Applications of Infinitary Lambda Calculus

Henk Barendregt* & Jan Willem Klop†

Dedicated to Giuseppe Longo at the occasion of his 60-th birthday

Abstract

We present an introduction to infinitary lambda calculus, highlighting its main properties. Subsequently we give three applications of infinitary lambda calculus. The first addresses the non-definability of Surjective Pairing, which in Barendregt [1984] is shown to be not definable in lambda calculus. We show how this result follows easily as an application of Berry's Sequentiality Theorem, which itself can be proved in the setting of infinitary lambda calculus. The second pertains to the notion of relative recursiveness of number-theoretic functions. The third application concerns an explanation of counterexamples to confluence of lambda calculus extended with non-left-linear reduction rules: Adding non-left-linear reduction rules such as $\delta xx \to x$ or the reduction rules for Surjective Pairing to the lambda calculus yields non-confluence, as proved in Klop [1980]. We discuss how an extension to the infinitary lambda calculus, where Böhm trees can be directly manipulated as infinite terms, yields a more simple and intuitive explanation of the correctness of these Church-Rosser counterexamples.

1. Introduction

The aim of this paper is to present some well-known results in λ-calculus from the point of view of infinitary λ-calculus, where terms may be infinitely deep and reduction sequences may be of transfinite length α, for a countable ordinal α. Infinitary λ-terms are already familiar in λ-calculus in the form of Böhm trees (BTs), but in the extended setting of infinitary λ-calculus (or λ^∞ for short) BTs are just a particular kind of infinite normal forms, and in this extended setting we can even apply a BT to another BT. In Section 2 we will give a somewhat more detailed exposition of λ^∞ with β-reduction, $\lambda^\infty\boldsymbol{\beta}$ for short. (We will not consider η-reduction in this paper.) First we will describe why in our view infinitary λ-calculus is of interest.

The first reason pertains to *semantics* of λ-calculus. By now it is classic that infinite λ-terms constitute a syntactic approach to the semantics of finite λ-terms with (e.g.) β-reduction, in various forms, in particular the semantics given by the three families of infinite λ-trees known as Böhm trees, Lévy-Longo trees, and Berarducci trees. Whereas the first family seems to be the most important, the second family is instrumental for a closer connection to the practice of functional programming using notions as lazy reduction and weak head normal form, see Abramsky and Ong [1993], while the third family is a sophisticated tool for consistency studies as demonstrated in Berarducci and Intrigila [1996].

The second reason concerns the *pragmatics* of computing with λ-terms. Some computations are most naturally presented as transfinite sequences, rather than as compressed sequences of length at most ordinal ω, even though this always can be done by dove-tailing. Below we give some illustrating examples.

*Radboud University, Nijmegen, The Netherlands.
†Radboud University, Nijmegen, and VU University, Amsterdam, The Netherlands.

The third reason is found in the feature of *expressivity*. Infinite λ-terms can be nonrecursive. This can be used to give a direct representation of notions that otherwise need some circumlocution for their definition: a recursion-theoretic oracle, used in the definition of relative computability, can be defined in various ways, but the representation as an infinite λ-term has an appealing directness, since the oracle can now directly be processed by a finite λ-term, standing for a finite program. Below, in section 4, we will substantiate this.

The last reason, illustrated by Section 3 on Berry's Sequentiality Theorem (BST) and Section 5 on the failure of confluence in extensions of λ-calculus with non-left linear reduction rules, is theoretical *coherence and transparency*, including a better understanding of phenomena in finite (!) λ-calculus. The Section on BST provides such a better understanding for the inherent sequentiality of finitary λ-calculus, with as corollaries some non-definability results treated there, among them the fundamental fact that (just like parallel-or), it is not possible to define Surjective Pairing in λ-calculus. We present a succinct and new proof of this non-definability fact. Finally, Section 5 contributes to a better understanding of the extension of λ-calculus with rules like $\delta xx \to x$, encoding a discriminator δ for syntactic equality (of its two arguments); such an extension $\lambda + \delta$ looses the confluence property, but the deeper reason is best understood via an excursion to the realm of infinite λ-terms.

Concluding this Introduction, let us point out once more that our paper has in part the character of a survey and introduction, albeit of modest scope. This entails that our primary concern is not to communicate new results on this subject. Yet there are some new elements. Next to some new proofs, such as for the undefinability of Surjective Pairing in (finitary and now also in infinitary) λ-calculus, and for the non-confluence of this same system viewed as a rewrite system, there are a few new results, notably the short solution of an open problem of Scott [1975a], and a theorem building on work of Kleene [1963], capturing the notion of relative recursiveness directly in (infinitary) λ-calculus.

2. Preliminaries

In this section we will lay out various notions and notations, and some basic properties, of finitary as well as infinitary λ-calculus.

2.1. Lambda calculus and two extensions

We assume familiarity with ordinary untyped λ-calculus, see e.g. Barendregt [1984]. In particular the following notations will be used. The notation follows common practise. Closed λ-terms are usually denoted by Roman capitals, but sometimes by Greek letters (upper or lower case). As often in mathematics and programming languages, there are sometimes innocent examples of overloading: for example ω is a λ-term, but also the first infinite ordinal, in which sense it is used in the notation M^ω, an infinite λ-term.

2.1. NOTATION. $M \equiv N$ stands for syntactic equality between the (possibly infinitary) terms M, N and $M = N$ for their convertibility (w.r.t. a notion of reduction clear from the context, usually β or an extension). We use the combinators (closed λ-terms) $\mathsf{I} \equiv \lambda x.x$, $\mathsf{K} \equiv \lambda xy.x$, $\mathsf{S} \equiv \lambda xyz.xz(yz)$, $\mathsf{Y} \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$, $\mathsf{B} \equiv \lambda xyz.x(yz)$, $\Theta \equiv (\lambda xy.y(xxy))(\lambda xy.y(xxy))$. We also often use the combinators $\omega \equiv (\lambda x.xx)$, in some papers denoted by Δ, and $\Omega \equiv (\omega\omega)$.

The set of λ-terms is denoted by Λ, that of normal forms (under β-reduction) by Λ_{NF}. The set of closed λ-terms is denoted by $\Lambda^\emptyset$. For $M, N \in \Lambda$ the following notations are used. For pairing $[M, N] \equiv \lambda z.zMN$, with z a fresh variable; for

applicative iteration M^nN is defined recursively: $M^0N \equiv N$; $M^{k+1}N \equiv M(M^kN)$. Using this notation, the Church numerals are $\mathbf{c}_n \equiv \lambda fx.f^nx$. For iterated arguments $MN^{\sim n}$ is also defined recursively: $MN^{\sim 0} \equiv M$; $MN^{\sim(k+1)} \equiv MN^{\sim k}N$.

2.2. DEFINITION. (i) Extend the set of λ-terms Λ with a constant f, intended to represent an f:$\mathbb{N}\to\mathbb{N}$. The resulting set of terms will be denoted by $\Lambda(\mathsf{f})$.

