Theoretical Computer Science 464 (2012) 48–71

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Highlights in infinitary rewriting and lambda calculus

Jörg Endrullis *, Dimitri Hendriks, Jan Willem Klop

Vrije Universiteit Amsterdam, Department of Computer Science, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

A B S T R A C T

We present some highlights from the emerging theory of infinitary rewriting, both for first-order term rewriting systems and λ-calculus.

In the first section we introduce the framework of infinitary rewriting for first-order rewrite systems, so without bound variables. We present a recent observation concerning the continuity of infinitary rewriting.

In the second section we present an excursion to the infinitary λ-calculus. After the main definitions, we mention a recent observation about infinite looping λ-terms, that is, terms that reduce in one step to themselves. Next we describe the fundamental trichotomy in the semantics of λ-calculus: Böhm trees, Lévy–Longo trees, and Berarducci trees. We conclude with a short description of a new refinement of Böhm tree semantics, called clocked semantics.

1. Introduction

In the cradle of the information age, with the emergence of the notions of computability and decidability some eighty years ago, the formal systems of λ-calculus and Combinatory Logic saw light. A descendant of these systems, one or two generations later, was formed by the more general notion of term rewriting systems, together with the rise of functional programming languages and the theory of algebraic specifications. Again a generation later several extensions and applications of this format were developed, in particular infinitary rewriting, term graph theory, the technology of narrowing and completion, the termination proof tools, and automated deduction and verification tools. In almost all these areas our colleague and friend Yoshihito Toyama has made several prominent contributions, that have shaped and enriched the field. Our paper is dedicated to him, on the occasion of his 60th birthday, in admiration and gratitude for his many accomplishments and his everlasting inspiration.

On the suggestion of this volume's editors, and also in the spirit of Toyama's interests, we have endeavoured to present in this paper an outlook on a strand of research that has emerged in the last two decades, concerning the infinitary extension of the λ-calculus and more general (orthogonal) term rewriting systems.

Our paper will present some of the highlights of infinitary rewriting, mostly in an informal way, leaving the completely detailed, formal proofs to the literature to which pointers are provided. Most of the material is by now well-established, but we have inserted some new results and observations, and at these points we have also included the proofs.

A few words about the rationale of infinitary rewriting. After the initial set-up of the λ-calculus and Combinatory Logic (CL) in the 1930s and their subsequent analysis and employment in mathematical logic, a next major step of mountainous importance was formed by the discovery by Scott, Plotkin, Engeler and others of the famous mathematical models of λ-calculus and CL, in the form of $\mathcal{D}^\infty$, $\mathcal{P}\omega$ and their variants. To describe the equality in these models infinitary λ-terms were

* Corresponding author. Tel.: +31 205989886.
E-mail addresses: j.endrullis@vu.nl (J. Endrullis), r.d.a.hendriks@vu.nl (D. Hendriks), j.w.klop@vu.nl (J.W. Klop).

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2012.08.018

used, known as Böhm trees (and later variants such as Lévy–Longo and Berarducci trees). The employment of infinite λ-terms thus entered the field in a natural way. This was still in a restricted form, Böhm trees are infinite normal forms but cannot be applied to each other. Now rewriting theory took the dimension of infinity seriously, and developed a full theory of possibly infinite terms, including their application to each other. Thus we find operational versions, as normal form models, for the main classic models $\mathcal{D}^\infty$ and $\mathcal{P}\omega$.

A benefit of the infinitary λ-calculus and rewrite systems is the ease of calculations that directly correspond to the equality in the models. Of course there were means for establishing equations such as Scott's Induction Rule, but calculating directly with the infinite terms seems more convenient. Examples are given in this paper.

The original interest in this infinitary extension was triggered by term graph rewriting [25], where we typically have cyclic term graphs, which after infinite unwinding give rise to infinite trees.

It is arguable whether the *transfinite* extension of infinitary rewriting is necessary or useful. In fact, by the Compression Lemma, we can restrict our attention to reduction lengths not exceeding the ordinal ω, but it is much more fun (besides facilitating reasoning) to create the vastly more extended space of reductions of length of any countable ordinal, and consider rewrite systems that contain computations of the giant ordinals ϵ_0 and Γ_0. If desired, one can always be satisfied with the initial segment of rewrite theory up to ω. We note that not all systems have compression, e.g. $\lambda^\infty\beta\eta$-calculus, see [58, page 691].

2. Infinitary rewriting for first order systems

2.1. Basics of infinitary rewriting

In this section we will consider possibly infinite terms over a first-order signature Σ. We assume familiarity with these notions, for which precise definitions can be found, e.g., in [58,37] and many other sources. For the extension to infinite terms we observe that the rules of $\mathcal{R} = \langle \Sigma, R\rangle$ apply just as well to finite as to infinite terms; their applicability just depends on the presence of a finite 'redex pattern'. Infinite terms arise from the set of finite terms, $Ter(\Sigma)$, by metric completion, using the well-known distance function d such that for $t, s \in Ter(\Sigma)$, $d(t, s) = 2^{-n}$ if the n-th level of the terms t, s (viewed as labelled trees) is the first level where a difference appears, in case t and s are not identical; furthermore, $d(t, t) = 0$. It is standard that this construction yields $\langle Ter(\Sigma), d\rangle$ as a metric space. Now infinite terms are obtained by taking the completion of this metric space, and they are represented by infinite trees. We will refer to the complete metric space arising in this way as $\langle Ter^\infty(\Sigma), d\rangle$, where $Ter^\infty(\Sigma)$ is the set of finite and infinite terms over Σ.

A natural consequence of this construction is the emergence of the notion of *Cauchy convergence*: we say that $t_0 \to t_1 \to t_2 \to \ldots$ is an infinite reduction sequence with limit t, if t is the limit of the sequence $t_0, t_1, t_2, \ldots$ in the usual sense of Cauchy convergence. Cauchy convergence is sometimes also called *weak convergence*. In fact, we will use throughout a stronger notion that has better properties. This is *strong convergence*, which in addition to the stipulation for Cauchy (or weak) convergence, requires that the depth of the redexes contracted in the successive steps tends to infinity when approaching a limit ordinal from below. So this rules out the possibility that the action of redex contraction stays confined at the top, or stagnates at some finite level of depth. See further Fig. 1 for an intuitive illustration.

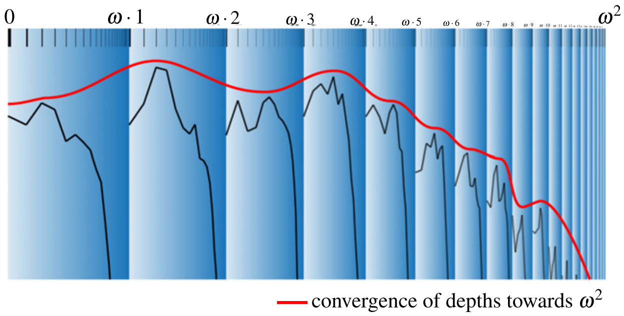


Fig. 1. Depth of redex contractions tends to infinity at each limit ordinal.

A more precise definition is as follows: a *transfinite rewrite sequence* (of ordinal length α) is a sequence of rewrite steps $(t_\beta \to_{\mathcal{R},p_\beta} t_{\beta+1})_{\beta<\alpha}$ such that for every limit ordinal $\lambda < \alpha$ we have that if β approaches λ from below, then

(i) the distance $d(t_\beta, t_\lambda)$ tends to 0 and, moreover,
(ii) the depth of the rewrite action, i.e., the length of the position p_β, tends to infinity.

The sequence is called *strongly convergent* if α is a successor ordinal, or there exists a term t_α such that the conditions (i) and (ii) are fulfilled for every limit ordinal $\lambda \le \alpha$. In this case we write $t_0 \twoheadrightarrow_{\mathcal{R}} t_\alpha$, or $t_0 \to^{\alpha} t_\alpha$ to explicitly indicate the length α of the sequence. The sequence is called *divergent* if it is not strongly convergent.

There are several reasons why strong convergence is beneficial; the foremost being that in this way we can define the notion of *descendant* (also *residual*) over limit ordinals. Also the well-known Parallel Moves Lemma (see Section 2.2) and the Compression Lemma (Theorem 2.8, below) fail for weak convergence, see [54] and [8] respectively. It is further easy to establish that strongly convergent reductions can have any countable length; weakly convergent reductions can have any length, as the one-rule TRS with $C \to C$ demonstrates.

The notion of *normal form*, which now may be an infinite term, is unproblematic: it is a term without a redex occurrence.

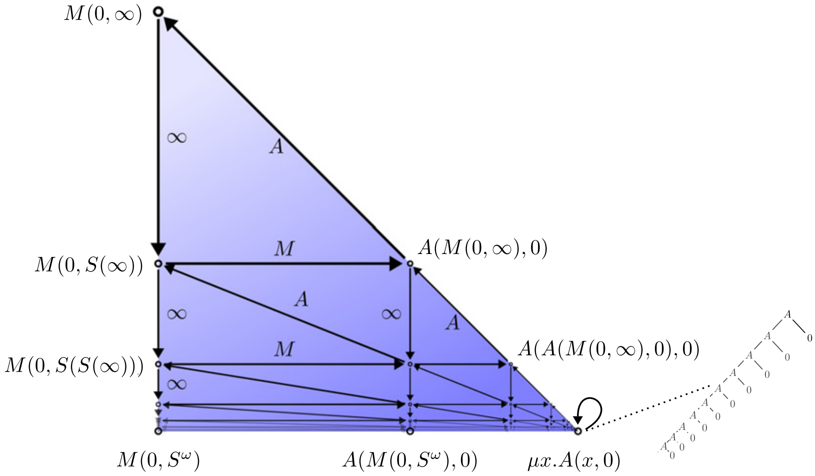


Fig. 2. Zero times infinity.

Example 2.1 (*Zero Times Infinity*)**.** Let us discuss all the concepts introduced so far by means of the following reduction rules for addition and multiplication due to Dedekind [7], in combination with a reduction rule defining the constant ∞ for 'infinity':

$$\begin{array}{lll} A(x,0) \to x & M(x,0) \to 0 & \infty \to S(\infty) \\ A(x,S(y)) \to S(A(x,y)) & M(x,S(y)) \to A(M(x,y),x). & \end{array}$$

The constant 0 and the unary S for successor generate the finite natural numbers. These rules compute some familiar identities for ∞, such as

$$A(S^n(0),\infty) = A(\infty,S^n(0)) = A(\infty,\infty) = \infty$$
$$M(S^{n+1}(0),\infty) = M(\infty,S^{n+1}(0)) = M(\infty,\infty) = \infty$$

in the sense that these terms reduce to the same infinite normal form, namely $S^\omega = S(S(S(\ldots)))$.

How about zero times infinity? The equation $M(\infty,0) = 0$ is immediate, but the term $M(0,\infty)$ is interesting, since it turns out to be undefined, as it allows for, e.g., the following reduction cycle:

$$M(0,\infty) \to M(0,S(\infty)) \to A(M(0,\infty),0) \to M(0,\infty).$$

The whole reduction graph including all finite and infinite reducts of $M(0,\infty)$ is displayed in Fig. 2. It turns out to be full of cycles, the shortest one constituting the top of the triangular reduction graph. All terms in the graph are hypercollapsing (a notion to be explained later); the term below right, a regular tree that we render in abbreviation as $\mu x.A(x,0)$ is reducible only to itself, even in infinitely many different one step reductions. None of the terms in the graph have a normal form, i.e., they are not WN^∞. There is no longest strongly convergent reduction, in fact there are strongly convergent reductions of any countable ordinal length. The same holds for divergent reductions. The diagonal steps are all collapsing steps, but no diagonal steps emanate from the term $\mu x.A(x,0)$; it only collapses to itself. This μ-term is only a convenient notation for an infinite term, namely the one depicted in the figure; it is *not* a term in our rewrite system. In general, we use $\mu x.C[x]$ to denote the infinite term t that is the solution of $t \equiv C[t]$.

We notice that in every TRS, even those with uncountably many symbols and rules, all transfinite reductions have countable length. All countable ordinals can indeed occur as the length of a strongly convergent reduction, e.g. for the TRS $\mathsf{a}(x) \to \mathsf{b}(x)$. For ordinary Cauchy-convergent reductions this is not so: the rewrite rule $C \to C$ yields arbitrarily long convergent reductions $C \to_c^{\alpha} C$. However, these are not strongly convergent, except the ones of finite length.

Strong convergence versus Cauchy convergence. We will have a closer look at the difference between Cauchy convergence (CC) and strong convergence (SC). First this is done with a signature extension, using a marker indicating activity; next the connection with reduction loops is shown. Consider the following abbreviations:

CC: *Cauchy convergence*, informally defined above;
SC: *Strong convergence*, also defined above;
CCC: *Cauchy convergence with colours*, explained below.

Given is the first-order signature Σ and a TRS (Σ, R). We extend this signature by adding a coloured 'activity marker' $\star$, a unary symbol with the reduction rule

$$\star(x) \to x.$$

The old reduction rules are changed in such a way that the right-hand side is prefixed with $\star$. For example, for Combinatory Logic (CL) this gives the following rules for S and I:

$$\mathsf{I}x \to \star(x) \qquad \mathsf{S}xyz \to \star(xz(yz)).$$

The resulting TRS is $\langle \Sigma', R' \rangle$. Now given an old reduction in $\langle \Sigma, R \rangle$, we can lift it to the coloured version $\langle \Sigma', R' \rangle$ by applying the rules as modified, introducing the markers $\star$. The markers are removed in the next step using the $\star$-rule (intuitively, the heat generated by the activity 'cools down'); formally, the immediate removal of the markers amounts to a reduction strategy.

We now define *Cauchy convergence with colours (CCC)* of a rewrite sequence in the original system $\langle \Sigma, R \rangle$ as Cauchy convergence of the lifted rewrite sequence in $\langle \Sigma', R' \rangle$. So the infinite reduction in CL:

$$\mathsf{SII(SII)} \to \mathsf{I(SII)(I(SII))} \to \mathsf{SII(I(SII))} \to \mathsf{SII(SII)} \to \ldots$$

is lifted to

$$\begin{aligned} \mathsf{SII(SII)} &\to \star(\mathsf{I(SII)(I(SII))}) \to \mathsf{I(SII)(I(SII))} \to \star(\mathsf{SII(I(SII))}) \\ &\to \mathsf{SII(I(SII))} \to \mathsf{SII}(\star(\mathsf{SII})) \to \mathsf{SII(SII)} \to \ldots. \end{aligned}$$

Proposition 2.2. *For all reductions:* SC $\iff$ CCC.

Proof. Both the introduction and the removal of the activity symbol $\star$ cause consecutive terms to differ at the depth of the rewrite step. Hence the depth of the rewrite steps tends to infinity if and only if the sequence is Cauchy convergent. □

Thus we can remove the depth requirement in the definition of SC in favour of a signature extension and the old concept CC. We could view CCC as 'the' definition, and then derive the depth requirement. As an alternative to the activity markers, we could have employed a maximal labelling, see Terese [58, Definition 8.4.14].

