Small Polynomial Path Orders in TCT

Naohi Eguchi Martin Avanzini Georg Moser Michael Schaper

Institute of Computer Science
University of Innsbruck, Austria

August 29, 2013

Tyrolean Complexity Tool TCT

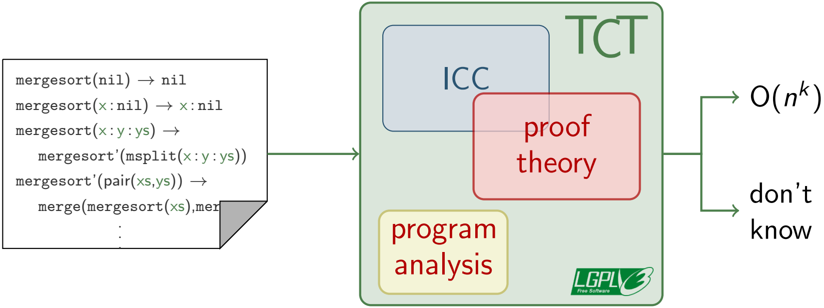
- (runtime) complexity analyser for term rewrite systems (TRSs)

  http://cl-informatik.uibk.ac.at/software/tct

Tyrolean Complexity Tool TCT

- (runtime) complexity analyser for term rewrite systems (TRSs)

  http://cl-informatik.uibk.ac.at/software/tct



Content

1. small polynomial path orders
2. extension to complexity problems of TCT
3. conclusion & experiments

Bellantoni & Cooks Definition of FP

Predicative Recursion on Notation

$$f(\varepsilon, \vec{x}; \vec{y}) = g(\vec{x}; \vec{y})$$
$$f(zi, \vec{x}; \vec{y}) = h_i(z, \vec{x}; \vec{y}, f(z, \vec{x}; \vec{y})) \quad \text{for } i = 0, 1$$

- uses separation of arguments

$$f(\underbrace{n_1, \ldots, n_l}_{\textit{normal}}; \underbrace{n_{l+1}, \ldots, n_{l+k}}_{\textit{safe}})$$

Stephen Bellantoni and Stephen A. Cook
A New Recursion-Theoretic Characterization of the Polytime Functions.
Computational Complexity, Vol. 2, pages 97–110, 1992

Bellantoni & Cooks Definition of FP

Predicative Recursion on Notation

$$f(\varepsilon, \vec{x}; \vec{y}) = g(\vec{x}; \vec{y})$$
$$f(zi, \vec{x}; \vec{y}) = h_i(z, \vec{x}; \vec{y}, f(z, \vec{x}; \vec{y})) \quad \text{for } i = 0, 1$$

- uses separation of arguments

$$f(\underbrace{n_1, \ldots, n_l}_{normal}; \underbrace{n_{l+1}, \ldots, n_{l+k}}_{safe})$$

Weak Safe Composition

$$f(\vec{x}; \vec{y}) = g(\vec{x}; h_1(\vec{x}; \vec{y}), \ldots, h_k(\vec{x}; \vec{y}))$$

Bellantoni & Cooks Definition of FP

Predicative Recursion on Notation

$$f(\varepsilon, \vec{x}; \vec{y}) = g(\vec{x}; \vec{y})$$
$$f(zi, \vec{x}; \vec{y}) = h_i(z, \vec{x}; \vec{y}, f(z, \vec{x}; \vec{y})) \quad \text{for } i = 0, 1$$

- uses separation of arguments

$$f(\underbrace{n_1, \dots, n_l}_{normal}; \underbrace{n_{l+1}, \dots, n_{l+k}}_{safe})$$

Weak Safe Composition

$$f(\vec{x}; \vec{y}) = g(\vec{x}; h_1(\vec{x}; \vec{y}), \dots, h_k(\vec{x}; \vec{y}))$$

- complexity depends essentially only on normal argument and nesting of recursive functions

Small Polynomial Path Order $>_{\mathsf{spop}*}$

$f(s_k, \ldots, s_1; s_{k+l}, \ldots, s_{k+1}) >_{\mathsf{spop}*} t$ if

1. $s_i \geqslant_{\mathsf{spop}*} t$ for some argument s_i

Small Polynomial Path Order $>_{\mathsf{spop}*}$

$f(s_k, \ldots, s_1; s_{k+l}, \ldots, s_{k+1}) >_{\mathsf{spop}*} t$ if

1. $s_i \geqslant_{\mathsf{spop}*} t$ for some argument s_i
2. $t = g(t_m, \ldots, t_1; t_{m+n}, \ldots, t_{m+1})$ where $f \in \mathcal{D}$ and $f > g$
   - $f(s_k, \ldots, s_1; s_{k+l}, \ldots, s_{k+1}) \rhd_{\mathsf{n}} t_j$ for all normal arguments t_j
   - $f(s_k, \ldots, s_1; s_{k+l}, \ldots, s_{k+1}) >_{\mathsf{spop}*} t_j$ for all safe arguments t_j
   - t contains symbol f at most once

