From Infinitary Term Rewriting to Cyclic Term Graph Rewriting and back

Patrick Bahr
paba@diku.dk

University of Copenhagen
Department of Computer Science

6th International Workshop on Computing with Terms and Graphs
Saarbrücken, Germany, April 2nd, 2011
Outline

1. Infinitary Term Rewriting

2. Term Graph Rewriting
   - Partial Order Model of Infinitary Rewriting
   - Convergence on Term Graphs

3. Outlook
Outline

1. Infinitary Term Rewriting

2. Term Graph Rewriting
   - Partial Order Model of Infinitary Rewriting
   - Convergence on Term Graphs

3. Outlook
Non-Terminating Rewriting Systems

Termination guarantees that every reduction sequence leads to a normal form, i.e. a final outcome.

Example (Infinite lists)

\[ \text{Rnats} = \{ \text{from}(x) \rightarrow x : \text{from}(s(x)) \} \]

Intuitively this converges to the infinite list 0:1:2:3:4:5:...
Non-Terminating Rewriting Systems

Termination guarantees that every reduction sequence leads to a normal form, i.e. a final outcome.

Non-terminating systems can be meaningful

- modelling reactive systems, e.g. by process calculi
- approximation algorithms which enhance the accuracy of the approximation with each iteration, e.g. computing $\pi$
- specification of infinite data structures, e.g. streams

Example (Infinite lists)

$\mathbb{N} = \{x : \text{from}(\text{from}(s(x)))\}$

intuitively this converges to the infinite list $0:1:2:3:4:5:...$.
Non-Terminating Rewriting Systems

Termination guarantees that every reduction sequence leads to a normal form, i.e. a final outcome.

Non-terminating systems can be meaningful

- modelling reactive systems, e.g. by process calculi
- approximation algorithms which enhance the accuracy of the approximation with each iteration, e.g. computing $\pi$
- specification of infinite data structures, e.g. streams

Example (Infinite lists)

$$\mathcal{R}_{\text{nats}} = \{ \text{from}(x) \rightarrow x : \text{from}(s(x)) \}$$

$$\text{from}(0)$$
Non-Terminating Rewriting Systems

Termination guarantees that every reduction sequence leads to a normal form, i.e. a final outcome.

Non-terminating systems can be meaningful

- modelling reactive systems, e.g. by process calculi
- approximation algorithms which enhance the accuracy of the approximation with each iteration, e.g. computing $\pi$
- specification of infinite data structures, e.g. streams

Example (Infinite lists)

$$\mathcal{R}_{\text{nats}} = \{ \text{from}(x) \to x : \text{from}(s(x)) \}$$

$$\text{from}(0) \to 0 : \text{from}(1)$$
Non-Terminating Rewriting Systems

Termination guarantees that every reduction sequence leads to a normal form, i.e. a final outcome.

Non-terminating systems can be meaningful

- modelling reactive systems, e.g. by process calculi
- approximation algorithms which enhance the accuracy of the approximation with each iteration, e.g. computing π
- specification of infinite data structures, e.g. streams

Example (Infinite lists)

\[ \mathcal{R}_{\text{nats}} = \begin{cases} \text{from}(x) \rightarrow x : \text{from}(s(x)) \\ \text{from}(0) \rightarrow^2 0 : 1 : \text{from}(2) \end{cases} \]
Non-Terminating Rewriting Systems

Termination guarantees that every reduction sequence leads to a normal form, i.e. a final outcome.

Non-terminating systems can be meaningful:

- modelling reactive systems, e.g. by process calculi
- approximation algorithms which enhance the accuracy of the approximation with each iteration, e.g. computing π
- specification of infinite data structures, e.g. streams

Example (Infinite lists)

\[ \mathcal{R}_{\text{nats}} = \left\{ \text{from}(x) \rightarrow x : \text{from}(s(x)) \right\} \]

\[ \text{from}(0) \rightarrow^3 0 : 1 : 2 : \text{from}(3) \]
Non-Terminating Rewriting Systems

Termination guarantees that every reduction sequence leads to a normal form, i.e. a final outcome.

Non-terminating systems can be meaningful

- modelling reactive systems, e.g. by process calculi
- approximation algorithms which enhance the accuracy of the approximation with each iteration, e.g. computing π
- specification of infinite data structures, e.g. streams

Example (Infinite lists)

\( \mathcal{R}_{nats} = \{ \text{from}(x) \to x : \text{from}(s(x)) \} \)

\( \text{from}(0) \to^4 0 : 1 : 2 : 3 : \text{from}(4) \)
Non-Terminating Rewriting Systems

Termination guarantees that every reduction sequence leads to a normal form, i.e. a final outcome.

Non-terminating systems can be meaningful

- modelling reactive systems, e.g. by process calculi
- approximation algorithms which enhance the accuracy of the approximation with each iteration, e.g. computing \( \pi \)
- specification of infinite data structures, e.g. streams

Example (Infinite lists)

\[ R_{nats} = \left\{ \text{from}(x) \rightarrow x : \text{from}(s(x)) \right\} \]

\[ \text{from}(0) \rightarrow^5 0 : 1 : 2 : 3 : 4 : \text{from}(5) \]
Non-Terminating Rewriting Systems

Termination guarantees that every reduction sequence leads to a normal form, i.e. a final outcome.

Non-terminating systems can be meaningful

- modelling reactive systems, e.g. by process calculi
- approximation algorithms which enhance the accuracy of the approximation with each iteration, e.g. computing $\pi$
- specification of infinite data structures, e.g. streams

Example (Infinite lists)

$$\mathcal{R}_{nats} = \left\{ \text{from}(x) \rightarrow x : \text{from}(s(x)) \right\}$$

$$\text{from}(0) \rightarrow^6 0 : 1 : 2 : 3 : 4 : 5 : \text{from}(6)$$
Non-Terminating Rewriting Systems

Termination guarantees that every reduction sequence leads to a normal form, i.e. a final outcome.

Non-terminating systems can be meaningful

- modelling reactive systems, e.g. by process calculi
- approximation algorithms which enhance the accuracy of the approximation with each iteration, e.g. computing $\pi$
- specification of infinite data structures, e.g. streams

Example (Infinite lists)

$$R_{nats} = \left\{ \text{from}(x) \rightarrow x : \text{from}(s(x)) \right\}$$

$$\text{from}(0) \rightarrow^6 0 : 1 : 2 : 3 : 4 : 5 : \text{from}(6) \rightarrow \ldots$$
Non-Terminating Rewriting Systems

Termination guarantees that every reduction sequence leads to a normal form, i.e. a final outcome.

