Infinitary Rewriting of Terms, Trees and Graphs

Patrick Bahr
paba@diku.dk

University of Copenhagen
Department of Computer Science

TF Lunch
Utrecht University
April 4, 2012
Outline

1. Introduction
   - Functional Programming & Lazy Evaluation
   - Infinite Reductions
   - From Terms to Graphs
   - Goals
   - Obstacles

2. Infinitary Term Graph Rewriting
   - Metric Approach
   - Partial Order Approach
   - Metric vs. Partial Order Approach
   - Soundness & Completeness Properties
Newton-Raphson Square Roots

Approximating $\sqrt{N}$

$$a_{n+1} = \frac{a_n + N/a_n}{2}$$
Newton-Raphson Square Roots

Approximating \( \sqrt{N} \)

\[
a_{n+1} = \frac{a_n + N/a_n}{2}
\]

Simple imperative algorithm

\[
x \leftarrow a_0 \\
\text{repeat} \\
\quad y \leftarrow x \\
\quad x \leftarrow (x + N/x)/2 \\
\text{until} \ |x - y| \leq \varepsilon \\
\text{return } x
\]
Newton-Raphson Square Roots

Approximating $\sqrt{N}$

$$a_{n+1} = \frac{a_n + N/a_n}{2}$$

Generates an infinite list

$[a_0, f(a_0), f(f(a_0)), f(f(f(a_0))), \ldots]$  

Simple imperative algorithm

```
x ← a_0
repeat
    y ← x
    x ← (x + N/x)/2
until |x - y| ≤ ε
return x
```
Newton-Raphson Square Roots

Approximating $\sqrt{N}$

\[
a_{n+1} = \frac{a_n + N/a_n}{2}
\]

Generates an infinite list

\[
[a_0, f(a_0), f(f(a_0)), f(f(f(a_0))), \ldots]
\]

Simple imperative algorithm

\[
\begin{align*}
x & \leftarrow a_0 \\
\text{repeat} & \\
\hspace{1em} y & \leftarrow x \\
\hspace{1em} x & \leftarrow (x + N/x)/2 \\
\text{until} & \ |x - y| \leq \varepsilon \\
\text{return} & \ x
\end{align*}
\]

\[
\text{repeat } f \ a = a :: \text{repeat } f \ (f \ a)
\]
Newton-Raphson Square Roots

Approximating $\sqrt{N}$

$$a_{n+1} = \frac{a_n + N/a_n}{2}$$

Generates an infinite list

$$[a_0, f(a_0), f(f(a_0)), f(f(f(a_0))), \ldots]$$

Simple imperative algorithm

```plaintext
x ← a_0
repeat
    y ← x
    x ← (x + N/x)/2
until |x - y| ≤ ε
return x
```

```
repeat f a = a :: repeat f (f a)
next N x = (x + N/x)/2
```
Newton-Raphson Square Roots

### Approximating $\sqrt{N}$

$$a_{n+1} = \frac{a_n + N/a_n}{2}$$

Generates an infinite list

$$[a_0, f(a_0), f(f(a_0)), f(f(f(a_0))), \ldots]$$

### Simple imperative algorithm

\[
\begin{align*}
&x \leftarrow a_0 \\
&\text{repeat} \\
&\quad y \leftarrow x \\
&\quad x \leftarrow (x + N/x)/2 \\
&\text{until } |x - y| \leq \varepsilon \\
&\text{return } x
\end{align*}
\]

```
repeat f a
next N x
within ε (a :: (b :: rest))
```

\[
= a :: repeat f (f a)
= (x + N/x)/2
= \text{if } |a - b| \leq \varepsilon
\text{ then } b
\text{ else within } ε \ (b :: \text{rest})
\]
Newton-Raphson Square Roots

Approximating $\sqrt{N}$

\[ a_{n+1} = \frac{a_n + N/a_n}{2} \]

Generates an infinite list

\[ [a_0, f(a_0), f(f(a_0)), f(f(f(a_0))), \ldots] \]

Simple imperative algorithm

\[
\begin{align*}
x & \leftarrow a_0 \\
\text{repeat} & \\
y & \leftarrow x \\
x & \leftarrow (x + N/x)/2 \\
\text{until} & \ |x - y| \leq \varepsilon \\
\text{return} & \ x
\end{align*}
\]

\[
\begin{align*}
\text{repeat } f \ a & = a :: \text{repeat } f \ (f \ a) \\
\text{next } N \ x & = (x + N/x)/2 \\
\text{within } \varepsilon \ (a :: (b :: \text{rest})) & = \text{if } |a - b| \leq \varepsilon \\
& \quad \text{then } b \\
& \quad \text{else } \text{within } \varepsilon \ (b :: \text{rest}) \\
\text{sqrt } a_0 \in \varepsilon \ N & = \text{within } \varepsilon \ (\text{repeat (next } N \ a_0) \ a_0)
\end{align*}
\]
Lazy Evaluation

Subexpressions are evaluated only when they are needed.
Lazy Evaluation

Subexpressions are evaluated only when they are needed.

\[
\begin{align*}
\text{repeat } f \ a &\quad = \ a :: \text{repeat } f \ (f \ a) \\
\text{next } N \ x &\quad = \ (x + N/x)/2 \\
\text{within } \epsilon \ (a :: (b :: \text{rest})) &\quad = \ \text{if } |a - b| \leq \epsilon \\
&\quad \text{then } b \\
&\quad \text{else within } \epsilon \ (b :: \text{rest}) \\
\sqrt{a_0} \in N &\quad = \ \text{within } \epsilon \ (\text{repeat } (\text{next } N) \ a_0)
\end{align*}
\]
Lazy Evaluation

Subexpressions are evaluated only when they are needed.

\[
\begin{align*}
\text{repeat } f \ a & \quad = \ a :: \text{repeat } f \ (f \ a) \\
\text{next } N \ x & \quad = \ (x + N/x)/2 \\
\text{within } \varepsilon \ (a :: (b :: \text{rest})) & \quad = \ \text{if } |a - b| \leq \varepsilon \\
& \quad \text{then } b \\
& \quad \text{else } \text{within } \varepsilon \ (b :: \text{rest}) \\
\sqrt{a_0} \in N & \quad = \ \text{within } \varepsilon \ (\text{repeat } (\text{next } N) \ a_0)
\end{align*}
\]

Infinitary term rewriting aims to model infinite reductions explicitly.
Formalising Infinitary Term Rewriting