Remark 2.3. There is another interesting way to pinpoint the difference between weak and strong convergence, which can be phrased in terms of *reduction loops*. Here we distinguish a 'loop' from a 'cycle': a loop is a reduction cycle consisting of a single reduction step. Now the difference between weak and strong convergence lies in the presence of reduction loops. An inkling of this fact is already seen in the one-rule TRS $C \to C$ seen above: its infinite reductions are weakly but not strongly convergent.

More general, loops arise by reduction rules whose left-hand side is unifiable with its right-hand side; the effect on weak versus strong convergence was noted in [23], but the statement there is flawed. The observation concerning loops and weak versus strong convergence is also present in [55], who arrived independently at it, and, moreover, notes that this fact is also valid in higher-order systems, in particular λ-calculus. The same observation, again arrived at independently, occurred in recent work by Endrullis et al., reported in the unpublished note [18].

2.2. Infinitary properties of transfinite term rewriting

We will now present and discuss the most important properties of infinitary rewriting, as in Table 1. Here the left column states the finitary properties, and the right column states the analogous properties for the infinitary case. Let us briefly enumerate and discuss the most salient facts.

Infinitary confluence. In finite rewriting with orthogonal rewrite systems, even with weakly orthogonal TRSs, we have the confluence property CR. A stepping stone towards CR is PML, the Parallel Moves Lemma, stating that one reduction step set out against a finite reduction admits converging reductions to a common reduct:

$$\begin{tikzcd} s \arrow[rr, ->>] \arrow[d] & & t_1 \arrow[d, "\Vert" marking] \\ t_2 \arrow[rr, dashed, ->>] & & u \end{tikzcd}$$

Table 1
The main properties in finite and infinitary rewriting.

Finitary rewriting	Infinitary or transfinite rewriting
finite reduction	strongly convergent reduction
infinite reduction	divergent reduction (“stagnating”)
normal form	(possibly infinite) normal form
CR: two coinitial finite reductions can be prolonged to a common term	CR^∞: two coinitial strongly convergent reductions can be prolonged by strongly convergent reductions to a common term
UN: two coinitial reductions ending in normal forms, end in the same normal form	UN^∞: two coinitial strongly convergent reductions ending in (possibly infinite) normal forms, end in the same normal form
SN: all reductions lead eventually to a normal form	SN^∞: all reductions lead eventually to a (possibly infinite) normal form, equivalently: there is no divergent reduction
WN: there is a finite reduction to a normal form	WN^∞: there is a strongly convergent reduction to a (possibly infinite) normal form

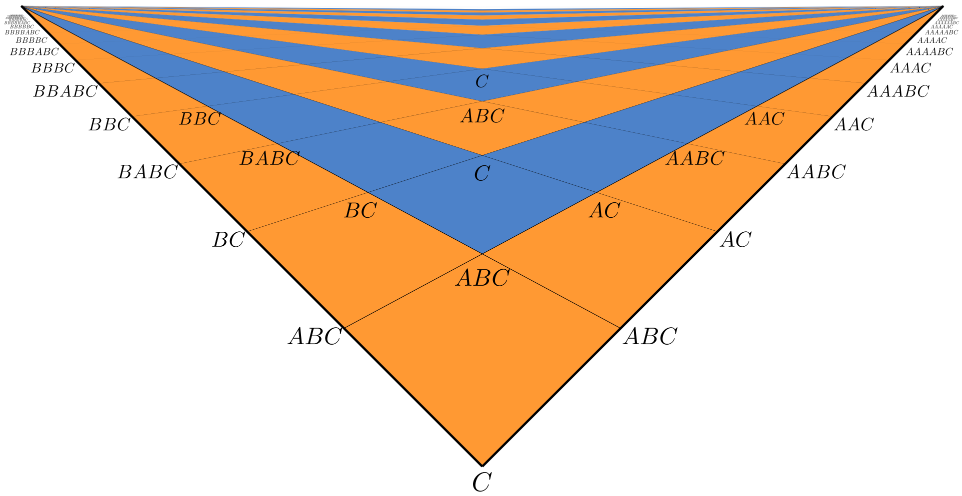


Fig. 3. The *ABC*-example (Example 2.4), in perspective. The reduction graph is rendered such that the distances in the Euclidean metric of the plane respect the tree metric.

The property PML is halfway to CR; a simple induction yields PML $\Longrightarrow$ CR. The generalisation of PML to its infinitary version PML^∞ is straightforward. Now for orthogonal and weakly orthogonal TRSs, we do have PML^∞, but CR^∞ fails, as the following example witnesses.

Example 2.4. Consider the orthogonal TRS with the three rules

$$A(x) \to x \qquad B(x) \to x \qquad C \to A(B(C)).$$

The first two rules are so-called *collapsing rules*, by virtue of their right-hand side being a single variable. Now we have reductions $C \twoheadrightarrow\!\!\!\!\!\rightarrow A^\omega$ and $C \twoheadrightarrow\!\!\!\!\!\rightarrow B^\omega$. Fig. 3 depicts the tiling diagram for these reductions. However, the infinite terms A^ω, B^ω only reduce to themselves; hence CR^∞ fails.

Example 2.5. The ‘*ABC*-example’ that we saw in the preceding example also works in the much more important rewrite system Combinatory Logic CL, with the usual three basic combinators I, K, S and their corresponding reduction rules (see, e.g., Barendregt [3]), and also in infinitary λ-calculus that we will consider in more detail in the next section. Fig. 4 with the infinite collapsing tower of two different *collapsing contexts* K□K and K□S shows how the *ABC*-counterexample can be simulated using a fixed-point construction in those calculi. To see that this is indeed a CR^∞-counterexample, note that $\mu x.\mathsf{K}(\mathsf{K}x\mathsf{S})\mathsf{K} \twoheadrightarrow\!\!\!\!\!\rightarrow \mu x.\mathsf{K}x\mathsf{S}$ and also $\mu x.\mathsf{K}(\mathsf{K}x\mathsf{S})K \twoheadrightarrow\!\!\!\!\!\rightarrow \mu x.\mathsf{K}x\mathsf{K}$, while $\mu x.\mathsf{K}x\mathsf{S}$ and $\mu x.\mathsf{K}x\mathsf{K}$ only reduce to themselves (in any countable ordinal number of steps).

Remark 2.6. The counterexample $\mu x.\mathsf{K}(\mathsf{K}x\mathsf{S})\mathsf{K}$ against CR^∞ gives us a hint as to what is the cause of the failure of CR^∞. First, let us recall the definition of *root-active* term: this is a term admitting an infinite reduction in which infinitely often the root redex is contracted (i.e., the whole term is a redex). Root-active terms are ‘problematic’, they can be considered

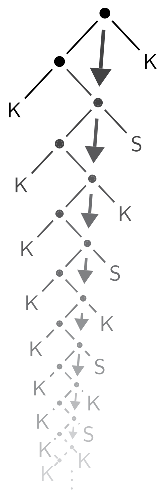


Fig. 4. Counterexample against CR^{∞} of combinatory logic.

as 'undefined': they never will reduce to a term where the root is *stable* and not subject to any further reduction. Indeed, working modulo the set RA of root-active terms, we restore CR^{∞}. Now RA contains a subset HC of *hypercollapsing* terms that is even more problematic or undefined. A hypercollapsing term is one that reduces to an infinite tower of stacked *collapsing contexts*. A context C is collapsing when $C[x] \twoheadrightarrow x$. The last step of such a collapsing reduction is by virtue of a collapsing reduction rule $t \to x$, with a variable as the right-hand side. Thus without loss of generality we may assume that all contexts C a collapsing tower is built of, collapse in a single step.

The notion CR^{∞} is fairly robust: only the hypercollapsing terms cause non-CR^{∞}. Even the root-active but not hypercollapsing terms do not disturb CR^{∞}. We can make this precise using the notion of *family* of a term t, $\mathit{Fam}(t)$ which is the set of all subterms of all reducts of t. The term t and its family $\mathit{Fam}(t)$ are shown in Fig. 5.

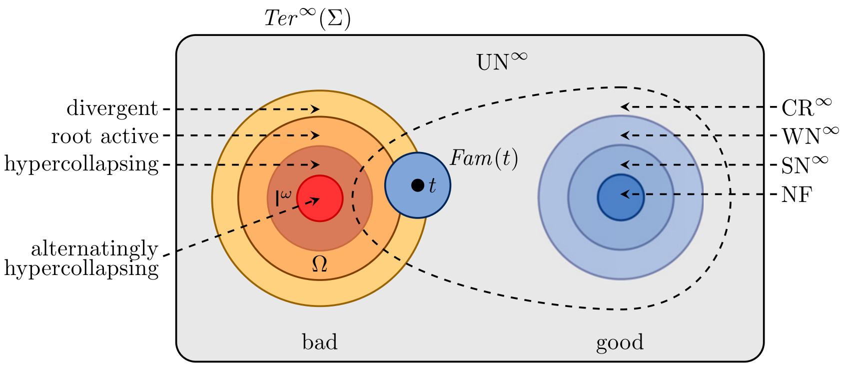


Fig. 5. Root-active and hypercollapsing terms.

Now we have the following theorem:

Theorem 2.7. *For all terms t in an orthogonal TRS, we have*

$$\mathit{Fam}(t) \cap \mathrm{HC} = \varnothing \implies \mathrm{CR}^{\infty}(t).$$

A proof of Theorem 2.7 can be given by the analysis of collapsing rules and ϵ-completion of rules, as mentioned in [26] and [58, Chapter 12, pages 705,706]. To give the intricate proof in its entirety is beyond the scope of this paper.

We conjecture that Theorem 2.7 can be sharpened by introducing a class of *alternating* hypercollapsing terms, reducing to an infinite alternating tower of two 'essentially' non-convertible collapsing contexts, like the term $\mu x.\mathsf{K}(\mathsf{K}x\mathsf{S})\mathsf{K}$.

Unique infinitary normal forms. Even though CR^{∞} fails, fortunately its consequence UN^{∞} does hold [24,38]. *Caveat*: Here it is important that we have orthogonal TRSs; for weakly orthogonal ones, UN^{∞} also fails, as we will see later.

Let us point out a notable consequence of UN^{∞}: for all orthogonal TRSs we have $\mathrm{SN}^{\infty} \implies \mathrm{CR}^{\infty}$, because SN^{∞} & $\mathrm{UN}^{\infty} \implies \mathrm{CR}^{\infty}$. And note that we also have the local version for all terms, i.e., $\forall t.\ \mathrm{SN}^{\infty}(t) \implies \mathrm{CR}^{\infty}(t)$.

Infinitary normalisation. As to infinitary normalisation, there are three noteworthy remarks.

(i) The first pertains to the definition of SN^∞, stating that all reductions eventually will normalise, i.e., reach a normal form. It is important to realise what the negation of this property means, namely that there is a depth n where infinitely many times a redex is contracted. Such a 'stagnation' reveals that the reduction is not strongly convergent, which we call *divergent*. So we can rephrase SN^∞ as stating: *there are no divergent reductions*.
(ii) The second remark is that in finitary rewriting the properties SN and WN as global properties of TRSs have a different strength: SN $\Longrightarrow$ WN but not vice versa. However, in infinitary rewriting (with orthogonal TRSs), we have somewhat surprisingly the equivalence $SN^\infty \iff WN^\infty$. *Caveat:* This is so for the *global* properties SN^∞ and WN^∞; on the term level the properties do have different strength, $SN^\infty(t)$ implies $WN^\infty(t)$, but not necessarily vice versa. For an exposition of these facts see [38].
(iii) Third, infinitary normalisation is closely related to *productivity*, that is, *infinitary constructor normalisation* where the infinite normal forms are required to consist of constructor symbols only. The *constructor symbols* are those symbols that do not occur as root symbols of left-hand sides of the rules. Methods for proving productivity of individual terms have been investigated in [13,15], and methods for proving productivity globally, for all finite terms, are studied in [63,64,19]. Techniques for proving infinitary normalisation have been developed in [62,16]. The properties infinitary normalisation and productivity are of course undecidable, see further [14,11].

Most of the transpositions of the finitary notions to their infinitary counterparts as in Table 1 are straightforward. We stress the basic analogy for infinitary reductions:

finite : infinite = *strongly converging : divergent.*

Infinite ordinals give us a large space to manoeuvre, but often it is convenient to stick to the first infinite ordinal ω. Indeed this can be done, for all orthogonal iTRSs, and even for a somewhat larger class. This is our next stepping stone, stating that for left-linear TRSs every reduction of length α can be compressed to one with the same start and finish, but with finite length, or length ω.

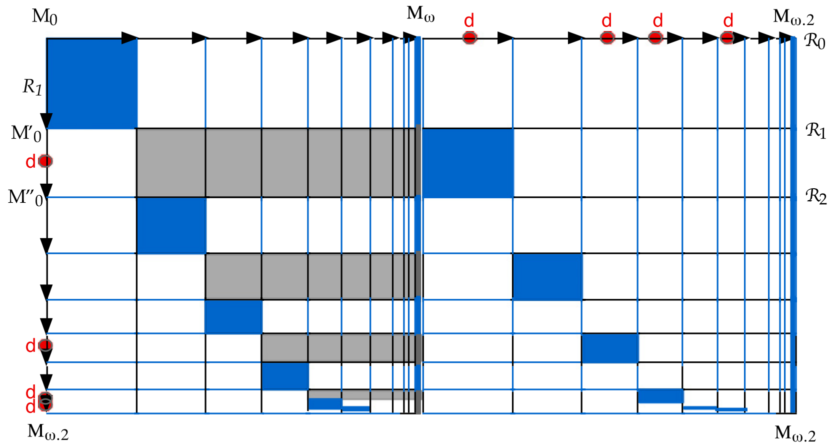


Fig. 6. The umc (uppermost contracted) reflection procedure.

Theorem 2.8 (*Compression Lemma [26,58]*)**.** *For every left-linear TRS we have*

$$t \to^{\alpha} t' \Longrightarrow t \to^{\leq\omega} t'.$$

To see that left-linearity is essential, consider the following TRS:

$$A \to C(A) \qquad B \to C(B) \qquad f(x,x) \to E. \tag{1}$$

Then the reduction $f(A, B) \to^{\omega+1} E$ cannot be compressed to length $\leq \omega$.

Fig. 6 illustrates how standardisation can be employed for compressing reductions to length $\leq \omega$. *Standardisation* is a method of transforming a reduction into a *standard* one, that is, one in which the steps are ordered in a top-down fashion. The original reduction γ_0 of ordinal length is displayed horizontally. Blue steps or reductions are empty. The blue elementary reduction diagrams are the ones in which 'coincidence' takes place; its initial sides are identical, its converging sides empty. Red spots indicate a point of stagnation, divergence, at depth d. (The procedure works for both strongly convergent and divergent rewrite sequences.) This divergence as well as its depth, is reflected into the compressed reduction at the left side, vertical, of the diagram. The right side and the bottom side are empty. The compressed reduction is a permutation of the original one; for orthogonal systems they are known to be Lévy-equivalent [26]. That the projections in the diagram are empty follows immediately from the analysis of reduction diagrams in the infinitary case present in [58, Chapter 12].