Non-terminating systems can be meaningful

- modelling reactive systems, e.g. by process calculi
- approximation algorithms which enhance the accuracy of the approximation with each iteration, e.g. computing $\pi$
- specification of infinite data structures, e.g. streams

Example (Infinite lists)

\[ R_{nats} = \{ \text{from}(x) \rightarrow x : \text{from}(s(x)) \} \]

\[ \text{from}(0) \rightarrow^6 0 : 1 : 2 : 3 : 4 : 5 : \text{from}(6) \rightarrow \ldots \]

Intuitively this converges to the infinite list $0 : 1 : 2 : 3 : 4 : 5 : \ldots$. 
What is infinitary rewriting?

- formalises the outcome of an infinite reduction sequence
- allows reduction sequences of any ordinal number length
- deals with (potentially) infinite terms
Infinitary Rewriting

What is infinitary rewriting?
- formalises the outcome of an infinite reduction sequence
- allows reduction sequences of any ordinal number length
- deals with (potentially) infinite terms

Why consider infinitary rewriting?
- model for lazy functional programming
- semantics for non-terminating systems
- semantics for process algebras
- arises in cyclic term graph rewriting
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:

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

\[ \text{sim}(s, t) = \text{minimum depth } d \]
\[ \text{s.t. } s \text{ and } t \text{ differ at depth } d \]
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:
  \[ d(s, t) = 2^{-\text{sim}(s, t)} \]

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

Example

\[
\begin{array}{ccc}
  & f & \\
  a & f & a \\
  & b & c & g \\
     & s & & t \\
\end{array}
\]
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:

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

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

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

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

Example

\[ d(s, t) = 2^{-1} \]
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:
  \[ d(s, t) = 2^{-\text{sim}(s, t)} \]

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

Example

\[
\begin{align*}
\text{d}(s, t) &= 2^{-1} \\
\text{d}(u, v) &= 2^{-2}
\end{align*}
\]
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:
  \[ d(s, t) = 2^{-\min\{d|s\neq t, s\text{ and } t\text{ differ at depth } d\}} \]

Example

\[
\begin{align*}
\text{Example} & \\
\text{sim}(s, t) &= \min \text{ depth } d \text{ s.t. } s \text{ and } t \text{ differ at depth } d \\
\end{align*}
\]
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:

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

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

Example

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

\[ d(u, v) = 2^{-2} \]
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:

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

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

**Example**

Example:

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

\[ d(u, v) = 2^{-2} \]
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:

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

\[ \text{sim}(s, t) = \text{minimum depth } d \text{ s.t. } s \text{ and } t \text{ differ at depth } d \]
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:
  \[ d(s, t) = 2^{-\text{sim}(s,t)} \]
  \[ \text{sim}(s, t) = \text{minimum depth } d \text{ s.t. } s \text{ and } t \text{ differ at depth } d \]
  \[ \text{sim}(s, t) = \text{maximum depth } d \text{ s.t. truncated at depth } d, s \text{ and } t \text{ are equal} \]

Example

1 level

\[ d(f, f) = 2^{-1} \]

2 levels

\[ d(f, f) = 2^{-2} \]
Weak Convergence of Transfinite Reductions

Weak convergence via metric \( d \)

- convergence in the metric space \( (\mathcal{T}^\infty(\Sigma, \mathcal{V}), d) \)
- \textbf{depth of the differences} between the terms has to tend to infinity
Example: Weak Convergence

\[ \text{from} \quad \rightarrow \quad \text{from}(s(x)) \]
Example: Weak Convergence

\[ \text{from} \quad \rightarrow \quad \text{from} \]

\[ 0 \quad \rightarrow \quad 1 \]

\[ \text{from}(x) \rightarrow x : \text{from}(s(x)) \]
Example: Weak Convergence

\[ \text{from}(x) \rightarrow x : \text{from}(s(x)) \]
Example: Weak Convergence

\[ \text{from}(x) \to x : \text{from}(s(x)) \]
Example: Weak Convergence

\[ \text{from} \]

\[
\begin{array}{c}
\text{from} \\
\Downarrow \\
0 \\
\end{array}
\]

\[
\begin{array}{c}
\text{from} \\
\uparrow \\
0 \\
\end{array} \quad 1 \text{ level} \quad \begin{array}{c}
\text{from} \\
\downarrow \\
1 \\
\end{array}
\]

\[
\begin{array}{c}
\text{from} \\
\leftarrow \\
2 \\
\end{array} \quad \begin{array}{c}
\text{from} \\
\rightarrow \\
3 \\
\end{array}
\]

\[
\text{from}(x) \rightarrow x : \text{from}(s(x))
\]
Example: Weak Convergence

\[ \text{from} (x) \rightarrow x : \text{from} (s(x)) \]
Example: Weak Convergence

\[ \text{from}(x) \rightarrow x : \text{from}(s(x)) \]
Example: Weak Convergence

\[ \text{from}(x) \rightarrow x : \text{from}(s(x)) \]
Example: Weak Convergence

\[ \text{from } 0 \rightarrow x : \text{from}(s(x)) \]
Example: Weak Convergence

\[ \text{from} \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \rightarrow x : \text{from}(s(x)) \]
Transfinite Reductions

Example (Infinite lists)

\[ R_{zip} = \begin{cases} 
    zip(nil, y) & \rightarrow \text{nil} \\
    zip(x, nil) & \rightarrow \text{nil} \\
    zip(x : x', y : y') & \rightarrow (x, y) : zip(x', y') 
\end{cases} \]
Transfinite Reductions

Example (Infinite lists)

\[ R_{zip} = \begin{cases} 
  \text{zip}(\text{nil}, y) \rightarrow \text{nil} \\
  \text{zip}(x, \text{nil}) \rightarrow \text{nil} \\
  \text{zip}(x : x', y : y') \rightarrow (x, y) : \text{zip}(x', y') 
\end{cases} \]

\[ \text{zip(from}(0), a : b : c : \text{nil}) \]
Transfinite Reductions

Example (Infinite lists)

\[ R_{zip} = \begin{cases} 
    \text{zip}(\text{nil}, y) \rightarrow \text{nil} \\
    \text{zip}(x, \text{nil}) \rightarrow \text{nil} \\
    \text{zip}(x : x', y : y') \rightarrow (x, y) : \text{zip}(x', y') 
\end{cases} \]

\[
\text{zip}(\text{from}(0), a : b : c : \text{nil}) \xrightarrow{\omega} \text{zip}(0 : 1 : 2 : 3 : 4 \ldots, a : b : c : \text{nil})
\]
Transfinite Reductions