$f(s_1, \ldots, s_k; \ldots) \rhd_{\mathsf{n}} t :\Leftrightarrow t$ is subterm of normal argument s_i

Small Polynomial Path Order $>_{\mathsf{spop*}}$

$f(s_k, \ldots, s_1; s_{k+l}, \ldots, s_{k+1}) >_{\mathsf{spop*}} t$ if

1. $s_i \geqslant_{\mathsf{spop*}} t$ for some argument s_i
2. $t = g(t_m, \ldots, t_1; t_{m+n}, \ldots, t_{m+1})$ where $f \in \mathcal{D}$ and $f > g$
   - $f(s_k, \ldots, s_1; s_{k+l}, \ldots, s_{k+1}) \rhd_{\mathsf{n}} t_j$ for all normal arguments t_j
   - $f(s_k, \ldots, s_1; s_{k+l}, \ldots, s_{k+1}) >_{\mathsf{spop*}} t_j$ for all safe arguments t_j
   - t contains symbol f at most once

Small Polynomial Path Order $>_{\mathsf{spop*}}$

$f(s_k, \ldots, s_1; s_{k+l}, \ldots, s_{k+1}) >_{\mathsf{spop*}} t$ if

1. $s_i \geqslant_{\mathsf{spop*}} t$ for some argument s_i
2. $t = g(t_m, \ldots, t_1; t_{m+n}, \ldots, t_{m+1})$ where $f \in \mathcal{D}$ and $f > g$
   - $f(s_k, \ldots, s_1; s_{k+l}, \ldots, s_{k+1}) \rhd_{\mathsf{n}} t_j$ for all normal arguments t_j
   - $f(s_k, \ldots, s_1; s_{k+l}, \ldots, s_{k+1}) >_{\mathsf{spop*}} t_j$ for all safe arguments t_j
   - t contains symbol f at most once

$$f(\vec{x}; \vec{y}) >_{\mathsf{spop*}} g(\vec{x}; h_1(\vec{x}; \vec{y}), \ldots, h_k(\vec{x}; \vec{y}))$$

Small Polynomial Path Order $>_{\mathsf{spop}*}$

$f(s_k, \dots, s_1; s_{k+l}, \dots, s_{k+1}) >_{\mathsf{spop}*} t$ if

1. $s_i \geqslant_{\mathsf{spop}*} t$ for some argument s_i
2. $t = g(t_m, \dots, t_1; t_{m+n}, \dots, t_{m+1})$ where $f \in \mathcal{D}$ and $f > g$
   - $f(s_k, \dots, s_1; s_{k+l}, \dots, s_{k+1}) \rhd_{\mathsf{n}} t_j$ for all normal arguments t_j
   - $f(s_k, \dots, s_1; s_{k+l}, \dots, s_{k+1}) >_{\mathsf{spop}*} t_j$ for all safe arguments t_j
   - t contains symbol f at most once
3. $t = f(t_k, \dots, t_1; t_{k+l}, \dots, t_{k+1})$ where $f \in \mathcal{D}$
   - $\langle s_1, \dots, s_k \rangle >_{\mathsf{spop}*} \langle t_{\pi(1)}, \dots, t_{\pi(k)} \rangle$
   - $\langle s_{k+1}, \dots, s_{k+l} \rangle >_{\mathsf{spop}*} \langle t_{\tau(k+1)}, \dots, t_{\tau(k+l)} \rangle$

Small Polynomial Path Order $>_{\mathsf{spop}*}$

$f(s_k, \dots, s_1; s_{k+l}, \dots, s_{k+1}) >_{\mathsf{spop}*} t$ if

1. $s_i \geqslant_{\mathsf{spop}*} t$ for some argument s_i
2. $t = g(t_m, \dots, t_1; t_{m+n}, \dots, t_{m+1})$ where $f \in \mathcal{D}$ and $f > g$
   - $f(s_k, \dots, s_1; s_{k+l}, \dots, s_{k+1}) \rhd_{\mathsf{n}} t_j$ for all normal arguments t_j
   - $f(s_k, \dots, s_1; s_{k+l}, \dots, s_{k+1}) >_{\mathsf{spop}*} t_j$ for all safe arguments t_j
   - t contains symbol f at most once
3. $t = f(t_k, \dots, t_1; t_{k+l}, \dots, t_{k+1})$ where $f \in \mathcal{D}$
   - $\langle s_1, \dots, s_k \rangle >_{\mathsf{spop}*} \langle t_{\pi(1)}, \dots, t_{\pi(k)} \rangle$
   - $\langle s_{k+1}, \dots, s_{k+l} \rangle >_{\mathsf{spop}*} \langle t_{\tau(k+1)}, \dots, t_{\tau(k+l)} \rangle$

$$f(\varepsilon, \vec{x}; \vec{y}) >_{\mathsf{spop}*} g(\vec{x}; \vec{y})$$
$$f(zi, \vec{x}; \vec{y}) >_{\mathsf{spop}*} h_i(z, \vec{x}; \vec{y}, f(z, \vec{x}; \vec{y})) \quad \text{for } i = 0, 1$$