Complete metric on terms

- terms are endowed with a **complete metric** in order to formalise the convergence of infinite reductions.
- metric distance between terms is inversely proportional to the shallowest depth at which they differ:

\[
d(s, t) = 2^{-\text{sim}(s, t)}
\]

\(\text{sim}(s, t)\) – depth of the shallowest discrepancy of \(s\) and \(t\)
Formalising Infinitary Term Rewriting

Complete metric on terms
- terms are endowed with a complete metric in order to formalise the convergence of infinite reductions.
- metric distance between terms is inversely proportional to the shallowest depth at which they differ:

\[
d(s, t) = 2^{-\text{sim}(s, t)}
\]

\text{sim}(s, t) – depth of the shallowest discrepancy of } s \text{ and } t

Example

\[\begin{array}{c}
  f \\
  \downarrow \\
  a \\
  \downarrow \\
  b \\
\end{array}
\quad
\begin{array}{c}
  f \\
  \downarrow \\
  a \\
  \downarrow \\
  c \\
  \downarrow \\
  g \\
  \downarrow \\
  a \\
\end{array}\]
Formalising Infinitary Term Rewriting

Complete metric on terms

- terms are endowed with a complete metric in order to formalise the convergence of infinite reductions.
- metric distance between terms is inversely proportional to the shallowest depth at which they differ:

\[ d(s, t) = 2^{-\text{sim}(s, t)} \]

\( \text{sim}(s, t) \) – depth of the shallowest discrepancy of \( s \) and \( t \)

Example

\[
\begin{array}{c}
\text{f} \\
\text{a} \\
\text{b} \\
\text{s}
\end{array}
\quad \triangleright \quad
\begin{array}{c}
\text{f} \\
\text{a} \\
\text{g} \\
\text{t}
\end{array}
\]

\[
\begin{array}{c}
\text{f} \\
\text{a} \\
\text{g} \\
\text{t}
\end{array}
\quad \triangleright \quad
\begin{array}{c}
\text{f} \\
\text{a} \\
\text{c} \\
\text{t}
\end{array}
\]

\[
\begin{array}{c}
\text{f} \\
\text{a} \\
\text{g} \\
\text{t}
\end{array}
\quad \triangleright \quad
\begin{array}{c}
\text{f} \\
\text{a} \\
\text{c} \\
\text{t}
\end{array}
\]
Formalising Infinitary Term Rewriting

Complete metric on terms

- terms are endowed with a complete metric in order to formalise the convergence of infinite reductions.
- metric distance between terms is inversely proportional to the shallowest depth at which they differ:

\[ d(s, t) = 2^{-\text{sim}(s,t)} \]

\text{sim}(s, t) – depth of the shallowest discrepancy of } s \text{ and } t

Example

\[ d(s) = \frac{1}{2} \]
Formalising Infinitary Term Rewriting

Complete metric on terms

- terms are endowed with a complete metric in order to formalise the convergence of infinite reductions.
- metric distance between terms is inversely proportional to the shallowest depth at which they differ:

\[ d(s, t) = 2^{-\text{sim}(s, t)} \]

\( \text{sim}(s, t) \) – depth of the shallowest discrepancy of \( s \) and \( t \)

Example

\[ d(s, t) = \frac{1}{2} \]

\[ d(s, t) = \frac{1}{4} \]
Formalising Infinitary Term Rewriting

Complete metric on terms

- terms are endowed with a complete metric in order to formalise the convergence of infinite reductions.
- metric distance between terms is inversely proportional to the shallowest depth at which they differ:

\[ d(s, t) = 2^{-\text{sim}(s, t)} \]

\( \text{sim}(s, t) \) – depth of the shallowest discrepancy of s and t

Example

\[
\begin{align*}
    &f \\
    &\quad \downarrow \quad \downarrow \\
    &a \quad f \\
    &\quad \downarrow \quad \downarrow \\
    &b \quad c \\
\end{align*}
\]

\[
\begin{align*}
    &f \\
    &\quad \downarrow \quad \downarrow \\
    &a \quad g \\
    &\quad \downarrow \quad \downarrow \\
    &u \quad b
\end{align*}
\]

\[ d(s, t) = \frac{1}{2} \]
Formalising Infinitary Term Rewriting

Complete metric on terms

- terms are endowed with a complete metric in order to formalise the convergence of infinite reductions.
- metric distance between terms is inversely proportional to the shallowest depth at which they differ:

\[ d(s, t) = 2^{-\text{sim}(s, t)} \]

\( \text{sim}(s, t) \) – depth of the shallowest discrepancy of \( s \) and \( t \)

Example

\[ d(s, t) = \frac{1}{2} \]

\[ d(u, v) = \frac{1}{4} \]
Convergence of Transfinite Reductions

Two different kinds of convergence

- **weak convergence**: convergence in the metric space of terms
  - for weak convergence the **depth of the discrepancies** of the terms has to tend to infinity

- **strong convergence**: convergence in the metric space + rewrite rules have to (eventually) be applied at increasingly large depth
  - for strong convergence the **depth of where the rewrite rules are applied** has to tend to infinity
Example: Weak Convergence

\[ f(x) \to f(g(x)) \]
Example: Weak Convergence

\[ f(x) \rightarrow f(g(x)) \]
Example: Weak Convergence

\[ f(x) \rightarrow f(g(x)) \]
Example: Weak Convergence

\[ f(x) \rightarrow f(g(x)) \]
Example: Weak Convergence

\[ f(x) \to f(g(x)) \]
Example: Weak Convergence

\[ f(x) \rightarrow f(g(x)) \]
Example: Weak Convergence

\[ f(x) \rightarrow f(g(x)) \]
Example: Weak Convergence

\[ f(x) \to f(g(x)) \]
Example: Weak Convergence

\[ f(x) \rightarrow f(g(x)) \]
Example: Weak Convergence

\[ f(x) \rightarrow f(g(x)) \]
Example: Weak Convergence

\[ f(x) \rightarrow f(g(x)) \]
Example: Weak Convergence

\[ f(x) \to f(g(x)) \]
Example: Strong Convergence

\[ a \rightarrow g(a) \]
Example: Strong Convergence

\[ a \rightarrow g(a) \]
Example: Strong Convergence

\[ a \rightarrow g(a) \]
Example: Strong Convergence

\[ a \rightarrow g(a) \]
Example: Strong Convergence

\[ a \rightarrow g(a) \]
Example: Strong Convergence

\[ a \rightarrow g(a) \]
Example: Strong Convergence

\[ a \rightarrow g(a) \]
Example: Strong Convergence

\[ a \to g(a) \]
Example: Strong Convergence

$a \rightarrow g(a)$
Example: Strong Convergence

\[
a \rightarrow g(a)
\]
Some Interesting Properties

Compression

Every reduction can be performed in at most $\omega$ steps:

$$s \xrightarrow{\alpha} t \quad \implies \quad s \xrightarrow{\leq \omega} t$$
Some Interesting Properties

