Exponential Path Order EPO*

Martin Avanzini[1] **Naohi Eguchi**[2] **Georg Moser**[1]

[1]Computational Logic
Faculty of Computer Science, University of Innsbruck, Austria

[2]School of Information Science
Japan Advanced Institute of Science and Technology, Japan

June 1 @ RTA'11

polynomial time

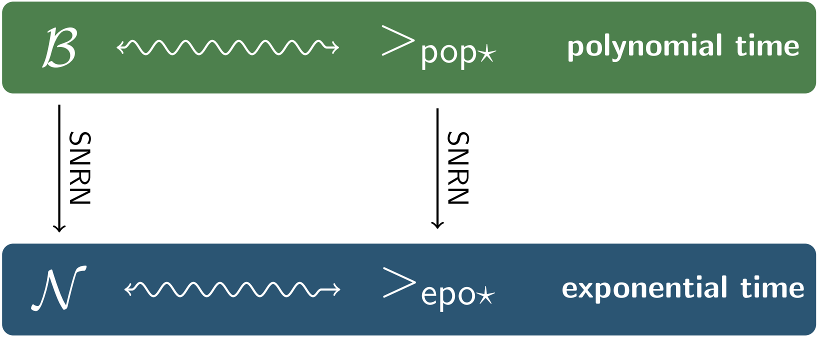
$\mathcal{B}$ polynomial time

Stephen Bellantoni and Stephen Cook
A new Recursion-Theoretic Characterization of the Polytime Functions.
CC, pages 97–110, 1992

$\mathcal{B}$ $\leftrightsquigarrow$ $>_{\mathsf{pop}\star}$ polynomial time

Martin Avanzini and Georg Moser
Complexity Analysis by Rewriting.
FLOPS '09, pages 130–146, 2008

Toshiyasu Arai and Naohi Eguchi
A new Function Algebra of EXPTIME Functions by Safe Nested Recursion.
TCL, pages 130–146, 2008



Martin Avanzini and Naohi Eguchi and Georg Moser
A Path Order for Rewrite Systems that Compute Exponential Time Functions.
RTA'11, pages 123–138, 2011

Main Result

Let FEXP denote class of functions computable in time $2^{O(n^k)}$ $(k \in \mathbb{N})$

Main Result

Let FEXP denote class of functions computable in time $2^{O(n^k)}$ $(k \in \mathbb{N})$

1. **Soundness**
   Let $\mathcal{R}$ be a constructor TRS that computes a function f.
   If $\mathcal{R} \subseteq >_{\mathsf{epo}\star}$ then $f \in \mathsf{FEXP}$.

Main Result

Let FEXP denote class of functions computable in time $2^{O(n^k)}$ $(k \in \mathbb{N})$

1. **Soundness**
   Let $\mathcal{R}$ be a constructor TRS that computes a function f.
   If $\mathcal{R} \subseteq >_{\mathsf{epo}\star}$ then $f \in \mathsf{FEXP}$.

2. **Completeness**
   Let $f \in \mathsf{FEXP}$.
   There exists a constructor TRS $\mathcal{R}_f$ computing f with $\mathcal{R}_f \subseteq >_{\mathsf{epo}\star}$.

Main Result

Let FEXP denote class of functions computable in time $2^{O(n^k)}$ $(k \in \mathbb{N})$

1. **Soundness**
   Let $\mathcal{R}$ be a constructor TRS that computes a function f.
   If $\mathcal{R} \subseteq >_{\mathsf{epo}\star}$ then $f \in$ FEXP.

2. **Completeness**
   Let $f \in$ FEXP.
   There exists a constructor TRS $\mathcal{R}_f$ computing f with $\mathcal{R}_f \subseteq >_{\mathsf{epo}\star}$.

Rewriting as Computational Model

We suppose ... in this talk

- $\mathcal{R}$ confluent and terminating
- signature $\mathcal{F}$ underlying TRS $\mathcal{R}$ partitioned into defined symbols $\mathcal{D}$ and constructors $\mathcal{C}$
- values $\mathcal{V}\mathsf{al} := \mathcal{T}(\mathcal{C}, \mathcal{V})$ are terms over constructors $\mathcal{C}$

Rewriting as Computational Model

We suppose . . . in this talk

- $\mathcal{R}$ confluent and terminating
- signature $\mathcal{F}$ underlying TRS $\mathcal{R}$ partitioned into defined symbols $\mathcal{D}$ and constructors $\mathcal{C}$
- values $\mathcal{V}\mathsf{al} := \mathcal{T}(\mathcal{C}, \mathcal{V})$ are terms over constructors $\mathcal{C}$