We construct the compressed, vertical reduction τ consisting of steps $\tau_0, \tau_1, \ldots$ as follows. For $i \in \mathbb{N}$ we let τ_i contract a fairly chosen redex, outermost among the redexes of which a descendant is contracted in γ_i, and define $\gamma_{i+1} = \gamma_i/\tau_i$ (that is, the projection of γ_i over τ_i). Here, by 'fair' we mean that every redex will be chosen after some finite number of steps. Note that the set of redexes of which a descendant is contracted is never empty unless γ_i is empty. It can be shown that the thus constructed reduction τ is strongly converging and has the same limit as γ_0. (In the case of a divergent sequence γ_0, τ also is divergent.) For more details, we refer to Ketema [31].

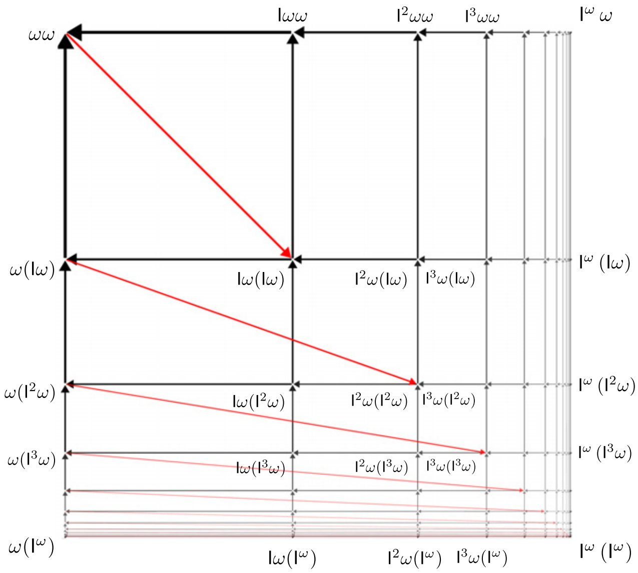


Fig. 7. The infinitary reduction graph of the term $\omega\omega$ with $\omega = \mathsf{SII}$ is not a closed graph. The red reduction steps are root steps. All infinite reductions in this graph are divergent. The accumulation or limit points in the Euclidean metric, as well as in the tree metric, at the right and bottom side, are themselves not $\twoheadrightarrow\!\!\!\!\rightarrow$-reducts, hence not contained in this $\twoheadrightarrow\!\!\!\!\rightarrow$-graph.

Example 2.9. The CL-term $\mathsf{SII}(\mathsf{SII})$ has the infinite reduction graph displayed in Fig. 7. Abbreviating $\omega = \mathsf{SII}$ the terms at the nodes of this graph are $\mathsf{I}^n\omega(\mathsf{I}^m\omega)$ for $n, m \geq 0$. Here are some observations:

(i) All the terms in this reduction graph are root-active, but not hypercollapsing. (Note that the accumulation points containing the subterm I^ω are not part of the reduction graph as they are not the limits of convergent reductions.)
(ii) There are continuum many infinite reductions contained in this reduction graph; all are divergent; in particular they are root-active.

2.3. Infinitary rewrite systems and subsystems

When we compare properties of rewrite systems we must be precise whether we mean finitary rewrite systems or infinitary rewrite systems. In particular we must be precise about the *domain* or universe of our TRS or iTRS. Although most of the time it will be clear from the context what is meant, sometimes some extra precision is desirable. Therefore we define the notion of 'sub-TRS' pertaining to a restriction of the domain (the set of terms), and not to a restriction of the set of reduction rules:

Definition 2.10.

(i) A *finitary* TRS (or TRS for short) $\mathcal{R} = \langle Ter(\Sigma), R\rangle$ over the signature Σ is a pair consisting of the domain $Ter(\Sigma)$, and a set of reduction rules R, generating the reduction relation $\to$ and its reflexive–transitive closure $\twoheadrightarrow$.
(ii) We may also consider TRSs $\mathcal{R}' = \langle T, R\rangle$ based on a subset $T \subseteq Ter(\Sigma)$, which then is required to be closed under $\twoheadrightarrow$. Such a TRS is called a *sub-TRS* of $\mathcal{R}$. Almost always our assertions and theorems about TRSs are in fact pertaining to all sub-TRSs. In case the *domain* T is all of $Ter(\Sigma)$, we call the TRS $\mathcal{R}'$ *full*.
(iii) An *infinitary TRS* (or iTRS) $\mathcal{R} = \langle Ter^\infty(\Sigma), R\rangle$ over Σ consists of the domain $Ter^\infty(\Sigma)$, the set of all finite and infinite terms over the signature Σ, and reduction rules R, generating the infinitary reduction relation $\to^\alpha$, or $\twoheadrightarrow\!\!\!\!\rightarrow$ for unspecified ordinal reduction length.
(iv) Again $\mathcal{R}' = \langle T, R\rangle$ is a *sub-iTRS* of $\mathcal{R}$ if $T \subseteq Ter^\infty(\Sigma)$ is closed under $\twoheadrightarrow\!\!\!\!\rightarrow$, and $\mathcal{R}'$ is called *full* iTRS if $T = Ter^\infty(\Sigma)$.

Definition 2.11. We define canonical transformations from finitary TRSs to iTRSs and vice versa.

(i) If $\mathcal{R} = \langle T, R\rangle$ is a finitary TRS, then $\mathcal{R}^\infty$ is the iTRS $\langle T^\infty, R\rangle$ where T^∞ is the closure of T under $\twoheadrightarrow\!\!\!\!\rightarrow$ in $Ter^\infty(\Sigma)$.
(ii) Vice versa, we obtain from iTRS $\mathcal{R} = (T^\infty, R)$ a finitary TRS $\mathcal{R}^{-\infty}$, by omitting the infinite terms from T^∞.

Remark 2.12. We can now be more precise in our assertions. First let us mention some of the TRSs and iTRSs connected to CL, Combinatory Logic: The full TRS CL has a sub-TRS CL(S) consisting of the finite S-terms. By closure under $\twoheadrightarrow\!\!\!\rightarrow$ it generates $\text{CL}(\text{S})^\infty$, not to be confused with the larger (full) $\text{CL}^\infty(\text{S})$, the sub-iTRS of CL^∞ consisting of all finite and infinite S-terms. Note, by the way, that there is an infinite S-term containing no S's! Now we have:

(i) (Barendregt [3]) $\text{CL}(\text{S}) \nvDash \text{SN}$. The well-known counterexample to SN is $\text{SSS}(\text{SSS})(\text{SSS})$.
(ii) (Waldmann [61]) $\text{CL}(\text{S})^\infty \vDash \text{SN}^\infty$.
(iii) (Zantema, personal communication) $\text{CL}^\infty(\text{S}) \nvDash \text{SN}^\infty$. The counterexample, obtained by unification of left and right-hand sides of the rewrite rule for the S-combinator, is $\text{S}(\text{S}^\omega)TT$ with $T = \mu x.xx$, the infinite binary tree of application nodes. Note that $\text{S}^\omega = \text{S}(\text{S}^\omega)$ and $TT = T$, and so the term is looping:

$$\text{S}(\text{S}^\omega)TT \to \text{S}^\omega T(TT) = \text{S}(\text{S}^\omega)TT.$$

We can write this whole term in μ-notation, $(\mu x.\text{S}x)(\mu y.yy)(\mu y.yy)$.

Remark 2.13. Note that for TRSs $\mathcal{R}$ we have $(\mathcal{R}^\infty)^{-\infty} \supseteq \mathcal{R}$,[1] and vice versa for iTRSs $\mathcal{R}$, $(\mathcal{R}^{-\infty})^\infty \subseteq \mathcal{R}$. In fact, $(\mathcal{R}^{-\infty})^\infty$ is the sub-iTRS consisting of the *finitely generated* terms from $\mathcal{R}$.

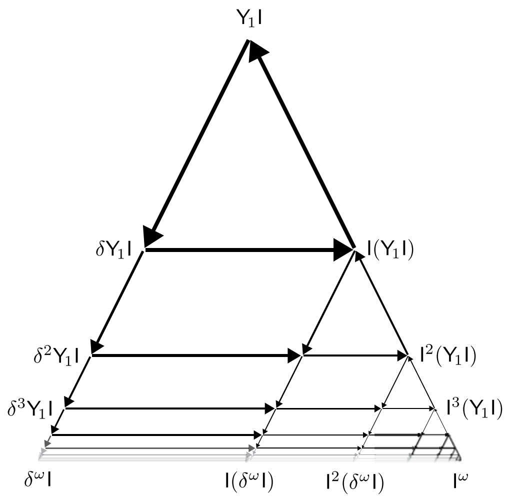


Fig. 8. Infinitary reduction graph of Y_1I, a closed graph.

Example 2.14. (i) Let δ be a constant with the rule $\delta xy \to y(xy)$. In Smullyan [57] δ is called the 'Owl'. Further, we will have a constant ω with the rule $\omega x \to xx$, and constant B with $\text{B}fgx \to f(gx)$. With these constants we can build Turing's fixed point combinator (fpc) Y_1 as ωZ where $\text{Z} = \text{B}\delta\omega$. Then indeed $\text{Y}_1 x \twoheadrightarrow x(\text{Y}_1 x)$, as follows:

$$\text{Y}_1 x = \omega\text{Z}x \to \text{ZZ}x = \text{B}\delta\omega\text{Z}x \to \delta(\omega\text{Z})x = \delta\text{Y}_1 x \to x(\text{Y}_1 x).$$

(ii) Consider the term Y_1I and its reduction graph $\mathcal{G}(\text{Y}_1\text{I})$ in Fig. 8. For the sub-iTRS generated by the combinators S, I, B, δ, ω it is easy to conclude that CR^∞ holds: invoke [26, Theorem 6.10] stating that iTRSs containing only a single non-parameterised collapsing rule (i.e., whose left-hand side contains only one variable) are CR^∞; in [26] these iTRSs are called *almost non-collapsing*.
(iii) In CL we can actually define δ as SI, ω as SII, and B as S(KS)K. For the more complicated iTRS with as domain the points of the graph $\mathcal{G}(\text{Y}_1\text{I})$, and with the rules for I, K, S, the property CR^∞ also holds, as can be seen from the explicit determination of the whole reduction graph. Note that now we cannot invoke [26, Theorem 6.10] due to the rule for K.
(iv) Turing's fpc Y_1 has as infinite normal form δ^ω, which we abbreviate by Δ. This Δ is an example of an infinitary fpc: $\Delta x = \delta\Delta x \to x(\Delta x) \twoheadrightarrow\!\!\!\rightarrow x^\omega$.
(v) $\Delta\Delta$ is an interesting term. We have

$$\Delta\Delta \twoheadrightarrow\!\!\!\rightarrow \Delta^\omega \twoheadrightarrow\!\!\!\rightarrow (\Delta^\omega)^\omega \twoheadrightarrow\!\!\!\rightarrow ((\Delta^\omega)^\omega)^\omega \twoheadrightarrow\!\!\!\rightarrow \cdots .$$

See Fig. 9. Somewhat surprisingly, $\Delta\Delta$ does have a normal form, viz. $\mu x.xx$; and moreover $\Delta\Delta$ has the property SN^∞. To see that $\mu x.xx$ is indeed the normal form, one may consider the reduction

$$\Delta\Delta \twoheadrightarrow\!\!\!\rightarrow (\Delta^\omega)^\omega \equiv \Delta^\omega((\Delta^\omega)^\omega) \twoheadrightarrow\!\!\!\rightarrow (\Delta^\omega)^\omega((\Delta^\omega)^\omega) \twoheadrightarrow\!\!\!\rightarrow \cdots$$

[1] We consider the TRS $\mathcal{R} = \langle T, R\rangle$ where R consists of the rules $A \to C(A)$, $B \to C(B)$ and $f(x,x) \to E$, over the set of terms $T = \{f(C^n(A), C^m(B)) \mid n, m \in \mathbb{N}\}$. Then $(\mathcal{R}^\infty)^{-\infty}$ contains the term E in its domain as a consequence of $f(A,B) \twoheadrightarrow\!\!\!\rightarrow f(C^\omega, C^\omega) \to E$.

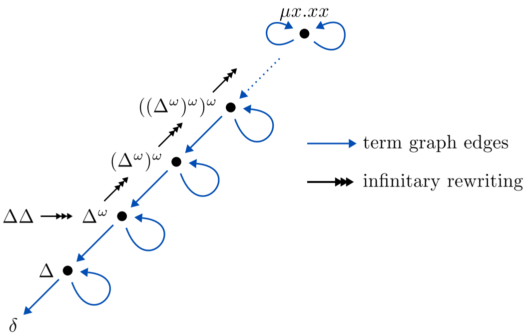


Fig. 9. Cyclic graphs for some reducts of $\Delta\Delta$, getting more and more complex but converging to the relatively simple normal form consisting of application nodes only. All the 'fuel' initially present in the form of the δ's, has been burnt out in the normal form.

and check that the reductions involved do not employ root redexes. (Only in the reduction $\Delta\Delta \twoheadrightarrow \Delta^\omega$ a root step is present; in the 'later' reductions there are no root steps.) In fact we have a strongly convergent reduction

$$\Delta\Delta \twoheadrightarrow \Delta^\omega \twoheadrightarrow (\Delta^\omega)^\omega \twoheadrightarrow ((\Delta^\omega)^\omega)^\omega \twoheadrightarrow \cdots \twoheadrightarrow \mu x.xx.$$

(vi) The term $\Delta\Delta$ has uncountably many reducts. It has reductions of any countable ordinal length. It is SN^∞ with $\mu x.xx$ as its unique normal form. This normal form is in fact a Berarducci tree. The example of $\Delta\Delta$ was also mentioned in [9]. SN^∞ can be proved as follows: We have CR^∞ as there are no collapsing rules in this TRS, which is a fragment (sub-TRS) of CL. Since there is a normal form, we have WN^∞. Hence, SN^∞ follows by the equivalence $\mathrm{SN}^\infty \iff \mathrm{WN}^\infty$ as global properties of TRSs.

2.4. Continuity of infinitary rewriting

Experimenting with several infinitary reduction graphs, we observe that they seem to have a certain closure property, or rather, continuity property. We will make this explicit now.

Definition 2.15. The Continuity Property (CP), is defined as follows:

$$\forall i \in \mathbb{N}.\ t \twoheadrightarrow s_i \text{ and } s = \lim_{i\to\infty} s_i \implies t \twoheadrightarrow s.$$

Note that by requiring $s = \lim_{i\to\infty} s_i$ we tacitly assume that the limit exists. The continuity property holds if and only if $\twoheadrightarrow$ is *pointwise closed*, see further [22, Section 4.1].

