006 Löchner

linear time algorithm for checking compatibility with *fixed* KBO



1970 Knuth & Bendix

KBO introduced in landmark paper

1987 Martin

1990 Dick, Kalmus & Martin

algorithm for deciding KBO termination based on solving systems of linear inequalities using method of complete description

2001 Korovin & Voronkov

polynomial time algorithm for deciding KBO termination

2006 Löchner

linear time algorithm for checking compatibility with *fixed* KBO

2007 Zankl & Middeldorp

SAT and PB encoding for 'deciding' KBO termination

## 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?

- more challenging than LPO

*Kurihara & Kondo*    1997 2004    LPO

*Codish, Lagoon & Stuckey*    2006    LPO, QLPO



## 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

*Codish, Lagoon & Stuckey*    2006    LPO, QLPO

- implementations (in T<sub>T</sub>T and AProVE) based on polynomial time algorithm

*Korovin & Voronkov*    2001/2003

are slow

- History
- Motivation
- Knuth-Bendix Order
- Three Encodings
  - SAT
  - PB
  - SMT
- Experimental Results
- Concluding Remarks



## Definition

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



## Definition

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



## Definition

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

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



## Definition

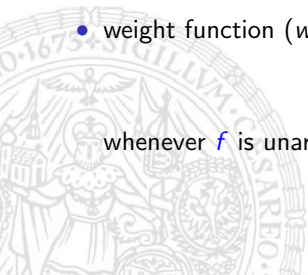
- quasi-precedence  $\succsim$  is quasi-order on signature  $\mathcal{F}$
- weight function  $(w, w_0)$  consists of mapping  $w: \mathcal{F} \rightarrow \mathbb{N}$  and constant  $w_0 > 0$  such that  $w(c) \geq w_0$  for all constants  $c \in \mathcal{F}$
- weight of term  $t$  is

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

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

$$f \succsim g \quad \forall g \in \mathcal{F}$$

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  $|s|_x \geq |t|_x$  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  $|s|_x \geq |t|_x$  for all  $x \in \mathcal{V}$  and either

❶  $w(s) > w(t)$

❷  $w(s) = w(t)$  and either

❶  $\exists n > 0$  such that  $s = f_n(\dots(f_1(t))\dots)$  and  $t \in \mathcal{V}$



## Definition

Knuth-Bendix order  $>_{\text{kbo}}$  on terms:

$s >_{\text{kbo}} t$  if  $|s|_x \geq |t|_x$  for all  $x \in \mathcal{V}$  and either

①  $w(s) > w(t)$

②  $w(s) = w(t)$  and either

①  $\exists n > 0$  such that  $s = f_n(\dots(f_1(t))\dots)$  and  $t \in \mathcal{V}$

②  $s = f(s_1, \dots, s_n)$  and  $t = g(t_1, \dots, t_m)$  and  $f \sim g$  and  $\exists i$

$$\forall j < i \quad s_j = t_j \quad s_i >_{\text{kbo}} t_i$$



## Definition

Knuth-Bendix order  $>_{\text{kbo}}$  on terms:

$s >_{\text{kbo}} t$  if  $|s|_x \geq |t|_x$  for all  $x \in \mathcal{V}$  and either

①  $w(s) > w(t)$

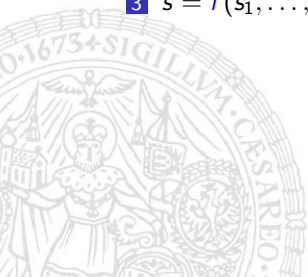
②  $w(s) = w(t)$  and either

①  $\exists n > 0$  such that  $s = f_n(\dots(f_1(t))\dots)$  and  $t \in \mathcal{V}$

②  $s = f(s_1, \dots, s_n)$  and  $t = g(t_1, \dots, t_m)$  and  $f \sim g$  and  $\exists i$

$$\forall j < i \quad s_j = t_j \quad s_i >_{\text{kbo}} t_i$$

③  $s = f(s_1, \dots, s_n)$  and  $t = g(t_1, \dots, t_m)$  and  $f > g$



## Definition

Knuth-Bendix order  $>_{\text{kbo}}$  on terms:

$s >_{\text{kbo}} t$  if  $|s|_x \geq |t|_x$  for all  $x \in \mathcal{V}$  and either

①  $w(s) > w(t)$

②  $w(s) = w(t)$  and either

①  $\exists n > 0$  such that  $s = f_n(\dots(f_1(t))\dots)$  and  $t \in \mathcal{V}$

②  $s = f(s_1, \dots, s_n)$  and  $t = g(t_1, \dots, t_m)$  and  $f \sim g$  and  $\exists i$

$$\forall j < i \quad s_j = t_j \quad s_i >_{\text{kbo}} t_i$$

③  $s = f(s_1, \dots, s_n)$  and  $t = g(t_1, \dots, t_m)$  and  $f > g$

## Theorem

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

$$\exists \text{ quasi-precedence } \succsim \quad \exists \text{ admissible weight function } (w, w_0)$$

such that  $\ell >_{\text{kbo}} r$  for all  $\ell \rightarrow r \in \mathcal{R}$

## Example

TRS/SK90\_2.42

$$\text{flat}(\text{nil}) \rightarrow \text{nil}$$

$$\text{flat}(\text{unit}(x)) \rightarrow \text{flat}(x)$$

$$\text{flat}(x ++ y) \rightarrow \text{flat}(x) ++ \text{flat}(y)$$

$$\text{flat}(\text{unit}(x) ++ y) \rightarrow \text{flat}(x) ++ \text{flat}(y)$$

$$\text{flat}(\text{flat}(x)) \rightarrow \text{flat}(x)$$

$$x ++ \text{nil} \rightarrow x$$

$$\text{rev}(\text{nil}) \rightarrow \text{nil}$$

$$\text{rev}(\text{unit}(x)) \rightarrow \text{unit}(x)$$

$$\text{rev}(x ++ y) \rightarrow \text{rev}(y) ++ \text{rev}(x)$$

$$\text{rev}(\text{rev}(x)) \rightarrow x$$

$$(x ++ y) ++ z \rightarrow x ++ (y ++ z)$$

$$\text{nil} ++ y \rightarrow y$$



## Example

TRS/SK90\_2.42

$$\text{flat}(\text{nil}) \rightarrow \text{nil}$$

$$\text{flat}(\text{unit}(x)) \rightarrow \text{flat}(x)$$

$$\text{flat}(x ++ y) \rightarrow \text{flat}(x) ++ \text{flat}(y)$$

$$\text{flat}(\text{unit}(x) ++ y) \rightarrow \text{flat}(x) ++ \text{flat}(y)$$