Compression
Every reduction can be performed in at most $\omega$ steps:

$$s \xrightarrow{\alpha} t \implies s \xrightarrow{\leq\omega} t$$

Finite approximation
Every outcome can be approximated by a finite reduction arbitrary well:

$$s \xrightarrow{\alpha} t \implies \forall d \in \mathbb{N} \exists t' \left\{ s \xrightarrow{*} t' \right\} \text{ t and t' coincide up to depth } d$$
The Full Story of Lazy Evaluation

Subexpressions are evaluated only when they are needed.
The Full Story of Lazy Evaluation

Subexpressions are evaluated only when they are needed.

\[
\begin{align*}
\text{repeat } f \ a & = a :: \text{repeat } f \ (f \ a) \\
\text{next } N \ x & = (x + N/x)/2 \\
\text{within } \varepsilon \ (a :: (b :: \text{rest})) & = \text{if } |a - b| \leq \varepsilon \\
& \quad \text{then } b \\
& \quad \text{else within } \varepsilon \ (b :: \text{rest}) \\
\text{sqrt } a_0 \in N & = \text{within } \varepsilon \ (\text{repeat } (\text{next } N) \ a_0)
\end{align*}
\]
The Full Story of Lazy Evaluation

Subexpressions are evaluated only when they are needed.

\[
\text{repeat } f \ a \quad = \quad a :: \text{repeat } f \ (f \ a)
\]

\[
\text{next } N \ x \quad = \quad (x + N/x)/2
\]

\[
\text{within } \varepsilon \ (a :: (b :: \text{rest})) \quad = \quad \text{if } |a - b| \leq \varepsilon
\]

\[
\quad \quad \quad \quad \quad \text{then } b
\]

\[
\quad \quad \quad \quad \quad \text{else } \text{within } \varepsilon \ (b :: \text{rest})
\]

\[
\text{sqrt } a_0 \in N \quad = \quad \text{within } \varepsilon \ (\text{repeat } (\text{next } N) \ a_0)
\]

Each subexpression is evaluated at most once.
The Full Story of Lazy Evaluation

Subexpressions are evaluated only when they are needed.

\[
\begin{align*}
\text{repeat } f \ a & \quad = \quad a :: \text{repeat } f \ (f \ a) \\
\text{next } N \ x & \quad = \quad (x + N/x)/2 \\
\text{within } \varepsilon \ (a :: (b :: \text{rest})) & \quad = \quad \begin{cases} 
  b & \text{if } |a - b| \leq \varepsilon \\
  \text{within } \varepsilon \ (b :: \text{rest}) & \text{else}
\end{cases} \\
\sqrt{a_0} \in N & \quad = \quad \text{within } \varepsilon \ (\text{repeat (next } N \ a_0))
\end{align*}
\]

Each subexpression is evaluated at most once even if its duplicated.
The Full Story of Lazy Evaluation

Subexpressions are evaluated only when they are needed.

\[
\begin{align*}
\text{repeat } f \ a & \quad = \ a :: \text{repeat } f \ (f \ a) \\
\text{next } N \ x & \quad = \ (x + N/x)/2 \\
\text{within } \varepsilon \ (a :: (b :: \text{rest})) & \quad = \ \text{if } |a - b| \leq \varepsilon \\
& \quad \text{then } b \\
& \quad \text{else } \text{within } \varepsilon \ (b :: \text{rest}) \\
\text{sqrt } a_0 \in N & \quad = \ \text{within } \varepsilon \ (\text{repeat } (\text{next } N) \ a_0)
\end{align*}
\]

Each subexpression is evaluated at most once even if its duplicated.

Term graph rewriting allows sharing of subexpressions
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

\[
\begin{align*}
 & f(g(a), h(g(a), a)) \\
\end{align*}
\]

Diagram:

```
   f
  / \
 g   h
 /   \
 a   a
```

Unravel:

```
   f
  / \
 g   h
 /   \
 a   a
```

Diagram:

```
   f
  / \
 g   h
 /   \
 a
```

\[a \rightarrow b\]
From Terms to Term Graphs

\[ f(g(a), h(g(a)), a) \]

unravel

\[ a \rightarrow b \]
From Terms to Term Graphs

\[ f(g(a), h(g(a), a)) \]

\[ a \rightarrow b \]
From Terms to Term Graphs
From Terms to Term Graphs

\[ f \left( g(a), h(g(a), a) \right) \]

unravel
From Terms to Term Graphs

\[ f(g(a)), h(g(a)), a \rightarrow b \]

unravel

\[ b \rightarrow c \]
From Terms to Term Graphs

\[ f(g(a), h(g(a), a)) \]

unravel

\[ b \rightarrow c \]
From Terms to Term Graphs

\[ f(g(a), h(g(a), a)) \]

\[ b \rightarrow c \]
Goals

What is this about?

- finding appropriate notions of converging term graph reductions
- generalising convergence for term reductions
Goals

What is this about?
- finding appropriate notions of converging term graph reductions
- generalising convergence for term reductions

Infinitary term graph rewriting – what is it for?
- common formalism to study correspondences between infinitary term rewriting and finitary term graph rewriting
- infinitary term graph rewriting to model lazy evaluation
  - infinitary term rewriting only covers non-strictness
  - however: lazy evaluation = non-strictness + sharing
- towards infinitary lambda calculi with letrec
  - Ariola & Blom. *Skew confluence and the lambda calculus with letrec.*
  - the calculus is non-confluent
  - but there is a notion of infinite normal forms
Obstacles

What is the an appropriate notion of convergence on term graph?

- It should **generalise convergence on terms.**
  - **But:** there are many quite different generalisations.
  - Most important issue: How to deal with **sharing**?
- It should simulate infinitary term rewriting in a sound & complete manner.
Obstacles

What is the appropriate notion of convergence on term graph?

- It should generalize convergence on terms.
  - But: there are many quite different generalizations.
  - Most important issue: How to deal with sharing?
- It should simulate infinitary term rewriting in a sound & complete manner.