Example (Infinite lists)

\[ R_{\text{zip}} = \begin{cases} 
    \text{zip}(\text{nil}, y) \rightarrow \text{nil} \\
    \text{zip}(x, \text{nil}) \rightarrow \text{nil} \\
    \text{zip}(x : x', y : y') \rightarrow (x, y) : \text{zip}(x', y') 
\end{cases} \]

\[ \text{zip}(\text{from}(0), a : b : c : \text{nil}) \rightarrow^\omega \text{zip}(0 : 1 : 2 : 3 : 4 \ldots, a : b : c : \text{nil}) \rightarrow (0, a) : \text{zip}(1 : 2 : 3 : 4 : \ldots, b : c : \text{nil}) \]
Transfinite Reductions

Example (Infinite lists)

\[ R_{zip} = \begin{cases} 
  \text{zip}(\text{nil}, y) \rightarrow \text{nil} \\
  \text{zip}(x, \text{nil}) \rightarrow \text{nil} \\
  \text{zip}(x : x', y : y') \rightarrow (x, y) : \text{zip}(x', y') 
\end{cases} \]

\[ \text{zip(from(0), a : b : c : nil)} \rightarrow^\omega \text{zip(0 : 1 : 2 : 3 : 4 : \ldots, a : b : c : nil)} \]
\[ \rightarrow (0, a) : \text{zip(1 : 2 : 3 : 4 : \ldots, b : c : nil)} \]
\[ \rightarrow (0, a) : (1, b) : \text{zip(2 : 3 : 4 : \ldots, c : nil)} \]
Transfinite Reductions

Example (Infinite lists)

\[ R_{\text{zip}} = \begin{cases} 
zip(\text{nil}, y) & \rightarrow \text{nil} \\
zip(x, \text{nil}) & \rightarrow \text{nil} \\
zip(x : x', y : y') & \rightarrow (x, y) : zip(x', y') 
\end{cases} \]

\[ zip(\text{from}(0), a : b : c : \text{nil}) \xrightarrow{\omega} zip(0 : 1 : 2 : 3 : 4 \ldots, a : b : c : \text{nil}) \]
\[ \rightarrow (0, a) : zip(1 : 2 : 3 : 4 \ldots, b : c : \text{nil}) \]
\[ \rightarrow (0, a) : (1, b) : zip(2 : 3 : 4 \ldots, c : \text{nil}) \]
\[ \rightarrow (0, a) : (1, b) : (2, c) : zip(3 : 4 \ldots, \text{nil}) \]
Transfinite Reductions

Example (Infinite lists)

\[ R_{zip} = \begin{cases} 
zip(nil, y) & \rightarrow \text{nil} \\
zip(x, nil) & \rightarrow \text{nil} \\
zip(x : x', y : y') & \rightarrow (x, y) : zip(x', y') 
\end{cases} \]

\[ zip(from(0), a : b : c : nil) \rightarrow^\omega \text{zip}(0 : 1 : 2 : 3 : 4 : \ldots, a : b : c : nil) \]

\[ \rightarrow (0, a) : \text{zip}(1 : 2 : 3 : 4 : \ldots, b : c : nil) \]

\[ \rightarrow (0, a) : (1, b) : \text{zip}(2 : 3 : 4 : \ldots, c : nil) \]

\[ \rightarrow (0, a) : (1, b) : (2, c) : \text{zip}(3 : 4 : \ldots, nil) \]

\[ \rightarrow (0, a) : (1, b) : (2, c) : \text{nil} \]
Transfinite Reductions

Example (Infinite lists)

\[ \mathcal{R}_{\text{zip}} = \begin{cases} 
\text{zip}(\text{nil}, y) \to \text{nil} \\
\text{zip}(x, \text{nil}) \to \text{nil} \\
\text{zip}(x : x', y : y') \to (x, y) : \text{zip}(x', y')
\end{cases} \]

\[
\text{zip}(\text{from}(0), a : b : c : \text{nil}) \to^\omega \text{zip}(0 : 1 : 2 : 3 : 4 \ldots, a : b : c : \text{nil}) \\
\to (0, a) : \text{zip}(1 : 2 : 3 : 4 \ldots, b : c : \text{nil}) \\
\to (0, a) : (1, b) : \text{zip}(2 : 3 : 4 \ldots, c : \text{nil}) \\
\to (0, a) : (1, b) : (2, c) : \text{zip}(3 : 4 \ldots, \text{nil}) \\
\to (0, a) : (1, b) : (2, c) : \text{nil}
\]

final outcome is a **finite term**!
Strong Convergence of Transfinite Reductions

Weak convergence is hard to deal with

- there might be terms only reachable after more than $\omega$ steps
- orthogonal systems are not confluent
- not necessarily normalising
Strong Convergence of Transfinite Reductions

Weak convergence is hard to deal with

- there might be terms only reachable after more than ω steps
- orthogonal systems are not confluent
- not necessarily normalising

Strong convergence via increasing redex depth

- conservative underapproximation of convergence in the metric space
- rewrite rules have to be applied at (eventually) increasingly large depth
- the limit is then defined by the metric space
  \[ \rightsquigarrow \]
- for strong convergence the depth of redexes has to tend to infinity
Example: Weakly but not Strongly Converging

\[ f(g(x)) \rightarrow f(g(g(x))) \]
Example: Weakly but not Strongly Converging

\[ f(g(x)) \rightarrow f(g(g(x))) \]
Example: Weakly but not Strongly Converging

\[ f(g(x)) \rightarrow f(g(g(x))) \]
Example: Weakly but not Strongly Converging

\[ f(g(x)) \rightarrow f(g(g(x))) \]
Example: Weakly but not Strongly Converging

\[ f(g(x)) \rightarrow f(g(g(x))) \]
Example: Weakly but not Strongly Converging

\[ f(g(x)) \rightarrow f(g(g(x))) \]
Example: Weakly but not Strongly Converging

\[ f(g(x)) \rightarrow f(g(g(x))) \]
Example: Weakly but not Strongly Converging

\[ f(g(x)) \rightarrow f(g(g(x))) \]
Example: Weakly but not Strongly Converging

\[ f(g(x)) \rightarrow f(g(g(x))) \]
Example: Weakly but not Strongly Converging

\[ f(g(x)) \rightarrow f(g(g(x))) \]
Example: Weakly but not Strongly Converging

\[ f(g(x)) \rightarrow f(g(g(x))) \]
Example: Weakly and Strongly Converging

\[ g(c) \rightarrow g(g(c)) \]
Example: Weakly and Strongly Converging

\[ g(c) \rightarrow g(g(c)) \]
Example: Weakly and Strongly Converging

\[ g(c) \rightarrow g(g(c)) \]
Example: Weakly and Strongly Converging

\[ g(c) \rightarrow g(g(c)) \]
Example: Weakly and Strongly Converging

\[ g(c) \rightarrow g(g(c)) \]
Example: Weakly and Strongly Converging

\[ g(c) \rightarrow g(g(c)) \]
Example: Weakly and Strongly Converging

\[ g(c) \rightarrow g(g(c)) \]
Example: Weakly and Strongly Converging

\[ g(c) \rightarrow g(g(c)) \]
Example: Weakly and Strongly Converging

\[ g(c) \rightarrow g(g(c)) \]
Example: Weakly and Strongly Converging

\[ g(c) \rightarrow g(g(c)) \]
Example: Weakly and Strongly Converging

$g(c) \rightarrow g(g(c))$
Example: Weakly and Strongly Converging

\[ g(c) \rightarrow g(g(c)) \]
Outline

1. Infinitary Term Rewriting

2. Term Graph Rewriting
   - Partial Order Model of Infinitary Rewriting
   - Convergence on Term Graphs

3. Outlook
Moving to Term Graphs – Why?

Simulating infinitary term rewriting

- term graphs allow to **finitely represent** rational terms
- certain infinite term reductions can be represented as finite term graph reductions [Kennaway et al.]
- infinitary term rewriting $\Leftrightarrow$ cyclic term graph rewriting?
Moving to Term Graphs – Why?