Theorem 2.16. *For orthogonal TRSs we have* $\mathrm{SN}^\infty \implies \mathrm{CP}$.

For the proof of the theorem we introduce the notion of *balanced standard* reductions which guarantees that parallel subterms are developed at equal speed. We stress that balancedness does not hold for the usual notion of *parallel standard* reductions [58] as the latter allows for parallel subterms to be ignored indefinitely. For a rewrite sequence σ of length α and an ordinal $\beta < \alpha$, we write $\sigma(\beta)$ to denote the step at index β. We use $\mathsf{pos}(\phi)$ to denote the position of the step ϕ.

Definition 2.17 (*Balanced Standard Reduction*)**.** Let σ be a rewrite sequence of length α. Then σ is *balanced standard* if $\mathsf{pos}(\sigma(\beta))$ is part of the redex pattern of $\sigma(\gamma)$ whenever $\beta < \gamma < \alpha$ and $\sigma(\gamma)$ is the closest step after $\sigma(\beta)$ such that $|\mathsf{pos}(\sigma(\gamma))| < |\mathsf{pos}(\sigma(\beta))|$.

The definition requires that every rewrite step ϕ contributes to the closest step ψ at a higher position; note that the position of ψ is not required to be a prefix of the position of ϕ as in the usual definitions of (parallel) standard reductions. The creation dependency between the steps is displayed in Fig. 10.

$$s_0 \xrightarrow{6} s_1 \xrightarrow{3} s_2 \xrightarrow{2} s_3 \xrightarrow{3} s_4 \xrightarrow{3} s_5 \xrightarrow{1} s_6 \xrightarrow{2} \cdots$$

Fig. 10. Illustration of balanced rewrite sequences. The steps are labelled by their depths; (direct) creation dependencies between the steps are indicated by dashed lines.

Theorem 2.18 (*Balanced Standardisation*)**.** *For every strongly convergent reduction $s \twoheadrightarrow t$ in an orthogonal TRS there exists a balanced standard reduction $s \to^{\leq\omega} t$ of length $\leq \omega$.*

Proof. By compression, we have a reduction $\sigma : s \to^{\leq\omega} t$. Then we transform the reduction σ to a balanced standard reduction by permutation of steps, in a way similar to the procedure in [36]. That is, by permutation we eliminate the 'anti-pairs' that conflict with the definition of balanced standard. Here an *anti-pair* is a subsequence of steps $\sigma(n), \sigma(n+1), \ldots, \sigma(n+k)$ in σ such that $|\mathrm{pos}(\sigma(n))| \leq |\mathrm{pos}(\sigma(n+i))|$ for all $1 \leq i < k$, $|\mathrm{pos}(\sigma(n))| > |\mathrm{pos}(\sigma(n+k))|$ and $\mathrm{pos}(\sigma(n))$ is not in the redex pattern of $\sigma(n+k)$. To transform σ to balanced standard, we repeatedly 'eliminate' the anti-pair $\sigma(n), \ldots, \sigma(n+k)$ such that the tuple $\langle n+k, k\rangle$ is minimal in the lexicographic order. That is, among the anti-pairs that end first, we pick the one that starts last. To eliminate the anti-pair, we permute (project) $\sigma(n)$ over the remainder of the subsequence $\sigma(n+1), \ldots, \sigma(n+k)$. From the choice of the anti-pair it follows that the step $\sigma(n)$ is parallel to $\sigma(n+1), \ldots, \sigma(n+k-1)$, and does not overlap with, but may be nested in or parallel to, the step $\sigma(n+k)$. For finite rewrite sequences σ, the argument for termination of this procedure is precisely as in [36]. For infinite rewrite sequences σ, the construction converges towards a strongly convergent sequence in the limit. This can be seen as follows. For every depth $d \in \mathbb{N}$, the construction terminates on the prefix of σ containing all steps at depth $\leq d$, transforming σ into the form $\sigma_1; \sigma_2$ (i.e., σ_1 followed by σ_2) such that σ_1 is balanced standard and ends with the last step at depth $\leq d$. Since permutations of steps at depth $> d$ cannot create steps at depth $\leq d$, all subsequent permutations of anti-pairs will be in σ_2. $\square$

For balanced standard reductions we obtain the following theorem and corollary providing a bound on the speed of the conversion. We emphasise that these properties do not hold for parallel standard reductions [58].

Theorem 2.19. *Let $\mathcal{R} = (\Sigma, R)$ be an orthogonal TRS, and $t \in Ter^{\infty}(\Sigma)$ a term with $\mathrm{SN}^{\infty}(t)$. For every $d \in \mathbb{N}$, there is only a finite set of balanced standard reductions starting from t and ending with a step at depth $\leq d$.*

Proof. Let σ be a balanced standard reduction that starts from t and ends in a root step. Then from the definition of balanced standard reduction it follows that all steps in this reduction are either root steps, or they are part of a creation chain for a root step in this reduction. As a consequence, all redexes contracted in the reduction σ are root-needed [44]. By $\mathrm{SN}^{\infty}(t)$ the term t admits a reduction to a root-stable form, and by Middeldorp [44, Corollary 5.7] root-needed reduction is root-normalising for orthogonal term rewrite systems. It immediately follows that t contains only finitely many root-needed redexes. Thus $\mathrm{SN}^{\infty}(t)$ implies that root-needed reduction is finitely branching and root-normalising.

Let Φ be the set of balanced standard reductions starting from t and ending in a root step. By König's lemma and by the above reasoning, the set Φ is finite, and each of the reductions in Φ is finite.

Let us consider a term of the form $s = f(s_1, \ldots, s_n)$. Then $(*)$ any balanced standard reduction γ starting from s, not containing root steps and ending in a step at depth d, can be seen as an interleaving (and placing in context) of balanced standard reductions $\gamma_1, \ldots, \gamma_n$ on the terms $s_1, \ldots, s_n$ each ending in a step at depth $\leq d-1$ (the '-1' stems from the removal of the context $f(\ldots, \Box, \ldots)$). From γ we obtain $\gamma_1, \ldots, \gamma_n$ by selecting the steps within the corresponding subterms. These selections are balanced standard again, since a step cannot contribute to another step in a parallel subterm; thus the interleaving of parallel reductions only results in additional requirements. The reductions $\gamma_1, \ldots, \gamma_n$ cannot end with a step ξ at depth $\geq d$ since then the last step of γ would be at a lower depth, and thus from a reduction in a parallel subterm (to which ξ cannot contribute).

Let $T = \{f_i(s_{i,1}, \ldots, s_{i,ar(f_i)}) \mid 1 \leq i \leq |\Phi|\}$ be the set of end terms of reductions in Φ. Every balanced standard rewrite sequence starting from t and ending in a step at depth $\leq d$ consists of a prefix in Φ (or an empty prefix) resulting in a term in T, and a suffix containing no root steps. We have already seen that Φ is finite, and by $(*)$ this suffix is an interleaving of reductions ending with a step at depth $\leq d-1$ on the subterms. Thus the claim follows by induction on $d \in \mathbb{N}$. $\square$

The following corollary is immediate.

Corollary 2.20 (*Modulus of Convergence*)**.** *Let $\mathcal{R} = (\Sigma, R)$ be an orthogonal TRS, and $t \in Ter^{\infty}(\Sigma)$ a term with $\mathrm{SN}^{\infty}(t)$. Every balanced standard reduction starting from t has length ω at most. Moreover, there exists a* modulus of convergence $\nu_t : \mathbb{N} \to \mathbb{N}$ *such that for every depth $d \in \mathbb{N}$ and every balanced standard reduction σ starting from t we have $|\mathrm{pos}(\sigma(n))| > d$ for all $n \geq \nu_t(d)$.*

We are now prepared for the proof of Theorem 2.16.

Proof of Theorem 2.16. For $i \in \mathbb{N}$ let $\sigma_i : t \twoheadrightarrow s_i$ be given. By Theorem 2.18 we may assume that the reductions are balanced standard. Let $I_0 = \mathbb{N}$ and for $d = 0, 1, \ldots$ define infinite sets $I_{d+1} \subset \mathbb{N}$ as follows. Let $d \in \mathbb{N}$. For every reduction σ_i with $i \in I_d$ we consider the prefix $\tau_{i,d}$ of σ_i ending with the last step at depth $\leq d$. By Theorem 2.19 there is only a finite number of these prefixes, and thus by the pigeonhole principle there is one prefix τ_d that occurs infinitely often. We then let $I_{d+1} = \{i \mid i \in I_d,\ \tau_{i,d} = \tau_d\}$.

As I_d is infinite for every $d \in \mathbb{N}$, the limit is preserved, that is, we have $\lim_{i \in I_d, i \to \infty} = s$. Moreover, for $d > 0$ we have $I_d \supseteq I_{d+1}$ and the sequences $\{\sigma_i \mid i \in I_d\}$ coincide on the prefix up to the last step of depth $\leq d$. Thus the sequences converge towards a strongly convergent rewrite sequence with limit s. $\square$

An alternative proof of Theorem 2.16 departing from the Standard Prefix Lemma [39, Lemma 1], was recently suggested to us by Vincent van Oostrom. This, however, would first require a generalisation of the Standard Prefix Lemma to the infinite setting.

Remark 2.21 (*Necessity of the Conditions in Theorem 2.16*).

(i) Orthogonality is necessary. For a non-orthogonal counterexample consider the following TRS (similar to [12, Definition 6.3] and [22, Example 4.5]):

$$c \to b(c) \qquad b(c) \to a(d) \qquad b(a(x)) \to a(a(x)).$$

Then $c \twoheadrightarrow b^n(c) \to b^{n-1}(a(d)) \twoheadrightarrow a^n(d)$ for all $n \in \mathbb{N}$, but not $c \twoheadrightarrow\!\!\!\!\!\!\twoheadrightarrow a^\omega$.

(ii) $\mathcal{G}(\mathsf{SII}(\mathsf{SII})) \not\models$ CP. This is because SN^∞ does not hold for the terms in this reduction graph, which is depicted in Fig. 7.

(iii) CR^∞ is not enough to imply CP. Consider the following rewrite rules

$$\begin{array}{lll} bU \to Ua & & \text{(walk up)} \\ tU \to tD & & \text{(turn at the top)} \\ Da \to bD & & \text{(walk down)} \\ Ds \to Uas & & \text{(turn at the bottom).} \end{array}$$

This system is orthogonal and has no collapsing rules, so it is CR^∞. We have:

$$tDs \to tUas \twoheadrightarrow tUaas \twoheadrightarrow tUaaas \twoheadrightarrow \cdots .$$

But not $tDs \twoheadrightarrow\!\!\!\!\!\!\twoheadrightarrow tU(a^\omega)$. Note, however, that this TRS is not SN^∞.

3. Infinitary λ-calculus

After our exposé of infinitary rewriting for first-order orthogonal TRSs, we now turn to the same for $\lambda\beta$-calculus [27]. For a generalisation of $\lambda^\infty\beta$-calculus to infinitary Combinatory Reduction Systems, we refer to [32,29,33–35]. At the end of this section we will also look at the infinitary extension of the $\lambda\beta\eta$-calculus, but there we encounter a negative state of affairs. As to $\lambda^\infty\beta$-calculus, the same notion of strongly convergent reduction applies. In this paper we will gloss over the details of taking care of α-conversion (renaming of bound variables); for a treatment of that issue we refer to [27,58,40]. The notion of β-reduction is entirely straightforward, we will not spell this out here. As before, the Compression Lemma holds, and, also as before, CR^∞ fails. In fact, now even the infinitary Parallel Moves Lemma, PML^∞, fails. Let us prove this fact.

Proposition 3.1 (*[27]*). *The properties* PML^∞*, and hence also* CR^∞*, fail in* $\lambda^\infty\beta$*-calculus.*

Proof. Let $\mathsf{Y}_0 = \lambda f.\omega_f\omega_f$ with $\omega_f = \lambda x.f(xx)$ and consider $\mathsf{Y}_0\mathsf{I}$. Then on the one hand $\mathsf{Y}_0\mathsf{I} \to_\beta (\lambda x.\mathsf{I}(xx))(\lambda x.\mathsf{I}(xx)) \twoheadrightarrow\!\!\!\!\!\!\twoheadrightarrow \mathsf{I}^\omega$, and on the other hand $\mathsf{Y}_0\mathsf{I} \to_\beta (\lambda x.\mathsf{I}(xx))(\lambda x.\mathsf{I}(xx)) \to_\beta^2 \Omega = (\lambda x.xx)(\lambda x.xx)$. Both I^ω and Ω reduce only to themselves, so they have no common reduct. □

We continue the analogy with the first-order case. Also now we have $\lambda^\infty\beta \models \mathrm{UN}^\infty$; unicity of infinite normal forms is guaranteed. Of course, a(n infinite) normal form is just a term without β-redex. As for the first-order case, we will have a brief look at what constitutes the difference between weak and strong convergence, now for infinitary β-reductions. The same remark as before about CCC, coloured Cauchy convergence, applies. And again, see Remark 2.3, the difference between weak and strong convergence manifests itself in the presence of β-reduction loops $M \to_\beta M$.

3.1. Looping terms

A *looping term* simply is a term M such that $M \to M$. For the finite $\lambda\beta$-calculus, the only looping terms are terms which have Ω as a subterm, see Lercher [41]. For the infinitary $\lambda\beta$-calculus, it is non-trivial to characterise the looping terms. This characterisation has been found by Polonsky and Endrullis [46].

Obviously we have:

(i) If $M \to_p M$ at position p, then $M|_p$ is looping.
(ii) If M is looping, then any term $C[M]$ is

and therefore the interesting cases are the terms that loop via a root step; we call these *root-looping terms*. In infinitary λ-calculus, a term is root looping if and only if it is of one of the following forms:

(i) Ω
(ii) I^ω
(iii) BB where B is the infinite solution of $B = \lambda x.xB$,
(iv) $(\lambda v_0.(\lambda v_1.(\lambda v_2.\ldots)t_2)t_1)t_0$ such that t_i is obtained from t_{i+1} by replacing v_0 by t_0 and all variables v_{j+1} by v_j. We call such a term a *cascade*.

Note that item (iv) is an infinite scheme of looping λ-terms, illustrated in Fig. 11. An example of a looping term is depicted in Fig. 12.

For the first-order case we have a complete characterisation of what causes the failure of CR^∞ for orthogonal TRSs. It is due to the presence of either two collapsing rules, as in the *ABC*-counterexample (Example 2.4), or to the presence of a parameterised collapsing rule like $\mathsf{K}xy \to x$ in CL (Example 2.5), see [26, Theorem 6.10].