(ii) On $\Lambda(\mathsf{f})$ one can extend $\boldsymbol{\beta}$-reduction with the notion of reduction f axiomatized by the contraction rule: $\mathsf{f}\mathbf{c}_n \to_{\mathsf{f}} \mathbf{c}_{f(n)}$.

2.3. LEMMA. *The notions of reduction* f *and* $\boldsymbol{\beta}\mathsf{f}$ *are Church-Rosser.*

PROOF. Similar to the proof of Mitschke's Theorem 15.3.3 in Barendregt [1984]. Alternatively, observe that f and $\boldsymbol{\beta}\mathsf{f}$ constitute orthogonal higher-order rewriting systems (in the form of CRSs or HRSs) and use Theorem 11.6.19 in Terese [2003]. ■

Remember that every λ-term M is of one of the following forms:

$$M \equiv \lambda x_1 \dots x_n.yM_1 \dots M_m \text{ or } \lambda x_1 \dots x_n.(\lambda y.P)QM_1 \dots M_k.$$

In the first case M is said to be a *head normal form (hnf)*; in the second case M has the head-redex $(\lambda y.P)Q$.

2.4. DEFINITION. (i) Another extension with one constant is $\Lambda(\bot)$.

(ii) On $\Lambda(\bot)$ one defines the notion of reduction $\boldsymbol{\Omega}$ by the contraction rules:

$$\begin{aligned} M &\to_{\boldsymbol{\Omega}} \bot, \qquad \text{if } M \not\equiv \bot \text{ and does not } \beta\text{-reduce to a hnf;} \\ \bot M &\to_{\boldsymbol{\Omega}} \bot; \\ \lambda x.\bot &\to_{\boldsymbol{\Omega}} \bot. \end{aligned}$$

2.5. LEMMA. *The notion of reduction* $\boldsymbol{\beta\Omega}$ *is Church-Rosser.*

PROOF. See Barendregt [1984] Lemma 15.2.5. ■

Below we will use Definition 2.4 and Lemma 2.5 in our dealings with Böhm Trees (BTs). We mention also at this point two notions related to hnf's, to be used below for two variants of BTs, to wit the Lévy-Longo Trees (LLTs) and the Berarducci Trees (BeTs). For the moment, the next Definition 2.6 and Remark 2.7 can be skipped.

2.6. DEFINITION. (i) A term M is a *weak head normal form* (weak hnf or whnf) if it is an *abstraction* $\lambda x.P$ or *vector* $xM_1 \dots M_m$, where x is a variable.

(ii) A λ-term M is *root stable*, if it is a variable, an abstraction $\lambda x.P$, or an application PQ where P does not reduce to an abstraction. Equivalently: M is root stable if it has no infinite reduction in which infinitely often a root reduction step is performed. A β-reduction step $C[(\lambda x.A(x))B] \to A(B)$ is a root step when the context $C[\,]$ is empty, so the contracted redex is 'at the root'.

2.7. REMARK. So, in a sense, whnf's as 'semantics building blocks' are parts of the hnf building blocks. This is not a coincidence, but is connected to the relationship between the various notions of semantics of λ-terms, regarding BTs, LLTs and BeTs that we briefly mentioned above, and on which we will elaborate below. The BeT building blocks are just abstractors λx, application nodes, and variables; in turn these building blocks are fragments of the whnf building blocks. The refinement of the 'bases of building blocks' can be seen as reflecting the coarseness of the corresponding semantical notions, which is stated more precisely in Remark 3.6.

2.2. Infinite λ-terms

In this section we will introduce infinite λ-terms. We first present the general notational format, called *applicative notation*, and then a specialized notation for a subset of the infinite λ-terms, where an abbreviated notation is more convenient, called the *hnf notation*.

2.8. DEFINITION. (i) Λ^∞ is the set of (possibly) infinite λ-terms coinductively defined by

$$\texttt{term ::= x} \mid \texttt{term @ term} \mid \lambda x\ \texttt{term}$$

(ii) $\Lambda^\infty(\bot)$ is defined similarly, also allowing the constant $\bot$.

(iii) Certain elements of $\Lambda^\infty(\bot)$ are known as Böhm trees of finite λ-terms $M \in \Lambda$, defined in Barendregt [1984] by the following coinductive definition.

$$\mathrm{BT}(M) \equiv \begin{cases} \bot, & \text{if } M \text{ does not have a hnf;} \\ \equiv \lambda\vec{x}.\ y \ \text{ (with children } \mathrm{BT}(M_1), \cdots, \mathrm{BT}(M_k)), & \text{if } M \text{ has hnf } \lambda\vec{x}.y\vec{M} \\ & \text{with } \vec{M} = M_1, \ldots, M_k. \end{cases}$$

So BT is a map from Λ to $\Lambda^\infty(\bot)$. Below we extend this map to all of $\Lambda^\infty(\bot)$, but this requires the definitions of infinitary β-reduction and hnf on $\Lambda^\infty(\bot)$.

Often we will present (both finite and infinite) λ-terms as unary-binary branching trees, with application nodes binary branching and abstraction nodes λx unary branching, and with variables or constants as terminal nodes. Such trees are displayed in Figure 1 (left window) and Figure 2 (left window).

2.9. REMARK. Note that in this last definition we have introduced an abbreviated notational format, introduced in Barendregt [1984], that we will call the hnf notation, which is especially suitable for terms that do not contain redexes. The BTs are among such terms. In Figures 1,2 it is shown how this hnf-notation can be 'expanded' to the general applicative notation, which costs several more application and abstraction nodes.

2.10. EXAMPLE. (i) Let $M \equiv [a_1, a_2, [a_3]] \equiv \lambda z.za_1a_2(\lambda z.za_3)$. Then M has the following two views.

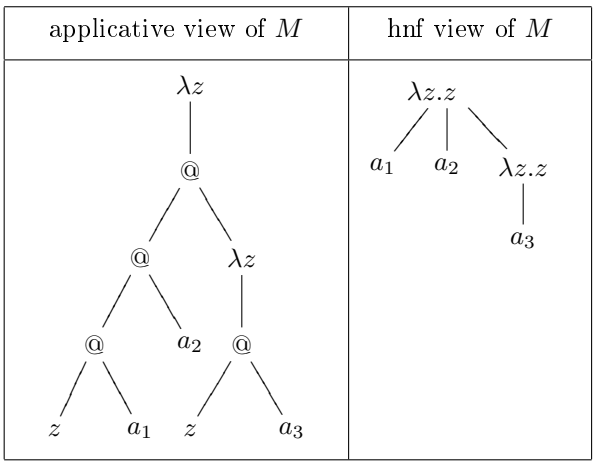


Figure 1: Two views of $M \equiv [a_1, a_2, [a_3]] \equiv \lambda z.za_1a_2(\lambda z.za_3)$

(ii) Let $\mathsf{Y}_2 \equiv \lambda f.f(f\bot)$ and Y as in 2.1. Then $\mathsf{Y}_2, \mathrm{BT}(\mathsf{Y})$ have the following two views.