Soundness of infinitary term graph rewriting:

\[
\text{U} (\cdot) \quad U (\mathcal{R}) \quad g \quad \overrightarrow{\mathcal{R}} \quad h 
\]
Obstacles

What is the appropriate notion of convergence on term graph?
- It should generalise convergence on terms.
  - But: there are many quite different generalisations.
  - Most important issue: How to deal with sharing?
- It should simulate infinitary term rewriting in a sound & complete manner.

Soundness of infinitary term graph rewriting

\[
\begin{align*}
\mathcal{R} & \quad g & \quad \mathcal{U}(\cdot) & \quad h \\
\mathcal{U}(\cdot) & \quad s & \quad \mathcal{U}(\cdot) & \quad t \\
\mathcal{U}(\mathcal{R}) & \quad \mathcal{U}(\cdot) & & \quad \mathcal{U}(\cdot)
\end{align*}
\]
Completeness of Term Graph Rewriting

An issue even for finitary acyclic term graph reductions!

\[
\begin{array}{c}
\text{f} \\
\downarrow \quad \downarrow \\
\text{a} \quad \text{a} \\
\end{array} \quad \rightarrow \quad \begin{array}{c}
\text{f} \\
\downarrow \quad \downarrow \\
\text{b} \quad \text{a} \\
\end{array}
\]
Completeness of Term Graph Rewriting

An issue even for **finitary acyclic** term graph reductions!

![Diagram showing term graph rewritings](image)
Completeness of Term Graph Rewriting

An issue even for **finitary acyclic** term graph reductions!

![Diagram showing term graph reductions](image)
Completeness of Term Graph Rewriting

An issue even for finitary acyclic term graph reductions!

\[ a \rightarrow b \]
Completeness of Term Graph Rewriting

An issue even for finitary acyclic term graph reductions!

Completeness w.r.t. term graph rewriting
Completeness of Term Graph Rewriting

An issue even for finitary acyclic term graph reductions!

\[ a \rightarrow b \]

Completeness w.r.t. term graph rewriting

\[ s \xrightarrow{U(\cdot)} t \]

\[ g \xrightarrow{*} h \]
Completeness of Term Graph Rewriting

An issue even for finitary acyclic term graph reductions!

Completeness w.r.t. term graph rewriting
Completeness of Term Graph Rewriting

An issue even for **finitary acyclic** term graph reductions!

\[
\begin{array}{c}
\text{a} \\
\text{a} \\
\text{b} \\
\text{b}
\end{array}
\rightarrow
\begin{array}{c}
\text{a} \\
\text{b} \\
\text{b}
\end{array}
\]

Completeness w.r.t. term graph rewriting

\[
\begin{array}{c}
\text{s} \\
\text{U}() \\
\text{g}
\end{array}
\rightarrow
\begin{array}{c}
\text{t} \\
\text{h}
\end{array}
\rightarrow
\begin{array}{c}
\text{t'} \\
\text{U}() \\
\text{h}
\end{array}
\]
Completeness of Infinitary Term Graph Rewriting?

We have a rule $n(x, y) \rightarrow n + 1(x, y)$ for each $n \in \mathbb{N}$. 

[Kennaway et al., 1994]
Completeness of Infinitary Term Graph Rewriting?

We have a rule \( n(x, y) \rightarrow n + 1(x, y) \) for each \( n \in \mathbb{N} \).

[Kennaway et al., 1994]
Completeness of Infinitary Term Graph Rewriting?

We have a rule \( n(x, y) \rightarrow n+1(x, y) \) for each \( n \in \mathbb{N} \).

[Kennaway et al., 1994]
Completeness of Infinitary Term Graph Rewriting?

We have a rule \( n(x, y) \rightarrow n+1(x, y) \) for each \( n \in \mathbb{N} \).

[Kennaway et al., 1994]
Outline

1. Introduction
   - Functional Programming & Lazy Evaluation
   - Infinite Reductions
   - From Terms to Graphs
   - Goals
   - Obstacles

2. Infinitary Term Graph Rewriting
   - Metric Approach
   - Partial Order Approach
   - Metric vs. Partial Order Approach
   - Soundness & Completeness Properties
Towards a Metric on Term Graphs

We want to generalise the metric on terms

\[ d(s, t) = 2^{-\text{sim}(s, t)} \]

\( \text{sim}(s, t) = \text{minimum depth } d \text{ s.t. } s \text{ and } t \text{ differ at depth } d \)

Alternative characterisation of \( \text{sim}(s, t) \) via truncation

**Truncation** \( t|d \) of a term \( t \) at depth \( d \):

\[ t|0 = \perp \]

\[ f(t_1, \ldots, t_k)|d + 1 = f(t_1|d, \ldots, t_k|d) \]

Then \( \text{sim}(s, t) = \text{maximum depth } d \text{ s.t. } s|d = t|d. \)
A Metric on Term Graphs

Depth of a node = length of a shortest path from the root to the node.
A 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.
A 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 $g^d$ is obtained from $g$ by

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

The simple metric on term graphs

$$d^d(g, h) = 2^{-\text{sim}^d(g, h)}$$

Where $\text{sim}^d(g, h) = \text{maximum depth } d \text{ s.t. } g^d \cong h^d.$
A 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.

The simple metric on term graphs

$$d^{\dagger}(g, h) = 2^{-\text{sim}^{\dagger}(g, h)}$$

Where $\text{sim}^{\dagger}(g, h) = \text{maximum depth } d \text{ s.t. } g^{\dagger d} \cong h^{\dagger d}$.

Strong convergence via metric $d^{\dagger}$ and redex depth

- convergence in the metric space $(G_{C}^{\infty}(\Sigma), d^{\dagger})$
- depth of redexes has to tend to infinity
Example: $rep(x) \rightarrow x :: rep(f(x))$
Example: \( \text{rep}(x) \rightarrow x :: \text{rep}(f(x)) \)
Example: $rep(x) \rightarrow x :: rep(f(x))$
Example: \( \text{rep}(x) \rightarrow x :: \text{rep}(f(x)) \)
Example: \( \text{rep}(x) \rightarrow x :: \text{rep}(f(x)) \)
Example: \( rep(x) \rightarrow x :: rep(f(x)) \)
Example: \( rep(x) \rightarrow x :: rep(f(x)) \)
Example: \( rep(x) \rightarrow x :: rep(f(x)) \)
Soundness & Completeness

Theorem (soundness of metric convergence)
For every left-linear, left-finite GRS \( R \) we have
\[ g \rightarrow \{ m \} R h = \Rightarrow U(g) \rightarrow \{ m \} U(R) U(h). \]

