universität innsbruck

The Weighted Path Order in $\mathrm{T_TT_2}$

Master Defense

Jonas Schöpf

Supervisors: Dr. Christian Sternagel & Dr. Sarah Winkler

April 28, 2020

Motivation

- WPO is successful technique implemented in NaTT
- impact of WPO in $\mathsf{T_TT_2}$ (better results in TermComp?)
- compare WPO with implementations of LPO, KBO, LPIs, ...
- more than 60 NaTT proofs where $\mathsf{T_TT_2}$ fails

Overview

- Term Rewriting
- The Weighted Path Order
- External Power
- Certification
- Experiments

Overview

- Term Rewriting
- The Weighted Path Order
- External Power
- Certification
- Experiments

Example – Term Rewriting

function symbols $\bot, \top$ (constants) $\neg$ (unary) $\wedge, \vee$ (binary)

Example – Term Rewriting

function symbols	$\bot, \top$ (constants) $\neg$ (unary) $\wedge, \vee$ (binary)
terms	if s and t are terms, then variables, $\bot, \top, \neg t, s \wedge t, s \vee t$ are terms

Example – Term Rewriting

function symbols $\bot, \top$ (constants) $\neg$ (unary) $\wedge, \vee$ (binary)

terms if s and t are terms, then variables, $\bot, \top, \neg t, s \wedge t, s \vee t$ are terms

TRS $\mathcal{R}$

$$
\begin{array}{lll}
\neg\bot \to \top & \neg\top \to \bot & \neg\neg x \to x \\
\top \wedge x \to x & \top \vee x \to \top & \neg(x \vee y) \to \neg x \wedge \neg y \\
\bot \wedge x \to \bot & \bot \vee x \to x & \neg(x \wedge y) \to \neg x \vee \neg y
\end{array}
$$

Example – Term Rewriting

function symbols $\bot, \top$ (constants) $\neg$ (unary) $\wedge, \vee$ (binary)

terms if s and t are terms, then variables, $\bot, \top, \neg t, s \wedge t, s \vee t$ are terms

TRS $\mathcal{R}$

$$
\begin{array}{lll}
\neg\bot \to \top & \neg\top \to \bot & \neg\neg x \to x \\
\top \wedge x \to x & \top \vee x \to \top & \neg(x \vee y) \to \neg x \wedge \neg y \\
\bot \wedge x \to \bot & \bot \vee x \to x & \neg(x \wedge y) \to \neg x \vee \neg y
\end{array}
$$

$$\neg(\bot \wedge y) \wedge (\neg\neg\top)$$

rewriting

Example – Term Rewriting

function symbols $\bot, \top$ (constants) $\neg$ (unary) $\wedge, \vee$ (binary)

terms if s and t are terms, then variables, $\bot, \top, \neg t, s \wedge t, s \vee t$ are terms

TRS $\mathcal{R}$

$$
\begin{array}{lll}
\neg\bot \to \top & \neg\top \to \bot & \neg\neg x \to x \\
\top \wedge x \to x & \top \vee x \to \top & \neg(x \vee y) \to \neg x \wedge \neg y \\
\bot \wedge x \to \bot & \bot \vee x \to x & \neg(x \wedge y) \to \neg x \vee \neg y
\end{array}
$$

$$\neg(\bot \wedge y) \wedge (\underline{\neg\neg\top})$$

rewriting

Example – Term Rewriting

function symbols	$\bot, \top$ (constants) $\neg$ (unary) $\wedge, \vee$ (binary)
terms	if s and t are terms, then variables, $\bot, \top, \neg t, s \wedge t, s \vee t$ are terms

TRS $\mathcal{R}$

$$\begin{array}{rclrclrcl} \neg\bot & \to & \top & \neg\top & \to & \bot & \neg\neg x & \to & x \\ \top \wedge x & \to & x & \top \vee x & \to & \top & \neg(x \vee y) & \to & \neg x \wedge \neg y \\ \bot \wedge x & \to & \bot & \bot \vee x & \to & x & \neg(x \wedge y) & \to & \neg x \vee \neg y \end{array}$$

$$\neg(\bot \wedge y) \wedge (\underline{\neg\neg\top}) \to \underline{\neg(\bot \wedge y)} \wedge \top$$

rewriting

Example – Term Rewriting

function symbols $\bot, \top$ (constants) $\neg$ (unary) $\wedge, \vee$ (binary)

terms if s and t are terms, then variables, $\bot, \top, \neg t, s \wedge t, s \vee t$ are terms

TRS $\mathcal{R}$

$$\begin{array}{lll} \neg\bot \to \top & \neg\top \to \bot & \neg\neg x \to x \\ \top \wedge x \to x & \top \vee x \to \top & \neg(x \vee y) \to \neg x \wedge \neg y \\ \bot \wedge x \to \bot & \bot \vee x \to x & \neg(x \wedge y) \to \neg x \vee \neg y \end{array}$$

rewriting

$$\begin{aligned} \neg(\bot \wedge y) \wedge (\underline{\neg\neg\top}) &\to \underline{\neg(\bot \wedge y)} \wedge \top \\ &\to (\underline{\neg\bot} \vee \neg y) \wedge \top \end{aligned}$$

Example – Term Rewriting

function symbols	$\bot, \top$ (constants) $\quad \neg$ (unary) $\quad \wedge, \vee$ (binary)
terms	if s and t are terms, then variables, $\bot, \top, \neg t, s \wedge t, s \vee t$ are terms

TRS $\mathcal{R}$

$$\begin{array}{lll} \neg\bot \to \top & \neg\top \to \bot & \neg\neg x \to x \\ \top \wedge x \to x & \top \vee x \to \top & \neg(x \vee y) \to \neg x \wedge \neg y \\ \bot \wedge x \to \bot & \bot \vee x \to x & \neg(x \wedge y) \to \neg x \vee \neg y \end{array}$$

rewriting

$$\begin{aligned} \neg(\bot \wedge y) \wedge (\underline{\neg\neg\top}) &\to \underline{\neg(\bot \wedge y)} \wedge \top \\ &\to (\underline{\neg\bot} \vee \neg y) \wedge \top \\ &\to (\underline{\top \vee \neg y}) \wedge \top \end{aligned}$$

Example – Term Rewriting

function symbols $\bot, \top$ (constants) $\neg$ (unary) $\wedge, \vee$ (binary)

terms if s and t are terms, then variables, $\bot, \top, \neg t, s \wedge t, s \vee t$ are terms

TRS $\mathcal{R}$

$$
\begin{array}{lll}
\neg\bot \to \top & \neg\top \to \bot & \neg\neg x \to x \\
\top \wedge x \to x & \top \vee x \to \top & \neg(x \vee y) \to \neg x \wedge \neg y \\
\bot \wedge x \to \bot & \bot \vee x \to x & \neg(x \wedge y) \to \neg x \vee \neg y
\end{array}
$$

rewriting

$$
\begin{aligned}
\neg(\bot \wedge y) \wedge (\underline{\neg\neg\top}) &\to \underline{\neg(\bot \wedge y)} \wedge \top \\
&\to (\underline{\neg\bot} \vee \neg y) \wedge \top \\
&\to (\underline{\top \vee \neg y}) \wedge \top \\
&\to \underline{\top \wedge \top}
\end{aligned}
$$

