From outermost termination to innermost termination

René Thiemann

Computational Logic
Institute of Computer Science

University of Innsbruck

SOFSEM ’09, Jan 26, 2009

Motivation

termination analysis of (functional) programs

- term rewriting is basic evaluation mechanism of func. programs

$\Rightarrow$ consider termination of term rewrite systems (TRSs)

Motivation

termination analysis of (functional) programs

- term rewriting is basic evaluation mechanism of func. programs

$\Rightarrow$ consider termination of term rewrite systems (TRSs)

- termination analysis of TRSs is well-studied

Knuth-Bendix order, recursive path order, polynomial orders, dependency pairs, semantic and predictive labeling, matchbounds, matrix interpretations, loop detection, ...

Motivation

termination analysis of (functional) programs

- term rewriting is basic evaluation mechanism of func. programs

$\Rightarrow$ consider termination of term rewrite systems (TRSs)

- termination analysis of TRSs is well-studied

  Knuth-Bendix order, recursive path order, polynomial orders, dependency pairs, semantic and predictive labeling, matchbounds, matrix interpretations, loop detection, ...

Motivation

termination analysis of (functional) programs

- term rewriting is basic evaluation mechanism of func. programs

$\Rightarrow$ consider termination of term rewrite systems (TRSs)

- termination analysis of TRSs is well-studied

  Knuth-Bendix order, recursive path order, polynomial orders, dependency pairs, semantic and predictive labeling, matchbounds, matrix interpretations, loop detection, ...

programs often use evaluation strategies like innermost or outermost

Motivation

termination analysis of (functional) programs

- term rewriting is basic evaluation mechanism of func. programs

$\Rightarrow$ consider termination of term rewrite systems (TRSs)

- termination analysis of TRSs is well-studied

  Knuth-Bendix order, recursive path order, polynomial orders, dependency pairs, semantic and predictive labeling, matchbounds, matrix interpretations, loop detection, ...

programs often use evaluation strategies like innermost or outermost

- innermost termination analysis is well-studied

  innermost recursive path order, (negative) polynomial orders, innermost dependency pairs, innermost semantic and predictive labeling, bounded increase, innermost loop detection, ...

Motivation

termination analysis of (functional) programs

- term rewriting is basic evaluation mechanism of func. programs

$\Rightarrow$ consider termination of term rewrite systems (TRSs)

- termination analysis of TRSs is well-studied

  Knuth-Bendix order, recursive path order, polynomial orders, dependency pairs, semantic and predictive labeling, matchbounds, matrix interpretations, loop detection, ...

programs often use evaluation strategies like innermost or outermost

- innermost termination analysis is well-studied

  innermost recursive path order, (negative) polynomial orders, innermost dependency pairs, innermost semantic and predictive labeling, bounded increase, innermost loop detection, ...

Analyzing outermost termination automatically

- direct approach of Fissore, Gnaedig, Kirchner (2002, Cariboo)

  - stand alone, does not profit much from other techniques
  - cannot be used to disprove outermost termination
  + applicable on every TRS

Analyzing outermost termination automatically

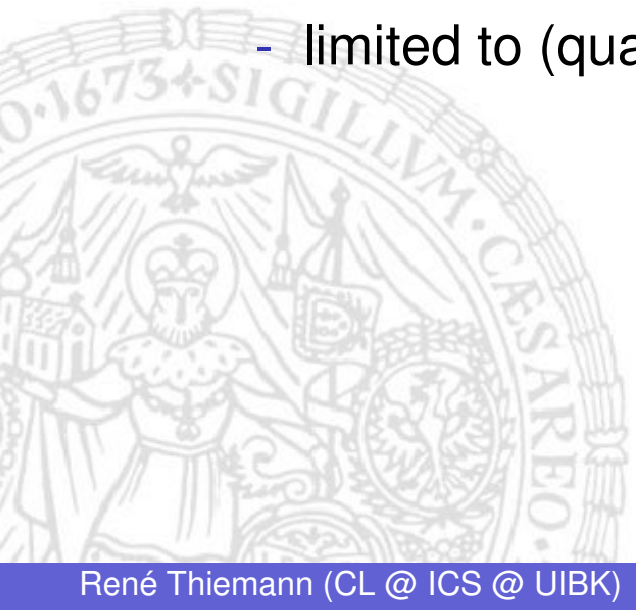
- direct approach of Fissore, Gnaedig, Kirchner (2002, Cariboo)

  - stand alone, does not profit much from other techniques
  - cannot be used to disprove outermost termination
  + applicable on every TRS

- transformation T of Raffelsieper and Zantema (2008):

  $\mathcal{R}$ is outermost terminating if $T(\mathcal{R})$ is terminating

  + profits from all existing techniques for termination
  - cannot be used to disprove outermost termination
  - limited to (quasi) left-linear TRSs

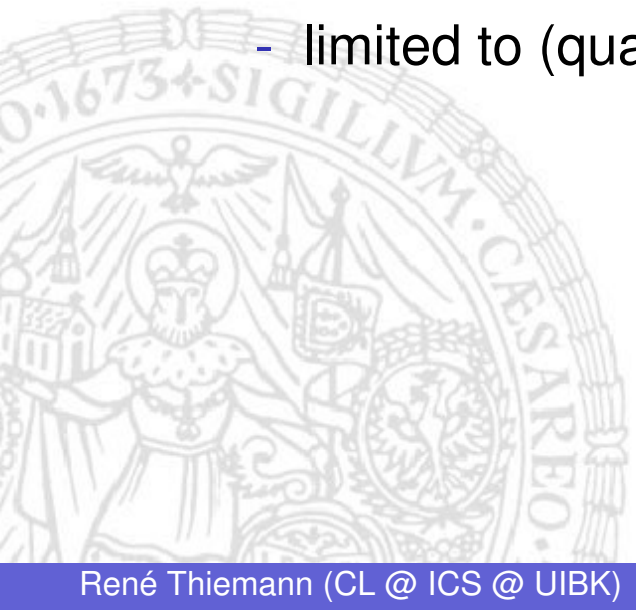


Analyzing outermost termination automatically

- direct approach of Fissore, Gnaedig, Kirchner (2002, Cariboo)
  - \+ works on original TRS
  - \- stand alone, does not profit much from other techniques
  - \- cannot be used to disprove outermost termination
  - \+ applicable on every TRS
- transformation T of Raffelsieper and Zantema (2008):

  $\mathcal{R}$ is outermost terminating if $T(\mathcal{R})$ is terminating

  - \+ profits from all existing techniques for termination
  - \- cannot be used to disprove outermost termination
  - \- limited to (quasi) left-linear TRSs



Analyzing outermost termination automatically

- direct approach of Fissore, Gnaedig, Kirchner (2002, Cariboo)
  - \+ works on original TRS
  - \- stand alone, does not profit much from other techniques
  - \- cannot be used to disprove outermost termination
  - \+ applicable on every TRS
- transformation T of Raffelsieper and Zantema (2008):

  $\mathcal{R}$ is outermost terminating if $T(\mathcal{R})$ is terminating

  - \+ profits from all existing techniques for termination
  - \- cannot be used to disprove outermost termination
  - \- limited to (quasi) left-linear TRSs
- contribution: a new transformation $'$

  $\mathcal{R}$ is outermost terminating iff $\mathcal{R}'$ is innermost terminating

  - \+ profits from all existing techniques for innermost termination
  - \+ can be used to disprove outermost termination
  - \+ applicable on every TRS

Outline

1. Motivation
2. Preliminaries
3. The Transformation
4. The Transformation - Soundness and Completeness
5. Experiments

Rewriting

Definition

- TRS: set of rules $\ell \to r$

Example

$$\mathsf{zeros} \to \mathsf{c}(0, \mathsf{zeros})$$
$$\mathsf{c}(x, xs) \to \mathsf{terminate}$$

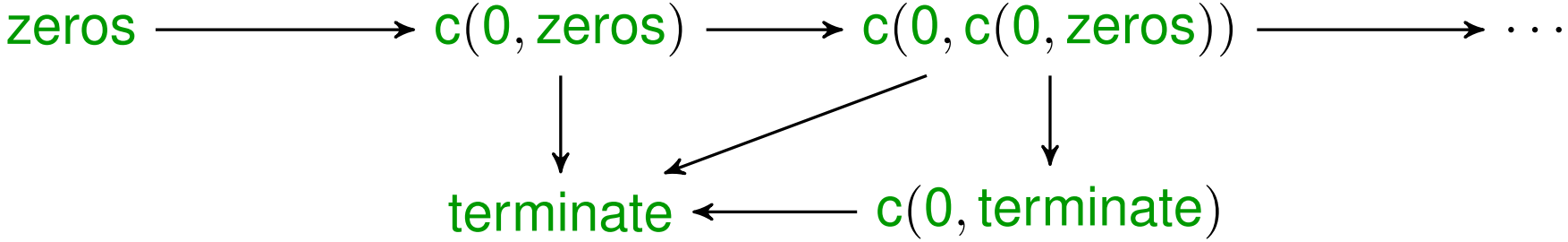
Rewriting

Definition

- TRS: set of rules $\ell \to r$
- rewrite step at position p: $t = C[\ell\sigma]_p \to_{\mathcal{R},p} C[r\sigma]_p = s$

