Type Introduction for Runtime Complexity Analysis

Martin Avanzini[1]
(Joint work with Bertram Felgenhauer[2])

[1] Dipartimento di Informatica
Scienza e Ingegneria
Università degli Studi di Bologna

[2] Institute for Computer Science
University of Innsbruck

Workshop on Termination, October 27, 2014

Runtime Complexity of TRSs

- derivation height $\mathrm{dh}(t, \to) := \max\{n \mid \exists\, s. t \to^{n} s\}$

1. derivational-complexity function *[Hofbauer & Lautemann, 1989]*

$$\mathrm{dc}_{\mathcal{R}}(n) = \max\{\, \mathrm{dh}(f(\vec{s}), \to_{\mathcal{R}}) \mid f(\vec{s}) \text{ of size up to } n \,\}$$

2. runtime-complexity function *[Hirokawa & Moser, 2008]*

$$\mathrm{rc}_{\mathcal{R}}(n) = \max\{\, \mathrm{dh}(f(\vec{v}), \to_{\mathcal{R}}) \mid \underbrace{f(\vec{v}) \text{ and } \vec{v} \text{ are values}}_{\text{basic term}} \text{ of size up to } n \,\}$$

Runtime Complexity of TRSs

- derivation height $\mathrm{dh}(t, \to) := \max\{n \mid \exists\, s. t \to^{n} s\}$

1. derivational-complexity function *[Hofbauer & Lautemann, 1989]*

$$\mathrm{dc}_{\mathcal{R}}(n) = \max\{\, \mathrm{dh}(f(\vec{s}), \to_{\mathcal{R}}) \mid f(\vec{s}) \text{ of size up to } n \,\}$$

2. runtime-complexity function *[Hirokawa & Moser, 2008]*

$$\mathrm{rc}_{\mathcal{R}}(n) = \max\{\, \mathrm{dh}(f(\vec{v}), \to_{\mathcal{R}}) \mid \underbrace{f(\vec{v}) \text{ and } \vec{v} \text{ are values}}_{\text{basic term}} \text{ of size up to } n \,\}$$

Runtime Complexity of TRSs

- derivation height $\mathrm{dh}(t, \to) := \max\{n \mid \exists\, s. t \to^{n} s\}$

1. derivational-complexity function *[Hofbauer & Lautemann, 1989]*

$$\mathrm{dc}_{\mathcal{R}}(n) = \max\{\, \mathrm{dh}(f(\vec{s}), \to_{\mathcal{R}}) \mid f(\vec{s}) \text{ of size up to } n \,\}$$

2. runtime-complexity function *[Hirokawa & Moser, 2008]*

$$\mathrm{rc}_{\mathcal{R}}(n) = \max\{\, \mathrm{dh}(f(\vec{v}),\ \to_{\mathcal{R}}) \mid \underbrace{f(\vec{v}) \text{ and } \vec{v} \text{ are values}}_{\text{basic term}} \text{ of size up to } n \,\}$$

Runtime Complexity of TRSs

- derivation height $\mathrm{dh}(t, \rightarrow) := \max\{n \mid \exists\, s. t \rightarrow^{n} s\}$

1. derivational-complexity function *[Hofbauer & Lautemann, 1989]*

$$\mathrm{dc}_{\mathcal{R}}(n) = \max\{\, \mathrm{dh}(f(\vec{s}), \rightarrow_{\mathcal{R}}) \mid f(\vec{s}) \text{ of size up to } n \,\}$$

2. runtime-complexity function *[Hirokawa & Moser, 2008]*

$$\mathrm{rc}_{\mathcal{R}}(n) = \max\{\, \mathrm{dh}(f(\vec{v}), \rightarrow_{\mathcal{R}}) \mid \underbrace{f(\vec{v}) \text{ and } \vec{v} \text{ are values}}_{\text{basic term}} \text{ of size up to } n \,\}$$

Runtime Complexity of TRSs

- derivation height $\mathrm{dh}(t, \to) := \max\{n \mid \exists\, s. t \to^n s\}$

1. derivational-complexity function *[Hofbauer & Lautemann, 1989]*

$$\mathrm{dc}_{\mathcal{R}}(n) = \max\{\, \mathrm{dh}(f(\vec{s}), \to_{\mathcal{R}}) \mid f(\vec{s}) \text{ of size up to } n \,\}$$

2. innermost runtime-complexity function *[Hirokawa & Moser, 2008]*

$$\mathrm{rc}^{\mathrm{i}}_{\mathcal{R}}(n) = \max\{\, \mathrm{dh}(f(\vec{v}), \xrightarrow{\mathrm{i}}_{\mathcal{R}}) \mid \underbrace{f(\vec{v}) \text{ and } \vec{v} \text{ are values}}_{\text{basic term}} \text{ of size up to } n \,\}$$

Automated Complexity Analysis of TRSs

- part of annual **termination competition** since 2008

- **Tools**
  - AProVE `http://aprove.informatik.rwth-aachen.de`
  - Cat `http://cl-informatik.uibk.ac.at/software/cat`
  - Matchbox/Poly `http://dfa.imn.htwk-leipzig.de/matchbox`
  - TCT `http://cl-informatik.uibk.ac.at/software/tct`

  - RAML prototype `http://raml.tcs.ifi.lmu.de/prototype`
    - automated, amortised cost analysis of functional programs
    - fast and powerful
    - theory recently reformulated in context of many-sorted TRSs

[Hofmann and Moser, 2014]