$$\text{flat}(\text{flat}(x)) \rightarrow \text{flat}(x)$$

$$x ++ \text{nil} \rightarrow x$$

$$\text{rev}(\text{nil}) \rightarrow \text{nil}$$

$$\text{rev}(\text{unit}(x)) \rightarrow \text{unit}(x)$$

$$\text{rev}(x ++ y) \rightarrow \text{rev}(y) ++ \text{rev}(x)$$

$$\text{rev}(\text{rev}(x)) \rightarrow x$$

$$(x ++ y) ++ z \rightarrow x ++ (y ++ z)$$

$$\text{nil} ++ y \rightarrow y$$

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

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

$$\text{flat} \sim \text{rev} > \text{unit} > ++ > \text{nil}$$



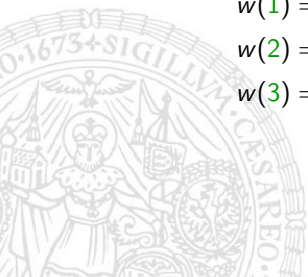
## Example

SRS/Zantema\_z113

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

## Example

SRS/Zantema\_z113

 $11 \rightarrow 43$  $33 \rightarrow 56$  $55 \rightarrow 62$  $12 \rightarrow 21$  $34 \rightarrow 11$  $56 \rightarrow 12$  $22 \rightarrow 111$  $44 \rightarrow 3$  $66 \rightarrow 21$  $w(1) = 32471712256$  $w(4) = 21696293888$  $w(2) = 48725750528$  $w(5) = 44731872512$  $w(3) = 43247130624$  $w(6) = 40598731520$  $3 > 1 > 2$  $1 > 4$ 

- History
- Motivation
- Knuth-Bendix Order
- **Three Encodings**
  - **SAT**
  - PB
  - SMT
- Experimental Results
- Concluding Remarks



## Aim

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

$\models \text{SAT}(\mathcal{R}) \implies \mathcal{R}$  is KBO terminating

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



## Aim

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

$\models \text{SAT}(\mathcal{R}) \implies \mathcal{R}$  is KBO terminating

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

## Problem

weight function  $w: \mathcal{F} \rightarrow \mathbb{N}$



## Aim

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

$$\models \text{SAT}(\mathcal{R}) \implies \mathcal{R} \text{ is KBO terminating}$$

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

## Problem

weight function  $w: \mathcal{F} \rightarrow \mathbb{N}$

## Solution

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



## Aim

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

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

## Problem

weight function  $w: \mathcal{F} \rightarrow \mathbb{N}$

## Solution

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

## Revised Aim

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

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

## Definition (Codish, Lagoon & Stuckey 2006)

- $\mathbf{a} = \langle a_k, \dots, a_1 \rangle$
- $\lceil \mathbf{a} \succ_j \mathbf{b} \rceil = \begin{cases} a_1 \wedge \neg b_1 & \text{if } j = 1 \\ a_j \wedge \neg b_j \vee (b_j \rightarrow a_j) \wedge \lceil \mathbf{a} \succ_{j-1} \mathbf{b} \rceil & \text{if } j > 1 \end{cases}$
- $\lceil \mathbf{a} \succ \mathbf{b} \rceil = \lceil \mathbf{a} \succ_k \mathbf{b} \rceil$
- $\lceil \mathbf{a} = \mathbf{b} \rceil = \bigwedge_{i=1}^k (a_i \leftrightarrow b_i)$
- $\lceil \mathbf{a} \geq \mathbf{b} \rceil = \lceil \mathbf{a} \succ \mathbf{b} \rceil \vee \lceil \mathbf{a} = \mathbf{b} \rceil$

## Definition

$$\lceil (\mathbf{a}, \varphi) + (\mathbf{b}, \psi) \rceil = (\mathbf{s}, \varphi \wedge \psi \wedge \gamma \wedge \sigma)$$

with

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

and

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

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

# Calculating Weights

## Definition

- weight of term  $t$

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

- comparing weights

$$\lceil (f, \varphi) \rceil > \lceil (g, \psi) \rceil = \lceil f \rceil > \lceil g \rceil \wedge \varphi \wedge \psi$$

- admissibility condition

$$\text{ADM}(\mathcal{F}) = \lceil w_0 \rceil > \mathbf{0} \wedge \bigwedge_{c \in \mathcal{F}^{(0)}} \lceil c \rceil \geq w_0 \wedge \bigwedge_{f \in \mathcal{F}^{(1)}} (\lceil f = \mathbf{0} \rceil \rightarrow \bigwedge_{g \in \mathcal{F}} (X_{fg} \vee Y_{fg}))$$

# Calculating Weights

## Definition

- weight of term  $t$

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

- comparing weights

$$\lceil (f, \varphi) \rceil > \lceil (g, \psi) \rceil = \lceil f \rceil > \lceil g \rceil \wedge \varphi \wedge \psi$$

- admissibility condition

$$\text{ADM}(\mathcal{F}) = \lceil w_0 \rceil > \mathbf{0} \rceil \wedge \bigwedge_{c \in \mathcal{F}^{(0)}} \lceil c \rceil \geq w_0 \rceil \wedge \bigwedge_{f \in \mathcal{F}^{(1)}} (\lceil f = \mathbf{0} \rceil \rightarrow \bigwedge_{g \in \mathcal{F}} (\mathbf{X}_{fg} \vee \mathbf{Y}_{fg}))$$

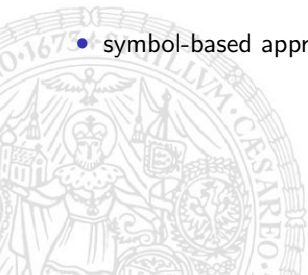
# Precedence

- atom-based approach

Kurihara & Kondo 1997 2004

- symbol-based approach

Codish, Lagoon & Stuckey 2006



# Precedence

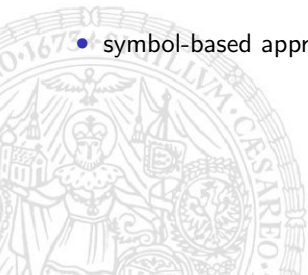
- atom-based approach

Kurihara & Kondo 1997 2004

$$\forall f, g \quad \left\{ \begin{array}{l} X_{fg} = \text{true} \\ \end{array} \right. \iff f > g$$

- symbol-based approach

Codish, Lagoon & Stuckey 2006



# Precedence

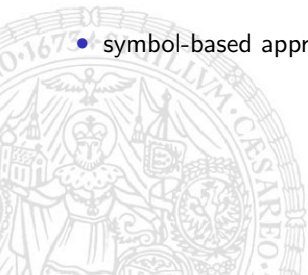
- atom-based approach

Kurihara & Kondo 1997 2004

$$\forall f, g \quad \left\{ \begin{array}{l} X_{fg} = \text{true} \iff f > g \\ Y_{fg} = \text{true} \iff f \sim g \end{array} \right.$$