Example

$$\mathsf{zeros} \to \mathsf{c}(0, \mathsf{zeros})$$
$$\mathsf{c}(x, xs) \to \mathsf{terminate}$$



Rewriting

Definition

- TRS: set of rules $\ell \to r$
- rewrite step at position p: $t = C[\ell\sigma]_p \to_{\mathcal{R},p} C[r\sigma]_p = s$
- innermost rewriting $\xrightarrow{\text{i}}_{\mathcal{R}}$: $t \to_{\mathcal{R},p} s$ and no step possible below p

Example

$$\text{zeros} \to \text{c}(0, \text{zeros})$$
$$\text{c}(x, xs) \to \text{terminate}$$

$$\text{zeros} \xrightarrow{\text{i}} \text{c}(0, \text{zeros}) \xrightarrow{\text{i}} \text{c}(0, \text{c}(0, \text{zeros})) \xrightarrow{\text{i}} \cdots$$

$$\text{terminate} \xleftarrow{\text{i}} \text{c}(0, \text{terminate})$$

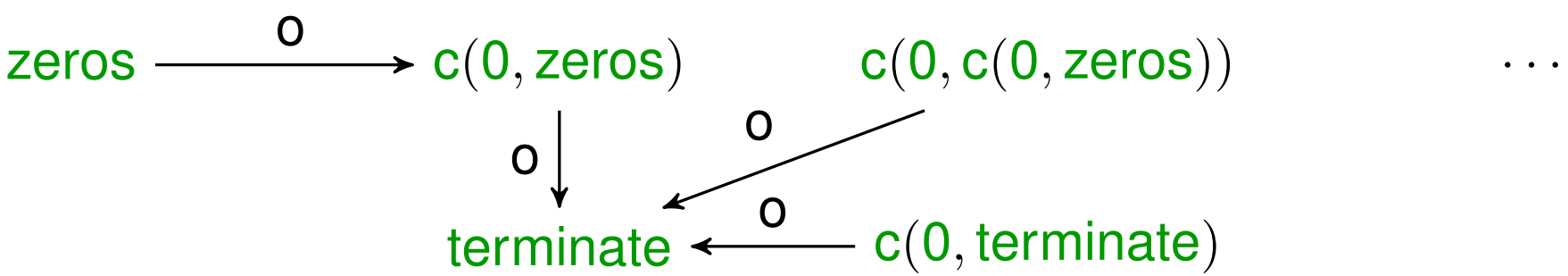
Rewriting

Definition

- TRS: set of rules $\ell \to r$
- rewrite step at position p: $t = C[\ell\sigma]_p \to_{\mathcal{R},p} C[r\sigma]_p = s$
- innermost rewriting $\xrightarrow{\text{i}}_{\mathcal{R}}$: $t \to_{\mathcal{R},p} s$ and no step possible below p
- outermost rewriting $\xrightarrow{\text{o}}_{\mathcal{R}}$: $t \to_{\mathcal{R},p} s$ and no step possible above p

Example

$$\mathsf{zeros} \to \mathsf{c}(0, \mathsf{zeros})$$
$$\mathsf{c}(x, xs) \to \mathsf{terminate}$$



Termination

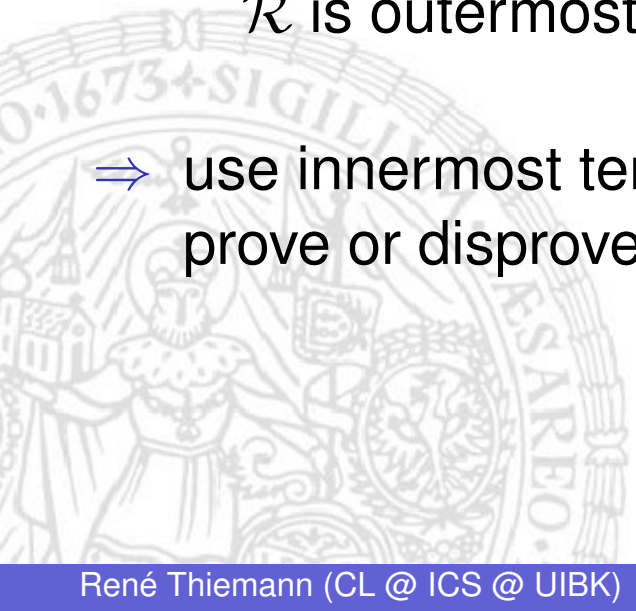
Definition

$\mathcal{R}$ is innermost / outermost terminating iff $\xrightarrow{\text{i}}_{\mathcal{R}}$ / $\xrightarrow{\text{o}}_{\mathcal{R}}$ is well-founded

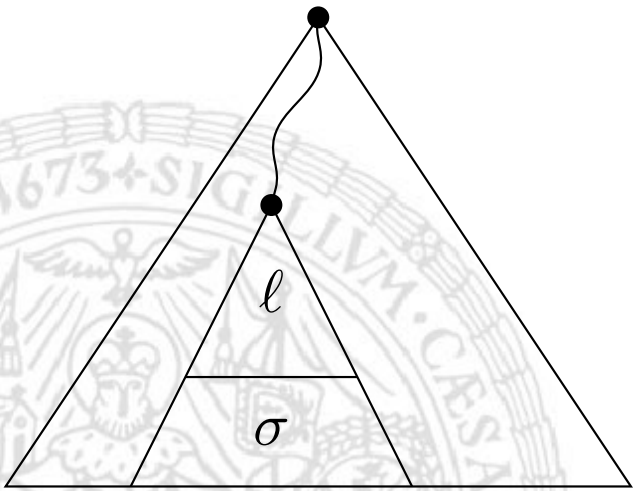
aim: transform given TRS $\mathcal{R}$ into TRS $\mathcal{R}'$ such that

$\mathcal{R}$ is outermost terminating iff $\mathcal{R}'$ is innermost terminating

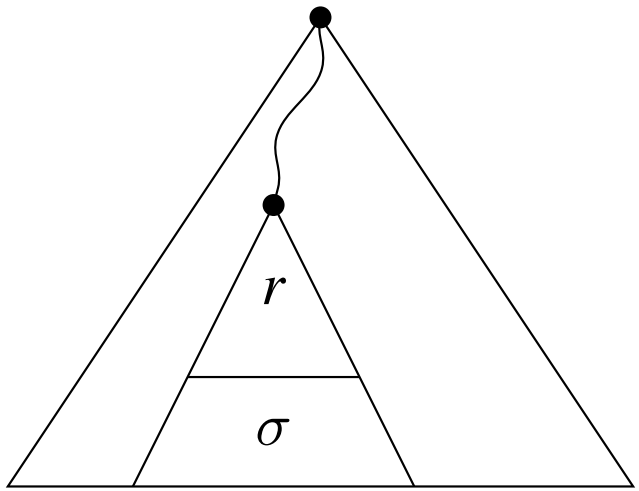
$\Rightarrow$ use innermost termination techniques to prove or disprove outermost termination



Simulation of outermost rewriting

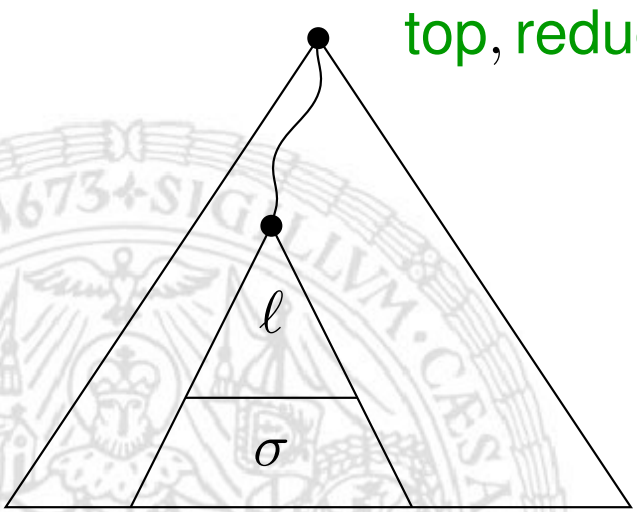


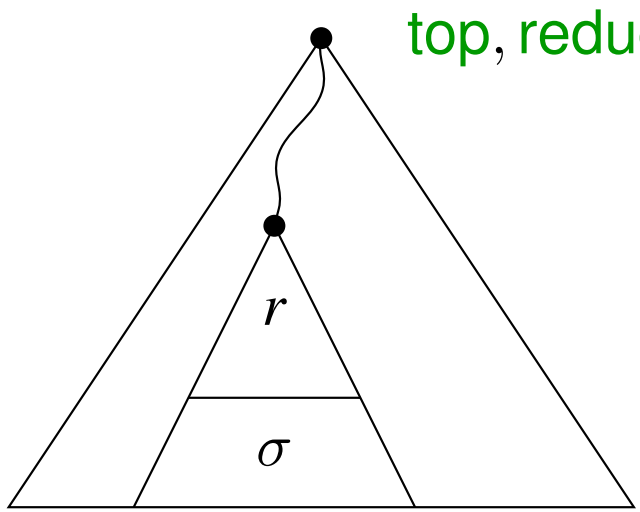
$\xrightarrow{\mathsf{o}}_{\mathcal{R}}$



