Böhm Reduction in Infinitary Term Graph Rewriting Systems

Patrick Bahr

IT University of Copenhagen

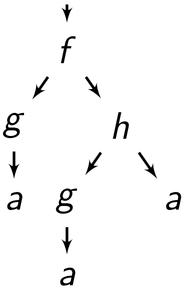
Overview

1. Motivation
   - Why term graphs?
   - Why infinitary term graph rewriting?
   - Why Böhm reduction?
2. Böhm Reduction on Terms
3. Böhm Reduction on Term Graphs

From Terms to Term Graphs

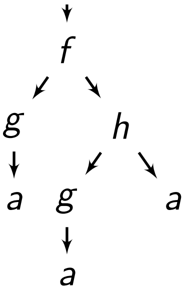
$$f(g(a), h(g(a), a))$$

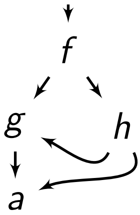
From Terms to Term Graphs



$f(g(a), h(g(a), a))$

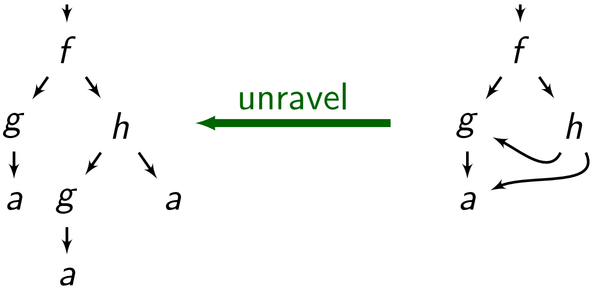
From Terms to Term Graphs





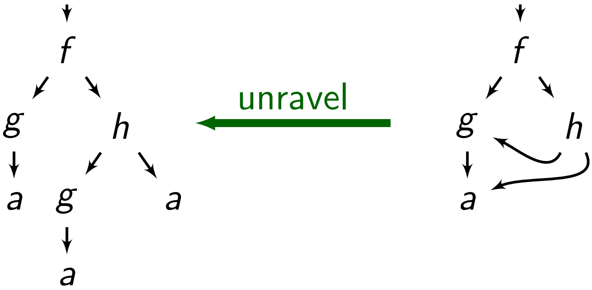
$$f(g(a), h(g(a), a))$$

From Terms to Term Graphs



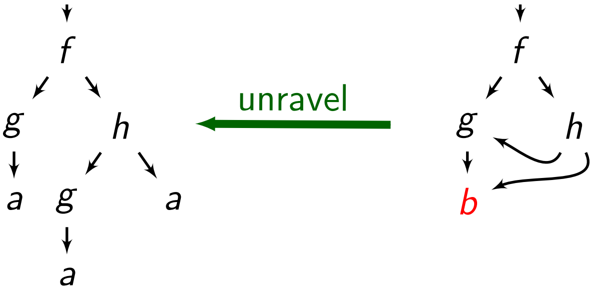
$f(g(a), h(g(a), a))$

From Terms to Term Graphs



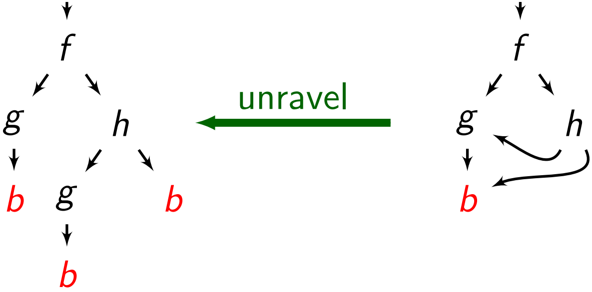
$$a \to b$$

From Terms to Term Graphs



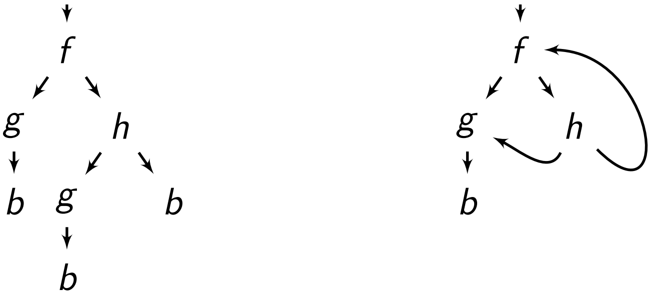
$a \to b$

From Terms to Term Graphs

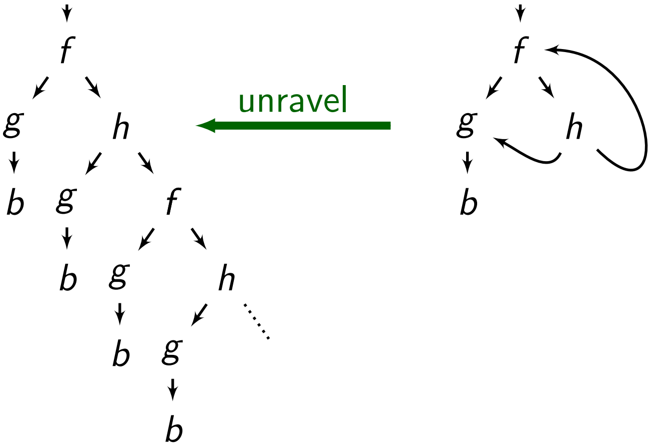


$a \to b$

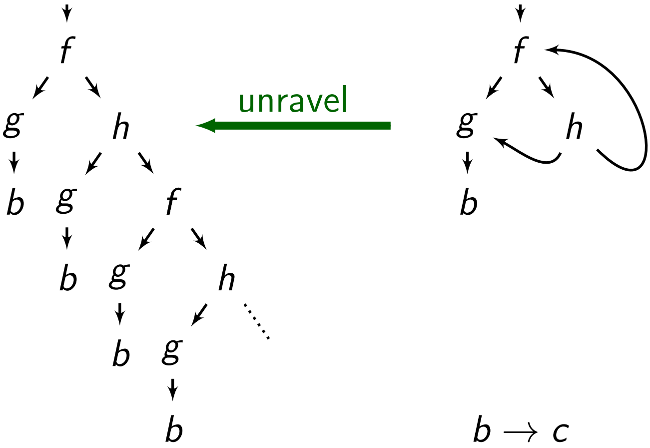
From Terms to Term Graphs



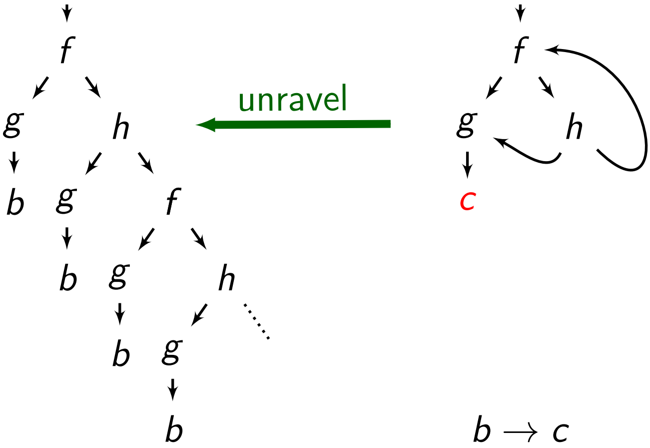
From Terms to Term Graphs



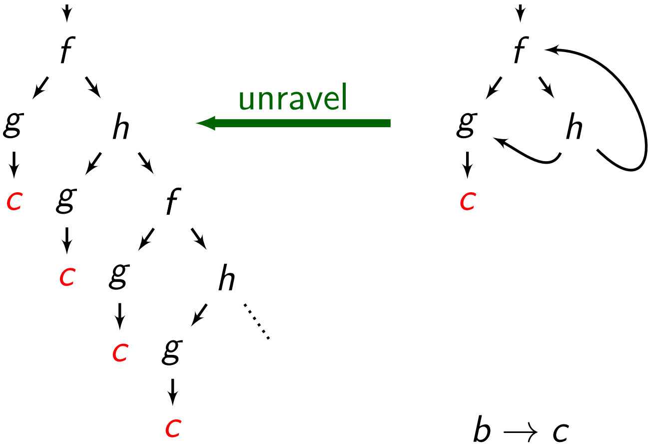
From Terms to Term Graphs



From Terms to Term Graphs



From Terms to Term Graphs



$b \to c$

Soundness & Completeness

Soundness of finite reductions

For every left-linear, left-finite GRS $\mathcal{R}$ we have

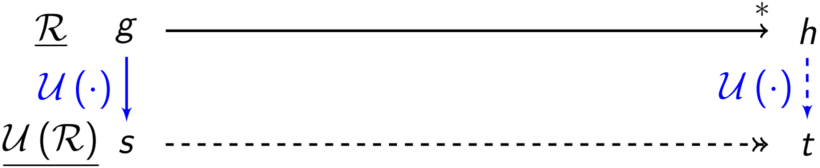
$$\begin{array}{ccc} \underline{\mathcal{R}} & g & \xrightarrow{\qquad\qquad\qquad *\qquad\qquad\qquad} & h \\ \mathcal{U}(\cdot) & \big\downarrow & & \\ \underline{\mathcal{U}(\mathcal{R})} & s & & \end{array}$$

[1] R. Kennaway et al. "On the adequacy of graph rewriting for simulating term rewriting". In: *ACM Transactions on Programming Languages and Systems* (1994).