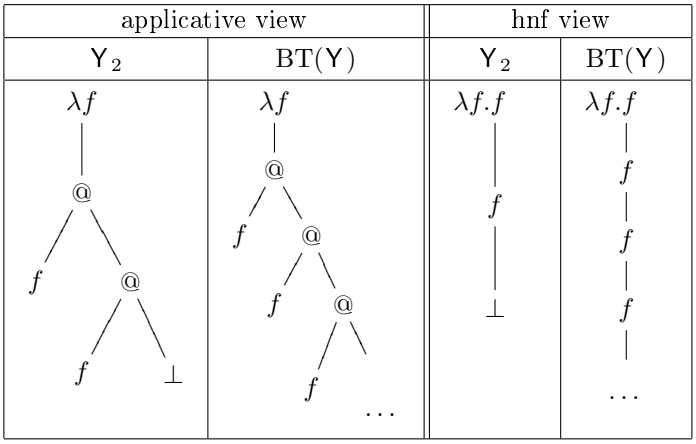
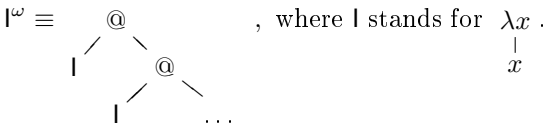


Figure 2: Two views of $\mathrm{BT}(\mathsf{Y})$ and its approximant Y_2

(iii) A notation that we will sometimes use for $M \in \Lambda^\infty(\bot)$, is M^ω, defined coinductively by $M^\omega \equiv M(M^\omega)$. For instance $\mathrm{BT}(\mathsf{Y}) \equiv \lambda f.f^\omega$.

(iv) An interesting term is I^ω. It will play a role in Lemma 2.20. In applicative notation one has

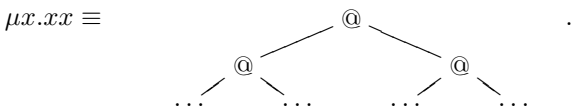


Note that this term contains infinitely many β-redexes; as we will see later, it reduces in one step to itself. There is no hnf view of I^ω.

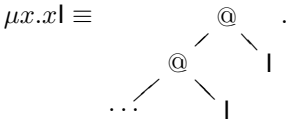
(v) We generalize (ii), especially for use in Section 5, to the well-known μ-notation; in this notation we have $M^\omega \equiv \mu x.Mx$, with x a fresh variable (i.e. $\notin \mathrm{FV}(M)$). This in accordance with the well-known μ-rule

$$\mu x.M \to_\mu M[x := \mu x.M].$$

Note that $\mu x.M$ can be emulated as $\Theta(\lambda x.M)$. So $\mathsf{A} \equiv \mu x.xx \in \Lambda^\infty(\bot)$ is the binary tree consisting of application nodes only.



Moreover, one has



The following remark needs Definition 2.29 and can be skipped at first reading.

2.11. Remark. Whether a term such as $\mu x.xx$ is useless (i.e. its 'semantics' equals $\bot$) depends from the semantical view that one is adopting. More precisely: let $M \in \Lambda$ be such that $M \twoheadrightarrow_\beta MM$. To this end, take $M \equiv \mathsf{Y}\omega$, where $\omega \equiv \lambda x.xx$. It

is an easy exercise to show that M has no hnf, and thus $\mathrm{BT}(M) \equiv \bot$. We could also take the BT after reducing M to its infinite normal form in $\Lambda^\infty(\bot)$; as we will see later, this infinite normal form of M is $\mu x.xx$. Now, residing in $\Lambda^\infty(\bot)$, we again have $\mathrm{BT}(\mu x.xx) \equiv \bot$, for the extension of BT to $\Lambda^\infty(\bot)$ to be defined below. This is so because $\mu x.xx$ is a normal form, which is not a hnf, hence has no hnf.

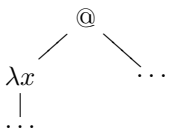
Also in the semantics of Lévy-Longo trees (LLTs), this term and its infinite normal form $\mu x.xx$, both have $\mathrm{LLT} \equiv \bot$.

However, in the Berarducci tree semantics, which gives a syntactic model of λ-calculus, these terms do have a non-trivial semantical value, viz. $\mu x.xx$, see Example 2.37.

In this paper we will focus on the coarsest of the three semantical views, namely that of the BTs. See also Remark 3.6.

2.3. Beta reduction on $\Lambda^\infty(\bot)$

The notion of β-reduction extends in a straightforward manner from $\Lambda(\bot)$ to $\Lambda^\infty(\bot)$, bearing in mind that a β-redex has a finite 'redex-pattern' that makes it recognizable as such, namely



Of course one has to define the usual notions of free and bound variable occurrences, and substitution without variable capture. But it is a matter of routine to spell out these details, from which we will refrain here; instead we refer to a detailed treatment in Terese [2003], section 12.4, where also α-conversion is treated, using Barendregt's variable convention, and including a proof of the Substitution Lemma as in Barendregt [1984], 2.1.16. Important is to realize that the contraction of a β-redex $(\lambda x.M)N$ to the reduct or contractum $M[x := N]$ now may require infinitely many copies of N to be substituted in as many occurrences of the free variable x in M. Examples are below in Example 1.3.1 and 1.3.2. As pointed out in Terese [2003], in practice one will avoid such 'ω-tasks', by adopting some computational scheme like explicit substitution, allowing a finite part of the reduct to be computed in finite time. Having defined single β-reduction steps on $\Lambda^\infty(\bot)$, with notation $\to_\beta$, we define the transitive-reflexive closure of $\to_\beta$, written as $\twoheadrightarrow_\beta$, just as for finite λ-terms, but now for possibly infinite terms, that is on $\Lambda^\infty(\bot)$. With this notion of reduction, the definition of head normal form (hnf) and thereby the coinductive definition of BT extends in an analogous way to all of the domain $\Lambda^\infty(\bot)$; we will not repeat the definitions as they are verbatim the same.

The definition of *normal form* with respect to β-reduction (β-nf) is simple: $M \in \Lambda^\infty(\bot)$ is a β-normal form if it contains no β-redex. As an advance warning, elaborated below after Lemma 2.30, we mention that every BT is a β-normal form, but not vice versa.

Next we introduce infinite β-reduction sequences. We will do this in an informal way, referring for a full detailed treatment to Terese [2003], Kennaway et al. [1995a], [1995b] and [1997], Ketema and Simonsen [2005] and [2006], Klop and de Vrijer [2005]. Reduction sequences now may have transfinite length:

$$M_0 \to_\beta M_1 \to_\beta \ldots M_\omega \to_\beta M_{\omega+1} \to_\beta \ldots M_{\omega.2} \to_\beta \ldots M_\alpha.$$

Here $M_0, M_1, \ldots \in \Lambda^\infty(\bot)$. We have single β-steps $M_\gamma \to_\beta M_{\gamma+1}$. The term M_λ is for a limit ordinal λ the Cauchy limit of the earlier M_μ, with $\mu < \lambda$, with the

usual distance metric d on the finite and infinite term trees: $d(M,N) = 2^{-n}$ if M, N coincide only up to depth n, and $d(M,M) = 0$.