Simulation of outermost rewriting

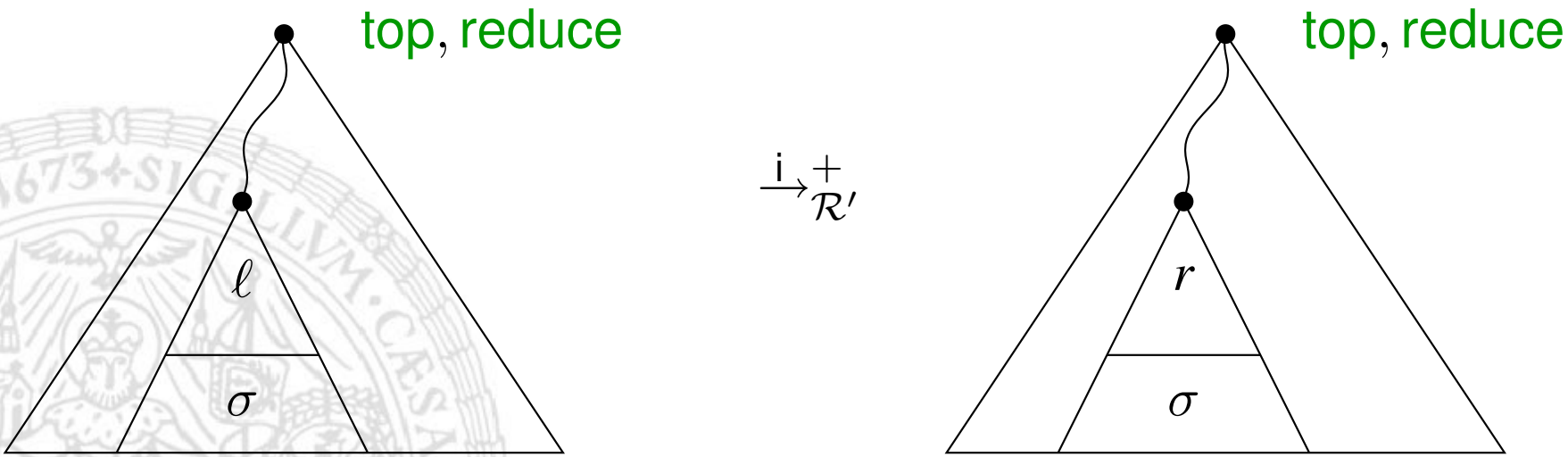
- enrich signature by marker-symbols (reduce, top, go_up)





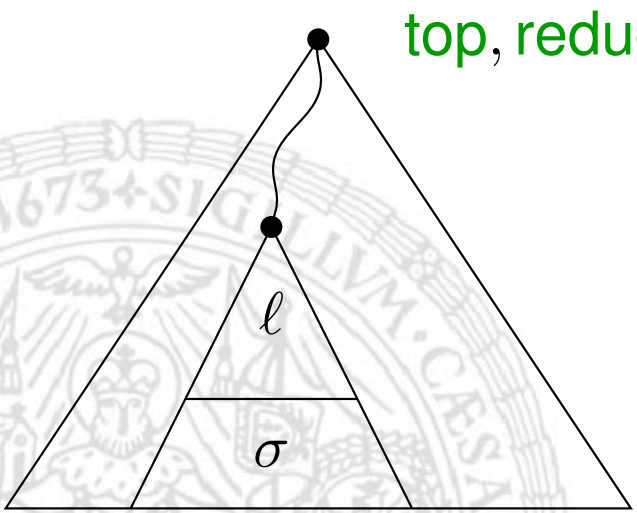
Simulation of outermost rewriting

- enrich signature by marker-symbols (reduce, top, go_up)
- simulate one outermost step in four phases by innermost steps



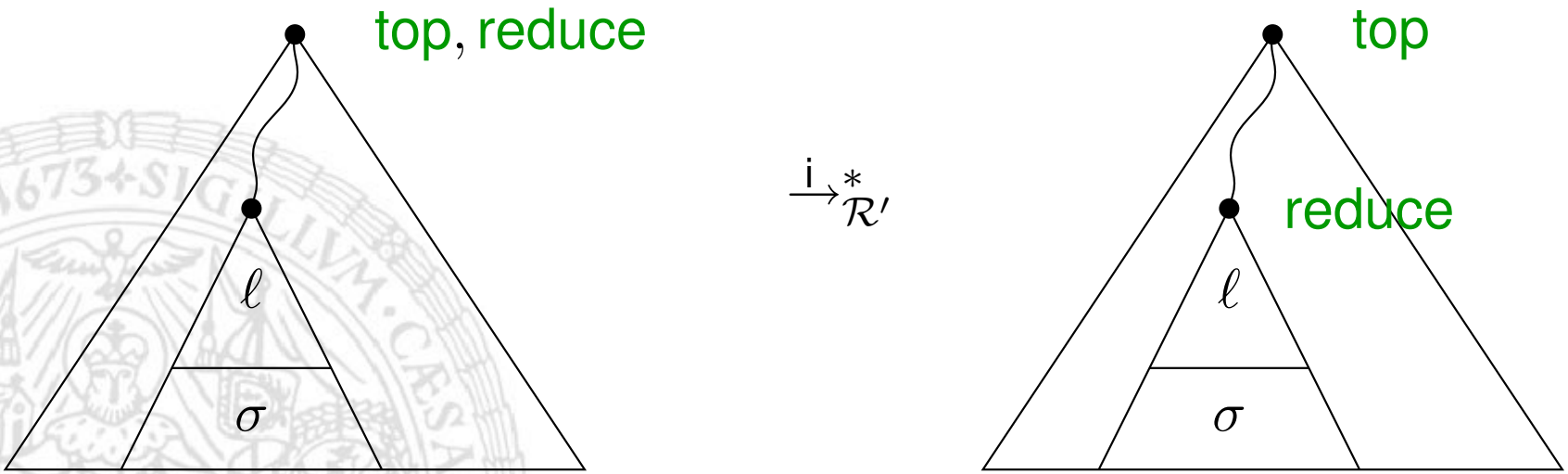
Simulation of outermost rewriting

- enrich signature by marker-symbols (reduce, top, go_up)
- simulate one outermost step in four phases by innermost steps



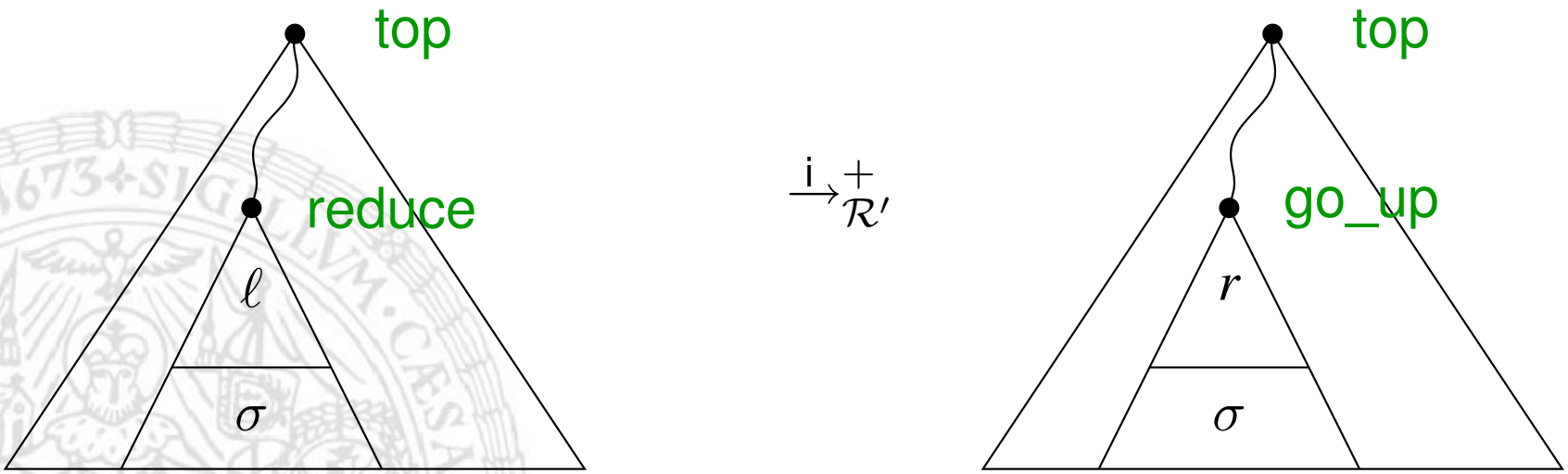
Simulation of outermost rewriting

- enrich signature by marker-symbols (reduce, top, go_up)
- simulate one outermost step in four phases by innermost steps
  1. move reduce-marker to position of redex