Completeness property
\[ g \cup (\cdot) \cup (R) \rightarrow (\cdot) \Rightarrow h \cup (\cdot) \cup (R) \rightarrow (\cdot). \]
Soundness & Completeness

Theorem (soundness of metric convergence)

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

$$g \xrightarrow{\mathcal{R}} h \implies \mathcal{U}(g) \xrightarrow{\mathcal{U}(\mathcal{R})} \mathcal{U}(h).$$
Soundness & Completeness

**Theorem (soundness of metric convergence)**

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

$$g \xrightarrow{m_{\mathcal{R}}} h \quad \implies \quad \mathcal{U}(g) \xrightarrow{m_{\mathcal{U}(\mathcal{R})}} \mathcal{U}(h).$$

**Completeness property**

$$\mathcal{U}(\mathcal{R}) \xrightarrow{s} \mathcal{U}(\mathcal{R}) \xrightarrow{t} \mathcal{U}(\cdot)$$
Soundness & Completeness

Theorem (soundness of metric convergence)

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

$$g \xrightarrow{m_{\mathcal{R}}} h \quad \implies \quad \mathcal{U}(g) \xrightarrow{m_{\mathcal{U}(\mathcal{R})}} \mathcal{U}(h).$$

Completeness property

\[
\begin{array}{ccc}
\mathcal{U}(\mathcal{R}) & s & t \\
\mathcal{U}(\cdot) & \searrow & \downarrow \\
\mathcal{R} & g & h \\
\end{array}
\]
Failure of Completeness for Metric Convergence

We have a rule \( n(x, y) \to n + 1(x, y) \) for each \( n \in \mathbb{N} \).
Partial Order Infinitary Term Rewriting

Partial order on terms

- **partial terms**: terms with additional constant $\perp$ (read as “undefined”)
- partial order $\leq_{\perp}$ reads as: “is less defined than”
- $\leq_{\perp}$ is a **complete semilattice** ($= \text{cpo} + \text{glbs of non-empty sets}$)

**Convergence** formalised by the limit inferior:

$$\liminf_{\iota \to \alpha} t_{\iota} = \bigsqcup_{\beta < \alpha} l_{\beta} \leq_{\iota < \alpha} t_{\iota}$$

intuition: eventual persistence of nodes of the terms

weak convergence: limit inferior of the terms of the reduction

strong convergence: limit inferior of the contexts of the reduction
Partial Order Infinitary Term Rewriting

Partial order on terms
- **partial terms**: terms with additional constant ⊥ (read as “undefined”)
- partial order \( \leq_\bot \) reads as: “is less defined than”
- \( \leq_\bot \) is a **complete semilattice** (= cpo + glbs of non-empty sets)

Convergence
- formalised by the limit inferior:
  \[
  \liminf_{\iota} t_\iota = \bigcup_{\beta \prec \alpha} \bigcap_{\beta \leq_\iota \alpha} t_\iota
  \]
- intuition: eventual persistence of nodes of the terms
- weak convergence: limit inferior of the terms of the reduction
Partial Order Infinitary Term Rewriting

Partial order on terms

- **partial terms**: terms with additional constant \( \perp \) (read as “undefined”)
- partial order \( \leq \perp \) reads as: “is less defined than”
- \( \leq \perp \) is a complete semilattice (= cpo + glbs of non-empty sets)

Convergence

- formalised by the limit inferior:
  \[
  \liminf_{t_l \rightarrow \alpha} t_l = \bigsqcup_{\beta < \alpha} \bigcap_{\beta \leq \ell < \alpha} t_l
  \]
- intuition: eventual persistence of nodes of the terms
- weak convergence: limit inferior of the terms of the reduction
- strong convergence: limit inferior of the contexts of the reduction
Partial Order Infinitary Term Rewriting

Partial order on terms

- **Partial terms**: terms with additional constant \( \bot \) (read as “undefined”)
- Partial order \( \leq \) reads as: “is less defined than”
- \( \leq \bot \) is a complete semilattice (= cpo + glbs of non-empty sets)

Convergence

- Formalised by the limit inferior:
  \[
  \liminf_{\alpha \rightarrow \alpha} t_{\alpha} = \bigsqcup \bigcap_{\beta < \alpha} t_{\beta} = \bigcup_{\alpha \rightarrow \beta} \beta \leq \text{term obtained by replacing the redex with } \bot
  \]
- Intuition: eventual persistence of nodes of the terms
- Weak convergence: limit inferior of the terms of the reduction
- Strong convergence: limit inferior of the contexts of the reduction
Partial Order Convergence vs. Metric Convergence

Intuition of partial order convergence

- subterms that would break \( m \)-convergence, converge to \( \bot \)
- every (continuous) reduction converges

Theorem (total \( p \)-convergence = \( m \)-convergence)

For every reduction \( S \) in a TRS the following equivalence holds:

\[ S : s \rightarrow p t \text{ total iff } S : s \rightarrow m t \]

Theorem (normalisation & confluence)

Every orthogonal TRS is infinitarily normalising and infinitarily confluent w.r.t. strong \( p \)-convergence.
Partial Order Convergence vs. Metric Convergence

Intuition of partial order convergence
- subterms that would break $m$-convergence, converge to $\bot$
- every (continuous) reduction converges

Theorem (total $p$-convergence = $m$-convergence)

For every reduction $S$ in a TRS the following equivalence holds:

$$S: s \overset{p}{\rightarrow} t \text{ total} \iff S: s \overset{m}{\rightarrow} t$$
Partial Order Convergence vs. Metric Convergence

Intuition of partial order convergence

- subterms that would break $m$-convergence, converge to $\perp$
- every (continuous) reduction converges

Theorem (total $p$-convergence $= m$-convergence)

For every reduction $S$ in a TRS the following equivalence holds:

$S: s \xrightarrow{p} t$ total \iff $S: s \xrightarrow{m} t$

Theorem (normalisation & confluence)

Every orthogonal TRS is infinitarily normalising and infinitarily confluent w.r.t. strong $p$-convergence.
A Partial Order on Term Graphs – How?

Specialise on terms

- Consider terms as term trees (i.e. term graphs with tree structure)
- How to define the partial order $\leq_\bot$ on term trees?
A Partial Order on Term Graphs – How?

Specialise on terms
- Consider terms as term trees (i.e. term graphs with tree structure)
- How to define the partial order $\leq_\bot$ on term trees?