At this point in our introduction, we would have reduction sequences of every ordinal length α, e.g. for $M_0 \equiv \Omega$ we would have

$$M_0 \equiv \Omega \to_\beta \Omega \to_\beta \ldots\ M_\omega \equiv \Omega \to_\beta \Omega \to_\beta \ldots\ \Omega \equiv M_\alpha.$$

However, in addition to Cauchy convergence we impose a crucial further requirement on the limit behaviour of reduction sequences: when approaching a limit λ, the depth d_γ of the contracted redex r_γ in step $M_\gamma \to_\beta M_{\gamma+1}$ must tend to infinity: $\lim_{\gamma<\lambda} d_\gamma = \infty$. Here the depth of a redex r in $M \in \Lambda^\infty(\bot)$ is the number of steps (edges) in the term tree of M from the root to r. Now our reduction sequence in spe $\Omega \to \Omega \to \ldots \Omega$ of arbitrary length α is not allowed, since there the contracted redex depth stays at level 0, and is not going down at each limit λ; the action is 'stagnating' at level 0. Reduction sequences satisfying our crucial redex-depth-to-infinity requirement, are called *strongly convergent*. The point of the redex depth requirement, i.e. of strong convergence, is that it entails a natural notion of 'descendant' or 'residual' carrying over to transfinite reductions, and the notion of descendant is a backbone of the theory of orthogonal rewriting, including λ-calculus. Actually, our definition above is in fact redundant, since the redex depth requirement already implies Cauchy convergence. It is not hard to see that strongly convergent reductions can have at most a countable ordinal as length; if not, we would have some level at which the action (redex contraction) would stagnate forever—but the depth requirement prohibits that. Reductions that are stagnating at some finite level, i.e. that are not strongly convergent, are called *divergent*. There is a helpful analogy between finitary reductions and infinitary (transfinite) reductions: in the former we have finite versus infinite reductions, to be compared with, in the latter, strongly convergent versus divergent reductions.

2.12. NOTATION (Infinitary β-reduction and conversion). (i) Let M, N be terms in $\Lambda^\infty(\bot)$ and suppose that there is a transfinite strongly convergent R-reduction from M to N. Then we write

$$M \twoheadrightarrow_R N.$$

(ii) $M \longrightarrow^\alpha_R N$ (respectively $M \longrightarrow^{\leqslant\alpha}_R N$, $M \longrightarrow^{<\alpha}_R N$) denotes that there is a strongly convergent infinitary R-reduction from M to N with length α (respectively $\leqslant\alpha$, $<\alpha$).

(iii) $=_{R\infty}$ is the infinitary conversion relation corresponding to $\twoheadrightarrow_R$. In fact $=_{R\infty}$ is $({}_R\twoheadleftarrow \circ \twoheadrightarrow_R)^*$, where '$\circ$' denotes relational composition and * transitive closure.

2.13. DEFINITION. (i) A term $M \in \Lambda^\infty$ is in $\boldsymbol{\beta}$-normal form ($\boldsymbol{\beta}$-nf) if it does not contain a $\boldsymbol{\beta}$-redex.

(ii) M has a $\boldsymbol{\beta}^\infty$-nf if $M \twoheadrightarrow_{\boldsymbol{\beta}} N$ and N is in $\boldsymbol{\beta}$-nf.

(iii) $\Lambda^\infty_{\mathrm{NF}} = \{M \in \Lambda^\infty \mid M \text{ is in } \boldsymbol{\beta}\text{-nf}\}$

2.14. EXAMPLE. (An infinite fixed point combinator.) In this example and the next we will present some brief excursions in the infinitary λ-calculus as introduced up to now. Next to illustrating the notions defined above, we also aim in these two examples to suggest the convenience of having available the additional infinitary domain for computations, and moreover that this leads to some observations that may be of interest on their own. In the present example we will encounter an infinite fixed point combinator (fpc). Using the notations for $\mathsf{S}, \mathsf{I}, \mathsf{Y}$ above, consider $\delta \equiv \lambda ab.b(ab)$. Note that $\delta = \mathsf{SI}$. The following is an observation of C. Böhm and G. van der Mey: if Y is a 'reducing fpc', i.e. $Yx \twoheadrightarrow_\beta x(Yx)$ for a variable x, then $Y\delta$ is again a reducing fpc. Indeed, we have

$$Y\delta x \twoheadrightarrow_\beta \delta(Y\delta)x \twoheadrightarrow_\beta x(Y\delta x) \twoheadrightarrow_\beta x^n(Y\delta x).$$

Now let us perform this reduction in an infinitary way, in $\omega + \omega$ steps:

$$Y\delta x \twoheadrightarrow_{\boldsymbol{\beta}} (\lambda f.f^\omega)\delta x \to_\beta \delta^\omega x \equiv \delta(\delta^\omega)x \twoheadrightarrow_\beta x(\delta^\omega x) \twoheadrightarrow_{\boldsymbol{\beta}} x^\omega.$$

Hence $Y\delta$ is indeed behaving as a fpc, and we have $Y\delta =_{\boldsymbol{\beta}^\infty} \lambda x.x^\omega =_{\boldsymbol{\beta}^\infty} Y$.

Note that the above reduction of length $\omega.2$ could have been 'compressed' to one of length ω between the same terms $Y\delta x$ and x^ω, but the resulting reduction would be less natural and informative.

In fact the infinite term $\delta^\omega \equiv \delta(\delta^\omega)$ is itself already a reducing fpc, as the reduction above shows, and we also have $\delta^\omega =_{\boldsymbol{\beta}^\infty} \lambda x.x^\omega =_{\boldsymbol{\beta}^\infty} Y$. So we have encountered a new infinite fpc, δ^ω, or in μ-notation: $\mu x.\delta x$. As an illustration of the richness of the infinitary domain, $\Lambda^\infty(\bot)$, we mention that one can find many more infinite fpc's, e.g., for every n the infinite term $(\mathsf{SS})^\omega \mathsf{S}^{\sim n}\mathsf{I}$ is a fpc. Here $\mathsf{S}^{\sim n}$ denotes a string of n occurrences of S's, with brackets associated to the left; thus for $n = 3$ we have $(\mathsf{SS})^\omega\mathsf{SSSI}$. The simple verification is left to the reader or can be found in Klop [2007].