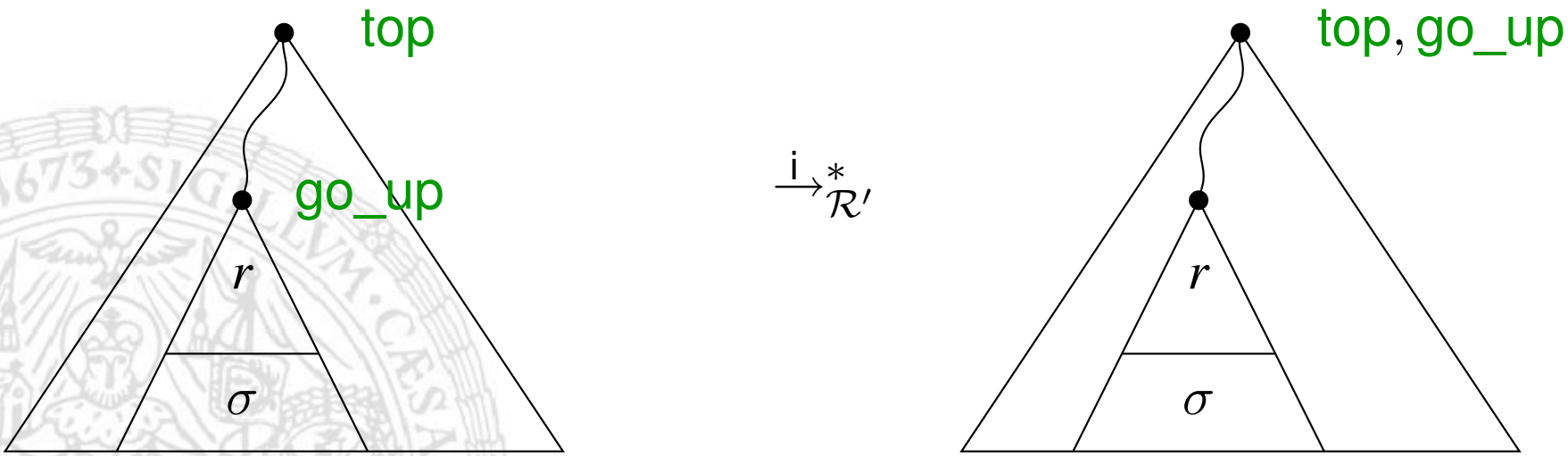
Simulation of outermost rewriting

- enrich signature by marker-symbols (reduce, top, go_up)
- simulate one outermost step in four phases by innermost steps
  1. move reduce-marker to position of redex
  2. perform step and change reduce- to go_up-marker



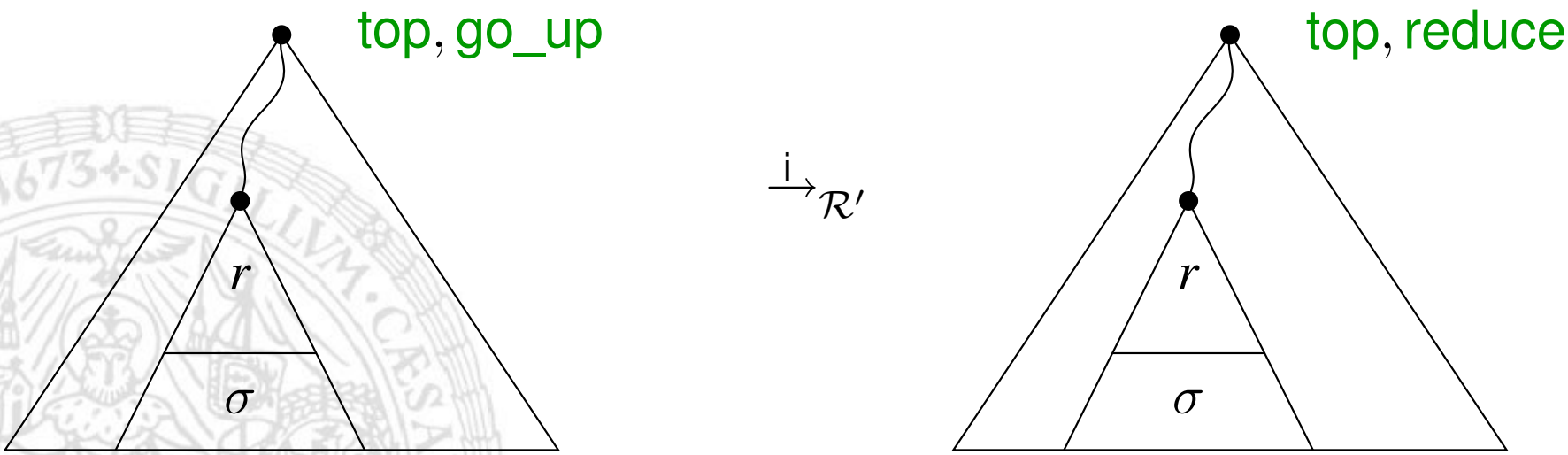
Simulation of outermost rewriting

- enrich signature by marker-symbols (reduce, top, go_up)
- simulate one outermost step in four phases by innermost steps
  1. move reduce-marker to position of redex
  2. perform step and change reduce- to go_up-marker
  3. move go_up-marker to top again



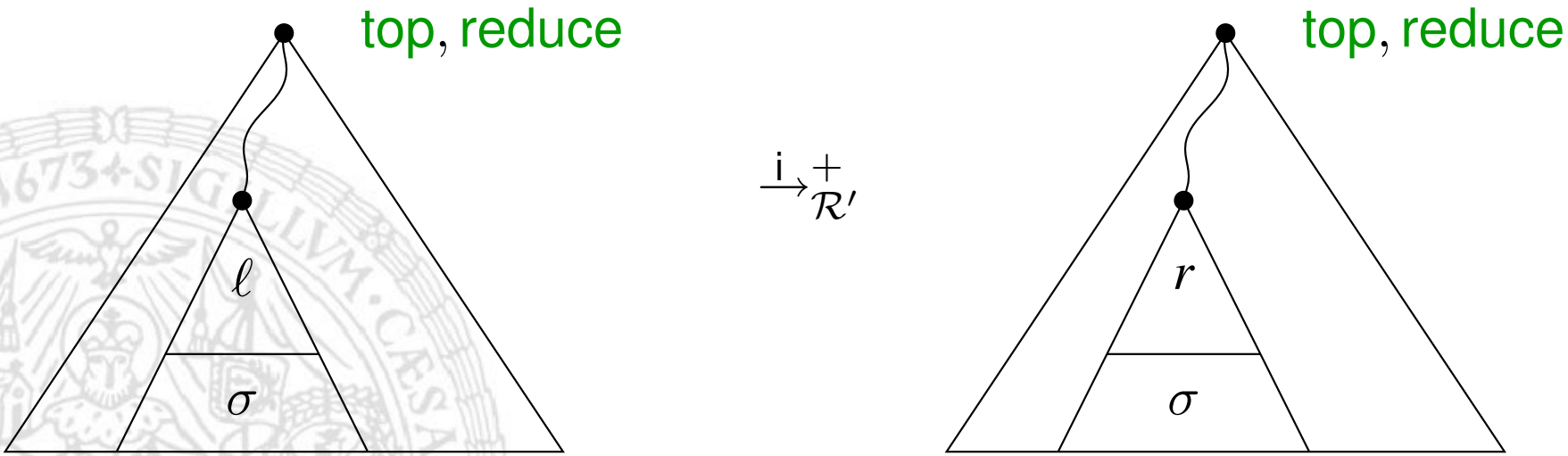
Simulation of outermost rewriting

- enrich signature by marker-symbols (reduce, top, go_up)
- simulate one outermost step in four phases by innermost steps
  1. move reduce-marker to position of redex
  2. perform step and change reduce- to go_up-marker
  3. move go_up-marker to top again
  4. switch go_up- to reduce-marker



Simulation of outermost rewriting

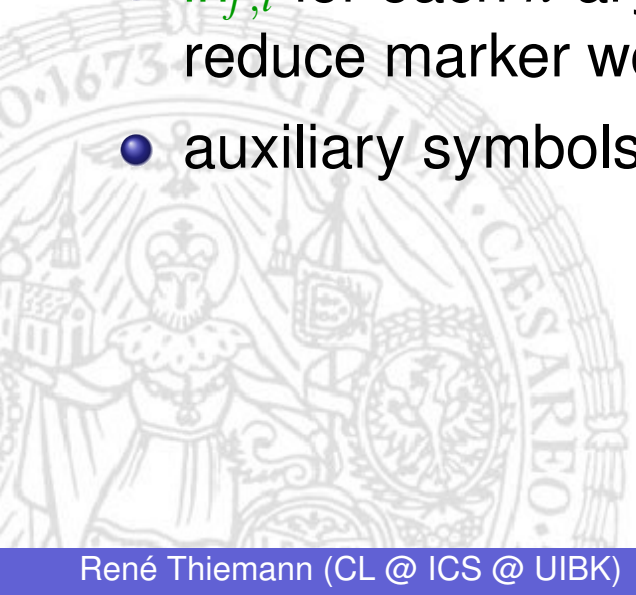
- enrich signature by marker-symbols (reduce, top, go_up)
- simulate one outermost step in four phases by innermost steps
  1. move reduce-marker to position of redex
  2. perform step and change reduce- to go_up-marker
  3. move go_up-marker to top again
  4. switch go_up- to reduce-marker