Small Polynomial Path Order $>_{\mathsf{spop}*}$

$f(s_k, \ldots, s_1; s_{k+l}, \ldots, s_{k+1}) >_{\mathsf{spop}*} t$ if

1. $s_i \geqslant_{\mathsf{spop}*} t$ for some argument s_i
2. $t = g(t_m, \ldots, t_1; t_{m+n}, \ldots, t_{m+1})$ where $f \in \mathcal{D}$ and $f > g$
   - $f(s_k, \ldots, s_1; s_{k+l}, \ldots, s_{k+1}) \rhd_{\mathsf{n}} t_j$ for all normal arguments t_j
   - $f(s_k, \ldots, s_1; s_{k+l}, \ldots, s_{k+1}) >_{\mathsf{spop}*} t_j$ for all safe arguments t_j
   - t contains symbol f at most once
3. $t = f(t_k, \ldots, t_1; t_{k+l}, \ldots, t_{k+1})$ where $f \in \mathcal{D}_{\mathsf{rec}}$
   - $\langle s_1, \ldots, s_k \rangle >_{\mathsf{spop}*} \langle t_{\pi(1)}, \ldots, t_{\pi(k)} \rangle$
   - $\langle s_{k+1}, \ldots, s_{k+l} \rangle >_{\mathsf{spop}*} \langle t_{\tau(k+1)}, \ldots, t_{\tau(k+l)} \rangle$

assume designated set of recursive symbols $\mathcal{D}_{\mathsf{rec}} \subseteq \mathcal{D}$

Small Polynomial Path Order $>_{\mathsf{spop}*}$

$f(s_k, \dots, s_1; s_{k+l}, \dots, s_{k+1}) >_{\mathsf{spop}*} t$ if

1. $s_i \geqslant_{\mathsf{spop}*} t$ for some argument s_i

2. $t = g(t_m, \dots, t_1; t_{m+n}, \dots, t_{m+1})$ where $f \in \mathcal{D}$ and $f > g$
   - $f(s_k, \dots, s_1; s_{k+l}, \dots, s_{k+1}) \rhd_{\mathsf{n}} t_j$ for all normal arguments t_j
   - $f(s_k, \dots, s_1; s_{k+l}, \dots, s_{k+1}) >_{\mathsf{spop}*} t_j$ for all safe arguments t_j
   - t contains symbol f at most once

3. $t = f(t_k, \dots, t_1; t_{k+l}, \dots, t_{k+1})$ where $f \in \mathcal{D}_{\mathsf{rec}}$
   - $\langle s_1, \dots, s_k \rangle >_{\mathsf{spop}*} \langle t_{\pi(1)}, \dots, t_{\pi(k)} \rangle$
   - $\langle s_{k+1}, \dots, s_{k+l} \rangle >_{\mathsf{spop}*} \langle t_{\tau(k+1)}, \dots, t_{\tau(k+l)} \rangle$

Small Polynomial Path Order $>_{\mathsf{spop}*}$

Runtime Complexity Analysis

- the depth of recursion $\mathsf{rd}(f)$ of symbol f is defined as follows:

$$\mathsf{rd}_{>}(f) := \begin{cases} 1 + \max\{\mathsf{rd}(g) \mid f > g\} & \text{if } f \in \mathcal{D}_{\mathsf{rec}} \\ \max\{\mathsf{rd}(g) \mid f > g\} & \text{otherwise} \end{cases}$$

Small Polynomial Path Order $>_{\mathsf{spop}*}$

Runtime Complexity Analysis

- the depth of recursion $\mathsf{rd}(f)$ of symbol f is defined as follows:

$$\mathsf{rd}_{>}(f) := \begin{cases} 1 + \max\{\mathsf{rd}(g) \mid f > g\} & \text{if } f \in \mathcal{D}_{\mathsf{rec}} \\ \max\{\mathsf{rd}(g) \mid f > g\} & \text{otherwise} \end{cases}$$

Theorem

Suppose $\mathcal{R}$ is *constructor* TRS compatible with $>_{\mathsf{spop}*}$. $\mathcal{R} \subseteq >_{\mathsf{spop}*}$

Then the *innermost runtime complexity* of $\mathcal{R}$ is polynomial, where the degree of the polynomial is the maximal depth of recursion.