2.15. Example. [The equation $\mathsf{BYS} = \mathsf{BY}$ and Scott's Induction Rule.] In Scott [1975b], p. 20, the following principle (Scott's Induction Rule) was introduced.

$$\frac{\Gamma, ax \sqsubseteq bx \vdash a(ux) \sqsubseteq b(ux)}{\Gamma, a\bot \sqsubseteq b\bot \vdash a(\mathsf{Y}u) \sqsubseteq b(\mathsf{Y}u)},$$

where $x \notin \mathrm{FV}(\Gamma)$. Scott mentions that the equation $\mathsf{BYS} = \mathsf{BY}$ can be proved using this rule. In Scott [1975a], p. 360, it is conjectured that, using techniques of Böhm, it can be shown that this equation cannot be proved in (finite) λ-calculus, i.e. $\mathsf{BYS} \neq_\beta \mathsf{BY}$. We first show this inconvertibility and then the validity of $\mathsf{BYS} = \mathsf{BY}$ under infinitary conversion $=_{\boldsymbol{\beta}^\infty}$.

2.16. Proposition. (i) *For Curry's fixed point combinator* Y *one has*

$$\mathsf{BYS} \neq_\beta \mathsf{BY}.$$

(ii) *For every fixed point combinator* Y *one has* $\mathsf{BYS} \neq_\beta \mathsf{B}Y$.

Proof. (i) That $\mathsf{BYS} \neq_\beta \mathsf{BY}$ follows immediately from the observation that applying an I to both sides of the equation in question, with result BYSI and BYI, we have $\mathsf{BYSI} =_\beta \Theta$ and $\mathsf{BYI} =_\beta \mathsf{Y}$, respectively Turing's and Curry's fixed point combinator (see notations in Section 2.1). It is well-known that $\Theta \neq_\beta \mathsf{Y}$; a non-trivial but easy exercise establishes this. It follows that $\mathsf{BYS} \neq_\beta \mathsf{BY}$.

Note that Scott [1975a] refers in this discussion to Curry's fpc Y. What if we take another fpc Y in the equation $\mathsf{BY}S = \mathsf{B}Y$? If Y is a fpc in the Böhm sequence

$$\mathsf{Y}^0 \equiv \mathsf{Y}, \mathsf{Y}^1 \equiv \Theta =_\beta \mathsf{Y}\delta, \mathsf{Y}^2 \equiv \mathsf{Y}\delta\delta, \mathsf{Y}^3 \equiv \mathsf{Y}\delta\delta\delta, \ldots,$$

then $\mathsf{BY}^n\mathsf{S} \neq_\beta \mathsf{BY}^n$ follows similarly from the fact that $\mathsf{Y}^n \neq_{\boldsymbol{\beta}} \mathsf{Y}^{n+1}$. In fact we even have $\mathsf{Y}^n \neq_{\boldsymbol{\beta}} \mathsf{Y}^{n+k}$, for all $n, k \geq 0$. For a proof of this result, see Böhm [1963] or Klop [2007].

(ii) Much more difficult it is to prove $\mathsf{B}Y\mathsf{S} \neq_{\boldsymbol{\beta}} \mathsf{B}Y$ for an arbitrary fixed point combinator Y! The proof runs via a deep result of Intrigila [1997], affirming a conjecture by Statman, stating that for no fpc Y we have $Y =_\beta Y\delta$. Indeed, suppose $\mathsf{B}Y\mathsf{S} =_{\boldsymbol{\beta}} \mathsf{B}Y$, for the fpc Y. Then $\mathsf{B}Y\mathsf{SI} =_{\boldsymbol{\beta}} \mathsf{B}Y\mathsf{I}$. Hence

$$Y\delta =_{\boldsymbol{\beta}} Y(\mathsf{SI}) =_{\boldsymbol{\beta}} \mathsf{B}Y\mathsf{SI} =_{\boldsymbol{\beta}} \mathsf{B}Y\mathsf{I} \equiv (\lambda abc.a(bc)Y\mathsf{I} =_{\boldsymbol{\beta}} \lambda c.Y(\mathsf{I}c) =_{\boldsymbol{\beta}} \lambda c.Yc =_{\boldsymbol{\beta}} Y.$$

The last step is justified as follows: $Y(\mathsf{KI}) =_{\boldsymbol{\beta}} \mathsf{KI}(Y(\mathsf{KI})) =_{\boldsymbol{\beta}} \mathsf{I}$, hence Y is solvable, and hence has a hnf, by Barendregt [1984], Theorem 8.3.14. Therefore Y, being closed is β-convertible to $\lambda x.Z$. Then

$$\lambda c.Yc =_{\boldsymbol{\beta}} \lambda c.(\lambda x.Z)c =_{\boldsymbol{\beta}} \lambda c.Z[x := c] \equiv_{\alpha} \lambda x.Z =_{\boldsymbol{\beta}} Y.$$

Therefore the assumption entails $Y\delta =_{\boldsymbol{\beta}} Y$, contradicting Intrigila [1997]. ■

2.17. PROPOSITION. *For every fixed point combinator Y one has* $\mathsf{B}Y\mathsf{S} =_{\boldsymbol{\beta}^\infty} \mathsf{B}Y$.

PROOF. $\mathsf{B}Y\mathsf{S} = \mathsf{B}Y$ (for an arbitrary fpc Y) can be proved conveniently in the framework of infinitary reductions. By a simple computation $\mathsf{B}Y \twoheadrightarrow_{\boldsymbol{\beta}} \lambda ab.(ab)^\omega$ and also $\mathsf{B}Y\mathsf{S} \twoheadrightarrow_{\boldsymbol{\beta}} \lambda ab.(ab)^\omega$. So

$$\mathsf{B}Y =_{\boldsymbol{\beta}^\infty} \lambda ab.(ab)^\omega =_{\boldsymbol{\beta}^\infty} \mathsf{B}Y\mathsf{S}.$$

Note that *en passant*, we have established that $=_{\boldsymbol{\beta}^\infty}$ is not conservative over $=_{\boldsymbol{\beta}}$. In Klop [2007] several other equations of this type are discussed, that do not hold with respect to $=_{\boldsymbol{\beta}}$, but do hold with respect to $=_{\boldsymbol{\beta}^\infty}$. ■

2.4. Basic properties of infinitary λ-calculus

We will briefly present some basic properties of the extended calculus, referring to Terese [2003] Chapter 12 for complete proofs.

In finitary λ-calculus, the two main issues for reduction are the confluence property or Church-Rosser property (CR), stating that two coinitial reductions can be prolonged to a common reduct, and the termination property in the strong variant of Strong Normalization (SN), stating that all reduction sequences eventually must terminate in a normal form, and the weak variant of Weak Normalization (WN), stating merely the existence of a normalizing reduction. The CR property has an important corollary, namely the uniqueness of normal forms (UN). For connections between these and other properties we refer to Barendregt [1984], Chapter 1 of Terese [2003], Klop [1992].

Naturally, the question arises how these properties generalize to the infinitary calculus $\lambda^\infty\boldsymbol{\beta}$. Notationally the extension is easy, and we will consider the properties of infinitary confluence (CR^∞), strong and weak infinitary normalization (SN^∞, WN^∞ respectively), and uniqueness of infinitary nomal forms (UN^∞). Connected to the property CR^∞ we also may consider PML^∞, the infinitary generalization of the fundamental Parallel Moves Lemma (PML), which for finite λ-calculus is the key lemma on the way to CR. Let us define these notions formally.