Example – Term Rewriting

function symbols $\bot, \top$ (constants) $\neg$ (unary) $\wedge, \vee$ (binary)

terms if s and t are terms, then variables, $\bot, \top, \neg t, s \wedge t, s \vee t$ are terms

TRS $\mathcal{R}$

$$
\begin{array}{rclrclrcl}
\neg\bot & \to & \top & \neg\top & \to & \bot & \neg\neg x & \to & x \\
\top \wedge x & \to & x & \top \vee x & \to & \top & \neg(x \vee y) & \to & \neg x \wedge \neg y \\
\bot \wedge x & \to & \bot & \bot \vee x & \to & x & \neg(x \wedge y) & \to & \neg x \vee \neg y
\end{array}
$$

rewriting

$$
\begin{aligned}
\neg(\bot \wedge y) \wedge (\underline{\neg\neg\top}) &\to \underline{\neg(\bot \wedge y)} \wedge \top \\
&\to (\underline{\neg\bot} \vee \neg y) \wedge \top \\
&\to (\underline{\top \vee \neg y}) \wedge \top \\
&\to \underline{\top \wedge \top} \to \top
\end{aligned}
$$

Reduction Order

$>$ is a reduction order if:

- $>$ is well-founded order on terms
- $s > t \Rightarrow C[s] > C[t]$
- $s > t \Rightarrow s\sigma > t\sigma$

Reduction Order

$>$ is a reduction order if:

- $>$ is well-founded order on terms
- $s > t \Rightarrow C[s] > C[t]$
- $s > t \Rightarrow s\sigma > t\sigma$

Theorem

TRS $\mathcal{R}$ terminates $\iff$ $\exists$ reduction order $>$. $\forall l \to r \in \mathcal{R}.\ l > r$

Overview

- Term Rewriting
- The Weighted Path Order
- External Power
- Certification
- Experiments

WPO as a Reduction Order

Consider a quasi-precedence $\gtrsim_{\mathcal{F}}$, a well-founded algebra $\mathcal{A}$ and a status function σ.
We define the relation $>_{\mathsf{WPO}(\mathcal{A},\sigma)}$ as follows:
For $s = f(s_1, \ldots, s_n) >_{\mathsf{WPO}(\mathcal{A},\sigma)} t$ iff

WPO as a Reduction Order

Consider a quasi-precedence $\gtrsim_{\mathcal{F}}$, a well-founded algebra $\mathcal{A}$ and a status function σ.
We define the relation $>_{\mathsf{WPO}(\mathcal{A},\sigma)}$ as follows:
For $s = f(s_1, \ldots, s_n) >_{\mathsf{WPO}(\mathcal{A},\sigma)} t$ iff

(1) $s >_{\mathcal{A}} t$, or

(2) $s \gtrsim_{\mathcal{A}} t$ and

WPO as a Reduction Order

Consider a quasi-precedence $\gtrsim_{\mathcal{F}}$, a well-founded algebra $\mathcal{A}$ and a status function σ.
We define the relation $>_{\mathsf{WPO}(\mathcal{A},\sigma)}$ as follows:
For $s = f(s_1, \ldots, s_n) >_{\mathsf{WPO}(\mathcal{A},\sigma)} t$ iff

(1) $s >_{\mathcal{A}} t$, or

(2) $s \gtrsim_{\mathcal{A}} t$ and

- (2.a) $\exists i \in \{1, \ldots, n\}.\ s_i \geq_{\mathsf{WPO}(\mathcal{A},\sigma)} t$, or
- (2.b) $t = g(t_1, \ldots, t_m)$, and $\forall j \in \{1, \ldots, m\}.\ s >_{\mathsf{WPO}(\mathcal{A},\sigma)} t_j$ and either

WPO as a Reduction Order

Consider a quasi-precedence $\gtrsim_{\mathcal{F}}$, a well-founded algebra $\mathcal{A}$ and a status function σ.
We define the relation $>_{\mathsf{WPO}(\mathcal{A},\sigma)}$ as follows:
For $s = f(s_1, \ldots, s_n) >_{\mathsf{WPO}(\mathcal{A},\sigma)} t$ iff

(1) $s >_{\mathcal{A}} t$, or

(2) $s \gtrsim_{\mathcal{A}} t$ and

- (2.a) $\exists i \in \{1, \ldots, n\}.\ s_i \geq_{\mathsf{WPO}(\mathcal{A},\sigma)} t$, or
- (2.b) $t = g(t_1, \ldots, t_m)$, and $\forall j \in \{1, \ldots, m\}.\ s >_{\mathsf{WPO}(\mathcal{A},\sigma)} t_j$ and either
  - $f >_{\mathcal{F}} g$ or
  - $f \sim_{\mathcal{F}} g$ and $[s_1, \ldots, s_n]^{\sigma(f)} >^{\mathsf{lex}}_{\mathsf{WPO}(\mathcal{A},\sigma)} [t_1, \ldots, t_m]^{\sigma(g)}$

WPO as a Reduction Order

Consider a quasi-precedence $\gtrsim_{\mathcal{F}}$, a well-founded algebra $\mathcal{A}$ and a status function σ.
We define the relation $>_{\mathsf{WPO}(\mathcal{A},\sigma)}$ as follows:
For $s = f(s_1, \ldots, s_n) >_{\mathsf{WPO}(\mathcal{A},\sigma)} t$ iff

(1) $s >_{\mathcal{A}} t$, or

(2) $s \gtrsim_{\mathcal{A}} t$ and

(2.a) $\exists i \in \{1, \ldots, n\}.\ s_i \geq_{\mathsf{WPO}(\mathcal{A},\sigma)} t$, or