Small Polynomial Path Order $>_{\mathsf{spop*}}$

Runtime Complexity Analysis

Example

$$
\begin{array}{llll}
1: & +(0; y) \to y & 3: & \times(0, y;) \to 0 \\
2: & +(\mathsf{s}(x); y) \to \mathsf{s}(+(x; y)) & 4: & \times(\mathsf{s}(x), y;) \to +(y; \times(x, y;)) \\
5: & \mathsf{sq}(x;) \to \times(x, x;) & &
\end{array}
$$

- TRS is compatible with $>_{\mathsf{spop*}}$ using precedence $\mathsf{sq} > \times > + > \mathsf{s}$
- only $\times$ and $+$ are recursive, but not sq
- innermost runtime complexity is bounded by quadratic polynomial

small polynomial path orders in TCT

Complexity Problem in TCT

- **(complexity) problem** $\mathcal{P}$ is tuple $\langle \mathcal{S}/\mathcal{W}, \mathcal{Q}, \mathcal{T} \rangle$
  1. $\to_{\mathcal{P}}$ is binary relation on terms

$$\xrightarrow{\mathcal{Q}}_{\mathcal{S}/\mathcal{W}} := \xrightarrow{\mathcal{Q}}{}^{*}_{\mathcal{W}} \cdot \xrightarrow{\mathcal{Q}}_{\mathcal{S}} \cdot \xrightarrow{\mathcal{Q}}{}^{*}_{\mathcal{W}}$$

    - $\mathcal{S}$,$\mathcal{W}$ are TRSs
    - $s \xrightarrow{\mathcal{Q}}_{\mathcal{R}} t$ if $s \to_{\mathcal{R}} t$ and arguments of redex in s are $\mathcal{Q}$ normal forms

  2. $\mathcal{T}$ is set of starting terms

- **complexity function** of $\mathcal{P}$ is

$$\mathsf{cp}_{\mathcal{P}}(n) := \max\{\mathsf{dh}(t, \to_{\mathcal{P}}) \mid t \in \mathcal{T} \text{ is term of size upto } n\}$$

Complexity Problem in TCT

- **(complexity) problem** $\mathcal{P}$ is tuple $\langle \mathcal{S}/\mathcal{W}, \mathcal{Q}, \mathcal{T} \rangle$

  ❶ $\rightarrow_{\mathcal{P}}$ is binary relation on terms

  $$\xrightarrow{\mathcal{Q}}_{\mathcal{S}/\mathcal{W}} := \xrightarrow{\mathcal{Q}}^{*}_{\mathcal{W}} \cdot \xrightarrow{\mathcal{Q}}_{\mathcal{S}} \cdot \xrightarrow{\mathcal{Q}}^{*}_{\mathcal{W}}$$

  - $\mathcal{S}$,$\mathcal{W}$ are TRSs
  - $s \xrightarrow{\mathcal{Q}}_{\mathcal{R}} t$ if $s \rightarrow_{\mathcal{R}} t$ and arguments of redex in s are $\mathcal{Q}$ normal forms

  ❷ $\mathcal{T}$ is set of starting terms

- **complexity function** of $\mathcal{P}$ is

  $$\mathrm{cp}_{\mathcal{P}}(n) := \max\{\mathrm{dh}(t, \rightarrow_{\mathcal{P}}) \mid t \in \mathcal{T} \text{ is term of size upto } n\}$$

- innermost problem if normal forms of $\mathcal{Q}$ are normal forms of $\mathcal{S}$, $\mathcal{W}$
- runtime problem if start terms $\mathcal{T}$ are constructor based

Small Polynomial Path Order $>_{\mathsf{spop}*}$

on complexity problems

Theorem *(A)*

Consider innermost runtime problem $\mathcal{P} = \langle \mathcal{S}/\mathcal{W}, \mathcal{Q}, \mathcal{T} \rangle$

1. $\mathcal{S}$ and $\mathcal{W}$ are *constructor* TRSs
2. $\mathcal{S} \subseteq >_{\mathsf{spop}*}$ and $\mathcal{W} \subseteq \geqslant_{\mathsf{spop}*}$

Then the complexity of $\mathcal{P}$ is polynomial, where the degree of the polynomial is the maximal depth of recursion.

Small Polynomial Path Order $>_{\mathsf{spop}*}$

on complexity problems

Example

consider innermost runtime problem $\mathcal{P}_\times := \langle \{3,4\}/\{1,2\}, \{1-4\}, \mathcal{T}\rangle$

$$1: \quad +(0,y) \to y \qquad\qquad 3: \quad \times(0,y) \to 0$$

$$2: +(\mathsf{s}(x),y) \to \mathsf{s}(+(x,y)) \qquad\qquad 4: \times(\mathsf{s}(x),y) \to +(y,\times(x,y))$$

Observations

- complexity analysis needs to trace only applications of $\times$-rules
- $\times$-rule applications only second argument of $+$

Small Polynomial Path Order $>_{\mathsf{spop}*}$

on complexity problems

Example

consider innermost runtime problem $\mathcal{P}_\times := \langle \{3,4\}/\{1,2\}, \{1-4\}, \mathcal{T} \rangle$

$$1: \quad +(0,y) \to y \qquad\qquad 3: \quad \times(0,y) \to 0$$
$$2: +(\mathsf{s}(x),y) \to \mathsf{s}(+(x,y)) \qquad\qquad 4: \times(\mathsf{s}(x),y) \to +(y,\times(x,y))$$

Observations

- complexity analysis needs to trace only applications of $\times$-rules
- $\times$-rule applications only second argument of $+$