$\bot$-homomorphisms $\phi: g \rightarrow_\bot h$
- homomorphism condition suspended on $\bot$-nodes
- allow mapping of $\bot$-nodes to arbitrary nodes
- same mechanism that formalises matching in term graph rewriting
A Partial Order on Term Graphs – How?

Specialise on terms

- Consider terms as **term trees** (i.e. term graphs with tree structure)
- How to define the partial order \( \leq \perp \) on term trees?

**\( \perp \)-homomorphisms** \( \phi : g \rightarrow \perp h \)

- homomorphism condition suspended on \( \perp \)-nodes
- allow mapping of \( \perp \)-nodes to arbitrary nodes
- same mechanism that formalises matching in term graph rewriting

**Proposition (** \( \perp \)-homomorphisms characterise \( \leq \perp \) on terms**)**

For all \( s, t \in \mathcal{T}^\infty(\Sigma_\perp) \):

\[ s \leq \perp t \iff \exists \phi : s \rightarrow \perp t \]
A Partial Order on Term Graphs – How?

Specialise on terms

- Consider terms as term trees (i.e. term graphs with tree structure)
- How to define the partial order $\leq_\perp$ on term trees?

$\perp$-homomorphisms $\phi: g \rightarrow_\perp h$

- homomorphism condition suspended on $\perp$-nodes
- allow mapping of $\perp$-nodes to arbitrary nodes
- same mechanism that formalises matching in term graph rewriting

Proposition ($\perp$-homomorphisms characterise $\leq_\perp$ on terms)

For all $s, t \in T^\infty(\Sigma_\perp)$: $s \leq_\perp t$ iff $\exists \phi: s \rightarrow_\perp t$

Definition (Simple partial order $\leq_{\perp}^S$ on term graphs)

For all $g, h \in G^\infty(\Sigma_\perp)$, let $g \leq_{\perp}^S h$ iff there is some $\phi: g \rightarrow_\perp h$. 
A \perp\text{-Homomorphism}

\[ \phi : g \rightarrow h \]
A $\perp$-Homomorphism

\[ \phi: \quad g \rightarrow h \]
Partial Order Convergence on Term Graphs

Convergence

- Weak conv.: limit inferior of the term graphs along the reduction.
- Strong conv.: limit inferior of the contexts along the reduction.
Partial Order Convergence on Term Graphs

Convergence

- Weak conv.: limit inferior of the term graphs along the reduction.
- Strong conv.: limit inferior of the contexts along the reduction.

Context

Term graph obtained by relabelling the root node of the redex with ⊥ (and removing all nodes that become unreachable).
Partial Order Convergence on Term Graphs

**Convergence**
- Weak conv.: limit inferior of the term graphs along the reduction.
- Strong conv.: limit inferior of the contexts along the reduction.

**Context**
Term graph obtained by relabelling the root node of the redex with $\bot$ (and removing all nodes that become unreachable).

**Example**

![Diagram of a term graph with nodes labeled $f$, $c$, and edges connecting them.](image-url)
Partial Order Convergence on Term Graphs

Convergence

- Weak conv.: limit inferior of the term graphs along the reduction.
- Strong conv.: limit inferior of the contexts along the reduction.

Context

Term graph obtained by relabelling the root node of the redex with $\perp$ (and removing all nodes that become unreachable).

Example
Partial Order Convergence on Term Graphs

Convergence
- Weak conv.: limit inferior of the term graphs along the reduction.
- Strong conv.: limit inferior of the contexts along the reduction.

Context
Term graph obtained by relabelling the root node of the redex with ⊥ (and removing all nodes that become unreachable).

Example

```
  f
 /\  
/   \  
f  f  C
  \  /
   \|
    C
```

context
Partial Order Convergence on Term Graphs

Convergence
- Weak conv.: limit inferior of the term graphs along the reduction.
- Strong conv.: limit inferior of the contexts along the reduction.

Context
Term graph obtained by relabelling the root node of the redex with $\bot$ (and removing all nodes that become unreachable).

Example

```
          f
         / \  \\
        /   \  \\
       f     f
      / \    / \  \\
     /   \  /   \  \\
    C     C C     C
```

```
          f
         / \  \\
        /   \  \\
       f     f
      / \    / \  \\
     /   \  /   \  \\
    C     C C     C
```

context
Partial Order Convergence on Term Graphs

Convergence
- Weak conv.: limit inferior of the term graphs along the reduction.
- Strong conv.: limit inferior of the contexts along the reduction.

Context
Term graph obtained by relabelling the root node of the redex with ⊥ (and removing all nodes that become unreachable).

Example

```
  f  f
   ↓   ↓
   C   C
```

context

```
  f
   ↓   ⊥
   C   C
```
Partial Order Convergence on Term Graphs

Convergence
- Weak conv.: limit inferior of the term graphs along the reduction.
- Strong conv.: limit inferior of the contexts along the reduction.

Context
Term graph obtained by relabelling the root node of the redex with ⊥ (and removing all nodes that become unreachable).

Example

```
f  f  
|   |   
V   V   
f  f  
|   |   
c  c  
```

```
f  ⊥
|   |
V   
c  c  
```
Partial Order Convergence on Term Graphs

Convergence

- Weak conv.: limit inferior of the term graphs along the reduction.
- Strong conv.: limit inferior of the contexts along the reduction.

Context

Term graph obtained by relabelling the root node of the redex with $\bot$ (and removing all nodes that become unreachable).

Example

```
  f
  |   |   |
  v   v   v
f   f   f
  |   |   |
  v   v   v
C   C   C
```

context

```
  f
  |   |   |
  v   v   v
f   f   ⊥
  |   |   |
  v   v   v
C   C   C
```
Metric vs. Partial Order Approach – Weak Conv.

Recall the situation on terms

For every reduction $S$ in a TRS

$$S: s \overset{c}{\rightarrow} t \text{ in } T^\infty(\Sigma) \iff S: s \overset{m}{\rightarrow} t.$$
Metric vs. Partial Order Approach – Weak Conv.

Recall the situation on terms

For every reduction $S$ in a TRS

$$S: s \xrightarrow{P} t \text{ in } T^\infty(\Sigma) \iff S: s \xrightarrow{m} t.$$ 

On term graphs

For every reduction $S$ in a GRS

$$S: s \xrightarrow{P} t \text{ in } G^\infty(\Sigma) \Rightarrow S: s \xrightarrow{m} t.$$
Metric vs. Partial Order Approach – Weak Conv.