Automated Complexity Analysis of TRSs

- part of annual **termination competition** since 2008
- **Tools**
  - APROVE http://aprove.informatik.rwth-aachen.de
  - Cat http://cl-informatik.uibk.ac.at/software/cat
  - Matchbox/Poly http://dfa.imn.htwk-leipzig.de/matchbox
  - TCT http://cl-informatik.uibk.ac.at/software/tct
  - RAML prototype http://raml.tcs.ifi.lmu.de/prototype
    - automated, amortised cost analysis of functional programs
    - fast and powerful
    - theory recently reformulated in context of many-sorted TRSs

[Hofmann and Moser, 2014]

Automated Complexity Analysis of TRSs

- part of annual **termination competition** since 2008
- **Tools**
  - AProVE `http://aprove.informatik.rwth-aachen.de`
  - Cat `http://cl-informatik.uibk.ac.at/software/cat`
  - Matchbox/Poly `http://dfa.imn.htwk-leipzig.de/matchbox`
  - TCT `http://cl-informatik.uibk.ac.at/software/tct`
  - RAML prototype `http://raml.tcs.ifi.lmu.de/prototype`
    - automated, amortised cost analysis of functional programs
    - fast and powerful
    - theory recently reformulated in context of many-sorted TRSs

[Hofmann and Moser, 2014]

Persistency

Definition

property P is **persistent** if for all many-sorted TRS $\mathcal{R}$

$$P(\mathcal{R}) \iff P(\Theta(\mathcal{R}))$$

- $\Theta(\mathcal{R})$ denotes un(i)sorted TRS underlying $\mathcal{R}$

1. confluence ✔ *[Aoto and Toyama, 1997]*
2. termination ✗
   - non-collapsing or non-duplicating TRSs ✔ *[Zantema,1994]*
   - all variables of same sort ✔ *[Aoto, 1998]*
   - non-overlapping or locally confluent overlay TRSs ✔ *[Iwama, 2003; 2004]*
3. innermost termination ✔ *[Iwama, 2004]*

Persistency

Definition

property P is **persistent** if for all many-sorted TRS $\mathcal{R}$

$$P(\mathcal{R}) \quad\Longleftrightarrow\quad P(\Theta(\mathcal{R}))$$

- $\Theta(\mathcal{R})$ denotes un(i)sorted TRS underlying $\mathcal{R}$

1. confluence ✔ *[Aoto and Toyama, 1997]*
2. termination ✗
   - non-collapsing or non-duplicating TRSs ✔ *[Zantema,1994]*
   - all variables of same sort ✔ *[Aoto, 1998]*
   - non-overlapping or locally confluent overlay TRSs ✔ *[Iwama, 2003; 2004]*
3. innermost termination ✔ *[Iwama, 2004]*

Persistency

Definition

property P is **persistent** if for all many-sorted TRS $\mathcal{R}$

$$P(\mathcal{R}) \iff P(\Theta(\mathcal{R}))$$

- $\Theta(\mathcal{R})$ denotes un(i)sorted TRS underlying $\mathcal{R}$

1. confluence ✔ *[Aoto and Toyama, 1997]*
2. termination ✗
   - non-collapsing or non-duplicating TRSs ✔ *[Zantema,1994]*
   - all variables of same sort ✔ *[Aoto, 1998]*
   - non-overlapping or locally confluent overlay TRSs ✔ *[Iwama, 2003; 2004]*
   - ...
3. innermost termination ✔ *[Iwama, 2004]*

Persistency

Definition

property P is **persistent** if for all many-sorted TRS $\mathcal{R}$

$$P(\mathcal{R}) \iff P(\Theta(\mathcal{R}))$$

- $\Theta(\mathcal{R})$ denotes un(i)sorted TRS underlying $\mathcal{R}$

1. confluence	✔	*[Aoto and Toyama, 1997]*
2. termination	✗	
– non-collapsing or non-duplicating TRSs	✔	*[Zantema,1994]*
– all variables of same sort	✔	*[Aoto, 1998]*
– non-overlapping or locally confluent overlay TRSs	✔	*[Iwama, 2003; 2004]*
– ...		
3. innermost termination	✔	*[Iwama, 2004]*

Persistency

Definition

property P is **persistent** if for all many-sorted TRS $\mathcal{R}$

$$P(\mathcal{R}) \iff P(\Theta(\mathcal{R}))$$

- $\Theta(\mathcal{R})$ denotes un(i)sorted TRS underlying $\mathcal{R}$

1. confluence ✔ *[Aoto and Toyama, 1997]*
2. termination ✗
   - non-collapsing or non-duplicating TRSs ✔ *[Zantema,1994]*
   - all variables of same sort ✔ *[Aoto, 1998]*
   - non-overlapping or locally confluent overlay TRSs ✔ *[Iwama, 2003; 2004]*
   - ...
3. innermost termination ✔ *[Iwama, 2004]*

Sorted Rewriting

notations (i)

1. finite set of sorts $\mathcal{S}$

2. for each sort $\alpha \in \mathcal{S}$, set of sorted variables $\mathcal{V}_\alpha = \{x^\alpha, y^\alpha, \ldots\}$

3. function symbols are equipped with sort declaration

$$f :: (\alpha_1, \ldots, \alpha_k) \to \alpha$$

4. term is well-sorted according to rules

$$\frac{x \in \mathcal{V}_\alpha}{x :: \alpha}\ (Var) \qquad \frac{t_1 :: \alpha_1 \quad \ldots \quad t_k :: \alpha_k \quad f :: (\alpha_1, \ldots, \alpha_k) \to \alpha}{f(t_1, \ldots, t_k) :: \alpha}\ (Fun)$$

Sorted Rewriting

notations (i)

1. finite set of sorts $\mathcal{S}$

2. for each sort $\alpha \in \mathcal{S}$, set of sorted variables $\mathcal{V}_\alpha = \{x^\alpha, y^\alpha, \dots\}$