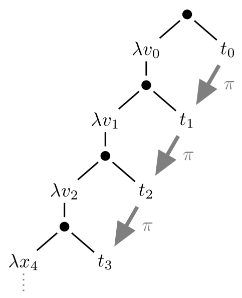


Fig. 11. The shape of cascades; here π stands for replacing all variables v_j by v_{j+1} followed by replacing an arbitrary (possibly infinite) number of occurrences of t_0 by v_0.

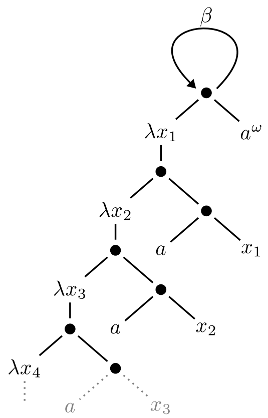


Fig. 12. An infinite looping λ-term.

For $\lambda^\infty\beta$ the failure of CR^∞ is a far more complicated phenomenon, see also [33]. We saw the counterexample given by the term $\mathsf{Y}_0\mathsf{I}$ in the proof of Proposition 3.1, reducing to both Ω and I^ω. But there are several counterexamples to CR^∞ that seem quite different. One counterexample is in fact given by the looping term in Fig. 12. A simpler one is in Fig. 13.

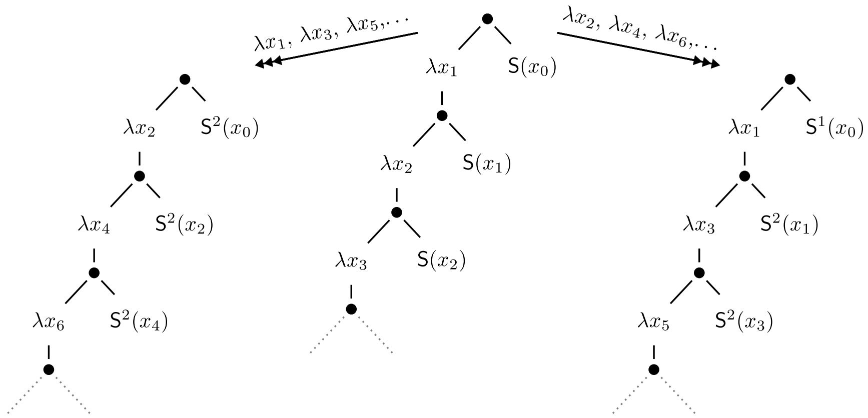


Fig. 13. Another counterexample to CR^∞ of $\lambda^\infty\beta$-calculus.

We will explain the interesting proofs that they indeed are CR^∞ counterexamples in the following two examples, both employing ARSs, abstract reduction systems. We describe first the easier one.

Example 3.2. Consider the ARS $\mathcal{A} = \langle \mathbb{N}^\omega, \to \rangle$ with as domain the set of streams of natural numbers, and reduction relation $\to$ consisting of the operation of adding two consecutive entries in the stream. Now it is easy to see that the element 1^ω is not CR^∞, as it reduces infinitarily to both 2^ω and 12^ω, two streams that have no common $\twoheadrightarrow\!\!\!\!\rightarrow$-reduct. It is easy to see that the reduction graph of the infinite λ-term in Fig. 13 is in fact isomorphic to the reduction graph $\mathcal{G}_{\!\mathcal{A}}(1^\omega)$ in this ARS $\mathcal{A}$.

Example 3.3. Now we consider the ARS $\mathcal{A} = \langle (\overline{\mathbb{N}})^{\omega}, \rightarrow \rangle$ consisting of the streams of *extended natural numbers* $\overline{\mathbb{N}} = \mathbb{N} \cup \{\infty\}$. The reduction relation is again the addition of two consecutive stream entries, now with the understanding that $n + \infty = \infty + n = \infty$. Now consider the stream $\infty 111\ldots$, corresponding in fact to the infinite looping term in Fig. 12. Also the reduction graph of this looping term is isomorphic to that of the stream as mentioned. That it is non-CR$^{\infty}$ is a nice puzzle, which we offer in particular to Yoshihito knowing his talent for devising and solving puzzles.

3.2. A topography of notions of 'undefined' in λ-calculus

In Section 2 we have zoomed in on the localisation of good and bad properties for infinitary first-order rewriting. Several of these notions have analogous counterparts in finite and infinitary λ-calculus, but we will have now a fresh look at the situation for λ-terms.

Just as for the first-order case, we find that equating a class of problematic terms restores CR$^{\infty}$. There it was tied up to hypercollapsing terms, but in the $\lambda^{\infty}\beta$-calculus it is more complicated as there is more choice in adopting a certain class as 'undefined' terms and then identifying them. The most well-known way is the one of Böhm trees. But there are two other canonical choices as we will see now. Besides these three paradigm notions of undefined, there are continuum many other possibilities, satisfying some basic requirements for 'undefined'.

For the three paradigm semantic frameworks there are important motivations: Böhm trees (BTs) [6], the most 'classical' one, is intimately connected to the theory of the model $\mathcal{P}\omega$, Lévy–Longo trees (LLTs) [42,43] has originated from desiderata that arose in the practice of functional programming languages, and Berarducci trees (BeTs) [5] came from consistency studies (which terms can be consistently equated; 'easy' terms).

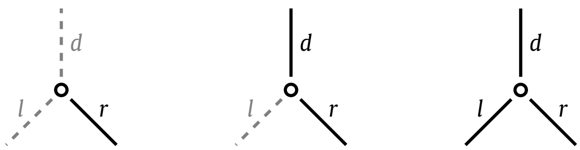
All the different notions of 'undefined' directly give rise to models for finite and infinitary λ-calculus. So in order to have a better view on the model theory of λ-calculus it is important to develop a *topography of notions of undefined*.

3.2.1. The threefold path

Böhm trees provide a semantics of λ-calculus where terms without head normal form are considered meaningless. In fact, this semantical view is one of three canonical semantical frameworks that arise in a uniform way by considering the three dimensions d, l, r in which λ-terms can grow:

d *down*, in an abstraction;
l *left* in an application;
r *right* in an application,

see Fig. 14.



λ_{001} λ_{101} λ_{111}

Fig. 14. Suppressed dimensions.

Each of these three dimensions d, l, r can be 'suppressed' in counting the depth of an occurrence in a λ-term, giving rise *a priori* to eight possible semantics, that are indicated by tuples $000, \ldots, 111$ stating which of the directions d, l, r, is nullified (0), or counted (1). For example, the 110-depth counts only d- and l-steps, disregarding the r-steps. Using this notion of depth in a term, we define the usual 2^{-n} notion of distance between λ-terms, referring to the least depth n where they differ. After metric completion this leads to eight complete metric spaces of finite and infinite λ-trees. They are equipped with generalisations of the finitary notions of substitution, α-conversion and β-reduction. Of these λ_{dlr}-calculi, λ_{000} is trivial as an infinitary calculus: it is the finite λ-calculus. Four others, λ_{010}, λ_{011}, λ_{100}, and λ_{110}, have to be discarded as they lack some basic properties, such as substitutivity of the reduction relation, see further [27].

Three remain: λ_{001}, λ_{101}, and λ_{111}, see Figs. 14 and 15. It turns out that these three infinitary calculi λ_{001}, λ_{101}, and λ_{111} when extended with the obvious Ω-rules (rules for replacing undefined terms with Ω, rules for moving the Ω's upwards; here Ω is understood to be a symbol) to get rid of meaningless terms (to wit, terms without head normal form (hnf), terms without weak head normal form (whnf), and 'mute' terms without root stable form, respectively), are the natural habitats for the three well-known notions of infinite λ-trees: λ_{001} contains the Böhm trees $\mathtt{BT}(M)$, with M a λ-term, λ_{101} contains the Lévy–Longo (or lazy) trees $\mathtt{LLT}(M)$, and λ_{111} contains the Berarducci trees $\mathtt{BeT}(M)$. In all three infinitary λ-calculi we obtain the Böhm trees, the Lévy–Longo trees, and the Berarducci trees in a uniform way as infinitary normal forms.

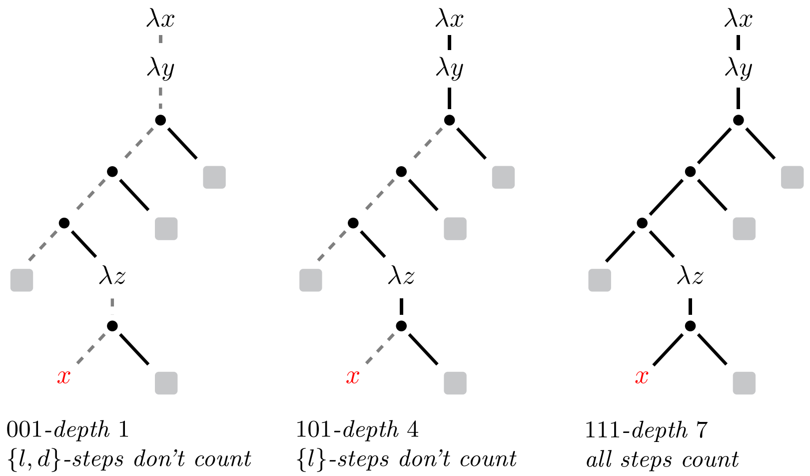


Fig. 15. Depth count of an occurrence of x in the three paradigm semantics.

Table 2
Survey of BT-LLT-BeT properties.

tree family	BT		LLT		BeT
dimensions d, ℓ, r	001		101		111
domain $Ter(\lambda_{d\ell r})$	$Ter(\lambda_{001})$	$\subseteq$	$Ter(\lambda_{101})$	$\subseteq$	$Ter(\lambda_{111})$
strategic redex	spine $\Leftarrow$ head	$\Leftarrow$	lazy	$\Leftarrow$	root
dlr-unsolvable	no hnf	$\Leftarrow$	no whnf	$\Leftarrow$	mute, no rnf
Ω-rules	$\Omega M \to \Omega$, $\lambda x.\Omega \to \Omega$		$\Omega M \to \Omega$		none
refinement	BT(M)	$\leq_\Omega$	LLT(M)	$\leq_\Omega$	BeT(M)
$\lambda\beta dlr$	$dl\beta$		$l\beta$		β
$\lambda\beta dlr$-normal forms	HNF	$\subseteq$	WHNF	$\subseteq$	non-redexes

In Table 2 we give a complete survey of the notions involved. The last row of this table describes the normal forms with respect to reduction at depth 0 in the respective metric; we refer to Table 4 for a characterisation of these redexes.

Definition 3.4.

(i) A term is a *head normal form (hnf)* if it is of the form $\lambda\vec{x}.y\vec{M}$ with $\vec{x} = x_1 \ldots x_n$ and $\vec{M} = M_1 \ldots M_m$.
(ii) A term is a *weak head normal form (whnf)* if it is an *abstraction* $\lambda x.M$ or a *vector* $xM_1 \ldots M_m$ where x is a variable.
(iii) A term is a *root normal form (rnf)*, or *root-stable*, if it is a variable, an abstraction $\lambda x.M$, or an application MN where M does not reduce to an abstraction.

The definition of the Böhm tree BT(M) of M is classic, and likewise that of the Lévy–Longo tree or lazy tree LLT(M). For completeness sake we repeat the definitions. See Table 3 and Fig. 16 for examples of these kinds of trees.

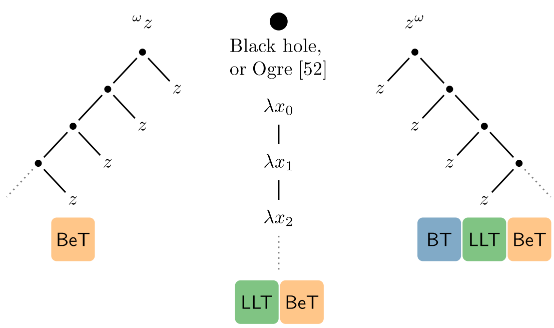


Fig. 16. Three infinite λ-terms. The colour flags mention to which families of trees they belong.

Table 3
BT, LLT, BeT-examples.

M	$\mathsf{BT}(M)$	$\mathsf{LLT}(M)$	$\mathsf{BeT}(M)$
$(\lambda x.xx)(\lambda x.xx)$	Ω	Ω	Ω
$(\lambda xy.xx)(\lambda xy.xx)$	Ω	$\bullet$	$\bullet$
$(\lambda x.xxz)(\lambda x.xxz)$	Ω	Ω	${}^{\omega}z$
$(\lambda x.z(xx))(\lambda x.z(xx))$	z^{ω}	z^{ω}	z^{ω}
$\lambda y.((\lambda x.xx)(\lambda x.xx))$	Ω	$\lambda y.\Omega$	$\lambda y.\Omega$
$(\lambda x.xx)(\lambda x.xx)y$	Ω	Ω	Ωy

Definition 3.5 (*Böhm trees*, $\mathsf{BT}(M)$).

$$\mathsf{BT}(M) = \begin{cases} \lambda\vec{x}.y\,\mathsf{BT}(M_1)\dots\mathsf{BT}(M_m) & \text{if } M \text{ has hnf } \lambda\vec{x}.yM_1\dots M_m, \\ \Omega & \text{otherwise.} \end{cases}$$

Definition 3.6 (*Lévy–Longo trees*, $\mathsf{LLT}(M)$).

$$\mathsf{LLT}(M) = \begin{cases} x\,\mathsf{LLT}(M_1)\dots\mathsf{LLT}(M_m) & \text{if } M \text{ has whnf } x\,M_1\dots M_m, \\ \lambda x.\mathsf{LLT}(M') & \text{if } M \text{ has whnf } \lambda x.M', \\ \Omega & \text{otherwise.} \end{cases}$$

A *term of order* 0 is a term that cannot be β-reduced to an abstraction term. A term M is *mute* [5] if it is a term of order 0 which cannot be reduced to a variable or to an application M_1M_2 with M_1 a term of order 0. Equivalently: M has an infinite reduction with at the root infinitely many times a redex contraction.

Definition 3.7 (*Berarducci trees*, $\mathsf{BeT}(M)$).

$$\mathsf{BeT}(M) = \begin{cases} y & \text{if } M \twoheadrightarrow y, \\ \lambda x.\mathsf{BeT}(N) & \text{if } M \twoheadrightarrow \lambda x.N, \\ \mathsf{BeT}(M_1)\,\mathsf{BeT}(M_2) & \text{if } M \twoheadrightarrow M_1\,M_2 \text{ such that } M_1 \text{ is of order } 0 \\ \Omega & \text{in all other cases (i.e., when } M \text{ is mute).} \end{cases}$$

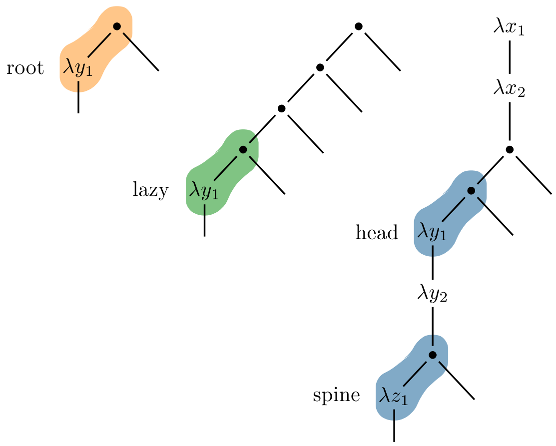


Fig. 17. The strategic redexes: root, lazy, head and spine.