Recall the situation on terms

For every reduction $S$ in a TRS

$$S: s \xrightarrow{P} t \text{ in } T^\infty(\Sigma)$$

$$\iff$$

$$S: s \xrightarrow{m} t.$$

On term graphs

For every reduction $S$ in a GRS

$$S: s \xrightarrow{P} t \text{ in } G^\infty(\Sigma)$$

$$\iff$$

$$S: s \xrightarrow{m} t.$$
Metric vs. Partial Order Approach – Weak Conv.

**Recall the situation on terms**

For every reduction $S$ in a TRS

\[ S: s \xrightarrow{P} t \text{ in } T^\infty(\Sigma) \iff S: s \xrightarrow{m} t. \]

**On term graphs**

For every reduction $S$ in a GRS

\[ S: s \xrightarrow{P} t \text{ in } G^\infty(\Sigma) \quad \iff \quad S: s \xrightarrow{m} t. \]
**Metric vs. Partial Order Approach – Weak Conv.**

**Recall the situation on terms**

For every reduction $S$ in a TRS

$$S: s \xrightarrow{\mathcal{P}} t \text{ in } T^\infty(\Sigma) \iff S: s \xrightarrow{m} t.$$  

**On term graphs**

For every reduction $S$ in a GRS

$$S: s \xrightarrow{\mathcal{P}} t \text{ in } G^\infty(\Sigma) \iff S: s \xrightarrow{m} t.$$  

**Counterexample**

```
  f  →  f  →  f  →  f  →  f  ......  
    C  →  C  →  C  →  C  →  C  
```


Recall the situation on terms

For every reduction \( S \) in a TRS

\[
S: s \xrightarrow{\mathcal{D}} t \text{ in } T^\infty(\Sigma) \quad \iff \quad S: s \xrightarrow{m} t.
\]

On term graphs

For every reduction \( S \) in a GRS

\[
S: s \xrightarrow{\mathcal{D}} t \text{ in } G^\infty(\Sigma) \quad \iff \quad S: s \xrightarrow{m} t.
\]

Counterexample

\[
\begin{array}{cccccccccc}
  f & \rightarrow & f & \rightarrow & f & \rightarrow & f & \rightarrow & \ldots & f \\
\downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & \\
  c & & c & & c & & c & & c & & c
\end{array}
\]
Metric vs. Partial Order Approach – Strong Conv.

Recall the situation on terms

For every reduction $S$ in a TRS

$$S: s \overset{p}{\longrightarrow} t \text{ in } \mathcal{T}^\infty(\Sigma) \iff S: s \overset{m}{\longrightarrow} t.$$
### Metric vs. Partial Order Approach – Strong Conv.

**Recall the situation on terms**

For every reduction $S$ in a **TRS**

$$S: s \overset{p}{\rightarrow} t \text{ in } \mathcal{T}_\infty(\Sigma) \iff S: s \overset{m}{\rightarrow} t.$$ 

**On term graphs**

For every reduction $S$ in a **GRS**

$$S: s \overset{p}{\rightarrow} t \text{ in } \mathcal{G}_\infty(\Sigma) \quad ? \quad S: s \overset{m}{\rightarrow} t.$$
Recall the situation on terms
For every reduction $S$ in a $\text{TRS}$

$$S: s \xrightarrow{p} t \text{ in } T^\infty(\Sigma) \iff S: s \xrightarrow{m} t.$$ 

On term graphs
For every reduction $S$ in a $\text{GRS}$

$$S: s \xrightarrow{p} t \text{ in } G^\infty(\Sigma) \iff S: s \xrightarrow{m} t.$$
Soundness – Partial Order Convergence

Proposition
Given: a step $g \rightarrow c$ in a left-linear, left-finite GRS $R$.
Then: $U(g) \rightarrow p U(R) U(h)$ and $U(c) = d \iota < \alpha c \iota$

Theorem (Soundness)
For every left-linear, left-finite GRS $R$ we have $g \rightarrow p R h \Rightarrow U(g) \rightarrow p U(R) U(h)$. 29
Soundness – Partial Order Convergence

Proposition
Given: a step $g \rightarrow c h$ in a left-linear, left-finite GRS $R$.
Then:
$U(g) \rightarrow p U(R) \cup U(h)$
and $U(c) = d \iota \alpha c \iota$

Theorem (Soundness)
For every left-linear, left-finite GRS $R$ we have $g \rightarrow p R h = \Rightarrow U(g) \rightarrow p U(R) \cup U(h)$. 

Proposition
Given: a step $g \rightarrow c$ in a left-linear, left-finite GRS $R$.
Then:
$U(g) \rightarrow p U(R) U(h)$ and $U(c) = d_{\lambda < \alpha} c_{\lambda}$.

Theorem (Soundness)
For every left-linear, left-finite GRS $R$ we have $g \rightarrow p R h = \Rightarrow U(g) \rightarrow p U(R) U(h)$. 

Soundness – Partial Order Convergence

**Proposition**

- **Given:** a step $g \rightarrow_c h$ in a left-linear, left-finite GRS $\mathcal{R}$.
- **Then:** $\mathcal{U}(g) \xrightarrow[\mathcal{U}(\mathcal{R})]{} \mathcal{U}(h)$ and $\mathcal{U}(c) = \bigcap_{i<\alpha} c_i$
Proposition

- Given: a step $g \rightarrow_c h$ in a left-linear, left-finite GRS $\mathcal{R}$.
- Then: $\mathcal{U}(g) \mathcal{P}_{\mathcal{U}(\mathcal{R})} \mathcal{U}(h)$ and $\mathcal{U}(c) = \prod_{i < \alpha} c_i$

Theorem (Soundness)

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

$$g \mathcal{P}_{\mathcal{R}} h \quad \implies \quad \mathcal{U}(g) \mathcal{P}_{\mathcal{U}(\mathcal{R})} \mathcal{U}(h).$$
Completeness for Partial Order Convergence

**Theorem (Infinitary normalisation)**

*For each term graph* \( g \), *there is a reduction* \( g \rightarrow h \) *to a normal form* \( h \).
Completeness for Partial Order Convergence

**Theorem (Infinitary normalisation)**

For each term graph $g$, there is a reduction $g \xrightarrow{\rho} h$ to a normal form $h$.