**Simulating infinitary term rewriting**

- Term graphs allow to **finitely represent** rational terms
- Certain infinite term reductions can be represented as finite term graph reductions [Kennaway et al.]
- Infinitary term rewriting \(\Leftrightarrow\) cyclic term graph rewriting?

**Calculi with explicit sharing and recursion**

- Adding **letrec** to \(\lambda\)-calculus breaks confluence
- However: unique **infinite normal forms** can be defined [Ariola & Blom]
- Infinitary confluence?
Moving to Term Graphs – Why?

Simulating infinitary term rewriting

- term graphs allow to \textit{finitely represent} rational terms
- certain infinite term reductions can be represented as finite term graph reductions [Kennaway et al.]
- infinitary term rewriting $\Leftrightarrow$ cyclic term graph rewriting?

Calculi with explicit sharing and recursion

- adding \texttt{letrec} to $\lambda$-calculus breaks confluence
- however: unique \textit{infinite normal forms} can be defined [Ariola & Blom]
- infinitary confluence?

We need a infinitary rewriting counterpart on term graphs!
## Convergence on Term Graph Reductions – How?

### A metric on term graphs?

- a metric seems too "unstructured" for the rich structure of term graphs
- how should sharing be captured by the metric?
- what is an appropriate notion of depth in a term graph?
Convergence on Term Graph Reductions – How?

A metric on term graphs?
- A metric seems too "unstructured" for the rich structure of term graphs.
- How should sharing be captured by the metric?
- What is an appropriate notion of depth in a term graph?

Example
A metric on term graphs?

- A metric seems too "unstructured" for the rich structure of term graphs.
- How should sharing be captured by the metric?
- What is an appropriate notion of depth in a term graph?

Example
Convergence on Term Graph Reductions – How?

A metric on term graphs?
- a metric seems too "unstructured" for the rich structure of term graphs
- how should sharing be captured by the metric?
- what is an appropriate notion of depth in a term graph?

Example

![Diagram of a triangle with depth label n]
Convergence on Term Graph Reductions – How?

A metric on term graphs?

- a metric seems too “unstructured” for the rich structure of term graphs
- how should sharing be captured by the metric?
- what is an appropriate notion of depth in a term graph?

Example
Convergence on Term Graph Reductions – How?

A metric on term graphs?
- A metric seems too “unstructured” for the rich structure of term graphs.
- How should sharing be captured by the metric?
- What is an appropriate notion of depth in a term graph?

Example
Convergence on Term Graph Reductions – How?

A metric on term graphs?

- a metric seems too “unstructured” for the rich structure of term graphs
- how should sharing be captured by the metric?
- what is an appropriate notion of depth in a term graph?

Example
Convergence on Term Graph Reductions – How?

A metric on term graphs?
- a metric seems too "unstructured" for the rich structure of term graphs
- how should sharing be captured by the metric?
- what is an appropriate notion of depth in a term graph?

Example

![Diagram showing depth comparison]

\[ d' < d \]
Reconsidering Infinitary Term Rewriting

Infinitary rewriting on terms “more structure”

- the metric on terms is beautifully simple
- it is just enough for convergence on terms
Reconsidering Infinitary Term Rewriting

Infinitary rewriting on terms “more structure”
- the metric on terms is beautifully simple
- it is just enough for convergence on terms

More structure on term graphs
- for term graphs, we need more structure
- but: maybe we can obtain a metric space in the end
Reconsidering Infinitary Term Rewriting

Infinitary rewriting on terms “more structure”
- the metric on terms is beautifully simple
- it is just enough for convergence on terms

More structure on term graphs
- for term graphs, we need more structure
- but: maybe we can obtain a metric space in the end

Infinitary term rewriting with more structure
- borrowing from domain theory
- partial orders have been widely used to obtain a more structure view on terms
Partial Order Model of Infinitary Rewriting

Described on the example of terms

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 ($= \text{cpo} + \text{glbs of non-empty sets}$)
Partial Order Model of Infinitary Rewriting

Described on the example of terms

Partial order on terms

- **partial terms**: terms with additional constant ⊥ (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**:

\[
\liminf_{\iota \to \alpha} t_{\iota} = \bigsqcup_{\beta < \alpha} \bigcap_{\beta \leq \iota < \alpha} t_{\iota}
\]

- intuition: eventual persistence of nodes of the terms
- weak convergence: limit inferior of the **terms** of the reduction
- strong convergence: limit inferior of the **contexts** of the reduction
An Example

Reduction sequence for \( f(x, y) \rightarrow f(y, x) \)
An Example

Reduction sequence for $f(x, y) \rightarrow f(y, x)$
An Example

Reduction sequence for $f(x, y) \rightarrow f(y, x)$
An Example

Reduction sequence for $f(x, y) \rightarrow f(y, x)$
An Example

Reduction sequence for $f(x, y) \rightarrow f(y, x)$
An Example

Reduction sequence for $f(x, y) \rightarrow f(y, x)$
An Example

Reduction sequence for \( f(x, y) \rightarrow f(y, x) \)

Weak convergence
An Example

Reduction sequence for $f(x, y) \rightarrow f(y, x)$

Weak convergence
An Example

Reduction sequence for $f(x, y) \rightarrow f(y, x)$

Weak convergence

Strong convergence
Properties of the Partial Order Model on Terms

Benefits

- more fine-grained than the metric model
- more intuitive than the metric model
- subsumes metric model
Properties of the Partial Order Model on Terms

Benefits

- more fine-grained than the metric model
- more intuitive than the metric model
- subsumes metric model

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

For every reduction $S$ in a TRS the following equivalences hold:

1. $S: s \xrightarrow{p} t$ is total iff $S: s \xrightarrow{m} t.$ (weak convergence)
2. $S: s \xrightarrow{p} t$ is total iff $S: s \xrightarrow{m} t.$ (strong 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 \( \leq \) on term trees?
- We need a means to substitute \( \bot \)’s.
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?
- We need a means to substitute $\bot$'s.