3.2.2. Strategic redexes: root, head, lazy and spine redex

To have a spine is very important, and for λ-terms it is the same. In fact, on the spine of a λ-term all the 'important' redexes are located. We will call them *strategic* redexes; they are known as *root*, *head* [4], *lazy* [1] and *spine redex* [4]. The spine of a λ-term, finite or infinite, is the leftmost path when viewing the term as a tree, that is, it is the maximal *dl*-branch consisting precisely of those positions that do not contain 2 (we never take a right branch of an application). Redexes whose

Table 4
Characterising redexes at depth 0, due to [59]. The rules d, l, r are also known as ξ, ν, μ [45].

$$\frac{}{(\lambda x.M(x))N \to M(N)}\ \beta$$

$$\frac{M \to N}{\lambda x.M \to \lambda x.N}\ d \qquad \frac{M \to N}{MZ \to NZ}\ l \qquad \frac{M \to N}{ZM \to ZN}\ r$$

patterns are on the spine are spine redexes. The uppermost one is the head redex. It is the root redex if its root is that of the whole term.

The definition is illustrated in Fig. 17, and proceeds, informally, as follows. In the BT (001) sense, there may be several redexes at depth 0, the spine redexes; the uppermost one in the syntactic sense is the head redex. In the LLT (101) sense, there is at most a unique redex at depth 0, which is the lazy redex. In the BeT (111) sense, there is at most one, unique, redex at depth 0, the root redex.

An elegant characterisation of depth-0 redexes is due to de Vries [59]. Depending on which of the derivation rules d, l, r is adopted, the inference systems given in Table 4 allows just the redexes of dlr-depth 0 to be contracted; e.g., with rules β, d, l we have spine reduction; with β, l we have lazy reduction, and with only β we have root reduction. The normal forms for these three notions of reduction are the hnfs, the whnfs, and the non-redexes, respectively.

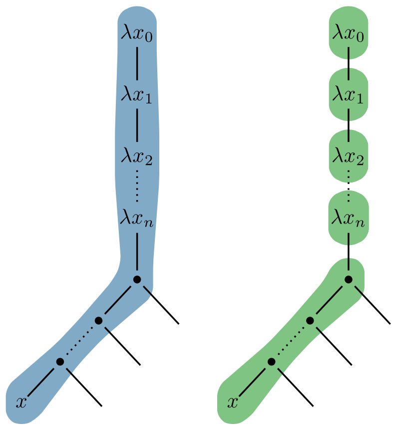


Fig. 18. A pair of socks: building blocks for BTs and LLTs.

For BTs and LLTs, the 'building blocks' are as depicted in Fig. 18. Note that in [3] another notation is used, which may be called the hnf-notation; there a 'building block' is obtained by pinching together the boomerang shaped figure of the form d^*l^* ending in a variable. (The left sock in Fig. 18.) Then we obtain the building block $\lambda\vec{x}.y$. The building blocks for LLTs are sub-blocks of those for BTs. And in turn, the building blocks for BeTs are even smaller sub-blocks, namely application nodes, abstraction nodes λx, variables, Ω. So the composition or decomposition of the building blocks parallels the refinement order in $\mathsf{BT}(M) \leq_\Omega \mathsf{LLT}(M) \leq_\Omega \mathsf{BeT}(M)$. In a slogan:

The finer the building blocks, the finer the semantics.

3.2.3. Head-normalisation theorems

A classical theorem in λ-calculus states that if a λ-term M has an infinite head reduction, it does not have (i.e., reduces to) an hnf, see [3, Theorem 8.3.11]. We abbreviate this as $M \in \Box\mathsf{head}$: M admits a reduction of which every step is a head step.

A stronger version, sometimes called the 'quasi-head normalisation theorem' [3, Exercise 13.6.13], states that if M has an infinite reduction with infinitely many head steps, it does not have an hnf. So here one is allowed to do something arbitrary in between the head steps. We abbreviate this as $M \in \Box\Diamond\mathsf{head}$. These notions are in fact equivalent; the proof is by pushing all the head steps to the front of the reduction sequence using some commuting diagrams. (See the proof of Theorem 13.2.6 in [3], there for quasi-leftmost reduction.)

We now capture in one diagram all these head/lazy/spine normalisation theorems, both in the $\Box$ and the $\Box\Diamond$-sense, see Fig. 19. The proofs are very much in the spirit of the one indicated above for $\Box\mathsf{head}$ and $\Box\Diamond\mathsf{head}$; see also [4].

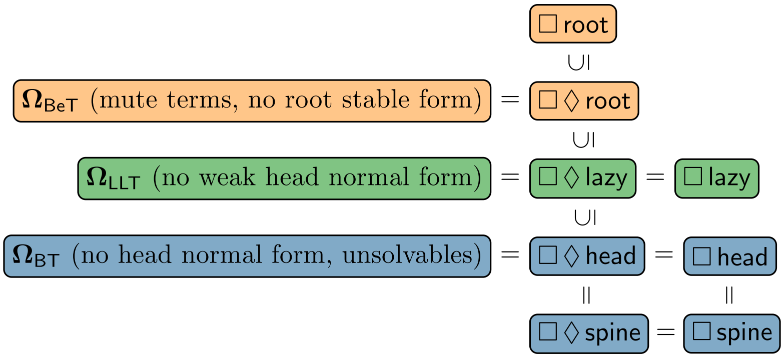


Fig. 19. Head-normalisation theorems.

3.2.4. *Continuum notions of 'undefined'*

Apart from the three main notions of undefined as given by the BT, LLT, BeT trichotomy, there are many more, in fact $2^{2^{\aleph}}$ many, that satisfy some 'reasonable' requirements. An analysis of what are these 'axioms' for notions of undefined has been made in [2,28,58,30,51,52]. The result of this analysis is that the important properties CR^{∞}, UN^{∞} are then uniformly proved for this large class of notions of undefined. And this yields just as many models for the λ-calculus. One might ask whether all these notions of undefined also have an accompanying notion of 'strategic' redex, like root, lazy, head. Such redex contractions should lead to defined results, like BT, LLT, BeT, if they exist; and if an infinite reduction exists with infinitely many contractions of a strategic redex, the begin term should be undefined.

3.2.5. *Lambda theories*

The syntactic analysis of finite and infinitary λ-calculus sheds more light on some of the main models of λ-calculus, $\mathcal{P}\omega$. It is long known that the theory of $\mathcal{P}\omega$ (i.e., all equations $M = N$ true in $\mathcal{P}\omega$) is that of BT-equality. It is interesting that we can split up this equality in two 'orthogonal' components: on the one hand there is equating all unsolvables (i.e., terms M with $\mathrm{BT}(M) = \Omega$), called the theory $\mathcal{H}$ in [3]; on the other hand, there is the 'infinite expansion' given by the theory of $\lambda^{\omega}\beta$. The supremum of both theories is the theory of $\mathcal{P}\omega$.

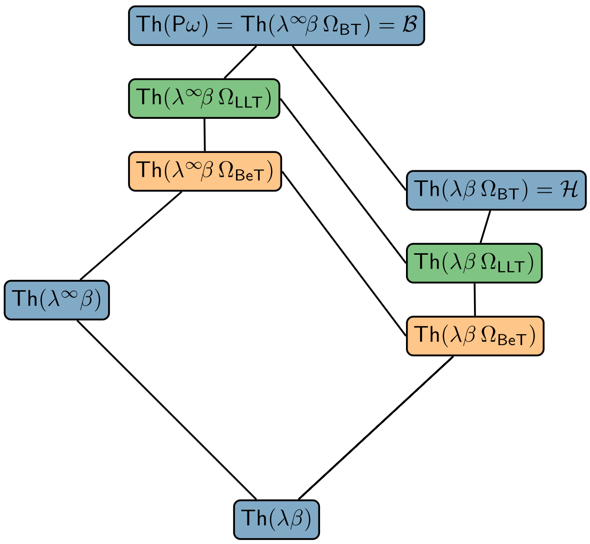


Fig. 20. Partial order of λ-calculus theories.

Fig. 20 gives the partial order of these theories, for the three different frameworks. The $\mathcal{B}$ in that figure is the theory of BT-equality described first in [3, Section 18.4]. This can be seen as a precursor of our $\lambda^{\infty}\beta\Omega_{\mathrm{BT}}$. BTs are there applied to each other by first taking their projections up to depth n, then applying these finite BTs to each other, and finally taking the limit. (It would be an interesting student assignment to prove the equivalence with the more direct set-up via the present $\lambda^{\infty}\beta\Omega_{\mathrm{BT}}$-calculus.) A related partial order of λ-theories is given in [3, p. 464] (there taking into account both the η-rule and the ω-rule). See also [50–53] where in addition the partial order of meaningless sets of terms is investigated. (As a second interesting research question, we suggest taking η and ω into account, starting from the partial order above.)

3.2.6. Restoring infinitary confluence by quotienting undefined terms

It is a pitfall to think that all normal forms from $\lambda^\infty\beta$-calculus are BTs. To see what is the difference, we formulate the following theorem. Of course one can characterise the normal forms from $\lambda^\infty\beta$-calculus in a *negative* way, by stating that they do not contain (the pattern of) a β-redex; but this does not give insight in their structure, from what components they are built, see Fig. 21. Now we see that the components with infinite spine are not possible in a Böhm tree. On the other hand, the normal forms from $\lambda^\infty\beta$-calculus are BeTs, Berarducci trees. Below we will use this fact.

It is interesting to consider the question which BTs are actually realisable by finite λ-terms, i.e., which of them are finitely generated. Note that we can compose continuum many BTs with their building blocks as given in Fig. 18, or equivalently, as normal forms of infinitary $\lambda^\infty\beta\Omega_{\mathsf{BT}}$-calculus. This question is answered in [3, Theorem 10.1.23], in the way one would expect; all and only the computably enumerable BTs are finitely generated, of course provided they have only finitely many free variables. Interestingly, this characterisation is much more subtle for the λI-version of the BTs; it then requires moreover the computability of a *variable indicator*, see [3, Theorem 10.1.25]. It would be interesting to do this exercise also for the case of LLT and BeT.

Theorem 3.8. *The normal forms from $\lambda^\infty\beta$-calculus are built (coinductively) from the four building block types as in Fig. 21, namely a variable, hnf-contexts, the Ogre [52], and d^*l^ω-terms.*

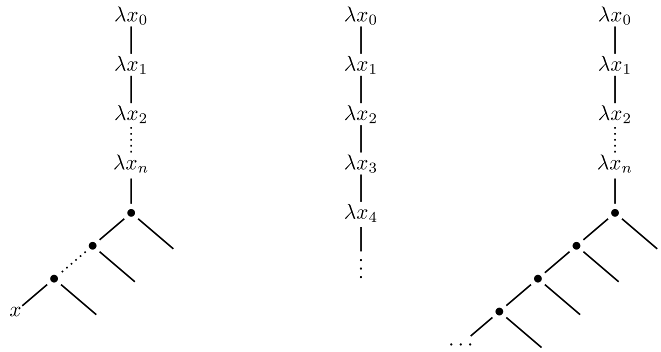


Fig. 21. Building blocks for $\lambda^\infty\beta$-normal forms.

Proof. That a possibly infinite composition of these building blocks contains no β-redex, and hence is a $\lambda^\infty\beta$-normal form, is clear.

Vice versa, given a $\lambda^\infty\beta$-normal form, we construct such a decomposition as follows. Colour the d, l-steps blue, and the r-steps red (see Fig. 14). (In the figure of the building blocks this is already done.) Now consider maximal connected blue fragments in the tree. This constitutes the desired composition, together with the occurrences of variables at the end of some branches. □

Table 5
The main properties for the λ-calculi.

	$\lambda^\infty\beta$	$\lambda^\infty\beta\eta$	λ_{001}	λ_{101}	λ_{111}
CR^∞	no	no	yes	yes	yes
UN^∞	yes	no	yes	yes	yes

We end this section with Table 5 summarising the main properties of the different λ-calculi, and remark that UN^∞ for $\lambda^\infty\beta$ is a corollary of UN^∞ for λ_{111} via de Vrijer's Triple Extension Lemma stated below, cf. [60, Proposition 3.1] for a slightly different version. This lemma states that if we can find an extended ARS such that the domain is extended, the reduction relation is extended, and the set of normal forms is extended, then UN of the extension implies UN of the original ARS.

Lemma 3.9 (*Triple Extension Lemma*)**.** *Let $\mathcal{A}_1 = \langle A_1, \to_1\rangle$ and $\mathcal{A}_2 = \langle A_2, \to_2\rangle$ be ARSs having normal forms $\mathrm{NF}_1 \subseteq A_1$ and $\mathrm{NF}_2 \subseteq A_2$ respectively. Assume that we have a* triple extension*:*

(i) $A_1 \subseteq A_2$, *and*
(ii) $\to_1 \subseteq \to_2$, *and*
(iii) $\mathrm{NF}_1 \subseteq \mathrm{NF}_2$.

Then $\mathrm{UN}(\mathcal{A}_2) \Longrightarrow \mathrm{UN}(\mathcal{A}_1)$.

Proof. Assume $\mathrm{UN}(\mathcal{A}_2)$, and consider a peak $s \leftarrow_1^* \cdot \to_1^* t$ in $\mathcal{A}_1$ with two normal forms $s, t \in \mathrm{NF}_1$. Then $s \leftarrow_2^* \cdot \to_2^* t$ by $\to_1 \subseteq \to_2$ and s, t are normal forms in $\mathcal{A}_2$ by $\mathrm{NF}_1 \subseteq \mathrm{NF}_2$. Hence, $s \equiv t$. □

We note that the Triple Extension Lemma is valid also for the conversion version of UN where it is required that convertible normal forms are syntactically equivalent.

To see that UN^{∞} for $\lambda^{\infty}\beta$ is a corollary of UN^{∞} for λ_{111} we choose: A_1 as the set of finite and infinite λ-terms with $\to_1 \;=\; \twoheadrightarrow\!\!\!\!\twoheadrightarrow_{\beta}$, and A_2 are the finite and infinite λ-terms over the extended signature with Ω the set of terms of λ_{111}, and $\to_2$ is $\twoheadrightarrow\!\!\!\!\twoheadrightarrow_{\beta\Omega}$ (that is, together with the usual Ω_{BeT}-rules for replacing root active terms with Ω). Note that the set of $\lambda^{\infty}\beta$-normal forms is a subset of the set of $\lambda^{\infty}\beta\Omega_{\mathsf{BeT}}$-normal forms. Not so for BT and LLT!