Definition

TRS $\mathcal{R}$ computes for each $\mathsf{f} \in \mathcal{D}$ partial function $f : \mathcal{V}\mathsf{al}^k \to \mathcal{V}\mathsf{al}_\perp$ s.t.

$$\forall \vec{s} \in \mathcal{V}\mathsf{al}^k \,.\, f(\vec{s}) = t \quad :\Longleftrightarrow \quad \mathsf{f}(\vec{s}) \to^!_{\mathcal{R}} t \text{ and } t \in \mathcal{V}\mathsf{al}$$

Rewriting as Computational Model

We suppose . . . in this talk

- $\mathcal{R}$ confluent and terminating
- signature $\mathcal{F}$ underlying TRS $\mathcal{R}$ partitioned into defined symbols $\mathcal{D}$ and constructors $\mathcal{C}$
- values $\mathcal{V}\mathsf{al} := \mathcal{T}(\mathcal{C}, \mathcal{V})$ are terms over constructors $\mathcal{C}$

Definition

TRS $\mathcal{R}$ computes for each $\mathsf{f} \in \mathcal{D}$ partial function $f : \mathcal{V}\mathsf{al}^k \to \mathcal{V}\mathsf{al}_\perp$ s.t.

$$\forall \vec{s} \in \mathcal{V}\mathsf{al}^k \, . \; f(\vec{s}) = t \quad :\Longleftrightarrow \quad \mathsf{f}(\vec{s}) \xrightarrow{\mathsf{i}}{}^{!}_{\mathcal{R}} \; t \text{ and } t \in \mathcal{V}\mathsf{al}$$

Exponential Path Order $>_{\mathsf{epo}\star}$

- constraints of $\mathcal{N}$ imposed on lexicographic path order $\quad >_{\mathsf{epo}\star} \subseteq >_{\mathsf{lpo}}$

Exponential Path Order $>_{\mathsf{epo}\star}$

- constraints of $\mathcal{N}$ imposed on lexicographic path order $>_{\mathsf{epo}\star} \subseteq >_{\mathsf{lpo}}$

  $\Longrightarrow$ sufficiently strong for **Completeness**

Exponential Path Order $>_{\mathsf{epo}\star}$

- constraints of $\mathcal{N}$ imposed on lexicographic path order $>_{\mathsf{epo}\star} \subseteq >_{\mathsf{lpo}}$

  $\Longrightarrow$ sufficiently strong for **Completeness**
- induces exponential bound on innermost runtime complexity $\mathsf{rc}^{\mathsf{i}}_{\mathcal{R}}$

Exponential Path Order $>_{\mathsf{epo}\star}$

- constraints of $\mathcal{N}$ imposed on lexicographic path order $>_{\mathsf{epo}\star} \subseteq >_{\mathsf{lpo}}$

  $\Longrightarrow$ sufficiently strong for **Completeness**
- induces exponential bound on innermost runtime complexity $\mathsf{rc}^{\mathsf{i}}_{\mathcal{R}}$

Theorem

Let $\mathcal{R}$ denote a constructor TRS. There exists $k \in \mathbb{N}$ such that

$$\mathcal{R} \subseteq >_{\mathsf{epo}\star} \quad \Longrightarrow \quad \mathsf{rc}^{\mathsf{i}}_{\mathcal{R}} \in 2^{\mathsf{O}(n^k)}$$

$$\mathsf{rc}^{\mathsf{i}}_{\mathcal{R}}(n) = \max\{\, \mathsf{dh}(f(\vec{s}), \xrightarrow{\mathsf{i}}_{\mathcal{R}}) \mid \vec{s} \in \mathcal{V}\mathsf{al}^k \text{ and } f(\vec{s}) \text{ of size upto } n \,\}$$

$$\mathsf{dh}(t, \xrightarrow{\mathsf{i}}_{\mathcal{R}}) = \max\{\ell \mid \exists (t_1, \ldots, t_\ell).\ t \xrightarrow{\mathsf{i}}_{\mathcal{R}} t_1 \xrightarrow{\mathsf{i}}_{\mathcal{R}} \ldots \xrightarrow{\mathsf{i}}_{\mathcal{R}} t_\ell\}$$

Exponential Path Order $>_{\mathsf{epo}\star}$

- constraints of $\mathcal{N}$ imposed on lexicographic path order $>_{\mathsf{epo}\star} \subseteq >_{\mathsf{lpo}}$

  $\Longrightarrow$ sufficiently strong for **Completeness**

- induces exponential bound on innermost runtime complexity $\mathsf{rc}^{\mathsf{i}}_{\mathcal{R}}$