Soundness & Completeness

Soundness of finite reductions

For every left-linear, left-finite GRS $\mathcal{R}$ we have

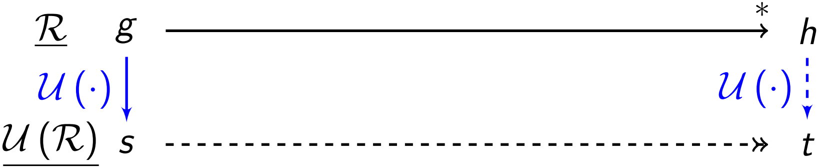


[1] R. Kennaway et al. "On the adequacy of graph rewriting for simulating term rewriting". In: *ACM Transactions on Programming Languages and Systems* (1994).

Soundness & Completeness

Soundness of finite reductions

For every left-linear, left-finite GRS $\mathcal{R}$ we have



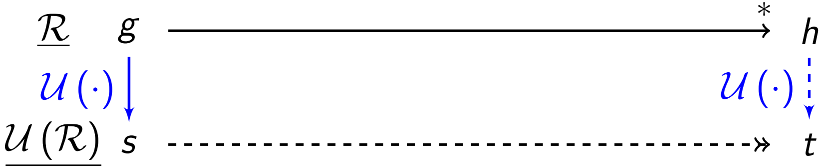
Completeness property

[1] R. Kennaway et al. "On the adequacy of graph rewriting for simulating term rewriting". In: *ACM Transactions on Programming Languages and Systems* (1994).

Soundness & Completeness

Soundness of finite reductions

For every left-linear, left-finite GRS $\mathcal{R}$ we have



Completeness property

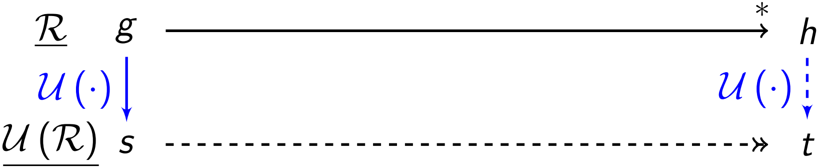
---

[1] R. Kennaway et al. "On the adequacy of graph rewriting for simulating term rewriting". In: *ACM Transactions on Programming Languages and Systems* (1994).

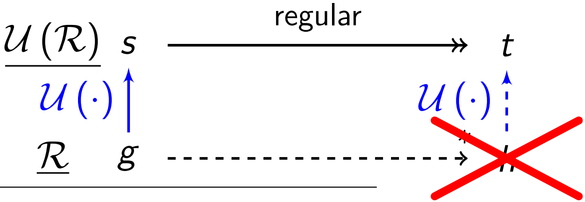
Soundness & Completeness

Soundness of finite reductions

For every left-linear, left-finite GRS $\mathcal{R}$ we have



Completeness property

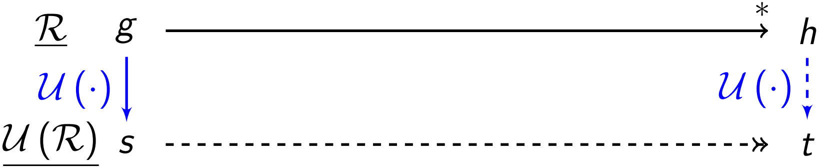


[1]R. Kennaway et al. "On the adequacy of graph rewriting for simulating term rewriting". In: *ACM Transactions on Programming Languages and Systems* (1994).

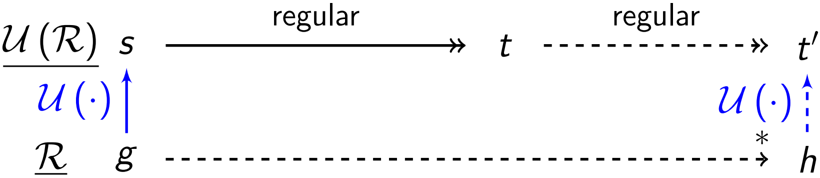
Soundness & Completeness

Soundness of finite reductions

For every left-linear, left-finite GRS $\mathcal{R}$ we have



Completeness property



[1] R. Kennaway et al. "On the adequacy of graph rewriting for simulating term rewriting". In: *ACM Transactions on Programming Languages and Systems* (1994).

Infinitary Graph Rewriting – Motivation

- A common formalism
  - study correspondences between infinitary TRSs and finitary GRSs

[2] Z. M. Ariola and S. Blom. "Skew confluence and the lambda calculus with letrec". In: *Annals of Pure and Applied Logic* (2002).

[3] C. Grabmayer and J. Rochel. "Maximal Sharing in the Lambda Calculus with Letrec". In: *ICFP*. 2014.

Infinitary Graph Rewriting – Motivation

- A common formalism
  - study correspondences between infinitary TRSs and finitary GRSs
- Lazy evaluation
  - infinitary term rewriting only covers non-strictness
  - however: lazy evaluation = non-strictness + sharing

---

[2] Z. M. Ariola and S. Blom. "Skew confluence and the lambda calculus with letrec". In: *Annals of Pure and Applied Logic* (2002).

[3] C. Grabmayer and J. Rochel. "Maximal Sharing in the Lambda Calculus with Letrec". In: *ICFP*. 2014.

Infinitary Graph Rewriting – Motivation

- A common formalism
  - study correspondences between infinitary TRSs and finitary GRSs
- Lazy evaluation
  - infinitary term rewriting only covers non-strictness
  - however: lazy evaluation = non-strictness + sharing
- lambda calculi with letrec[2,3]
  - these calculi are non-confluent
  - but there is a notion of infinite normal forms

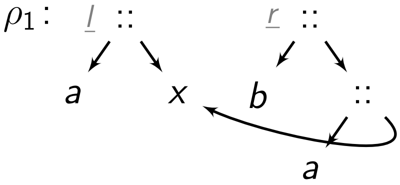
---

[2] Z. M. Ariola and S. Blom. "Skew confluence and the lambda calculus with letrec". In: *Annals of Pure and Applied Logic* (2002).

[3] C. Grabmayer and J. Rochel. "Maximal Sharing in the Lambda Calculus with Letrec". In: *ICFP*. 2014.

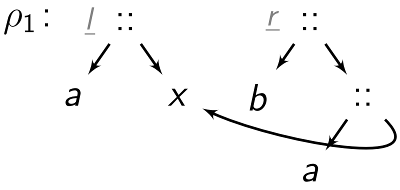
Example: Cyclic Sharing

Term graph rules for $a :: x \to b :: a :: x$

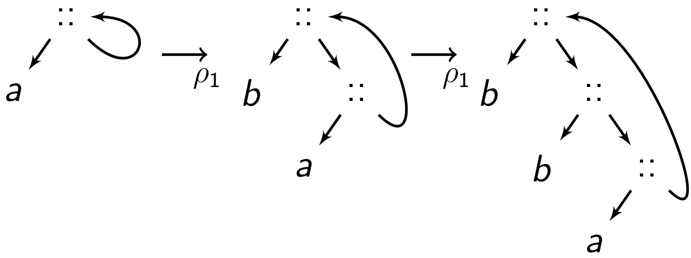


Example: Cyclic Sharing

Term graph rules for $a :: x \to b :: a :: x$

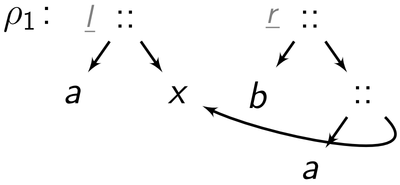


Reductions:

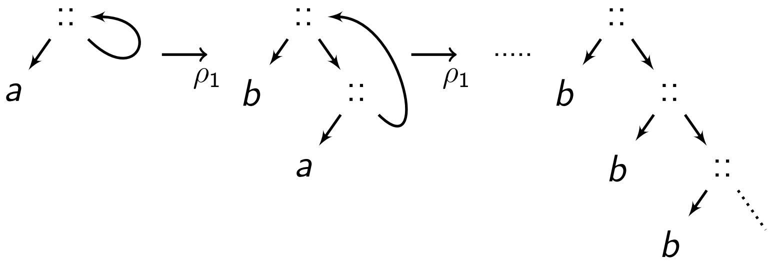


Example: Cyclic Sharing

Term graph rules for $a :: x \to b :: a :: x$

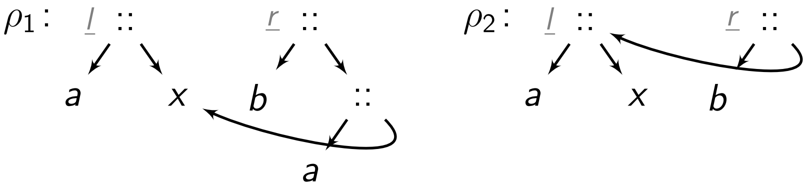


Reductions:

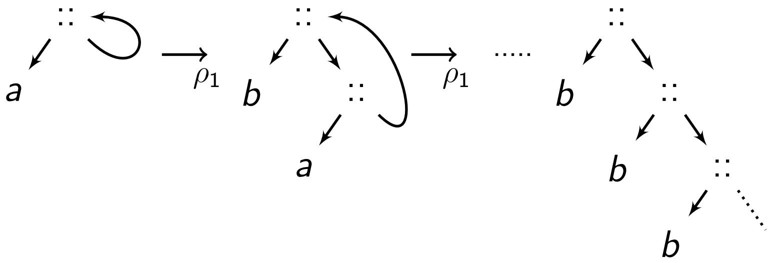


Example: Cyclic Sharing

Term graph rules for $a :: x \to b :: a :: x$

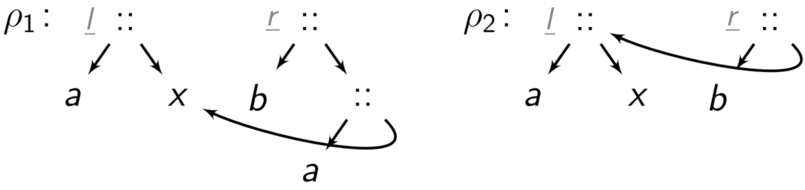


Reductions:

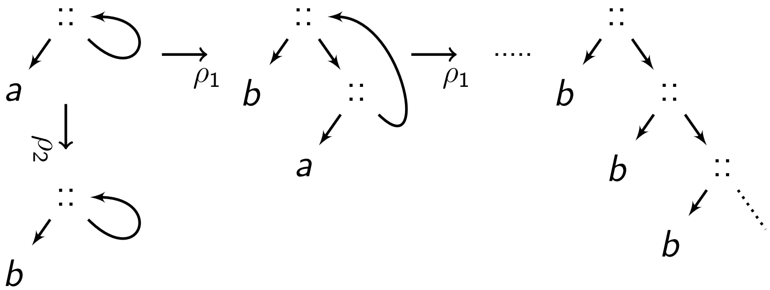


Example: Cyclic Sharing

Term graph rules for $a :: x \to b :: a :: x$



Reductions:



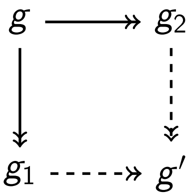
Problems of Infintary Graph Rewriting

Confluence of Orthogonal Systems

$$\begin{array}{ccc} g & \twoheadrightarrow & g_2 \\ \downarrow\!\!\downarrow & & \\ g_1 & & \end{array}$$

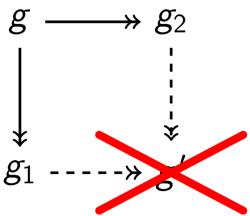
Problems of Infintary Graph Rewriting

Confluence of Orthogonal Systems



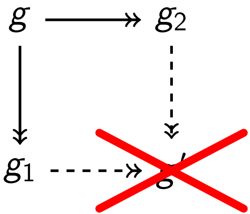
Problems of Infintary Graph Rewriting

Confluence of Orthogonal Systems

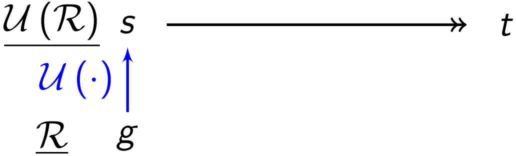


Problems of Infintary Graph Rewriting

Confluence of Orthogonal Systems

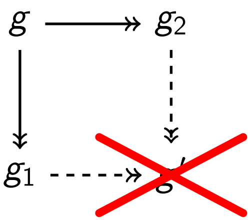


Completeness

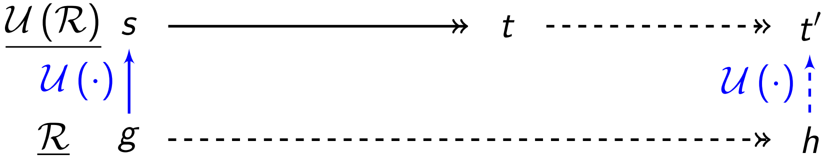


Problems of Infintary Graph Rewriting

Confluence of Orthogonal Systems

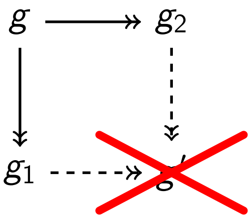


Completeness

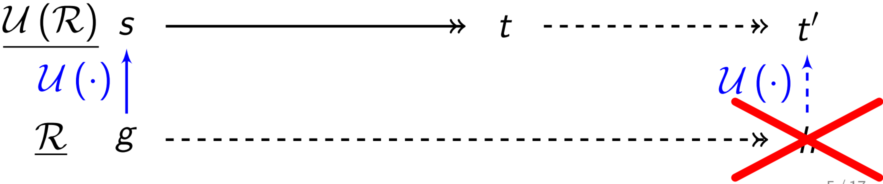


Problems of Infintary Graph Rewriting

Confluence of Orthogonal Systems



Completeness



This paper

Study two techniques to solve these problems

- Böhm reduction
- partial order infinitary rewriting

[4] R. Kennaway, V. van Oostrom, and F.-J. de Vries. "Meaningless Terms in Rewriting". In: *J. Funct. Logic Programming* (1999).

[5] B. "Partial Order Infinitary Term Rewriting". In: *LMCS* (2014).

[6] B. "Infinitary Term Graph Rewriting is Simple, Sound and Complete". In: *RTA*. 2012.

This paper

Study two techniques to solve these problems

- Böhm reduction
- partial order infinitary rewriting

In previous work

- both yield confluence for infinitary term rewriting[4,5]
- partial order approach yields completeness property for infinitary term graph rewriting[6]

[4] R. Kennaway, V. van Oostrom, and F.-J. de Vries. "Meaningless Terms in Rewriting". In: *J. Funct. Logic Programming* (1999).

[5] B. "Partial Order Infinitary Term Rewriting". In: *LMCS* (2014).

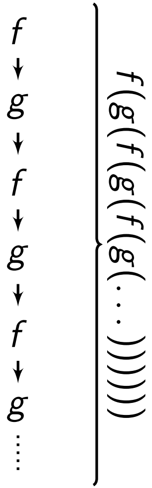
[6] B. "Infinitary Term Graph Rewriting is Simple, Sound and Complete". In: *RTA*. 2012.