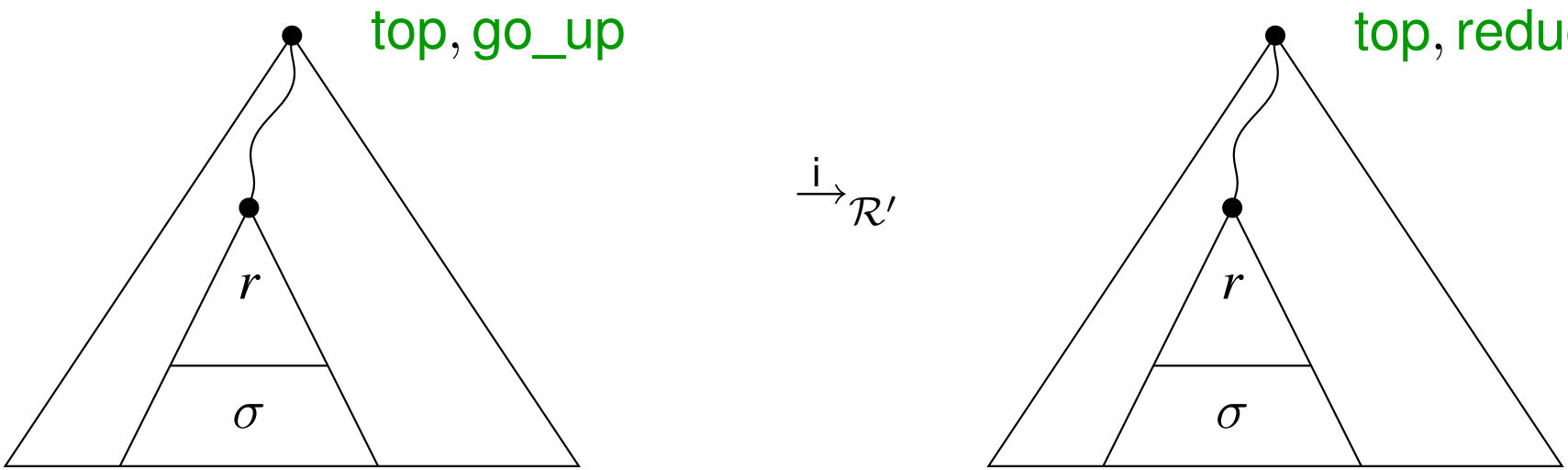
Enriching the signature

the enriched signature Σ' consists of

- all symbols of the old signature Σ
- top to mark the top of a term
- reduce to mark where we are allowed to reduce
- go_up to mark that we have to go up again
- $\text{in}_{f,i}$ for each n-ary symbol $f \in \Sigma$ and $1 \leq i \leq n$ to mark that the reduce marker went below an f into the i-th argument
- auxiliary symbols result, check_f, and redex_f



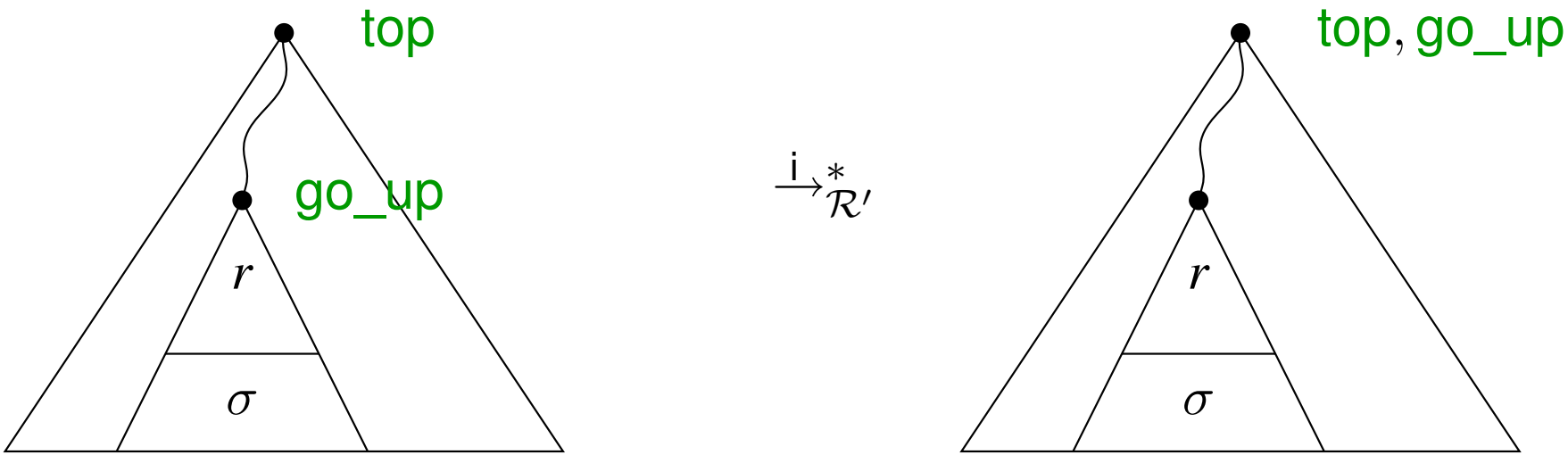
4. phase: switch from go_up to reduce at top position



add following rule to $\mathcal{R}'$:

$$\mathsf{top}(\mathsf{go\_up}(x)) \to \mathsf{top}(\mathsf{reduce}(x))$$

3. phase: move go_up back to top

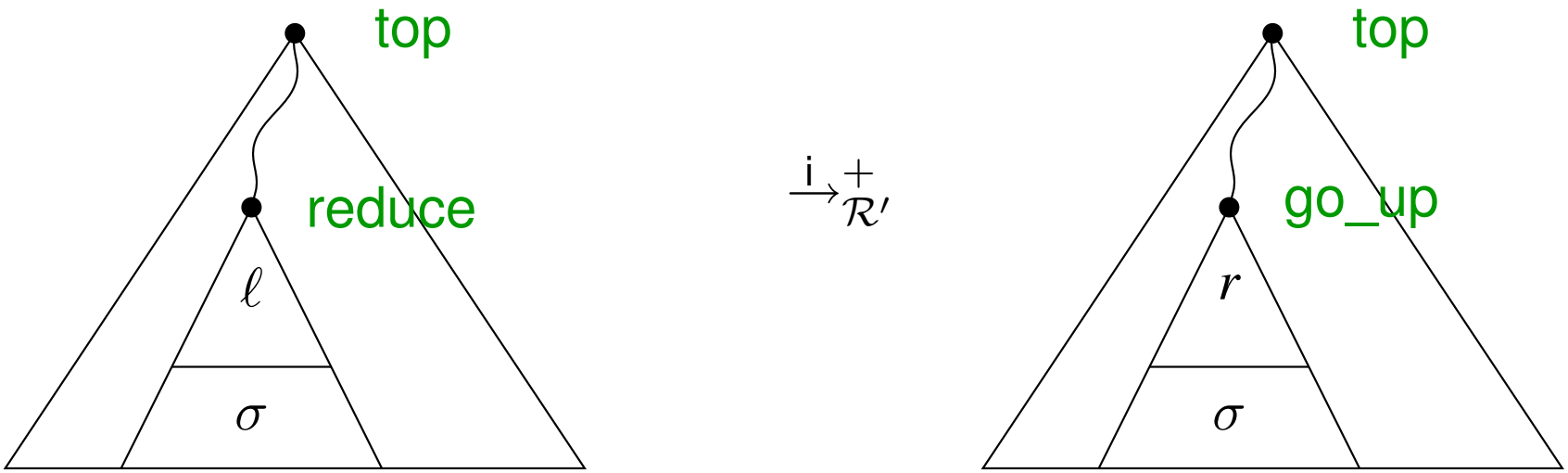


add following rules to $\mathcal{R}'$:

$$\mathsf{in}_{f,i}(x_1, \ldots, \mathsf{go\_up}(x_i), \ldots, x_n) \to \mathsf{go\_up}(f(x_1, \ldots, x_n))$$

for all $f \in \Sigma, 1 \leq i \leq n$

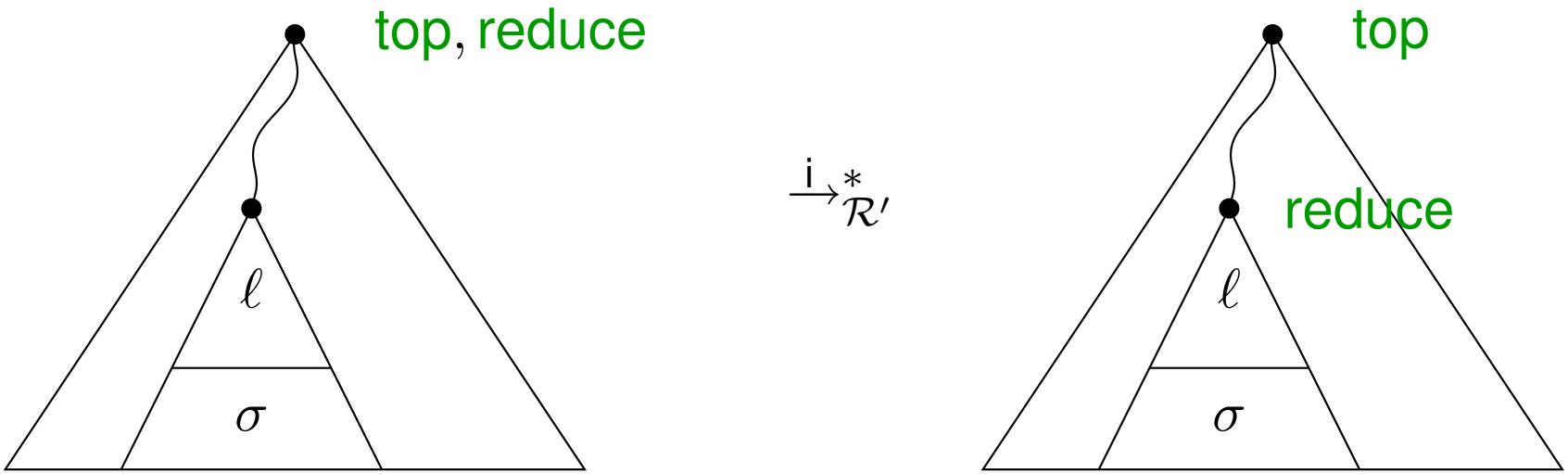
2. phase: perform reduction and switch to go_up