Idea

- arguments never containing $\times$-redex can be dropped
- to this end, incorporate argument filtering in order

Small Polynomial Path Order $>_{\mathsf{spop}*}$

with argument filtering

- argument filtering π is mapping on function symbols f to argument position or list of argument positions

Small Polynomial Path Order $>_{\mathsf{spop}*}$

with argument filtering

- argument filtering π is mapping on function symbols f to argument position or list of argument positions
- for order $>$ on terms, define

$$s >^{\pi} t :\Leftrightarrow \pi(s) > \pi(t)$$

$$\pi(t) := \begin{cases} t & \text{if } t \text{ a variable} \\ \pi(t_i) & \text{if } t = f(t_1, \dots, t_n) \text{ and } \pi(f) = i \\ f(\pi(t_{i_1}), \dots, \pi(t_{i_k})) & \text{if } t = f(t_1, \dots, t_n) \text{ and } \pi(f) = [i_1, \dots, i_k] \end{cases}$$

Small Polynomial Path Order $>_{\mathsf{spop}*}$

with argument filtering

Theorem *(B)*

Consider innermost runtime problem $\mathcal{P} = \langle \mathcal{S}/\mathcal{W}, \mathcal{Q}, \mathcal{T} \rangle$

1. $\mathcal{S}$ and $\mathcal{W}$ are *constructor* TRSs
2. $\mathcal{S} \subseteq >^{\pi}_{\mathsf{spop}*}$ and $\mathcal{W} \subseteq \geqslant^{\pi}_{\mathsf{spop}*}$
3. π is argument filtering
   - "π does not delete $\mathcal{S}$-redexes" *account for all $\mathcal{S}$-applications*

Then the complexity of $\mathcal{P}$ is polynomial, where the degree of the polynomial is the maximal depth of recursion.

Small Polynomial Path Order $>_{\mathsf{spop}*}$

with argument filtering

Theorem *(B)*

Consider innermost runtime problem $\mathcal{P} = \langle \mathcal{S}/\mathcal{W}, \mathcal{Q}, \mathcal{T} \rangle$

1. $\mathcal{S}$ and $\mathcal{W}$ are *constructor* TRSs
2. $\mathcal{S} \subseteq >^{\pi}_{\mathsf{spop}*}$ and $\mathcal{W} \subseteq \geqslant^{\pi}_{\mathsf{spop}*}$
3. π is argument filtering
   - "π does not delete $\mathcal{S}$-redexes" — *account for all $\mathcal{S}$-applications*
   - $\pi(f)$ is list for every symbol f defined by $\mathcal{S}$ — *don't break predicative recursion*

Then the complexity of $\mathcal{P}$ is polynomial, where the degree of the polynomial is the maximal depth of recursion.

Small Polynomial Path Order $>_{\mathsf{spop}*}$

with argument filtering – Example 1

Example

consider innermost runtime problem $\mathcal{P}_\times := \langle \{3,4\}/\{1,2\}, \{1-4\}, \mathcal{T} \rangle$

$$
\begin{array}{rlrl}
1: & +(0, y) \to y & 3: & \times(0, y) \to 0 \\
2: & +(\mathsf{s}(x), y) \to \mathsf{s}(+(x, y)) & 4: & \times(\mathsf{s}(x), y) \to +(y, \times(x, y))
\end{array}
$$

Small Polynomial Path Order $>_{\mathsf{spop}*}$

with argument filtering – Example 1

Example

consider innermost runtime problem $\mathcal{P}_\times := \langle \{3,4\}/\{1,2\}, \{1-4\}, \mathcal{T} \rangle$

$$
\begin{aligned}
1:&\quad +(0, y) \to y & 3:&\quad \times(0, y) \to 0 \\
2:&\quad +(\mathsf{s}(x), y) \to \mathsf{s}(+(x, y)) & 4:&\quad \times(\mathsf{s}(x), y) \to +(y, \times(x, y))
\end{aligned}
$$

$\pi(\mathsf{s}) = \{1\}$ $\quad$ $\pi(+) = \{1, 2\}$ $\quad$ $\pi(\times) = \{1, 2\}$ ✓

Small Polynomial Path Order $>_{\mathsf{spop}*}$

with argument filtering – Example 1

Example

consider innermost runtime problem $\mathcal{P}_\times := \langle \{3,4\}/\{1,2\}, \{1-4\}, \mathcal{T} \rangle$

$$
\begin{array}{llll}
1: & +(0,y) \to y & 3: & \times(0,y) \to 0 \\
2: & +(\mathsf{s}(x),y) \to \mathsf{s}(+(x,y)) & 4: & \times(\mathsf{s}(x),y) \to +(y,\times(x,y))
\end{array}
$$