$\bot$-homomorphisms $\varphi: g \to \bot h$

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

$\varphi: g \rightarrow h$
A \perp\text{-Homomorphism}

\[
\varphi: \begin{array}{c}
g \\
\perp \\
\perp
\end{array} \to \begin{array}{c}
\perp \\
g \\
g
\end{array}
\]
\(\bot\)-Homomorphisms as a Partial Order

**Proposition (partial order on terms)**

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

\[ s \leq_\bot t \text{ iff } \exists \varphi: s \rightarrow_\bot t \]
**⊥-Homomorphisms as a Partial Order**

**Proposition (partial order on terms)**

For all $s, t \in \mathcal{T}^\infty(\Sigma_\bot)$: $s \leq_\bot t$ iff $\exists \varphi: s \rightarrow_\bot t$

**Theorem**

For all $g, h \in \mathcal{G}^\infty(\Sigma_\bot)$, let $g \leq^1_\bot h$ be defined iff there is some $\varphi: g \rightarrow_\bot h$.

The pair $(\mathcal{G}_C^\infty(\Sigma_\bot), \leq^1_\bot)$ forms a **complete semilattice**.
\(-\)-Homomorphisms as a Partial Order

**Proposition (partial order on terms)**

For all \(s, t \in T^\infty(\Sigma_\bot)\):

\[ s \leq_\bot t \iff \exists \varphi: s \to_\bot t \]

**Theorem**

For all \(g, h \in G^\infty(\Sigma_\bot)\), let \(g \leq^1_\bot h\) be defined iff there is some \(\varphi: g \to_\bot h\).

The pair \((G^\infty_C(\Sigma_\bot), \leq^1_\bot)\) forms a complete semilattice.

**Alas, \(\leq^1_\bot\) has some quirks!**

\[
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\text{f} \\
\text{c} \\
\text{f}
\end{array}
\end{array}
\end{array}
\]

\[
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\text{c} \\
\text{c}
\end{array}
\end{array}
\end{array}
\]

\[
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\text{c} \\
\text{c}
\end{array}
\end{array}
\end{array}
\]
\(\perp\)-Homomorphisms as a Partial Order

**Proposition (partial order on terms)**

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

\[s \leq_\perp t \text{ iff } \exists \varphi : s \rightarrow_\perp t\]

**Theorem**

For all \(g, h \in G^\infty(\Sigma_\perp)\), let \(g \leq^1_\perp h\) be defined iff there is some \(\varphi : g \rightarrow_\perp h\).

The pair \((G^\infty_C(\Sigma_\perp), \leq^1_\perp)\) forms a complete semilattice.

Alas, \(\leq^1_\perp\) has some quirks!

\[
\begin{array}{ccc}
\xymatrix{& f \\
\& c \\
& \leq^1_\perp \\
& \& f \\
\& c \\
\& \& c
\end{array}
\]
Proposition (partial order on terms)

For all $s, t \in \mathcal{T}^{\infty}(\Sigma_{\bot})$: $s \leq_{\bot} t$ iff $\exists \varphi: s \rightarrow_{\bot} t$

Theorem

For all $g, h \in \mathcal{G}^{\infty}(\Sigma_{\bot})$, let $g \leq_{\bot}^{1} h$ be defined iff there is some $\varphi: g \rightarrow_{\bot} h$.

The pair $(\mathcal{G}^{\infty}_{C}(\Sigma_{\bot}), \leq_{\bot}^{1})$ forms a complete semilattice.

Alas, $\leq_{\bot}^{1}$ has some quirks!

- introduces sharing
- total term graphs not necessarily maximal
- but: we should not dismiss it too fast!
Avoiding Sharing

Definition (injective \( \bot \)-homomorphisms)

For all \( g, h \in G^\infty(\Sigma_\bot) \), let \( g \leq_2 \bot h \) be defined iff there is some \( \varphi: g \rightarrow \bot h \) injective on all (non-\( \bot \)-) nodes.
Avoiding Sharing

Definition (injective $\bot$-homomorphisms)

For all $g, h \in G^\infty(\Sigma_\bot)$, let $g \leq_2^\bot h$ be defined iff there is some $\varphi: g \rightarrow_\bot h$ injective on all (non-$\bot$-) nodes.

Greatest lower bounds w.r.t. $\leq_2^\bot$

\[
\begin{array}{c}
\text{f} \\
\downarrow \\
\text{c} \\
\end{array}
\quad
\begin{array}{c}
\text{f} \\
\downarrow \\
\text{c} \\
\end{array}
\quad
\begin{array}{c}
\text{f} \\
\uparrow \\
\text{c} \\
\end{array}
\
\begin{array}{c}
\text{c} \\
\end{array}
\quad
\begin{array}{c}
\text{c} \\
\end{array}
\quad
\begin{array}{c}
\text{c} \\
\end{array}
\]

In particular, $\leq_2^\bot$ is not a complete semilattice!
Avoiding Sharing

Definition (injective $\perp$-homomorphisms)

For all $g, h \in G^{\infty}(\Sigma_{\perp})$, let $g \leq_{\perp}^2 h$ be defined iff there is some $\varphi: g \rightarrow_{\perp} h$ injective on all (non-$\perp$-) nodes.

Greatest lower bounds w.r.t. $\leq_{\perp}^2$

$$\begin{array}{ccc}
f & \perp^2 & f \\
c & c & c \\
\end{array} = ?$$
Avoiding Sharing

Definition (injective $\bot$-homomorphisms)
For all $g, h \in \mathcal{G}_\infty(\Sigma_\bot)$, let $g \leq_{\bot}^2 h$ be defined iff there is some $\varphi: g \rightarrow_{\bot} h$ injective on all (non-$\bot$-) nodes.

Greatest lower bounds w.r.t. $\leq_{\bot}^2$

\[
\begin{array}{c}
\text{f} \\
\downarrow \\
\text{c} \\
\end{array}
\quad
\begin{array}{c}
\text{f} \\
\downarrow \\
\text{c} \\
\end{array}
\quad
\begin{array}{c}
\text{f} \\
\downarrow \\
\bot \\
\end{array}
\quad
\begin{array}{c}
\text{f} \\
\downarrow \\
\text{c} \\
\end{array}
\quad
\begin{array}{c}
\text{f} \\
\downarrow \\
\text{c} \\
\end{array}
\quad
\begin{array}{c}
\text{f} \\
\downarrow \\
\bot \\
\end{array}
\]

In particular, $\leq_{\bot}^2$ is not a complete semilattice!
Avoiding Sharing

Definition (injective $\perp$-homomorphisms)
For all $g, h \in G^\infty(\Sigma_{\perp})$, let $g \leq^2_{\perp} h$ be defined iff there is some $\varphi: g \rightarrow_{\perp} h$ injective on all (non-$\perp$-) nodes.