add following rules to $\mathcal{R}'$:

$$\mathsf{reduce}(f(x_1,\ldots,x_n)) \to \mathsf{check}_f(\mathsf{redex}_f(x_1,\ldots,x_n)) \qquad \text{for all } f \in \Sigma$$
$$\mathsf{redex}_f(\ell_1,\ldots,\ell_n) \to \mathsf{result}(r) \qquad \text{for all } f(\ell_1,\ldots,\ell_n) \to r \in \mathcal{R}$$
$$\mathsf{check}_f(\mathsf{result}(x)) \to \mathsf{go\_up}(x) \qquad \text{for all } f \in \Sigma$$

1. phase: move reduce to outermost redexes



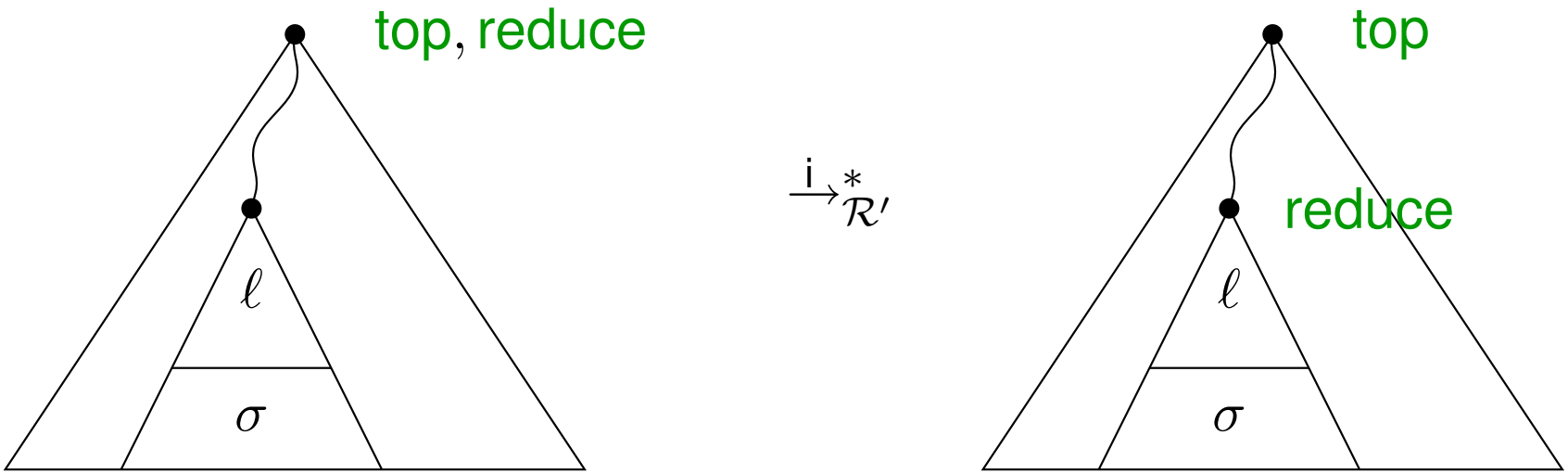
recall existing rules of $\mathcal{R}'$:

$$\mathsf{reduce}(f(x_1,\ldots,x_n)) \to \mathsf{check}_f(\mathsf{redex}_f(x_1,\ldots,x_n)) \qquad \text{for all } f \in \Sigma$$

add following rules to $\mathcal{R}'$:

$$\mathsf{check}_f(\mathsf{redex}_f(x_1,\ldots,x_n)) \to \mathsf{in}_{f,i}(x_1,\ldots,\mathsf{reduce}(x_i),\ldots,x_n)$$
$$\text{for all } f \in \Sigma, 1 \le i \le n$$

1. phase: move reduce to outermost redexes



recall existing rules of $\mathcal{R}'$:

$$\mathsf{reduce}(f(x_1,\ldots,x_n)) \to \mathsf{check}_f(\mathsf{redex}_f(x_1,\ldots,x_n)) \qquad \text{for all } f \in \Sigma$$

$$\mathsf{redex}_f(\ell_1,\ldots,\ell_n) \to \mathsf{result}(r) \qquad \text{for all } f(\ell_1,\ldots,\ell_n) \to r \in \mathcal{R}$$

add following rules to $\mathcal{R}'$:

$$\mathsf{check}_f(\mathsf{redex}_f(x_1,\ldots,x_n)) \to \mathsf{in}_{f,i}(x_1,\ldots,\mathsf{reduce}(x_i),\ldots,x_n)$$
$$\text{for all } f \in \Sigma, 1 \le i \le n$$

The transformed TRS $\mathcal{R}'$

$$\begin{aligned}
\mathsf{reduce}(f(x_1,\ldots,x_n)) &\to \mathsf{check}_f(\mathsf{redex}_f(x_1,\ldots,x_n))\\
\mathsf{check}_f(\mathsf{redex}_f(x_1,\ldots,x_n)) &\to \mathsf{in}_{f,i}(x_1,\ldots,\mathsf{reduce}(x_i),\ldots,x_n)\\
\mathsf{redex}_f(\ell_1,\ldots,\ell_n) &\to \mathsf{result}(r)\\
\mathsf{check}_f(\mathsf{result}(x)) &\to \mathsf{go_up}(x)\\
\mathsf{in}_{f,i}(x_1,\ldots,\mathsf{go_up}(x_i),\ldots,x_n) &\to \mathsf{go_up}(f(x_1,\ldots,x_n))\\
\mathsf{top}(\mathsf{go_up}(x)) &\to \mathsf{top}(\mathsf{reduce}(x))
\end{aligned}$$

Theorem

$\mathcal{R}$ is outermost terminating iff $\mathcal{R}'$ is innermost terminating

soundness proof.

from

$$t_1 \xrightarrow{\mathrm{o}}_{\mathcal{R}} t_2 \xrightarrow{\mathrm{o}}_{\mathcal{R}} t_3 \xrightarrow{\mathrm{o}}_{\mathcal{R}} \cdots$$

obtain

$$\mathsf{top}(\mathsf{reduce}(t_1)) \xrightarrow{\mathrm{i}}{}^{+}_{\mathcal{R}'} \mathsf{top}(\mathsf{reduce}(t_2)) \xrightarrow{\mathrm{i}}{}^{+}_{\mathcal{R}'} \mathsf{top}(\mathsf{reduce}(t_3)) \xrightarrow{\mathrm{i}}{}^{+}_{\mathcal{R}'} \cdots$$

The transformed TRS $\mathcal{R}'$

$$\begin{aligned}
\mathsf{reduce}(f(x_1,\ldots,x_n)) &\to \mathsf{check}_f(\mathsf{redex}_f(x_1,\ldots,x_n))\\
\mathsf{check}_f(\mathsf{redex}_f(x_1,\ldots,x_n)) &\to \mathsf{in}_{f,i}(x_1,\ldots,\mathsf{reduce}(x_i),\ldots,x_n)\\
\mathsf{redex}_f(\ell_1,\ldots,\ell_n) &\to \mathsf{result}(r)\\
\mathsf{check}_f(\mathsf{result}(x)) &\to \mathsf{go_up}(x)\\
\mathsf{in}_{f,i}(x_1,\ldots,\mathsf{go_up}(x_i),\ldots,x_n) &\to \mathsf{go_up}(f(x_1,\ldots,x_n))\\
\mathsf{top}(\mathsf{go_up}(x)) &\to \mathsf{top}(\mathsf{reduce}(x))
\end{aligned}$$

Theorem

$\mathcal{R}$ is outermost terminating iff $\mathcal{R}'$ is innermost terminating

soundness proof.

from

$$t_1 \xrightarrow{\mathsf{o}}_{\mathcal{R}} t_2 \xrightarrow{\mathsf{o}}_{\mathcal{R}} t_3 \xrightarrow{\mathsf{o}}_{\mathcal{R}} \ldots$$

obtain

$$\mathsf{top}(\mathsf{reduce}(t_1)) \xrightarrow{\mathsf{i}}{}^{+}_{\mathcal{R}'} \mathsf{top}(\mathsf{reduce}(t_2)) \xrightarrow{\mathsf{i}}{}^{+}_{\mathcal{R}'} \mathsf{top}(\mathsf{reduce}(t_3)) \xrightarrow{\mathsf{i}}{}^{+}_{\mathcal{R}'} \ldots$$