  $\Longrightarrow$ implies **Soundness**

Theorem

Let $\mathcal{R}$ denote a constructor TRS. There exists $k \in \mathbb{N}$ such that

$$\mathcal{R} \subseteq >_{\mathsf{epo}\star} \quad \Longrightarrow \quad \mathsf{rc}^{\mathsf{i}}_{\mathcal{R}} \in 2^{\mathsf{O}(n^k)}$$

$$\mathsf{rc}^{\mathsf{i}}_{\mathcal{R}}(n) = \max\{\, \mathsf{dh}(f(\vec{s}), \xrightarrow{\mathsf{i}}_{\mathcal{R}}) \mid \vec{s} \in \mathcal{V}\mathsf{al}^k \text{ and } f(\vec{s}) \text{ of size upto } n \,\}$$

$$\mathsf{dh}(t, \xrightarrow{\mathsf{i}}_{\mathcal{R}}) = \max\{\ell \mid \exists (t_1, \dots, t_\ell).\ t \xrightarrow{\mathsf{i}}_{\mathcal{R}} t_1 \xrightarrow{\mathsf{i}}_{\mathcal{R}} \dots \xrightarrow{\mathsf{i}}_{\mathcal{R}} t_\ell\}$$

Exponential Path Order $>_{\mathsf{epo}\star}$

- constraints of $\mathcal{N}$ imposed on lexicographic path order $>_{\mathsf{epo}\star} \subseteq >_{\mathsf{lpo}}$

  $\Longrightarrow$ sufficiently strong for **Completeness**

- induces exponential bound on innermost runtime complexity $\mathsf{rc}^{\mathsf{i}}_{\mathcal{R}}$

  $\Longrightarrow$ implies **Soundness**

Ugo Dal Lago and Simone Martini
On Constructor Rewrite Systems and the Lambda-Calculus.
36th ICALP, pages 163–174, 2009

Exponential Path Order $>_{\mathsf{epo}\star}$

- constraints of $\mathcal{N}$ imposed on lexicographic path order $>_{\mathsf{epo}\star} \subseteq >_{\mathsf{lpo}}$

  $\Longrightarrow$ sufficiently strong for **Completeness**

- induces exponential bound on innermost runtime complexity $\mathsf{rc}^{\mathsf{i}}_{\mathcal{R}}$

  $\Longrightarrow$ implies **Soundness**

Ugo Dal Lago and Simone Martini
On Constructor Rewrite Systems and the Lambda-Calculus.
36th ICALP, pages 163–174, 2009

Martin Avanzini and Georg Moser
Closing the Gap Between Runtime Complexity and Polytime Computability.
RTA'10, pages 33–48, 2010

The class $\mathcal{N}$

Syntactic, Recursion-theoretic Characterisation of FEXP

The class $\mathcal{N}$

is the smallest class . . .

1. containing certain initial function — projections, successors, . . .
2. closed under safe nested recursion on notation
3. closed under weak safe composition

Safe Recursion on Notation

- syntactical restriction of primitive recursion scheme

$$f(\underbrace{x_1, \ldots, x_k}_{\text{normal}}; \underbrace{y_1, \ldots, y_l}_{\text{safe}})$$

Safe Recursion on Notation

- syntactical restriction of primitive recursion scheme

$$f(\underbrace{x_1, \ldots, x_k}_{\text{normal}}; \underbrace{y_1, \ldots, y_l}_{\text{safe}})$$

- separates recursion parameters from recursively computed results

$$\begin{aligned} f(\epsilon, \vec{x}; \vec{y}) &= g(\vec{x}; \vec{y}) \\ f(\mathbf{zi}, \vec{x}; \vec{y}) &= h_i(z, \vec{x}; \vec{y}, \mathbf{f}(\mathbf{z}, \vec{\mathbf{x}}; \vec{\mathbf{y}})) \qquad (i \in \{0, 1\}) \end{aligned}$$

Safe Recursion on Notation

- syntactical restriction of primitive recursion scheme

$$f(\underbrace{x_1, \ldots, x_k}_{\text{normal}}; \underbrace{y_1, \ldots, y_l}_{\text{safe}})$$

- separates recursion parameters from recursively computed results

$$f(\epsilon, \vec{x}; \vec{y}) = g(\vec{x}; \vec{y})$$
$$f(\mathbf{zi}, \vec{x}; \vec{y}) = h_i(z, \vec{x}; \vec{y}, \mathbf{f(z, \vec{x}; \vec{y})}) \qquad (i \in \{0, 1\})$$