3. function symbols are equipped with sort declaration

$$f :: (\alpha_1, \dots, \alpha_k) \to \alpha$$

4. term is well-sorted according to rules

$$\frac{x \in \mathcal{V}_\alpha}{x :: \alpha}\ (Var) \qquad \frac{t_1 :: \alpha_1 \quad \dots \quad t_k :: \alpha_k \quad f :: (\alpha_1, \dots, \alpha_k) \to \alpha}{f(t_1, \dots, t_k) :: \alpha}\ (Fun)$$

notations (ii)

5. **many-sorted** TRS $\mathcal{R}$ satisfies for each $l \to r \in \mathcal{R}$:

$$l :: \alpha \text{ and } r :: \alpha \text{ for some sort } \alpha \in \mathcal{S}$$

6. **rewrite relation** $\to_{\mathcal{R}}$ defined on well-sorted terms only

Sorted Rewriting

notations (ii)

5. **many-sorted** TRS $\mathcal{R}$ satisfies for each $l \to r \in \mathcal{R}$:

$$l :: \alpha \text{ and } r :: \alpha \text{ for some sort } \alpha \in \mathcal{S}$$

6. **rewrite relation** $\to_{\mathcal{R}}$ defined on well-sorted terms only

Main Result

Theorem

The (innermost) runtime-complexity functions of $\mathcal{R}$ and $\Theta(\mathcal{R})$ coincide.

proof outline.

- $\mathrm{rc}^{(i)}_{\Theta(\mathcal{R})}(n) \leqslant \mathrm{rc}^{(i)}_{\mathcal{R}}(n)$:

  every reduction

  $$\mathsf{f}(v_1, \dots, v_n) \xrightarrow{(i)}_{\Theta(\mathcal{R})} t_1 \xrightarrow{(i)}_{\Theta(\mathcal{R})} t_2 \xrightarrow{(i)}_{\Theta(\mathcal{R})} \cdots$$

  simulated stepwise by

  $$\mathsf{f}(v_1', \dots, v_n') \xrightarrow{(i)}_{\mathcal{R}} t_1' \xrightarrow{(i)}_{\mathcal{R}} t_2' \xrightarrow{(i)}_{\mathcal{R}} \cdots$$

  where $|v_i'| \leqslant |v_i|$.

- $\mathrm{rc}^{(i)}_{\Theta(\mathcal{R})}(n) \geqslant \mathrm{rc}^{(i)}_{\mathcal{R}}(n)$: trivial

□

Main Result

Theorem

The (innermost) runtime-complexity functions of $\mathcal{R}$ and $\Theta(\mathcal{R})$ coincide.

proof outline.

- $\mathsf{rc}^{(i)}_{\Theta(\mathcal{R})}(n) \leqslant \mathsf{rc}^{(i)}_{\mathcal{R}}(n)$:

  every reduction

  $$\mathtt{f}(v_1, \ldots, v_n) \xrightarrow{(i)}_{\Theta(\mathcal{R})} t_1 \xrightarrow{(i)}_{\Theta(\mathcal{R})} t_2 \xrightarrow{(i)}_{\Theta(\mathcal{R})} \cdots$$

  simulated stepwise by

  $$\mathtt{f}(v'_1, \ldots, v'_n) \xrightarrow{(i)}_{\mathcal{R}} t'_1 \xrightarrow{(i)}_{\mathcal{R}} t'_2 \xrightarrow{(i)}_{\mathcal{R}} \cdots$$

  where $|v'_i| \leqslant |v_i|$.

- $\mathsf{rc}^{(i)}_{\Theta(\mathcal{R})}(n) \geqslant \mathsf{rc}^{(i)}_{\mathcal{R}}(n)$: trivial

$\square$

Persistency

Toyama's example

Example

TRS $\mathcal{R}_\mathsf{T}$ consists of

$$\mathtt{f}(0,1,x) \to \mathtt{f}(x,x,x) \qquad \mathrm{g}(y,z) \to y \qquad \mathrm{g}(y,z) \to z$$

where

$$0 :: \bullet \qquad 1 :: \bullet \qquad \mathtt{f} :: (\bullet,\bullet,\bullet) \to \bullet \qquad \mathrm{g} :: (\bullet,\bullet) \to \star$$

- uni-sorted variant $\Theta(\mathcal{R}_\mathsf{T})$ gives cycle

$$\underline{\mathtt{f}(0,1,\mathrm{g}(0,1))} \to_{\Theta(\mathcal{R}_\mathsf{T})} \mathtt{f}(\underline{\mathrm{g}(0,1)},\mathrm{g}(0,1),\mathrm{g}(0,1)) \\ \to_{\Theta(\mathcal{R}_\mathsf{T})} \mathtt{f}(0,\underline{\mathrm{g}(0,1)},\mathrm{g}(0,1)) \to_{\Theta(\mathcal{R}_\mathsf{T})} \mathtt{f}(0,1,\mathrm{g}(0,1))$$

- $\mathcal{R}_\mathsf{T}$ is terminating

Persistency

Toyama's example

Example

TRS $\mathcal{R}_\mathsf{T}$ consists of

$$\mathtt{f}(0,1,x) \to \mathtt{f}(x,x,x) \qquad \mathtt{g}(y,z) \to y \qquad \mathtt{g}(y,z) \to z$$

where

$$0 :: \bullet \qquad 1 :: \bullet \qquad \mathtt{f} :: (\bullet,\bullet,\bullet) \to \bullet \qquad \mathtt{g} :: (\bullet,\bullet) \to \star$$

