Modes of Convergence for Infinitary Rewriting

Patrick Bahr
paba@diku.dk

University of Copenhagen
Department of Computer Science

16th Estonian Winter School in Computer Science,
Palmse, Estonia, February 27-March 4, 2011
Outline

1. Introduction
   - From Finitary Rewriting to Infinitary Rewriting
   - Why Infinitary Rewriting?

2. Partial Order Model of Infinitary Rewriting

3. Beyond Term Rewriting
   - Abstract Models
   - Term Graph Rewriting
   - Higher-Order Term Rewriting
Outline

1 Introduction
   - From Finitary Rewriting to Infinitary Rewriting
   - Why Infinitary Rewriting?

2 Partial Order Model of Infinitary Rewriting

3 Beyond Term Rewriting
   - Abstract Models
   - Term Graph Rewriting
   - Higher-Order Term Rewriting
Rewriting Systems

What are rewriting systems?
- consist of directed symbolic equations over objects such as strings, terms, graphs etc.
- based on the idea of replacing equals by equals
- provide a formal model of computation
- term rewriting is the foundation of functional programming

Example (Term rewriting system defining addition and multiplication)

\[ R_{++} = \begin{cases} 
    x + 0 & \rightarrow x \\
    x + s(y) & \rightarrow s(x + y) \\
    x * 0 & \rightarrow 0 \\
    x * s(y) & \rightarrow x + (x * y) \\
\end{cases} \]

\[ s^2(0) * s^2(0) \rightarrow^7 s^4(0) \]

\[ R_{++} \text{ is terminating!} \]
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)

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

\[
\text{from}(0) \rightarrow \ldots
\]

intuitively this converges to the infinite list 0 : 1 : 2 : 3 : 4 : 5 : ....
## 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?
- because we can
- 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 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

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

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

1. Introduction
   - From Finitary Rewriting to Infinitary Rewriting
   - Why Infinitary Rewriting?

2. Partial Order Model of Infinitary Rewriting

3. Beyond Term Rewriting
   - Abstract Models
   - Term Graph Rewriting
   - Higher-Order Term Rewriting
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** (= cpo + glbs of non-empty sets)

Convergence

- formalised by the **limit inferior**:

\[
\liminf_{\iota \rightarrow \alpha} t_{\iota} = \bigsqcup \bigcap_{\beta < \alpha} \beta \leq_{\iota} \beta < \alpha \leq_{\iota} \alpha \quad t_{\iota}
\]

- 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 $f(x, y) \rightarrow f(y, x)$

Weak convergence

Strong convergence
Properties of the Partial Order Model

Benefits
- reduction sequences always converge
- 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)
**Strong Convergence on Orthogonal System**

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

### Infinitary Confluence

- \( t \) 
- \( \overrightarrow{t_1} \)
- \( \overrightarrow{t_2} \)
- \( \overrightarrow{t_3} \)

### Infinitary Normalisation

\[
t \quad \overrightarrow{} \quad \rightarrow \quad \perp
\]

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

### Böhm Trees

- The same properties can be achieved by allowing so-called root active terms to be immediately rewritten to \( \perp \).
- Böhm extensions

**In fact:**

metric convergence + Böhm extension = partial-order convergence
Outline

1 Introduction
   - From Finitary Rewriting to Infinitary Rewriting
   - Why Infinitary Rewriting?

2 Partial Order Model of Infinitary Rewriting

3 Beyond Term Rewriting
   - Abstract Models
   - Term Graph Rewriting
   - Higher-Order Term Rewriting
Abstraction

Abstract Modes of Convergence

- Abstract models of infinitary rewriting in the tradition of abstract reduction systems.
- Simply replace “finite reductions” with “converging reductions”.
- Abstract reduction systems are special case with only finite reductions converging
- Recover most abstract properties, e.g. parts of the termination resp. confluence hierarchy

Can we do better?

- This abstraction is rather ad-hoc!
- Open question: Is there a (nice) common generalisation of metric convergence and partial-order convergence?
Term Graph Rewriting

Problem
- There are several alternatives for a metric resp. partial order.
- However, there is no obvious/natural choice! (or is there?)

Candidate Structures
- isomorphism “up to depth n” $\leadsto$ complete ultrametric
- isomorphism “modulo $\bot$” $\leadsto$ complete semilattice
  $\leadsto$ total $p$-convergence $= m$-convergence! (at least for weak conv.)

But
- These structures are quite intricate.
- There are other “more natural” choices for metric spaces based on truncations.
- However, it is not clear what the corresponding partial order is.
Infinitary Term Graph Rewriting – Why Bother?

Applications
- syntax-based semantics (provided we can obtain unique normal forms)
- simulation of infinitary term rewriting by finitary graph rewriting

Partial order higher-order term rewriting
- e.g. lambda calculus or combinatory reduction systems
- variable binding of higher-order calculi can be represented as cycle(?)
- comparison with Böhm-extensions of higher-order calculi
- **Challenge:** parametrised metric \( \sim \) different notions of Böhm(-like) trees
- How can this be captured by a partial order?