# Term Rewriting

## Basic Concepts, Tools, and Applications

Sarah Winkler

Logical Perspectives Summer School
June 14–16 2021, Moscow

# Course Content

**Day 1**

abstract rewriting, properties of abstract rewrite systems, Newman's Lemma, term rewriting

**Day 2**

termination, polynomial interpretations, lexicographic path order, Knuth-Bendix order, derivational complexity

**Day 3**

critical pairs, confluence, orthogonality, Knuth-Bendix completion

# Outline

**Definition**

TRS $\mathcal{R}$ is terminating if $\quad \nexists\ t_1 \to_{\mathcal{R}} t_2 \to_{\mathcal{R}} t_3 \to_{\mathcal{R}} \dots$

## Definition

TRS $\mathcal{R}$ is terminating if $\nexists\ t_1 \to_{\mathcal{R}} t_2 \to_{\mathcal{R}} t_3 \to_{\mathcal{R}} \dots$

## Lemma

*TRS $\mathcal{R}$ is terminating $\iff$*
*$\exists$ well-founded order $>$ such that $s \to_{\mathcal{R}} t$ implies $s > t$*

## Definition

TRS $\mathcal{R}$ is terminating if $\nexists\ t_1 \to_{\mathcal{R}} t_2 \to_{\mathcal{R}} t_3 \to_{\mathcal{R}} \dots$

## Lemma

*TRS $\mathcal{R}$ is terminating $\iff$*
*$\exists$ well-founded order $>$ such that $s \to_{\mathcal{R}} t$ implies $s > t$*

## Example

- TRS

$$0 + y \to y \qquad\qquad \mathsf{s}(x) + y \to \mathsf{s}(x + y)$$

**Definition**

TRS $\mathcal{R}$ is terminating if $\nexists\ t_1 \to_{\mathcal{R}} t_2 \to_{\mathcal{R}} t_3 \to_{\mathcal{R}} \dots$

**Lemma**

*TRS $\mathcal{R}$ is terminating $\iff$*
*$\exists$ well-founded order $>$ such that $s \to_{\mathcal{R}} t$ implies $s > t$*

**Example**

- TRS

$$0 + y \to y \qquad\qquad \mathsf{s}(x) + y \to \mathsf{s}(x + y)$$

- well-founded order $>$ defined by mapping $[\cdot]$

$$s > t \quad\iff\quad [s] >_{\mathbb{N}} [t] \text{ with } [u] = \begin{cases} 1 & \text{if } u = 0 \\ [v] + 1 & \text{if } u = \mathsf{s}(v) \\ 2[v] + [w] & \text{if } u = v + w \\ 0 & \text{otherwise} \end{cases}$$

**Definition**

TRS $\mathcal{R}$ is terminating if $\nexists\ t_1 \to_{\mathcal{R}} t_2 \to_{\mathcal{R}} t_3 \to_{\mathcal{R}} \dots$

**Lemma**

*TRS $\mathcal{R}$ is terminating $\iff$*
*$\exists$ well-founded order $>$ such that $s \to_{\mathcal{R}} t$ implies $s > t$*

**Example**

- TRS

$$0 + y \to y \qquad\qquad \mathsf{s}(x) + y \to \mathsf{s}(x + y)$$

- well-founded order $>$ defined by mapping $[\cdot]$

$$s > t \quad\iff\quad [s] >_{\mathbb{N}} [t] \text{ with } [u] = \begin{cases} 1 & \text{if } u = 0 \\ [v] + 1 & \text{if } u = \mathsf{s}(v) \\ 2[v] + [w] & \text{if } u = v + w \\ 0 & \text{otherwise} \end{cases}$$

**Remark**

*infeasible to check all rewrite steps*

## Definition (Reduction Order)

- reduction order is proper order $>$ on terms which is

## Definition (Reduction Order)

- reduction order is proper order $>$ on terms which is
  - well-founded

## Definition (Reduction Order)

- reduction order is proper order $>$ on terms which is
  - well-founded
  - closed under contexts, i.e.
    $s > t \quad \Longrightarrow \quad C[\,s\,] > C[\,t\,]$ for all contexts $C$

## Definition (Reduction Order)

- reduction order is proper order $>$ on terms which is
  - well-founded
  - closed under contexts, i.e. (monotone)
    $s > t \implies C[\,s\,] > C[\,t\,]$ for all contexts $C$

## Definition (Reduction Order)

- reduction order is proper order $>$ on terms which is
  - well-founded
  - closed under contexts, i.e. (monotone)
    $s > t \implies C[\,s\,] > C[\,t\,]$ for all contexts $C$
  - closed under substitutions, i.e.
    $s > t \implies s\sigma > t\sigma$ for all substitution $\sigma$

## Definition (Reduction Order)

- reduction order is proper order $>$ on terms which is
  - well-founded
  - closed under contexts, i.e. (monotone)
    $s > t \implies C[\,s\,] > C[\,t\,]$ for all contexts $C$
  - closed under substitutions, i.e. (stable)
    $s > t \implies s\sigma > t\sigma$ for all substitution $\sigma$

## Definition (Reduction Order)

- reduction order is proper order $>$ on terms which is
  - well-founded
  - closed under contexts, i.e. (monotone)

    $s > t \implies C[\,s\,] > C[\,t\,]$ for all contexts $C$
  - closed under substitutions, i.e. (stable)

    $s > t \implies s\sigma > t\sigma$ for all substitution $\sigma$
- TRS $\mathcal{R}$ and reduction order $>$ are compatible
  if $\ell > r$ for all rules $\ell \to r$ in $\mathcal{R}$

## Definition (Reduction Order)

- reduction order is proper order $>$ on terms which is
  - well-founded
  - closed under contexts, i.e. (monotone)
    $s > t \implies C[\,s\,] > C[\,t\,]$ for all contexts $C$
  - closed under substitutions, i.e. (stable)
    $s > t \implies s\sigma > t\sigma$ for all substitution $\sigma$
- TRS $\mathcal{R}$ and reduction order $>$ are compatible
  if $\ell > r$ for all rules $\ell \to r$ in $\mathcal{R}$

## Theorem

*TRS $\mathcal{R}$ is terminating $\iff$ $\mathcal{R}$ is compatible with reduction order $>$*

## Definition (Reduction Order)

- reduction order is proper order $>$ on terms which is
  - well-founded
  - closed under contexts, i.e. (monotone)
    $s > t \implies C[\,s\,] > C[\,t\,]$ for all contexts $C$
  - closed under substitutions, i.e. (stable)
    $s > t \implies s\sigma > t\sigma$ for all substitution $\sigma$
- TRS $\mathcal{R}$ and reduction order $>$ are compatible
  if $\ell > r$ for all rules $\ell \to r$ in $\mathcal{R}$

## Theorem

*TRS $\mathcal{R}$ is terminating $\iff$ $\mathcal{R}$ is compatible with reduction order $>$*

## Proof.

$\implies$ $\mathcal{R}$ is compatible with reduction order $\to_{\mathcal{R}}^{+}$

$\impliedby$ by reduction order have strict $>$-decrease in every rewrite step $\square$

## Definitions

- $\mathcal{F}$-algebra $\mathcal{A} = (A, \{ f_{\mathcal{A}} \}_{f \in \mathcal{F}})$ consists of
  - carrier $A$
  - interpretation $f_{\mathcal{A}} \colon A^n \to A$ for every $f \in \mathcal{F}$ of arity $n$

## Definitions

- $\mathcal{F}$-algebra $\mathcal{A} = (A, \{ f_{\mathcal{A}} \}_{f \in \mathcal{F}})$ consists of
  - carrier $A$
  - interpretation $f_{\mathcal{A}} \colon A^n \to A$ for every $f \in \mathcal{F}$ of arity $n$
- assignment $\alpha \colon \mathcal{V} \to A$

## Definitions

- $\mathcal{F}$-algebra $\mathcal{A} = (A, \{ f_{\mathcal{A}} \}_{f \in \mathcal{F}})$ consists of
  - carrier $\quad A$
  - interpretation $\quad f_{\mathcal{A}} \colon A^n \to A \quad$ for every $f \in \mathcal{F}$ of arity $n$
- assignment $\quad \alpha \colon \mathcal{V} \to A$
- interpretation function $\quad [\alpha]_{\mathcal{A}}(\,\cdot\,) \colon \mathcal{T}(\mathcal{F}, \mathcal{V}) \to A$

## Definitions

- $\mathcal{F}$-algebra $\mathcal{A} = (A, \{ f_{\mathcal{A}} \}_{f \in \mathcal{F}})$ consists of
  - carrier $A$
  - interpretation $f_{\mathcal{A}} \colon A^n \to A$ for every $f \in \mathcal{F}$ of arity $n$
- assignment $\alpha \colon \mathcal{V} \to A$
- interpretation function $[\alpha]_{\mathcal{A}}(\,\cdot\,) \colon \mathcal{T}(\mathcal{F}, \mathcal{V}) \to A$

$$[\alpha]_{\mathcal{A}}(t) = \begin{cases} \alpha(t) & \text{if } t \in \mathcal{V} \\ & \end{cases}$$

## Definitions

- $\mathcal{F}$-algebra $\mathcal{A} = (A, \{ f_{\mathcal{A}} \}_{f \in \mathcal{F}})$ consists of
  - carrier $A$
  - interpretation $f_{\mathcal{A}} \colon A^n \to A$ for every $f \in \mathcal{F}$ of arity $n$
- assignment $\alpha \colon \mathcal{V} \to A$
- interpretation function $[\alpha]_{\mathcal{A}}(\,\cdot\,) \colon \mathcal{T}(\mathcal{F}, \mathcal{V}) \to A$

$$[\alpha]_{\mathcal{A}}(t) = \begin{cases} \alpha(t) & \text{if } t \in \mathcal{V} \\ f_{\mathcal{A}}([\alpha]_{\mathcal{A}}(t_1), \ldots, [\alpha]_{\mathcal{A}}(t_n)) & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

## Definitions

- $\mathcal{F}$-algebra $\mathcal{A} = (A, \{ f_{\mathcal{A}} \}_{f \in \mathcal{F}})$ consists of
  - carrier $A$
  - interpretation $f_{\mathcal{A}} \colon A^n \to A$ for every $f \in \mathcal{F}$ of arity $n$
- assignment $\alpha \colon \mathcal{V} \to A$
- interpretation function $[\alpha]_{\mathcal{A}}(\,\cdot\,) \colon \mathcal{T}(\mathcal{F}, \mathcal{V}) \to A$

$$[\alpha]_{\mathcal{A}}(t) = \begin{cases} \alpha(t) & \text{if } t \in \mathcal{V} \\ f_{\mathcal{A}}([\alpha]_{\mathcal{A}}(t_1), \ldots, [\alpha]_{\mathcal{A}}(t_n)) & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

## Example

two $\{ 0, \mathsf{s}, + \}$- algebras:

- $\mathcal{A} = (\mathbb{N}, \{0_{\mathcal{A}}, \mathsf{s}_{\mathcal{A}}, +_{\mathcal{A}}\})$ with

  $0_{\mathcal{A}} = 0$ $\quad \mathsf{s}_{\mathcal{A}}(x) = x + 1$ $\quad +_{\mathcal{A}}(x, y) = x + y$

## Definitions

- $\mathcal{F}$-algebra $\mathcal{A} = (A, \{ f_{\mathcal{A}} \}_{f \in \mathcal{F}})$ consists of
  - carrier $A$
  - interpretation $f_{\mathcal{A}} \colon A^n \to A$ for every $f \in \mathcal{F}$ of arity $n$
- assignment $\alpha \colon \mathcal{V} \to A$
- interpretation function $[\alpha]_{\mathcal{A}}(\,\cdot\,) \colon \mathcal{T}(\mathcal{F}, \mathcal{V}) \to A$

$$[\alpha]_{\mathcal{A}}(t) = \begin{cases} \alpha(t) & \text{if } t \in \mathcal{V} \\ f_{\mathcal{A}}([\alpha]_{\mathcal{A}}(t_1), \ldots, [\alpha]_{\mathcal{A}}(t_n)) & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

## Example

two $\{ 0, \mathsf{s}, + \}$- algebras:

- $\mathcal{A} = (\mathbb{N}, \{0_{\mathcal{A}}, \mathsf{s}_{\mathcal{A}}, +_{\mathcal{A}}\})$ with
  $0_{\mathcal{A}} = 0$ $\mathsf{s}_{\mathcal{A}}(x) = x + 1$ $+_{\mathcal{A}}(x, y) = x + y$
- $\mathcal{B} = (\{\blacklozenge, \spadesuit, \clubsuit, \heartsuit\}, \{ 0_{\mathcal{B}}, \mathsf{s}_{\mathcal{B}}, +_{\mathcal{B}} \})$ with
  $0_{\mathcal{B}} = \spadesuit$ $\mathsf{s}_{\mathcal{B}}(x) = \heartsuit$ if $x = \heartsuit$ else $\clubsuit$ $+_{\mathcal{B}}(x, y) = \blacklozenge$

**Definition (Well-Founded Monotone Algebra (WFMA))**

WFMA is algebra $\mathcal{A} = (A, \{ f_{\mathcal{A}} \}_{f \in \mathcal{F}})$ with well-founded order $>$ on $A$

## Definition (Well-Founded Monotone Algebra (WFMA))

WFMA is algebra $\mathcal{A} = (A, \{\, f_{\mathcal{A}} \,\}_{f \in \mathcal{F}})$ with well-founded order $>$ on $A$ such that every $f_{\mathcal{A}}$ is strictly monotone in all arguments:

$$f_{\mathcal{A}}(a_1,\ \ldots,\ a_i,\ \ldots,\ a_n) > f_{\mathcal{A}}(a_1,\ \ldots,\ b,\ \ldots,\ a_n)$$

for all $a_1, \ldots, a_n, b \in A$ and $i \in \{1,\ \ldots,\ n\}$ with $a_i > b$