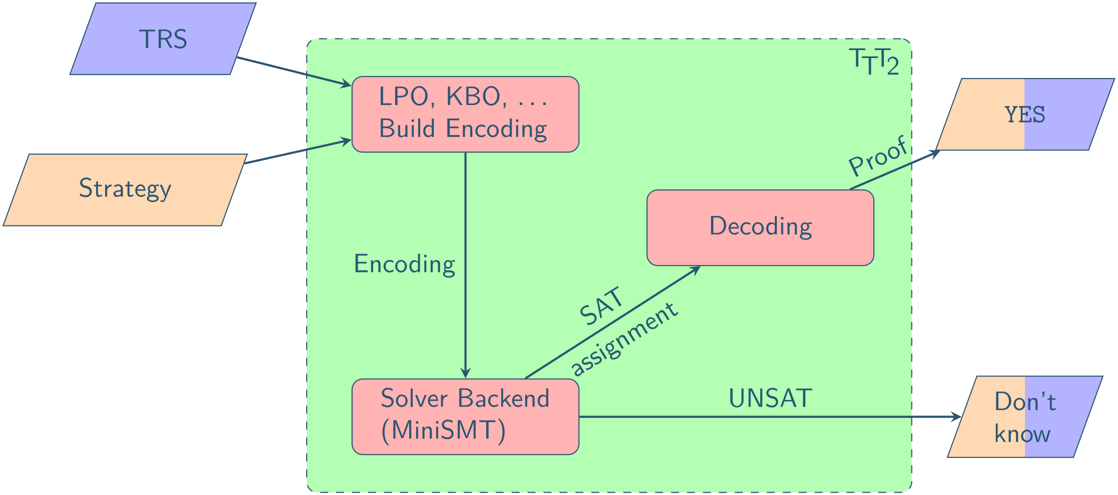
(2.b) $t = g(t_1, \ldots, t_m)$, and $\forall j \in \{1, \ldots, m\}.\ s >_{\mathsf{WPO}(\mathcal{A},\sigma)} t_j$ and either

- $f >_{\mathcal{F}} g$ or
- $f \sim_{\mathcal{F}} g$ and $[s_1, \ldots, s_n]^{\sigma(f)} >^{\mathsf{lex}}_{\mathsf{WPO}(\mathcal{A},\sigma)} [t_1, \ldots, t_m]^{\sigma(g)}$

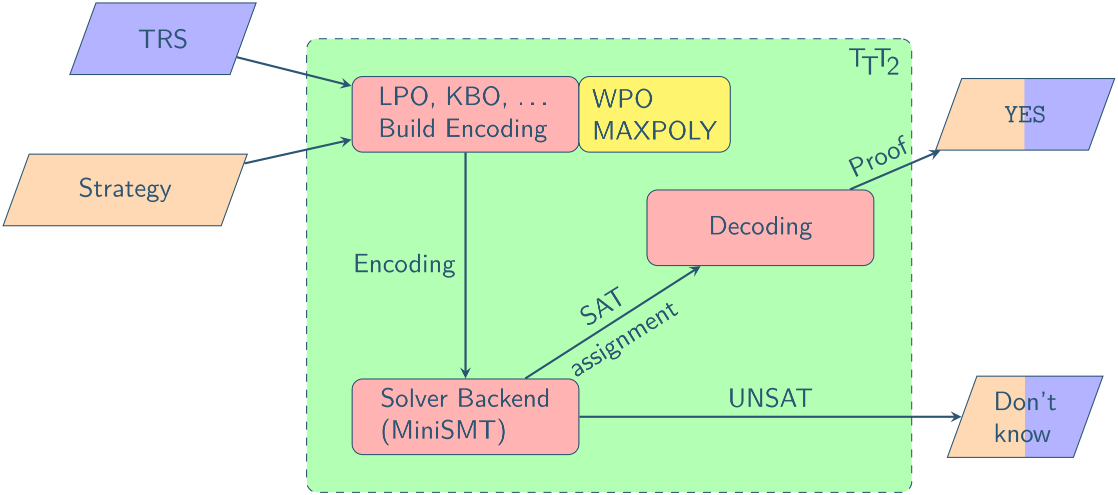
Theorem

If $\mathcal{A}$ is weakly monotone and weakly simple then $>_{\mathsf{WPO}(\mathcal{A})}$ is a reduction order

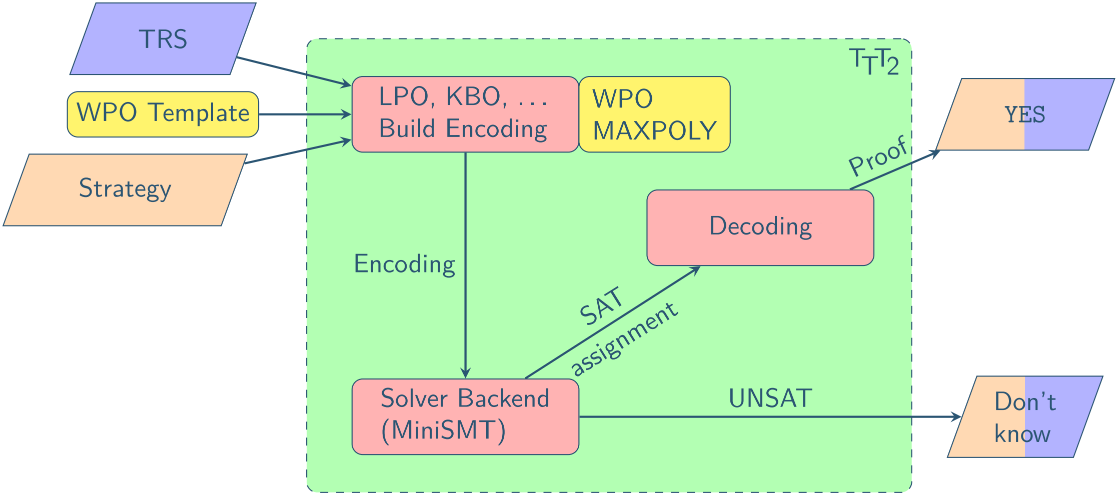
Contributions to TTT2



Contributions to T_TT_2



Contributions to $\mathsf{T_TT_2}$



WPO Implementation

- implementation of 5 algebras ($\mathcal{S}um$, $\mathcal{P}ol$, $\mathcal{M}ax$, $\mathcal{MS}um$, $\mathcal{MP}ol$)

WPO Implementation

- implementation of 5 algebras ($\mathcal{S}um$, $\mathcal{P}ol$, $\mathcal{M}ax$, $\mathcal{MS}um$, $\mathcal{MP}ol$)
- WPO as reduction pair with partial Status

WPO Implementation

- implementation of 5 algebras ($\mathcal{S}um$, $\mathcal{P}ol$, $\mathcal{M}ax$, $\mathcal{MS}um$, $\mathcal{MP}ol$)
- WPO as reduction pair with partial Status
- experimental implementation with matrix interpretations in WPO

WPO Implementation

- implementation of 5 algebras ($\mathcal{S}um$, $\mathcal{P}ol$, $\mathcal{M}ax$, $\mathcal{MS}um$, $\mathcal{MP}ol$)
- WPO as reduction pair with partial Status
- experimental implementation with matrix interpretations in WPO

WPO Improves $\mathsf{T_TT_2}$?

- actually not really
- ≈ 60 termination proofs of NaTT, where $\mathsf{T_TT_2}$ has `MAYBE`

WPO Implementation

- implementation of 5 algebras ($\mathcal{S}um$, $\mathcal{P}ol$, $\mathcal{M}ax$, $\mathcal{MS}um$, $\mathcal{MP}ol$)
- WPO as reduction pair with partial Status
- experimental implementation with matrix interpretations in WPO

WPO Improves $\mathsf{T_TT_2}$?