$\pi(\mathsf{s}) = \{1\}$ $\pi(+) = \{1,2\}$ $\pi(\times) = \{1,2\}$ ✓

$\pi(\mathsf{s}) = \{1\}$ $\pi(+) = 2$ $\pi(\times) = \{1,2\}$ ✓

Small Polynomial Path Order $>_{\mathsf{spop}*}$

with argument filtering – Example 1

Example

consider innermost runtime problem $\mathcal{P}_\times := \langle \{3,4\}/\{1,2\}, \{1-4\}, \mathcal{T}\rangle$

$$1: \quad +(0,y) \to y \qquad\qquad 3: \quad \times(0,y) \to 0$$

$$2: +(\mathsf{s}(x),y) \to \mathsf{s}(+(x,y)) \qquad\qquad 4: \times(\mathsf{s}(x),y) \to +(y,\times(x,y))$$

$\pi(\mathsf{s}) = \{1\}$ $\quad \pi(+) = \{1,2\}$ $\quad \pi(\times) = \{1,2\}$ ✓

$\pi(\mathsf{s}) = \{1\}$ $\quad \pi(+) = 2$ $\quad \pi(\times) = \{1,2\}$ ✓

$\pi(\mathsf{s}) = \{1\}$ $\quad \pi(+) = 2$ $\quad \pi(\times) = \{1\}$ ✓

Small Polynomial Path Order $>_{\mathsf{spop}*}$

with argument filtering – Example 1

Example

consider innermost runtime problem $\mathcal{P}_{\times} := \langle \{3,4\}/\{1,2\}, \{1-4\}, \mathcal{T}\rangle$

$$
\begin{aligned}
1:&\quad +(0,y) \to y & 3:&\quad \times(0,y) \to 0\\
2:&\quad +(\mathsf{s}(x),y) \to \mathsf{s}(+(x,y)) & 4:&\quad \times(\mathsf{s}(x),y) \to +(y,\times(x,y))
\end{aligned}
$$

$\pi(\mathsf{s}) = \{1\}$	$\pi(+) = \{1,2\}$	$\pi(\times) = \{1,2\}$	✓
$\pi(\mathsf{s}) = \{1\}$	$\pi(+) = 2$	$\pi(\times) = \{1,2\}$	✓
$\pi(\mathsf{s}) = \{1\}$	$\pi(+) = 2$	$\pi(\times) = \{1\}$	✓
$\pi(\mathsf{s}) = \{1\}$	$\pi(+) = 1$	$\pi(\times) = \{1\}$	✗

Small Polynomial Path Order $>_{\mathsf{spop}*}$

with argument filtering – Example 1

Example

consider innermost runtime problem $\mathcal{P}_\times := \langle \{3,4\}/\{1,2\}, \{1-4\}, \mathcal{T} \rangle$

$$1: \quad +(0, y) \to y \qquad\qquad 3: \quad \times(0, y) \to 0$$

$$2: +(\mathsf{s}(x), y) \to \mathsf{s}(+(x, y)) \qquad 4: \times(\mathsf{s}(x), y) \to +(y, \times(x, y))$$

$\pi(\mathsf{s}) = \{1\}$	$\pi(+) = \{1, 2\}$	$\pi(\times) = \{1, 2\}$	✓
$\pi(\mathsf{s}) = \{1\}$	$\pi(+) = 2$	$\pi(\times) = \{1, 2\}$	✓
$\pi(\mathsf{s}) = \{1\}$	$\pi(+) = 2$	$\pi(\times) = \{1\}$	✓
$\pi(\mathsf{s}) = \{1\}$	$\pi(+) = 1$	$\pi(\times) = \{1\}$	✗
$\pi(\mathsf{s}) = \{1\}$	$\pi(+) = 2$	$\pi(\times) = 1$	✗

Small Polynomial Path Order $>_{\mathsf{spop}*}$

with argument filtering – Example 2

Example

consider innermost runtime problem $\mathcal{P}_{\mathsf{log}} := \langle \{3,4\}/\{1,2\}, \{1-4\}, \mathcal{T} \rangle$

$$1: \quad \mathsf{half}(0) \to 0 \qquad\qquad 3: \mathsf{half}^\sharp(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{half}^\sharp(x)$$

$$2: \mathsf{half}(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{s}(\mathsf{half}(x)) \qquad 4: \mathsf{log}^\sharp(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{log}^\sharp(\mathsf{s}(\mathsf{half}(x)))$$

- obtained by dependency pair transformation on **AG01/#3.7**
- rule 4 not orientable by $>_{\mathsf{spop}*}$

Small Polynomial Path Order $>_{\mathsf{spop}*}$

with argument filtering – Example 2

Example

consider innermost runtime problem $\mathcal{P}_{\mathsf{log}} := \langle \{3,4\}/\{1,2\}, \{1-4\}, \mathcal{T}\rangle$

$$
\begin{aligned}
1 &: \quad \mathsf{half}(0) \to 0 & 3 &: \mathsf{half}^\sharp(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{half}^\sharp(x) \\
2 &: \mathsf{half}(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{s}(\mathsf{half}(x)) & 4 &: \mathsf{log}^\sharp(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{log}^\sharp(\mathsf{s}(\mathsf{half}(x)))
\end{aligned}
$$