- symbol-based approach

Codish, Lagoon & Stuckey 2006



# Precedence

- atom-based approach

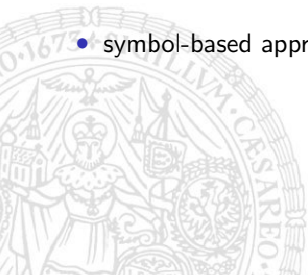
Kurihara & Kondo 1997 2004

$$\forall f, g \quad \begin{cases} X_{fg} = \text{true} & \iff f > g \\ Y_{fg} = \text{true} & \iff f \sim g \end{cases}$$

$$\text{PREC}(\mathcal{F}) = \bigwedge_{\substack{f, g, h \in \mathcal{F} \\ f \neq g \neq h \neq f}} (X_{fg} \wedge X_{gh} \rightarrow X_{fh}) \wedge \bigwedge_{\substack{f, g \in \mathcal{F} \\ f \neq g}} (X_{fg} \rightarrow \neg(Y_{fg} \vee Y_{gf})) \wedge \dots$$

- symbol-based approach

Codish, Lagoon & Stuckey 2006



# Precedence

- atom-based approach

Kurihara & Kondo 1997 2004

$$\forall f, g \quad \begin{cases} X_{fg} = \text{true} & \iff f > g \\ Y_{fg} = \text{true} & \iff f \sim g \end{cases}$$

$$\text{PREC}(\mathcal{F}) = \bigwedge_{\substack{f, g, h \in \mathcal{F} \\ f \neq g \neq h \neq f}} (X_{fg} \wedge X_{gh} \rightarrow X_{fh}) \wedge \bigwedge_{\substack{f, g \in \mathcal{F} \\ f \neq g}} (X_{fg} \rightarrow \neg(Y_{fg} \vee Y_{gf})) \wedge \dots$$

- symbol-based approach

Codish, Lagoon & Stuckey 2006

interpret function symbols in  $\{0, \dots, |\mathcal{F}| - 1\}$ , using  $l = \lceil \log_2(|\mathcal{F}|) \rceil$  bits

$$\text{PREC}(\mathcal{F}) = \bigwedge_{f, g \in \mathcal{F}} ((X_{fg} \rightarrow \lceil \mathbf{f}' >_l \mathbf{g}' \rceil) \wedge (Y_{fg} \rightarrow \lceil \mathbf{f}' =_l \mathbf{g}' \rceil))$$

## Definition

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

where  $\text{SAT}(s >_{\text{kbo}}' t)$  is

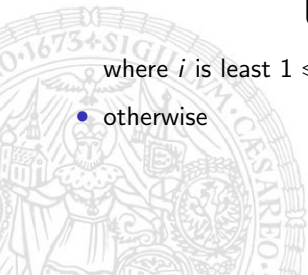
- if  $s = f(s_1, \dots, s_n)$  and  $t = g(t_1, \dots, t_m)$

$$\begin{cases} \text{SAT}(s_i >_{\text{kbo}} t_i) & \text{if } f = g \\ X_{fg} \vee Y_{fg} \wedge \text{SAT}(s_i >_{\text{kbo}} t_i) & \text{if } f \neq g \end{cases}$$

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

- otherwise

⊤



## Theorem

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

$$\bigwedge_{\ell \rightarrow r \in \mathcal{R}} \text{SAT}(\ell >_{\text{kbo}} r)$$

*is satisfiable*



## Theorem

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

$$\text{ADM}(\mathcal{F}) \wedge \text{PREC}(\mathcal{F}) \wedge \bigwedge_{\ell \rightarrow r \in \mathcal{R}} \text{SAT}(\ell >_{\text{kbo}} r)$$

*is satisfiable*



## Theorem

TRS  $\mathcal{R}$  is KBO terminating in  $k$  bits if and only if

$$\text{SAT}_k(\mathcal{R}) = \text{ADM}(\mathcal{F}) \wedge \text{PREC}(\mathcal{F}) \wedge \bigwedge_{\ell \rightarrow r \in \mathcal{R}} \text{SAT}(\ell >_{\text{kbo}} r)$$

is satisfiable



## Theorem

*TRS  $\mathcal{R}$  is KBO terminating in  $k$  bits if and only if*

$$\text{SAT}_k(\mathcal{R}) = \text{ADM}(\mathcal{F}) \wedge \text{PREC}(\mathcal{F}) \wedge \bigwedge_{\ell \rightarrow r \in \mathcal{R}} \text{SAT}(\ell >_{\text{kbo}} r)$$

*is satisfiable*

## Remark

*satisfying assignment encodes precedence and weight function*



## Theorem

*TRS  $\mathcal{R}$  is KBO terminating in  $k$  bits if and only if*

$$\text{SAT}_k(\mathcal{R}) = \text{ADM}(\mathcal{F}) \wedge \text{PREC}(\mathcal{F}) \wedge \bigwedge_{\ell \rightarrow r \in \mathcal{R}} \text{SAT}(\ell >_{\text{kbo}} r)$$

*is satisfiable*

## Remark

*satisfying assignment encodes precedence and weight function*

## Remark

$\forall k > 0 \exists \text{ TRS } \mathcal{R}_k \text{ that is KBO terminating in } k \text{ bits but not in } k - 1 \text{ bits}$

## Theorem

*TRS  $\mathcal{R}$  is KBO terminating in  $k$  bits if and only if*

$$\text{SAT}_k(\mathcal{R}) = \text{ADM}(\mathcal{F}) \wedge \text{PREC}(\mathcal{F}) \wedge \bigwedge_{\ell \rightarrow r \in \mathcal{R}} \text{SAT}(\ell >_{\text{kbo}} r)$$

*is satisfiable*

## Remark

*satisfying assignment encodes precedence and weight function*

## Remark

$\forall k > 0 \exists \text{ TRS } \mathcal{R}_k \text{ that is KBO terminating in } k \text{ bits but not in } k - 1 \text{ bits}$

$$f(g(x, y)) \rightarrow g(f(x), f(y))$$

$$h(x) \rightarrow f(f(x))$$

$$i(x) \rightarrow h^{2^k}(x)$$

- History
- Motivation
- Knuth-Bendix Order
- **Three Encodings**
  - SAT
  - **PB**
  - SMT
- Experimental Results
- Concluding Remarks



# Pseudo Boolean Constraints

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



# Pseudo Boolean Constraints

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



# Pseudo Boolean Constraints

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

## Why PB Encoding?

- **addition** and **comparisons** for free

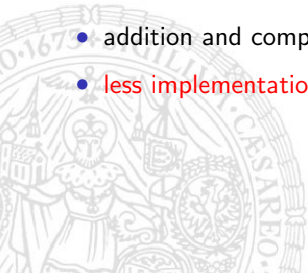


# Pseudo Boolean Constraints

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

## Why PB Encoding?

- addition and comparisons for free
- less implementation work



# Pseudo Boolean Constraints

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

## Why PB Encoding?

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

# Pseudo Boolean Constraints

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

## Why PB Encoding?

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

# Pseudo Boolean Constraints

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

## Why PB Encoding?

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

## Admissibility

ADM-PB( $\mathcal{F}$ )

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

with  $n = |\mathcal{F}|$ ,  $\bar{w}(f) = 2^{k-1} * f_k + \dots + 2^0 * f_1$ ,  $\bar{w}_0 = 2^{k-1} * w_{0k} + \dots + 2^0 * w_{01}$