Infinitary Term Rewriting

Failure of Infinitary Confluence

for Orthogonal Term Rewriting Systems

$$f(x) \to x \qquad g(x) \to x$$



Failure of Infinitary Confluence

for Orthogonal Term Rewriting Systems

$$f(x) \to x \qquad\qquad g(x) \to x$$

Failure of Infinitary Confluence

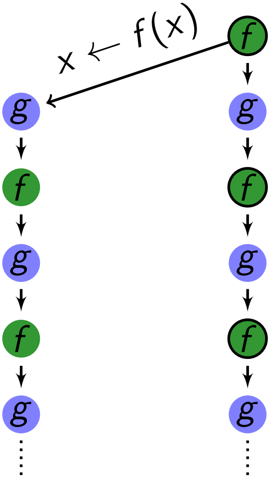
for Orthogonal Term Rewriting Systems

$$f(x) \to x \qquad\qquad g(x) \to x$$

Failure of Infinitary Confluence

for Orthogonal Term Rewriting Systems

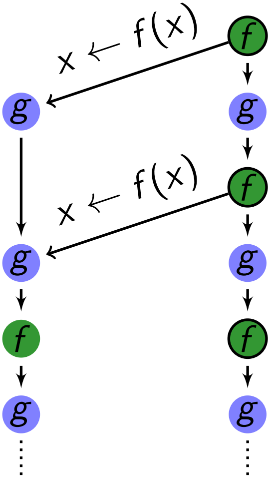
$$f(x) \to x \qquad g(x) \to x$$



Failure of Infinitary Confluence

for Orthogonal Term Rewriting Systems

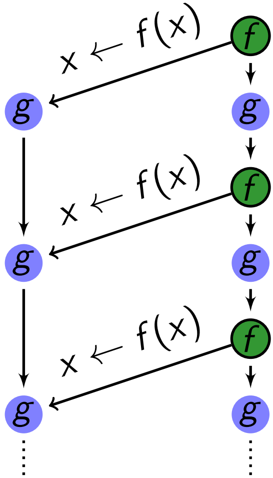
$f(x) \to x \qquad g(x) \to x$



Failure of Infinitary Confluence

for Orthogonal Term Rewriting Systems

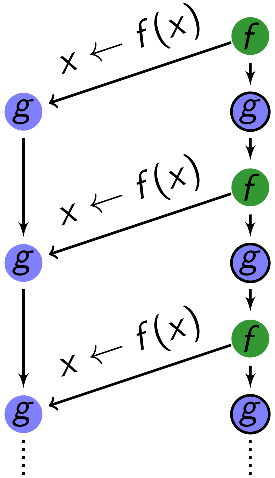
$$f(x) \to x \qquad g(x) \to x$$



Failure of Infinitary Confluence

for Orthogonal Term Rewriting Systems

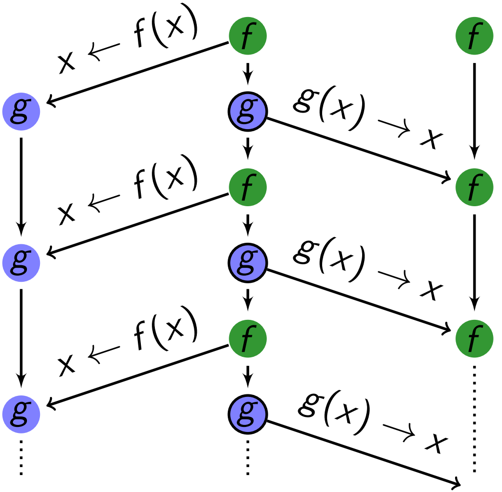
$f(x) \to x$ $\qquad$ $g(x) \to x$



Failure of Infinitary Confluence

for Orthogonal Term Rewriting Systems

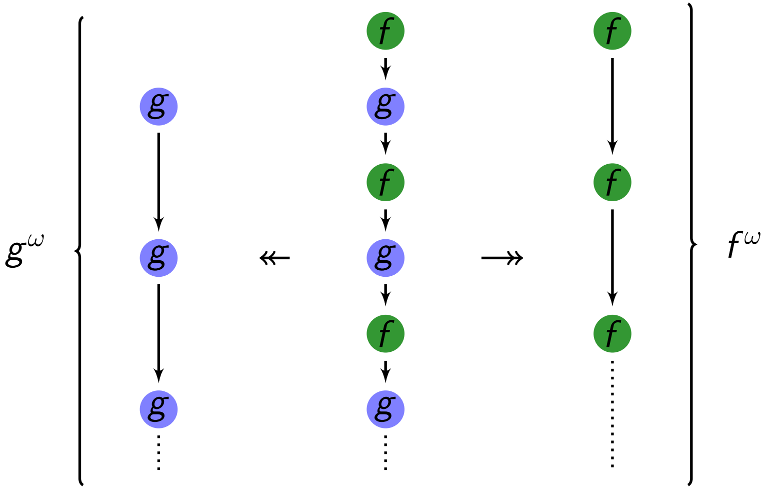
$f(x) \to x$ $\qquad$ $g(x) \to x$



Failure of Infinitary Confluence

for Orthogonal Term Rewriting Systems

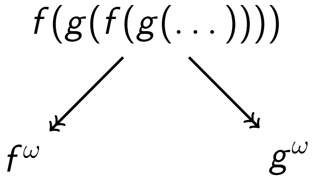
$f(x) \to x \qquad g(x) \to x$



Böhm Reduction

Idea

- terms like f^ω and g^ω are considered meaningless
- for each meaningless term t, add rule $t \to \perp$
- meaningless terms are characterised by a set of axioms



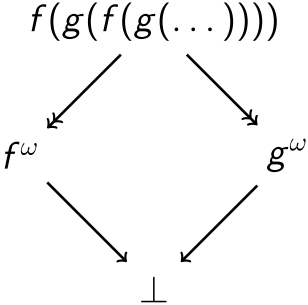
[7] R. Kennaway et al. "Infinitary lambda calculus". In: *Theoretical Computer Science* (1997).

[8] R. Kennaway, V. van Oostrom, and F.-J. de Vries. "Meaningless Terms in Rewriting". In: *J. Funct. Logic Programming* (1999).

Böhm Reduction

Idea

- terms like f^ω and g^ω are considered meaningless
- for each meaningless term t, add rule $t \to \bot$
- meaningless terms are characterised by a set of axioms



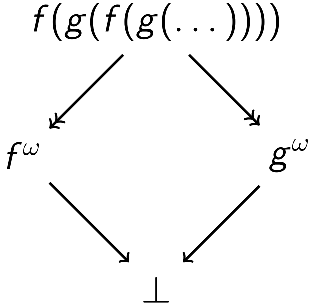
[7] R. Kennaway et al. "Infinitary lambda calculus". In: *Theoretical Computer Science* (1997).

[8] R. Kennaway, V. van Oostrom, and F.-J. de Vries. "Meaningless Terms in Rewriting". In: *J. Funct. Logic Programming* (1999).

Böhm Reduction

Idea

- terms like f^ω and g^ω are considered meaningless
- for each meaningless term t, add rule $t \to \bot$
- meaningless terms are characterised by a set of axioms



Böhm reduction = infinitary rewriting with $\bot$-rules

[7] R. Kennaway et al. "Infinitary lambda calculus". In: *Theoretical Computer Science* (1997).

[8] R. Kennaway, V. van Oostrom, and F.-J. de Vries. "Meaningless Terms in Rewriting". In: *J. Funct. Logic Programming* (1999).

Partial Order Infinitary Rewriting

- Alternative characterisation of Böhm reduction
- Changes the notion of convergence instead of adding rules
  (uses a partial order instead of a metric)

[9] B. “Partial Order Infinitary Term Rewriting”. In: *LMCS* (2014).

Partial Order Infinitary Rewriting

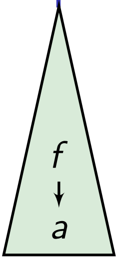
- Alternative characterisation of Böhm reduction
- Changes the notion of convergence instead of adding rules
  (uses a partial order instead of a metric)

The Good & The Bad

+ less ad hoc
+ no need for infinitely many reduction rules
- captures only a particular set of meaningless terms (namely: root-active terms)

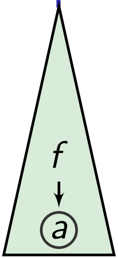
[9] B. "Partial Order Infinitary Term Rewriting". In: *LMCS* (2014).