## Definition (Well-Founded Monotone Algebra (WFMA))

WFMA is algebra $\mathcal{A} = (A, \{\, f_{\mathcal{A}} \,\}_{f \in \mathcal{F}})$ with well-founded order $>$ on $A$ such that every $f_{\mathcal{A}}$ is strictly monotone in all arguments:

$$f_{\mathcal{A}}(a_1,\, \ldots,\, a_i,\, \ldots,\, a_n) > f_{\mathcal{A}}(a_1,\, \ldots,\, b,\, \ldots,\, a_n)$$

for all $a_1, \ldots, a_n, b \in A$ and $i \in \{1,\, \ldots,\, n\}$ with $a_i > b$

## Definition ($>_{\mathcal{A}}$)

relation $>_{\mathcal{A}}$ on terms defined as $s >_{\mathcal{A}} t$ iff $[\alpha]_{\mathcal{A}}(s) > [\alpha]_{\mathcal{A}}(t) \quad \forall\, \alpha$

## Definition (Well-Founded Monotone Algebra (WFMA))

WFMA is algebra $\mathcal{A} = (A, \{\, f_\mathcal{A} \,\}_{f \in \mathcal{F}})$ with well-founded order $>$ on $A$ such that every $f_\mathcal{A}$ is strictly monotone in all arguments:

$$f_\mathcal{A}(a_1, \ldots, a_i, \ldots, a_n) > f_\mathcal{A}(a_1, \ldots, b, \ldots, a_n)$$

for all $a_1, \ldots, a_n, b \in A$ and $i \in \{1, \ldots, n\}$ with $a_i > b$

## Definition ($>_\mathcal{A}$)

relation $>_\mathcal{A}$ on terms defined as $s >_\mathcal{A} t$ iff $[\alpha]_\mathcal{A}(s) > [\alpha]_\mathcal{A}(t) \quad \forall\, \alpha$

## Lemma

*$>_\mathcal{A}$ is reduction order for every well-founded monotone algebra $(\mathcal{A}, >)$*

## Definition (Well-Founded Monotone Algebra (WFMA))

WFMA is algebra $\mathcal{A} = (A, \{ f_\mathcal{A} \}_{f \in \mathcal{F}})$ with well-founded order $>$ on $A$ such that every $f_\mathcal{A}$ is strictly monotone in all arguments:

$$f_\mathcal{A}(a_1, \ldots, a_i, \ldots, a_n) > f_\mathcal{A}(a_1, \ldots, b, \ldots, a_n)$$

for all $a_1, \ldots, a_n, b \in A$ and $i \in \{1, \ldots, n\}$ with $a_i > b$

## Definition ($>_\mathcal{A}$)

relation $>_\mathcal{A}$ on terms defined as $s >_\mathcal{A} t$ iff $[\alpha]_\mathcal{A}(s) > [\alpha]_\mathcal{A}(t) \quad \forall\, \alpha$

## Lemma

$>_\mathcal{A}$ *is reduction order for every well-founded monotone algebra* $(\mathcal{A}, >)$

## Theorem

*TRS* $\mathcal{R}$ *is terminating* $\iff$ $\mathcal{R} \subseteq >_\mathcal{A}$ *for WFMA* $(\mathcal{A}, >)$

**Definition (Well-Founded Monotone Algebra (WFMA))**

WFMA is algebra $\mathcal{A} = (A, \{ f_\mathcal{A} \}_{f \in \mathcal{F}})$ with well-founded order $>$ on $A$ such that every $f_\mathcal{A}$ is strictly monotone in all arguments:

$$f_\mathcal{A}(a_1, \ldots, a_i, \ldots, a_n) > f_\mathcal{A}(a_1, \ldots, b, \ldots, a_n)$$

for all $a_1, \ldots, a_n, b \in A$ and $i \in \{1, \ldots, n\}$ with $a_i > b$

**Definition ($>_\mathcal{A}$)**

relation $>_\mathcal{A}$ on terms defined as $s >_\mathcal{A} t$ iff $[\alpha]_\mathcal{A}(s) > [\alpha]_\mathcal{A}(t) \quad \forall\, \alpha$

**Lemma**

$>_\mathcal{A}$ *is reduction order for every well-founded monotone algebra* $(\mathcal{A}, >)$

**Theorem**

*TRS* $\mathcal{R}$ *is terminating* $\iff$ $\mathcal{R} \subseteq >_\mathcal{A}$ *for WFMA* $(\mathcal{A}, >)$

**Proof.**

$\Longleftarrow \quad \mathcal{R}$ is compatible with reduction order $>_\mathcal{A}$

## Definition (Well-Founded Monotone Algebra (WFMA))

WFMA is algebra $\mathcal{A} = (A, \{\, f_\mathcal{A} \,\}_{f \in \mathcal{F}})$ with well-founded order $>$ on $A$ such that every $f_\mathcal{A}$ is strictly monotone in all arguments:

$$f_\mathcal{A}(a_1, \ldots, a_i, \ldots, a_n) > f_\mathcal{A}(a_1, \ldots, b, \ldots, a_n)$$

for all $a_1, \ldots, a_n, b \in A$ and $i \in \{1, \ldots, n\}$ with $a_i > b$

## Definition ($>_\mathcal{A}$)

relation $>_\mathcal{A}$ on terms defined as $s >_\mathcal{A} t$ iff $[\alpha]_\mathcal{A}(s) > [\alpha]_\mathcal{A}(t) \quad \forall\, \alpha$

## Lemma

$>_\mathcal{A}$ *is reduction order for every well-founded monotone algebra* $(\mathcal{A}, >)$

## Theorem

*TRS* $\mathcal{R}$ *is terminating* $\iff$ $\mathcal{R} \subseteq >_\mathcal{A}$ *for WFMA* $(\mathcal{A}, >)$

## Proof.

$\Longleftarrow$ $\mathcal{R}$ is compatible with reduction order $>_\mathcal{A}$

$\Longrightarrow$ term algebra $(\mathcal{T}(\mathcal{F}, \mathcal{V}), \to_\mathcal{R}^+)$ is WFMA

$\square$

**Definition (Polynomial Termination)**

TRS $\mathcal{R}$ is polynomially terminating (over $\mathbb{N}$) if $\mathcal{R} \subseteq >_{\mathcal{A}}$ for well-founded monotone algebra $(\mathcal{A}, >)$ such that

- carrier of $\mathcal{A}$ is $\mathbb{N}$
- $>$ is standard order on $\mathbb{N}$
- $f_{\mathcal{A}} \in \mathbb{Z}[x_1, \ldots, x_n]$ for every $n$-ary $f$

**Definition (Polynomial Termination)**

TRS $\mathcal{R}$ is polynomially terminating (over $\mathbb{N}$) if $\mathcal{R} \subseteq >_{\mathcal{A}}$ for well-founded monotone algebra $(\mathcal{A}, >)$ such that

- carrier of $\mathcal{A}$ is $\mathbb{N}$
- $>$ is standard order on
- $f_{\mathcal{A}} \in \mathbb{Z}[x_1, \ldots, x_n]$ for every $n$-ary $f$

polynomials with indeterminates $x_1, \ldots, x_n$ and coefficients in $\mathbb{Z}$

**Definition (Polynomial Termination)**

TRS $\mathcal{R}$ is polynomially terminating (over $\mathbb{N}$) if $\mathcal{R} \subseteq >_{\mathcal{A}}$ for well-founded monotone algebra $(\mathcal{A}, >)$ such that

- carrier of $\mathcal{A}$ is $\mathbb{N}$
- $>$ is standard order on
- $f_{\mathcal{A}} \in \mathbb{Z}[x_1, \ldots, x_n]$ for every $n$-ary $f$

polynomials with indeterminates $x_1, \ldots, x_n$ and coefficients in $\mathbb{Z}$

**Definition (Polynomial Termination)**

TRS $\mathcal{R}$ is polynomially terminating (over $\mathbb{N}$) if $\mathcal{R} \subseteq >_{\mathcal{A}}$ for well-founded monotone algebra $(\mathcal{A}, >)$ such that

- carrier of $\mathcal{A}$ is $\mathbb{N}$
- $>$ is standard order on $\mathbb{N}$
- $f_{\mathcal{A}} \in \mathbb{Z}[x_1, \ldots, x_n]$ for every $n$-ary $f$

**Lemma**

*$\mathcal{R}$ is polynomially terminating over $\mathbb{N}$*

$$\iff$$

*$\mathcal{R}$ is polynomially terminating over $\{ n \in \mathbb{N} \mid n \geqslant N \}$ for some $N \geqslant 0$*

## Example

- TRS

  $0 + y \to y \quad \mathsf{s}(x) + y \to \mathsf{s}(x + y) \quad 0 \times y \to 0 \quad \mathsf{s}(x) \times y \to y + (x \times y)$

## Example

- TRS

  $0 + y \to y \quad \mathsf{s}(x) + y \to \mathsf{s}(x + y) \quad 0 \times y \to 0 \quad \mathsf{s}(x) \times y \to y + (x \times y)$

- interpretations in $\mathbb{N}$

$$\begin{aligned} 0_{\mathcal{A}} &= 1 & +_{\mathcal{A}}(x, y) &= 2x + y \\ \mathsf{s}_{\mathcal{A}}(x) &= x + 1 & \times_{\mathcal{A}}(x, y) &= 2xy + x + y + 1 \end{aligned}$$

## Example

- TRS

$$0 + y \to y \quad \mathsf{s}(x) + y \to \mathsf{s}(x + y) \quad 0 \times y \to 0 \quad \mathsf{s}(x) \times y \to y + (x \times y)$$

- interpretations in $\mathbb{N}$

$$0_{\mathcal{A}} = 1 \qquad +_{\mathcal{A}}(x, y) = 2x + y$$
$$\mathsf{s}_{\mathcal{A}}(x) = x + 1 \qquad \times_{\mathcal{A}}(x, y) = 2xy + x + y + 1$$

- constraints $\quad \forall\, x,\ y \in \mathbb{N}$

$$y + 2 > y \qquad 2x + y + 2 > 2x + y + 1$$
$$3y + 2 > 1 \qquad 2xy + x + 3y + 2 > 2xy + x + 3y + 1$$

## Example

- TRS

$$0 + y \to y \quad \mathsf{s}(x) + y \to \mathsf{s}(x + y) \quad 0 \times y \to 0 \quad \mathsf{s}(x) \times y \to y + (x \times y)$$

- interpretations in $\mathbb{N}$

$$\begin{aligned} 0_{\mathcal{A}} &= 1 & \qquad +_{\mathcal{A}}(x, y) &= 2x + y \\ \mathsf{s}_{\mathcal{A}}(x) &= x + 1 & \qquad \times_{\mathcal{A}}(x, y) &= 2xy + x + y + 1 \end{aligned}$$

- constraints $\quad \forall\, x,\ y \in \mathbb{N}$

$$\begin{aligned} 2 &> 0 & \qquad 1 &> 0 \\ 3y + 1 &> 0 & \qquad 1 &> 0 \end{aligned}$$

## Example

- TRS

  $0 + y \to y \quad \mathsf{s}(x) + y \to \mathsf{s}(x + y) \quad 0 \times y \to 0 \quad \mathsf{s}(x) \times y \to y + (x \times y)$

- interpretations in $\mathbb{N}$

$$
\begin{aligned}
0_{\mathcal{A}} &= 1 & \qquad +_{\mathcal{A}}(x, y) &= 2x + y \\
\mathsf{s}_{\mathcal{A}}(x) &= x + 1 & \qquad \times_{\mathcal{A}}(x, y) &= 2xy + x + y + 1
\end{aligned}
$$

- constraints $\quad \forall\, x,\ y \in \mathbb{N}$

$$
\begin{aligned}
2 &> 0 & \qquad 1 &> 0 \\
3y + 1 &> 0 & \qquad 1 &> 0
\end{aligned}
$$

- $\mathsf{s}(0) \times \mathsf{s}(\mathsf{s}(0)) \;\to\; \mathsf{s}(\mathsf{s}(0)) + (0 \times \mathsf{s}(\mathsf{s}(0))) \;\to\; \mathsf{s}(\mathsf{s}(0)) + 0 \;\to\; \mathsf{s}(\mathsf{s}(0) + 0)$

$$\to \quad \mathsf{s}(\mathsf{s}(0 + 0)) \quad \to \quad \mathsf{s}(\mathsf{s}(0))$$

## Example

- TRS

$$0 + y \to y \quad \mathsf{s}(x) + y \to \mathsf{s}(x + y) \quad 0 \times y \to 0 \quad \mathsf{s}(x) \times y \to y + (x \times y)$$

- interpretations in $\mathbb{N}$

$$\begin{aligned} 0_{\mathcal{A}} &= 1 & +_{\mathcal{A}}(x, y) &= 2x + y \\ \mathsf{s}_{\mathcal{A}}(x) &= x + 1 & \times_{\mathcal{A}}(x, y) &= 2xy + x + y + 1 \end{aligned}$$

- constraints $\quad \forall\, x,\ y \in \mathbb{N}$

$$\begin{aligned} 2 &> 0 & 1 &> 0 \\ 3y + 1 &> 0 & 1 &> 0 \end{aligned}$$

- $$\begin{array}{ccccccc} \mathsf{s}(0) \times \mathsf{s}(\mathsf{s}(0)) & \to & \mathsf{s}(\mathsf{s}(0)) + (0 \times \mathsf{s}(\mathsf{s}(0))) & \to & \mathsf{s}(\mathsf{s}(0)) + 0 & \to & \mathsf{s}(\mathsf{s}(0) + 0) \\ 18 & > & 17 & > & 7 & > & 6 \\ & \to & \mathsf{s}(\mathsf{s}(0 + 0)) & \to & \mathsf{s}(\mathsf{s}(0)) & & \\ & > & 5 & > & 3 & & \end{array}$$