## Admissibility

ADM-PB( $\mathcal{F}$ )

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

with  $n = |\mathcal{F}|$ ,  $\bar{w}(f) = 2^{k-1} * f_k + \dots + 2^0 * f_1$ ,  $\bar{w}_0 = 2^{k-1} * w_{0k} + \dots + 2^0 * w_{01}$

## Precedence

PREC-PB( $\mathcal{F}$ )

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

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



## Admissibility

ADM-PB( $\mathcal{F}$ )

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

with  $n = |\mathcal{F}|$ ,  $\bar{w}(f) = 2^{k-1} * f_k + \dots + 2^0 * f_1$ ,  $\bar{w}_0 = 2^{k-1} * w_{0k} + \dots + 2^0 * w_{01}$

## Precedence

PREC-PB( $\mathcal{F}$ )

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

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

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

## Admissibility

ADM-PB( $\mathcal{F}$ )

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

with  $n = |\mathcal{F}|$ ,  $\bar{w}(f) = 2^{k-1} * f_k + \dots + 2^0 * f_1$ ,  $\bar{w}_0 = 2^{k-1} * w_{0k} + \dots + 2^0 * w_{01}$

## Precedence

PREC-PB( $\mathcal{F}$ )

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

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

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

## Admissibility

ADM-PB( $\mathcal{F}$ )

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

with  $n = |\mathcal{F}|$ ,  $\bar{w}(f) = 2^{k-1} * f_k + \dots + 2^0 * f_1$ ,  $\bar{w}_0 = 2^{k-1} * w_{0k} + \dots + 2^0 * w_{01}$

## Precedence

PREC-PB( $\mathcal{F}$ )

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

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

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

## Admissibility

ADM-PB( $\mathcal{F}$ )

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

with  $n = |\mathcal{F}|$ ,  $\bar{w}(f) = 2^{k-1} * f_k + \dots + 2^0 * f_1$ ,  $\bar{w}_0 = 2^{k-1} * w_{0k} + \dots + 2^0 * w_{01}$

## Precedence

PREC-PB( $\mathcal{F}$ )

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

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

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

## Definition

$PB(s \succ_{kbo} t)$  consists of  $KBO_{s,t} = 0$  if  $s \in \mathcal{V}$  or  $s = t$  or  $\exists x |s|_x < |t|_x$



## Definition

$PB(s >_{kbo} t)$  consists of  $KBO_{s,t} = 0$  if  $s \in \mathcal{V}$  or  $s = t$  or  $\exists x |s|_x < |t|_x$ , and

$$- (m + 1) * KBO_{s,t} + \bar{w}(s) - \bar{w}(t) + KBO'_{s,t} \geq -m; PB(s >'_{kbo} t)$$

otherwise



## Definition

$PB(s >_{kbo} t)$  consists of  $KBO_{s,t} = 0$  if  $s \in \mathcal{V}$  or  $s = t$  or  $\exists x |s|_x < |t|_x$ , and

$$- (m + 1) * KBO_{s,t} + \bar{w}(s) - \bar{w}(t) + KBO'_{s,t} \geq -m; PB(s >'_{kbo} t)$$

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

otherwise



## Definition

$\text{PB}(s >_{\text{kbo}} t)$  consists of  $KBO_{s,t} = 0$  if  $s \in \mathcal{V}$  or  $s = t$  or  $\exists x |s|_x < |t|_x$ , and

$$- (m + 1) * KBO_{s,t} + \bar{w}(s) - \bar{w}(t) + KBO'_{s,t} \geq -m; \text{PB}(s >'_{\text{kbo}} t)$$

where  $m = 2^k * |t|$  and  $\text{PB}(s >'_{\text{kbo}} t)$  is

otherwise



## Definition

$PB(s >_{kbo} t)$  consists of  $KBO_{s,t} = 0$  if  $s \in \mathcal{V}$  or  $s = t$  or  $\exists x |s|_x < |t|_x$ , and

$$- (m + 1) * KBO_{s,t} + \bar{w}(s) - \bar{w}(t) + KBO'_{s,t} \geq -m; \quad PB(s >'_{kbo} t)$$

where  $m = 2^k * |t|$  and  $PB(s >'_{kbo} t)$  is

- if  $s = f(s_1, \dots, s_n)$  and  $t = g(t_1, \dots, t_m)$

$$PB(s_i >_{kbo} t_j); \quad \begin{cases} -KBO'_{s,t} + KBO_{s_i,t_i} \geq 0 & \text{if } f = g \\ -2 * KBO'_{s,t} + 2 * X_{fg} + Y_{fg} + KBO_{s_i,t_i} \geq 0 & \text{if } f \neq g \end{cases}$$

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

- otherwise

(empty constraint)

otherwise

# Example

$\text{PB}(s >_{\text{kbo}} t)$  for  $s = f(g(x), g(g(x)))$  and  $t = f(g(g(x)), x)$



# Example

$PB(s >_{kbo} t)$  for  $s = f(g(x), g(g(x)))$  and  $t = f(g(g(x)), x)$

$$-KBO_{s,t} + w(g) + KBO'_{s,t} \geq 0;$$



# Example

$\text{PB}(s >_{\text{kbo}} t)$  for  $s = f(g(x), g(g(x)))$  and  $t = f(g(g(x)), x)$

$$-KBO_{s,t} + w(g) + KBO'_{s,t} \geq 0;$$

$$-KBO'_{s,t} + KBO_{g(x),g(g(x))} \geq 0;$$



# Example

$PB(s >_{kbo} t)$  for  $s = f(g(x), g(g(x)))$  and  $t = f(g(g(x)), x)$

$$-KBO_{s,t} + w(g) + KBO'_{s,t} \geq 0;$$

$$-KBO'_{s,t} + KBO_{g(x),g(g(x))} \geq 0;$$

$$-(2^k + 1) * KBO_{g(x),g(g(x))} - w(g) + KBO'_{g(x),g(g(x))} \geq -2^k;$$



# Example

$PB(s >_{kbo} t)$  for  $s = f(g(x), g(g(x)))$  and  $t = f(g(g(x)), x)$

$$-KBO_{s,t} + w(g) + KBO'_{s,t} \geq 0;$$

$$-KBO'_{s,t} + KBO_{g(x),g(g(x))} \geq 0;$$

$$-(2^k + 1) * KBO_{g(x),g(g(x))} - w(g) + KBO'_{g(x),g(g(x))} \geq -2^k;$$

$$KBO'_{g(x),g(g(x))} + KBO_{x,g(x)} \geq 0;$$

$$KBO_{x,g(x)} = 0;$$



## Theorem

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

$$\text{ADM-PB}(\mathcal{F}); \text{PREC-PB}(\mathcal{F}); \left( \text{PB}(\ell >_{\text{kbo}} r); \text{KBO}_{\ell,r} = 1; \right)_{\ell \rightarrow r \in \mathcal{R}}$$