Example: Convergence of a Reduction



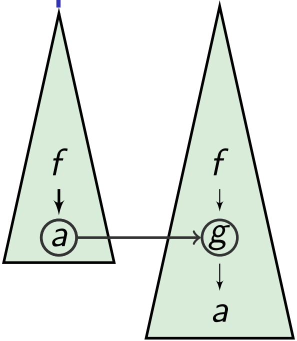
$$\mathcal{R} = \{a \to g(a)\}$$

Example: Convergence of a Reduction



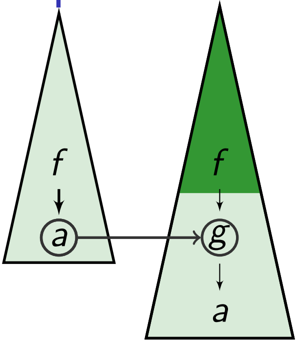
$$\mathcal{R} = \{a \to g(a)\}$$

Example: Convergence of a Reduction



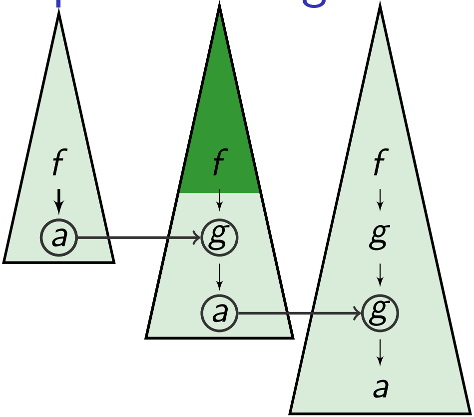
$$\mathcal{R} = \{a \to g(a)\}$$

Example: Convergence of a Reduction



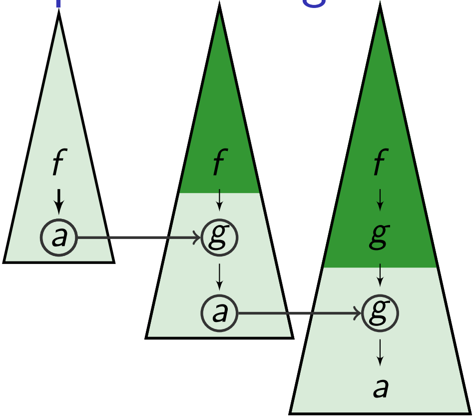
$$\mathcal{R} = \{a \to g(a)\}$$

Example: Convergence of a Reduction



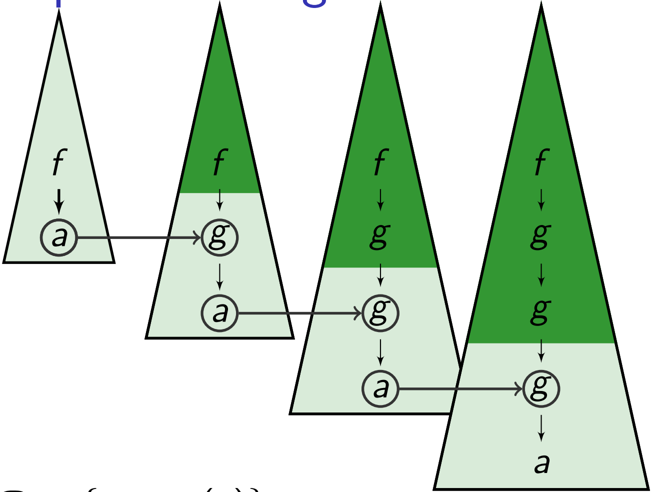
$$\mathcal{R} = \{a \to g(a)\}$$

Example: Convergence of a Reduction



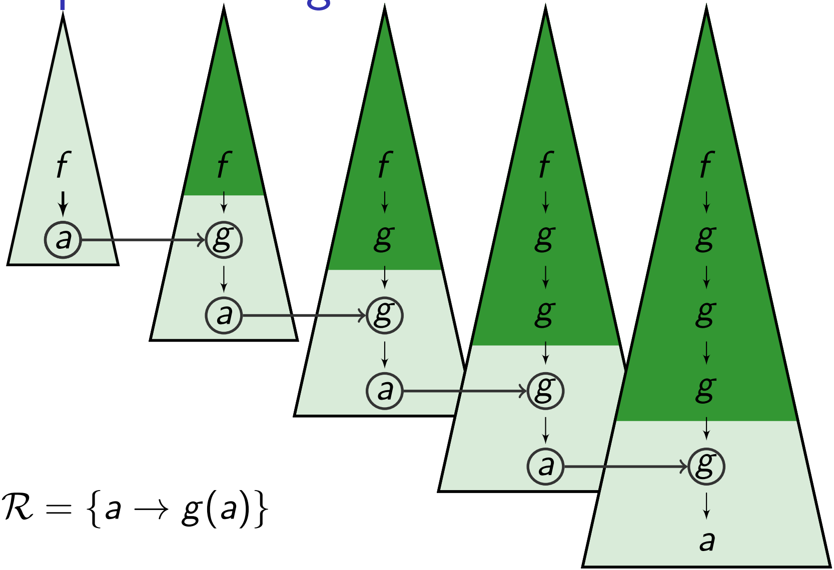
$$\mathcal{R} = \{a \to g(a)\}$$

Example: Convergence of a Reduction



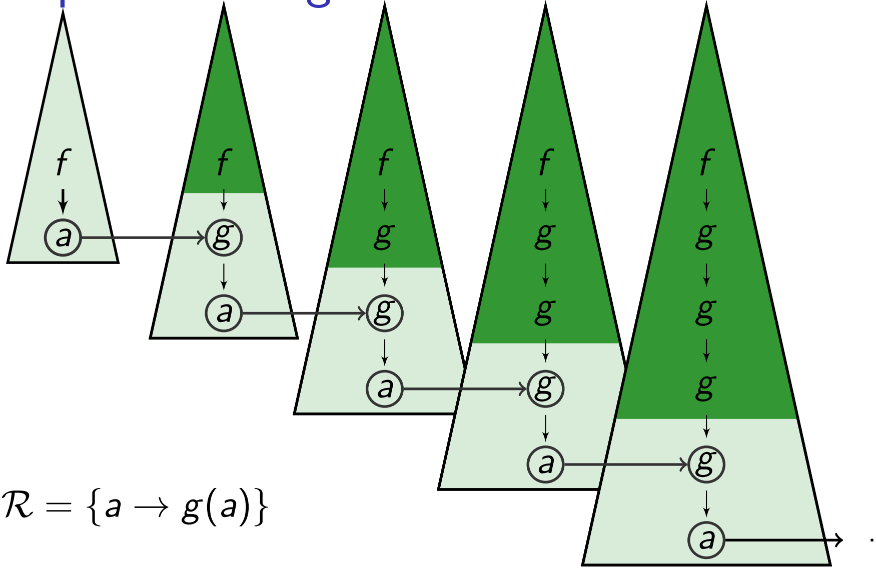
$\mathcal{R} = \{a \to g(a)\}$

Example: Convergence of a Reduction



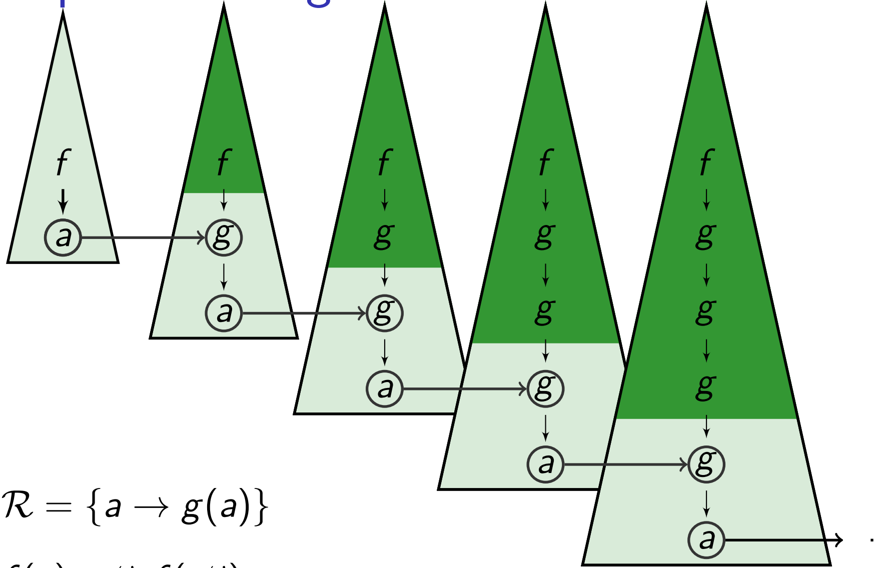
$\mathcal{R} = \{a \to g(a)\}$

Example: Convergence of a Reduction



$$\mathcal{R} = \{a \to g(a)\}$$

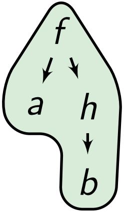
Example: Convergence of a Reduction



$\mathcal{R} = \{a \to g(a)\}$

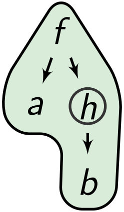
$f(a) \twoheadrightarrow^{\omega}_{\mathcal{R}} f(g^{\omega})$

Example: Non-Convergence



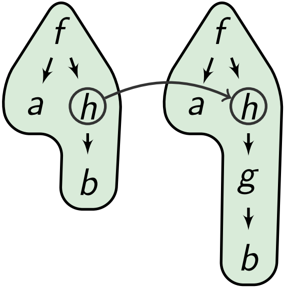
$$\mathcal{R} = \begin{cases} a \to g(a) \\ h(x) \to h(g(x)) \end{cases}$$

Example: Non-Convergence



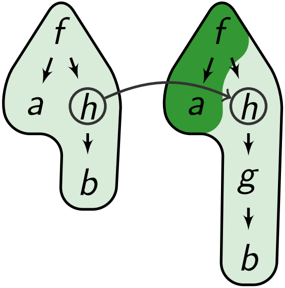
$$\mathcal{R} = \begin{cases} a \to g(a) \\ h(x) \to h(g(x)) \end{cases}$$

Example: Non-Convergence



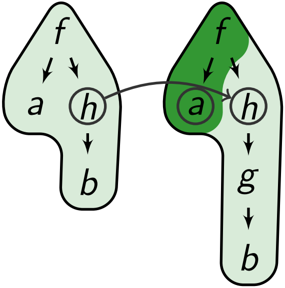
$$\mathcal{R} = \begin{cases} a \to g(a) \\ h(x) \to h(g(x)) \end{cases}$$

Example: Non-Convergence



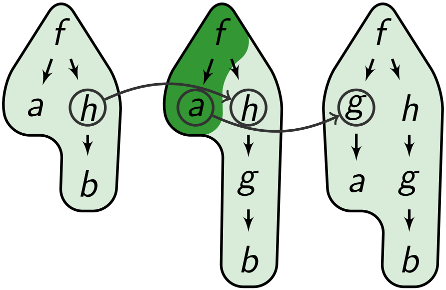
$$\mathcal{R} = \begin{cases} a \to g(a) \\ h(x) \to h(g(x)) \end{cases}$$

Example: Non-Convergence



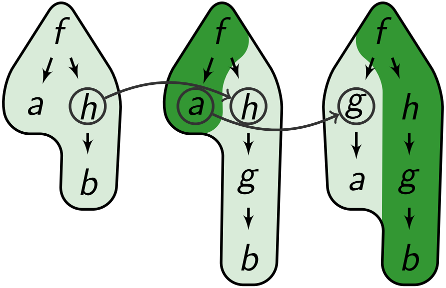
$$\mathcal{R} = \begin{cases} a \to g(a) \\ h(x) \to h(g(x)) \end{cases}$$

Example: Non-Convergence



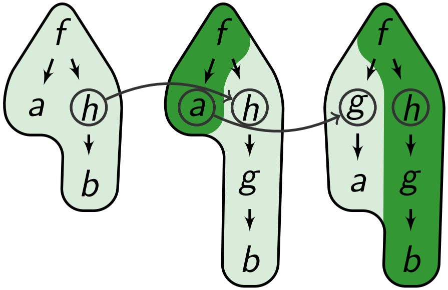
$$\mathcal{R} = \begin{cases} a \to g(a) \\ h(x) \to h(g(x)) \end{cases}$$

Example: Non-Convergence



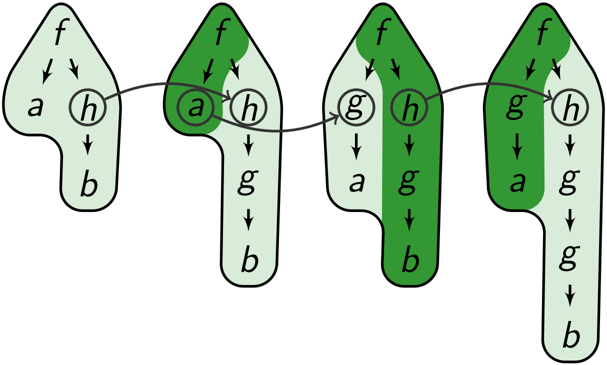
$$\mathcal{R} = \begin{cases} a \to g(a) \\ h(x) \to h(g(x)) \end{cases}$$

Example: Non-Convergence



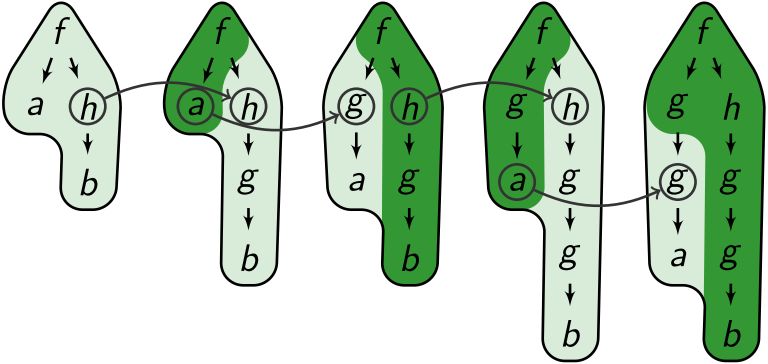
$$\mathcal{R} = \begin{cases} a \to g(a) \\ h(x) \to h(g(x)) \end{cases}$$