Safe Recursion on Notation

- syntactical restriction of primitive recursion scheme

$$f(\underbrace{x_1, \ldots, x_k}_{\text{normal}}; \underbrace{y_1, \ldots, y_l}_{\text{safe}})$$

- separates recursion parameters from recursively computed results

$$
\begin{aligned}
f(\epsilon, \vec{x}; \vec{y}) &= g(\vec{x}; \vec{y}) \\
f(\mathbf{zi}, \vec{x}; \vec{y}) &= h_i(z, \vec{x}; \vec{y}, \mathbf{f(z, \vec{x}; \vec{y})}) \qquad (i \in \{0,1\})
\end{aligned}
$$

$$
\begin{aligned}
\text{where } h_i(\epsilon, \vec{x}; \vec{y}, \mathbf{r}) &= r_i(\vec{x}; \vec{y}, \mathbf{r}) \\
h_i(\mathbf{zi}, \vec{x}; \vec{y}, \mathbf{r}) &= s_{i,j}(z, \vec{x}; \vec{y}, h_i(z, \vec{x}; \vec{y}, \mathbf{r}))
\end{aligned}
$$

no recursion on recursively computed result

Safe Nested Recursion on Notation

extends safe recursion on notation with . . .

1. nesting of recursive function calls — nested recursion

$$f(\epsilon; y) = g(; y)$$
$$f(xi; y) = r_i(x; y, f(x; s_i(x; y, f(x; \ldots))))$$

Safe Nested Recursion on Notation

extends safe recursion on notation with . . .

1. nesting of recursive function calls — nested recursion

$$f(\epsilon; y) = g(; y)$$
$$f(xi; y) = r_i(x; y, f(x; s_i(x; y, f(x; \ldots))))$$

2. simultaneous recursion on all normal arguments — multiple recursion

$$f(\epsilon, \epsilon; z) = g(; z)$$
$$f(xi, \epsilon; z) = r_{i,\epsilon}(x, \epsilon; z, f(x, \epsilon; s_{i,\epsilon}(x, \epsilon; f(x, \epsilon; z))))$$
$$f(\epsilon, yj; z) = r_{\epsilon,j}(\epsilon, y; z, f(\epsilon, y; s_{\epsilon,j}(\epsilon, y; f(\epsilon, y; z))))$$
$$f(xi, yj; z) = r_{i,j}(x, y; z, f(xi, y; s_{i,j}(x, y; f(x, yj; z))))$$

- case analysis on least significant “bits” of recursion parameters

Safe Nested Recursion on Notation

extends safe recursion on notation with ...

1. nesting of recursive function calls — nested recursion

$$
\begin{aligned}
f(\epsilon; y) &= g(; y) \\
f(xi; y) &= r_i(x; y, f(x; s_i(x; y, f(x; \ldots))))
\end{aligned}
$$

2. simultaneous recursion on all normal arguments — multiple recursion

$$
\begin{aligned}
f(\epsilon, \epsilon; z) &= g(; z) \\
f(xi, \epsilon; z) &= r_{i,\epsilon}(x, \epsilon; z, f(x, \epsilon; s_{i,\epsilon}(x, \epsilon; f(x, \epsilon; z)))) \\
f(\epsilon, yj; z) &= r_{\epsilon,j}(\epsilon, y; z, f(\epsilon, y; s_{\epsilon,j}(\epsilon, y; f(\epsilon, y; z)))) \\
f(xi, yj; z) &= r_{i,j}(x, y; z, f(xi, y; s_{i,j}(x, y; f(x, yj; z))))
\end{aligned}
$$

- case analysis on least significant "bits" of recursion parameters
- lexicographic decreasing recursion parameters

Safe Composition

Requirements

1. composition maintains separation of safe and normal arguments

Safe Composition

Requirements

1. composition maintains separation of safe and normal arguments

Safe Composition

employed in $\mathcal{B}$

$$f(\vec{x}; \vec{y}) = g(\vec{r}(\vec{x};); \vec{s}(\vec{x}; \vec{y}))$$

Safe Composition

Requirements

1. composition maintains separation of safe and normal arguments
2. reflects that FEXP is not closed under composition

Safe Composition

employed in $\mathcal{B}$

$$f(\vec{x};\vec{y}) = g(\vec{r}(\vec{x};);\vec{s}(\vec{x};\vec{y}))$$

Weak Safe Composition

employed in $\mathcal{N}$

$$f(\vec{x};\vec{y}) = g(x_{i_1},\dots,x_{i_k};\vec{s}(\vec{x};\vec{y})) \qquad \{x_{i_1},\dots,x_{i_k}\} \subseteq \{\vec{x}\}$$

The class $\mathcal{N}$

is the smallest class . . .