# Outline

Termination

- Interpretations
- Lexicographic Path Order
- Knuth-Bendix Order

Complexity

## Definition (Precedence)

precedence is strict order $>$ on $\mathcal{F}$

## Definition (Precedence)

precedence is strict order $>$ on $\mathcal{F}$

## Definition (Lexicographic Path Order)

relation $>_{\mathsf{lpo}}$ on terms defined by

## Definition (Precedence)

precedence is strict order $>$ on $\mathcal{F}$

## Definition (Lexicographic Path Order)

relation $>_{\mathsf{lpo}}$ on terms defined by $s >_{\mathsf{lpo}} t$ if $s = f(s_1, \ldots, s_n)$ and either

## Definition (Precedence)

precedence is strict order $>$ on $\mathcal{F}$

## Definition (Lexicographic Path Order)

relation $>_{\mathsf{lpo}}$ on terms defined by $s >_{\mathsf{lpo}} t$ if $s = f(s_1, \ldots, s_n)$ and either

1. $\exists\, i \quad s_i >_{\mathsf{lpo}} t$ or $s_i = t$

## Definition (Precedence)

precedence is strict order $>$ on $\mathcal{F}$

## Definition (Lexicographic Path Order)

relation $>_{\mathsf{lpo}}$ on terms defined by $s >_{\mathsf{lpo}} t$ if $s = f(s_1, \ldots, s_n)$ and either

1. $\exists\, i \quad s_i >_{\mathsf{lpo}} t$ or $s_i = t$
2. $t = g(t_1, \ldots, t_m)$ and $f > g$ and $\forall\, j \quad s >_{\mathsf{lpo}} t_j$

### Definition (Precedence)

precedence is strict order $>$ on $\mathcal{F}$

### Definition (Lexicographic Path Order)

relation $>_{\mathsf{lpo}}$ on terms defined by $s >_{\mathsf{lpo}} t$ if $s = f(s_1, \ldots, s_n)$ and either

1. $\exists\, i \quad s_i >_{\mathsf{lpo}} t$ or $s_i = t$
2. $t = g(t_1, \ldots, t_m)$ and $f > g$ and $\forall\, j \quad s >_{\mathsf{lpo}} t_j$

## Definition (Precedence)

precedence is strict order $>$ on $\mathcal{F}$

## Definition (Lexicographic Path Order)

relation $>_{\mathsf{lpo}}$ on terms defined by $s >_{\mathsf{lpo}} t$ if $s = f(s_1, \dots, s_n)$ and either

1. $\exists\, i \quad s_i >_{\mathsf{lpo}} t$ or $s_i = t$
2. $t = g(t_1, \dots, t_m)$ and $f > g$ and $\forall\, j \quad s >_{\mathsf{lpo}} t_j$

## Definition (Precedence)

precedence is strict order $>$ on $\mathcal{F}$

## Definition (Lexicographic Path Order)

relation $>_{\mathsf{lpo}}$ on terms defined by $s >_{\mathsf{lpo}} t$ if $s = f(s_1, \ldots, s_n)$ and either

1. $\exists\, i \quad s_i >_{\mathsf{lpo}} t$ or $s_i = t$
2. $t = g(t_1, \ldots, t_m)$ and $f > g$ and $\forall\, j \quad s >_{\mathsf{lpo}} t_j$
3. $t = f(t_1, \ldots, t_n)$ and $\exists\, i$ such that
   - $\forall\, j < i \quad s_j = t_j$
   - $s_i >_{\mathsf{lpo}} t_i$
   - $\forall\, j > i \quad s >_{\mathsf{lpo}} t_j$

## Definition (Precedence)

precedence is strict order $>$ on $\mathcal{F}$

## Definition (Lexicographic Path Order)

relation $>_{\mathsf{lpo}}$ on terms defined by $s >_{\mathsf{lpo}} t$ if $s = f(s_1, \ldots, s_n)$ and either

1. $\exists\, i \quad s_i >_{\mathsf{lpo}} t$ or $s_i = t$
2. $t = g(t_1, \ldots, t_m)$ and $f > g$ and $\forall\, j \quad s >_{\mathsf{lpo}} t_j$
3. $t = f(t_1, \ldots, t_n)$ and $\exists\, i$ such that
   - $\forall\, j < i \quad s_j = t_j$
   - $s_i >_{\mathsf{lpo}} t_i$
   - $\forall\, j > i \quad s >_{\mathsf{lpo}} t_j$

## Definition (Precedence)

precedence is strict order $>$ on $\mathcal{F}$

## Definition (Lexicographic Path Order)

relation $>_{\mathsf{lpo}}$ on terms defined by $s >_{\mathsf{lpo}} t$ if $s = f(s_1, \dots, s_n)$ and either

1. $\exists\, i \quad s_i >_{\mathsf{lpo}} t$ or $s_i = t$
2. $t = g(t_1, \dots, t_m)$ and $f > g$ and $\forall\, j \quad s >_{\mathsf{lpo}} t_j$
3. $t = f(t_1, \dots, t_n)$ and $\exists\, i$ such that
   - $\forall\, j < i \quad s_j = t_j$
   - $s_i >_{\mathsf{lpo}} t_i$
   - $\forall\, j > i \quad s >_{\mathsf{lpo}} t_j$

## Definition (Precedence)

precedence is strict order $>$ on $\mathcal{F}$

## Definition (Lexicographic Path Order)

relation $>_{\mathsf{lpo}}$ on terms defined by $s >_{\mathsf{lpo}} t$ if $s = f(s_1, \dots, s_n)$ and either

1. $\exists\, i \quad s_i >_{\mathsf{lpo}} t$ or $s_i = t$
2. $t = g(t_1, \dots, t_m)$ and $f > g$ and $\forall\, j \quad s >_{\mathsf{lpo}} t_j$
3. $t = f(t_1, \dots, t_n)$ and $\exists\, i$ such that
   - $\forall\, j < i \quad s_j = t_j$
   - $s_i >_{\mathsf{lpo}} t_i$
   - $\forall\, j > i \quad s >_{\mathsf{lpo}} t_j$

## Definition (Precedence)

precedence is strict order $>$ on $\mathcal{F}$

## Definition (Lexicographic Path Order)

relation $>_{\mathsf{lpo}}$ on terms defined by $s >_{\mathsf{lpo}} t$ if $s = f(s_1, \dots, s_n)$ and either

1. $\exists\, i \quad s_i >_{\mathsf{lpo}} t$ or $s_i = t$
2. $t = g(t_1, \dots, t_m)$ and $f > g$ and $\forall\, j \quad s >_{\mathsf{lpo}} t_j$
3. $t = f(t_1, \dots, t_n)$ and $\exists\, i$ such that
   - $\forall\, j < i \quad s_j = t_j$
   - $s_i >_{\mathsf{lpo}} t_i$
   - $\forall\, j > i \quad s >_{\mathsf{lpo}} t_j$

## Example

$$\mathsf{s}(x) \times y >^{?}_{\mathsf{lpo}} (x \times y) + y$$

## Definition (Precedence)

precedence is strict order $>$ on $\mathcal{F}$

## Definition (Lexicographic Path Order)

relation $>_{\mathsf{lpo}}$ on terms defined by $s >_{\mathsf{lpo}} t$ if $s = f(s_1, \dots, s_n)$ and either

1. $\exists\, i \quad s_i >_{\mathsf{lpo}} t$ or $s_i = t$
2. $t = g(t_1, \dots, t_m)$ and $f > g$ and $\forall\, j \quad s >_{\mathsf{lpo}} t_j$
3. $t = f(t_1, \dots, t_n)$ and $\exists\, i$ such that
   - $\forall\, j < i \quad s_j = t_j$
   - $s_i >_{\mathsf{lpo}} t_i$
   - $\forall\, j > i \quad s >_{\mathsf{lpo}} t_j$

### Example

$$\mathsf{s}(x) \times y >^{?}_{\mathsf{lpo}} (x \times y) + y$$

## Definition (Precedence)

precedence is strict order $>$ on $\mathcal{F}$

## Definition (Lexicographic Path Order)

relation $>_{\mathsf{lpo}}$ on terms defined by $s >_{\mathsf{lpo}} t$ if $s = f(s_1, \ldots, s_n)$ and either

1. $\exists\, i \quad s_i >_{\mathsf{lpo}} t$ or $s_i = t$
2. $t = g(t_1, \ldots, t_m)$ and $f > g$ and $\forall\, j \quad s >_{\mathsf{lpo}} t_j$
3. $t = f(t_1, \ldots, t_n)$ and $\exists\, i$ such that
   - $\forall\, j < i \quad s_j = t_j$
   - $s_i >_{\mathsf{lpo}} t_i$
   - $\forall\, j > i \quad s >_{\mathsf{lpo}} t_j$

### Example

$$\mathsf{s}(x) \times y >^{?}_{\mathsf{lpo}} (x \times y) + y$$

$\times > +$ $\qquad$ $\mathsf{s}(x) \times y >^{?}_{\mathsf{lpo}} x \times y$ $\qquad$ $\mathsf{s}(x) \times y >^{?}_{\mathsf{lpo}} y$

## Definition (Precedence)

precedence is strict order $>$ on $\mathcal{F}$

## Definition (Lexicographic Path Order)

relation $>_{\mathsf{lpo}}$ on terms defined by $s >_{\mathsf{lpo}} t$ if $s = f(s_1, \ldots, s_n)$ and either

1. $\exists\, i \quad s_i >_{\mathsf{lpo}} t$ or $s_i = t$
2. $t = g(t_1, \ldots, t_m)$ and $f > g$ and $\forall\, j \quad s >_{\mathsf{lpo}} t_j$
3. $t = f(t_1, \ldots, t_n)$ and $\exists\, i$ such that
   - $\forall\, j < i \quad s_j = t_j$
   - $s_i >_{\mathsf{lpo}} t_i$
   - $\forall\, j > i \quad s >_{\mathsf{lpo}} t_j$

### Example

$$\mathsf{s}(x) \times y >^{?}_{\mathsf{lpo}} (x \times y) + y$$

$$\times > + \qquad \mathsf{s}(x) \times y >^{?}_{\mathsf{lpo}} x \times y \qquad \mathsf{s}(x) \times y >^{?}_{\mathsf{lpo}} y$$

## Definition (Precedence)

precedence is strict order $>$ on $\mathcal{F}$

## Definition (Lexicographic Path Order)

relation $>_{\mathsf{lpo}}$ on terms defined by $s >_{\mathsf{lpo}} t$ if $s = f(s_1, \ldots, s_n)$ and either

1. $\exists\, i \quad s_i >_{\mathsf{lpo}} t$ or $s_i = t$
2. $t = g(t_1, \ldots, t_m)$ and $f > g$ and $\forall\, j \quad s >_{\mathsf{lpo}} t_j$
3. $t = f(t_1, \ldots, t_n)$ and $\exists\, i$ such that
   - $\forall\, j < i \quad s_j = t_j$
   - $s_i >_{\mathsf{lpo}} t_i$
   - $\forall\, j > i \quad s >_{\mathsf{lpo}} t_j$

### Example

$$\mathsf{s}(x) \times y >^{?}_{\mathsf{lpo}} (x \times y) + y$$

$$\times > + \qquad \mathsf{s}(x) \times y >^{?}_{\mathsf{lpo}} x \times y \qquad \mathsf{s}(x) \times y >^{?}_{\mathsf{lpo}} y$$

## Definition (Precedence)

precedence is strict order $>$ on $\mathcal{F}$

## Definition (Lexicographic Path Order)

relation $>_{\mathsf{lpo}}$ on terms defined by $s >_{\mathsf{lpo}} t$ if $s = f(s_1, \ldots, s_n)$ and either

1. $\exists\, i \quad s_i >_{\mathsf{lpo}} t$ or $s_i = t$
2. $t = g(t_1, \ldots, t_m)$ and $f > g$ and $\forall\, j \quad s >_{\mathsf{lpo}} t_j$
3. $t = f(t_1, \ldots, t_n)$ and $\exists\, i$ such that
   - $\forall\, j < i \quad s_j = t_j$
   - $s_i >_{\mathsf{lpo}} t_i$
   - $\forall\, j > i \quad s >_{\mathsf{lpo}} t_j$

### Example

$$\mathsf{s}(x) \times y >^?_{\mathsf{lpo}} (x \times y) + y$$

$$\times > + \qquad \mathsf{s}(x) \times y >^?_{\mathsf{lpo}} x \times y \qquad \mathsf{s}(x) \times y >^?_{\mathsf{lpo}} y$$

$$\mathsf{s}(x) >^?_{\mathsf{lpo}} x \qquad \mathsf{s}(x) \times y >^?_{\mathsf{lpo}} y$$

## Definition (Precedence)

precedence is strict order $>$ on $\mathcal{F}$

## Definition (Lexicographic Path Order)

relation $>_{\mathsf{lpo}}$ on terms defined by $s >_{\mathsf{lpo}} t$ if $s = f(s_1, \ldots, s_n)$ and either

1. $\exists\, i \quad s_i >_{\mathsf{lpo}} t$ or $s_i = t$
2. $t = g(t_1, \ldots, t_m)$ and $f > g$ and $\forall\, j \quad s >_{\mathsf{lpo}} t_j$
3. $t = f(t_1, \ldots, t_n)$ and $\exists\, i$ such that
   - $\forall\, j < i \quad s_j = t_j$
   - $s_i >_{\mathsf{lpo}} t_i$
   - $\forall\, j > i \quad s >_{\mathsf{lpo}} t_j$

## Example

$$\mathsf{s}(x) \times y >^?_{\mathsf{lpo}} (x \times y) + y$$

$$\times > + \qquad \mathsf{s}(x) \times y >^?_{\mathsf{lpo}} x \times y \qquad \mathsf{s}(x) \times y >^?_{\mathsf{lpo}} y$$

$$\mathsf{s}(x) >^?_{\mathsf{lpo}} x \qquad \mathsf{s}(x) \times y >^?_{\mathsf{lpo}} y$$

## Definition (Precedence)

precedence is strict order $>$ on $\mathcal{F}$

## Definition (Lexicographic Path Order)

relation $>_{\mathsf{lpo}}$ on terms defined by $s >_{\mathsf{lpo}} t$ if $s = f(s_1, \dots, s_n)$ and either