Example: Non-Convergence



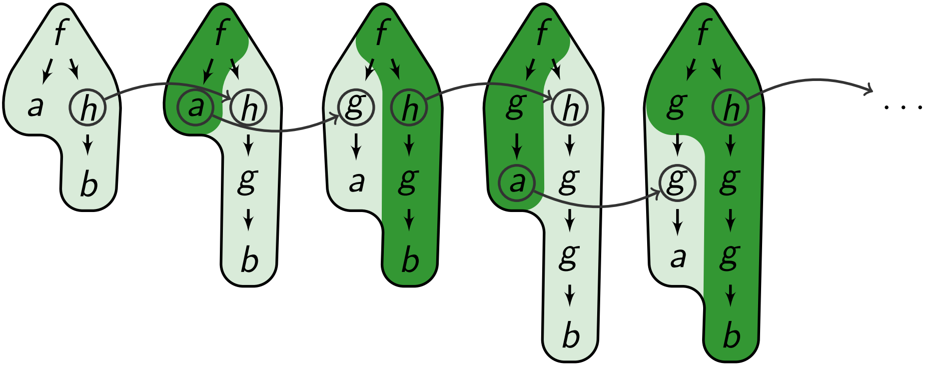
$$\mathcal{R} = \begin{cases} a \to g(a) \\ h(x) \to h(g(x)) \end{cases}$$

Example: Non-Convergence



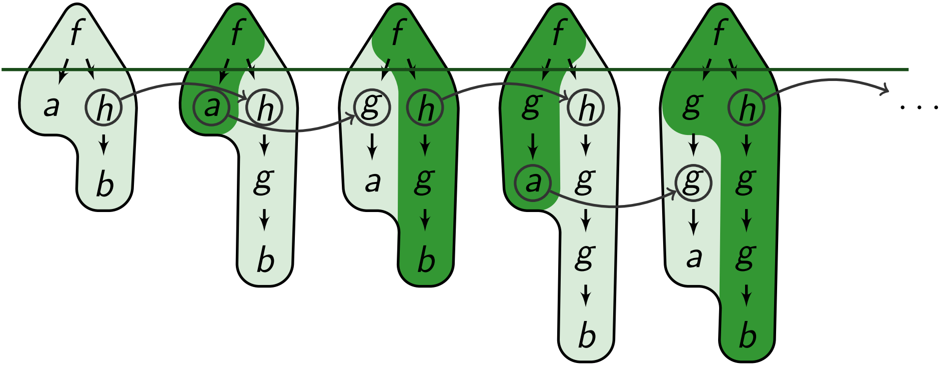
$$\mathcal{R} = \begin{cases} a \to g(a) \\ h(x) \to h(g(x)) \end{cases}$$

Example: Non-Convergence



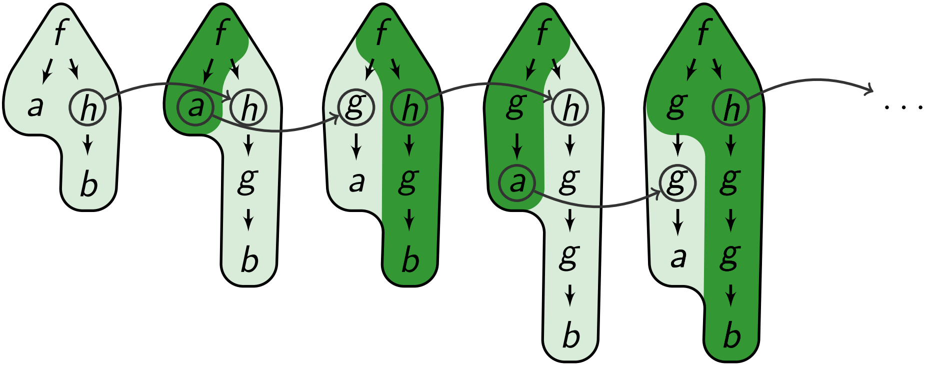
$$\mathcal{R} = \begin{cases} a \to g(a) \\ h(x) \to h(g(x)) \end{cases}$$

Example: Non-Convergence

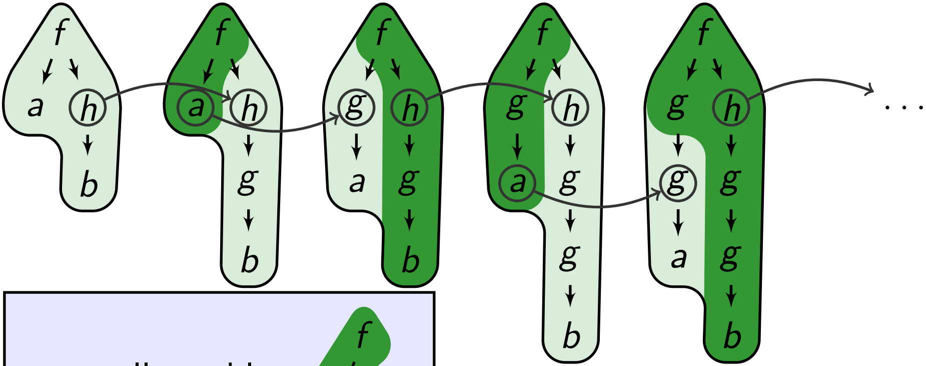


$$\mathcal{R} = \begin{cases} a \to g(a) \\ h(x) \to h(g(x)) \end{cases}$$

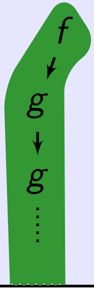
Partial Order Convergence



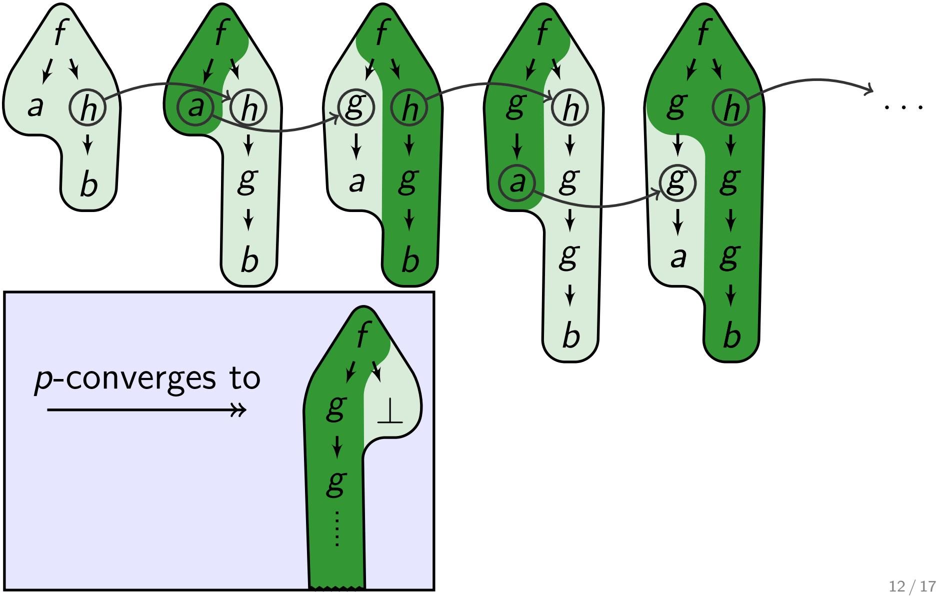
Partial Order Convergence



eventually stable:



Partial Order Convergence



Properties of Orthogonal TRS

property	metric	Böhm red.
compression	✔	✔
inf. strip lemma	✔	✔
developments	✘	✔
inf. confluence	✘	✔
inf. normalisation	✘	✔

Properties of Orthogonal TRS

property	**metric**	**Böhm red.**	**part. order**
compression	✔	✔	✔
inf. strip lemma	✔	✔	✔
developments	✘	✔	✔
inf. confluence	✘	✔	✔
inf. normalisation	✘	✔	✔

Properties of Orthogonal TRS

property	**metric**	**Böhm red.**	**part. order**
compression	✔	✔	✔
inf. strip lemma	✔	✔	✔
developments	✘	✔	✔
inf. confluence	✘	✔	✔
inf. normalisation	✘	✔	✔

Theorem

If $\mathcal{R}$ is an orthogonal TRS and $\mathcal{B}$ the Böhm extension of $\mathcal{R}$ (w.r.t. root-active terms), then

$$s \xrightarrow{p}\!\!\!\!\twoheadrightarrow_{\mathcal{R}} t \quad \textit{iff} \quad s \xrightarrow{m}\!\!\!\!\twoheadrightarrow_{\mathcal{B}} t.$$

Term Graph Rewriting

Properties of Orthogonal GRS

property	metric	Böhm red.	part. order
compression	✔	?	✔
inf. strip lemma	✔	✔	✔
developments	✘	✔	✔
inf. normalisation	✘	✔	✔
inf. confluence	✘	?	?