1. containing certain initial function projections, successors, . . .
2. closed under safe nested recursion on notation
3. closed under weak safe composition

The class $\mathcal{N}$

is the smallest class ...

1. containing certain initial function — projections, successors, ...
2. closed under safe nested recursion on notation
3. closed under weak safe composition

Theorem

$$\mathcal{N} = \mathrm{FEXP}$$

Toshiyasu Arai and Naohi Eguchi
A new Function Algebra of EXPTIME Functions by Safe Nested Recursion.
TCL, pages 130–146, 2008

Exponential Path Order $>_{\mathsf{epo}\star}$

A Path Order based on $\mathcal{N}$

Exponential Path Order $>_{\mathsf{epo}\star}$

- induced by precedence $>$ *and* safe mapping $\mathsf{safe} : \mathcal{F} \to 2^{\mathbb{N}}$

Exponential Path Order $>_{\mathsf{epo}\star}$

- induced by precedence $>$ *and* safe mapping $\mathsf{safe} : \mathcal{F} \to 2^{\mathbb{N}}$

```
> tct -s "epo*" fib.trs
  YES(?,EXPO)

  We consider the following Problem:
   Strict Trs:
     { fib(s(s(x)), y) -> fib(s(x), fib(x, y))
     , fib(s(0()), y)  -> s(y)
     , fib(0(), y)     -> s(y) }
   StartTerms: basic terms
   Strategy:   innermost

  The system is compatible with 'epo*' induced by
    Precedence:   fib > s,0
    Safe Mapping: safe(fib) = {2}, safe(s) = {1}
```

Exponential Path Order $>_{\mathsf{epo}\star}$

- induced by precedence $>$ *and* safe mapping $\mathsf{safe} \colon \mathcal{F} \to 2^{\mathbb{N}}$

Definition

precedence $>$ and safe mapping safe are admissible if

1. constructors are minimal $f > g \Rightarrow f \notin \mathcal{C}$
2. all argument positions of constructors are safe $f \in \mathcal{C} \Rightarrow \mathsf{safe}(f) = \{1, \ldots, \mathsf{ar}(f)\}$

Exponential Path Order $>_{\mathsf{epo}\star}$

- induced by precedence $>$ *and* safe mapping $\mathsf{safe} : \mathcal{F} \to 2^{\mathbb{N}}$

Definition

precedence $>$ and safe mapping safe are admissible if

1. constructors are minimal $f > g \Rightarrow f \notin \mathcal{C}$
2. all argument positions of constructors are safe $f \in \mathcal{C} \Rightarrow \mathsf{safe}(f) = \{1, \dots, \mathsf{ar}(f)\}$

Notation

we suppose $\mathsf{safe}(f) = \{l + 1, \dots, l + m\}$, we write

$$f(s_1, \dots, s_l; s_{l+1}, \dots, s_{l+m})$$

Exponential Path Order $>_{\mathsf{epo}\star}$

Preliminary Definition

Let $s = f(s_1, \dots, s_l; s_{l+1}, \dots, s_{l+m})$, let $>$ and safe be admissible.

Exponential Path Order $>_{\text{epo}\star}$

Preliminary Definition

Let $s = f(s_1, \ldots, s_l; s_{l+1}, \ldots, s_{l+m})$, let $>$ and safe be admissible.

ST

$$\frac{s_i \geqslant_{\text{epo}\star} t}{s >_{\text{epo}\star} t}$$

WSC

$$\frac{\text{“}t_i \text{ are normal arguments of } s\text{”} \qquad s >_{\text{epo}\star} t_{k+1} \cdots s >_{\text{epo}\star} t_{k+n}}{s >_{\text{epo}\star} g(t_1, \ldots, t_k; t_{k+1}, \ldots, t_{k+n})}\ f > g$$

SNRN

$$\frac{\langle s_1, \ldots, s_l \rangle >_{\text{lex}'} \langle t_1, \ldots, t_l \rangle \qquad s >_{\text{epo}\star} t_{l+1} \ \cdots \ s >_{\text{epo}\star} t_m}{s >_{\text{epo}\star} f(t_1, \ldots, t_l; t_{l+1}, \ldots, t_{l+m})}$$

Exponential Path Order $>_{\mathsf{epo}\star}$

Preliminary Definition

Let $s = f(s_1, \dots, s_l; s_{l+1}, \dots, s_{l+m})$, let $>$ and safe be admissible.