- uni-sorted vari alien $\mathcal{R}_\mathsf{T})$ gives cycle

$$\underline{\mathtt{f}(0,1,\mathtt{g}(0,1))} \to_{\Theta(\mathcal{R}_\mathsf{T})} \mathtt{f}(\underline{\mathtt{g}(0,1)},\mathtt{g}(0,1),\mathtt{g}(0,1))$$
$$\to_{\Theta(\mathcal{R}_\mathsf{T})} \mathtt{f}(0,\underline{\mathtt{g}(0,1)},\mathtt{g}(0,1)) \to_{\Theta(\mathcal{R}_\mathsf{T})} \mathtt{f}(0,1,\mathtt{g}(0,1))$$

- $\mathcal{R}_\mathsf{T}$ is terminating

Observations

1. “alien s of t contributes to reduction of t only if s itself reducible”
2. “no new aliens are created during reduction”

Observations

1. “alien s of t contributes to reduction of t only if s itself reducible”
2. “no new aliens are created during reduction”

Observations

1. "alien s of t contributes to reduction of t only if s itself reducible"
2. "no new aliens are created during reduction"

Example

TRS $\mathcal{R}_{\mathsf{sum}}$ consists of

$$\mathbf{0} + y \to y \qquad \mathrm{sum}([\,]) \to [\,]$$
$$\mathrm{s}(x) + y \to \mathrm{s}(x + y) \qquad \mathrm{sum}(x : xs) \to x + \mathrm{sum}(xs)$$

where

$$\mathbf{0} :: \mathsf{N} \qquad [\,] :: \mathsf{L} \qquad (+) :: (\mathsf{N}, \mathsf{N}) \to \mathsf{N}$$
$$\mathrm{s} :: \mathsf{N} \to \mathsf{N} \qquad (:) :: (\mathsf{N}, \mathsf{L}) \to \mathsf{L} \qquad \mathrm{sum} :: \mathsf{L} \to \mathsf{N}$$

$$\begin{aligned}
\mathrm{sum}(\mathrm{s}([\,]) : (\mathbf{0} : [\,]) : \mathbf{s}([\,])) &\to_{\Theta(\mathcal{R}_{\mathsf{sum}})} \mathrm{s}([\,]) + \mathrm{sum}((\mathbf{0} : [\,]) : \mathbf{s}([\,])) \\
&\to_{\Theta(\mathcal{R}_{\mathsf{sum}})} \mathrm{s}([\,]) + (\mathbf{0} : [\,]) + \mathrm{sum}(\mathbf{s}([\,])) \\
&\to_{\Theta(\mathcal{R}_{\mathsf{sum}})} \mathrm{s}([\,] + (\mathbf{0} : [\,]) + \mathrm{sum}(\mathbf{s}([\,])))
\end{aligned}$$

Observations

1. "alien s of t contributes to reduction of t only if s itself reducible"
2. "no new aliens are created during reduction"

Example

TRS $\mathcal{R}_{\mathsf{sum}}$ consists of

$$\mathbf{0} + y \to y \qquad \mathrm{sum}([\,]) \to [\,]$$

$$\mathrm{s}(x) + y \to \mathrm{s}(x + y) \qquad \mathrm{sum}(x : xs) \to x + \mathrm{sum}(xs)$$

where

$$\mathbf{0} :: \mathsf{N} \qquad [\,] :: \mathsf{L} \qquad (+) :: (\mathsf{N}, \mathsf{N}) \to \mathsf{N}$$

$$\mathrm{s} :: \mathsf{N} \to \mathsf{N} \qquad (:) :: (\mathsf{N}, \mathsf{L}) \to \mathsf{L} \qquad \mathrm{sum} :: \mathsf{L} \to \mathsf{N}$$

$$\begin{aligned}
\mathrm{sum}(\mathrm{s}(x^{\mathsf{N}}) : x^{\mathsf{N}} : x^{\mathsf{L}}) &\to_{\mathcal{R}_{\mathsf{sum}}} \mathrm{s}(x^{\mathsf{N}}) + \mathrm{sum}(x^{\mathsf{N}} : x^{\mathsf{L}}) \\
&\to_{\mathcal{R}_{\mathsf{sum}}} \mathrm{s}(x^{\mathsf{N}}) + x^{\mathsf{N}} + \mathrm{sum}(x^{\mathsf{L}}) \\
&\to_{\mathcal{R}_{\mathsf{sum}}} \mathrm{s}(x^{\mathsf{N}} + x^{\mathsf{N}} + \mathrm{sum}(x^{\mathsf{L}}))
\end{aligned}$$

Simulating “ill-sorted” Reductions

1. “alien s of t contributes to reduction of t only if s itself reducible”
2. “no new aliens are created during reduction”

- $C[\![s_1, \ldots, s_k]\!] := C[s_1, \ldots, s_k]$ where
  - $s_1, \ldots, s_k$ are the aliens
  - C maximal, well-sorted context
- $s \to_{\Theta(\mathcal{R})} t$ called outer if does not takes place in an alien

Lemma (Step Lemma I)

For outer step $s = C[\![s_1, \ldots, s_k]\!] \to_{\Theta(\mathcal{R})} D[\![t_1, \ldots, t_l]\!] = t$*, either*

Simulating "ill-sorted" Reductions

1. "alien s of t contributes to reduction of t only if s itself reducible"
2. "no new aliens are created during reduction"

- $C[\![s_1,\ldots,s_k]\!] := C[s_1,\ldots,s_k]$ where
  - $s_1,\ldots,s_k$ are the aliens
  - C maximal, well-sorted context
- $s \to_{\Theta(\mathcal{R})} t$ called outer if does not takes place in an alien

Lemma (Step Lemma I)