*is satisfiable*

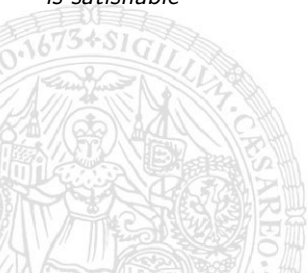


## Theorem

TRS  $\mathcal{R}$  is KBO terminating in  $k$  bits if and only if

$$\text{PB}_k(\mathcal{R}) = \text{ADM-PB}(\mathcal{F}); \text{PREC-PB}(\mathcal{F}); \left( \text{PB}(\ell >_{\text{kbo}} r); \text{KBO}_{\ell,r} = 1; \right)_{\ell \rightarrow r \in \mathcal{R}}$$

*is satisfiable*



- History
- Motivation
- Knuth-Bendix Order
- **Three Encodings**
  - SAT
  - PB
  - **SMT**
- Experimental Results
- Concluding Remarks



# Satisfiability Modulo Theories

- theory: integer linear arithmetic



# Satisfiability Modulo Theories

- theory: integer linear arithmetic
- solvers: Barcelogic, CVC3, MathSAT, Yices, Z3, ...

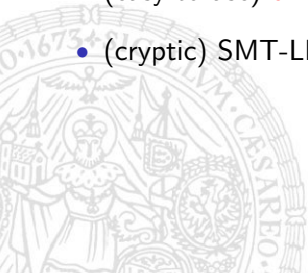


# Satisfiability Modulo Theories

- theory: integer linear arithmetic
- solvers: Barcelogic, CVC3, MathSAT, **Yices**, Z3, ...

supports

- (easy to use) **own** format
- (cryptic) SMT-LIB format



## Example

part of Yices encoding of  $f(x) \rightarrow g(x)$

```
(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 Yices encoding of  $f(x) \rightarrow g(x)$