Completeness: proof sketch

1. if $\mathcal{R}'$ is not innermost terminating then there is infinite derivation

$$\mathsf{top}(\mathsf{go\_up}(t_1)) \xrightarrow{\mathsf{i}}_{\mathcal{R}',\epsilon} \mathsf{top}(\mathsf{reduce}(t_1)) \xrightarrow{\mathsf{i}}^*_{\mathcal{R}',>\epsilon} \mathsf{top}(\mathsf{go\_up}(t_2)) \xrightarrow{\mathsf{i}}_{\mathcal{R}',\epsilon} \cdots$$

2. from

$$\mathsf{reduce}(t_i) \xrightarrow{\mathsf{i}}^*_{\mathcal{R}'} \mathsf{go\_up}(t_{i+1})$$

conclude

$$\mathcal{O}(t_i) \xrightarrow{\mathsf{o}}_{\mathcal{R}} \mathcal{O}(t_{i+1})$$

where $\mathcal{O}$ maps terms of extended signature to terms of original signature

$\Rightarrow$ obtain infinite outermost derivation

$$\mathcal{O}(t_1) \xrightarrow{\mathsf{o}}_{\mathcal{R}} \mathcal{O}(t_2) \xrightarrow{\mathsf{o}}_{\mathcal{R}} \mathcal{O}(t_3) \xrightarrow{\mathsf{o}}_{\mathcal{R}} \cdots$$

Completeness: proof sketch

1. if $\mathcal{R}'$ is not innermost terminating then there is infinite derivation

   $$\mathsf{top}(\mathsf{go\_up}(t_1)) \xrightarrow{\mathsf{i}}_{\mathcal{R}',\epsilon} \mathsf{top}(\mathsf{reduce}(t_1)) \xrightarrow{\mathsf{i}}^*_{\mathcal{R}',>\epsilon} \mathsf{top}(\mathsf{go\_up}(t_2)) \xrightarrow{\mathsf{i}}_{\mathcal{R}',\epsilon} \cdots$$

2. from

   $$\mathsf{reduce}(t_i) \xrightarrow{\mathsf{i}}^*_{\mathcal{R}'} \mathsf{go\_up}(t_{i+1})$$

   conclude

   $$\mathcal{O}(t_i) \xrightarrow{\mathsf{o}}_{\mathcal{R}} \mathcal{O}(t_{i+1})$$

   where $\mathcal{O}$ maps terms of extended signature to terms of original signature

   $\Rightarrow$ obtain infinite outermost derivation

   $$\mathcal{O}(t_1) \xrightarrow{\mathsf{o}}_{\mathcal{R}} \mathcal{O}(t_2) \xrightarrow{\mathsf{o}}_{\mathcal{R}} \mathcal{O}(t_3) \xrightarrow{\mathsf{o}}_{\mathcal{R}} \cdots$$

Completeness: proof sketch

1. if $\mathcal{R}'$ is not innermost terminating then there is infinite derivation

$$\mathsf{top}(\mathsf{go\_up}(t_1)) \xrightarrow{\mathsf{i}}_{\mathcal{R}',\epsilon} \mathsf{top}(\mathsf{reduce}(t_1)) \xrightarrow{\mathsf{i}}^*_{\mathcal{R}',>\epsilon} \mathsf{top}(\mathsf{go\_up}(t_2)) \xrightarrow{\mathsf{i}}_{\mathcal{R}',\epsilon} \cdots$$

2. from

$$\mathsf{reduce}(t_i) \xrightarrow{\mathsf{i}}^*_{\mathcal{R}'} \mathsf{go\_up}(t_{i+1})$$

conclude

$$\mathcal{O}(t_i) \xrightarrow{\mathsf{o}}_{\mathcal{R}} \mathcal{O}(t_{i+1})$$

where $\mathcal{O}$ maps terms of extended signature to terms of original signature

$\Rightarrow$ obtain infinite outermost derivation

$$\mathcal{O}(t_1) \xrightarrow{\mathsf{o}}_{\mathcal{R}} \mathcal{O}(t_2) \xrightarrow{\mathsf{o}}_{\mathcal{R}} \mathcal{O}(t_3) \xrightarrow{\mathsf{o}}_{\mathcal{R}} \cdots$$

Completeness: proof sketch

1. if $\mathcal{R}'$ is not innermost terminating then there is infinite derivation

$$\mathsf{top}(\mathsf{go\_up}(t_1)) \xrightarrow{\mathsf{i}}_{\mathcal{R}',\epsilon} \mathsf{top}(\mathsf{reduce}(t_1)) \xrightarrow{\mathsf{i}}^*_{\mathcal{R}',>\epsilon} \mathsf{top}(\mathsf{go\_up}(t_2)) \xrightarrow{\mathsf{i}}_{\mathcal{R}',\epsilon} \cdots$$

2. from

$$\mathsf{reduce}(t_i) \xrightarrow{\mathsf{i}}^*_{\mathcal{R}'} \mathsf{go\_up}(t_{i+1})$$

conclude

$$\mathcal{O}(t_i) \xrightarrow{\mathsf{o}}_{\mathcal{R}} \mathcal{O}(t_{i+1})$$

where $\mathcal{O}$ maps terms of extended signature to terms of original signature

$\Rightarrow$ obtain infinite outermost derivation

$$\mathcal{O}(t_1) \xrightarrow{\mathsf{o}}_{\mathcal{R}} \mathcal{O}(t_2) \xrightarrow{\mathsf{o}}_{\mathcal{R}} \mathcal{O}(t_3) \xrightarrow{\mathsf{o}}_{\mathcal{R}} \cdots$$

Experimental Results

- consider those 434 TRSs of TPDB 4.0 where termination prover AProVE could not show termination
- one minute timeout, Intel Core 2 Duo processor, 2GB RAM
- four approaches
  - Cariboo
  - Raffelsieper Zantema transformation + AProVE
  - $\mathcal{R}'$ + AProVE
  - improved version $\mathcal{R}''$ of our transformation + AProVE

Transformation	Cariboo	RZ	$\mathcal{R}'$	$\mathcal{R}''$
outermost t/ proven	(≥ 6)	6 (160s)	7 (158s)	7 (139s)

Experimental Results

- consider those 434 TRSs of TPDB 4.0 where termination prover AProVE could not show termination
- one minute timeout, Intel Core 2 Duo processor, 2GB RAM
- four approaches
  - Cariboo
  - Raffelsieper Zantema transformation + AProVE
  - $\mathcal{R}'$ + AProVE
  - improved version $\mathcal{R}''$ of our transformation + AProVE

Transformation	Cariboo	RZ	$\mathcal{R}'$	$\mathcal{R}''$
outermost t. proven	(≥ 6)	6 (160s)	7 (158s)	7 (139s)

Experimental Results

- consider those 434 TRSs of TPDB 4.0 where termination prover AProVE could not show termination
- one minute timeout, Intel Core 2 Duo processor, 2GB RAM
- four approaches
  - Cariboo
  - Raffelsieper Zantema transformation + AProVE
  - $\mathcal{R}'$ + AProVE
  - improved version $\mathcal{R}''$ of our transformation + AProVE

Transformation	Cariboo	RZ	$\mathcal{R}'$	$\mathcal{R}''$
outermost t. proven	(≥ 6)	6 (160s)	7 (158s)	7 (139s)

Experimental Results

- consider those 434 TRSs of TPDB 4.0 where termination prover AProVE could not show termination (162 non-terminating)
- one minute timeout, Intel Core 2 Duo processor, 2GB RAM
- four approaches
  - Cariboo
  - Raffelsieper Zantema transformation + AProVE
  - $\mathcal{R}'$ + AProVE
  - improved version $\mathcal{R}''$ of our transformation + AProVE

Transformation	Cariboo	RZ	$\mathcal{R}'$	$\mathcal{R}''$
outermost t. proven	(≥ 6)	6 (160s)	7 (158s)	7 (139s)
outermost t. disproven	–	–	39 (440s)	40 (379s)

Experimental Results

- consider those 434 TRSs of TPDB 4.0 where termination prover AProVE could not show termination (162 non-terminating)
- one minute timeout, Intel Core 2 Duo processor, 2GB RAM
- four approaches
  - Cariboo
  - Raffelsieper Zantema transformation + AProVE
  - $\mathcal{R}'$ + AProVE
  - improved version $\mathcal{R}''$ of our transformation + AProVE

Transformation	Cariboo	RZ	$\mathcal{R}'$	$\mathcal{R}''$
outermost t. proven	(≥ 6)	6 (160s)	7 (158s)	7 (139s)
outermost t. disproven	–	–	39 (440s)	40 (379s)

first method to automatically disprove outermost termination

Improved Transformation $\mathcal{R}''$

special treatment for constructors C and constants

$$\mathsf{reduce}(f(x_1,\ldots,x_n)) \to \mathsf{check}_f(\mathsf{redex}_f(x_1,\ldots,x_n)) \quad \text{for all } f \in \Sigma$$

$$\mathsf{check}_f(\mathsf{redex}_f(x_1,\ldots,x_n)) \to \mathsf{in}_{f,i}(x_1,\ldots,\mathsf{reduce}(x_i),\ldots,x_n)$$
$$\text{for all } f \in \Sigma, 1 \le i \le \mathrm{arity}(f)$$