2.18. DEFINITION. (i) The infinitary Church-Rosser (or confluence) property CR^∞ for $\twoheadrightarrow_R$ is: for all $M_0, M_1, M_2 \in \Lambda^\infty(\bot)$ there exists an $M_3 \in \Lambda^\infty(\bot)$ such that

$$M_0 \twoheadrightarrow_R M_1 \;\&\; M_0 \twoheadrightarrow_R M_2 \;\Rightarrow\; M_1 \twoheadrightarrow_R M_3 \;\&\; M_2 \twoheadrightarrow_R M_3.$$

(Note: we could have given the CR^∞ property mentioning explicitly the length in ordinals of the reductions involved; in view of the Compression property, appearing later, this amounts to the same as the present definition.)

(ii) PML^∞ for $\twoheadrightarrow_R$ is the property similar to CR^∞, but but with one of the coinitial reductions finite: for all $M_0, M_1, M_2 \in \Lambda^\infty(\bot)$ there exists a $M_3 \in \Lambda^\infty(\bot)$ such that

$$M_0 \to_R M_1 \;\&\; M_0 \twoheadrightarrow_R M_2 \;\Rightarrow\; M_1 \twoheadrightarrow_R M_3 \;\&\; M_2 \twoheadrightarrow_R M_3.$$

(iii) A term $M \in \Lambda^\infty(\bot)$ has the infinitary Strong Normalization Property, notation M is SN^∞, if M admits no divergent reductions. In other words all reductions of M eventually terminate in a normal form, possibly after a transfinite β-reduction.

(iv) $M \in \Lambda^\infty(\bot)$ has the WN$^\infty$ property if there exists a $\lambda^\infty\beta$-nf $N \in \Lambda^\infty(\bot)$ such that $M \twoheadrightarrow_{\boldsymbol{\beta}} N$.

2.19. Example. (i) Every fpc Y is WN$^\infty$, its normal form being $\lambda a.a^\omega$. For the fpc's $\mathsf{Y}^0 \equiv \mathsf{Y}, \mathsf{Y}^1 \equiv \Theta =_{\boldsymbol{\beta}} \mathsf{Y}\delta, \mathsf{Y}^n \equiv \mathsf{Y}\delta^{\sim n}$, considered in Example 2.15, we even have SN$^\infty$.

(ii) A term which is WN$^\infty$ but not SN$^\infty$ is $\mathsf{KI}\Omega$. This involves a term which is 'erasing', i.e. not a $\lambda\mathsf{I}$-term, so one may ask whether possibly Church's theorem, stating that for $\lambda\mathsf{I}$-terms M one has the equivalence

$$M \text{ is SN} \iff M \text{ is WN},$$

generalizes to the infinitary setting. However, this is not the case, and a counterexample to this generalization is the fpc $Y^\Omega \equiv \zeta\zeta\Omega$, where $\zeta \equiv \lambda xpf.f(xxpf)$, mentioned in Klop [2007]. This fpc is WN$^\infty$ but not SN$^\infty$, and it is a $\lambda\mathsf{I}$-term.

The following counterexample was independently given in Ariola and Klop [1994] and Berarducci [1996]. The latter paper moreover presented a method to restore CR$^\infty$ by equating a class of problematic terms, namely the ones that have no root stable form (in Berarducci's paper called 'mute' terms) as will be discussed below.

2.20. Lemma (Failure of PML$^\infty$ and CR$^\infty$). *The properties PML$^\infty$ and a fortiori CR$^\infty$, do not hold for infinitary $\lambda^\infty\boldsymbol{\beta}$-calculus.*

Proof. Consider YI. Then on the one hand

$$\mathsf{YI} \to_\beta (\lambda x.\mathsf{I}(xx))(\lambda x.\mathsf{I}(xx)) \twoheadrightarrow_{\boldsymbol{\beta}} \mathsf{I}^\omega,$$

and on the other hand

$$\mathsf{YI} \to_\beta (\lambda x.\mathsf{I}(xx))(\lambda x.\mathsf{I}(xx)) \twoheadrightarrow_\beta (\lambda x.xx)(\lambda x.xx) \equiv \Omega.$$

Both I^ω and Ω only reduce to themselves, so they have no common reduct and PML$^\infty$ and hence also CR$^\infty$ fail. ■

After these negative findings, we now turn in two ways to the positive state of affairs.

The first way of restoring aspects of confluence is as follows. Note that both I^ω and Ω in the proof of Lemma 2.20 are not normal forms. Now, when we impose that one of the terms that are the end points of the coinitial reductions considered for the confluence is a normal form, then confluence does hold.

This fundamental theorem has some beneficial consequences, among which the property UN$^\infty$, the unique normal form property. It was proved in Kennaway et al. [1995b] for first order infinitary TRSs, there called iTRSs, and extended by Ketema and Simonsen [2005] to a wider context, generalizing iTRSs and also our present framework, namely for all orthogonal and 'fully-extended' infinitary Combinatory Reduction Systems (iCRSs, as they are called in Ketema and Simonsen [2006] and [2005]). The notion 'fully extended' excludes a variable condition such as present in the η-reduction rule. For our purpose, we only mention that infinitary λ-calculus extended with the oracle f-rules $\lambda^\infty\boldsymbol{\beta}\mathsf{f}$, is among this large class of higher-order rewrite systems. First we will state formally the unique normal form property together with two variants. We will do this in Definition 2.21 in a general way, namely for Abstract Reduction Systems; then we specify the notation of these properties for the present infinitary λ-calculi.

2.21. Definition. Let $\to_R$ be a reduction relation on some set A, with corresponding conversion relation $=_R$.

(i) R has the *unique normal form property w.r.t. reduction*, notation $\mathrm{UN}(\twoheadrightarrow_R)$, if for all $a, b_1, b_2 \in A$ with b_1, b_2 in R-nf one has

$$a \twoheadrightarrow_R b_1 \;\&\; a \twoheadrightarrow_R b_2 \;\Rightarrow\; b_1 \equiv b_2.$$

(ii) We say that R has the *unique normal form property w.r.t. conversion*, notation $\mathrm{UN}(=_R)$, if for all b_1, b_2 in R-nf one has

$$b_1 =_R b_2 \;\Rightarrow\; b_1 \equiv b_2.$$

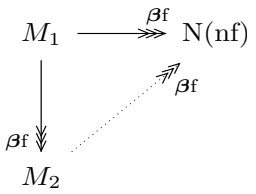
(iii) R has the *normal form property* w.r.t. R, notation $\mathrm{NF}(R)$, if for all $a, b \in A$ with b in R-nf one has

$$a =_R b \;\Rightarrow\; a \twoheadrightarrow_R b.$$

Note that $\mathrm{UN}(=_R) \;\Rightarrow\; \mathrm{UN}(\twoheadrightarrow_R)$, but in general not vice versa.