```
(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)
```

no encoding!

- History
- Motivation
- Knuth-Bendix Order
- Three Encodings
  - SAT
  - PB
  - SMT
- Experimental Results
- Concluding Remarks



# Strict Precedence

865 TRSs in version 3.2 of TPDB

method (# bits)	total time	# successes	# timeouts (60s)
$\top\top$	119.89	77	1



# Strict Precedence

865 TRSs in version 3.2 of TPDB

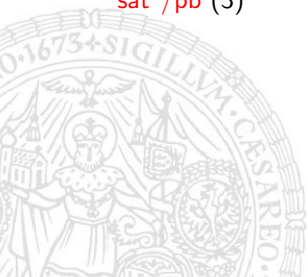
method (# bits)	total time	# successes	# timeouts (60s)
$\top\top$	119.89	77	1
sat /pb (2)	17.67/ 15.29	72/ 76	0/0



# Strict Precedence

865 TRSs in version 3.2 of TPDB

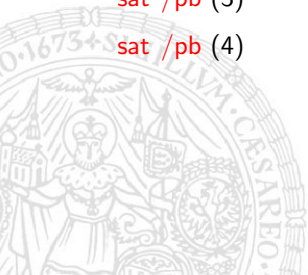
method (# bits)	total time	# successes	# timeouts (60s)
$\top\top$	119.89	77	1
sat /pb (2)	17.67/ 15.29	72/ 76	0/0
sat /pb (3)	19.05/ 15.20	77/ 77	0/0



# Strict Precedence

865 TRSs in version 3.2 of TPDB

method (# bits)	total time	# successes	# timeouts (60s)
$\top\top$	119.89	77	1
sat /pb (2)	17.67/ 15.29	72/ 76	0/0
sat /pb (3)	19.05/ 15.20	77/ 77	0/0
sat /pb (4)	20.71/ 15.43	78/ 78	0/0



# Strict Precedence

865 TRSs in version 3.2 of TPDB

method (# bits)	total time	# successes	# timeouts (60s)
$\top\top$	119.89	77	1
sat /pb (2)	17.67/ 15.29	72/ 76	0/0
sat /pb (3)	19.05/ 15.20	77/ 77	0/0
sat /pb (4)	20.71/ 15.43	78/ 78	0/0
sat /pb (10)	108.61/ 23.30	78/ 78	1/0

# Strict Precedence

865 TRSs in version 3.2 of TPDB

method (# bits)	total time	# successes	# timeouts (60s)
$\top\top$	119.89	77	1
sat /pb (2)	17.67/ 15.29	72/ 76	0/0
sat /pb (3)	19.05/ 15.20	77/ 77	0/0
sat /pb (4)	20.71/ 15.43	78/ 78	0/0
sat /pb (10)	108.61/ 23.30	78/ 78	1/0
smt	14.86	78	0

# Strict Precedence

865 TRSs in version 3.2 of TPDB

method (# bits)	total time	# successes	# timeouts (60s)
$\top\top$	119.89	77	1
sat <sup>†</sup> /pb (2)	17.76/ 15.29	76/ 76	0/0
sat <sup>†</sup> /pb (3)	19.38/ 15.20	77/ 77	0/0
sat <sup>†</sup> /pb (4)	21.45/ 15.43	78/ 78	0/0
sat <sup>†</sup> /pb (10)	102.23/ 23.30	78/ 78	1/0
smt	14.86	78	0

<sup>†</sup> extra bits for immediate results on demand

# Strict Precedence

1358 TRSs in version 4.0 of TPDB

method (# bits)	total time	# successes	# timeouts (60s)
$\top\top$	335.20	101	1
sat /pb (2)	423.52/ 98.18	90/104	2/0
sat /pb (3)	425.86/ 97.28	106/106	2/0
sat /pb (4)	434.12/ 98.43	107/107	2/0
sat /pb (10)	779.91/178.99	107/107	5/1
smt	29.83	107	0

# Strict Precedence

1358 TRSs in version 4.0 of TPDB

method (# bits)	total time	# successes	# timeouts (60s)
$\top\top$	335.20	101	1
sat <sup>†</sup> /pb (2)	421.99/ 98.18	104/104	2/0
sat <sup>†</sup> /pb (3)	428.99/ 97.28	106/106	2/0
sat <sup>†</sup> /pb (4)	437.43/ 98.43	107/107	2/0
sat <sup>†</sup> /pb (10)	790.50/178.99	107/107	5/1
smt	29.83	107	0

<sup>†</sup> extra bits for immediate results on demand

## Example

TRS/variou\_21

$$p1 + p1 \rightarrow p2$$

$$p2 + p1 \rightarrow p1 + p2$$

$$p5 + p1 \rightarrow p1 + p5$$

$$p5 + p2 \rightarrow p2 + p5$$

$$p5 + p5 \rightarrow p10$$

$$p10 + p1 \rightarrow p1 + p10$$

$$p10 + p2 \rightarrow p2 + p10$$

$$p10 + p5 \rightarrow p5 + p10$$

$$p1 + (p2 + p2) \rightarrow p5$$

$$p2 + (p2 + p2) \rightarrow p1 + p5$$

$$p1 + (p1 + x) \rightarrow p2 + x$$

$$p2 + (p1 + x) \rightarrow p1 + (p2 + x)$$

$$p5 + (p1 + x) \rightarrow p1 + (p5 + x)$$

$$p5 + (p2 + x) \rightarrow p2 + (p5 + x)$$

$$p5 + (p5 + x) \rightarrow p10 + x$$

$$p10 + (p1 + x) \rightarrow p1 + (p10 + x)$$

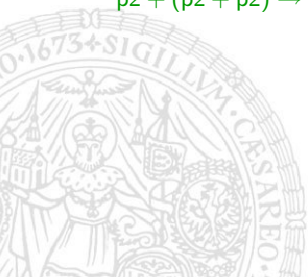
$$p10 + (p2 + x) \rightarrow p2 + (p10 + x)$$

$$p10 + (p5 + x) \rightarrow p5 + (p10 + x)$$

$$p1 + (p2 + (p2 + x)) \rightarrow p5 + x$$

$$p2 + (p2 + (p2 + x)) \rightarrow p1 + (p5 + x)$$

$$(x + y) + z \rightarrow x + (y + z)$$



## Example

TRS/variou\_21

$$p1 + p1 \rightarrow p2$$

$$p2 + p1 \rightarrow p1 + p2$$

$$p5 + p1 \rightarrow p1 + p5$$

$$p5 + p2 \rightarrow p2 + p5$$

$$p5 + p5 \rightarrow p10$$

$$p10 + p1 \rightarrow p1 + p10$$

$$p10 + p2 \rightarrow p2 + p10$$

$$p10 + p5 \rightarrow p5 + p10$$

$$p1 + (p2 + p2) \rightarrow p5$$

$$p2 + (p2 + p2) \rightarrow p1 + p5$$

$$p1 + (p1 + x) \rightarrow p2 + x$$

$$p2 + (p1 + x) \rightarrow p1 + (p2 + x)$$

$$p5 + (p1 + x) \rightarrow p1 + (p5 + x)$$

$$p5 + (p2 + x) \rightarrow p2 + (p5 + x)$$

$$p5 + (p5 + x) \rightarrow p10 + x$$

$$p10 + (p1 + x) \rightarrow p1 + (p10 + x)$$

$$p10 + (p2 + x) \rightarrow p2 + (p10 + x)$$

$$p10 + (p5 + x) \rightarrow p5 + (p10 + x)$$

$$p1 + (p2 + (p2 + x)) \rightarrow p5 + x$$

$$p2 + (p2 + (p2 + x)) \rightarrow p1 + (p5 + x)$$

$$(x + y) + z \rightarrow x + (y + z)$$

$$w(p1) = w(p2) = 4 \quad w(p5) = 6 \quad w(p10) = 11 \quad w(+) = 0$$

$$p2 > p1 \quad p5 > p1 \quad p10 > p1$$

## Example

TRS/variou\_21

$p1 + p1 \rightarrow p2$	$p1 + (p1 + x) \rightarrow p2 + x$
$p2 + p1 \rightarrow p1 + p2$	$p2 + (p1 + x) \rightarrow p1 + (p2 + x)$
$p5 + p1 \rightarrow p1 + p5$	$p5 + (p1 + x) \rightarrow p1 + (p5 + x)$
$p5 + p2 \rightarrow p2 + p5$	$p5 + (p2 + x) \rightarrow p2 + (p5 + x)$
$p5 + p5 \rightarrow p10$	$p5 + (p5 + x) \rightarrow p10 + x$
$p10 + p1 \rightarrow p1 + p10$	$p10 + (p1 + x) \rightarrow p1 + (p10 + x)$
$p10 + p2 \rightarrow p2 + p10$	$p10 + (p2 + x) \rightarrow p2 + (p10 + x)$
$p10 + p5 \rightarrow p5 + p10$	$p10 + (p5 + x) \rightarrow p5 + (p10 + x)$
$p1 + (p2 + p2) \rightarrow p5$	$p1 + (p2 + (p2 + x)) \rightarrow p5 + x$
$p2 + (p2 + p2) \rightarrow p1 + p5$	$p2 + (p2 + (p2 + x)) \rightarrow p1 + (p5 + x)$
	$(x + y) + z \rightarrow x + (y + z)$

$w(p1) = w(p2) = 4$      $w(p5) = 6$      $w(p10) = 11$      $w(+) = 0$   
 $p2 > p1$      $p5 > p1$      $p10 > p1$

<b>sat</b> (4)	0.09	<b>smt</b>	0.06
<b>pb</b> (4)	0.05		

## Example

TRS/variou\_21

$p1 + p1 \rightarrow p2$	$p1 + (p1 + x) \rightarrow p2 + x$
$p2 + p1 \rightarrow p1 + p2$	$p2 + (p1 + x) \rightarrow p1 + (p2 + x)$
$p5 + p1 \rightarrow p1 + p5$	$p5 + (p1 + x) \rightarrow p1 + (p5 + x)$
$p5 + p2 \rightarrow p2 + p5$	$p5 + (p2 + x) \rightarrow p2 + (p5 + x)$
$p5 + p5 \rightarrow p10$	$p5 + (p5 + x) \rightarrow p10 + x$