**Theorem (Completeness)**

Strong $p$-convergence in an orthogonal, left-finite GRS $\mathcal{R}$ is complete w.r.t. strong $p$-convergence in $\mathcal{U}(\mathcal{R})$.

\[
\begin{align*}
\mathcal{U}(\mathcal{R}) & \quad s \quad \rightarrow \quad t \quad \xrightarrow{\mathcal{U}(\cdot)} \quad t' \\
\mathcal{U}(\cdot) & \quad \uparrow \\
\mathcal{R} & \quad g \quad \rightarrow \quad \mathcal{U}(\cdot) \\
\mathcal{U}(\cdot) & \quad h \quad \downarrow
\end{align*}
\]
Completeness for Partial Order Convergence

**Theorem (Infinitary normalisation)**

For each term graph $g$, there is a reduction $g \xrightarrow{p} h$ to a normal form $h$.

**Theorem (Completeness)**

Strong $p$-convergence in an orthogonal, left-finite GRS $\mathcal{R}$ is complete w.r.t. strong $p$-convergence in $\mathcal{U}(\mathcal{R})$.

**Proof.**

\[
\begin{array}{ccc}
\mathcal{U}(\mathcal{R}) & \xrightarrow{s} & t \\
\mathcal{U}(\cdot) & \downarrow & \\
\mathcal{R} & \xrightarrow{g} & \\
\end{array}
\]
Completeness for Partial Order Convergence

Theorem (Infinitary normalisation)
For each term graph \( g \), there is a reduction \( g \xrightarrow{p} h \) to a normal form \( h \).

Theorem (Completeness)
Strong \( p \)-convergence in an orthogonal, left-finite GRS \( \mathcal{R} \) is complete w.r.t. strong \( p \)-convergence in \( \mathcal{U}(\mathcal{R}) \).

Proof.

\[
\begin{align*}
\mathcal{U}(\mathcal{R}) \quad s & \quad \longrightarrow \quad t \\
\mathcal{U}(\cdot) \quad \mathcal{R} \quad g & \quad \text{normalising} \quad \longrightarrow \quad h
\end{align*}
\]
Completeness for Partial Order Convergence

**Theorem (Infinitary normalisation)**

For each term graph $g$, there is a reduction $g \xrightarrow{p} h$ to a normal form $h$.

**Theorem (Completeness)**

Strong $p$-convergence in an orthogonal, left-finite GRS $\mathcal{R}$ is complete w.r.t. strong $p$-convergence in $\mathcal{U}(\mathcal{R})$.

**Proof.**

![Diagram showing soundness and normalising relationships between term graphs and terms](diagram.png)
Completeness for Partial Order Convergence

Theorem (Infinitary normalisation)

For each term graph \( g \), there is a reduction \( g \Rightarrow h \) to a normal form \( h \).

Theorem (Completeness)

Strong \( p \)-convergence in an orthogonal, left-finite GRS \( \mathcal{R} \) is complete w.r.t. strong \( p \)-convergence in \( \mathcal{U}(\mathcal{R}) \).

Proof.

\[
\begin{align*}
\mathcal{U}(\mathcal{R}) & \xrightarrow{s} t \\
\mathcal{U}(\cdot) & \xrightarrow{\text{soundness}} \mathcal{U}(\cdot) \\
\mathcal{R} & \xrightarrow{g} h \\
& \xrightarrow{\text{normalising}} \mathcal{U}(\cdot) \\
& \xrightarrow{\text{confluence}} t' \\
& \xrightarrow{t'} \mathcal{U}(\cdot)
\end{align*}
\]
Failure of Completeness for Metric Convergence

We have a rule \( n(x, y) \rightarrow n + 1(x, y) \) for each \( n \in \mathbb{N} \).
**Theorem**

**Strong** $m$-**convergence in an orthogonal, left-finite GRS $R$ that is normalising w.r.t. strongly $m$-converging reductions** is complete for normalising reductions in $U(R)$. 


Weak(er) Completeness for Metric Convergence

**Theorem**

*Strong m-convergence in an orthogonal, left-finite GRS $\mathcal{R}$ that is normalising w.r.t. strongly m-converging reductions is complete for normalising reductions in $\mathcal{U}(\mathcal{R})$.***

\[
\begin{array}{cccccc}
U(\mathcal{R}) & s & \text{normalising} & \Rightarrow \\
U(\cdot) & \Downarrow \\
\mathcal{R} & g & \Rightarrow \\
\end{array}
\]
Weak(er) Completeness for Metric Convergence

**Theorem**

Strong m-convergence in an orthogonal, left-finite GRS $\mathcal{R}$ that is normalising w.r.t. strongly m-converging reductions is complete for normalising reductions in $\mathcal{U}(\mathcal{R})$.

**Proof.**

\[ \mathcal{U}(\mathcal{R}) \xrightarrow{s} \mathcal{U}(\cdot) \xrightarrow{\mathcal{R}} g \xrightarrow{\cdot} t \]
Weak(er) Completeness for Metric Convergence

**Theorem**

*Strong m-convergence in an orthogonal, left-finite GRS $\mathcal{R}$ that is normalising w.r.t. strongly m-converging reductions is complete for normalising reductions in $\mathcal{U}(\mathcal{R})$.*

**Proof.**

$$
\begin{align*}
\mathcal{U}(\mathcal{R}) & \quad s \quad \longrightarrow \quad t \\
\mathcal{U}(\cdot) & \quad \uparrow \\
\mathcal{R} & \quad g \quad \longmapsto \quad h
\end{align*}
$$

normalising
Weak(er) Completeness for Metric Convergence

**Theorem**

*Strong m-convergence in an orthogonal, left-finite GRS $\mathcal{R}$ that is normalising w.r.t. strongly m-converging reductions is complete for normalising reductions in $\mathcal{U}(\mathcal{R})$.*

**Proof.**
Weak(er) Completeness for Metric Convergence

**Theorem**

*Strong m-convergence in an orthogonal, left-finite GRS \( \mathcal{R} \) that is normalising w.r.t. strongly m-converging reductions is complete for normalising reductions in \( U(\mathcal{R}) \).*

**Proof.**
Weak(er) Completeness for Metric Convergence

**Theorem**

Strong $m$-convergence in an orthogonal, left-finite GRS $\mathcal{R}$ that is normalising w.r.t. strongly $m$-converging reductions is complete for normalising reductions in $\mathcal{U}(\mathcal{R})$.

**Conjecture**

**Proof.**

\[
\begin{align*}
\mathcal{U}(\mathcal{R}) & \xrightarrow{s} t \xrightarrow{\text{UN w.r.t. } m} t' \\
\mathcal{U}(\cdot) & \xrightarrow{g} h \\
\mathcal{R} & \xrightarrow{\cdot} \mathcal{U}(\cdot)
\end{align*}
\]