1. $\exists\, i \quad s_i >_{\mathsf{lpo}} t$ or $s_i = t$
2. $t = g(t_1, \dots, t_m)$ and $f > g$ and $\forall\, j \quad s >_{\mathsf{lpo}} t_j$
3. $t = f(t_1, \dots, t_n)$ and $\exists\, i$ such that
   - $\forall\, j < i \quad s_j = t_j$
   - $s_i >_{\mathsf{lpo}} t_i$
   - $\forall\, j > i \quad s >_{\mathsf{lpo}} t_j$

### Example

$$\mathsf{s}(x) \times y >^?_{\mathsf{lpo}} (x \times y) + y$$

$\times > +$ $\qquad$ $\mathsf{s}(x) \times y >^?_{\mathsf{lpo}} x \times y$ $\qquad$ $\mathsf{s}(x) \times y >^?_{\mathsf{lpo}} y$

$\mathsf{s}(x) >_{\mathsf{lpo}} x$ $\qquad$ $\mathsf{s}(x) \times y >^?_{\mathsf{lpo}} y$

## Definition (Precedence)

precedence is strict order $>$ on $\mathcal{F}$

## Definition (Lexicographic Path Order)

relation $>_{\mathsf{lpo}}$ on terms defined by $s >_{\mathsf{lpo}} t$ if $s = f(s_1, \ldots, s_n)$ and either

1. $\exists\, i \quad s_i >_{\mathsf{lpo}} t$ or $s_i = t$
2. $t = g(t_1, \ldots, t_m)$ and $f > g$ and $\forall\, j \quad s >_{\mathsf{lpo}} t_j$
3. $t = f(t_1, \ldots, t_n)$ and $\exists\, i$ such that
   - $\forall\, j < i \quad s_j = t_j$
   - $s_i >_{\mathsf{lpo}} t_i$
   - $\forall\, j > i \quad s >_{\mathsf{lpo}} t_j$

### Example

$$\mathsf{s}(x) \times y >^?_{\mathsf{lpo}} (x \times y) + y$$

$\times > +$ $\qquad$ $\mathsf{s}(x) \times y >^?_{\mathsf{lpo}} x \times y$ $\qquad$ $\mathsf{s}(x) \times y >^?_{\mathsf{lpo}} y$

$\mathsf{s}(x) >_{\mathsf{lpo}} x$ $\qquad$ $\mathsf{s}(x) \times y >^?_{\mathsf{lpo}} y$

## Definition (Precedence)

precedence is strict order $>$ on $\mathcal{F}$

## Definition (Lexicographic Path Order)

relation $>_{\mathsf{lpo}}$ on terms defined by $s >_{\mathsf{lpo}} t$ if $s = f(s_1, \ldots, s_n)$ and either

1. $\exists\, i \quad s_i >_{\mathsf{lpo}} t$ or $s_i = t$
2. $t = g(t_1, \ldots, t_m)$ and $f > g$ and $\forall\, j \quad s >_{\mathsf{lpo}} t_j$
3. $t = f(t_1, \ldots, t_n)$ and $\exists\, i$ such that
   - $\forall\, j < i \quad s_j = t_j$
   - $s_i >_{\mathsf{lpo}} t_i$
   - $\forall\, j > i \quad s >_{\mathsf{lpo}} t_j$

### Example

$$\mathsf{s}(x) \times y >^{?}_{\mathsf{lpo}} (x \times y) + y$$

$\times > +$ $\qquad$ $\mathsf{s}(x) \times y >_{\mathsf{lpo}} x \times y$ $\qquad$ $\mathsf{s}(x) \times y >^{?}_{\mathsf{lpo}} y$

$\mathsf{s}(x) >_{\mathsf{lpo}} x$ $\qquad$ $\mathsf{s}(x) \times y >_{\mathsf{lpo}} y$

## Definition (Precedence)

precedence is strict order $>$ on $\mathcal{F}$

## Definition (Lexicographic Path Order)

relation $>_{\mathsf{lpo}}$ on terms defined by $s >_{\mathsf{lpo}} t$ if $s = f(s_1, \ldots, s_n)$ and either

1. $\exists\, i \quad s_i >_{\mathsf{lpo}} t$ or $s_i = t$
2. $t = g(t_1, \ldots, t_m)$ and $f > g$ and $\forall\, j \quad s >_{\mathsf{lpo}} t_j$
3. $t = f(t_1, \ldots, t_n)$ and $\exists\, i$ such that
   - $\forall\, j < i \quad s_j = t_j$
   - $s_i >_{\mathsf{lpo}} t_i$
   - $\forall\, j > i \quad s >_{\mathsf{lpo}} t_j$

### Example

$$\mathsf{s}(x) \times y >_{\mathsf{lpo}} (x \times y) + y$$

$$\times > + \qquad \mathsf{s}(x) \times y >_{\mathsf{lpo}} x \times y \qquad \mathsf{s}(x) \times y >_{\mathsf{lpo}} y$$

$$\mathsf{s}(x) >_{\mathsf{lpo}} x \qquad \mathsf{s}(x) \times y >_{\mathsf{lpo}} y$$

## Definition (Precedence)

precedence is strict order $>$ on $\mathcal{F}$

## Definition (Lexicographic Path Order)

relation $>_{\mathsf{lpo}}$ on terms defined by $s >_{\mathsf{lpo}} t$ if $s = f(s_1, \ldots, s_n)$ and either

1. $\exists\, i \quad s_i >_{\mathsf{lpo}} t$ or $s_i = t$
2. $t = g(t_1, \ldots, t_m)$ and $f > g$ and $\forall\, j \quad s >_{\mathsf{lpo}} t_j$
3. $t = f(t_1, \ldots, t_n)$ and $\exists\, i$ such that
   - $\forall\, j < i \quad s_j = t_j$
   - $s_i >_{\mathsf{lpo}} t_i$
   - $\forall\, j > i \quad s >_{\mathsf{lpo}} t_j$

## Theorem

$>_{\mathsf{lpo}}$ *is reduction order for every well-founded precedence* $>$

## Definition (Precedence)

precedence is strict order $>$ on $\mathcal{F}$

## Definition (Lexicographic Path Order)

relation $>_{\mathsf{lpo}}$ on terms defined by $s >_{\mathsf{lpo}} t$ if $s = f(s_1, \ldots, s_n)$ and either

1. $\exists\, i \quad s_i >_{\mathsf{lpo}} t$ or $s_i = t$
2. $t = g(t_1, \ldots, t_m)$ and $f > g$ and $\forall\, j \quad s >_{\mathsf{lpo}} t_j$
3. $t = f(t_1, \ldots, t_n)$ and $\exists\, i$ such that
   - $\forall\, j < i \quad s_j = t_j$
   - $s_i >_{\mathsf{lpo}} t_i$
   - $\forall\, j > i \quad s >_{\mathsf{lpo}} t_j$

## Theorem

$>_{\mathsf{lpo}}$ *is reduction order for every well-founded precedence* $>$

## Corollary

*if* $>$ *is well-founded and* $\ell >_{\mathsf{lpo}} r$ *for all* $\ell \to r \in \mathcal{R}$ *then* $\mathcal{R}$ *is terminating*

## Lemma

- *if* $> \subseteq \succ$ *then* $>_{\mathsf{lpo}} \subseteq \succ_{\mathsf{lpo}}$ *incrementality*

**Lemma**

- *if* $> \subseteq \succ$ *then* $>_{\mathsf{lpo}} \subseteq \succ_{\mathsf{lpo}}$ *incrementality*
- *if* $>$ *is total on* $\mathcal{F}$ *then* $>_{\mathsf{lpo}}$ *is total on* $\mathcal{T}(\mathcal{F})$ *ground-totality*

## Lemma

- *if* $> \subseteq \succ$ *then* $>_{\mathsf{lpo}} \subseteq \succ_{\mathsf{lpo}}$ *incrementality*
- *if* $>$ *is total on* $\mathcal{F}$ *then* $>_{\mathsf{lpo}}$ *is total on* $\mathcal{T}(\mathcal{F})$ *ground-totality*

## Definition (LPO Decision Problems)

$P_1$ : instance: TRS $\mathcal{R}$ and precedence $>$

question: does $\mathcal{R} \subseteq >_{\mathsf{lpo}}$ hold?

**Lemma**

- *if* $> \subseteq \succ$ *then* $>_{\mathsf{lpo}} \subseteq \succ_{\mathsf{lpo}}$ *incrementality*
- *if* $>$ *is total on* $\mathcal{F}$ *then* $>_{\mathsf{lpo}}$ *is total on* $\mathcal{T}(\mathcal{F})$ *ground-totality*

**Definition (LPO Decision Problems)**

| | | |
|---|---|---|
| $P_1$ : | instance: | TRS $\mathcal{R}$ and precedence $>$ |
| | question: | does $\mathcal{R} \subseteq >_{\mathsf{lpo}}$ hold? |
| $P_2$ : | instance: | TRS $\mathcal{R}$ |
| | question: | $\exists$ precedence $>$ such that $\mathcal{R} \subseteq >_{\mathsf{lpo}}$ holds? |

## Lemma

- *if* $> \subseteq \succ$ *then* $>_{\mathsf{lpo}} \subseteq \succ_{\mathsf{lpo}}$ *incrementality*
- *if* $>$ *is total on* $\mathcal{F}$ *then* $>_{\mathsf{lpo}}$ *is total on* $\mathcal{T}(\mathcal{F})$ *ground-totality*

## Definition (LPO Decision Problems)

| | | |
|---|---|---|
| $P_1$ : | instance: | TRS $\mathcal{R}$ and precedence $>$ |
| | question: | does $\mathcal{R} \subseteq >_{\mathsf{lpo}}$ hold? |
| $P_2$ : | instance: | TRS $\mathcal{R}$ |
| | question: | $\exists$ precedence $>$ such that $\mathcal{R} \subseteq >_{\mathsf{lpo}}$ holds? |

## Lemma

*$P_1$ and $P_2$ are decidable*

## Lemma

- *if* $> \subseteq \succ$ *then* $>_{\mathsf{lpo}} \subseteq \succ_{\mathsf{lpo}}$ *incrementality*
- *if* $>$ *is total on* $\mathcal{F}$ *then* $>_{\mathsf{lpo}}$ *is total on* $\mathcal{T}(\mathcal{F})$ *ground-totality*

## Definition (LPO Decision Problems)

| | | |
|---|---|---|
| $P_1$ : | instance: | TRS $\mathcal{R}$ and precedence $>$ |
| | question: | does $\mathcal{R} \subseteq >_{\mathsf{lpo}}$ hold? |
| $P_2$ : | instance: | TRS $\mathcal{R}$ |
| | question: | $\exists$ precedence $>$ such that $\mathcal{R} \subseteq >_{\mathsf{lpo}}$ holds? |

## Lemma

*$P_1$ and $P_2$ are decidable (in time exponential in $|s| + |t|$)*

## Examples

| TRS | precedence |
| --- | --- |
| $$\begin{aligned} 0 + y &\to y \\ \mathsf{s}(x) + y &\to \mathsf{s}(x + y) \\ 0 \times y &\to 0 \\ \mathsf{s}(x) \times y &\to (x \times y) + y \end{aligned}$$ | |

## Examples

TRS

$$\begin{aligned} 0 + y &\to y \\ \mathsf{s}(x) + y &\to \mathsf{s}(x + y) \\ 0 \times y &\to 0 \\ \mathsf{s}(x) \times y &\to (x \times y) + y \end{aligned}$$

precedence

$$\times > + > \mathsf{s}$$

## Examples

| TRS | precedence |
| --- | --- |
| $\begin{aligned} 0 + y &\to y \\ s(x) + y &\to s(x + y) \\ 0 \times y &\to 0 \\ s(x) \times y &\to (x \times y) + y \end{aligned}$ | $\times > + > s$ |
| $\begin{aligned} ack(0, y) &\to s(y) \\ ack(s(x), 0) &\to ack(x, s(0)) \\ ack(s(x), s(y)) &\to ack(x, ack(s(x), y)) \end{aligned}$ | |

# Examples

| TRS | precedence |
|---|---|
| $0 + y \to y$<br>$\mathsf{s}(x) + y \to \mathsf{s}(x + y)$<br>$0 \times y \to 0$<br>$\mathsf{s}(x) \times y \to (x \times y) + y$ | $\times > + > \mathsf{s}$ |
| $\mathsf{ack}(0, y) \to \mathsf{s}(y)$<br>$\mathsf{ack}(\mathsf{s}(x), 0) \to \mathsf{ack}(x, \mathsf{s}(0))$<br>$\mathsf{ack}(\mathsf{s}(x), \mathsf{s}(y)) \to \mathsf{ack}(x, \mathsf{ack}(\mathsf{s}(x), y))$ | $\mathsf{ack} > \mathsf{s}$ |

# Examples

| TRS | precedence |
|---|---|
| $0 + y \to y$<br>$\mathsf{s}(x) + y \to \mathsf{s}(x + y)$<br>$0 \times y \to 0$<br>$\mathsf{s}(x) \times y \to (x \times y) + y$ | $\times > + > \mathsf{s}$ |
| $\mathsf{ack}(0, y) \to \mathsf{s}(y)$<br>$\mathsf{ack}(\mathsf{s}(x), 0) \to \mathsf{ack}(x, \mathsf{s}(0))$<br>$\mathsf{ack}(\mathsf{s}(x), \mathsf{s}(y)) \to \mathsf{ack}(x, \mathsf{ack}(\mathsf{s}(x), y))$ | $\mathsf{ack} > \mathsf{s}$ |

$$
\begin{aligned}
\mathsf{e} \cdot x &\to x & x \cdot \mathsf{e} &\to x \\
x^- \cdot x &\to \mathsf{e} & x \cdot x^- &\to \mathsf{e} \\
(x \cdot y) \cdot z &\to x \cdot (y \cdot z) & x^{--} &\to x \\
\mathsf{e}^- &\to \mathsf{e} & (x \cdot y)^- &\to y^- \cdot x^- \\
x^- \cdot (x \cdot y) &\to y & x \cdot (x^- \cdot y) &\to y
\end{aligned}
$$