$p10 + p1 \rightarrow p1 + p10$	$p10 + (p1 + x) \rightarrow p1 + (p10 + x)$
$p10 + p2 \rightarrow p2 + p10$	$p10 + (p2 + x) \rightarrow p2 + (p10 + x)$
$p10 + p5 \rightarrow p5 + p10$	$p10 + (p5 + x) \rightarrow p5 + (p10 + x)$
$p1 + (p2 + p2) \rightarrow p5$	$p1 + (p2 + (p2 + x)) \rightarrow p5 + x$
$p2 + (p2 + p2) \rightarrow p1 + p5$	$p2 + (p2 + (p2 + x)) \rightarrow p1 + (p5 + x)$
	$(x + y) + z \rightarrow x + (y + z)$

 $w(p1) = w(p2) = 4$      $w(p5) = 6$      $w(p10) = 11$      $w(+) = 0$ 
 $p2 > p1$      $p5 > p1$      $p10 > p1$ 

<b>sat</b> (4)	0.09	<b>smt</b>	0.06
----------------	------	------------	------

<b>pb</b> (4)	0.05	<b>T</b> $\bar{T}$	4016.23
---------------	------	--------------------	---------

## Example

TRS/HM\_t000

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



## Example

TRS/HM\_t000

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

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

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

## Example

TRS/HM\_t000

$$\begin{array}{lll}
 0 + 0 \rightarrow 0 & 1 + 0 \rightarrow 1 & \dots \quad 9 + 0 \rightarrow 9 \\
 0 + 1 \rightarrow 1 & 1 + 1 \rightarrow 2 & \dots \quad 9 + 1 \rightarrow 1 : 0 \\
 0 + 2 \rightarrow 2 & 1 + 2 \rightarrow 3 & \dots \quad 9 + 2 \rightarrow 1 : 1 \\
 & \vdots & \vdots \\
 0 + 8 \rightarrow 8 & 1 + 8 \rightarrow 9 & \dots \quad 9 + 8 \rightarrow 1 : 7 \\
 0 + 9 \rightarrow 9 & 1 + 9 \rightarrow 1 : 0 & \dots \quad 9 + 9 \rightarrow 1 : 8 \\
 x + (y : z) \rightarrow y : (x + z) & & 0 : x \rightarrow x \\
 (x : y) + z \rightarrow x : (y + z) & & x : (y : z) \rightarrow (x + y) : z
 \end{array}$$
 $w(0) = w(1) = w(2) = w(3) = w(4) = 1 \quad w(+) = 7$ 
 $w(5) = w(6) = w(7) = w(8) = w(9) = 2 \quad w(:) = 8 \quad + > :$ 
 $\text{sat}(4) \quad 0.41$ 
 $\text{smt} \quad 0.15$ 
 $\text{pb}(4) \quad 0.35$

## Example

TRS/HM\_t000

$$\begin{array}{lll}
 0 + 0 \rightarrow 0 & 1 + 0 \rightarrow 1 & \dots \quad 9 + 0 \rightarrow 9 \\
 0 + 1 \rightarrow 1 & 1 + 1 \rightarrow 2 & \dots \quad 9 + 1 \rightarrow 1 : 0 \\
 0 + 2 \rightarrow 2 & 1 + 2 \rightarrow 3 & \dots \quad 9 + 2 \rightarrow 1 : 1 \\
 & \vdots & \\
 0 + 8 \rightarrow 8 & 1 + 8 \rightarrow 9 & \dots \quad 9 + 8 \rightarrow 1 : 7 \\
 0 + 9 \rightarrow 9 & 1 + 9 \rightarrow 1 : 0 & \dots \quad 9 + 9 \rightarrow 1 : 8 \\
 x + (y : z) \rightarrow y : (x + z) & & 0 : x \rightarrow x \\
 (x : y) + z \rightarrow x : (y + z) & x : (y : z) \rightarrow (x + y) : z & 
 \end{array}$$
 $w(0) = w(1) = w(2) = w(3) = w(4) = 1 \quad w(+ ) = 7$ 
 $w(5) = w(6) = w(7) = w(8) = w(9) = 2 \quad w(:) = 8 \quad + > :$ 
 $\text{sat}(4) \quad 0.41$ 
 $\text{smt} \quad 0.15$ 
 $\text{pb}(4) \quad 0.35$ 
 $\text{T}\overline{\text{T}} \quad ??$

# Quasi Precedence

865 TRSs in version 3.2 of TPDB

method (# bits)	total time	# successes	# timeouts (60s)
sat/pb (2)	20.13/ 15.96	73/ 77	0/0
sat/pb (3)	21.70/ 15.97	78/ 78	0/0
sat/pb (4)	23.60/ 16.28	79/ 79	0/0
sat/pb (10)	103.48/ 28.64	79/ 79	1/0
smt	14.60	79	0

# Quasi Precedence

1358 TRSs in version 4.0 of TPDB

method (# bits)	total time	# successes	# timeouts (60s)
sat/pb (2)	451.70/160.26	92/104	2/1
sat/pb (3)	457.77/133.13	107/107	2/0
sat/pb (4)	466.39/142.50	108/108	2/0
sat/pb (10)	833.87/229.08	108/108	5/1
smt	30.80	108	0

# String Rewrite Systems

322 SRSs in version 3.2 of TPDB

method (# bits)	total time	# successes	# timeouts (60s)
$T \bar{T}$	48.01	30	0



# String Rewrite Systems

322 SRSs in version 3.2 of TPDB

method (# bits)	total time	# successes	# timeouts (60s)
$T \bar{T} T$	48.01	30	0
sat/pb (2)	5.20/ 4.93	8/19	0/0



# String Rewrite Systems

322 SRSs in version 3.2 of TPDB

method (# bits)	total time	# successes	# timeouts (60s)
$T \bar{T}$	48.01	30	0
sat/pb (2)	5.20/ 4.93	8/19	0/0
sat/pb (3)	5.92/ 4.77	17/24	0/0
sat/pb (4)	5.90/ 4.80	24/30	0/0
sat/pb (7)	8.93/ 5.54	33/33	0/0
sat/pb (10)	13.64/ 8.91	33/33	0/0

# String Rewrite Systems

322 SRSs in version 3.2 of TPDB

method (# bits)	total time	# successes	# timeouts (60s)
$T \bar{T}$	48.01	30	0
sat/pb (2)	5.20/ 4.93	8/19	0/0
sat/pb (3)	5.92/ 4.77	17/24	0/0
sat/pb (4)	5.90/ 4.80	24/30	0/0
sat/pb (7)	8.93/ 5.54	33/33	0/0
sat/pb (10)	13.64/ 8.91	33/33	0/0
smt	5.67	33	0

# String Rewrite Systems

703 SRSs in version 4.0 of TPDB

method (# bits)	total time	# successes	# timeouts (60s)
$T \bar{T}$	55.83	30	0
sat/pb (2)	10.29/ 9.51	8/19	0/0
sat/pb (3)	11.15/ 9.39	17/24	0/0
sat/pb (4)	12.01/ 9.50	24/30	0/0
sat/pb (7)	15.97/10.50	33/33	0/0
sat/pb (10)	22.20/13.43	33/33	0/0
smt	11.78	33	0

## Example

SRS/Zantema\_z113

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

## Example

SRS/Zantema\_z113

 $11 \rightarrow 43$  $33 \rightarrow 56$  $55 \rightarrow 62$  $12 \rightarrow 21$  $34 \rightarrow 11$  $56 \rightarrow 12$  $22 \rightarrow 111$  $44 \rightarrow 3$  $66 \rightarrow 21$  $\tau\tau$  $w(1) = 32471712256$  $w(4) = 21696293888$  $w(2) = 48725750528$  $w(5) = 44731872512$  $w(3) = 43247130624$  $w(6) = 40598731520$  $3 > 1 > 2$  $1 > 4$ 

## Example

SRS/Zantema\_z113

 $11 \rightarrow 43$  $33 \rightarrow 56$  $55 \rightarrow 62$  $12 \rightarrow 21$  $34 \rightarrow 11$  $56 \rightarrow 12$  $22 \rightarrow 111$  $44 \rightarrow 3$  $66 \rightarrow 21$  $\tau\tau$  $w(1) = 32471712256$  $w(4) = 21696293888$  $w(2) = 48725750528$  $w(5) = 44731872512$  $w(3) = 43247130624$  $w(6) = 40598731520$  $3 > 1 > 2 \quad 1 > 4$ 

sat(7)

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

## Example

SRS/Zantema\_z113

 $11 \rightarrow 43$  $33 \rightarrow 56$  $55 \rightarrow 62$  $12 \rightarrow 21$  $34 \rightarrow 11$  $56 \rightarrow 12$  $22 \rightarrow 111$  $44 \rightarrow 3$  $66 \rightarrow 21$  $\tau\tau$  $w(1) = 32471712256$  $w(4) = 21696293888$  $w(2) = 48725750528$  $w(5) = 44731872512$  $w(3) = 43247130624$  $w(6) = 40598731520$  $3 > 1 > 2 \quad 1 > 4$  $w(1) = 64$  $w(2) = 97$  $w(3) = 85$  $w(4) = 43$  $w(5) = 85$  $w(6) = 89$  $3 > 1 > 5 > 2 \sim 6 > 4$  $pb(7)$