- actually not really
- ≈ 60 termination proofs of NaTT, where $\mathsf{T_TT_2}$ has `MAYBE`
- maximal polynomial interpretations

WPO Implementation

- implementation of 5 algebras ($\mathcal{S}um$, $\mathcal{P}ol$, $\mathcal{M}ax$, $\mathcal{MS}um$, $\mathcal{MP}ol$)
- WPO as reduction pair with partial Status
- experimental implementation with matrix interpretations in WPO

WPO Improves $\mathsf{T_TT_2}$?

- actually not really
- ≈ 60 termination proofs of NaTT, where $\mathsf{T_TT_2}$ has `MAYBE`
- maximal polynomial interpretations
- 4 new proofs (found with specific strategy)

WPO Implementation

- implementation of 5 algebras ($\mathcal{S}um$, $\mathcal{P}ol$, $\mathcal{M}ax$, $\mathcal{MS}um$, $\mathcal{MP}ol$)
- WPO as reduction pair with partial Status
- experimental implementation with matrix interpretations in WPO

WPO Improves $\mathsf{T_TT_2}$?

- actually not really
- ≈ 60 termination proofs of NaTT, where $\mathsf{T_TT_2}$ has `MAYBE`
- maximal polynomial interpretations
- 4 new proofs (found with specific strategy)

Templates for WPO

- how to find proof with specific parameters?

WPO Implementation

- implementation of 5 algebras ($\mathcal{S}um$, $\mathcal{P}ol$, $\mathcal{M}ax$, $\mathcal{MS}um$, $\mathcal{MP}ol$)
- WPO as reduction pair with partial Status
- experimental implementation with matrix interpretations in WPO

WPO Improves $\mathsf{T_TT_2}$?

- actually not really
- ≈ 60 termination proofs of NaTT, where $\mathsf{T_TT_2}$ has `MAYBE`
- maximal polynomial interpretations
- 4 new proofs (found with specific strategy)

Templates for WPO

- how to find proof with specific parameters?
- templates to insert preconditions

Example

$$\begin{aligned} \mathsf{f}(\mathsf{g}(\mathsf{g}(x,\mathsf{a}),\mathsf{g}(\mathsf{b},y))) &\to \mathsf{f}(\mathsf{g}(\mathsf{g}(\mathsf{h}(x,x),\mathsf{b}),\mathsf{g}(y,\mathsf{a}))) \\ \mathsf{g}(x,y) &\to x \\ \mathsf{h}(x,\mathsf{h}(y,z)) &\to y \end{aligned}$$

Example

$$\mathsf{f}(\mathsf{g}(\mathsf{g}(x,\mathsf{a}),\mathsf{g}(\mathsf{b},y))) \to \mathsf{f}(\mathsf{g}(\mathsf{g}(\mathsf{h}(x,x),\mathsf{b}),\mathsf{g}(y,\mathsf{a})))$$
$$\mathsf{g}(x,y) \to x$$
$$\mathsf{h}(x,\mathsf{h}(y,z)) \to y$$

Termination Proof with WPO($\mathcal{M}ax$)

statuses:	$\mathsf{st}(\mathsf{a}) = \mathsf{st}(\mathsf{b}) = []$	$\mathsf{st}(\mathsf{h}) = \mathsf{st}(\mathsf{g}) = [2,1]$	$\mathsf{st}(\mathsf{f}) = [1]$
interpretations:	$\mathsf{a}_{\mathcal{A}} = 0$	$\mathsf{h}_{\mathcal{A}}(x,y) = \max(2,x,y)$	$\mathsf{f}_{\mathcal{A}}(x) = \max(0,x)$
	$\mathsf{b}_{\mathcal{A}} = 2$	$\mathsf{g}_{\mathcal{A}}(x,y) = \max(1,x,1+y)$	
precedence:	$\mathsf{f} > \mathsf{g} > \mathsf{h} \sim \mathsf{b} \sim \mathsf{a}$		

Example

$$\begin{aligned} \mathsf{f}(\mathsf{g}(\mathsf{g}(x,\mathsf{a}),\mathsf{g}(\mathsf{b},y))) &\to \mathsf{f}(\mathsf{g}(\mathsf{g}(\mathsf{h}(x,x),\mathsf{b}),\mathsf{g}(y,\mathsf{a}))) \\ \mathsf{g}(x,y) &\to x \\ \mathsf{h}(x,\mathsf{h}(y,z)) &\to y \end{aligned}$$

Termination Proof

- AProVE: after 20 min aborted

Example

$$\begin{aligned} \mathsf{f}(\mathsf{g}(\mathsf{g}(x,\mathsf{a}),\mathsf{g}(\mathsf{b},y))) &\to \mathsf{f}(\mathsf{g}(\mathsf{g}(\mathsf{h}(x,x),\mathsf{b}),\mathsf{g}(y,\mathsf{a}))) \\ \mathsf{g}(x,y) &\to x \\ \mathsf{h}(x,\mathsf{h}(y,z)) &\to y \end{aligned}$$

Termination Proof

- AProVE: after 20 min aborted
- TTT2 without WPO: 65 s (DP, TDG, SCCs, UR,. . . , ARCTIC)
- TTT2 with WPO: 0,45 s (WPO($\mathcal{M}ax$)) $\Rightarrow$ both TTT2 proofs certified

Example

$$\mathsf{f}(\mathsf{g}(\mathsf{g}(x,\mathsf{a}),\mathsf{g}(\mathsf{b},y))) \rightarrow \mathsf{f}(\mathsf{g}(\mathsf{g}(\mathsf{h}(x,x),\mathsf{b}),\mathsf{g}(y,\mathsf{a})))$$
$$\mathsf{g}(x,y) \rightarrow x$$
$$\mathsf{h}(x,\mathsf{h}(y,z)) \rightarrow y$$