3.3. Clock semantics[2]

Böhm trees are invariant under β-reduction. This yields a simple method to discriminate (finite) λ-terms M, N: just compute $\mathsf{BT}(M)$ and $\mathsf{BT}(N)$, and if $\mathsf{BT}(M) \neq \mathsf{BT}(N)$, then $M \neq_{\beta} N$.[3] But what if we want to β-discriminate M, N when their BTs do coincide? In the following example this is actually the case.

In [47] Scott mentions the equation $\mathsf{BY} = \mathsf{BYS}$ (also discussed in [9]) as an interesting example of an equation not provable in $\lambda\beta$ (that is, it does not hold in finitary λ-calculus), while easily provable with Scott's Induction Rule. Here $\mathsf{B} = \lambda fgx.f(gx)$ and $\mathsf{S} = \lambda xyz.xz(yz)$ are the usual combinators, and Y is a fixed point combinator, that is, a term with $\mathsf{Y}f =_{\beta} f(\mathsf{Y}f)$. Scott mentions that he expects that using 'methods of Böhm' the non-convertibility in $\lambda\beta$ can be established, but that he did not attempt a proof. On the other hand, with the induction rule (of Scott) the equality is easily established. Indeed this equation holds in the infinitary λ-calculus $\lambda\beta^{\infty}$: a straightforward calculation shows that in $\lambda\beta^{\infty}$, we have $\mathsf{BY} = \mathsf{BYS} = \lambda ab.(ab)^{\omega}$. That the equation is not provable in $\lambda\beta$, is a nice short proof. Here we take for the fpc Y, Curry's fpc Y_0, (as in [47]), defined by $\mathsf{Y}_0 = \lambda f.\omega_f\omega_f$ where $\omega_f = \lambda x.f(xx)$.

Proposition 3.10. $\mathsf{BY}_0 \neq_{\beta} \mathsf{BY}_0\mathsf{S}$.

Proof. Postfixing the combinator $\mathsf{I} = \lambda x.x$ yields $\mathsf{BY}_0\mathsf{I}$ and $\mathsf{BY}_0\mathsf{SI}$. Now $\mathsf{BY}_0\mathsf{I} =_{\beta} \mathsf{Y}_0$ and $\mathsf{BY}_0\mathsf{SI} =_{\beta} \mathsf{Y}_0(\mathsf{SI}) = \mathsf{Y}_1$, where Y_1 is Turing's fpc, $\mathsf{Y}_1 = \mathsf{ZZ}$ with $\mathsf{Z} = \lambda xf.f(xxf)$. Because $\mathsf{Y}_0 \neq_{\beta} \mathsf{Y}_1$ (see, e.g., [20] for a proof), the result follows. In the same breath we can strengthen this non-equation to all fpcs Y, by the same calculation followed by an application of Intrigila's theorem [21] stating that for no fpc Y we have $\mathsf{Y} = \mathsf{Y}\delta = \mathsf{Y}(\mathsf{SI})$. $\square$

Here we could profit from some lucky coincidences. But how can we in more general circumstances β-discriminate M, N when their BTs do coincide? A clue is given by inspecting the BTs of the terms BY_0 and $\mathsf{BY}_0\mathsf{S}$, and in particular how they are computed, in what 'tempo'.

The idea is that we will extract from a λ-term more than just its BT, but also how the BT was formed; one could say, in what tempo, or in what rhythm. A BT is formed from static pieces of information, but these are rendered in a clock-wise fashion, where the ticks of the internal clock are head reduction steps. Thus we arrive at a refined notion of BT, where we annotate at the nodes the necessary ticks of the clock, i.e., the number of head reduction steps, needed to go from one position in the BT to a successor position. The equality thus arising is strictly intermediate between β-convertibility $=_{\beta}$, and Böhm tree equality $=_{\mathsf{BT}}$. The clocked Böhm trees of BY_0 and $\mathsf{BY}_0\mathsf{S}$ are displayed in Fig. 22.

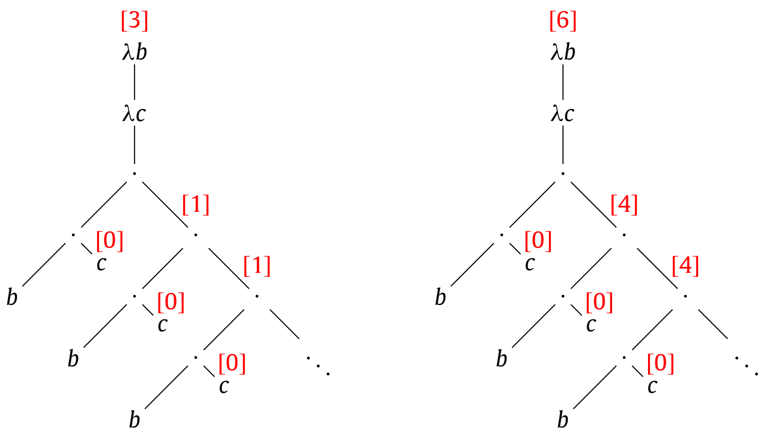


Fig. 22. Clocked Böhm trees of BY_0 and $\mathsf{BY}_0\mathsf{S}$.

2 This section is based on our work [20].

3 For another method to prove terms being not convertible, we refer to [10].

Definition 3.11 (*Simple Terms*)**.** A term M is *simple*, if in no reduction of M a redex is multiplied. So every redex $(\lambda x.A)B$ contracted in a reduct of M has the property that x occurs at most once in A, or B is in normal form. An equivalent and useful reformulation is that in reduction diagrams involving reducts of *M no splitting in elementary diagrams* occurs.

An example of a term that is not simple is $\mathsf{Y}_0\delta$ with $\delta = \lambda xy.y(xy)$; it reduces to $\omega_\delta\omega_\delta$ and this term may duplicate the redex in the second ω_δ. But the reduct $\mathsf{ZZ} = \mathsf{Y}_1$ of $\omega_\delta\omega_\delta$ is simple, and likewise all $\mathsf{ZZ}\delta^{\sim n}$. (Here we use the notation $AB^{\sim n}$, defined by $AB^{\sim 0} = A$ and $AB^{\sim n+1} = ABB^{\sim n}$.) This example illustrates that although sometimes the terms under consideration are not simple, with some luck they can be reduced to simple terms. Another example is $\mathsf{Y}_1(\mathsf{SS})\mathsf{SI}$ as in the example above. Due to the presence of the redex (SS) this term is not simple. But it can easily be made simple, by reducing SS to its normal form $\lambda yzz'.zz'(yzz')$. (But there are also terms that have no simple reduct, i.e., cannot be simplified in this sense.)

In order to discriminate λ-terms M and N, we are of course allowed to consider convertible terms $M' =_\beta M$ and $N' =_\beta N$, in particular simple reducts. For the latter, different clock behavior proves non-convertibility.

Theorem 3.12 (*[20]*)**.** *For simple terms, clocks are invariant under reduction.*

This theorem enables us to prove non-convertibility of λ-terms with simple reducts, by inspection of their clock behavior: if they have different clocks they are non-convertible.

3.4. $\lambda^\infty\beta\eta$-calculus

The preceding theory begs the question how it can be generalised from the infinitary $\lambda^\infty\beta$-calculus to the infinitary $\lambda^\infty\beta\eta$-calculus, which arises by adding the η-rule. That is, the rewrite rules of $\lambda^\infty\beta\eta$ are:

$$(\lambda x.M)N \to M[x{:=}N] \qquad (\beta)$$
$$\lambda x.Mx \to M \quad \text{if } x \text{ is not free in } M. \qquad (\eta)$$

Familiarity with the finite λ-calculus learns that the extension of $\lambda\beta$-calculus to $\lambda\beta\eta$-calculus preserves many desirable properties, the foremost being the Church–Rosser property (CR). Working with the $\lambda^\infty\beta$-calculus we do not have the infinitary CR-property, CR^∞, as we saw, but we do have its corollary, UN^∞. So it is natural to ask whether this property is preserved in the $\lambda^\infty\beta\eta$-calculus. However, this property breaks down dramatically. The essence of this breakdown is already clearly visible in the first-order framework, as we will now show, to form a stepping stone to the infinitary lambda calculus setting.

3.4.1. Failure of UN^∞ for weakly orthogonal iTRSs

While orthogonal TRSs enjoy the property UN^∞ (see [26,38]), UN^∞ breaks down for weakly orthogonal TRSs (see [17]). The following simple counterexample can be used: for the signature consisting of the unary symbols P and S, consider the rewrite rules $\mathsf{P}(\mathsf{S}(x)) \to x$ and $\mathsf{S}(\mathsf{P}(x)) \to x$. For convenience, we drop the brackets and consider the corresponding string rewriting system (SRS):

$$\mathsf{PS} \to \varepsilon \qquad \mathsf{SP} \to \varepsilon$$

where ε is the empty word. This system has two trivial critical pairs:

$$\mathsf{P} \leftarrow \underline{\mathsf{P}\overline{\mathsf{S}}}\overline{\mathsf{P}} \to \mathsf{P} \qquad \mathsf{S} \leftarrow \underline{\mathsf{S}\overline{\mathsf{P}}}\overline{\mathsf{S}} \to \mathsf{S},$$

and hence is weakly orthogonal.

Now consider the term ψ defined as follows:

$$\psi = \mathsf{P}\ \mathsf{SS}\ \mathsf{PPP}\ \mathsf{SSSS}\ \mathsf{PPPPP}\ \mathsf{SSSSSS}\ \ldots$$

that is, $\psi = \mathsf{P}^1\mathsf{S}^2\mathsf{P}^3\mathsf{S}^4\mathsf{P}^5\mathsf{S}^6\ldots$. If we only apply rule $\mathsf{PS} \to \varepsilon$ the P-blocks are absorbed by the larger S-blocks to their right (that is: $\mathsf{P}^n\mathsf{S}^{n+1} \to^* \varepsilon$), leaving the normal form S^ω. Likewise, applying only $\mathsf{SP} \to \varepsilon$ yields P^ω:

$$\mathsf{S}^\omega \twoheadleftarrow\!\!\leftarrow \psi \to\!\!\twoheadrightarrow \mathsf{P}^\omega.$$

Note that S^ω and P^ω are normal forms, the only infinite normal forms. It is not difficult to prove that $\psi \to\!\!\twoheadrightarrow w$ for every infinite PS-word w. In particular $\psi \to (\mathsf{PS})^\infty$ which has no normal form, it rewrites only to itself.

Given an infinite PS-word w we can plot in a graph the surplus number of S's of w when stepping through the word w from left to right, see e.g. Fig. 23. The graph is obtained by counting S for $+1$ and P for -1. For $w = (\mathsf{SP})^\omega$ the graph takes values, consecutively, $1, 0, 1, 0, \ldots$, for $w = \mathsf{S}^\omega$ it takes $1, 2, 3, \ldots$, and for $w = \mathsf{P}^\omega$ we have $-1, -2, -3, \ldots$. The graph of the counterexample ψ is displayed in Fig. 23.

It can be shown that if the graph of a word w:

(i) has no upper bound, then $w \to\!\!\twoheadrightarrow \mathsf{S}^\omega$,
(ii) has no lower bound, then $w \to\!\!\twoheadrightarrow \mathsf{P}^\omega$,
(iii) has no upper and lower bound, then $w \to\!\!\twoheadrightarrow v$ for any infinite PS-word v.

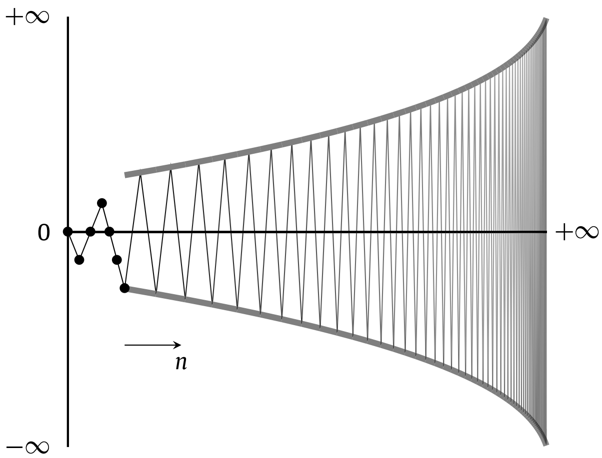


Fig. 23. Graph for the oscillating PS-word $\psi = \mathsf{P}^1\,\mathsf{S}^2\,\mathsf{P}^3\ \ldots$.

3.4.2. Failure of UN^∞ *for* $\lambda^\infty\beta\eta$*-calculus.*

Like $\mathsf{P}(\mathsf{S}(x)) \to x$ and $\mathsf{S}(\mathsf{P}(x)) \to x$, the $\lambda^\infty\beta\eta$-calculus [49,48] is a weakly orthogonal rewrite system. More precisely, the $\lambda^\infty\beta\eta$-calculus is a weakly orthogonal higher order rewrite system, see [58, Def. 11.6.10] and [35]. The $\lambda^\infty\beta\eta$-calculus allows for two critical pairs:

$$Mx \overset{\beta}{\leftarrow} (\lambda x.Mx)x \overset{\eta}{\to} Mx \qquad \lambda x.M[y{:=}x] \overset{\beta}{\leftarrow} \lambda x.(\lambda y.M)x \overset{\eta}{\to} \lambda y.M.$$

The terms $\lambda x.M[y{:=}x]$ and $\lambda y.M$ are equal modulo renaming of bound variables. Hence both critical pairs are trivial and $\lambda^\infty\beta\eta$ is weakly orthogonal.

It turns out that the counterexample $\psi = \mathsf{P}^1\mathsf{S}^2\mathsf{P}^3\mathsf{S}^4\mathsf{P}^5\mathsf{S}^6\ldots$ from the previous section has a direct translation to $\lambda^\infty\beta\eta$, see [17]. This translation can be made formally precise as follows:

Definition 3.13. We define $(\!|\_|\!) : \{\mathsf{P},\mathsf{S}\}^\omega \to \mathit{Ter}^\infty(\lambda)$ by $(\!|w|\!) = (\!|w|\!)_0$, for all $w \in \{\mathsf{P},\mathsf{S}\}^\omega$, where $(\!|w|\!)_i$ is defined coinductively, for all $i \in \mathbb{Z}$, as follows:

$$(\!|\mathsf{P}w|\!)_i = (\!|w|\!)_{i-1}\, x_i \qquad (\!|\mathsf{S}w|\!)_i = \lambda x_{i+1}.(\!|w|\!)_{i+1}.$$

The translation of ψ is the λ-term $(\!|\psi|\!)$, displayed in the middle of Fig. 24. This term has two normal forms (corresponding to S^ω and P^ω), as indicated in the figure. In [49] positive CR^∞ results are mentioned for Böhm Tree normal forms in $\lambda^\infty\beta\eta$-calculus.