Greatest lower bounds w.r.t. $\leq^2_{\perp}$

```
f
  /\c
 c   c
 ,   ,
```

```
f
  /\c
 c   c
≥^2_{\perp}
```

```
f
  /\c
 c   c
 ,   ,
```

```
f
  /\⊥
 ⊥   ⊥
```

In particular, $\leq^2_{\perp}$ is not a complete semilattice!
Avoiding Sharing

Definition (injective \(\bot\)-homomorphisms)

For all \(g, h \in \mathcal{G}^\infty(\Sigma_\bot)\), let \(g \leq^2_\bot h\) be defined iff there is some \(\varphi: g \rightarrow_\bot h\) injective on all (non-\(\bot\)) nodes.

Greatest lower bounds w.r.t. \(\leq^2_\bot\)

In particular, \(\leq^2_\bot\) is not a complete semilattice!
Avoiding Sharing

Definition (injective \( \bot \)-homomorphisms)
For all \( g, h \in G^\infty(\Sigma_\bot) \), let \( g \leq^2_\bot h \) be defined iff there is some \( \varphi: g \to_\bot h \) injective on all (non-\( \bot \)-) nodes.

Greatest lower bounds w.r.t. \( \leq^2_\bot \)

In particular, \( \leq^2_\bot \) is not a complete semilattice!

Theorem

The pair \( (G^\infty_\Sigma(\Sigma_\bot), \leq^2_\bot) \) forms a complete partial order.
Maintaining Sharing

Goal

\[ g \preceq^G \bot h \text{ iff } g \text{ is isomorphic to initial part of } h \text{ above } \bot \text{'s in } g \]
Maintaining Sharing

Goal

\[ g \leq^G \bot h \text{ iff } g \text{ is isomorphic to initial part of } h \text{ above } \bot \text{'s in } g \]

What is sharing?

a node \( n \) is shared if it is reachable via multiple paths from the root

the set of all paths \( P(g(n)) \) to node describes its sharing
Maintaining Sharing

Goal

\( g \leq^G \perp h \) iff \( g \) is isomorphic to initial part of \( h \) above \( \perp \)'s in \( g \)

1. \( g \)
2. \( h \)
Maintaining Sharing

**Goal**

\[ g \leq^G \perp h \iff g \text{ is isomorphic to initial part of } h \text{ above } \perp \text{'s in } g \]

Diagram:

- Node \( g \) is shared if it is reachable via multiple paths from the root.
- The set of all paths \( P_g(n) \) to node describes its sharing.

Diagram illustrating the relationship between \( g \) and \( h \) with \( \leq^G \perp \).
Maintaining Sharing

Goal

\[ g \preceq^G h \iff g \text{ is isomorphic to initial part of } h \text{ above } \bot \text{'s in } g \]

What is sharing?

A node is shared if it is reachable via multiple paths from the root.

The set of all paths \( P_g(n) \) to a node describes its sharing.
Maintaining Sharing

Goal

\[ g \leq^g_\bot h \iff g \text{ is isomorphic to initial part of } h \text{ above } \bot \text{'s in } g \]

What is sharing?

A node \( n \) is shared if it is reachable via multiple paths from the root.

The set of all paths \( P_g(n) \) to node describes its sharing.

Diagram:
- Node \( g \)
- Node \( h \)
- Arrow indicating preservation of sharing from \( g \) to \( h \)

Maintaining Sharing

Goal

\[ g \preceq^G h \text{ iff } g \text{ is isomorphic to initial part of } h \text{ above } ' \perp ' \text{’s in } g \]

What is sharing?

- A node \( n \) is shared if it is reachable via \textbf{multiple paths} from the root
- The set of all paths \( P_g(n) \) to a node describes its sharing
Sharing-Preserving $\bot$-homomorphisms

**Definition**

For all $g, h \in G^\infty(\Sigma_\bot)$, let $g \leq^3 \bot h$ be defined iff there is some $\varphi: g \rightarrow \bot h$ with $P_g(n) = P_h(\varphi(n))$ for all non-$\bot$-nodes $n$ in $g$. 

Theorem

The pair $(G^\infty(\Sigma_\bot), \leq^3 \bot)$ forms a complete semilattice.

$\leq^3 \bot$ is quite restrictive!
Sharing-Preserving \( \bot \)-homomorphisms

**Definition**

For all \( g, h \in G^\infty(\Sigma_\bot) \), let \( g \leq^\bot_3 h \) be defined iff there is some \( \varphi : g \rightarrow \bot h \) with \( P_g(n) = P_h(\varphi(n)) \) for all non-\( \bot \)-nodes \( n \) in \( g \).

**Theorem**

The pair \( (G_C^\infty(\Sigma_\bot), \leq^\bot_3) \) forms a *complete semilattice*.
Sharing-Preserving \( \bot \)-homomorphisms

**Definition**
For all \( g, h \in \mathcal{G}^\infty(\Sigma_\bot) \), let \( g \leq^3_{\bot} h \) be defined iff there is some \( \varphi : g \to_{\bot} h \) with \( P_g(n) = P_h(\varphi(n)) \) for all non-\( \bot \)-nodes \( n \) in \( g \).

**Theorem**
The pair \( (\mathcal{G}_C^\infty(\Sigma_\bot), \leq^3_{\bot}) \) forms a complete semilattice.

\( \leq^3_{\bot} \) is quite restrictive!
Sharing-Preserving $\bot$-homomorphisms

**Definition**
For all $g, h \in G^\infty(\Sigma_\bot)$, let $g \leq^3 \bot h$ be defined iff there is some $\varphi: g \to \bot h$ with $P_g(n) = P_h(\varphi(n))$ for all non-$\bot$-nodes $n$ in $g$.

**Theorem**
The pair $(G^\infty_C(\Sigma_\bot), \leq^3_\bot)$ forms a complete semilattice.

$\leq^3_\bot$ is quite restrictive!

```
  h    h
 /\   /\  \\
 h  h  h
```

Sharing-Preserving \(\bot\)-homomorphisms

**Definition**

For all \(g, h \in G^\infty(\Sigma_\bot)\), let \(g \leq^3_\bot h\) be defined iff there is some \(\varphi : g \rightarrow_\bot h\) with \(P_g(n) = P_h(\varphi(n))\) for all non-\(\bot\)-nodes \(n\) in \(g\).

**Theorem**

The pair \((G^\infty_C(\Sigma_\bot), \leq^3_\bot)\) forms a complete semilattice.