2.22. Notation. To indicate that we are dealing with infinitary reduction, we will write the properties of Definition 2.21 as UN^∞, NF^∞, specifying always the considered reduction or conversion relation. E.g. we will state 'UN^∞ holds for $\twoheadrightarrow_\beta$', 'UN^∞ holds for $=_{\beta^\infty}$' or 'NF^∞ holds for $=_{\beta^\infty}$'.

2.23. Lemma (Ketema and Simonsen [2006]). *Suppose $M_1 \twoheadrightarrow_{\beta\mathsf{f}} N$ and $M_1 \twoheadrightarrow_{\beta\mathsf{f}} M_2$, with N in $\beta^\infty\mathsf{f}$-nf. Then $M_2 \twoheadrightarrow_{\beta\mathsf{f}} N$.*



This lemma has some useful consequences.

2.24. Corollary. (i) NF^∞ *holds for $=_{\beta^\infty}$ and for $=_{\beta\mathsf{f}^\infty}$.*
(ii) UN^∞ *holds for $\twoheadrightarrow_\beta$ and $=_{\beta^\infty}$; also for $\twoheadrightarrow_{\beta\mathsf{f}}$ and $=_{\beta\mathsf{f}^\infty}$.*
(iii) *Let $M \in \Lambda^\infty(\mathsf{f})$. Suppose $M \in \mathrm{WN}^\infty$ for $\twoheadrightarrow_{\beta\mathsf{f}}$, i.e. M has an infinitary $\beta\mathsf{f}$-nf. Then M is CR^∞ for $\twoheadrightarrow_{\beta\mathsf{f}}$, i.e. two $\twoheadrightarrow_{\beta\mathsf{f}}$-reducts of M have a common reduct.*

The other way of reaching confluence properties is by taking a *congruence*, that is, by working modulo a class of undefined terms, e.g. the class of terms without hnf. This works, because the problematic terms causing non-confluence are always undefined terms. Below in the subsection about Böhm reduction, we will elaborate this route. First we pay attention to the following important feature of infinitary reductions.

Compression

The introduction of reduction sequences of transfinite length α is a natural generalization of finite reductions. But often we do not need the fine distinctions that this length measuring with countable ordinals makes possible. Indeed we can remove the use of transfinite ordinals, by compressing a reduction of length α to one between the same terms of length $\beta \leqslant \omega$. In fact, the infinitary λ-calculus of Berarducci and Intrigila [1996] does without transfinite reductions, and just considers reductions of length at most ω. (Their infinitary λ-calculus can easily be extended to transfinite reductions, though.) So, we have the following Compression property.

2.25. LEMMA. (i) *Let $R : M \to^{\alpha}_{\boldsymbol{\beta}} N$, for some countable ordinal α. Then there exists an infinitary reduction R' of at most ω steps, i.e. $R' : M \to^{\leqslant\omega}_{\boldsymbol{\beta}} N$. This R' is obtained from R* by compression.

(ii) *Compression also holds for $\lambda^\infty\boldsymbol{\beta}\mathsf{f}$-calculus, where the oracle rules for f are added.*

PROOF. (i) See Kennaway and de Vries [2003], p. 690. The compression is a straightforward application of 'dove-tailing'.

(ii) See Ketema and Simonsen [2006] and [2005]. ■

2.26. EXAMPLE. The following reduction

$$[\mathsf{Y}\,a, \mathsf{Y}\,b] \to^{\omega}_{\boldsymbol{\beta}} [a^\omega, \mathsf{Y}\,b] \to^{\omega}_{\boldsymbol{\beta}} [a^\omega, b^\omega],$$

see Notation 2.1, has length $\omega.2$. It can be compressed to length ω by alternating the contraction of a redex 'to the left and to the right.' Since the reduction ends in a nf, in this case all compressed reductions R' are Lévy equivalent with R, see Terese [2003], p. 690.

2.27. REMARK. For the Compression property our definition of strongly convergent reductions is essential. For infinitary reductions that are merely Cauchy convergent, without the depth-to-infinity requirement, compression does not hold. For counterexamples see Terese [2003].

For use in Section 4 we mention the following, anticipating the notion of reduction $\boldsymbol{\beta\Omega}$, treated in the next subsection.

2.28. PROPOSITION. *Let $N \in \Lambda$ be a finite term. Then*

(i) $M \twoheadrightarrow_{\boldsymbol{\beta}} N \;\Rightarrow\; M \twoheadrightarrow_{\beta} N.$

(ii) $M \twoheadrightarrow_{\boldsymbol{\beta\Omega}} N \;\Rightarrow\; M \twoheadrightarrow_{\beta} N.$

(iii) $M \twoheadrightarrow_{\boldsymbol{\beta}\mathsf{f}\boldsymbol{\Omega}} N \;\Rightarrow\; M \twoheadrightarrow_{\boldsymbol{\beta}\mathsf{f}} N.$

PROOF. (i) By compression $M \longrightarrow^{\leqslant\omega}_{\boldsymbol{\beta}} N$. Since $N \in \Lambda$ is finite, α cannot be ω, by the definition of strong convergence.

(ii), (iii) Similarly. ■

Infinitary $\lambda^\infty\boldsymbol{\beta}$-calculus with Böhm reduction

We will now briefly focus on the extension of $\lambda^\infty\boldsymbol{\beta}$-calculus with Ω-reduction rules. Actually, as mentioned in the Introduction, the theory forks in three main directions. (See Terese [2003], chapter 12, for a more elaborate presentation. As a reminder, the definition of weak hnf and of root stable term were already stated in Definition 2.6(ii) and discussed in Remark 2.7.) We introduce the following three infinitary rewrite systems.

2.29. DEFINITION. (i) (For Böhm trees, BTs) The $\lambda^\infty\boldsymbol{\beta\Omega}_3$-calculus is the $\lambda^\infty\boldsymbol{\beta}$-calculus extended with the three Ω-reduction rules given in Definition 2.4.

(ii) (For Lévy-Longo trees, LLTs) The $\lambda^\infty\boldsymbol{\beta\Omega}_2$-calculus is the $\lambda^\infty\boldsymbol{\beta}$-calculus extended with the two Ω-reduction rules:

$$\begin{aligned} M &\to_{\boldsymbol{\Omega}} \bot \qquad \text{if } M \not\equiv \bot \text{ and } M \text{ does not } \beta\text{-reduce to a weak hnf;} \\ \bot M &\to_{\boldsymbol{\Omega}} \bot. \end{aligned}$$

(iii) (For Berarducci trees, BeTs) The $\lambda^\infty\boldsymbol{\beta\Omega}_1$-calculus is the $\lambda^\infty\boldsymbol{\beta}$-calculus extended with the single Ω-reduction rule:

$$M \to_{\boldsymbol{\Omega}} \bot, \qquad \text{if } M \not\equiv \bot \text{ and } M \text{ does not } \beta\text{-reduce to a root stable term.}$$

Note that these three rewrite systems are not orthogonal rewrite systems; the rules display several overlaps, giving rise to non-trivial 'critical pairs'.