# Examples

| TRS | | precedence |
|---|---|---|
| $\begin{aligned} 0 + y &\to y \\ \mathsf{s}(x) + y &\to \mathsf{s}(x + y) \\ 0 \times y &\to 0 \\ \mathsf{s}(x) \times y &\to (x \times y) + y \end{aligned}$ | | $\times > + > \mathsf{s}$ |
| $\begin{aligned} \mathsf{ack}(0, y) &\to \mathsf{s}(y) \\ \mathsf{ack}(\mathsf{s}(x), 0) &\to \mathsf{ack}(x, \mathsf{s}(0)) \\ \mathsf{ack}(\mathsf{s}(x), \mathsf{s}(y)) &\to \mathsf{ack}(x, \mathsf{ack}(\mathsf{s}(x), y)) \end{aligned}$ | | $\mathsf{ack} > \mathsf{s}$ |
| $\begin{aligned} \mathsf{e} \cdot x &\to x \\ x^- \cdot x &\to \mathsf{e} \\ (x \cdot y) \cdot z &\to x \cdot (y \cdot z) \\ \mathsf{e}^- &\to \mathsf{e} \\ x^- \cdot (x \cdot y) &\to y \end{aligned}$ | $\begin{aligned} x \cdot \mathsf{e} &\to x \\ x \cdot x^- &\to \mathsf{e} \\ x^{--} &\to x \\ (x \cdot y)^- &\to y^- \cdot x^- \\ x \cdot (x^- \cdot y) &\to y \end{aligned}$ | ${}^- > \cdot > \mathsf{e}$ |

# Outline

## Definitions (Weight Function)

- 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 $c \in \mathcal{F}$

## Definitions (Weight Function)

- 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 $t$:

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

## Example

- rewrite rules

$$
\begin{aligned}
\mathrm{e} \cdot x &\to x & x \cdot \mathrm{e} &\to x \\
x^{-} \cdot x &\to \mathrm{e} & x \cdot x^{-} &\to \mathrm{e} \\
(x \cdot y) \cdot z &\to x \cdot (y \cdot z) & x^{--} &\to x \\
\mathrm{e}^{-} &\to \mathrm{e} & (x \cdot y)^{-} &\to y^{-} \cdot x^{-} \\
x^{-} \cdot (x \cdot y) &\to y & x \cdot (x^{-} \cdot y) &\to y
\end{aligned}
$$

## Example

- rewrite rules

$$
\begin{aligned}
e \cdot x &\to x & x \cdot e &\to x \\
x^- \cdot x &\to e & x \cdot x^- &\to e \\
(x \cdot y) \cdot z &\to x \cdot (y \cdot z) & x^{--} &\to x \\
e^- &\to e & (x \cdot y)^- &\to y^- \cdot x^- \\
x^- \cdot (x \cdot y) &\to y & x \cdot (x^- \cdot y) &\to y
\end{aligned}
$$

- weight function: $w(e) = w(\cdot) = w_0 = 1 \qquad w(^-) = 0$

## Example

- rewrite rules

$$
\begin{aligned}
\mathsf{e} \cdot x &\to x & x \cdot \mathsf{e} &\to x \\
x^- \cdot x &\to \mathsf{e} & x \cdot x^- &\to \mathsf{e} \\
(x \cdot y) \cdot z &\to x \cdot (y \cdot z) & x^{--} &\to x \\
\mathsf{e}^- &\to \mathsf{e} & (x \cdot y)^- &\to y^- \cdot x^- \\
x^- \cdot (x \cdot y) &\to y & x \cdot (x^- \cdot y) &\to y
\end{aligned}
$$

- weight function: $w(\mathsf{e}) = w(\cdot) = w_0 = 1 \qquad w(^-) = 0$

  $w(\mathsf{e} \cdot x) = 3$

## Example

- rewrite rules

$$
\begin{aligned}
e \cdot x &\to x & x \cdot e &\to x \\
x^- \cdot x &\to e & x \cdot x^- &\to e \\
(x \cdot y) \cdot z &\to x \cdot (y \cdot z) & x^{--} &\to x \\
e^- &\to e & (x \cdot y)^- &\to y^- \cdot x^- \\
x^- \cdot (x \cdot y) &\to y & x \cdot (x^- \cdot y) &\to y
\end{aligned}
$$

- weight function: $w(e) = w(\cdot) = w_0 = 1 \qquad w(^-) = 0$

$w(e \cdot x) = 3 \qquad w(x) = 1$

## Example

- rewrite rules

$$\begin{aligned} \mathsf{e} \cdot x &\to x & x \cdot \mathsf{e} &\to x \\ x^- \cdot x &\to \mathsf{e} & x \cdot x^- &\to \mathsf{e} \\ (x \cdot y) \cdot z &\to x \cdot (y \cdot z) & x^{--} &\to x \\ \mathsf{e}^- &\to \mathsf{e} & (x \cdot y)^- &\to y^- \cdot x^- \\ x^- \cdot (x \cdot y) &\to y & x \cdot (x^- \cdot y) &\to y \end{aligned}$$

- weight function: $w(\mathsf{e}) = w(\cdot) = w_0 = 1 \qquad w(^-) = 0$

$$w(\mathsf{e} \cdot x) = 3 \qquad w(x) = 1 \qquad w((x \cdot y)^-) = ?$$

## Example

- rewrite rules

$$
\begin{aligned}
\mathsf{e} \cdot x &\to x & x \cdot \mathsf{e} &\to x \\
x^- \cdot x &\to \mathsf{e} & x \cdot x^- &\to \mathsf{e} \\
(x \cdot y) \cdot z &\to x \cdot (y \cdot z) & x^{--} &\to x \\
\mathsf{e}^- &\to \mathsf{e} & (x \cdot y)^- &\to y^- \cdot x^- \\
x^- \cdot (x \cdot y) &\to y & x \cdot (x^- \cdot y) &\to y
\end{aligned}
$$

- weight function: $w(\mathsf{e}) = w(\cdot) = w_0 = 1 \qquad w(^-) = 0$

$$w(\mathsf{e} \cdot x) = 3 \qquad w(x) = 1 \qquad w((x \cdot y)^-) = 3$$

## Definitions (Weight Function)

- 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 $t$:

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

- weight function $(w, w_0)$ is admissible for precedence $>$ if

$$f > g \qquad \forall\, g \in \mathcal{F} \setminus \{\, f \,\}$$

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

## Example

- rewrite rules

$$
\begin{aligned}
\mathrm{e} \cdot x &\to x & \qquad x \cdot \mathrm{e} &\to x \\
x^{-} \cdot x &\to \mathrm{e} & \qquad x \cdot x^{-} &\to \mathrm{e} \\
(x \cdot y) \cdot z &\to x \cdot (y \cdot z) & \qquad x^{--} &\to x \\
\mathrm{e}^{-} &\to \mathrm{e} & \qquad (x \cdot y)^{-} &\to y^{-} \cdot x^{-} \\
x^{-} \cdot (x \cdot y) &\to y & \qquad x \cdot (x^{-} \cdot y) &\to y
\end{aligned}
$$

- weight function: $w(\mathrm{e}) = w(\cdot) = w_0 = 1 \qquad w(^{-}) = 0$

$$w(\mathrm{e} \cdot x) = 3 \qquad w(x) = 1 \qquad w((x \cdot y)^{-}) = 3$$

- precedence: $^{-} > \cdot > \mathrm{e}$

## Example

- rewrite rules

$$\begin{aligned} \mathsf{e} \cdot x &\to x & x \cdot \mathsf{e} &\to x \\ x^{-} \cdot x &\to \mathsf{e} & x \cdot x^{-} &\to \mathsf{e} \\ (x \cdot y) \cdot z &\to x \cdot (y \cdot z) & x^{--} &\to x \\ \mathsf{e}^{-} &\to \mathsf{e} & (x \cdot y)^{-} &\to y^{-} \cdot x^{-} \\ x^{-} \cdot (x \cdot y) &\to y & x \cdot (x^{-} \cdot y) &\to y \end{aligned}$$

- weight function: $w(\mathsf{e}) = w(\cdot) = w_0 = 1$ $\quad w(^{-}) = 0$

$$w(\mathsf{e} \cdot x) = 3 \qquad w(x) = 1 \qquad w((x \cdot y)^{-}) = 3$$

- precedence: $^{-} > \cdot > \mathsf{e}$
- admissible because $^{-}$ is maximal in precedence

## Definitions (Knuth–Bendix Order)

relation $>_{\mathsf{kbo}}$ on terms: $s >_{\mathsf{kbo}} t$ if $|s|_x \geqslant |t|_x \ \forall\, x \in \mathcal{V}$ and

**Definitions (Knuth–Bendix Order)**

relation $>_{\mathsf{kbo}}$ on terms: $s >_{\mathsf{kbo}} t$ if $|s|_x \geqslant |t|_x \;\; \forall\, x \in \mathcal{V}$ and either

- $w(s) > w(t)$

## Definitions (Knuth–Bendix Order)

relation $>_{\mathsf{kbo}}$ on terms: $s >_{\mathsf{kbo}} t$ if $|s|_x \geqslant |t|_x \;\; \forall\, x \in \mathcal{V}$ and either

- $w(s) > w(t)$
- $w(s) = w(t)$ and either
  1. $\exists\, n > 0$ such that $s = f^n(t)$ and $t \in \mathcal{V}$

## Definitions (Knuth–Bendix Order)

relation $>_{\mathsf{kbo}}$ on terms: $s >_{\mathsf{kbo}} t$ if $|s|_x \geqslant |t|_x \;\; \forall\, x \in \mathcal{V}$ and either

- $w(s) > w(t)$
- $w(s) = w(t)$ and either
  1. $\exists\, n > 0$ such that $s = f^n(t)$ and $t \in \mathcal{V}$
  2. $s = f(s_1, \ldots, s_n)$ and $t = f(t_1, \ldots, t_n)$ and $\exists\, i$ such that
     - $\forall\, j < i \quad s_j = t_j$
     - $s_i >_{\mathsf{kbo}} t_i$

## Definitions (Knuth–Bendix Order)

relation $>_{\mathsf{kbo}}$ on terms: $s >_{\mathsf{kbo}} t$ if $|s|_x \geqslant |t|_x \;\; \forall\, x \in \mathcal{V}$ and either

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

## Definitions (Knuth–Bendix Order)

relation $>_{\mathsf{kbo}}$ on terms: $s >_{\mathsf{kbo}} t$ if $|s|_x \geqslant |t|_x \;\; \forall\, x \in \mathcal{V}$ and either

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

**Theorem**

$>_{\mathsf{kbo}}$ *is reduction order if precedence $>$ is well-founded and weight function $(w, w_0)$ is admissible for $>$*

## Example

- rewrite rules

$$
\begin{aligned}
\mathsf{e} \cdot x &\to x & x \cdot \mathsf{e} &\to x \\
x^- \cdot x &\to \mathsf{e} & x \cdot x^- &\to \mathsf{e} \\
(x \cdot y) \cdot z &\to x \cdot (y \cdot z) & x^{--} &\to x \\
\mathsf{e}^- &\to \mathsf{e} & (x \cdot y)^- &\to y^- \cdot x^- \\
x^- \cdot (x \cdot y) &\to y & x \cdot (x^- \cdot y) &\to y
\end{aligned}
$$

- weight function $\quad w(\mathsf{e}) = w(\cdot) = w_0 = 1 \qquad w(^-) = 0$
- precedence $\quad ^- > \cdot > \mathsf{e}$

## Example

- rewrite rules

$$
\begin{aligned}
e \cdot x &\to x & x \cdot e &\to x \\
x^- \cdot x &\to e & x \cdot x^- &\to e \\
(x \cdot y) \cdot z &\to x \cdot (y \cdot z) & x^{--} &\to x \\
e^- &\to e & (x \cdot y)^- &\to y^- \cdot x^- \\
x^- \cdot (x \cdot y) &\to y & x \cdot (x^- \cdot y) &\to y
\end{aligned}
$$

- weight function $\quad w(e) = w(\cdot) = w_0 = 1 \quad w(^-) = 0$
- precedence $\quad {}^- > \cdot > e$

$e \cdot x >_{\mathrm{kbo}} x$

## Example

- rewrite rules

$$
\begin{aligned}
\mathsf{e} \cdot x &\to x & x \cdot \mathsf{e} &\to x \\
x^- \cdot x &\to \mathsf{e} & x \cdot x^- &\to \mathsf{e} \\
(x \cdot y) \cdot z &\to x \cdot (y \cdot z) & x^{--} &\to x \\
\mathsf{e}^- &\to \mathsf{e} & (x \cdot y)^- &\to y^- \cdot x^- \\
x^- \cdot (x \cdot y) &\to y & x \cdot (x^- \cdot y) &\to y
\end{aligned}
$$

- weight function $\quad w(\mathsf{e}) = w(\cdot) = w_0 = 1 \qquad w(^-) = 0$
- precedence $\quad ^- > \cdot > \mathsf{e}$

$$\mathsf{e} \cdot x >_{\mathsf{kbo}} x \qquad x^{--} >_{\mathsf{kbo}} x$$

## Example

- rewrite rules

$$\begin{aligned}
\mathsf{e} \cdot x &\to x & x \cdot \mathsf{e} &\to x \\
x^- \cdot x &\to \mathsf{e} & x \cdot x^- &\to \mathsf{e} \\
(x \cdot y) \cdot z &\to x \cdot (y \cdot z) & x^{--} &\to x \\
\mathsf{e}^- &\to \mathsf{e} & (x \cdot y)^- &\to y^- \cdot x^- \\
x^- \cdot (x \cdot y) &\to y & x \cdot (x^- \cdot y) &\to y
\end{aligned}$$

- weight function $\quad w(\mathsf{e}) = w(\cdot) = w_0 = 1 \qquad w({}^-) = 0$
- precedence $\quad {}^- > \cdot > \mathsf{e}$

$$\mathsf{e} \cdot x >_{\mathsf{kbo}} x \qquad x^{--} >_{\mathsf{kbo}} x \qquad (x \cdot y)^- >_{\mathsf{kbo}} y^- \cdot x^-$$

## Lemma

- *if* $> \subseteq \succ$ *then* $>_{\mathsf{kbo}} \subseteq \succ_{\mathsf{kbo}}$ *incrementality*

### Lemma

- *if* $> \subseteq \succ$ *then* $>_{\mathsf{kbo}} \subseteq \succ_{\mathsf{kbo}}$ *incrementality*
- *if* $>$ *is total on* $\mathcal{F}$ *then* $>_{\mathsf{kbo}}$ *is total on* $\mathcal{T}(\mathcal{F})$ *ground-totality*

## Lemma

- *if* $> \subseteq \succ$ *then* $>_{\mathsf{kbo}} \subseteq \succ_{\mathsf{kbo}}$ *incrementality*
- *if* $>$ *is total on* $\mathcal{F}$ *then* $>_{\mathsf{kbo}}$ *is total on* $\mathcal{T}(\mathcal{F})$ *ground-totality*

## Definition (KBO Decision Problems)

$P_1$: instance: TRS $\mathcal{R}$, weight function $(w, w_0)$, precedence $>$

question: does $\mathcal{R} \subseteq >_{\mathsf{kbo}}$ hold?