\(\leq^3_\bot\) is quite restrictive!
Sharing-Preserving \( \perp \)-homomorphisms

**Definition**

For all \( g, h \in G^\infty(\Sigma_\perp) \), let \( g \leq^3 \perp h \) be defined iff there is some \( \varphi : g \rightarrow \perp h \) with \( P_g(n) = P_h(\varphi(n)) \) for all non-\( \perp \)-nodes \( n \) in \( g \).

**Theorem**

The pair \((G^\infty_C(\Sigma_\perp), \leq^3 \perp)\) forms a complete semilattice.

\( \leq^3 \perp \) is quite restrictive!

![Diagram](image)
Sharing-Preserving $\bot$-homomorphisms

**Definition**

For all $g, h \in G^\infty(\Sigma_\bot)$, let $g \leq_\bot^3 h$ be defined iff there is some $\varphi : g \to h$ with $P_g(n) = P_h(\varphi(n))$ for all non-$\bot$-nodes $n$ in $g$.

**Theorem**

The pair $(G^\infty_C(\Sigma_\bot), \leq_\bot^3)$ forms a complete semilattice.

$\leq_\bot^3$ is quite restrictive!

---

Diagram of nodes and edges illustrating the relations and the structure of the semilattice.
Acyclic Sharing

**Acyclic Paths**

We only consider the set $\mathcal{P}_g^a(n)$ of **minimal paths** to $n$. 
Acyclic Sharing

**Acyclic Paths**

We only consider the set $\mathcal{P}_g^a(n)$ of minimal paths to $n$.

**Definition**

For all $g, h \in G^\infty(\Sigma_\bot)$, let $g \preceq_4 h$ be defined iff there is some $\varphi: g \rightarrow_\bot h$ with $\mathcal{P}_g(n) = \mathcal{P}_g(\varphi(n))$ for all non-$\bot$-nodes $n$ in $g$. 
Acyclic Sharing

Acyclic Paths
We only consider the set \( P_g^a(n) \) of minimal paths to \( n \).

Definition
For all \( g, h \in G^\infty(\Sigma_\bot) \), let \( g \leq_{\bot}^4 h \) be defined iff there is some \( \varphi: g \to_{\bot} h \) with \( P_g(n) = P_g(\varphi(n)) \) for all non-\( \bot \)-nodes \( n \) in \( g \).

Theorem
The pair \( (G^\infty_C(\Sigma_\bot), \leq_{\bot}^4) \) forms a complete semilattice.
What Have We Gained?

**Insight into convergence over term graphs**

- Partial orders honour the rich structure of term graphs.
- All discussed partial orders specialise to $\leq \bot$ on terms.

**Theorem (total $p$-convergence = weak convergence)**

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

$S: g \hookrightarrow h$ is total if $S: g \hookrightarrow h$. (weak convergence)
What Have We Gained?

Insight into convergence over term graphs
- partial orders honour the rich structure of term graphs
- all discussed partial orders specialise to $\leq_{\bot}$ on terms

complete semilattices induce a complete metric space
- complete semilattices induce a canonical metric (except for $\leq_{\bot}^1$)
- common structure of two term graphs $g$ and $h$: $g \sqcap_{\bot} h$
- metric distance $d(g, h) = 2^{-d}$, where $d = \bot$-depth($g \sqcap_{\bot} h$)
- resulting complete metric specialises to the metric $d$ on terms
What Have We Gained?

Insight into convergence over term graphs
- partial orders honour the rich structure of term graphs
- all discussed partial orders specialise to $\leq_{\bot}$ on terms

Complete semilattices induce a complete metric space
- complete semilattices induce a canonical metric (except for $\leq_{1_{\bot}}$)
- common structure of two term graphs $g$ and $h$: $g \sqcap_{\bot} h$
- metric distance $d(g, h) = 2^{-d}$, where $d = \bot$-depth($g \sqcap_{\bot} h$)
- resulting complete metric specialises to the metric $d$ on terms

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

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

$S : g \xrightarrow{p} h$ is total iff $S : g \xrightarrow{m} h$. (weak convergence)
Next Steps

Partial order $\leq_{1\bot}$ based on $\bot$-homomorphisms

- it behaves weird but it might still be suited for convergence e.g.
Partial order $\leq_1^\perp$ based on $\perp$-homomorphisms

- it behaves weird but it might still be suited for convergence, e.g.

```
from
\downarrow
0
```
Next Steps

Partial order $\leq^{1}_{\bot}$ based on $\bot$-homomorphisms

- it behaves weirdly but it might still be suited for convergence, e.g.

```
0   0   from
\downarrow
\quad \quad \quad \quad
\quad \quad \quad \quad
\quad \quad \quad \quad
\quad \quad \quad \quad
s
```
Next Steps

Partial order $\preceq_1$ based on $\perp$-homomorphisms

- it behaves weird but it might still be suited for convergence, e.g.

```
from 0 0 from 0:
    ↓  ↓  ↓  ↓
     s  s  s  s
```

Strong convergence on term graphs

what is a proper notion of strong convergence?

using the partial order approach might again be helpful
Next Steps

Partial order $\leq^1_\bot$ based on $\bot$-homomorphisms

- it behaves weirdly but it might still be suited for convergence, e.g.

\[
\begin{array}{cccc}
0 & \rightarrow & 0 & \rightarrow \\
\downarrow & & & \\
s & \rightarrow & s & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & \rightarrow & 0 & \rightarrow \\
\downarrow & & & \\
s & \rightarrow & s & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & \rightarrow & 0 & \rightarrow \\
\downarrow & & & \\
s & \rightarrow & s & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & \rightarrow & 0 & \rightarrow \\
\downarrow & & & \\
s & \rightarrow & s & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & \rightarrow & 0 & \rightarrow \\
\downarrow & & & \\
s & \rightarrow & s & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & \rightarrow & 0 & \rightarrow \\
\downarrow & & & \\
s & \rightarrow & s & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & \rightarrow & 0 & \rightarrow \\
\downarrow & & & \\
s & \rightarrow & s & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & \rightarrow & 0 & \rightarrow \\
\downarrow & & & \\
s & \rightarrow & s & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & \rightarrow & 0 & \rightarrow \\
\downarrow & & & \\
s & \rightarrow & s & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & \rightarrow & 0 & \rightarrow \\
\downarrow & & & \\
s & \rightarrow & s & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & \rightarrow & 0 & \rightarrow \\
\downarrow & & & \\
s & \rightarrow & s & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & \rightarrow & 0 & \rightarrow \\
\downarrow & & & \\
s & \rightarrow & s & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & \rightarrow & 0 & \rightarrow \\
\downarrow & & & \\
s & \rightarrow & s & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & \rightarrow & 0 & \rightarrow \\
\downarrow & & & \\
s & \rightarrow & s & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & \rightarrow & 0 & \rightarrow \\
\downarrow & & & \\
s & \rightarrow & s & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & \rightarrow & 0 & \rightarrow \\
\downarrow & & & \\
s & \rightarrow & s & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & \rightarrow & 0 & \rightarrow \\
\downarrow & & & \\
s & \rightarrow & s & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & \rightarrow & 0 & \rightarrow \\
\downarrow & & & \\
s & \rightarrow & s & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & \rightarrow & 0 & \rightarrow \\
\downarrow & & & \\
s & \rightarrow & s & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & \rightarrow & 0 & \rightarrow \\
\downarrow & & & \\
s & \rightarrow & s & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & \rightarrow & 0 & \rightarrow \\
\downarrow & & & \\
s & \rightarrow & s & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & \rightarrow & 0 & \rightarrow \\
\downarrow & & & \\
s & \rightarrow & s & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & \rightarrow & 0 & \rightarrow \\
\downarrow & & & \\
s & \rightarrow & s & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & \rightarrow & 0 & \rightarrow \\
\downarrow & & & \\
s & \rightarrow & s & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & \rightarrow & 0 & \rightarrow \\
\downarrow & & & \\
s & \rightarrow & s & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & \rightarrow & 0 & \rightarrow \\
\downarrow & & & \\
s & \rightarrow & s & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & \rightarrow & 0 & \rightarrow \\
\downarrow & & & \\
s & \rightarrow & s & \rightarrow \\
\end{array}
\]
Next Steps

Partial order $\leq^1 \bot$ based on $\bot$-homomorphisms

- it behaves weird but it might still be suited for convergence, e.g.
- is there a metric space counterpart?
Next Steps

Partial order $\leq_{\bot}$ based on $\bot$-homomorphisms

- it behaves weird but it might still be suited for convergence e.g.
- is there a metric space counterpart?

\[
\begin{array}{cccccc}
f & \rightarrow & f & \rightarrow & f & \rightarrow \ f \\
\downarrow & & \downarrow & & \downarrow & \\
c & & c & & c & \\
\end{array}
\]
Next Steps

Partial order $\leq^{1}_\perp$ based on $\perp$-homomorphisms

- it behaves weird but it might still be suited for convergence e.g.
- is there a metric space counterpart?

\[
\begin{array}{ccccccccc}
  & f & \rightarrow & f & \rightarrow & f & \rightarrow & f & \rightarrow & \ldots & f \\
  c & \rightarrow & c & \rightarrow & c & \rightarrow & c & \rightarrow & c & \rightarrow & c
\end{array}
\]
Next Steps

Partial order $\leq^1_{\bot}$ based on $\bot$-homomorphisms

- it behaves weird but it might still be suited for convergence e.g.
- is there a metric space counterpart?

\[
\begin{array}{ccccccc}
  & f & \rightarrow & f & \rightarrow & f & \rightarrow & f & \rightarrow & f & \rightarrow & f \\
  & c & \downarrow & c & \downarrow & c & \downarrow & c & \downarrow & c & \downarrow & c \\
  & c & \downarrow & c & \downarrow & c & \downarrow & c & \downarrow & c & \downarrow & c \\
  \end{array}
\]

Strong convergence on term graphs

- what is a proper notion of strong convergence?
- using the partial order approach might again be helpful
Outline

1. Infinitary Term Rewriting

2. Term Graph Rewriting
   - Partial Order Model of Infinitary Rewriting
   - Convergence on Term Graphs

3. Outlook
Back to Term Graph Rewriting

Partial order approach to infinitary term rewriting

- more fine grained notion of convergence
- reductions always converge \( \rightsquigarrow \) semantics
- naturally captures meaningless terms
Strong Convergence on Orthogonal System

**Metric convergence**

- Normal forms are *unique*.
- However: terms might have **no normal forms** (only reductions that do not converge).

Infinite confluence:

Every term has a normal form reachable by a possibly infinite reduction.

Unique normal forms!
Strong Convergence on Orthogonal System

Metric convergence

- normal forms are unique
- however: terms might have no normal forms (only reductions that do not converge)

With partial order model, we gain normalisation and thus confluence.

Infinitary confluence

```
  t ----> t_1
  |      |
  v      v
  t_2 ----> t_3
```

Infinitary normalisation

```
t --> t~
```

Every term has a normal form reachable by a possibly infinite reduction.
Strong Convergence on Orthogonal System

**Metric convergence**

- normal forms are **unique**
- however: terms might have **no normal forms** (only reductions that do not converge)

With partial order model, we gain normalisation and thus **confluence**.

**Infinitary confluence**

\[ t \xrightarrow{} t_1 \xrightarrow{} t_3 \]

\[ t \xrightarrow{} t_2 \xrightarrow{} t_3 \]

**Infinitary normalisation**

\[ t \xrightarrow{} \bar{t} \xrightarrow{} \]

Every term has a normal form reachable by a possibly infinite reduction.

Unique normal forms!
Meaningless Terms

<table>
<thead>
<tr>
<th>Böhm extensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Given a TRS $\mathcal{R}$, its Böhm extension $\mathcal{B}_\mathcal{R}$ is obtained by adding rules of the form $r \rightarrow \perp$, where $r$ are root-active terms</td>
</tr>
</tbody>
</table>
Böhm extensions

Given a TRS $\mathcal{R}$, its Böhm extension $\mathcal{B}_\mathcal{R}$ is obtained by adding rules of the form $r \rightarrow \bot$, where $r$ are root-active terms.

Unique normal forms

- Böhm extensions are used to obtain unique normal forms (Böhm trees).
- $\mathcal{B}_\mathcal{R}$ is infinitary normalising and confluent.
Meaningless Terms

Böhm extensions
Given a TRS $\mathcal{R}$, its Böhm extension $B_\mathcal{R}$ is obtained by adding rules of the form $r \rightarrow \bot$, where $r$ are root-active terms.

Unique normal forms
- Böhm extensions are used to obtain unique normal forms (Böhm trees)
- $B_\mathcal{R}$ is infinitary normalising and confluent

Theorem ($m$-convergence + Böhm extension = $p$-convergence)
If $\mathcal{R}$ is an orthogonal TRS and $B$ the Böhm extension of $\mathcal{R}$, then

$$s \overset{p}{\Rightarrow}_\mathcal{R} t \quad \text{iff} \quad s \overset{m}{\Rightarrow}_B t.$$
Further Steps

**Strong convergence on term graphs**

- unique normal forms $\rightsquigarrow$ Böhm-graphs
- correspondence infinitary term rewriting $\iff$ cyclic term graph rewriting
Further Steps

Strong convergence on term graphs
- unique normal forms $\leadsto$ Böhm-graphs
- correspondence infinitary term rewriting $\Leftrightarrow$ cyclic term graph rewriting

Higher-Order Systems
- application to $\lambda$-calculus with letrec?