Properties of Orthogonal GRS

property	**metric**	**Böhm red.**	**part. order**
compression	✔	?	✔
inf. strip lemma	✔	✔	✔
developments	✘	✔	✔
inf. normalisation	✘	✔	✔
inf. confluence	✘	?	?
inf. confluence modulo bisim.	✘	✔	✔

Properties of Orthogonal GRS

property	metric	Böhm red.	part. order
compression	✔	?	✔
inf. strip lemma	✔	✔	✔
developments	✘	✔	✔
inf. normalisation	✘	✔	✔
inf. confluence	✘	?	?
inf. confluence modulo bisim.	✘	✔	✔

Theorem

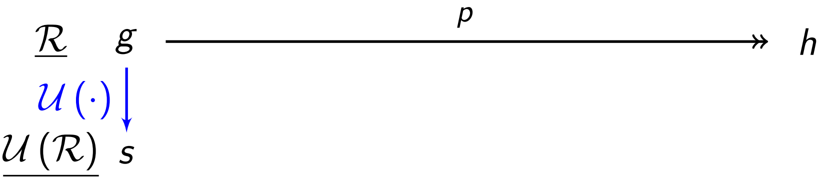
If $\mathcal{R}$ is an orth. GRS and $\mathcal{B}$ the Böhm extension of $\mathcal{R}$ (w.r.t. root-active term graphs), then

$$g \xrightarrow{p}\!\!\!\!\rightarrow_{\mathcal{R}} h \quad \text{iff} \quad g \xrightarrow{m}\!\!\!\!\rightarrow_{\mathcal{B}} h.$$

Soundness & Completeness

Soundness of metric convergence

For every left-linear, left-finite GRS $\mathcal{R}$ we have

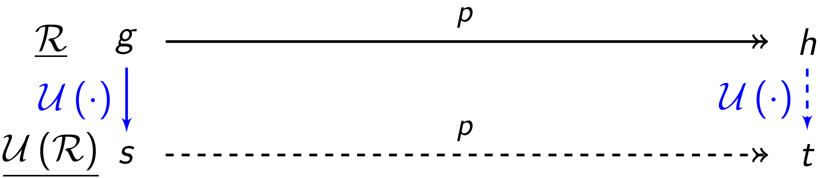


[10] B. "Infinitary Term Graph Rewriting is Simple, Sound and Complete". In: *RTA*. 2012.

Soundness & Completeness

Soundness of metric convergence

For every left-linear, left-finite GRS $\mathcal{R}$ we have

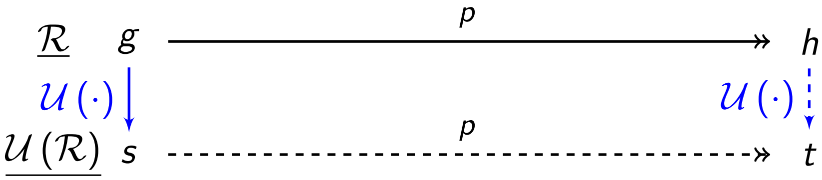


[10] B. "Infinitary Term Graph Rewriting is Simple, Sound and Complete". In: *RTA*. 2012.

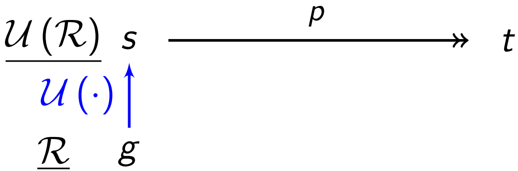
Soundness & Completeness

Soundness of metric convergence

For every left-linear, left-finite GRS $\mathcal{R}$ we have



Completeness property



[10] B. "Infinitary Term Graph Rewriting is Simple, Sound and Complete". In: *RTA*. 2012.

Soundness & Completeness

Soundness of metric convergence

For every left-linear, left-finite GRS $\mathcal{R}$ we have

$$
\begin{array}{rcccc}
\underline{\mathcal{R}} & g & \xrightarrow{\quad\quad p \quad\quad} \!\!\!\!\!\rightarrow & h \\
\mathcal{U}(\cdot) & \big\downarrow & & \mathcal{U}(\cdot) \ \vdots\!\downarrow \\
\underline{\mathcal{U}(\mathcal{R})} & s & \stackrel{p}{\dashrightarrow}\!\!\!\!\rightarrow & t
\end{array}
$$

Completeness property

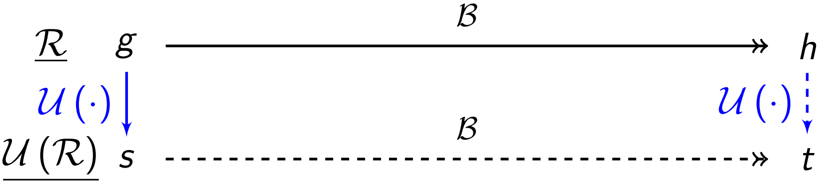
$$
\begin{array}{rccccc}
\underline{\mathcal{U}(\mathcal{R})} & s & \xrightarrow{\quad p \quad} \!\!\!\!\!\rightarrow & t & \stackrel{p}{\dashrightarrow}\!\!\!\!\rightarrow & t' \\
\mathcal{U}(\cdot) & \big\uparrow & & & \mathcal{U}(\cdot) & \vdots\!\uparrow \\
\underline{\mathcal{R}} & g & & \stackrel{p}{\dashrightarrow}\!\!\!\!\rightarrow & & h
\end{array}
$$

[10] B. "Infinitary Term Graph Rewriting is Simple, Sound and Complete". In: *RTA*. 2012.

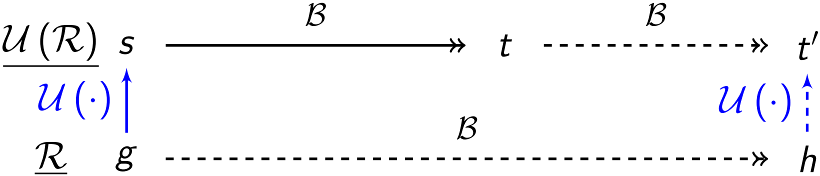
Soundness & Completeness

Soundness of metric convergence

For every left-linear, left-finite GRS $\mathcal{R}$ we have



Completeness property



[10] B. “Infinitary Term Graph Rewriting is Simple, Sound and Complete”. In: *RTA*. 2012.

Working with Term Graphs

Some Observations

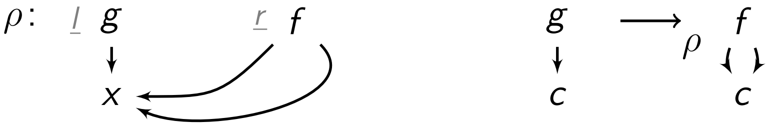
- Term graphs can be messy
  - Very operational style of term graph rewriting
  - Böhm reduction is not left-linear
- But: sharing simplifies some things
  - Reduction produces no duplication
  - Residuals & developments are easier

Working with Term Graphs

Some Observations

- Term graphs can be messy
  - Very operational style of term graph rewriting
  - Böhm reduction is not left-linear
- But: sharing simplifies some things
  - Reduction produces no duplication
  - Residuals & developments are easier

Example ($g(x) \to f(x, x)$)



Working with Term Graphs

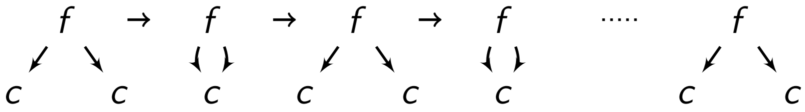
Some Observations

- Term graphs can be messy
  - Very operational style of term graph rewriting
  - Böhm reduction is not left-linear
- But: sharing simplifies some things
  - Reduction produces no duplication
  - Residuals & developments are easier
- Weak convergence is even weirder than on terms:

Working with Term Graphs

Some Observations

- Term graphs can be messy
  - Very operational style of term graph rewriting
  - Böhm reduction is not left-linear
- But: sharing simplifies some things
  - Reduction produces no duplication
  - Residuals & developments are easier
- Weak convergence is even weirder than on terms:



Future Work

- Infinitary confluence for term graphs
- Coinductive definition of infinitary term graph rewriting
- Axiomatic account of meaningless term graphs
- Partial-order reduction corresponding to Böhm reductions other than root-active terms

Böhm Reduction in Infinitary Term Graph Rewriting Systems

Patrick Bahr

IT University of Copenhagen

The Metric Model of Infinitary Rewriting

Convergence

based on the ‘usual’ complete metric space on terms

$$\mathbf{d}(s, t) = 2^{-n}$$

$n =$ depth of the shallowest discrepancy of s and t

The Metric Model of Infinitary Rewriting

Convergence

based on the 'usual' complete metric space on terms

$$\mathbf{d}(s, t) = 2^{-n}$$

$n =$ depth of the shallowest discrepancy of s and t

Convergence of reductions

(a.k.a. strong convergence)

- convergence in the metric space, and
- rewrite rules are applied (eventually) at increasingly large depth

The Metric Model of Infinitary Rewriting

Convergence

based on the ‘usual’ complete metric space on terms

$$\mathbf{d}(s, t) = 2^{-n}$$

$n =$ depth of the shallowest discrepancy of s and t

Convergence of reductions

(a.k.a. strong convergence)

- convergence in the metric space, and
- rewrite rules are applied (eventually) at increasingly large depth

⇝ convergence of a reduction: depth at which the rewrite rules are applied tends to infinity

Partial Order Infinitary Rewriting

Partial order on terms

- partial terms: terms with additional constant $\bot$
- partial order $\leq_\bot$ reads as: "is less defined than"
- $\leq_\bot$ is a complete semilattice
  (= cpo + glbs of non-empty sets)

Partial Order Infinitary Rewriting

Partial order on terms

- partial terms: terms with additional constant $\perp$
- partial order $\leq_\perp$ reads as: "is less defined than"
- $\leq_\perp$ is a complete semilattice
  (= cpo + glbs of non-empty sets)

Convergence: limit inferior

$$\liminf_{\iota \to \alpha} t_\iota = \bigsqcup_{\beta < \alpha} \bigsqcap_{\beta \leq \iota < \alpha} t_\iota$$

Partial Order Infinitary Rewriting

Partial order on terms

- partial terms: terms with additional constant $\bot$
- partial order $\leq_\bot$ reads as: "is less defined than"
- $\leq_\bot$ is a complete semilattice
  (= cpo + glbs of non-empty sets)

Convergence: limit inferior

$$\liminf_{\iota \to \alpha} t_\iota = \bigsqcup_{\beta < \alpha} \bigsqcap_{\beta \leq \iota < \alpha} t_\iota$$

- intuition: eventual persistence of nodes in the tree
- strong convergence: limit inferior of the contexts of the reduction

Metric on Term Graphs

Depth of a node = length of a shortest path from the root to the node.