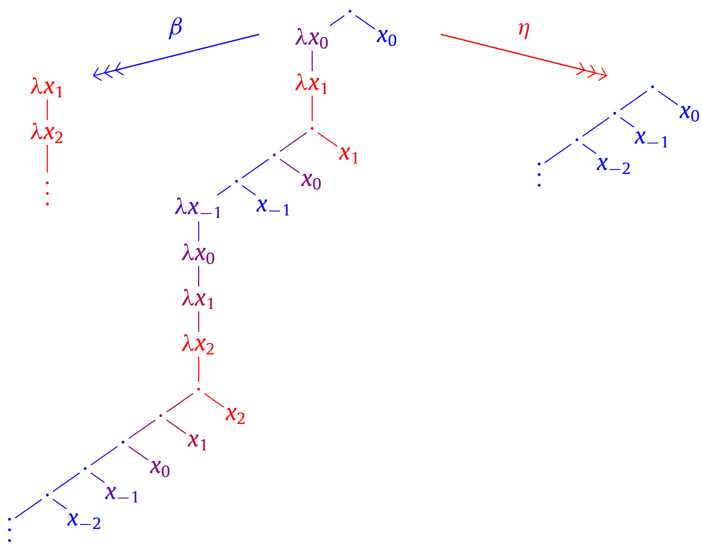


Fig. 24. Counterexample to UN^∞ in $\lambda^\infty\beta\eta$.

References

[1] S. Abramsky, C.-H.L. Ong, Full abstraction in the lazy lambda calculus, Information and Computation 105 (2) (1993) 159–267.
[2] Z.M. Ariola, R. Kennaway, J.W. Klop, M.R. Sleep, F.-J. de Vries, Syntactic definitions of undefined: on defining the undefined, in: Proc. Conf. on Theoretical Aspects of Computer Software (TACS 1994), 1994, pp. 543–554.
[3] H.P. Barendregt, The Lambda Calculus, its Syntax and Semantics, revised ed., in: Studies in Logic and The Foundations of Mathematics, vol. 103, North-Holland, 1984.
[4] H.P. Barendregt, R. Kennaway, J.W. Klop, M.R. Sleep, Needed reduction and spine strategies for the lambda calculus, Information and Computation 75 (3) (1987) 191–231.
[5] A. Berarducci, Infinite λ-calculus and non-sensible models, in: Logic and Algebra (Pontignano, 1994), Dekker, New York, 1996, pp. 339–377.
[6] C. Böhm (Ed.), Proc. Symp. on Lambda-Calculus and Computer Science Theory, in: Lecture Notes in Computer Science, vol. 37, Springer, 1975.
[7] R. Dedekind, Was Sind und Was Sollen die Zahlen? Friedrich Vieweg und Sohn, 1893.
[8] N. Dershowitz, S. Kaplan, D.A. Plaisted, Rewrite, rewrite, rewrite, rewrite, rewrite, Theoretical Computer Science 83 (1) (1991) 71–96.
[9] M. Dezani-Ciancaglini, P. Severi, F.-J. de Vries, Infinitary lambda calculus and discrimination of Berarducci trees, Theoretical Computer Science 2 (298) (2003) 275–302.
[10] J. Endrullis, R. de Vrijer, Reduction under substitution, in: Proc. 19th Int. Conf. on Rewriting Techniques and Applications (RTA 2008), in: Lecture Notes in Computer Science, vol. 5117, Springer, 2008, pp. 425–440.
[11] J. Endrullis, H. Geuvers, J.G. Simonsen, H. Zantema, Levels of undecidability in rewriting, Information and Computation 209 (2) (2011) 227–245.
[12] J. Endrullis, H. Geuvers, H. Zantema, Degrees of undecidability in term rewriting, in: Proc. Int. Workshop on Computer Science Logic (CSL 2009), in: LNCS, vol. 5771, Springer, 2009, pp. 255–270.
[13] J. Endrullis, C. Grabmayer, D. Hendriks, Data-oblivious stream productivity, in: Proc. Conf. on Logic for Programming Artificial Intelligence and Reasoning (LPAR 2008), in: LNCS, vol. 5330, Springer, 2008, pp. 79–96.
[14] J. Endrullis, C. Grabmayer, D. Hendriks, Complexity of Fractran and productivity, in: Proc. Conf. on Automated Deduction (CADE 22), in: LNCS, vol. 5663, 2009, pp. 371–387.
[15] J. Endrullis, C. Grabmayer, D. Hendriks, A. Isihara, J.W. Klop, Productivity of stream definitions, Theoretical Computer Science 411 (2010) 765–782.
[16] J. Endrullis, C. Grabmayer, D. Hendriks, J.W. Klop, R.C. de Vrijer, Proving infinitary normalization, in: Postproc. Int. Workshop on Types for Proofs and Programs (TYPES 2008), in: Lecture Notes in Computer Science, vol. 5497, Springer, 2009, pp. 64–82.
[17] J. Endrullis, C. Grabmayer, D. Hendriks, J.W. Klop, V. van Oostrom, Unique normal forms in infinitary weakly orthogonal rewriting, in: Proc. 21st Int. Conf. on Rewriting Techniques and Applications (RTA 2010), in: Leibniz International Proceedings in Informatics, vol. 6, Schloss Dagstuhl, 2010, pp. 85–102.
[18] J. Endrullis, C. Grabmayer, R.C. de Vrijer, Equivalence of two versions of SN^∞ (w-SN^∞ and s-SN^∞), 2008.
[19] J. Endrullis, D. Hendriks, Lazy productivity via termination, Theoretical Computer Science 412 (28) (2011) 3203–3225.
[20] J. Endrullis, D. Hendriks, J.W. Klop, Modular construction of fixed point combinators and clocked Böhm trees, in: Proc. Symp. on Logic in Computer Science (LICS 2010), IEEE Computer Society, 2010, pp. 111–119.
[21] B. Intrigila, Non-existent statman's double fixed point combinator does not exist, indeed, Information and Computation 137 (1) (1997) 35–40.
[22] S. Kahrs, Infinitary rewriting: foundations revisited, in: Proc. 21st Int. Conf. on Rewriting Techniques and Applications (RTA 2010), in: Leibniz International Proceedings in Informatics, vol. 6, Schloss Dagstuhl, 2010, pp. 161–176.
[23] J.R. Kennaway, J.W. Klop, M.R. Sleep, F.-J. de Vries, An infinitary Church–Rosser property for non-collapsing orthogonal term rewriting systems, Technical Report CS-R9043, CWI, 1990.
[24] J.R. Kennaway, J.W. Klop, M.R. Sleep, F.-J. de Vries, Transfinite reductions in orthogonal term rewriting systems, in: Proc. 4th Int. Conf. on Rewriting Techniques and Applications (RTA 1991), in: Lecture Notes in Computer Science, vol. 488, Springer, 1991, pp. 1–12.
[25] J.R. Kennaway, J.W. Klop, M.R. Sleep, F.-J. de Vries, The adequacy of term graph rewriting for simulating term rewriting, ACM TOPLAS 16 (1994) 493–523 (An earlier version appeared as Chapter 12 of [58]).
[26] J.R. Kennaway, J.W. Klop, M.R. Sleep, F.-J. de Vries, Transfinite reductions in orthogonal term rewriting systems, Information and Computation 119 (1) (1995) 18–38.
[27] J.R. Kennaway, J.W. Klop, M.R. Sleep, F.-J. de Vries, Infinitary lambda calculus, Theoretical Computer Science 175 (1) (1997) 93–125.
[28] J.R. Kennaway, V. van Oostrom, F.-J. de Vries, Meaningless terms in rewriting, The Journal of Functional and Logic Programming 1 (1999).
[29] J. Ketema, On normalisation of infinitary combinatory reduction systems, in: Proc. 19th Int. Conf. on Rewriting Techniques and Applications (RTA 2008), in: Lecture Notes in Computer Science, vol. 5117, Springer, 2008, pp. 172–186.
[30] J. Ketema, Comparing Böhm-like trees, in: Proc. 20th Int. Conf. on Rewriting Techniques and Applications (RTA 2009), in: Lecture Notes in Computer Science, vol. 5595, Springer, 2009, pp. 239–254.
[31] J. Ketema, Reinterpreting compression in infinitary rewriting, in: Proc. 23rd Int. Conf. on Rewriting Techniques and Applications (RTA 2012), in: LIPIcs, vol. 15, Schloss Dagstuhl, 2012, pp. 209–224.
[32] J. Ketema, J.G. Simonsen, Infinitary combinatory reduction systems, in: Proc. 16th Int. Conf. on Rewriting Techniques and Applications (RTA 2005), in: Lecture Notes in Computer Science, vol. 3467, Springer, 2005, pp. 438–452.
[33] J. Ketema, J.G. Simonsen, Infinitary combinatory reduction systems: confluence, Logical Methods in Computer Science 5 (4) (2009) 1–29.
[34] J. Ketema, J.G. Simonsen, Infinitary combinatory reduction systems: normalising reduction strategies, Logical Methods in Computer Science 6 (1:7) (2010) 1–35.
[35] J. Ketema, J.G. Simonsen, Infinitary combinatory reduction systems, Information and Computation 209 (6) (2011) 893–926.
[36] J.W. Klop, Combinatory Reduction Systems, in: Mathematical Centre Tracts, vol. 127, Mathematisch Centrum, 1980.
[37] J.W. Klop, Term rewriting systems, in: Handbook of Logic in Computer Science, vol. II, Oxford University Press, 1992, pp. 1–116.
[38] J.W. Klop, R.C. de Vrijer, Infinitary normalization, in: We Will Show Them: Essays in Honour of Dov Gabbay (2), College Publications, 2005, pp. 169–192.
[39] J.W. Klop, V. van Oostrom, F. van Raamsdonk, Reduction strategies and acyclicity, in: Rewriting, Computation and Proof, Essays Dedicated to Jean-Pierre Jouannaud on the Occasion of his 60th Birthday, in: Lecture Notes in Computer Science, vol. 4600, 2007, pp. 89–112.
[40] A. Kurz, D. Petrisan, P. Severi, F.-J. de Vries, An alpha-corecursion principle for the infinitary lambda calculus, in: Proc. 11th Int. Workshop on Coalgebraic Methods in Computer Science, Springer, 2012.
[41] B. Lercher, Lambda-calculus terms that reduce to themselves, Notre Dame Journal of Formal Logic 17 (2) (1976) 291–292.
[42] J.-J. Lévy, An algebraic interpretation of the $\lambda\beta K$-calculus, and an application of a labelled λ-calculus, Theoretical Computer Science 2 (1) (1976) 97–114.
[43] G. Longo, Set-theoretical models of λ-calculus: theories, expansions, isomorphisms, Annals of Pure and Applied Logic 24 (2) (1983) 153–188.
[44] A. Middeldorp, Call by need computations to root-stable form, in: Proc. Symp. on Principles of Programming Languages (POPL 1997), ACM, 1997, pp. 94–105.
[45] C.-H.L. Ong, The lazy lambda calculus: an investigation into the foundations of functional programming, Ph.D. Thesis, University of Cambridge, 1992.
[46] A. Polonsky, J. Endrullis, A characterization of reduction 1-cycles in infinitary lambda calculus, 2012, submitted for publication.
[47] D.S. Scott, Some philosophical issues concerning theories of combinators, in: Lambda Calculus and Computer Science Theory, in: Lecture Notes in Computer Science, vol. 37, 1975, pp. 346–366.
[48] P. Severi, F.-J. de Vries, A lambda calculus for D_∞, Technical Report, University of Leicester, 2002.
[49] P. Severi, F.-J. de Vries, An extensional Böhm model, in: Rewriting Techniques and Applications, in: LNCS, vol. 2378, Springer, 2002, pp. 159–173.
[50] P. Severi, F.-J. de Vries, Order structures for Böhm-like models, in: Computer Science Logic, in: LNCS, vol. 3634, Springer, 2005.

[51] P. Severi, F.-J. de Vries, Decomposing the lattice of meaningless sets in the infinitary lambda calculus, in: Proc. of 18th Int. Workshop on Logic, Language, Information and Computation (WoLLIC 2011), in: Lecture Notes in Computer Science, vol. 6642, Springer, 2011, pp. 210–227.
[52] P. Severi, F.-J. de Vries, Weakening the axiom of overlap in infinitary lambda calculus, in: Proc. 22nd Int. Conf. on Rewriting Techniques and Applications (RTA 2011), in: Leibniz International Proceedings in Informatics, vol. 10, Schloss Dagstuhl, 2011, pp. 313–328.
[53] P. Severi, F.-J. de Vries, Meaningless sets in infinitary combinatory logic, in: Proc. 23rd Int. Conf. on Rewriting Techniques and Applications (RTA 2012), in: LIPics, vol. 15, Schloss Dagstuhl, 2012, pp. 288–304.
[54] J.G. Simonsen, On confluence and residuals in cauchy convergent transfinite rewriting, Information Processing Letters 91 (3) (2004) 141–146.
[55] J.G. Simonsen, Weak convergence and uniform normalization in infinitary rewriting, in: Proc. 20th Int. Conf. on Rewriting Techniques and Applications (RTA 2009), in: Leibniz International Proceedings in Informatics, vol. 6, Schloss Dagstuhl, 2010, pp. 311–324.
[56] M.R. Sleep, M.J. Plasmeijer, M.C.J.D. van Eekelen (Eds.), Term Graph Rewriting: Theory and Practice, John Wiley, 1993.
[57] R. Smullyan, To Mock a Mockingbird, and Other Logic Puzzles, Alfred A. Knopf, New York, 1985.
[58] Terese, Term Rewriting Systems, in: Cambridge Tracts in Theoretical Computer Science, vol. 55, Cambridge University Press, 2003.
[59] F.-J. de Vries, Böhm trees, bisimulations and observations in lambda calculus, in: Proc. 2nd Fuji Int. Workshop on Functional and Logic Programming, World Scientific, 1997, pp. 230–245.
[60] R.C. de Vrijer, Conditional linearization, Indagationes Mathematicae 10 (1) (1999) 145–159.
[61] J. Waldmann, The combinator S, Information and Computation 159 (1–2) (2000) 2–21.
[62] H. Zantema, Normalization of infinite terms, in: Proc. 19th Int. Conf. on Rewriting Techniques and Applications (RTA 2008), in: Lecture Notes in Computer Science, vol. 5117, 2008, pp. 441–455.
[63] H. Zantema, M. Raffelsieper, Stream productivity by outermost termination, in: Proc. Workshop on Reduction Strategies in Rewriting and Programming (WRS 2009), in: Electronic Proceedings in Theoretical Computer Science, vol. 15, 2009, pp. 83–95.
[64] H. Zantema, M. Raffelsieper, Proving productivity in infinite data structures, in: Proc. 21st Int. Conf. on Rewriting Techniques and Applications (RTA 2010), in: Leibniz International Proceedings in Informatics, vol. 6, Schloss Dagstuhl, 2010, pp. 401–416.