- take argument filtering

$$\pi(\mathsf{half}) = 1 \quad \pi(\mathsf{s}) = \{1\} \quad \pi(\mathsf{half}^\sharp) = \{1\} \quad \pi(\mathsf{log}^\sharp) = \{1\}$$

- take empty precedence, recursive symbols are $\mathsf{half}^\sharp, \mathsf{log}^\sharp$

$$
\begin{aligned}
1 &: \quad 0 \geqslant_{\mathsf{spop}*} 0 & 3 &: \mathsf{half}^\sharp(\mathsf{s}(\mathsf{s}(x));) >_{\mathsf{spop}*} \mathsf{half}^\sharp(x;) \\
2 &: \mathsf{s}(\mathsf{s}(x)) \geqslant_{\mathsf{spop}*} \mathsf{s}(x) & 4 &: \mathsf{log}^\sharp(\mathsf{s}(\mathsf{s}(x));) >_{\mathsf{spop}*} \mathsf{log}^\sharp(\mathsf{s}(x);)
\end{aligned}
$$

Small Polynomial Path Order $>_{\mathsf{spop}*}$

with argument filtering – Example 2

Example

consider innermost runtime problem $\mathcal{P}_{\mathsf{log}} := \langle \{3,4\} / \{1,2\}, \{1-4\}, \mathcal{T} \rangle$

$$1: \quad \mathsf{half}(0) \to 0 \qquad 3: \mathsf{half}^\sharp(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{half}^\sharp(x)$$

$$2: \mathsf{half}(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{s}(\mathsf{half}(x)) \qquad 4: \mathsf{log}^\sharp(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{log}^\sharp(\mathsf{s}(\mathsf{half}(x)))$$

- take argument filtering

$$\pi(\mathsf{half}) = 1 \quad \pi(\mathsf{s}) = \{1\} \quad \pi(\mathsf{half}^\sharp) = \{1\} \quad \pi(\mathsf{log}^\sharp) = \{1\}$$

- take empty precedence, recursive symbols are $\mathsf{half}^\sharp, \mathsf{log}^\sharp$

- maximal recursion depth is one $\Rightarrow$ complexity of $\mathcal{P}_{\mathsf{log}}$ is linear

Small Polynomial Path Order $>_{\mathsf{spop}*}$

with argument filtering – Example 2

Example

consider innermost runtime problem $\mathcal{P}_{\mathsf{log}} := \langle \{3, 4\} / \{1, 2\}, \{1 - 4\}, \mathcal{T} \rangle$

$$
\begin{aligned}
1 &: \quad \mathsf{half}(0) \to 0 & 3 &: \mathsf{half}^{\sharp}(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{half}^{\sharp}(x) \\
2 &: \mathsf{half}(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{s}(\mathsf{half}(x)) & 4 &: \mathsf{log}^{\sharp}(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{log}^{\sharp}(\mathsf{s}(\mathsf{half}(x)))
\end{aligned}
$$

- take argument filtering

$$\pi(\mathsf{half}) = 1 \qquad \pi(\mathsf{s}) = \{1\} \qquad \pi(\mathsf{half}^{\sharp}) = 1 \qquad \pi(\mathsf{log}^{\sharp}) = 1 \qquad ✗$$

- take empty precedence

$$
\begin{aligned}
1 &: \quad 0 \geqslant_{\mathsf{spop}*} 0 & 3 &: \mathsf{s}(\mathsf{s}(x)) >_{\mathsf{spop}*} x \\
2 &: \mathsf{s}(\mathsf{s}(x)) \geqslant_{\mathsf{spop}*} \mathsf{s}(x) & 4 &: \mathsf{s}(\mathsf{s}(x)) >_{\mathsf{spop}*} \mathsf{s}(x)
\end{aligned}
$$

Small Polynomial Path Order $>_{\mathsf{spop}*}$

with argument filtering – Example 2

Example

consider innermost runtime problem $\mathcal{P}_{\mathsf{log}} := \langle \{3, 4\} / \{1, 2\}, \{1 - 4\}, \mathcal{T} \rangle$

$$1 : \mathsf{half}(0) \to 0 \qquad 3 : \mathsf{half}^\sharp(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{half}^\sharp(x)$$

$$2 : \mathsf{half}(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{s}(\mathsf{half}(x)) \qquad 4 : \mathsf{log}^\sharp(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{log}^\sharp(\mathsf{s}(\mathsf{half}(x)))$$

- take argument filtering

$$\pi(\mathsf{half}) = 1 \quad \pi(\mathsf{s}) = \{1\} \quad \pi(\mathsf{half}^\sharp) = 1 \quad \pi(\mathsf{log}^\sharp) = 1 \quad ✗$$

- take empty precedence
- maximal recursion depth is 0 $\Rightarrow$ complexity of $\mathcal{P}_{\mathsf{log}}$ wrongly inferred constant

Conclusion

- small polynomial path orders have been extended to notion of complexity problem in TCT
  - *complexity pair* $(\geqslant_{\mathsf{spop}*}, >_{\mathsf{spop}*})$ used in relative setting
  - *argument filterings* π integrated

Conclusion

- small polynomial path orders have been extended to notion of complexity problem in TCT
  - *complexity pair* $(\geqslant_{\mathsf{spop}*}, >_{\mathsf{spop}*})$ used in relative setting
  - *argument filterings* π integrated

not shown...