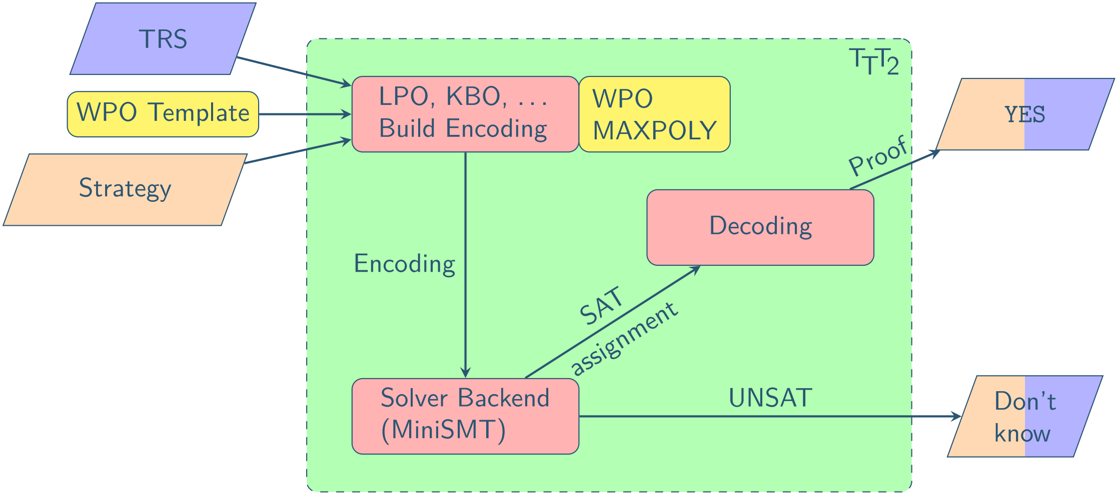
Termination Proof

- AProVE: after 20 min aborted
- $\mathsf{T_TT_2}$ without WPO: 65 s (DP, TDG, SCCs, UR,. . . , ARCTIC)
- $\mathsf{T_TT_2}$ with WPO: 0,45 s (WPO($\mathcal{M}ax$)) $\Rightarrow$ both $\mathsf{T_TT_2}$ proofs certified
- NaTT needs 0,076 s (WPO($\mathcal{MS}um$) with partial Status)
- NaTT needs 21 s (WPO($\mathcal{M}ax$))

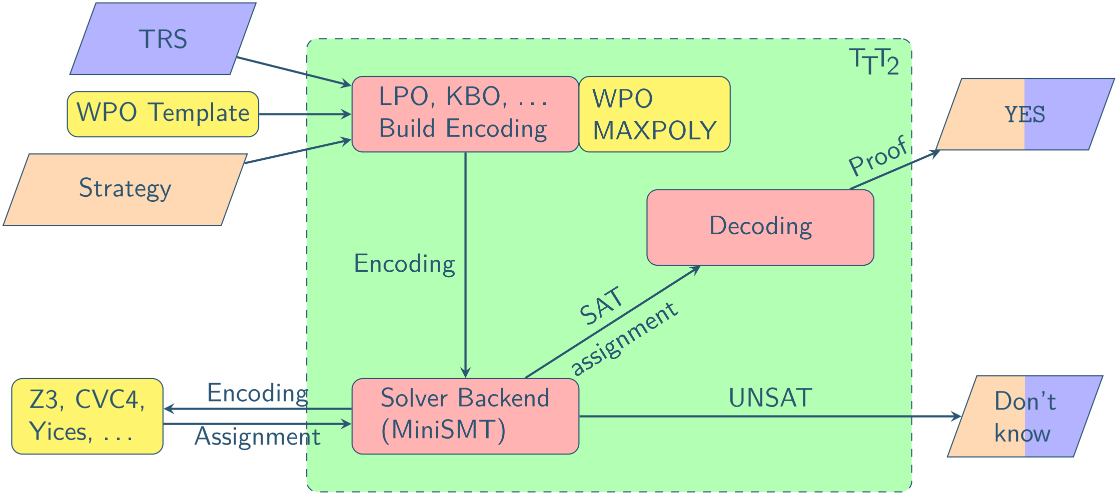
Overview

- Term Rewriting
- The Weighted Path Order
- External Power
- Certification
- Experiments

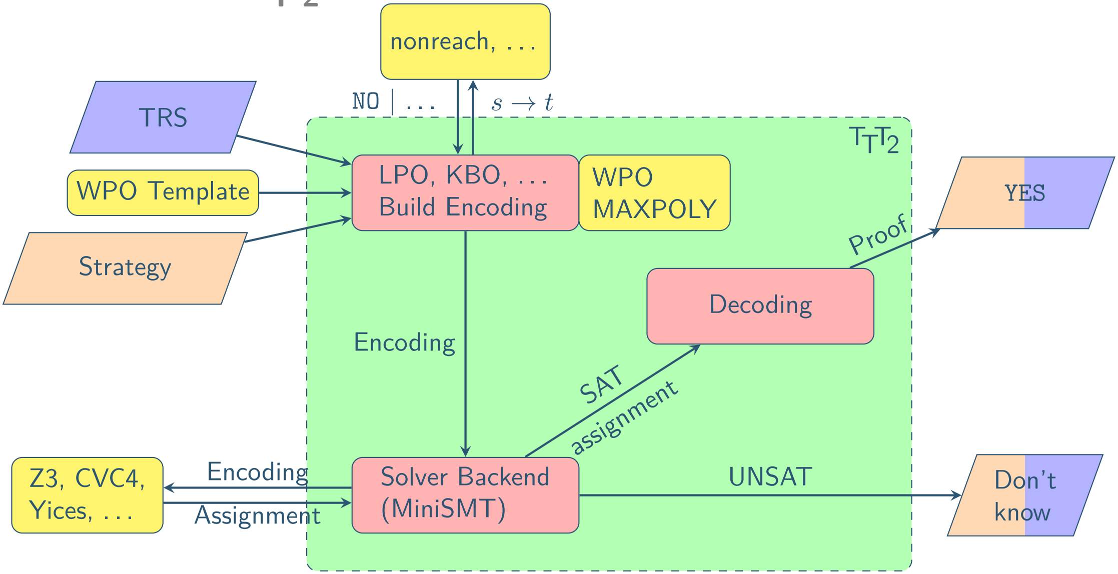
Contributions to $\mathsf{T_TT_2}$



Contributions to $\mathsf{T_TT_2}$



Contributions to $\mathsf{T_TT_2}$



External SMT Solver

- avoid manually adapting number of bits

External SMT Solver

- avoid manually adapting number of bits
- incorporate state-of-the-art SMT solvers

External SMT Solver

- avoid manually adapting number of bits
- incorporate state-of-the-art SMT solvers
- Z3 is better with negative and higher numbers

External SMT Solver

- avoid manually adapting number of bits
- incorporate state-of-the-art SMT solvers
- Z3 is better with negative and higher numbers
- Z3 enhances MAXPOLY

External SMT Solver

- avoid manually adapting number of bits
- incorporate state-of-the-art SMT solvers
- Z3 is better with negative and higher numbers
- Z3 enhances MAXPOLY

External Nonreachility Analysis

- nonreachability used for edge estimation, usable rules, ...