## Lemma

- *if* $> \subseteq \succ$ *then* $>_{\mathsf{kbo}} \subseteq \succ_{\mathsf{kbo}}$ *incrementality*
- *if* $>$ *is total on* $\mathcal{F}$ *then* $>_{\mathsf{kbo}}$ *is total on* $\mathcal{T}(\mathcal{F})$ *ground-totality*

## Definition (KBO Decision Problems)

| | | |
|---|---|---|
| $P_1$: | instance: | TRS $\mathcal{R}$, weight function $(w, w_0)$, precedence $>$ |
| | question: | does $\mathcal{R} \subseteq >_{\mathsf{kbo}}$ hold? |
| $P_2$: | instance: | TRS $\mathcal{R}$ |
| | question: | $\exists$ precedence $>$, weight function $(w, w_0)$ s.t. $\mathcal{R} \subseteq >_{\mathsf{kbo}}$? |

### Lemma

- *if* $> \subseteq \succ$ *then* $>_{\mathsf{kbo}} \subseteq \succ_{\mathsf{kbo}}$ *incrementality*
- *if* $>$ *is total on* $\mathcal{F}$ *then* $>_{\mathsf{kbo}}$ *is total on* $\mathcal{T}(\mathcal{F})$ *ground-totality*

### Definition (KBO Decision Problems)

| | | |
|---|---|---|
| $P_1$ : | instance: | TRS $\mathcal{R}$, weight function $(w, w_0)$, precedence $>$ |
| | question: | does $\mathcal{R} \subseteq >_{\mathsf{kbo}}$ hold? |
| $P_2$ : | instance: | TRS $\mathcal{R}$ |
| | question: | $\exists$ precedence $>$, weight function $(w, w_0)$ s.t. $\mathcal{R} \subseteq >_{\mathsf{kbo}}$? |

### Lemma

*$P_1$ and $P_2$ are decidable*

**Lemma**

- *if* $> \subseteq \succ$ *then* $>_{\mathsf{kbo}} \subseteq \succ_{\mathsf{kbo}}$ *incrementality*
- *if* $>$ *is total on* $\mathcal{F}$ *then* $>_{\mathsf{kbo}}$ *is total on* $\mathcal{T}(\mathcal{F})$ *ground-totality*

**Definition (KBO Decision Problems)**

$P_1$: instance: TRS $\mathcal{R}$, weight function $(w, w_0)$, precedence $>$

question: does $\mathcal{R} \subseteq >_{\mathsf{kbo}}$ hold?

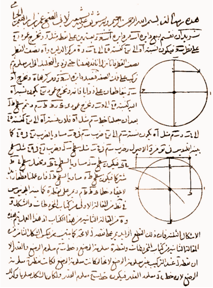
$P_2$: instance: TRS $\mathcal{R}$

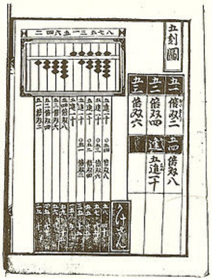
question: $\exists$ precedence $>$, weight function $(w, w_0)$ s.t. $\mathcal{R} \subseteq >_{\mathsf{kbo}}$?

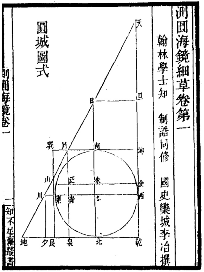
**Lemma**

*$P_1$ and $P_2$ are decidable (in time polynomial in $|s| + |t|$)*

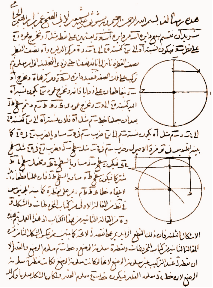
# Formal Analysis Technologies

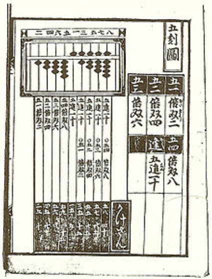


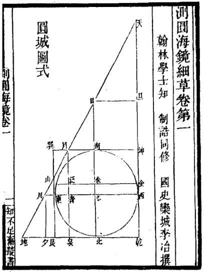





# Formal Analysis Technologies

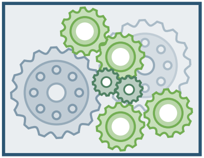







# $\mathsf{T_TT_2}$: Tyrolean Termination Tool 2

input: TRS $\mathcal{R}$

$\mathcal{R} \longrightarrow$ 

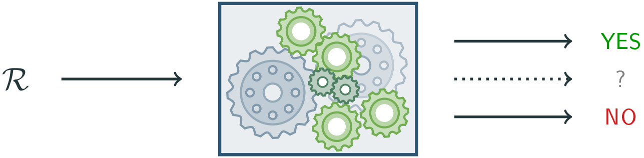
---

Korp et al., **Tyrolean Termination Tool 2**. RTA 2009.

## $\mathsf{T_TT_2}$: Tyrolean Termination Tool 2

input: TRS $\mathcal{R}$

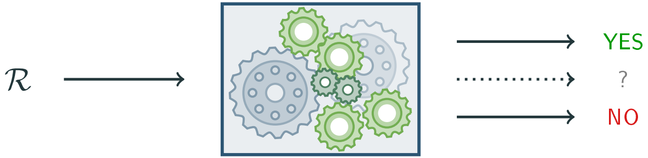
output: YES + termination proof, or NO + counterexample



# $\mathsf{T_TT_2}$: Tyrolean Termination Tool 2

input: TRS $\mathcal{R}$

output: YES + termination proof, or NO + counterexample



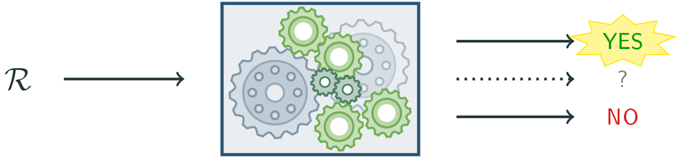

## Example (Addition)

$$0 + x \to x \qquad\qquad \mathsf{s}(x) + y \to \mathsf{s}(x + y)$$

# $\mathsf{T_TT_2}$: Tyrolean Termination Tool 2

input: TRS $\mathcal{R}$

output: YES + termination proof, or NO + counterexample



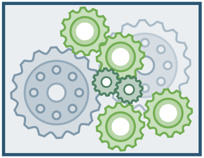

**Example (Addition)**

$$0 + x \to x \qquad\qquad \mathsf{s}(x) + y \to \mathsf{s}(x + y)$$

## TTT2: Tyrolean Termination Tool 2

input: TRS $\mathcal{R}$

output: YES + termination proof, or NO + counterexample

$\mathcal{R} \longrightarrow$  $\longrightarrow$ YES / $\cdots\rightarrow$ ? / $\longrightarrow$ NO

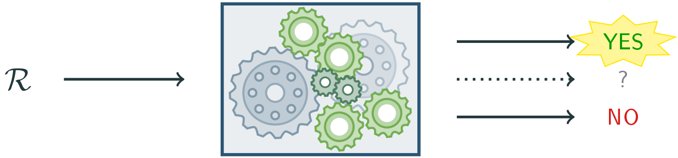
### Example (Bean Game)

$\bullet\ \bullet \to \circ \qquad \circ\ \circ \to \circ \qquad \bullet\ \circ \to \bullet \qquad \circ\ \bullet \to \bullet$

## $\mathsf{T_TT_2}$: Tyrolean Termination Tool 2

input: TRS $\mathcal{R}$

output: YES + termination proof, or NO + counterexample



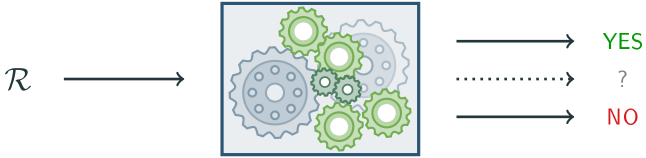

### Example (Bean Game)

$\bullet\ \bullet \to \circ \qquad \circ\ \circ \to \circ \qquad \bullet\ \circ \to \bullet \qquad \circ\ \bullet \to \bullet$

## $\mathsf{T_TT_2}$: Tyrolean Termination Tool 2

input: TRS $\mathcal{R}$

output: YES + termination proof, or NO + counterexample



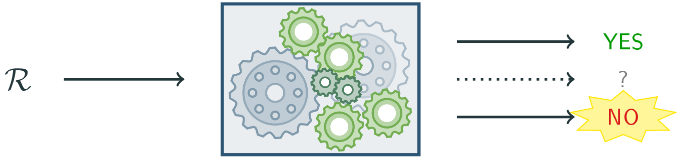

### Example (Sieve of Eratosthenes)

$$
\begin{aligned}
\mathsf{primes} &\to \mathsf{sieve}(\mathsf{from}(\mathsf{s}(\mathsf{s}(0)))) & \mathsf{sieve}(0 : y) &\to \mathsf{sieve}(y) \\
\mathsf{from}(x) &\to x : \mathsf{from}(\mathsf{s}(x)) & \mathsf{sieve}(\mathsf{s}(x) : y) &\to \mathsf{s}(x) : \mathsf{sieve}(\mathsf{filter}(x, y, x)) \\
\mathsf{hd}(x : y) &\to x & \mathsf{filter}(0, y : z, w) &\to 0 : \mathsf{filter}(w, z, w) \\
\mathsf{tl}(x : y) &\to y & \mathsf{filter}(\mathsf{s}(x), y : z, w) &\to y : \mathsf{filter}(x, z, w)
\end{aligned}
$$

## $\mathsf{T_TT_2}$: Tyrolean Termination Tool 2

input: TRS $\mathcal{R}$

output: YES + termination proof, or NO + counterexample




### Example (Sieve of Eratosthenes)

$$\mathsf{primes} \to \mathsf{sieve}(\mathsf{from}(\mathsf{s}(\mathsf{s}(0)))) \qquad \mathsf{sieve}(0 : y) \to \mathsf{sieve}(y)$$

$$\boxed{\mathsf{from}(x) \to x : \mathsf{from}(\mathsf{s}(x))} \qquad \mathsf{sieve}(\mathsf{s}(x) : y) \to \mathsf{s}(x) : \mathsf{sieve}(\mathsf{filter}(x, y, x))$$

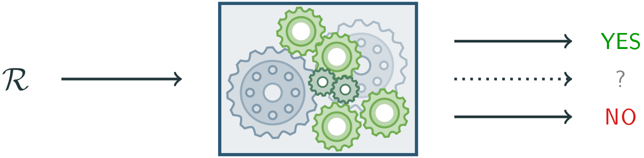
$$\mathsf{hd}(x : y) \to x \qquad \mathsf{filter}(0, y : z, w) \to 0 : \mathsf{filter}(w, z, w)$$

$$\mathsf{tl}(x : y) \to y \qquad \mathsf{filter}(\mathsf{s}(x), y : z, w) \to y : \mathsf{filter}(x, z, w)$$

## TTT2: Tyrolean Termination Tool 2

input: TRS $\mathcal{R}$

output: YES + termination proof, or NO + counterexample



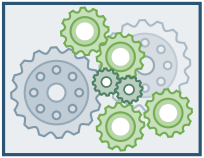

### Implemented techniques

- dependency pair (DP) framework, dependency graphs
- interpretation methods: polynomials, matrices, arctic, ordinals
- reduction orders: LPO, KBO, weighted path order
- labeling techniques: semantic labelling, matchbounds
- non-termination: loops and unfoldings, ...

## T<sub></sub>TT2: Tyrolean Termination Tool 2

input: TRS $\mathcal{R}$

output: YES + termination proof, or NO + counterexample

$\mathcal{R}$ ⟶ 

⟶ YES

⇢ ?

⟶ NO

## Implemented techniques

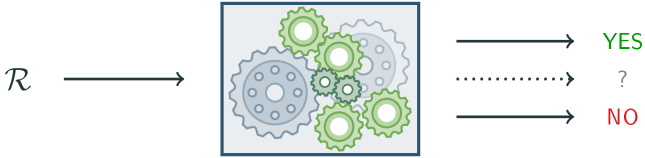
▸ TTT2

- dependency pair (DP) framework, dependency graphs
- interpretation methods: polynomials, matrices, arctic, ordinals
- reduction orders: LPO, KBO, weighted path order
- labeling techniques: semantic labelling, matchbounds
- non-termination: loops and unfoldings, ...