ST

$$\frac{s_i \geqslant_{\mathsf{epo}\star} t}{s >_{\mathsf{epo}\star} t}$$

WSC

$$\frac{\text{“}t_i \text{ are normal arguments of } s\text{”} \qquad s >_{\mathsf{epo}\star} t_{k+1} \cdots s >_{\mathsf{epo}\star} t_{k+n}}{s >_{\mathsf{epo}\star} g(t_1, \dots, t_k; t_{k+1}, \dots, t_{k+n})} \; f > g$$

SNRN

$$\frac{\langle s_1, \dots, s_l \rangle >_{\mathsf{lex}'} \langle t_1, \dots, t_l \rangle \qquad s >_{\mathsf{epo}\star} t_{l+1} \;\cdots\; s >_{\mathsf{epo}\star} t_m}{s >_{\mathsf{epo}\star} f(t_1, \dots, t_l; t_{l+1}, \dots, t_{l+m})}$$

Recall Weak Safe Composition ...

$$f(\vec{x}; \vec{y}) = g(x_{i_1}, \dots, x_{i_k}; \vec{s}(\vec{x}; \vec{y})) \qquad \{x_{i_1}, \dots, x_{i_k}\} \subseteq \{\vec{x}\}$$

Exponential Path Order $>_{\mathsf{epo}\star}$

Preliminary Definition

Let $s = f(s_1, \dots, s_l; s_{l+1}, \dots, s_{l+m})$, let $>$ and safe be admissible.

ST

$$\frac{s_i \geqslant_{\mathsf{epo}\star} t}{s >_{\mathsf{epo}\star} t}$$

WSC

$$\frac{\text{"}t_i \text{ are normal arguments of } s\text{"} \qquad s >_{\mathsf{epo}\star} t_{k+1} \cdots s >_{\mathsf{epo}\star} t_{k+n}}{s >_{\mathsf{epo}\star} g(t_1, \dots, t_k; t_{k+1}, \dots, t_{k+n})} \; f > g$$

SNRN

$$\frac{\langle s_1, \dots, s_l \rangle >_{\mathsf{lex}'} \langle t_1, \dots, t_l \rangle \qquad s >_{\mathsf{epo}\star} t_{l+1} \; \cdots \; s >_{\mathsf{epo}\star} t_m}{s >_{\mathsf{epo}\star} f(t_1, \dots, t_l; t_{l+1}, \dots, t_{l+m})}$$

Recall Safe Nested Recursion on Notation ...

$$f(xi, yj; \dots) = r(\dots; \dots, f(xi, y; \vec{s}(\dots; \dots, f(x, yj; \dots))))$$

Exponential Path Order $>_{\text{epo}\star}$

Preliminary Definition

Let $s = f(s_1, \dots, s_l; s_{l+1}, \dots, s_{l+m})$, let $>$ and safe be admissible.

ST
$$\frac{s_i \geqslant_{\text{epo}\star} t}{s >_{\text{epo}\star} t}$$

WSC
$$\frac{\text{“}t_i \text{ are normal arguments of } s\text{”} \qquad s >_{\text{epo}\star} t_{k+1} \cdots s >_{\text{epo}\star} t_{k+n}}{s >_{\text{epo}\star} g(t_1, \dots, t_k; t_{k+1}, \dots, t_{k+n})}\ f > g$$

SNRN
$$\frac{\langle s_1, \dots, s_l \rangle >_{\text{lex}'} \langle t_1, \dots, t_l \rangle \qquad s >_{\text{epo}\star} t_{l+1} \ \cdots \ s >_{\text{epo}\star} t_m}{s >_{\text{epo}\star} f(t_1, \dots, t_l; t_{l+1}, \dots, t_{l+m})}$$

Auxiliary Order $\sqsupset_{\mathsf{epo\star}}$

Order for $\mathcal{V}$al

$$\mathbf{ST_n}\ \frac{s_i \sqsupseteq_{\mathsf{epo\star}} t}{f(s_1, \dots, s_l; s_{l+1}, \dots, s_{l+m}) \sqsupset_{\mathsf{epo\star}} t}\ \text{if } f \in \mathcal{D} \text{ then } i \in \{1, \dots, l\}$$

Note

1. $\sqsupset_{\mathsf{epo\star}} = \rhd$ on $\mathcal{V}$al

Auxiliary Order $\sqsupset_{\mathsf{epo}\star}$

Order for $\mathcal{V}$al

$$\mathbf{ST_n}\ \frac{s_i \sqsupseteq_{\mathsf{epo}\star} t}{f(s_1, \ldots, s_l; s_{l+1}, \ldots, s_{l+m}) \sqsupset_{\mathsf{epo}\star} t}\ \text{if } f \in \mathcal{D} \text{ then } i \in \{1, \ldots, l\}$$