External SMT Solver

- avoid manually adapting number of bits
- incorporate state-of-the-art SMT solvers
- Z3 is better with negative and higher numbers
- Z3 enhances MAXPOLY

External Nonreachility Analysis

- nonreachability used for edge estimation, usable rules, ...
- nonreach developed at CL

External SMT Solver

- avoid manually adapting number of bits
- incorporate state-of-the-art SMT solvers
- Z3 is better with negative and higher numbers
- Z3 enhances MAXPOLY

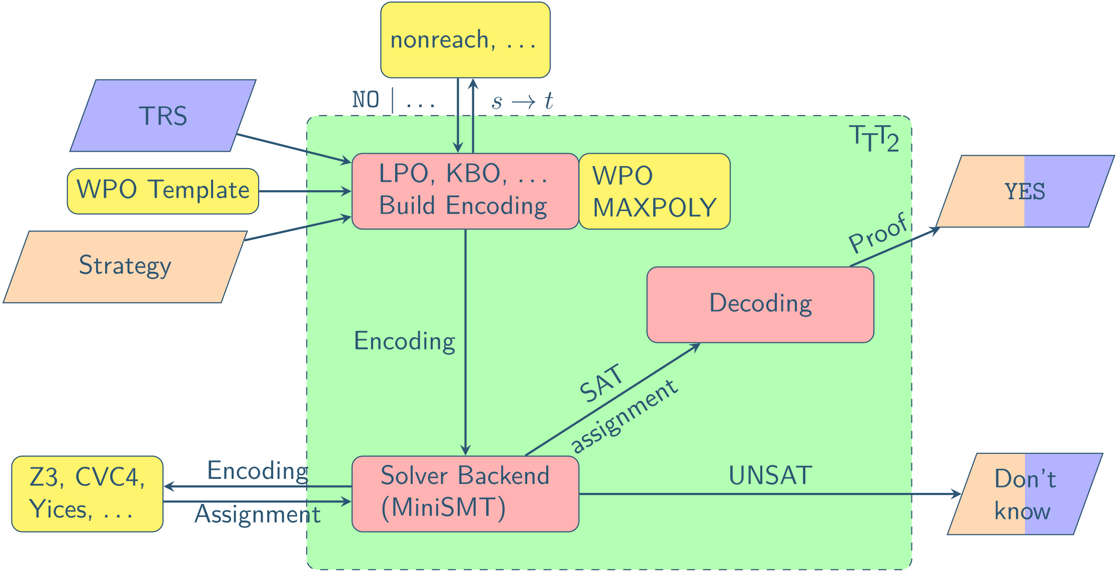
External Nonreachility Analysis

- nonreachability used for edge estimation, usable rules, ...
- nonreach developed at CL
- the stronger the nonreachability analysis is, the less usable rules we have

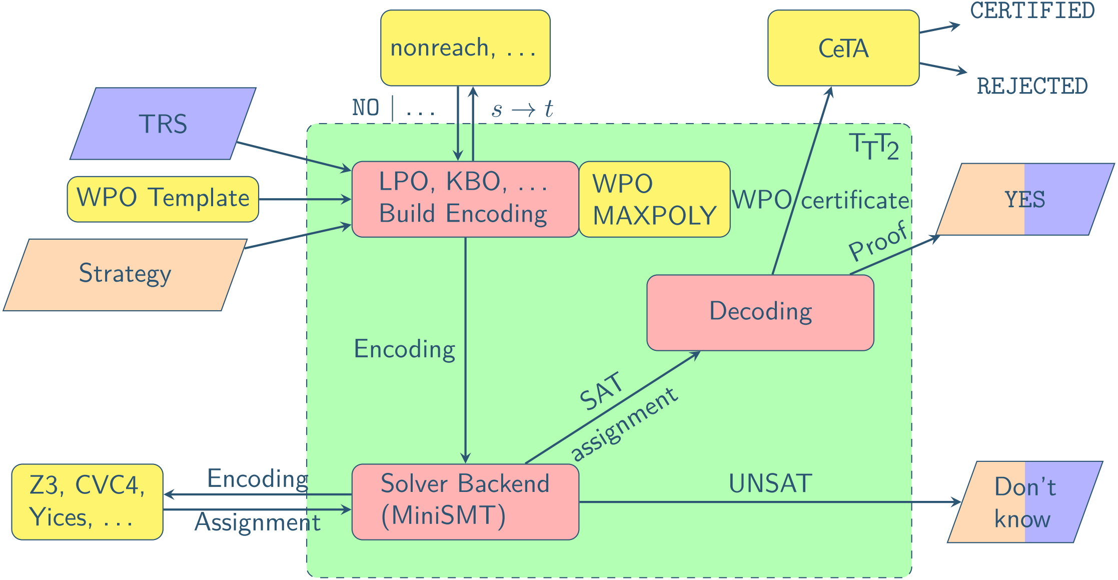
Overview

- Term Rewriting
- The Weighted Path Order
- External Power
- Certification
- Experiments