For outer step $s = C[\![s_1,\ldots,s_k]\!] \to_{\Theta(\mathcal{R})} D[\![t_1,\ldots,t_l]\!] = t$, either

Simulating "ill-sorted" Reductions

1. "alien s of t contributes to reduction of t only if s itself reducible"
2. "no new aliens are created during reduction"

- $C[\![s_1, \ldots, s_k]\!] := C[s_1, \ldots, s_k]$ where
  - $s_1, \ldots, s_k$ are the aliens
  - C maximal, well-sorted context
- $s \to_{\Theta(\mathcal{R})} t$ called outer if does not takes place in an alien

Lemma (Step Lemma I)

For outer step $s = C[\![s_1, \ldots, s_k]\!] \to_{\Theta(\mathcal{R})} D[\![t_1, \ldots, t_l]\!] = t$, *either*

1. $C \to_{\mathcal{R}} D$ *and* $\mathrm{alien}(t) \subseteq \mathrm{alien}(s)$; *or*
2. $C \to_{\mathcal{R}} \Box^{\alpha_i}$ *and* $t \in \mathrm{alien}(s)$

Simulating “ill-sorted” Reductions

1. “alien s of t contributes to reduction of t only if s itself reducible”
2. “no new aliens are created during reduction”

- $C[\![s_1, \ldots, s_k]\!] := C[s_1, \ldots, s_k]$ where
  - $s_1, \ldots, s_k$ are the aliens
  - C maximal, well-sorted context
- $s \to_{\Theta(\mathcal{R})} t$ called outer if does not takes place in an alien

Lemma (Step Lemma I)

For outer step $s = C[\![s_1, \ldots, s_k]\!] \to_{\Theta(\mathcal{R})} D[\![t_1, \ldots, t_l]\!] = t$*, either*

1. $C[x^{\alpha_1}, \ldots, x^{\alpha_k}] \to_{\mathcal{R}} D[x^{\alpha_{i_1}}, \ldots, x^{\alpha_{i_l}}]$ *and* $\mathrm{alien}(t) \subseteq \mathrm{alien}(s)$*; or*
2. $C[x^{\alpha_1}, \ldots, x^{\alpha_k}] \to_{\mathcal{R}} x^{\alpha_i}$ *and* $t \in \mathrm{alien}(s)$

Simulating “ill-sorted” Innermost Reductions

Example

$$\mathtt{f} :: (\bullet, \bullet) \to \bullet \qquad \mathtt{g} :: \bullet \to \bullet \qquad \mathbf{a} :: \star \qquad \mathbf{b} :: \star$$

$$\mathtt{f}(x, x) \to x \qquad \mathtt{g}(\mathtt{f}(x, y)) \to \mathtt{g}(\mathtt{f}(x, y))$$

- $\Theta(\mathcal{R})$ gives rise to infinite reduction

$$\mathtt{g}(\mathtt{f}(\mathbf{a}, \mathbf{b})) \xrightarrow{\mathrm{i}}_{\Theta(\mathcal{R})} \mathtt{g}(\mathtt{f}(\mathbf{a}, \mathbf{b})) \xrightarrow{\mathrm{i}}_{\Theta(\mathcal{R})} \mathtt{g}(\mathtt{f}(\mathbf{a}, \mathbf{b})) \xrightarrow{\mathrm{i}}_{\Theta(\mathcal{R})} \cdots$$

- corresponding reduction

$$\mathtt{g}(\mathtt{f}(x^\bullet, x^\bullet)) \to_{\mathcal{R}} \mathtt{g}(\mathtt{f}(x^\bullet, x^\bullet)) \to_{\mathcal{R}} \mathtt{g}(\mathtt{f}(x^\bullet, x^\bullet)) \to_{\mathcal{R}} \cdots$$

not innermost

Simulating "ill-sorted" Innermost Reductions

Example

$$\mathtt{f} :: (\bullet, \bullet) \to \bullet \qquad \mathtt{g} :: \bullet \to \bullet \qquad \mathbf{a} :: \star \qquad \mathbf{b} :: \star$$

$$\mathtt{f}(x, x) \to x \qquad \mathtt{g}(\mathtt{f}(x, y)) \to \mathtt{g}(\mathtt{f}(x, y))$$

- $\Theta(\mathcal{R})$ gives rise to infinite reduction

$$\mathtt{g}(\mathtt{f}(\mathbf{a}, \mathbf{b})) \xrightarrow{\mathrm{i}}_{\Theta(\mathcal{R})} \mathtt{g}(\mathtt{f}(\mathbf{a}, \mathbf{b})) \xrightarrow{\mathrm{i}}_{\Theta(\mathcal{R})} \mathtt{g}(\mathtt{f}(\mathbf{a}, \mathbf{b})) \xrightarrow{\mathrm{i}}_{\Theta(\mathcal{R})} \cdots$$

- corresponding reduction

$$\mathtt{g}(\mathtt{f}(x^{\bullet}_{\mathbf{a}}, x^{\bullet}_{\mathbf{b}})) \to_{\mathcal{R}} \mathtt{g}(\mathtt{f}(x^{\bullet}_{\mathbf{a}}, x^{\bullet}_{\mathbf{b}})) \to_{\mathcal{R}} \mathtt{g}(\mathtt{f}(x^{\bullet}_{\mathbf{a}}, x^{\bullet}_{\mathbf{b}})) \to_{\mathcal{R}} \cdots$$

not innermost

Simulating "ill-sorted" Innermost Reductions

Lemma (Step Lemma II)

For outer step $s = C[\![s_1, \ldots, s_k]\!] \xrightarrow{(i)}_{\Theta(\mathcal{R})} D[\![t_1, \ldots, t_l]\!] = t$*, either*