# $\mathsf{T_TT_2}$: Tyrolean Termination Tool 2

input: TRS $\mathcal{R}$

output: YES + termination proof, or NO + counterexample




## Implemented techniques

▸ $\mathsf{T_TT_2}$

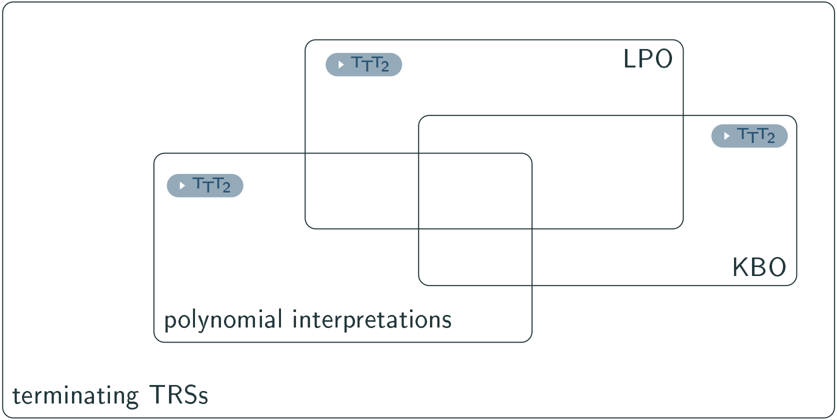
- dependency pair (DP) framework, dependency graphs
- interpretation methods: polynomials, matrices, arctic, ordinals
- reduction orders: LPO, KBO, weighted path order
- labeling techniques: semantic labelling, matchbounds
- non-termination: loops and unfoldings, ...

## Annual termination competition

`http://termination-portal.org`

**Remark**

*KBO, LPO and polynomial interpretations are incomparable*



## Fun/Research Example: Battle of Hercules and Hydra (1)

Hydra is a tree-shaped monster which grows whenever Hercules chops off a head:

If the cut-off head has a grandparent in the tree then the branch from this grandparent multiplies.

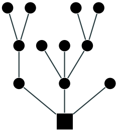
Hydra gets more and more angry: the number of copies corresponds to the number of heads already cut off.

## Fun/Research Example: Battle of Hercules and Hydra (1)

Hydra is a tree-shaped monster which grows whenever Hercules chops off a head:

If the cut-off head has a grandparent in the tree then the branch from this grandparent multiplies.

Hydra gets more and more angry: the number of copies corresponds to the number of heads already cut off.



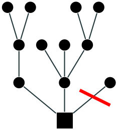
$H_1$

## Fun/Research Example: Battle of Hercules and Hydra (1)

Hydra is a tree-shaped monster which grows whenever Hercules chops off a head:

If the cut-off head has a grandparent in the tree then the branch from this grandparent multiplies.

Hydra gets more and more angry: the number of copies corresponds to the number of heads already cut off.



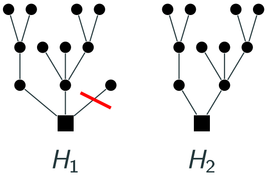
$H_1$

## Fun/Research Example: Battle of Hercules and Hydra (1)

Hydra is a tree-shaped monster which grows whenever Hercules chops off a head:

If the cut-off head has a grandparent in the tree then the branch from this grandparent multiplies.

Hydra gets more and more angry: the number of copies corresponds to the number of heads already cut off.

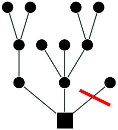



## Fun/Research Example: Battle of Hercules and Hydra (1)

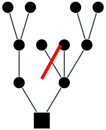
Hydra is a tree-shaped monster which grows whenever Hercules chops off a head:

If the cut-off head has a grandparent in the tree then the branch from this grandparent multiplies.

Hydra gets more and more angry: the number of copies corresponds to the number of heads already cut off.



$H_1$



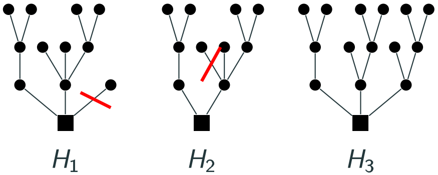
$H_2$

## Fun/Research Example: Battle of Hercules and Hydra (1)

Hydra is a tree-shaped monster which grows whenever Hercules chops off a head:

If the cut-off head has a grandparent in the tree then the branch from this grandparent multiplies.

Hydra gets more and more angry: the number of copies corresponds to the number of heads already cut off.

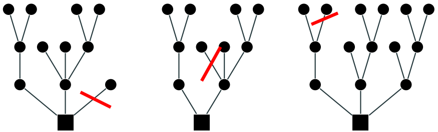



## Fun/Research Example: Battle of Hercules and Hydra (1)

Hydra is a tree-shaped monster which grows whenever Hercules chops off a head:

If the cut-off head has a grandparent in the tree then the branch from this grandparent multiplies.

Hydra gets more and more angry: the number of copies corresponds to the number of heads already cut off.



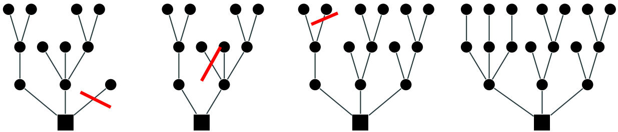
$H_1$ $H_2$ $H_3$

## Fun/Research Example: Battle of Hercules and Hydra (1)

Hydra is a tree-shaped monster which grows whenever Hercules chops off a head:

If the cut-off head has a grandparent in the tree then the branch from this grandparent multiplies.

Hydra gets more and more angry: the number of copies corresponds to the number of heads already cut off.



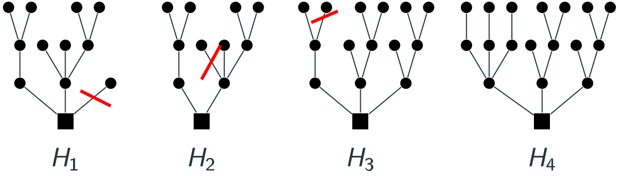
$H_1$ $H_2$ $H_3$ $H_4$

## Fun/Research Example: Battle of Hercules and Hydra (1)

Hydra is a tree-shaped monster which grows whenever Hercules chops off a head:

If the cut-off head has a grandparent in the tree then the branch from this grandparent multiplies.

Hydra gets more and more angry: the number of copies corresponds to the number of heads already cut off.




Will Hydra ever die, such that Hercules wins?

## Fun/Research Example: Battle of Hercules and Hydra (2)

process is modelled by TRS $\mathcal{R}$:

$$
\begin{aligned}
\circ(x) &\to \bullet(\Box(x)) & \bullet(\Box(x)) &\to \Box(\bullet(\bullet(x))) & \bullet(x) &\to x \\
\Box(\circ(x)) &\to \circ(\Box(x)) & \bullet(\mathsf{c}_1(x,y)) &\to \mathsf{c}_1(x,\mathsf{H}(x,y)) \\
\mathsf{H}(0,x) &\to \circ(x) & \bullet(\mathsf{H}(\mathsf{H}(0,y),z)) &\to \mathsf{c}_1(y,z) \\
\mathsf{c}_2(x,y,z) &\to \circ(\mathsf{H}(y,z)) & \bullet(\mathsf{H}(\mathsf{H}(\mathsf{H}(0,x),y),z)) &\to \mathsf{c}_2(x,y,z) \\
\mathsf{c}_1(y,z) &\to \circ(z) & \bullet(\mathsf{c}_2(x,y,z)) &\to \mathsf{c}_2(x,\mathsf{H}(x,y),z)
\end{aligned}
$$

## Fun/Research Example: Battle of Hercules and Hydra (2)

process is modelled by TRS $\mathcal{R}$:

$$
\begin{aligned}
\circ(x) &\to \bullet(\Box(x)) & \bullet(\Box(x)) &\to \Box(\bullet(\bullet(x))) & \bullet(x) &\to x \\
\Box(\circ(x)) &\to \circ(\Box(x)) & \bullet(\mathsf{c}_1(x,y)) &\to \mathsf{c}_1(x,\mathsf{H}(x,y)) \\
\mathsf{H}(0,x) &\to \circ(x) & \bullet(\mathsf{H}(\mathsf{H}(0,y),z)) &\to \mathsf{c}_1(y,z) \\
\mathsf{c}_2(x,y,z) &\to \circ(\mathsf{H}(y,z)) & \bullet(\mathsf{H}(\mathsf{H}(\mathsf{H}(0,x),y),z)) &\to \mathsf{c}_2(x,y,z) \\
\mathsf{c}_1(y,z) &\to \circ(z) & \bullet(\mathsf{c}_2(x,y,z)) &\to \mathsf{c}_2(x,\mathsf{H}(x,y),z)
\end{aligned}
$$

- the TRS is terminating (so Hydra will die)
- but the derivational complexity is beyond multiple recursive

---

Touzet, **Encoding the Hydra Battle as a Rewrite System**. MFCS 1998.

## Fun/Research Example: Battle of Hercules and Hydra (2)

process is modelled by TRS $\mathcal{R}$:

$$
\begin{aligned}
\circ(x) &\to \bullet(\Box(x)) & \bullet(\Box(x)) &\to \Box(\bullet(\bullet(x))) & \bullet(x) &\to x \\
\Box(\circ(x)) &\to \circ(\Box(x)) & \bullet(\mathsf{c}_1(x,y)) &\to \mathsf{c}_1(x,\mathsf{H}(x,y)) & & \\
\mathsf{H}(0,x) &\to \circ(x) & \bullet(\mathsf{H}(\mathsf{H}(0,y),z)) &\to \mathsf{c}_1(y,z) & & \\
\mathsf{c}_2(x,y,z) &\to \circ(\mathsf{H}(y,z)) & \bullet(\mathsf{H}(\mathsf{H}(\mathsf{H}(0,x),y),z)) &\to \mathsf{c}_2(x,y,z) & & \\
\mathsf{c}_1(y,z) &\to \circ(z) & \bullet(\mathsf{c}_2(x,y,z)) &\to \mathsf{c}_2(x,\mathsf{H}(x,y),z) & &
\end{aligned}
$$

- the TRS is terminating (so Hydra will die)
- but the derivational complexity is beyond multiple recursive

### Long-Standing Open Problem

show termination of $\mathcal{R}$ automatically

# Fun/Research Example: Battle of Hercules and Hydra (2)

process is modelled by TRS $\mathcal{R}$:

$$
\begin{array}{rlrlrl}
\circ(x) & \to \bullet(\square(x)) & \bullet(\square(x)) & \to \square(\bullet(\bullet(x))) & \bullet(x) & \to x \\
\square(\circ(x)) & \to \circ(\square(x)) & \bullet(\mathsf{c}_1(x,y)) & \to \mathsf{c}_1(x,\mathsf{H}(x,y)) & & \\
\mathsf{H}(0,x) & \to \circ(x) & \bullet(\mathsf{H}(\mathsf{H}(0,y),z)) & \to \mathsf{c}_1(y,z) & & \\
\mathsf{c}_2(x,y,z) & \to \circ(\mathsf{H}(y,z)) & \bullet(\mathsf{H}(\mathsf{H}(\mathsf{H}(0,x),y),z)) & \to \mathsf{c}_2(x,y,z) & & \\
\mathsf{c}_1(y,z) & \to \circ(z) & \bullet(\mathsf{c}_2(x,y,z)) & \to \mathsf{c}_2(x,\mathsf{H}(x,y),z) & &
\end{array}
$$

- the TRS is terminating (so Hydra will die)
- but the derivational complexity is beyond multiple recursive

## Long-Standing Open Problem

show termination of $\mathcal{R}$ automatically

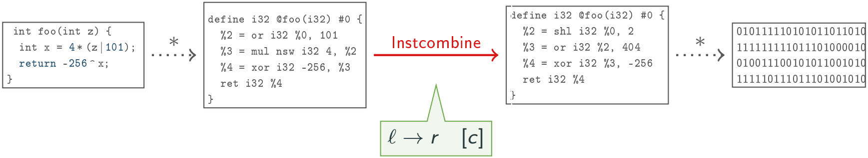
## Solution: Ordinal Interpretations

$\mathsf{H}(x,y) = \omega^x \oplus y \quad \mathsf{c}_2(x,y,z) = \omega^{y+\omega^{x+1}} \oplus z \quad \ldots$

TTT$_2$

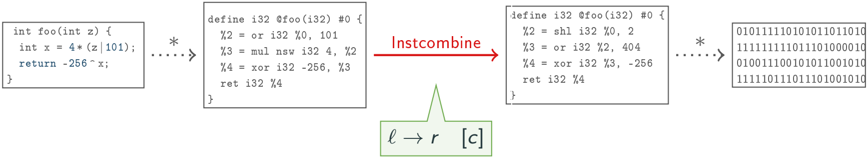
Zankl, W, and Middeldorp, **Beyond Polynomials and Peano Arithmetic.** JSC 69, 2015.