Note

1. $\sqsupset_{\mathsf{epo}\star} = \rhd$ on $\mathcal{V}$al
2. $\sqsupset_{\mathsf{epo}\star} \subsetneq \rhd$
   - if $f \in \mathcal{D}$, $\mathsf{safe}(f) = \{2\}$ then $f(x; z) \sqsupset_{\mathsf{epo}\star} x$ but $f(x; z) \not\sqsupset_{\mathsf{epo}\star} z$

Exponential Path Order $>_{\mathsf{epo}\star}$

Preliminary Definition

Let $s = f(s_1, \dots, s_l; s_{l+1}, \dots, s_{l+m})$, let $>$ and safe be admissible.

ST
$$\frac{s_i \geqslant_{\mathsf{epo}\star} t}{s >_{\mathsf{epo}\star} t}$$

WSC
$$\frac{\text{“}t_i \text{ are normal arguments of } s\text{”} \qquad s >_{\mathsf{epo}\star} t_{k+1} \cdots s >_{\mathsf{epo}\star} t_{k+n}}{s >_{\mathsf{epo}\star} g(t_1, \dots, t_k; t_{k+1}, \dots, t_{k+n})} \; f > g$$

SNRN
$$\frac{\langle s_1, \dots, s_l \rangle >_{\mathsf{lex}'} \langle t_1, \dots, t_l \rangle \qquad s >_{\mathsf{epo}\star} t_{l+1} \;\cdots\; s >_{\mathsf{epo}\star} t_m}{s >_{\mathsf{epo}\star} f(t_1, \dots, t_l; t_{l+1}, \dots, t_{l+m})}$$

Exponential Path Order $>_{\mathsf{epo}\star}$

Preliminary Definition

Let $s = f(s_1, \dots, s_l; s_{l+1}, \dots, s_{l+m})$, let $>$ and safe be admissible.

ST $$\frac{s_i \geqslant_{\mathsf{epo}\star} t}{s >_{\mathsf{epo}\star} t}$$

WSC $$\frac{s \sqsupset_{\mathsf{epo}\star} t_1 \ \cdots \ s \sqsupset_{\mathsf{epo}\star} t_k \qquad s >_{\mathsf{epo}\star} t_{k+1} \cdots s >_{\mathsf{epo}\star} t_{k+n}}{s >_{\mathsf{epo}\star} g(t_1, \dots, t_k; t_{k+1}, \dots, t_{k+n})} \ f > g$$

SNRN $$\frac{\langle s_1, \dots, s_l \rangle >_{\mathsf{lex}'} \langle t_1, \dots, t_l \rangle \qquad s >_{\mathsf{epo}\star} t_{l+1} \ \cdots \ s >_{\mathsf{epo}\star} t_m}{s >_{\mathsf{epo}\star} f(t_1, \dots, t_l; t_{l+1}, \dots, t_{l+m})}$$

Exponential Path Order $>_{\mathsf{epo}\star}$

Preliminary Definition

Let $s = f(s_1, \dots, s_l; s_{l+1}, \dots, s_{l+m})$, let $>$ and safe be admissible.

ST $$\frac{s_i \geqslant_{\mathsf{epo}\star} t}{s >_{\mathsf{epo}\star} t}$$

WSC $$\frac{s \sqsupset_{\mathsf{epo}\star} t_1 \ \cdots \ s \sqsupset_{\mathsf{epo}\star} t_k \qquad s >_{\mathsf{epo}\star} t_{k+1} \cdots s >_{\mathsf{epo}\star} t_{k+n}}{s >_{\mathsf{epo}\star} g(t_1, \dots, t_k; t_{k+1}, \dots, t_{k+n})} \ f > g$$

SNRN $$\frac{\langle s_1, \dots, s_l \rangle >_{\mathsf{lex}'} \langle t_1, \dots, t_l \rangle \qquad s >_{\mathsf{epo}\star} t_{l+1} \ \cdots \ s >_{\mathsf{epo}\star} t_m}{s >_{\mathsf{epo}\star} f(t_1, \dots, t_l; t_{l+1}, \dots, t_{l+m})}$$

Exponential Path Order $>_{\mathsf{epo}\star}$

Preliminary Definition

Let $s = f(s_1, \ldots, s_l; s_{l+1}, \ldots, s_{l+m})$, let $>$ and safe be admissible.