1. $C[x_{s_1}^{\alpha_1}, \ldots, x_{s_k}^{\alpha_k}] \xrightarrow{(i)}_{\mathcal{R}} D[x_{t_1}^{\beta_1}, \ldots, x_{t_l}^{\beta_l}]$ *and* $\mathsf{alien}(t) \subseteq \mathsf{alien}(s)$*; or*
2. $C[x_{s_1}^{\alpha_1}, \ldots, x_{s_k}^{\alpha_k}] \xrightarrow{(i)}_{\mathcal{R}} x_{s_i}^{\alpha_i}$ *and* $t \in \mathsf{alien}(s)$

Simulating "ill-sorted" Innermost Reductions

Lemma (Step Lemma II)

For outer step $s = C[\![s_1, \ldots, s_k]\!] \xrightarrow{(i)}_{\Theta(\mathcal{R})} D[\![t_1, \ldots, t_l]\!] = t$, *either*

1. $C[x_{s_1}^{\alpha_1}, \ldots, x_{s_k}^{\alpha_k}] \xrightarrow{(i)}_{\mathcal{R}} D[x_{t_1}^{\beta_1}, \ldots, x_{t_l}^{\beta_l}]$ *and* $\mathrm{alien}(t) \subseteq \mathrm{alien}(s)$*; or*
2. $C[x_{s_1}^{\alpha_1}, \ldots, x_{s_k}^{\alpha_k}] \xrightarrow{(i)}_{\mathcal{R}} x_{s_i}^{\alpha_i}$ *and* $t \in \mathrm{alien}(s)$

Definition

$$t{\downharpoonleft} := C[x_{t_1}^{\alpha_1}, \ldots, x_{t_k}^{\alpha_k}] \text{ where } t = C[\![t_1, \ldots, t_k]\!]$$

Simulating "ill-sorted" Innermost Reductions

Lemma (Step Lemma II)

For outer step $s = C[\![s_1, \ldots, s_k]\!] \xrightarrow{(i)}_{\Theta(\mathcal{R})} D[\![t_1, \ldots, t_l]\!] = t$*, either*

1. $s{\downarrow} \xrightarrow{(i)}_{\mathcal{R}} t{\downarrow}$ *and* $\mathsf{alien}(t) \subseteq \mathsf{alien}(s)$*; or*
2. $s{\downarrow} \xrightarrow{(i)}_{\mathcal{R}} x^{\alpha_i}_{s_i}$ *and* $t \in \mathsf{alien}(s)$

Definition

$$t{\downarrow} := C[x^{\alpha_1}_{t_1}, \ldots, x^{\alpha_k}_{t_k}] \text{ where } t = C[\![t_1, \ldots, t_k]\!]$$

Simulating "ill-sorted" Innermost Reductions

Lemma (Step Lemma II)

For outer step $s = C[\![s_1, \ldots, s_k]\!] \xrightarrow{(i)}_{\Theta(\mathcal{R})} D[\![t_1, \ldots, t_l]\!] = t$, *either*

1. $s{\downarrow} \xrightarrow{(i)}_{\mathcal{R}} t{\downarrow}$ *and* $\mathsf{alien}(t) \subseteq \mathsf{alien}(s)$; *or*
2. $s{\downarrow} \xrightarrow{(i)}_{\mathcal{R}} x^{\alpha_i}_{s_i}$ *and* $t \in \mathsf{alien}(s)$

Proof of Main Result.

- consider t_0 whose aliens are in normal form

$$\begin{array}{ccccccccc} t_0 & \xrightarrow{(i)}_{\Theta(\mathcal{R})} & t_1 & \xrightarrow{(i)}_{\Theta(\mathcal{R})} & t_2 & \xrightarrow{(i)}_{\Theta(\mathcal{R})} & \cdots & \xrightarrow{(i)}_{\Theta(\mathcal{R})} & t_l \\ \vdots & & & & & & & & \vdots \\ t_0{\downarrow} & & & & & & & & t'_l \end{array}$$

$\square$

Simulating "ill-sorted" Innermost Reductions

Lemma (Step Lemma II)

For outer step $s = C[\![s_1, \dots, s_k]\!] \xrightarrow{(i)}_{\Theta(\mathcal{R})} D[\![t_1, \dots, t_l]\!] = t$, *either*

1. $s\!\downarrow \xrightarrow{(i)}_{\mathcal{R}} t\!\downarrow$ *and* $\mathsf{alien}(t) \subseteq \mathsf{alien}(s)$; *or*
2. $s\!\downarrow \xrightarrow{(i)}_{\mathcal{R}} x^{\alpha_i}_{s_i}$ *and* $t \in \mathsf{alien}(s)$

Proof of Main Result.

- consider t_0 whose aliens are in normal form

$$\begin{array}{ccccccccc}
t_0 & \xrightarrow{(i)}_{\Theta(\mathcal{R})} & t_1 & \xrightarrow{(i)}_{\Theta(\mathcal{R})} & t_2 & \xrightarrow{(i)}_{\Theta(\mathcal{R})} & \cdots & \xrightarrow{(i)}_{\Theta(\mathcal{R})} & t_l \\
\vdots & & \vdots & & & & & & \vdots \\
t_0\!\downarrow & \xrightarrow{(i)}_{\mathcal{R}} & t'_1 & & & & & & t'_l
\end{array}$$

$\square$

Simulating "ill-sorted" Innermost Reductions

Lemma (Step Lemma II)

For outer step $s = C[\![s_1, \ldots, s_k]\!] \xrightarrow{(i)}_{\Theta(\mathcal{R})} D[\![t_1, \ldots, t_l]\!] = t$, *either*

1. $s{\downarrow} \xrightarrow{(i)}_{\mathcal{R}} t{\downarrow}$ *and* $\mathrm{alien}(t) \subseteq \mathrm{alien}(s)$; *or*
2. $s{\downarrow} \xrightarrow{(i)}_{\mathcal{R}} x_{s_i}^{\alpha_i}$ *and* $t \in \mathrm{alien}(s)$

Proof of Main Result.