Improved Transformation $\mathcal{R}''$

special treatment for constructors C and constants

$$\mathsf{reduce}(f(x_1,\ldots,x_n)) \to \mathsf{check}_f(\mathsf{redex}_f(x_1,\ldots,x_n)) \quad \text{for all } f \notin C$$

$$\mathsf{check}_f(\mathsf{redex}_f(x_1,\ldots,x_n)) \to \mathsf{in}_{f,i}(x_1,\ldots,\mathsf{reduce}(x_i),\ldots,x_n) \quad \text{for all } f \notin C, 1 \le i \le \mathrm{arity}(f)$$

$$\mathsf{reduce}(f(x_1,\ldots,x_n)) \to \mathsf{in}_{f,i}(x_1,\ldots,\mathsf{reduce}(x_i),\ldots,x_n) \quad \text{for all } f \in C, 1 \le i \le \mathrm{arity}(f)$$

Improved Transformation $\mathcal{R}''$

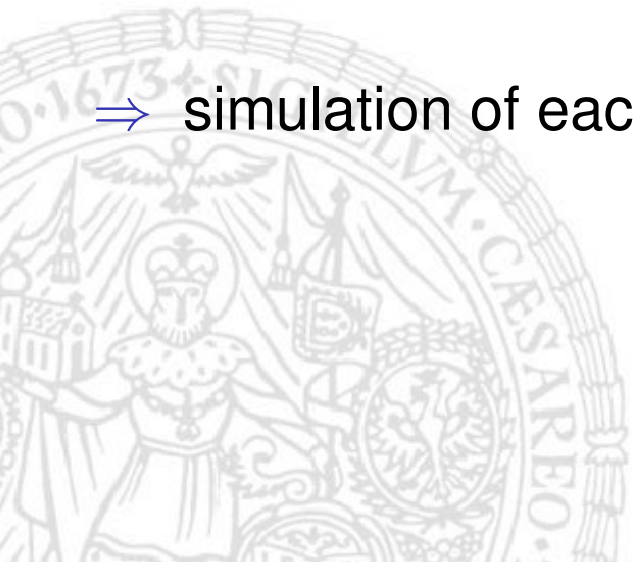
special treatment for constructors C and constants

$$\mathsf{reduce}(f(x_1,\ldots,x_n)) \to \mathsf{check}_f(\mathsf{redex}_f(x_1,\ldots,x_n)) \quad \text{for all } f \notin C$$

$$\mathsf{check}_f(\mathsf{redex}_f(x_1,\ldots,x_n)) \to \mathsf{in}_{f,i}(x_1,\ldots,\mathsf{reduce}(x_i),\ldots,x_n) \quad \text{for all } f \notin C, 1 \le i \le \mathrm{arity}(f)$$

$$\mathsf{reduce}(f(x_1,\ldots,x_n)) \to \mathsf{in}_{f,i}(x_1,\ldots,\mathsf{reduce}(x_i),\ldots,x_n) \quad \text{for all } f \in C, 1 \le i \le \mathrm{arity}(f)$$

$\Rightarrow$ simulation of each outermost step in less innermost steps



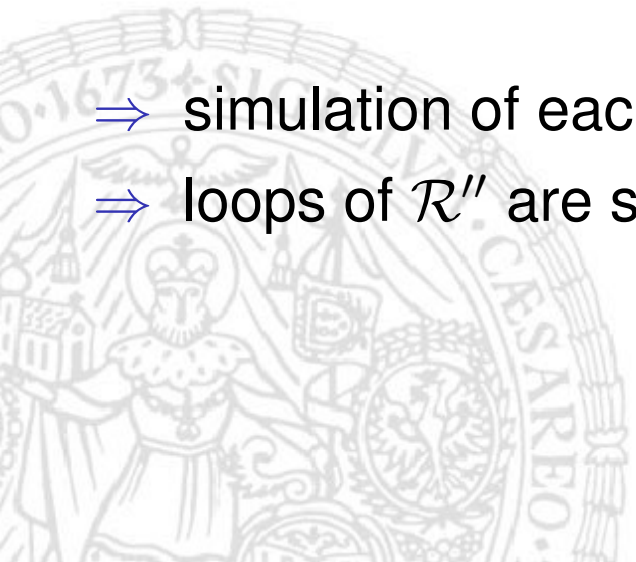
Improved Transformation $\mathcal{R}''$

special treatment for constructors C and constants

$$\mathsf{reduce}(f(x_1,\ldots,x_n)) \to \mathsf{check}_f(\mathsf{redex}_f(x_1,\ldots,x_n)) \quad \text{for all } f \notin C$$

$$\mathsf{check}_f(\mathsf{redex}_f(x_1,\ldots,x_n)) \to \mathsf{in}_{f,i}(x_1,\ldots,\mathsf{reduce}(x_i),\ldots,x_n) \quad \text{for all } f \notin C, 1 \le i \le \mathrm{arity}(f)$$

$$\mathsf{reduce}(f(x_1,\ldots,x_n)) \to \mathsf{in}_{f,i}(x_1,\ldots,\mathsf{reduce}(x_i),\ldots,x_n) \quad \text{for all } f \in C, 1 \le i \le \mathrm{arity}(f)$$



$\Rightarrow$ simulation of each outermost step in less innermost steps

$\Rightarrow$ loops of $\mathcal{R}''$ are shorter than loops of $\mathcal{R}'$

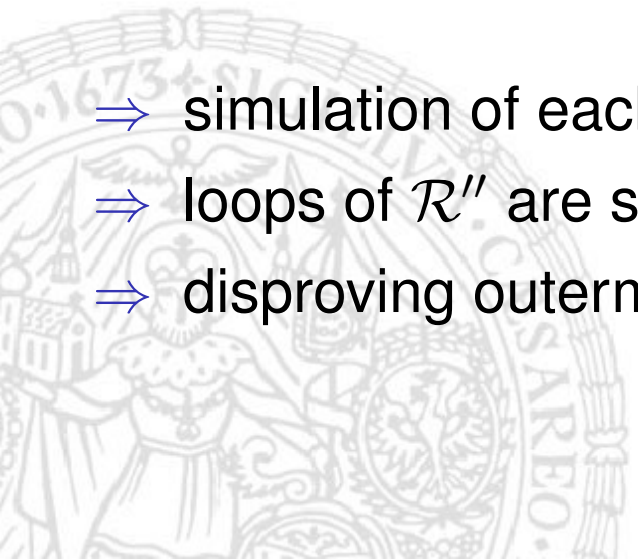
Improved Transformation $\mathcal{R}''$

special treatment for constructors C and constants

$$\mathsf{reduce}(f(x_1,\ldots,x_n)) \to \mathsf{check}_f(\mathsf{redex}_f(x_1,\ldots,x_n)) \quad \text{for all } f \notin C$$

$$\mathsf{check}_f(\mathsf{redex}_f(x_1,\ldots,x_n)) \to \mathsf{in}_{f,i}(x_1,\ldots,\mathsf{reduce}(x_i),\ldots,x_n) \quad \text{for all } f \notin C, 1 \le i \le \mathrm{arity}(f)$$

$$\mathsf{reduce}(f(x_1,\ldots,x_n)) \to \mathsf{in}_{f,i}(x_1,\ldots,\mathsf{reduce}(x_i),\ldots,x_n) \quad \text{for all } f \in C, 1 \le i \le \mathrm{arity}(f)$$



- $\Rightarrow$ simulation of each outermost step in less innermost steps
- $\Rightarrow$ loops of $\mathcal{R}''$ are shorter than loops of $\mathcal{R}'$
- $\Rightarrow$ disproving outermost termination becomes easier

Summary

- complete transformations such that

  $\mathcal{R}$ is outermost terminating iff $\mathcal{R}'$ or $\mathcal{R}''$ is innermost terminating
- first automatic approach to disprove outermost termination

Future own work

- not all outermost loops of $\mathcal{R}$ are transformed to innermost loops of $\mathcal{R}'$ or $\mathcal{R}''$

  $\rightarrow$ develop direct method which can detect all outermost loops

Future related work

- Raffelsieper Zantema: incompleteness transformation
- Endrullis: transformation to context-sensitive termination

Summary

- complete transformations such that

  $\mathcal{R}$ is outermost terminating iff $\mathcal{R}'$ or $\mathcal{R}''$ is innermost terminating
- first automatic approach to disprove outermost termination

Future own work

- not all outermost loops of $\mathcal{R}$ are transformed to innermost loops of $\mathcal{R}'$ or $\mathcal{R}''$

$\Rightarrow$ develop direct method which can detect all outermost loops

Future related work

- Raffelsieper Zantema: incompleteness transformation
- Endrullis: transformation to context-sensitive termination

Summary

- complete transformations such that

  $\mathcal{R}$ is outermost terminating iff $\mathcal{R}'$ or $\mathcal{R}''$ is innermost terminating
- first automatic approach to disprove outermost termination

Future own work

- not all outermost loops of $\mathcal{R}$ are transformed to innermost loops of $\mathcal{R}'$ or $\mathcal{R}''$

$\Rightarrow$ develop direct method which can detect all outermost loops

Future related work

- Raffelsieper Zantema: incompleteness transformation
- Endrullis: transformation to context-sensitive termination