- *quasi-precedence* can be used

Conclusion

- small polynomial path orders have been extended to notion of complexity problem in TCT
  - *complexity pair* $(\geqslant_{\mathsf{spop*}}, >_{\mathsf{spop*}})$ used in relative setting
  - *argument filterings* π integrated

not shown. . .

- *quasi-precedence* can be used
- order can be extended to allow *parameter substitution*

Conclusion

- small polynomial path orders have been extended to notion of complexity problem in TCT
  - *complexity pair* $(\geqslant_{\mathsf{spop}*}, >_{\mathsf{spop}*})$ used in relative setting
  - *argument filterings* π integrated

not shown...

- *quasi-precedence* can be used
- order can be extended to allow *parameter substitution*
- generalises *safe reduction pairs* on (innermost) DP problems

Conclusion

- small polynomial path orders have been extended to notion of complexity problem in TCT
  - *complexity pair* $(\geqslant_{\mathsf{spop}*}, >_{\mathsf{spop}*})$ used in relative setting
  - *argument filterings* π integrated

not shown. . .

- *quasi-precedence* can be used
- order can be extended to allow *parameter substitution*
- generalises *safe reduction pairs* on (innermost) DP problems
- *non-collapsing* constraint on π can be *dropped*
  - degree of bounding function doubles
  - bound is tight

Experiments

bound	sPOP*	DP+sPOP*	DP+MI
O(1)	4/0.17	20/0.28	20/0.27
$O(n)$	20/0.17	72/0.31	98/0.48
$O(n^2)$	23/0.19	11/0.44	17/4.67
$O(n^3)$	6/0.23	3/0.60	8/14.7
total	54/0.19	106/0.32	143/1.55
maybe	703/0.34	651/1.20	614/18.3

Table : # oriented problems / average execution times (secs.)

setup

- 757 well-formed constructor TRSs from TPDB 8.0
- 8 Dual-Core Opteron™ 885 processors (2.6GHz)

Experiments

bound	sPOP*	DP+sPOP*	DP+MI
O(1)	4/0.17	20/0.28	20/0.27
O(n)	20/0.17	72/0.31	98/0.48
O(n^2)	23/0.19	11/0.44	17/4.67
O(n^3)	6/0.23	3/0.60	8/14.7
total	**54**/0.19	**106**/0.32	143/1.55
maybe	703/0.34	651/1.20	614/18.3

Table : # oriented problems / average execution times (secs.)

setup

- 757 well-formed constructor TRSs from TPDB 8.0
- 8 Dual-Core Opteron™ 885 processors (2.6GHz)

Experiments

bound	sPOP*	DP+sPOP*	DP+MI
O(1)	4/0.17	20/0.28	20/0.27
$O(n)$	20/0.17	72/0.31	98/0.48
$O(n^2)$	23/0.19	11/0.44	17/4.67
$O(n^3)$	6/0.23	3/0.60	8/14.7
total	54/0.19	**106**/0.32	**143**/1.55
maybe	703/0.34	651/1.20	614/18.3

Table : # oriented problems / average execution times (secs.)

setup

- 757 well-formed constructor TRSs from TPDB 8.0
- 8 Dual-Core Opteron™ 885 processors (2.6GHz)

Experiments

bound	sPOP*	DP+sPOP*	DP+MI
O(1)	4/0.17	20/0.28	20/0.27
O(n)	20/0.17	72/0.31	98/0.48
O(n^2)	23/0.19	11/0.44	17/4.67
O(n^3)	6/0.23	3/0.60	8/14.7
total	54/0.19	106/**0.32**	143/**1.55**
maybe	703/0.34	651/1.20	614/18.3

Table : # oriented problems / average execution times (secs.)

setup

- 757 well-formed constructor TRSs from TPDB 8.0
- 8 Dual-Core Opteron™ 885 processors (2.6GHz)

Experiments

bound	sPOP*	DP+sPOP*	DP+MI	$\sum$
O(1)	4/0.17	20/0.28	20/0.27	20
O(n)	20/0.17	72/0.31	98/0.48	98
O(n^2)	23/0.19	11/0.44	17/4.67	22
O(n^3)	6/0.23	3/0.60	8/14.7	9
total	**54**/0.19	**106**/0.32	**143**/1.55	**149**
maybe	703/0.34	651/1.20	614/18.3	608

Table : # oriented problems / average execution times (secs.)

setup

- 757 well-formed constructor TRSs from TPDB 8.0
- 8 Dual-Core Opteron™ 885 processors (2.6GHz)