# Practice Example: LLVM Expression Simplification




- LLVM Instcombine: >1000 algebraic simplifications of expressions
- applies "rewrite rules" with side constraints expressible in SMT

# Practice Example: LLVM Expression Simplification

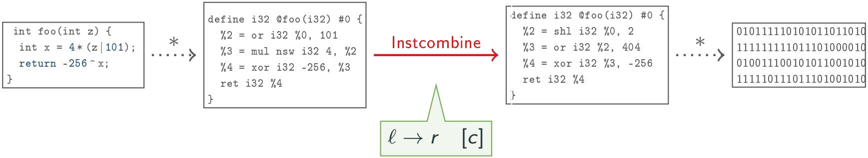



- LLVM Instcombine: >1000 algebraic simplifications of expressions
- applies "rewrite rules" with side constraints expressible in SMT

## Example (Loop detection)

simplification seeking opportunity to replace mul by shift:

# Practice Example: LLVM Expression Simplification



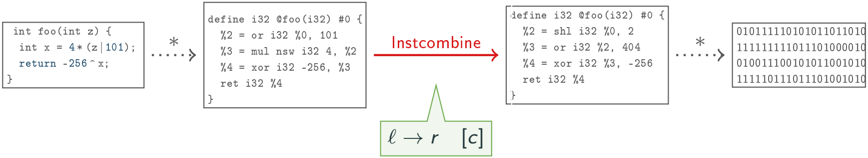

- LLVM Instcombine: >1000 algebraic simplifications of expressions
- applies "rewrite rules" with side constraints expressible in SMT

## Example (Loop detection)

simplification seeking opportunity to replace mul by shift:

$$\mathsf{mul}(\mathsf{sub}(y, x), z) \to \mathsf{mul}(\mathsf{sub}(x, y), \mathsf{abs}(z)) \qquad [z < 0_8 \wedge \mathsf{isPowerOf2}(\mathsf{abs}(z))]$$

# Practice Example: LLVM Expression Simplification



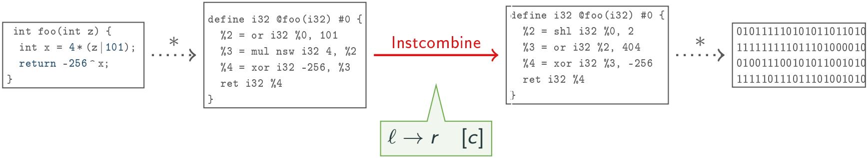

- LLVM Instcombine: >1000 algebraic simplifications of expressions
- applies "rewrite rules" with side constraints expressible in SMT

## Example (Loop detection)

simplification seeking opportunity to replace mul by shift:

$$\mathrm{mul}(\mathrm{sub}(y, x), z) \to \mathrm{mul}(\mathrm{sub}(x, y), \mathrm{abs}(z)) \quad [z < 0_8 \wedge \mathrm{isPowerOf2}(\mathrm{abs}(z))]$$

# Practice Example: LLVM Expression Simplification




- LLVM Instcombine: >1000 algebraic simplifications of expressions
- applies "rewrite rules" with side constraints expressible in SMT

## Example (Loop detection)

simplification seeking opportunity to replace mul by shift:

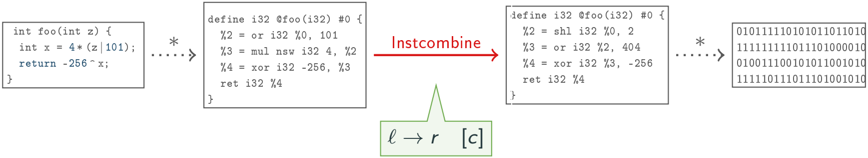
$$\mathsf{mul}(\mathsf{sub}(y, x), z) \to \mathsf{mul}(\mathsf{sub}(x, y), \mathsf{abs}(z)) \qquad [z < 0_8 \wedge \mathsf{isPowerOf2}(\mathsf{abs}(z))]$$

termination tool detects non-termination:

$$\mathsf{mul}(\mathsf{sub}(1_8, 1_8), (-128)_8) \to_{\mathcal{R}} \mathsf{mul}(\mathsf{sub}(1_8, 1_8), \mathsf{abs}((-128)_8))$$

Nishida and W, **Loop Detection by Logically Constrained Term Rewriting.** VSTTE 2018.

# Practice Example: LLVM Expression Simplification




- LLVM Instcombine: >1000 algebraic simplifications of expressions
- applies "rewrite rules" with side constraints expressible in SMT

## Example (Loop detection)

simplification seeking opportunity to replace mul by shift:

$$\mathsf{mul}(\mathsf{sub}(y,x),z) \to \mathsf{mul}(\mathsf{sub}(x,y),\mathsf{abs}(z)) \quad [z < 0_8 \wedge \mathsf{isPowerOf2}(\mathsf{abs}(z))]$$

termination tool detects non-termination:

$$\begin{aligned}\mathsf{mul}(\mathsf{sub}(1_8,1_8),(-128)_8) &\to_{\mathcal{R}} \mathsf{mul}(\mathsf{sub}(1_8,1_8),\mathsf{abs}((-128)_8)) \\ &\to_{\mathcal{R}} \mathsf{mul}(\mathsf{sub}(1_8,1_8),(-128)_8)\end{aligned}$$

---

Nishida and W, **Loop Detection by Logically Constrained Term Rewriting.** VSTTE 2018.

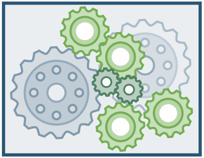
# Outline

# Complexity

## TcT

input: term rewrite system $\mathcal{R}$

output: worst-case derivational complexity $\mathsf{dc}_{\mathcal{R}}$

$\mathcal{R} \longrightarrow$  $\longrightarrow \mathcal{O}(n^k)$

$\dashrightarrow$ ?

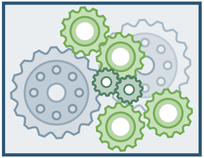
Avanzini, Moser, and Schaper, **TcT: Tyrolean Complexity Tool**. 2016.

# Complexity

## TcT

input: term rewrite system $\mathcal{R}$

output: worst-case derivational complexity $\mathsf{dc}_{\mathcal{R}}$

$\mathcal{R} \longrightarrow$  $\longrightarrow \mathcal{O}(n^k)$

$\dashrightarrow$ ?

### Definitions

- derivation height $\qquad \mathsf{dh}_{\mathcal{R}}(t) = \max \{ n \mid \exists\, u \colon t \to_{\mathcal{R}}^{n} u \}$

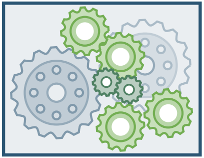
# Complexity

### TcT

input: term rewrite system $\mathcal{R}$

output: worst-case derivational complexity $\mathrm{dc}_{\mathcal{R}}$

$\mathcal{R} \longrightarrow$



$\longrightarrow \mathcal{O}(n^k)$

$\cdots\cdots\rightarrow$ ?

**Definitions**

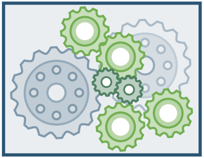
- derivation height $\qquad \mathrm{dh}_{\mathcal{R}}(t) = \max \{\, n \mid \exists\, u \colon t \rightarrow_{\mathcal{R}}^{n} u \,\}$
- derivational complexity $\qquad \mathrm{dc}_{\mathcal{R}}(n) = \max \{\, \mathrm{dh}_{\mathcal{R}}(t) \mid |t| = n \,\}$

### TCT

input: term rewrite system $\mathcal{R}$

output: worst-case derivational complexity $\mathsf{dc}_{\mathcal{R}}$

$\mathcal{R} \longrightarrow$



$\longrightarrow \mathcal{O}(n^k)$

$\dashrightarrow$ ?

**Example (Bean Game)**

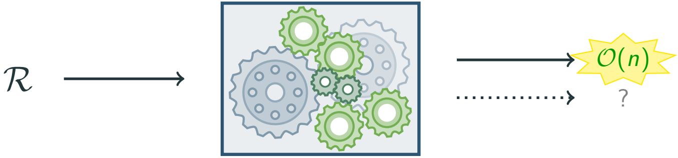
●● → ○  ○○ → ○  ●○ → ●  ○● → ●

# Complexity

## T<sub>C</sub>T

input: term rewrite system $\mathcal{R}$

output: worst-case derivational complexity $\mathrm{dc}_{\mathcal{R}}$




**Example (Bean Game)**

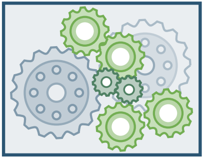
$\bullet\bullet \to \circ \qquad \circ\circ \to \circ \qquad \bullet\circ \to \bullet \qquad \circ\bullet \to \bullet$

# Complexity

## TcT

input: term rewrite system $\mathcal{R}$

output: worst-case derivational complexity $\mathrm{dc}_{\mathcal{R}}$

$\mathcal{R} \longrightarrow$  $\longrightarrow \mathcal{O}(n^k)$

$\dashrightarrow$ ?

**Example**

$$[] \mathbin{@} xs \to xs \qquad (x : xs) \mathbin{@} ys \to x : (xs \mathbin{@} ys)$$

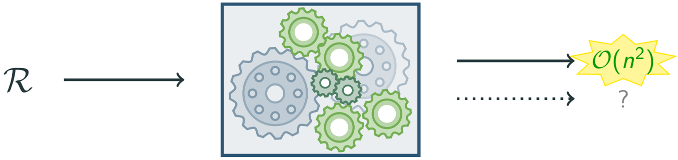
$$\mathsf{flatten}([]) \to [] \qquad \mathsf{flatten}(x : xs) \to x \mathbin{@} \mathsf{flatten}(xs)$$

# Complexity

## TcT

input: term rewrite system $\mathcal{R}$

output: worst-case derivational complexity $\mathsf{dc}_{\mathcal{R}}$




**Example**

$$[] \mathbin{@} xs \to xs \qquad (x : xs) \mathbin{@} ys \to x : (xs \mathbin{@} ys)$$

$$\mathsf{flatten}([]) \to [] \qquad \mathsf{flatten}(x : xs) \to x \mathbin{@} \mathsf{flatten}(xs)$$

**Theorem**

*if TRS $\mathcal{R}$ is polynomially terminating then its derivational complexity is at most doubly exponential*

**Theorem**

*if TRS $\mathcal{R}$ is LPO terminating then its derivational complexity is a multiply recursive function*

**Theorem**

*if TRS $\mathcal{R}$ is KBO terminating then its derivational complexity is bounded by an Ackermann function*

## Research Example: Program Analysis

$$\mathsf{init}(x,y,z) \to \mathsf{sort}(x,y,z)$$
$$\mathsf{sort}(x,y,z) \to \langle \mathsf{ms}_0(x,u,v), \mathsf{ms}_1(x,u,v), \mathsf{ms}_2(x,u,v), \mathsf{ms}_3(x,u,v)\rangle$$
$$[x \geq 2 \wedge u \geq 0 \wedge v \geq 0 \wedge x+1 \geq 2u \wedge 2u \geq x \wedge x \geq 2v \wedge 2v+1 \geq x]$$
$$\mathsf{ms}_0(x,y,z) \to \mathsf{split}(x,y,z) \qquad \mathsf{split}(x,y,z) \to \mathsf{split}(x-2,y,z)\ [x \geq 2]$$
$$\mathsf{ms}_1(x,y,z) \to \mathsf{ms}(y,y,z) \qquad \mathsf{merge}(x,y,z) \to \mathsf{merge}(x-1,y,z)\ [x \geq 1 \wedge y \geq 1]$$
$$\mathsf{ms}_2(x,y,z) \to \mathsf{ms}(z,y,z) \qquad \mathsf{merge}(x,y,z) \to \mathsf{merge}(x,y-1,z)\ [x \geq 1 \wedge y \geq 1]$$
$$\mathsf{ms}_3(x,y,z) \to \mathsf{merge}(y,z,z)$$

- TRSs with SMT arithmetic constraints can represent imperative program over integers

W and Moser, **Runtime Complexity Analysis of Logically Constrained Rewriting**. LOPSTR 2020.

## Research Example: Program Analysis

$$\begin{aligned}
&\mathsf{init}(x, y, z) \to \mathsf{sort}(x, y, z)\\
&\mathsf{sort}(x, y, z) \to \langle \mathsf{ms}_0(x, u, v), \mathsf{ms}_1(x, u, v), \mathsf{ms}_2(x, u, v), \mathsf{ms}_3(x, u, v)\rangle\\
&\qquad\qquad [x \geq 2 \wedge u \geq 0 \wedge v \geq 0 \wedge x + 1 \geq 2u \wedge 2u \geq x \wedge x \geq 2v \wedge 2v + 1 \geq x]
\end{aligned}$$

$$\begin{aligned}
&\mathsf{ms}_0(x, y, z) \to \mathsf{split}(x, y, z) && \mathsf{split}(x, y, z) \to \mathsf{split}(x - 2, y, z)\ [x \geq 2]\\
&\mathsf{ms}_1(x, y, z) \to \mathsf{ms}(y, y, z) && \mathsf{merge}(x, y, z) \to \mathsf{merge}(x - 1, y, z)\ [x \geq 1 \wedge y \geq 1]\\
&\mathsf{ms}_2(x, y, z) \to \mathsf{ms}(z, y, z) && \mathsf{merge}(x, y, z) \to \mathsf{merge}(x, y - 1, z)\ [x \geq 1 \wedge y \geq 1]\\
&\mathsf{ms}_3(x, y, z) \to \mathsf{merge}(y, z, z)
\end{aligned}$$

- TRSs with SMT arithmetic constraints can represent imperative program over integers
- mergesort: new $\mathsf{T_CT}$ derives optimal $\mathcal{O}(n \log(n))$ bound (CoFloCo, KoAT, and previous version of $\mathsf{T_CT}$ at best quadratic)

W and Moser, **Runtime Complexity Analysis of Logically Constrained Rewriting**. LOPSTR 2020.

## Exercises

1. Which of the following TRSs on are polynomially terminating?

$$\mathcal{R}_1: \quad \begin{aligned} \mathsf{f}(x, \mathsf{g}(y, z)) &\to \mathsf{g}(\mathsf{f}(x, y), \mathsf{f}(x, z)) \\ \mathsf{f}(\mathsf{g}(x, y), z) &\to \mathsf{g}(\mathsf{f}(x, z), \mathsf{f}(y, z)) \\ \mathsf{g}(\mathsf{g}(x, y), z) &\to \mathsf{g}(x, \mathsf{g}(y, z)) \end{aligned}$$

$$\mathcal{R}_2: \quad \begin{aligned} \mathsf{f}(\mathsf{g}(x), y) &\to \mathsf{g}(\mathsf{f}(x, \mathsf{f}(x, y))) \\ \mathsf{f}(x, x) &\to \mathsf{g}(\mathsf{g}(x)) \end{aligned}$$

2. Which of the TRSs of Exercise 1 is LPO terminating?

3. Which of the TRSs on slide 13 can be shown terminating by KBO?

4. Show that if a TRS $\mathcal{R}$ is compatible with a KBO with weight function $(w, w_0)$ and precedence $>$ but $(w, w_0)$ is not admissible for $>$ then $\mathcal{R}$ may not be terminating.