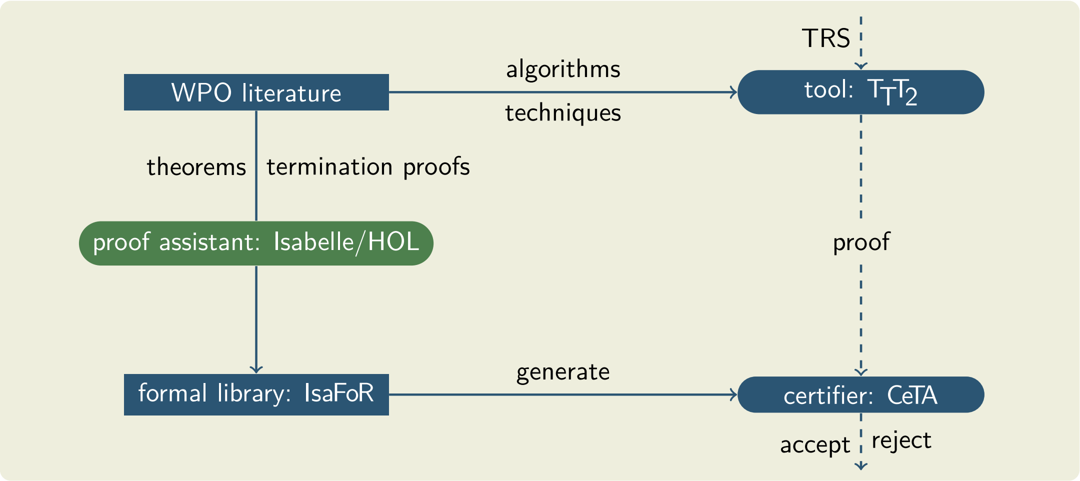
Contributions to $\mathsf{T_TT_2}$



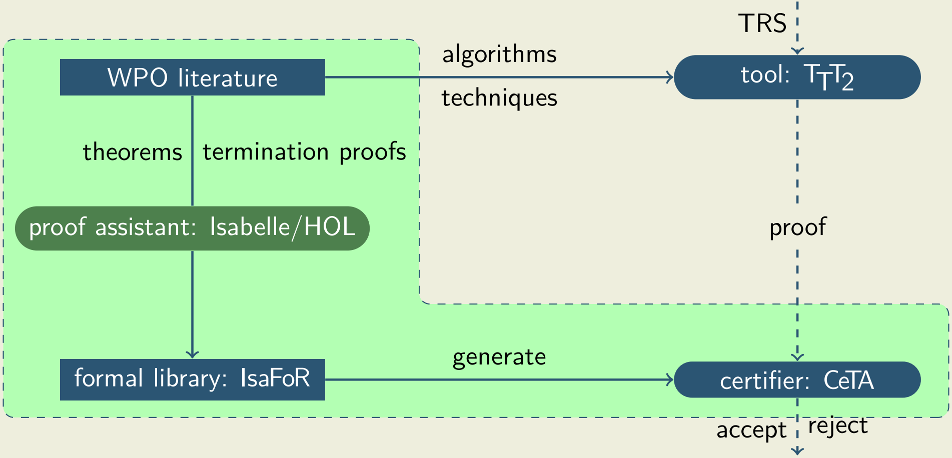
Contributions to $\mathsf{T_TT_2}$



Certification of WPO Proofs

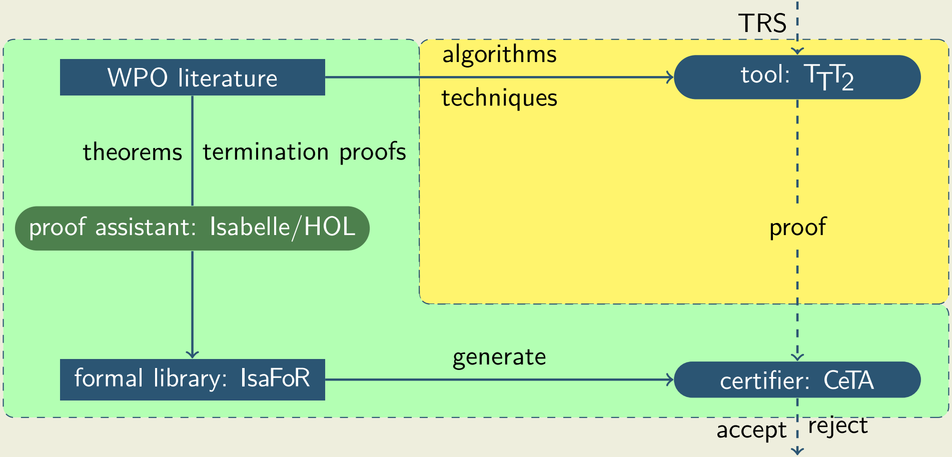


Certification of WPO Proofs



- theory of WPO was already formalized and added to CeTA

Certification of WPO Proofs



- theory of WPO was already formalized and added to CeTA
- generate output according to certificate grammar

Overview

- Term Rewriting
- The Weighted Path Order
- External Power
- Certification
- Experiments

Experiments

Method	$\mathsf{T_TT_2}$	Cert. Method	CeTA ∘ $\mathsf{T_TT_2}$
WPO	295	WPOC	295
NaTT WPO	255	–	–
MAXPOLYDP	541	–	–
old COMP	998	old COMPCERT	925
COMP	1031	COMPCERT	947
NaTT COMP	1033	NaTT CERT	≥751

Summary

- implementation of WPO in $\mathsf{T_TT_2}$

Summary

- implementation of WPO in $\mathsf{T_TT_2}$
- certificates and a template mechanism for WPO

Summary

- implementation of WPO in $\mathsf{T_TT_2}$
- certificates and a template mechanism for WPO
- employing external (state-of-the-art) SMT solvers (Z3, CVC4, Yices, . . .)

Summary

- implementation of WPO in $\mathsf{T_TT_2}$
- certificates and a template mechanism for WPO
- employing external (state-of-the-art) SMT solvers (Z3, CVC4, Yices, ...)
- backend for nonreachability analysis (edge estimation, usable rules, ...)

Summary

- implementation of WPO in $\mathsf{T_TT_2}$
- certificates and a template mechanism for WPO
- employing external (state-of-the-art) SMT solvers (Z3, CVC4, Yices, . . .)
- backend for nonreachability analysis (edge estimation, usable rules, . . .)
- maximal polynomial interpretations

Summary

- implementation of WPO in $\mathsf{T_TT_2}$
- certificates and a template mechanism for WPO
- employing external (state-of-the-art) SMT solvers (Z3, CVC4, Yices, ...)
- backend for nonreachability analysis (edge estimation, usable rules, ...)
- maximal polynomial interpretations

Future Work

- improve the strategy COMP & COMPCERT towards WPO and MAXPOLY

Summary

- implementation of WPO in $\mathsf{T_TT_2}$
- certificates and a template mechanism for WPO
- employing external (state-of-the-art) SMT solvers (Z3, CVC4, Yices, . . .)
- backend for nonreachability analysis (edge estimation, usable rules, . . .)
- maximal polynomial interpretations

Future Work

- improve the strategy COMP & COMPCERT towards WPO and MAXPOLY
- performance differences WPO($\mathcal{MS}um$)

Summary

- implementation of WPO in $\mathsf{T_TT_2}$
- certificates and a template mechanism for WPO
- employing external (state-of-the-art) SMT solvers (Z3, CVC4, Yices, ...)
- backend for nonreachability analysis (edge estimation, usable rules, ...)
- maximal polynomial interpretations

Future Work

- improve the strategy COMP & COMPCERT towards WPO and MAXPOLY
- performance differences WPO($\mathcal{MS}um$)
- combine matrix interpretations with WPO

Summary

- implementation of WPO in $\mathsf{T_TT_2}$
- certificates and a template mechanism for WPO
- employing external (state-of-the-art) SMT solvers (Z3, CVC4, Yices, ...)
- backend for nonreachability analysis (edge estimation, usable rules, ...)
- maximal polynomial interpretations

Future Work

- improve the strategy COMP & COMPCERT towards WPO and MAXPOLY
- performance differences WPO($\mathcal{MS}um$)
- combine matrix interpretations with WPO
- formalize maximal polynomial interpretations

Thank you for your attention!

References

Akihisa Yamada, Keiichirou Kusakari, and Toshiki Sakabe.
A Unified Ordering for Termination Proving.
Science of Computer Programming, 111:110–134, 2015.
doi: 10.1016/j.scico.2014.07.00.