Metric on Term Graphs

Depth of a node = length of a shortest path from the root to the node.

Truncation of term graphs

The truncation $g \dagger d$ is obtained from g by

- relabelling all nodes at depth d with $\bot$, and
- removing all nodes that thus become unreachable from the root.

Metric on Term Graphs

Depth of a node = length of a shortest path from the root to the node.

Truncation of term graphs

The truncation $g\dagger d$ is obtained from g by

- relabelling all nodes at depth d with $\bot$, and
- removing all nodes that thus become unreachable from the root.

Metric on term graphs

$$\mathbf{d}(g, h) = 2^{-n}$$

Where n = maximum depth d s.t. $g\dagger d \cong h\dagger d$.

A Partial Order on Term Graphs – How?

$\bot$-homomorphisms $\phi\colon g \to_\bot h$

- homomorphism condition suspended on $\bot$-nodes
- allow mapping of $\bot$-nodes to arbitrary nodes

A Partial Order on Term Graphs – How?

$\bot$-homomorphisms $\phi\colon\ g \to_\bot h$

- homomorphism condition suspended on $\bot$-nodes
- allow mapping of $\bot$-nodes to arbitrary nodes

Proposition

For all terms s, t: $\quad s \leq_\bot t \quad$ *iff* $\quad \exists \phi\colon\ s \to_\bot t$

A Partial Order on Term Graphs – How?

$\bot$-homomorphisms $\phi\colon g \to_\bot h$

- homomorphism condition suspended on $\bot$-nodes
- allow mapping of $\bot$-nodes to arbitrary nodes

Proposition

For all terms s, t: $\quad s \leq_\bot t \quad$ *iff* $\quad \exists \phi\colon s \to_\bot t$

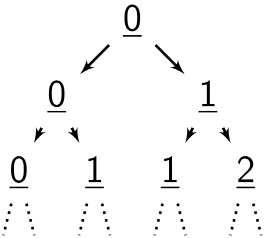
Definition

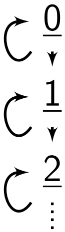
For all term graphs g, h, let $g \leq_\bot h$ iff there is some $\phi\colon g \to_\bot h$.

$$\mathcal{R} = \{ \quad \underline{n}(x, y) \to \underline{n+1}(x, y) \quad | \quad n \in \mathbb{N} \quad \}.$$

[11] R. Kennaway et al. “On the adequacy of graph rewriting for simulating term rewriting”. In: *ACM Transactions on Programming Languages and Systems* (1994).

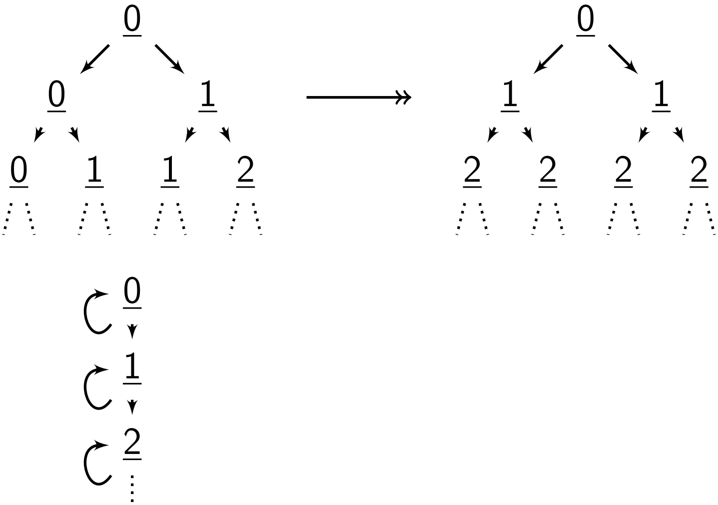
$$\mathcal{R} = \{ \quad \underline{n}(x,y) \rightarrow \underline{n+1}(x,y) \quad | \quad n \in \mathbb{N} \quad \}.$$





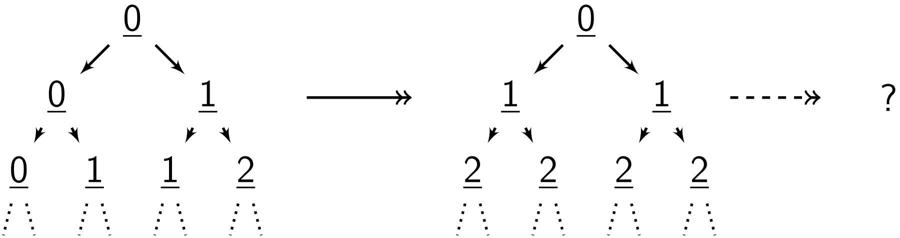
[11] R. Kennaway et al. "On the adequacy of graph rewriting for simulating term rewriting". In: *ACM Transactions on Programming Languages and Systems* (1994).

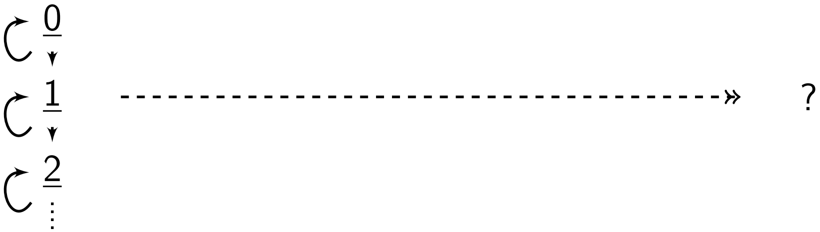
$$\mathcal{R} = \{ \quad \underline{n}(x,y) \to \underline{n+1}(x,y) \quad | \quad n \in \mathbb{N} \quad \}.$$



[11] R. Kennaway et al. “On the adequacy of graph rewriting for simulating term rewriting”. In: *ACM Transactions on Programming Languages and Systems* (1994).

$$\mathcal{R} = \{ \quad \underline{n}(x,y) \to \underline{n+1}(x,y) \quad | \quad n \in \mathbb{N} \quad \}.$$

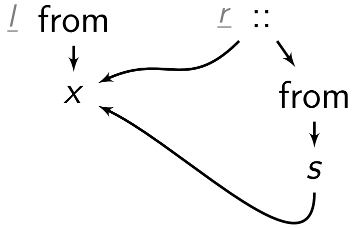




[11] R. Kennaway et al. "On the adequacy of graph rewriting for simulating term rewriting". In: *ACM Transactions on Programming Languages and Systems* (1994).

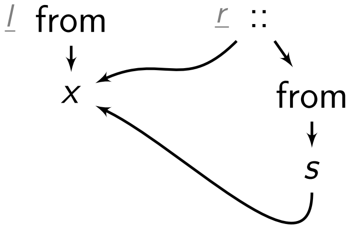
Example: Acyclic Sharing

Term graph rule for $\mathit{from}(x) \to x :: \mathit{from}(s(x))$

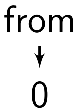


Example: Acyclic Sharing

Term graph rule for *from*(x) $\to$ x :: *from*(s(x))

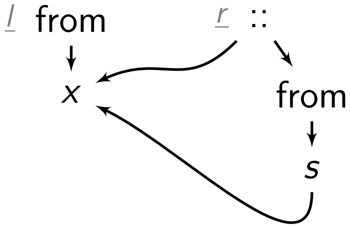


Reductions:

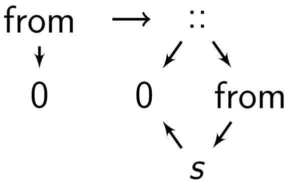


Example: Acyclic Sharing

Term graph rule for $\mathit{from}(x) \to x :: \mathit{from}(s(x))$

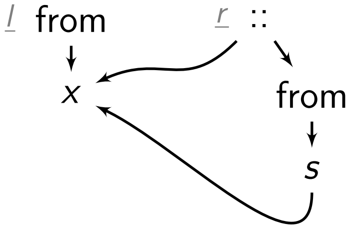


Reductions:

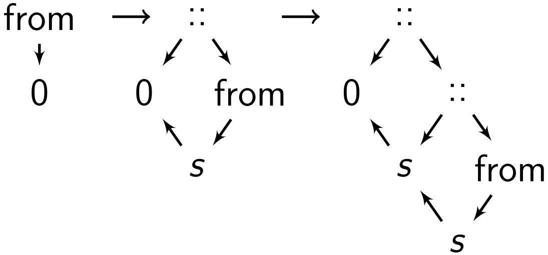


Example: Acyclic Sharing

Term graph rule for $from(x) \to x :: from(s(x))$

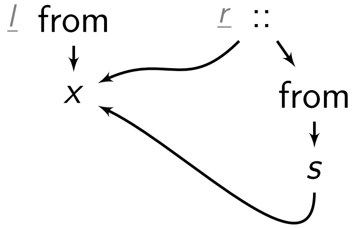


Reductions:



Example: Acyclic Sharing

Term graph rule for $from(x) \to x :: from(s(x))$



Reductions: