View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Dagstuhl Research Online Publication Server

Infinitary Rewriting Coinductively

Jörg Endrullis[1] and Andrew Polonsky[2]

1 **Department of Computer Science, VU University Amsterdam**
De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands.
`j.endrullis@vu.nl`

2 **Institute for Computing and Information Sciences, Radboud University Nijmegen,**
P.O. Box 9010, 6500 GL Nijmegen, The Netherlands
`andrew.polonsky@gmail.com`

Abstract

We provide a coinductive definition of strongly convergent reductions between infinite lambda terms. This approach avoids the notions of ordinals and metric convergence which have appeared in the earlier definitions of the concept. As an illustration, we prove the existence part of the infinitary standardization theorem. The proof is fully formalized in Coq using coinductive types. The paper concludes with a characterization of infinite lambda terms which reduce to themselves in a single beta step.

1998 ACM Subject Classification D.1.1 Applicative (Functional) Programming, D.3.1 Formal Definitions and Theory, F.4.1 Mathematical Logic, F.4.2 Grammars and Other Rewriting Systems, I.1.1 Expressions and Their Representation, I.1.3 Languages and Systems

Keywords and phrases infinitary rewriting, coinduction, lambda calculus, standardization

Digital Object Identifier 10.4230/LIPIcs.TYPES.2011.16

1 Introduction

In the untyped lambda calculus [1], one observes that the fixed point combinator Y has Böhm tree

$$\lambda f.f(f(f\cdots))$$

which looks like a "limit" of the infinite reduction sequence

$$Y \to \lambda f.Y f \to \lambda f.f(Y f) \to \dots$$

Infinitary rewriting [6, 13, 2, 3, 17, 9] makes such statements precise by considering infinite reduction sequences together with the topology on infinite terms generated by finite prefixes: the basic opens are of the form

$$\mathcal{O}_{C[]} = \{t \mid \exists t_1, \dots, t_n.\ t = C[t_1, \dots, t_n]\}$$

where $C[]$ is a finite multi-hole context. Alternatively, this topology is given by the metric d where

$$d(s,t) = \inf\{2^{-n} \mid s \text{ and } t \text{ have the same symbols up to depth } n\}$$

Since the infinite terms can themselves be seen as formal limits of Cauchy sequences of finite terms with the metric above, it is natural to consider rewriting sequences together with this topological structure. Specifically, a reduction sequence

$$t_0 \to t_1 \to t_2 \to \cdots \to t_n \to \cdots$$

© J. Endrullis and A. Polonsky;
licensed under Creative Commons License BY-ND
18th International Workshop on Types for Proofs and Programs (TYPES 2011).
Editors: Nils Anders Danielsson, Bengt Nordström; pp. 16–27
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

is said to *converge weakly to the limit* t if the sequence $\{t_i\}$ converges to t in the metric d.

For example, reducing Curry's fixed point combinator $Yf = WW$, where $W = \lambda x.f(xx)$, yields the infinite sequence

$$Yf = WW \to f(WW) \to f^2(WW) \to \cdots \to f^n(WW) \to \ldots$$

which converges to the limit f^ω.

However, the above notion of infinite reductions does not yet yield a satisfactory rewriting theory (intuitively, because topology does not respect the "rewriting structure" in any way). As has been often stressed by Jan Willem Klop, a much superior notion of transfinite reduction is the so-called *strongly convergent* reduction. This is a reduction as above which satisfies the additional condition that the depth of redexes contracted in the infinite sequence must tend to infinity. This constraint is sufficient to recover fundamental rewriting notions, including descendants, projections of reductions, and standardization.

In the present paper, we observe that an alternative, "coordinate-free" definition of strongly convergent reductions results from interpreting the binary reduction relation as a coinductive type family.

1.0.0.1 Related Work.

Catarina Coquand and Thierry Coquand have explored a similar approach in [4], giving a coinductive definition of standard reductions in infinitary combinatory logic. In his PhD thesis [11] and the paper [12], Felix Joachimski investigates finite reductions between coinductively defined infinite terms. To prove confluence, Joachimski introduces a coinductive definition of infinite developments, but not infinite reductions in general. Our proof of standardization for infinite reductions is a generalization of Plotkin's proof of standardization [15] for finitary rewriting; see also [19].

2 Setup

The set of infinite lambda terms is generated coinductively by the grammar

$$\Lambda^\infty ::= x \mid \Lambda^\infty\Lambda^\infty \mid \lambda x.\Lambda^\infty$$

For infinite terms s and t, we write $s = t$ if s and t are bisimilar, that is, the predicate $=$ is coinductively defined by:

$$\frac{}{x = x} \qquad\qquad \frac{s = s' \quad t = t'}{st = s't'} \qquad\qquad \frac{r = r'}{\lambda x.r = \lambda x.r'}$$

Thus $=$ is the largest relation R such that every $s\,R\,t$ is of one of the forms:

1. $x\,R\,x$,
2. $st\,R\,s't'$ for terms s, s', t, t' with $s\,R\,s'$ and $t\,R\,t'$, or
3. $\lambda x.r\,R\,\lambda x.r'$ for terms r, r' and a variable x with $r\,R\,r'$.

Here and henceforth, we use double inference lines to emphasize that the given derivation system defines a predicate or type family by coinduction rather than by induction.

We frequently denote regular infinite terms by systems of equations, e.g.:

$$M = (\lambda x.x)M$$

It is to be understood that the term M is the infinite tree unfolding of this equation, see Figure 1.

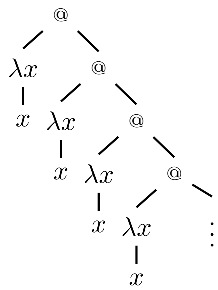


Figure 1 *A regular infinite term.*

The operation of capture-avoiding *substitution*, written $s[u/x]$, is defined by guarded corecursion

s	$s[u/x]$	
x	u	
y	y	$y \neq x$
$s_1 s_2$	$s_1[u/x] s_2[u/x]$	
$\lambda y.r$	$\lambda y.r[u/x]$	$y \notin \mathsf{FV}(u)$

We will not discuss here the problem of implementing Barendregt's variable convention in the infinitary setting. It does present an interesting issue: if the variables are represented by a countable set, then each variable might occur freely in a lambda term. Then it is not possible to find a fresh name which does not occur in it. (We note that the trick of Hilbert's Hotel is not applicable here, since we cannot rename free variables.)

In our Coq formalization, we have used classical deBruijn representation which successfully solves this problem, but it entails proving a number of lifting lemmas. Perhaps the most natural approach to formalizing infinitary rewriting would be to use an explicit substitution calculus based on explicit scope delimiters, as in [10], [18].

The substitution operator satisfies the following, provided that $x \notin \mathsf{FV}(u)$:

$$s[t/x][u/y] = s[u/y][t[u/y]/x] \tag{1}$$

The *one-step beta reduction* is a binary relation on Λ^∞, defined inductively by the rules

$$\frac{}{(\lambda x.r)t \longrightarrow r[t/x]} \qquad \frac{s \longrightarrow s'}{st \longrightarrow s't} \qquad \frac{t \longrightarrow t'}{st \longrightarrow st'} \qquad \frac{r \longrightarrow r'}{\lambda x.r \longrightarrow \lambda x.r'}$$

The relation $\twoheadrightarrow$ of a *finite* beta reduction is the reflexive-transitive closure of $\longrightarrow$, defined inductively by the rules

$$\frac{}{t \twoheadrightarrow t} \qquad \frac{s \twoheadrightarrow t \quad t \longrightarrow t'}{s \twoheadrightarrow t'}$$

(We note that we could also append the single beta step on the left.)

The notion of *one-step weak head reduction* $\longrightarrow_w$ is obtained by restricting $\longrightarrow$ to only the first two rules:

$$\frac{}{(\lambda x.r)t \longrightarrow_w r[t/x]} \qquad \frac{s \longrightarrow_w s'}{st \longrightarrow_w s't}$$

Correspondingly, $\twoheadrightarrow_w$ is the reflexive-transitive closure of $\longrightarrow_w$.

The *infinite* beta reduction $\twoheadrightarrow\!\!\!\!\rightarrow$ is defined coinductively by requiring that every node in the syntax tree becomes "frozen" after finitely many steps. This is made explicit by the following derivation rules, which are this time interpreted coinductively:

$$\frac{s \twoheadrightarrow x}{s \mathrel{\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow} x}$$

$$\frac{s \twoheadrightarrow t_1 t_2 \qquad t_1 \mathrel{\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow} t_1' \qquad t_2 \mathrel{\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow} t_2'}{s \mathrel{\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow} t_1' t_2'}$$

$$\frac{s \twoheadrightarrow \lambda x.r \qquad r \mathrel{\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow} r'}{s \mathrel{\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow} \lambda x.r'}$$

▶ **Example 1.** Let us reconsider Curry's fixed point combinator $Y = \lambda f.WW$ with $W = \lambda x.f(xx)$. Then the infinite rewrite sequence $Yf \mathrel{\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow} f^\omega$ with $f^\omega = f(f^\omega)$ can be derived as follows:

$$\frac{Yf \twoheadrightarrow f(WW) \qquad f \mathrel{\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow} f \qquad \dfrac{WW \twoheadrightarrow f(WW) \qquad f \mathrel{\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow} f \qquad \dfrac{}{WW \twoheadrightarrow f^\omega}}{WW \mathrel{\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow} f^\omega}}{Yf \mathrel{\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow} f^\omega}$$

Note that this is an infinite proof term, as indicated by the loop $\cdots\!\!\blacktriangleright$.

Classically, transfinite reduction sequences are defined as follows (here we view ordinals α as the set of all smaller ordinals $\alpha = \{\beta \mid \beta < \alpha\}$):

▶ **Definition 2.** Let $s \in \Lambda^\infty$, and let α be an ordinal.

A map $t : (\alpha \cup \{\alpha\}) \to \Lambda^\infty$, together with steps $\sigma_\beta : t(\beta) \to t(\beta + 1))_{\beta<\alpha}$ for every $\beta < \alpha$, is a *strongly convergent reduction of length* α *from* $t(0)$ *to* $t(\alpha)$, if the following conditions hold:

1. If $\gamma \le \alpha$ is a limit ordinal, then $t(\gamma)$ is the limit, in the metric topology on infinite terms, of the ordinal-indexed sequence $(t(\beta))_{\beta<\gamma}$;
2. If $\gamma \le \alpha$ is a limit ordinal, then for every $d \in \mathbb{N}$, there exists $\beta < \gamma$, such that, for all β' with $\beta \le \beta' < \gamma$, the redex contracted in the step $\sigma_{\beta'}$ occurs at depth greater than d.

The proof of the following theorem will be given in Section 4.

▶ **Theorem 3.** $s \mathrel{\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow} t$ *if and only if* s *reduces to* t *via a strongly convergent reduction sequence.*

One advantage of the coinductive approach is that it provides a simple and natural definition of standard reductions.

The *infinitary standard reduction* is obtained by the same rules as the infinite beta reductions, except that the finite prefixes are now required to be weak head reductions.

$$\frac{s \twoheadrightarrow_w x}{s \mathrel{\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow}_s x}$$

$$\frac{s \twoheadrightarrow_w t_1 t_2 \qquad t_1 \mathrel{\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow}_s t_1' \qquad t_2 \mathrel{\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow}_s t_2'}{s \mathrel{\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow}_s t_1' t_2'}$$

$$\frac{s \twoheadrightarrow_w \lambda x.r \qquad r \mathrel{\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow}_s r'}{s \mathrel{\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow}_s \lambda x.r'}$$

3 Standardization

We now seek to prove the following fact:

$$s \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow t \implies s \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow_s t$$

The intuition is as follows. In order to replace beta-prefixes with weak head-prefixes, we standardize the beta prefix, extract the initial weak head reduction, and absorb the remainder into the coinductive call. However, the standardization of a finite beta reduction can give rise to an infinite reduction, as in the following counterexample to the Church–Rosser theorem for *finite* reductions between infinite terms:

$$(\lambda f.f^\omega)(\mathtt{I}x) \longrightarrow (\lambda f.f^\omega)x \longrightarrow x^\omega$$

when standardized, yields

$$(\lambda f.f^\omega)(\mathtt{I}x) \longrightarrow (\mathtt{I}x)^\omega \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow x^\omega$$

As an intermediate step, we therefore first convert the prefixes to infinite standard reductions. This suggests the introduction of one more auxiliary reduction $\longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow_a$, which follows the above scheme but takes for prefixes infinite standard reductions defined previously.

$$\frac{s \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow_s x}{s \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow_a x}$$

$$\frac{s \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow_s t_1t_2 \qquad t_1 \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow_a t_1' \qquad t_2 \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow_a t_2'}{s \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow_a t_1't_2'}$$

$$\frac{s \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow_s \lambda x.r \qquad r \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow_a r'}{s \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow_a \lambda x.r'}$$

Infinitary standardization theorem now follows by a series of simple lemmas:

▶ **Lemma 4.** *We have*

1. $s \twoheadrightarrow_w t,\ t \twoheadrightarrow_w u \implies s \twoheadrightarrow_w u$
2. $s \twoheadrightarrow_w t,\ t \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow_s u \implies s \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow_s u$
3. $s \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow_s s',\ t \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow_s t' \implies s[t/x] \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow_s s'[t'/x]$
4. *For* $\longrightarrow_R \in \{\longrightarrow, \twoheadrightarrow, \twoheadrightarrow_w\}$,

$$s \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow_s t, t \longrightarrow_R u \implies s \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow_s u$$

5. $s \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow_s t,\ t \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow_s u \implies s \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow_s u$

Proof. **1.** By induction.

2. By case distinction, using 1 to concatenate the prefix.

3. By coinduction, using that

$$\begin{aligned} s \longrightarrow_w t &\implies s[u/x] \longrightarrow_w t[u/x] \\ s \twoheadrightarrow_w t &\implies s[u/x] \twoheadrightarrow_w t[u/x] \end{aligned}$$

4. By induction on $t \longrightarrow_R u$, using 3 for the redex base case.

5. By coinduction on $t \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow_s u$

Case 1 $t \twoheadrightarrow_w x = u$. Then $s \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow_s x$ by 4.

Case 2 $u = u_1u_2$, $t \twoheadrightarrow_w t_1t_2$, and $t_i \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow_s u_i$. By 4, $s \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow_s t_1t_2$. Hence $s \twoheadrightarrow_w t_1't_2'$, with $t_i' \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow_s t_i$. By coinduction, $t_i' \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow_s u_i$. Using that $s \twoheadrightarrow_w t_1't_2'$, we get $s \longrightarrow\!\!\!\!\!\twoheadrightarrow\!\!\!\!\!\twoheadrightarrow_s u_1u_2$.

Case 3 $u = \lambda x.v$, $t \twoheadrightarrow_w \lambda x.r$, and $r \twoheadrightarrow_s v$. By 4, $s \twoheadrightarrow_s \lambda x.r$. Hence $s \twoheadrightarrow_w \lambda x.r'$, with $r' \twoheadrightarrow_s r$. By coinduction, $r' \twoheadrightarrow_s v$. Using that $s \twoheadrightarrow_w \lambda x.r'$, we get $s \twoheadrightarrow_s \lambda x.v$. ◀

▶ **Lemma 5.** *We have*

1. $s \twoheadrightarrow_s t,\ t \twoheadrightarrow_s u \Longrightarrow s \twoheadrightarrow_s u$
2. $s \twoheadrightarrow_s t,\ t \twoheadrightarrow_a u \Longrightarrow s \twoheadrightarrow_a u$.
3. $s \twoheadrightarrow_a s',\ t \twoheadrightarrow_a t' \Longrightarrow s[t/x] \twoheadrightarrow_a s'[t'/x]$
4. *For* $\longrightarrow_R \in \{\longrightarrow, \twoheadrightarrow, \twoheadrightarrow_w, \twoheadrightarrow_s\}$,

$$s \twoheadrightarrow_a t, t \longrightarrow_R u \Longrightarrow s \twoheadrightarrow_a u$$

5. $s \twoheadrightarrow_a t,\ t \twoheadrightarrow_a u \Longrightarrow s \twoheadrightarrow_a u$

Proof. 1 was proved in the previous lemma. The rest follows the proof there mutatis mutandis. ◀

▶ **Lemma 6.** *We have*

1. $s \twoheadrightarrow_s t \Longrightarrow s \twoheadrightarrow t$
2. $s \twoheadrightarrow t \Longrightarrow s \twoheadrightarrow_s t$
3. $s \twoheadrightarrow t \Longrightarrow s \twoheadrightarrow_a t$
4. $s \twoheadrightarrow_s t \Longrightarrow s \twoheadrightarrow_a t$
5. $s \twoheadrightarrow_a t \Longrightarrow s \twoheadrightarrow_s t$

Proof. **1.** Immediate: every weak head prefix is also a beta prefix.
2. By induction on $s \twoheadrightarrow t$, using Lemma 4.4 and reflexively of $\twoheadrightarrow_s$.
3. Immediate by 2.
4. By composition of 1 and 3.
5. By coinduction on $s \twoheadrightarrow_a t$:

Case 1 $s \twoheadrightarrow_s x = t$. Done.

Case 2 $t = t_1 t_2$, $s \twoheadrightarrow_s s_1 s_2$, and $s_i \twoheadrightarrow_a t_i$. Hence $s \twoheadrightarrow_w s'_1 s'_2$, with $s'_i \twoheadrightarrow_s s_i$. By 4, $s'_i \twoheadrightarrow_a s_i$. By Lemma 5.5, $s'_i \twoheadrightarrow_a t_i$. By coinduction, $s'_i \twoheadrightarrow_s t_i$. Using that $s \twoheadrightarrow_w s'_1 s'_2$, we get $s \twoheadrightarrow_s t_1 t_2$ by constructor.

Case 3 $t = \lambda x.v$, $s \twoheadrightarrow_s \lambda x.r$, and $r \twoheadrightarrow_a v$. Hence $s \twoheadrightarrow_w \lambda x.r'$, with $r' \twoheadrightarrow_s r$. By 4, $r' \twoheadrightarrow_a r$. By Lemma 5.5, $r' \twoheadrightarrow_a v$. By coinduction, $r' \twoheadrightarrow_s v$. Using that $s \twoheadrightarrow_w \lambda x.r'$, we get $s \twoheadrightarrow_s \lambda x.v$.

◀

▶ **Theorem 7.** $s \twoheadrightarrow t \quad \Longrightarrow \quad s \twoheadrightarrow_s t$

Proof. By composing parts 3 and 5 of Lemma 6. ◀

▶ Remark. Technically speaking, we have only proved the existence part of Curry's standardization theorem; as some rewriting theorists would argue, in the finitary case, the theorem also asserts that the standard reduction is strongly equivalent with the given one in the sense of Lévy, and is furthermore a unique representative of this equivalence class.

We find it an interesting problem to give a coinductive formulation of the notion of Lévy-equivalence for infinite reductions.

The Coq formalization of the coinductive treatment of infinitary rewriting — in particular, the proof of standardization — can be downloaded from `http://joerg.endrullis.de`. All coinductive proofs in Coq have to adhere to a strict syntactic guardedness condition [5] for guaranteeing constructive well-definedness, also known as productivity [7]. We have employed a proof transformation method from [8], in order to transform productive into guarded proofs.

4 Coinductive Reductions are Strongly Convergent

We now prove Theorem 3:

$$s \longrightarrow\!\!\!\!\!\twoheadrightarrow t \quad \Longleftrightarrow \quad s \text{ reduces to } t \text{ via a strongly convergent reduction sequence}$$

Theorem 3. ($\Rightarrow$) Suppose that $s \longrightarrow\!\!\!\!\!\twoheadrightarrow t$. By traversing the infinite derivation tree of $s \longrightarrow\!\!\!\!\!\twoheadrightarrow t$ in the breadth-first order, and accumulating the finite beta-prefixes by concatenation, we get a reduction sequence of length ω which satisfies the depth requirement by construction.

($\Leftarrow$) Let R be a strongly convergent reduction sequence from s to t of length α; we write this as $s \xrightarrow{R}_\alpha t$. By induction on α, we show that $s \longrightarrow\!\!\!\!\!\twoheadrightarrow_a t$. This suffices for $s \longrightarrow\!\!\!\!\!\twoheadrightarrow t$ by Lemma 6.5 and 6.1.

Zero case: $s \xrightarrow{R}_0 t$. Then $s = t$, hence $s \longrightarrow\!\!\!\!\!\twoheadrightarrow_s t$ and $s \longrightarrow\!\!\!\!\!\twoheadrightarrow_a t$.

Successor: $s \xrightarrow{R}_{\alpha+1} t$. Then $s \xrightarrow{R}_\alpha s' \longrightarrow t$. Then $s' \longrightarrow\!\!\!\!\!\twoheadrightarrow_s t$ and $s' \longrightarrow\!\!\!\!\!\twoheadrightarrow_a t$, and by the induction hypothesis, $s \longrightarrow\!\!\!\!\!\twoheadrightarrow_a s'$. Thus $s \longrightarrow\!\!\!\!\!\twoheadrightarrow_a t$ by Lemma 5.5.

Limit: $s \xrightarrow{R}_\alpha t$, α a limit ordinal. We define an infinite derivation of $s \longrightarrow\!\!\!\!\!\twoheadrightarrow t$ coinductively. By the depth condition, there exists $\beta < \alpha$ such that, for every $\gamma \geq \beta$, the redex contracted by R at γ occurs at depth greater than zero. Let t_β be the term at index β in R. Then by induction hypothesis we have $s \longrightarrow\!\!\!\!\!\twoheadrightarrow_a t_\beta$, and $s \longrightarrow\!\!\!\!\!\twoheadrightarrow_s t_\beta$ by Lemma 6.5. We distinguish three possible shapes of t_β.

- **Variable:** $t_\beta = x$. This is impossible, since then t_β cannot reduce to anything, while we assumed that $\beta < \gamma$.
- **Abstraction:** $t_\beta = \lambda x.r$. Then $t = \lambda x.u$, and $r \longrightarrow_{\leq\alpha} u$. Then $r \longrightarrow\!\!\!\!\!\twoheadrightarrow_a u$ by coinduction. Now $s \longrightarrow\!\!\!\!\!\twoheadrightarrow \lambda x.u$ by the abstraction constructor of $\longrightarrow\!\!\!\!\!\twoheadrightarrow_a$.
- **Application:** $t_\beta = t_1 t_2$. Then $t = u_1 u_2$ and the tail of reduction R past β can be split into two parts $\{t_i \longrightarrow_{\leq\alpha} u_i \mid i = 0, 1\}$ of length at most α. Then $t_0 \longrightarrow\!\!\!\!\!\twoheadrightarrow_a u_0$ and $t_1 \longrightarrow\!\!\!\!\!\twoheadrightarrow_a u_1$ by coinduction. Now $s \longrightarrow\!\!\!\!\!\twoheadrightarrow u_1 u_2$ by the application constructor of $\longrightarrow\!\!\!\!\!\twoheadrightarrow_a$.

◀

5 Loops Loops Loops Loops Loops Loops Loops Loops Loops ...

One might wonder which infinite reductions converge in the weak sense of topology but not in the strong/coinductive sense above. One example is the infinite head reduction of $\Omega = (\lambda x.xx)(\lambda x.xx)$.

$$\Omega \to \Omega \to \Omega \to \cdots \tag{2}$$

which converges to Ω in the metric on infinite terms, but is not strongly convergent. Here we nevertheless have $\Omega \longrightarrow\!\!\!\!\!\twoheadrightarrow \Omega$ due to finite prefixes of the infinite reduction (in particular, the empty reduction). Not every topologically convergent reduction has a strongly convergent counterpart. This is illustrated by the following reduction:

$$\begin{aligned} M &= (\lambda x_0.(\lambda x_1.(\lambda x_2.\ldots)(x_1\mathsf{I}))(x_0\mathsf{I}))\mathsf{I} \\ &\to (\lambda x_0.(\lambda x_1.(\lambda x_2.\ldots)(x_1\mathsf{I}))(x_0\mathsf{I}))(\mathsf{II}) \\ &\to (\lambda x_0.(\lambda x_1.(\lambda x_2.\ldots)(x_1\mathsf{I}))(x_0\mathsf{I}))(\mathsf{III}) \\ &\ \ \vdots \\ &\to (\lambda x_0.(\lambda x_1.(\lambda x_2.\ldots)(x_1\mathsf{I}))(x_0\mathsf{I}))(\mathsf{I}^\omega) = N \end{aligned} \tag{3}$$

This reduction converges only topologically, every rewrite step occurs at the root. In fact, there exists no strongly convergent reduction from M to N, we do not have $M \longrightarrow\!\!\!\!\!\twoheadrightarrow N$.

We note that both examples of topologically convergent reductions (2) and (3) contain a term that admits a loop: $\Omega \to \Omega$ and $N \to N$, respectively. A recent theorem of [16] states that these examples are paradigmatic: if R is a reduction sequence which is weakly, but not strongly, convergent, then R contains a term which reduces to itself in one beta-reduction step.

We conclude this paper by giving a characterization of all such terms.

▶ **Definition 8.** For $M \in \Lambda^\infty$, we define:

1. A *one-cycle* is a rewrite step $M \to M$.
2. A *loop* is a rewrite step $M \to M$ at the root of the term.

Note that every one-cycle $M \to M$ is of the form $M \equiv C[M'] \to C[M']$ for some context C and a loop $M' \to M'$. As a consequence, the interesting objects are the loops, and we are interested in a characterization of terms that admit loops. For the case of (ordinary) finitary λ-calculus, this problem has been studied and solved by Lercher in 1976 [14] who showed that Ω is the only finite looping λ-term:

▶ **Theorem 9** (Lercher). *The only finite λ-term M such that $M \to M$ via a root step is $\Omega \equiv (\lambda x.\, xx)(\lambda x.\, xx)$.*

In infinitary lambda calculus, the situation becomes more involved. It turns out, that there are 3 looping terms with a finite spine (among which of course Ω), and there is a whole scheme of uncountably many terms with an infinite spine.

▶ **Theorem 10.** *The looping terms in infinitary λ-calculus are precisely the terms that are of one of the following forms:*

1. I^ω,
2. $\Omega \equiv (\lambda x.\, xx)(\lambda x.\, xx)$,
3. *BB where B is the infinite solution of $B \equiv \lambda x.xB$, or*
4. *$(\lambda x_0.(\lambda x_1.(\lambda x_2.\ldots)s_2)s_1)s_0$ such that for every $i \in \mathbb{N}$, the term s_{i+1} is obtained from s_i by replacing all x_j by x_{j+1} followed by replacing an arbitrary (possibly infinite) number of occurrences of s_0 by x_0. We call such a term a* cascade.

The terms in cases (1), (2) and (3) are displayed in Figure 2.

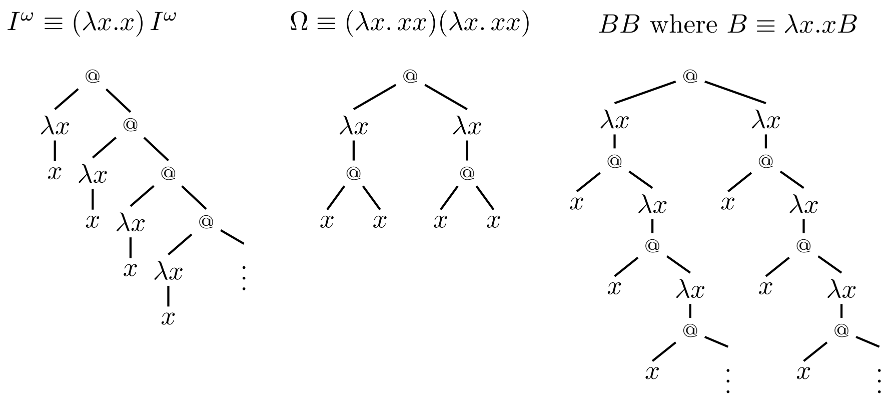
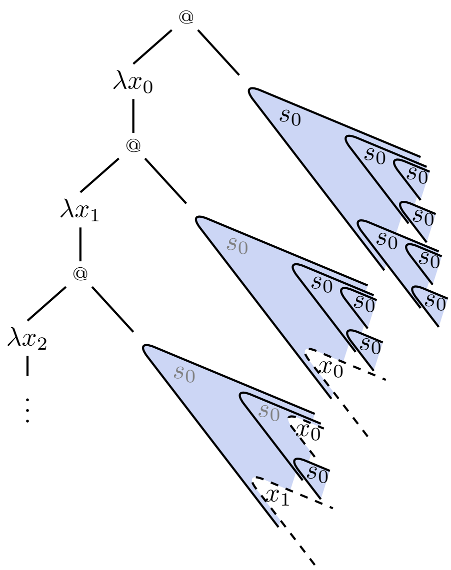


Figure 2 *Looping terms in infinitary λ-calculus, except for cascades.*

The case (4) of cascades is illustrated in Figure 3, and an example of a cascade is shown in Figure 4. A cascade $(\lambda x_0.(\lambda x_1.(\lambda x_2....)s_2)s_1)s_0$ can equivalently be characterized as follows: for every $n \in \mathbb{N}$, the term s_i is obtained from s_{i+1} by a substitution replacing x_0 by s_0 and all variables x_{j+1} by x_j.

4th (class of) solution(s): $M \equiv (\lambda x_0.(\lambda x_1.(\lambda x_2.\ldots)s_2)s_1)s_0$
with $s_i = s_{i+1}[\, x_0 = s_0,\;\; x_1 = x_0, \ldots, x_{i+1} = x_i \,]$ for $i \geq 1$



The recipe for cascades:

- take any term s_0
- obtain s_{i+1} from s_i by:
  (a) replacing all occurrences of x_i by x_{i+1} (for all $i \in \mathbb{N}$ in parallel),
  (b) replacing some (zero or more) occurrences of subterms s_0 by x_0

Figure 3 *The structure of cascades in infinitary λ-calculus. The gray occurrences s_0 indicate that this term is obtained from s_0 by replacing subterms by variables.*

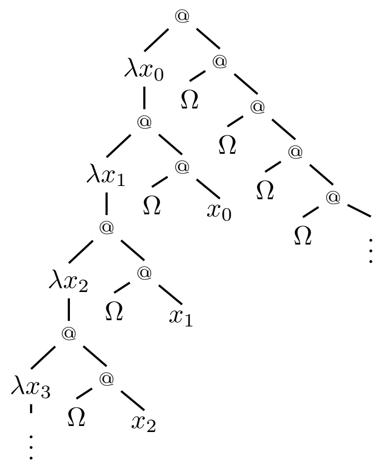


Figure 4 Example of a cascade.

Proof of Theorem 10. Let $M \in \Lambda^\infty$ be a term that admits a loop $M \to M$. Then M has a redex at the root, thus $M \equiv (\lambda x.M')C$ for some $M', C \in \Lambda^\infty$. We distinguish the following cases for M':

(ia) *M' is a variable, $M' \equiv x$.* Then $M \equiv (\lambda x.x)C \to C \equiv M$, and hence $M \equiv \mathsf{I}^\omega$. This is case (1) in the theorem.

(ia) *M' is a variable, $M' \equiv y \neq x$.* Then $M \equiv (\lambda x.y)C \to y \not\equiv M$, contradiction.

(ii) *M' is an abstraction.* Then the reduct would be an abstraction, contradiction

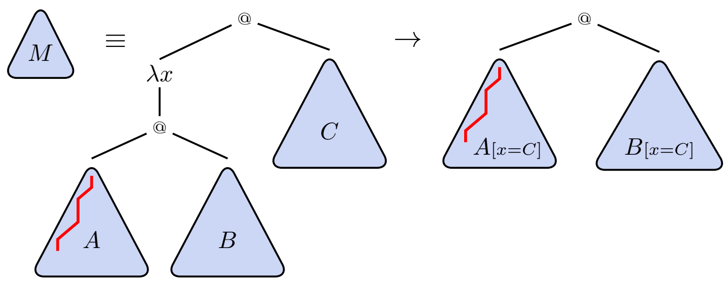
(iii) *M' is an application, $M \equiv AB$.* We analyse this case below.

For (iii) we have: $M \equiv (\lambda x.AB)C$ and by assumption $M \equiv (AB)[x := C]$. Hence

(a) $A[x := C] \equiv \lambda x.AB$, and

(b) $C \equiv B[x := C]$.

We consider the left spine L of A, depicted thick and red in the following picture:

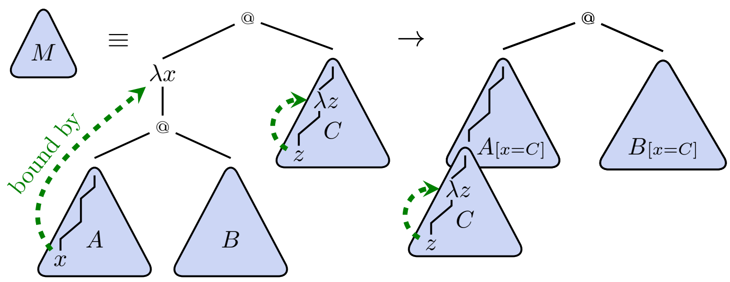


Now there are two possibilities, either the spine L is finite or infinite:

(1) *L is finite.*

Assume that the spine would end in a variable $y \not\equiv x$. This assumption yields a contradiction by (a) since then the spine of $A[x := C]$ in the reduct would be shorter than the left spine of $(\lambda x.AB)$.

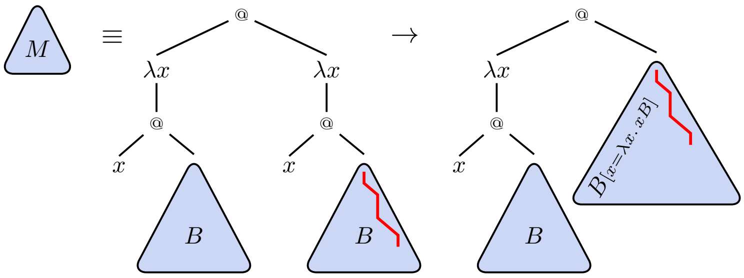
As a consequence, the spine ends in the variable x. This situation is surveyed in the following picture:



We conclude that $A \equiv x$ as otherwise the variable at the end of the spine in $A[x := C]$ cannot be bound at the root as in $(\lambda x.AB)$. Then $C \equiv \lambda x.xB$ by (a) and together with (b) we get:

$$\lambda x.xB \equiv B[x := \lambda x.xB] \qquad (\dagger)$$

We consider the right spine R of B, displayed red in the following picture:



Again, there are the following possibilities:

(i) *R is finite.* As before, it follows that $B \equiv x$ since otherwise the right spine of the reduct would be shorter than the right spine of M. Hence we have found the well-known looping term $M \equiv \Omega \equiv (\lambda.xx)(\lambda.xx)$.

(ii) *R is infinite.* Then the right spine of $\lambda x.xB$ is the same as that of B, and hence is an alternation of abstraction and application. Thus:

$$B \equiv \lambda x_0.s_0(\lambda x_1.s_1(\lambda x_2.s_2(\ldots)))$$

for some terms s_i. From (†) it follows $s_0 \equiv x_0$, and this in turn implies that $s_1 \equiv x_1$, and then $s_2 \equiv x_2$, ans so forth. Using induction we obtain $s_i \equiv x_i$. Thus $B \equiv \lambda x.xB$, $C \equiv B$ and $M \equiv (\lambda x.xB)B \equiv BB$.

(2) *L is infinite.*

Then the spine of A must be the same as that of $(\lambda x.AB)$, and thus is an alteration of lambda and application. As a consequence, we have

$$M \equiv (\lambda x_0.(\lambda x_1.(\lambda x_2....)s_2)s_1)s_0$$

for some terms s_i. As a consequence the loop $M \to M$, it follows that:

$$M \equiv (\lambda x_0.(\lambda x_1.(\lambda x_2....)s_2)s_1)s_0 \;=_\alpha\; (\lambda x_1.(\lambda x_2....)s_2)s_1)[x_0 := s_0]$$

Thus, for every $i \geq 1$ we have that s_i is be obtained from s_{i+1} by replacing x_0 by s_0 and all variables x_{j+1} by x_j (the α-renaming).

◀

References

1 H.P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics.* Elsevier Science, revised edition, 1985.

2 A. Berarducci. Infinite λ-Calculus and Non-Sensible Models. In *Logic and Algebra (Pontignano, 1994)*, pages 339–377. Dekker, New York, 1996.

3 A. Berarducci and B. Intrigila. Church–Rosser λ-theories, Infinite λ-calculus and Consistency Problems. *Logic: From Foundations to Applications*, pages 33–58, 1996.

4 C. Coquand and T. Coquand. On the Definition of Reduction for Infinite Terms. *Comptes Rendus de l'Académie des Sciences. Série I*, 323(5):553–558, 1996.

5 Th. Coquand. Infinite objects in type theory. In Henk Barendregt and Tobias Nipkow, editors, *Types for Proofs and Programs, International Workshop TYPES'93, Nijmegen, The Netherlands, May 24–28, 1993, Selected Papers*, volume 806 of *Lecture Notes in Computer Science*, pages 62–78. Springer, 1994.

6 N. Dershowitz, S. Kaplan, and D.A. Plaisted. Rewrite, Rewrite, Rewrite, Rewrite, Rewrite,.... *Theoretical Computer Science*, 83(1):71–96, 1991.

7 J. Endrullis, C. Grabmayer, D. Hendriks, A. Isihara, and J.W. Klop. Productivity of Stream Definitions. *Theoretical Computer Science*, 411:765–782, 2010.

8 J. Endrullis, D. Hendriks, and M. Bodin. Circular Coinduction in Coq using Bisimulation-Up-To Techniques. Unpublished note.

9 J. Endrullis, D. Hendriks, and J.W. Klop. Highlights in Infinitary Rewriting and Lambda Calculus. *Theoretical Computer Science*, 464:48–71, 2012.

10 D. Hendriks and V. van Oostrom. Adbmal. In *Proc. Conf. on Automated Deduction (CADE 2003)*, volume 2741 of *Lecture Notes in Artificial Intelligence*, pages 136–150. Springer, 2003.

11 F. Joachimski. *Reduction Properties of* Π*IE-Systems*. PhD thesis, LMU München, 2001.

12 F. Joachimski. Confluence of the Coinductive [Lambda]-Calculus. *Theoretical Computer Science*, 311(1-3):105–119, 2004.

13 J.R. Kennaway, J.W. Klop, M.R. Sleep, and F.-J. de Vries. Transfinite Reductions in Orthogonal Term Rewriting Systems. *Information and Computation*, 119(1):18–38, 1995.

14 B. Lercher. Lambda-Calculus Terms That Reduce To Themselves. *Notre Dame Journal of Formal Logic*, 17(2):291–292, 1976.

15 G.D. Plotkin. Call-by-Name, Call-by-Value and the Lambda-Calculus. *Theoretical Computer Science*, 1(2):125–159, 1975.

16 J.G. Simonsen. Weak Convergence and Uniform Normalization in Infinitary Rewriting. In *Proc. 20th Int. Conf. on Rewriting Techniques and Applications (RTA 2009)*, volume 6 of *Leibniz International Proceedings in Informatics*, pages 311–324. Schloss Dagstuhl, 2010.

17 Terese. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.

18 V. van Oostrom. Explicit Substitution for Graphs. In *Nieuwsbrief van de Nederlandse Vereniging voor Theoretische Informatica*, number 9, pages 34–39, 2005.

19 H. Xi. Upper bounds for standardizations and an application. *J. Symb. Log.*, 64(1):291–303, 1999.