Certificate

status & interpretation: $\mathsf{st}(\mathsf{h}) = [1,2]$ $\mathsf{h}_{\mathcal{A}}(x,y) = \max(2, 0+x, 0+y)$:

```
<precedenceStatusEntry>
  <name>h</name>
  <arity>2</arity>
  <precedence>0</precedence>
  <status>
    <position>1</position>
    <position>2</position>
  </status>
</precedenceStatusEntry>
```

```
<interpret>
  <name>h</name>
  <arity>2</arity>
  <max>
    <constant>2</constant>
    <sum>
      <constant>0</constant>
      <variable>1</variable>
    </sum>
    <sum>
      <constant>0</constant>
      <variable>2</variable>
    </sum>
  </max>
</interpret>
```

WPO as a Reduction Order

Consider a quasi-precedence $\gtrsim_{\mathcal{F}}$, a well-founded algebra $\mathcal{A}$ and a status function σ.
We define the relation $>_{\mathsf{WPO}(\mathcal{A},\sigma)}$ as follows:
For $s = f(s_1, \ldots, s_n) >_{\mathsf{WPO}(\mathcal{A},\sigma)} t$ iff

WPO as a Reduction Order

Consider a quasi-precedence $\gtrsim_{\mathcal{F}}$, a well-founded algebra $\mathcal{A}$ and a status function σ.
We define the relation $>_{\mathsf{WPO}(\mathcal{A},\sigma)}$ as follows:
For $s = f(s_1, \ldots, s_n) >_{\mathsf{WPO}(\mathcal{A},\sigma)} t$ iff

(1) $s >_{\mathcal{A}} t$, or

(2) $s \gtrsim_{\mathcal{A}} t$ and

WPO as a Reduction Order

Consider a quasi-precedence $\gtrsim_{\mathcal{F}}$, a well-founded algebra $\mathcal{A}$ and a status function σ.
We define the relation $>_{\mathsf{WPO}(\mathcal{A},\sigma)}$ as follows:
For $s = f(s_1, \ldots, s_n) >_{\mathsf{WPO}(\mathcal{A},\sigma)} t$ iff

(1) $s >_{\mathcal{A}} t$, or

(2) $s \gtrsim_{\mathcal{A}} t$ and

  (2.a) $\exists i \in \{1, \ldots, n\}.\ s_i \geq_{\mathsf{WPO}(\mathcal{A},\sigma)} t$, or
  (2.b) $t = g(t_1, \ldots, t_m)$, and $\forall j \in \{1, \ldots, m\}.\ s >_{\mathsf{WPO}(\mathcal{A},\sigma)} t_j$ and either

WPO as a Reduction Order

Consider a quasi-precedence $\gtrsim_{\mathcal{F}}$, a well-founded algebra $\mathcal{A}$ and a status function σ.
We define the relation $>_{\mathsf{WPO}(\mathcal{A},\sigma)}$ as follows:
For $s = f(s_1, \ldots, s_n) >_{\mathsf{WPO}(\mathcal{A},\sigma)} t$ iff

(1) $s >_{\mathcal{A}} t$, or

(2) $s \gtrsim_{\mathcal{A}} t$ and

  (2.a) $\exists i \in \{1, \ldots, n\}.\ s_i \geq_{\mathsf{WPO}(\mathcal{A},\sigma)} t$, or

  (2.b) $t = g(t_1, \ldots, t_m)$, and $\forall j \in \{1, \ldots, m\}.\ s >_{\mathsf{WPO}(\mathcal{A},\sigma)} t_j$ and either

- $f >_{\mathcal{F}} g$ or
- $f \sim_{\mathcal{F}} g$ and $[s_1, \ldots, s_n]^{\sigma(f)} >^{\mathsf{lex}}_{\mathsf{WPO}(\mathcal{A},\sigma)} [t_1, \ldots, t_m]^{\sigma(g)}$

WPO as a Reduction Order

Consider a quasi-precedence $\gtrsim_{\mathcal{F}}$, a well-founded algebra $\mathcal{A}$ and a status function σ.
We define the relation $>_{\mathsf{WPO}(\mathcal{A},\sigma)}$ as follows:
For $s = f(s_1, \ldots, s_n) >_{\mathsf{WPO}(\mathcal{A},\sigma)} t$ iff

(1) $s >_{\mathcal{A}} t$, or

(2) $s \gtrsim_{\mathcal{A}} t$ and

- (2.a) $\exists i \in \{1, \ldots, n\}.\ s_i \geq_{\mathsf{WPO}(\mathcal{A},\sigma)} t$, or
- (2.b) $t = g(t_1, \ldots, t_m)$, and $\forall j \in \{1, \ldots, m\}.\ s >_{\mathsf{WPO}(\mathcal{A},\sigma)} t_j$ and either
  - $f >_{\mathcal{F}} g$ or
  - $f \sim_{\mathcal{F}} g$ and $[s_1, \ldots, s_n]^{\sigma(f)} >^{\mathsf{lex}}_{\mathsf{WPO}(\mathcal{A},\sigma)} [t_1, \ldots, t_m]^{\sigma(g)}$

Theorem

If $\mathcal{A}$ is weakly monotone and weakly simple then $>_{\mathsf{WPO}(\mathcal{A})}$ is a reduction order

WPO as a Reduction Pair

Consider a quasi-precedence $\gtrsim_{\mathcal{F}}$, a well-founded algebra $\mathcal{A}$ and a partial status function σ. We define the relations $>_{\mathsf{WPO}(\mathcal{A},\sigma)}$ and $\gtrsim_{\mathsf{WPO}(\mathcal{A},\sigma)}$ as follows: For $s \in \mathcal{V}$, $s \gtrsim_{\mathsf{WPO}(\mathcal{A},\sigma)} t$ iff $s = t$. For $s = f(s_1, \dots, s_n) \;\overset{(\sim)}{>}_{\mathsf{WPO}(\mathcal{A},\sigma)}\; t$ iff

(1) $s >_{\mathcal{A}} t$, or

(2) $s \gtrsim_{\mathcal{A}} t$ and

- (2.a) $\exists i \in \sigma(f).\ s_i \gtrsim_{\mathsf{WPO}(\mathcal{A},\sigma)} t$, or
- (2.b) $t = g(t_1, \dots, t_m)$, and $\forall j \in \sigma(g).\ s >_{\mathsf{WPO}(\mathcal{A},\sigma)} t_j$ and either
  - (2.b.i) $f >_{\mathcal{F}} g$ or
  - (2.b.ii) $f \sim_{\mathcal{F}} g$ and $[s_1, \dots, s_n]^{\sigma(f)} \;\overset{(\sim)}{>}{}^{\mathsf{lex}}_{\mathsf{WPO}(\mathcal{A},\sigma)}\; [t_1, \dots, t_m]^{\sigma(g)}$

Theorem

If $\mathcal{A}$ is weakly monotone and weakly simple then $>_{\mathsf{WPO}(\mathcal{A})}$ is a reduction order

Maximal Polynomial Integer Interpretations

Consider a weight function $\langle \mathsf{w}, w_0 \rangle$ and a subterm penalty function sp where $\mathsf{sp}(f, i) \in \mathbb{Z}$. The maximal polynomial algebra looks as follows:

$$f_{\mathcal{A}}(a_1, \ldots, a_n) = \begin{cases} \max(0, \mathsf{w}_f + \sum_{i=1}^{n} \mathsf{sp}_{f,i} + a_{i\mathcal{A}}) & \text{if } \mathsf{ws}(f) = \mathsf{pol} \\ \max\left(\mathsf{w}_f, \max_{i=1}^{n}(\mathsf{sp}_{f,i} + a_{i\mathcal{A}})\right) & \text{if } \mathsf{ws}(f) = \mathsf{max} \end{cases}$$