## Example

SRS/Zantema\_z113

 $11 \rightarrow 43$  $33 \rightarrow 56$  $55 \rightarrow 62$  $12 \rightarrow 21$  $34 \rightarrow 11$  $56 \rightarrow 12$  $22 \rightarrow 111$  $44 \rightarrow 3$  $66 \rightarrow 21$  $\top\top$  $w(1) = 32471712256$  $w(4) = 21696293888$  $w(2) = 48725750528$  $w(5) = 44731872512$  $w(3) = 43247130624$  $w(6) = 40598731520$  $3 > 1 > 2 \quad 1 > 4$  $w(1) = 61$  $w(2) = 92$  $w(3) = 81$  $w(4) = 41$  $w(5) = 85$  $w(6) = 77$  $6 > 3 > 1 > 2 > 5 > 4$ 

smt

- History
- Motivation
- Knuth-Bendix Order
- Three Encodings
  - SAT
  - PB
  - SMT
- Experimental Results
- Concluding Remarks



## Summary

- PB is better than SAT
  - less implementation effort



## Summary

- **PB is better than SAT**
  - less implementation effort
  - our SAT encoding (based on simulating adders) destroys arc-consistency



## Summary

- PB is better than SAT
  - less implementation effort
  - our SAT encoding (based on simulating adders) destroys arc-consistency
- **SMT is best**
  - 'no' implementation effort



## Summary

- PB is better than SAT
  - less implementation effort
  - our SAT encoding (based on simulating adders) destroys arc-consistency
- **SMT is best**
  - 'no' implementation effort
  - very fast



## Summary

- PB is better than SAT
  - less implementation effort
  - our SAT encoding (based on simulating adders) destroys arc-consistency
- **SMT is best**
  - 'no' implementation effort
  - very fast
  - no restriction on number of bits



## Summary

- PB is better than SAT
  - less implementation effort
  - our SAT encoding (based on simulating adders) destroys arc-consistency
- SMT is best
  - 'no' implementation effort
  - very fast
  - no restriction on number of bits

## Three Remarks

- **objective function** in PB is useful for finding minimal weights



## Summary

- PB is better than SAT
  - less implementation effort
  - our SAT encoding (based on simulating adders) destroys arc-consistency
- SMT is best
  - 'no' implementation effort
  - very fast
  - no restriction on number of bits

## Three Remarks

- objective function in PB is useful for finding minimal weights
- results 'extend' to combination of KBO with **dependency pairs**

## Summary

- PB is better than SAT
  - less implementation effort
  - our SAT encoding (based on simulating adders) destroys arc-consistency
- SMT is best
  - 'no' implementation effort
  - very fast
  - no restriction on number of bits

## Three Remarks

- objective function in PB is useful for finding minimal weights
- results 'extend' to combination of KBO with dependency pairs
- our experiments revealed **bugs**

## Summary

- PB is better than SAT
  - less implementation effort
  - our SAT encoding (based on simulating adders) destroys arc-consistency
- SMT is best
  - 'no' implementation effort
  - very fast
  - no restriction on number of bits

## Three Remarks

- objective function in PB is useful for finding minimal weights
- results 'extend' to combination of KBO with dependency pairs
- our experiments revealed **bugs**:  
`sat` → AProVE 1.2 (implementation of KBO with quasi-precedence)

## Summary

- PB is better than SAT
  - less implementation effort
  - our SAT encoding (based on simulating adders) destroys arc-consistency
- SMT is best
  - 'no' implementation effort
  - very fast
  - no restriction on number of bits

## Three Remarks

- objective function in PB is useful for finding minimal weights
- results 'extend' to combination of KBO with dependency pairs
- our experiments revealed **bugs**:  
**pb** → MiniSAT+

## Summary

- PB is better than SAT
  - less implementation effort
  - our SAT encoding (based on simulating adders) destroys arc-consistency
- SMT is best
  - 'no' implementation effort
  - very fast
  - no restriction on number of bits

## Three Remarks

- objective function in PB is useful for finding minimal weights
- results 'extend' to combination of KBO with dependency pairs
- our experiments revealed **bugs**:  
`smt` → CVC3

## Much Related Work

- LPO/RPO
- argument filtering in dependency pair setting
- matrix interpretations
- polynomial interpretations
- semantic labeling
- nontermination



# International School on Rewriting

21 – 26 July 2008

Obergurgl, Tirol



# International School on Rewriting

21–26 July 2008

Obergurgl, Tirol



# International School on Rewriting

21–26 July 2008

Obergurgl, Tirol

## Accessibility

- 2 hours from Innsbruck (train to Ötztal, bus to Obergurgl)
- 5 hours from Linz (RTA 2008 in Hagenberg), via Innsbruck



# International School on Rewriting

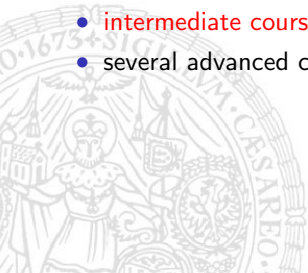
21–26 July 2008      Obergurgl, Tirol

## Accessibility

- 2 hours from Innsbruck (train to Ötztal, bus to Obergurgl)
- 5 hours from Linz (RTA 2008 in Hagenberg), via Innsbruck

## Lectures (track 1)

- full-fledged introductory term rewriting course (including exercises)
- intermediate course on resolution
- several advanced courses



# International School on Rewriting

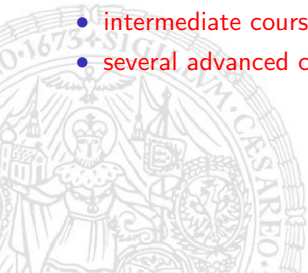
21–26 July 2008      Obergurgl, Tirol

## Accessibility

- 2 hours from Innsbruck (train to Ötztal, bus to Obergurgl)
- 5 hours from Linz (RTA 2008 in Hagenberg), via Innsbruck

## Lectures (track 2)

- full-fledged introductory term rewriting course (including exercises)
- intermediate course on resolution
- several advanced courses



# International School on Rewriting

21–26 July 2008      Obergurgl, Tirol

## Accessibility

- 2 hours from Innsbruck (train to Ötztal, bus to Obergurgl)
- 5 hours from Linz (RTA 2008 in Hagenberg), via Innsbruck

## Lectures (track 1)

- full-fledged introductory term rewriting course (including exercises)
- intermediate course on resolution
- several advanced courses

## Lecturers

- Comon, Hofbauer, Klop, Lescanne, Mackie, Nipkow, van Oostrom, van Raamsdonk, Voronkov



send your students !