- consider t_0 whose aliens are in normal form

$$
\begin{array}{ccccccccc}
t_0 & \xrightarrow{(i)}_{\Theta(\mathcal{R})} & t_1 & \xrightarrow{(i)}_{\Theta(\mathcal{R})} & t_2 & \xrightarrow{(i)}_{\Theta(\mathcal{R})} & \cdots & \xrightarrow{(i)}_{\Theta(\mathcal{R})} & t_l \\
\vdots & & \vdots & & & & & & \vdots \\
t_0{\downarrow} & \xrightarrow{(i)}_{\mathcal{R}} & t_1{\downarrow} & & & & & & t_l'
\end{array}
$$

$\square$

Simulating “ill-sorted” Innermost Reductions

Lemma (Step Lemma II)

For outer step $s = C[\![s_1, \ldots, s_k]\!] \xrightarrow{(i)}_{\Theta(\mathcal{R})} D[\![t_1, \ldots, t_l]\!] = t$*, either*

1. $s{\downarrow} \xrightarrow{(i)}_{\mathcal{R}} t{\downarrow}$ *and* $\mathsf{alien}(t) \subseteq \mathsf{alien}(s)$*; or*
2. $s{\downarrow} \xrightarrow{(i)}_{\mathcal{R}} x^{\alpha_i}_{s_i}$ *and* $t \in \mathsf{alien}(s)$

Proof of Main Result.

- consider t_0 whose aliens are in normal form

$$\begin{array}{ccccccccc}
t_0 & \xrightarrow{(i)}_{\Theta(\mathcal{R})} & t_1 & \xrightarrow{(i)}_{\Theta(\mathcal{R})} & t_2 & \xrightarrow{(i)}_{\Theta(\mathcal{R})} & \cdots & \xrightarrow{(i)}_{\Theta(\mathcal{R})} & t_l \\
\vdots & & \vdots & & \vdots & & & & \vdots \\
t_0{\downarrow} & \xrightarrow{(i)}_{\mathcal{R}} & t_1{\downarrow} & \xrightarrow{(i)}_{\mathcal{R}} & t_2' & & & & t_l'
\end{array}$$

□

Simulating "ill-sorted" Innermost Reductions

Lemma (Step Lemma II)

For outer step $s = C[\![s_1, \ldots, s_k]\!] \xrightarrow{(i)}_{\Theta(\mathcal{R})} D[\![t_1, \ldots, t_l]\!] = t$, *either*

1. $s{\downarrow} \xrightarrow{(i)}_{\mathcal{R}} t{\downarrow}$ *and* $\mathrm{alien}(t) \subseteq \mathrm{alien}(s)$*; or*
2. $s{\downarrow} \xrightarrow{(i)}_{\mathcal{R}} x_{s_i}^{\alpha_i}$ *and* $t \in \mathrm{alien}(s)$

Proof of Main Result.

- consider t_0 whose aliens are in normal form

$$\begin{tikzcd}
t_0 \arrow[r, "(i)"] \arrow[d, dotted] & _{\Theta(\mathcal{R})}\; t_1 \arrow[r, "(i)"] \arrow[d, dotted] & _{\Theta(\mathcal{R})}\; t_2 \arrow[r, "(i)"] \arrow[d, dotted] & _{\Theta(\mathcal{R})} \cdots \arrow[r, "(i)"] & _{\Theta(\mathcal{R})}\; t_l \arrow[d, dotted] \\
t_0{\downarrow} \arrow[r, "(i)"] & _{\mathcal{R}}\; t_1{\downarrow} \arrow[r, "(i)"] & _{\mathcal{R}}\; t_2{\downarrow} \arrow[r, "(i)"] & _{\mathcal{R}} \cdots \arrow[r, "(i)"] & _{\mathcal{R}}\; t'_l
\end{tikzcd}$$

□

Lesson Learned…

“sorting is not a limiting factor”

Applications

- formative rules *[Fuhs and Kop, 2014]*
- interpretations

$$p(|\mathtt{f}(v_1,\dots,v_k)|) \geqslant [\![\mathtt{f}(v_1,\dots,v_k)]\!] \geqslant \mathsf{dh}(\mathtt{f}(v_1,\dots,v_k),\to)$$

  - consider interpretation where $[\![x : xs]\!] := 2 \cdot x$
  - define

$$t_n = \underbrace{[\dots[}_{n}\,0\,]\dots]]$$

$$[\![t^n]\!] = 2^n \cdot [\![0]\!]$$

$$(\alpha_1,\dots,\alpha_k,\beta,\dots,\beta) \to \beta$$

$$(\underbrace{u_1,\dots,u_k}_{\text{non-recursive}};\underbrace{v_1,\dots,v_l}_{\text{recursive}}) = p_c(u_1,\dots,u_k) + \sum_i v_i$$

- …

Applications

- formative rules *[Fuhs and Kop, 2014]*
- interpretations

$$p(|\mathtt{f}(v_1,\ldots,v_k)|) \geqslant [\![\mathtt{f}(v_1,\ldots,v_k)]\!] \geqslant \mathsf{dh}(\mathtt{f}(v_1,\ldots,v_k),\rightarrow)$$

  - consider interpretation where $[\![x : xs]\!] := 2 \cdot x$
  - define

$$t_n = \underbrace{[[\ldots[}_{n}\,\mathbf{0}\,]\ldots]]$$

$$\implies [\![t^n]\!] = 2^n \cdot [\![\mathbf{0}]\!]$$

$$\mathbf{c} :: (\alpha_1,\ldots,\alpha_k,\beta,\ldots,\beta) \rightarrow \beta$$

$$[\![\mathbf{c}]\!](\underbrace{u_1,\ldots,u_k}_{\text{non-recursive}};\underbrace{v_1,\ldots,v_l}_{\text{recursive}}) = p_{\mathbf{c}}(u_1,\ldots,u_k) + \sum_i v_i$$

- ...