ST
$$\frac{s_i \geqslant_{\mathsf{epo}\star} t}{s >_{\mathsf{epo}\star} t}$$

WSC
$$\frac{s \sqsupset_{\mathsf{epo}\star} t_1 \quad \cdots \quad s \sqsupset_{\mathsf{epo}\star} t_k \qquad s >_{\mathsf{epo}\star} t_{k+1} \cdots s >_{\mathsf{epo}\star} t_{k+n}}{s >_{\mathsf{epo}\star} g(t_1, \ldots, t_k; t_{k+1}, \ldots, t_{k+n})} \; f > g$$

SNRN
$$\frac{(\dagger) \qquad s >_{\mathsf{epo}\star} t_{l+1} \quad \cdots \quad s >_{\mathsf{epo}\star} t_m}{s >_{\mathsf{epo}\star} f(t_1, \ldots, t_l; t_{l+1}, \ldots, t_{l+m})}$$

$(\dagger)$

1. $s_1 = t_1 \quad \cdots \quad s_{i-1} = t_{i-1}$,
2. $s_i \sqsupset_{\mathsf{epo}\star} t_i$, and
3. $s \sqsupset_{\mathsf{epo}\star} t_{i+1} \quad \cdots \quad s \sqsupset_{\mathsf{epo}\star} t_l$.

Exponential Path Order $>_{\mathsf{epo}\star}$

Let $s = f(s_1, \dots, s_l; s_{l+1}, \dots, s_{l+m})$, let $>$ and safe be admissible.

ST $$\frac{s_i \geqslant_{\mathsf{epo}\star} t}{s >_{\mathsf{epo}\star} t}$$

WSC $$\frac{s \sqsupset_{\mathsf{epo}\star} t_1 \quad \cdots \quad s \sqsupset_{\mathsf{epo}\star} t_k \qquad s >_{\mathsf{epo}\star} t_{k+1} \cdots s >_{\mathsf{epo}\star} t_{k+n}}{s >_{\mathsf{epo}\star} g(t_1, \dots, t_k; t_{k+1}, \dots, t_{k+n})} \; f > g$$

SNRN $$\frac{(\dagger) \qquad s >_{\mathsf{epo}\star} t_{l+1} \quad \cdots \quad s >_{\mathsf{epo}\star} t_m}{s >_{\mathsf{epo}\star} f(t_1, \dots, t_l; t_{l+1}, \dots, t_{l+m})}$$

$(\dagger)$

1. $s_1 = t_1 \quad \cdots \quad s_{i-1} = t_{i-1}$,
2. $s_i \sqsupset_{\mathsf{epo}\star} t_i$, and
3. $s \sqsupset_{\mathsf{epo}\star} t_{i+1} \quad \cdots \quad s \sqsupset_{\mathsf{epo}\star} t_l$.

The Good ...

the exponential path order $\mathsf{EPO}^{\star}$ is ...

- a restriction of LPO that induces exponentially bounded $\mathsf{rc}^{\mathsf{i}}_{\mathcal{R}}$
- sound *and* complete for FEXP, implemented in our tool $\mathsf{T_CT}$
  http://cl-informatik.uibk.ac.at/software/tct

The Good ...

the exponential path order EPO* is ...

- a restriction of LPO that induces exponentially bounded $\mathsf{rc}^{\mathsf{i}}_{\mathcal{R}}$
- sound *and* complete for FEXP, implemented in our tool $\mathsf{T_CT}$
  `http://cl-informatik.uibk.ac.at/software/tct`

The Bad ...

- rules out some natural definitions:

$$\mathsf{d}(0) \to 0 \qquad \mathsf{e}(0) \to \mathsf{s}(0)$$

$$\mathsf{d}(\mathsf{s}(x)) \to \mathsf{s}(\mathsf{s}(\mathsf{d}(x))) \qquad \mathsf{e}(\mathsf{s}(x)) \to \mathsf{d}(\mathsf{e}(x))$$

The Good ...

the exponential path order EPO* is ...

- a restriction of LPO that induces exponentially bounded $\mathsf{rc}^{\mathsf{i}}_{\mathcal{R}}$
- sound *and* complete for FEXP, implemented in our tool T$_\mathsf{C}$T
  http://cl-informatik.uibk.ac.at/software/tct

The Bad ...

- rules out some natural definitions:

$$\mathsf{d}(0) \to 0 \qquad \mathsf{e}(0) \to \mathsf{s}(0)$$

$$\mathsf{d}(\mathsf{s}(x)) \to \mathsf{s}(\mathsf{s}(\mathsf{d}(x))) \qquad \mathsf{e}(\mathsf{s}(x)) \to \mathsf{d}(\mathsf{e}(x))$$

The Ugly ...

- $>_{\mathsf{pop}\star} \not\subseteq >_{\mathsf{epo}\star}$