Applications

- formative rules *[Fuhs and Kop, 2014]*
- interpretations

$$p(|\mathtt{f}(v_1,\ldots,v_k)|) \geqslant [\![\mathtt{f}(v_1,\ldots,v_k)]\!] \geqslant \mathsf{dh}(\mathtt{f}(v_1,\ldots,v_k),\rightarrow)$$

  - consider interpretation where $[\![x : xs]\!] := 2 \cdot x$
  - define

$$t_n = \underbrace{[[\ldots[}_{n}\,\mathbf{0}\,]\ldots]]$$

$$\Longrightarrow\ [\![t^n]\!] = 2^n \cdot [\![\mathbf{0}]\!]$$

$$\mathbf{c} :: (\alpha_1,\ldots,\alpha_k,\beta,\ldots,\beta) \rightarrow \beta$$

$$[\![\mathbf{c}]\!](\underbrace{u_1,\ldots,u_k}_{\text{non-recursive}};\underbrace{v_1,\ldots,v_l}_{\text{recursive}}) = p_{\mathbf{c}}(u_1,\ldots,u_k) + \sum_i v_i$$

- ...

Applications

- formative rules *[Fuhs and Kop, 2014]*
- interpretations

$$p(|\mathtt{f}(v_1,\ldots,v_k)|) \geqslant [\![\mathtt{f}(v_1,\ldots,v_k)]\!] \geqslant \mathsf{dh}(\mathtt{f}(v_1,\ldots,v_k),\to)$$

  - consider interpretation where $[\![x : xs]\!] := 2 \cdot x$
  - define

$$t_n = \underbrace{[[\ldots[}_{n}\,\mathbf{0}\,]\ldots]]$$

$$\Rightarrow \quad [\![t^n]\!] = 2^n \cdot [\![\mathbf{0}]\!]$$

$$\mathbf{c} :: (\alpha_1,\ldots,\alpha_k,\beta,\ldots,\beta) \to \beta$$

$$[\![\mathbf{c}]\!](\underbrace{u_1,\ldots,u_k}_{\text{non-recursive}};\underbrace{v_1,\ldots,v_l}_{\text{recursive}}) = p_{\mathbf{c}}(u_1,\ldots,u_k) + \sum_i v_i$$

- ...

Applications

- formative rules *[Fuhs and Kop, 2014]*
- interpretations

$$p(|\mathtt{f}(v_1,\dots,v_k)|) \geqslant [\![\mathtt{f}(v_1,\dots,v_k)]\!] \geqslant \mathsf{dh}(\mathtt{f}(v_1,\dots,v_k),\to)$$

  - consider interpretation where $[\![x : xs]\!] := 2 \cdot x$
  - define

$$t_n = \underbrace{[[\dots[}_{n}\,\mathbf{0}\,]\dots]]$$

$$\Longrightarrow\ [\![t^n]\!] = 2^n \cdot [\![\mathbf{0}]\!]$$

$$\mathbf{c} :: (\alpha_1,\dots,\alpha_k,\beta,\dots,\beta) \to \beta$$

$$[\![\mathbf{c}]\!](\underbrace{u_1,\dots,u_k}_{\text{non-recursive}};\underbrace{v_1,\dots,v_l}_{\text{recursive}}) = p_{\mathbf{c}}(u_1,\dots,u_k) + \sum_i v_i$$

- ...

Applications

- formative rules *[Fuhs and Kop, 2014]*
- interpretations

$$p(|\mathtt{f}(v_1,\dots,v_k)|) \geqslant [\![\mathtt{f}(v_1,\dots,v_k)]\!] \geqslant \mathsf{dh}(\mathtt{f}(v_1,\dots,v_k),\to)$$

  - consider interpretation where $[\![x : xs]\!] := 2 \cdot x$
  - define

$$t_n = \underbrace{[[\dots[}_{n}\,\mathbf{0}\,]\dots]]$$

$$\implies [\![t^n]\!] = 2^n \cdot [\![\mathbf{0}]\!]$$

$$\mathbf{c} :: (\alpha_1,\dots,\alpha_k,\beta,\dots,\beta) \to \beta$$

$$[\![\mathbf{c}]\!](\underbrace{u_1,\dots,u_k}_{\text{non-recursive}};\underbrace{v_1,\dots,v_l}_{\text{recursive}}) = p_{\mathbf{c}}(u_1,\dots,u_k) + \sum_i v_i$$

- ...

Future Work

- extensions

  1. order-sorted rewriting

  $$\alpha \geqslant_{\mathcal{S}} \beta$$

  2. "polymorphism"

  $$\texttt{rev} :: \mathsf{List}(\alpha) \to \mathsf{List}(\alpha)$$

  3. ...

- development of techniques for sorted rewrite systems

Future Work

- extensions
  1. order-sorted rewriting
     $$\alpha \geqslant_{\mathcal{S}} \beta$$
  2. “polymorphism”
     $$\texttt{rev} :: \mathsf{List}(\alpha) \to \mathsf{List}(\alpha)$$
  3. ...
- development of techniques for sorted rewrite systems