We now give a rather different definition of BTs. Whereas the first definition in 2.3 was in a coinductive fashion, the present alternative one is employing infinitary rewriting. We will only treat BTs, and refer just to $\lambda^\infty\boldsymbol{\beta\Omega}$-calculus; the definitions of LLTs and BeTs are entirely analogous.

Also for the calculi yielding LLTs and BeTs we have CR^∞ and the other properties of Corollary 2.34 below. In particular CR^∞ for $\lambda^\infty\boldsymbol{\beta\Omega}_1$ for the BeTs provides an interesting alternative route to UN^∞ for $\lambda^\infty\boldsymbol{\beta}$, based on the following lemma from de Vrijer [1999] on abstract reduction systems. We note that this route was first employed by Berarducci [1996].

2.30. Lemma. *Let $\mathcal{A} = (A, \to_1)$ and $\mathcal{B} = (B, \to_2)$ be two abstract reduction systems (ARSs). Suppose*

(i) $A \subseteq B$;

(ii) $\twoheadrightarrow_1 \subseteq \twoheadrightarrow_2$;

(iii) $\mathrm{NF}(\mathcal{A}) \subseteq \mathrm{NF}(\mathcal{B})$, *where NF of an ARS is the set of its nfs.*

Then $\mathcal{B}$ is UN($\twoheadrightarrow_2$) $\Rightarrow$ $\mathcal{A}$ is UN($\twoheadrightarrow_1$).

Proof. The proof is trivial. If for $a \in A$ has two nfs n_1, n_2, so $a \twoheadrightarrow_1 n_i$, $i = 1, 2$, then $a, n_1, n_2 \in B$ and $a \twoheadrightarrow_2 n_i$, $i = 1, 2$, so $n_1 = n_2$. ■

Now the infinitary calculus $\lambda\boldsymbol{\beta\Omega}^\infty_1$ for BeTs is indeed an extension of $\lambda^\infty\boldsymbol{\beta}$ as ARSs in this Lemma. As CR^∞ holds for $\twoheadrightarrow_{\boldsymbol{\beta\Omega}_1}$, we have UN^∞ for $\twoheadrightarrow_{\boldsymbol{\beta}}$, by Lemma 2.30. Note that this proof manoeuvre would not work for BTs or LLTs: there the third condition in Lemma 2.30 is not satisfied. Namely, for BTs the problem is that L, as in Example 2.37, is a $\boldsymbol{\beta}$-nf, but not a $\lambda^\infty\boldsymbol{\beta\Omega}_3$-nf, the calculus defining BTs. For LLTs an offending term would be the term A, as in Example 2.37, which is also a $\boldsymbol{\beta}$-nf, but not a $\lambda^\infty\boldsymbol{\beta\Omega}_2$-nf, the calculus defining LLTs.

2.31. Definition. Let $M \in \Lambda^\emptyset(\mathbf{c})$, where $\mathbf{c}$ is some set of constants (or variables that we will not bind). Then $BT(M)$ is defined as above, where the $\mathbf{c}$ are treated as constants. We will apply this to various versions of $\lambda(\delta)$ in Section 5, which is a rewriting system consisting of $\Lambda(\delta)$ with some varying notions of reduction, involving the constants δ. Although $\delta xx \to_{\beta\delta} x$, one has $\mathrm{BT}(\delta xx) \not\equiv \mathrm{BT}(x)$, but $\mathrm{BT}(\delta xx) \equiv \delta xx$.

2.32. Lemma. *Let $M \in \Lambda^\infty(\bot)$. Then $\mathrm{BT}(M)$ is a $\boldsymbol{\beta\Omega}^\infty$-nf of M.*

Proof. Definition 2.8(iii) of $\mathrm{BT}(M)$, extended to elements of $\Lambda^\infty(\bot)$ can be seen as an infinitary reduction; the 'depth-to-infinity' requirement clearly is satisfied. ■

2.33. Proposition. *We have CR^∞ for $\lambda^\infty\boldsymbol{\beta\Omega}$.*

Proof. See Terese [2003] Theorem 12.9.6, p. 699. ■

2.34. Corollary. *We have WN^∞ and UN^∞ for $\lambda^\infty\boldsymbol{\beta\Omega}$. More specifically, in the $\lambda^\infty\boldsymbol{\beta\Omega}$-calculus all terms M have $\mathrm{BT}(M)$ as unique $\lambda^\infty\boldsymbol{\beta\Omega}$-nf.*

Proof. By Lemma 2.32 and Proposition 2.33. ■

2.35. Corollary. *Let $M, N \in \Lambda^\infty$. Then*

(i) $M \twoheadrightarrow_{\boldsymbol{\beta\Omega}} \mathrm{BT}(M)$.

(ii) $\mathrm{BT}(M)\mathrm{BT}(N) \twoheadrightarrow_{\boldsymbol{\beta\Omega}} \mathrm{BT}(MN)$.

(iii) $\mathrm{BT}(\mathrm{BT}(M)) \equiv \mathrm{BT}(M)$.

(iv) $M =_{\boldsymbol{\beta\Omega}^\infty} N \iff \mathrm{BT}(M) \equiv \mathrm{BT}(N)$.

(v) $\mathrm{BT}(MN) \equiv \mathrm{BT}(\mathrm{BT}(M)\mathrm{BT}(N))$.

PROOF. (i) By Lemma 2.32.

(ii) Note that $MN \twoheadrightarrow_{\boldsymbol{\beta\Omega}} \mathrm{BT}(M)\mathrm{BT}(N)$, and that $\mathrm{BT}(MN)$ is the $\lambda^\infty\boldsymbol{\beta\Omega}$-nf of MN. Then the result follows by CR^∞ for $\lambda^\infty\boldsymbol{\beta\Omega}$.

(iii) By Corollary 2.34.

(iv) By Corollary 2.34.

(v) By (ii), (iv) and (iii). ■

2.36. REMARK. (i) If a priority is imposed between the $\boldsymbol{\Omega}$-reduction rules and β-reduction, to the effect that the first have precedence over the latter, then the $\lambda^\infty\boldsymbol{\beta\Omega}$-calculus is even SN^∞. If not, SN^∞ fails: Ω has a divergent reduction

$$\Omega \to_\beta \Omega \to_\beta \ldots.$$

(ii) These definitions and facts generalize straightforward to the presence of the oracle f-rules in Definition 2.1.

To conclude this part on BTs, LLTs and BeTs we mention that *mutatis mutandis* similar statements hold for the LLT and BeT setting, most importantly concerning the properties CR^∞, WN^∞ and UN^∞. In the remainder of this paper we will not need LLTs and BeTs.

2.37. EXAMPLE. Write $\mathsf{L} \equiv \lambda x_0(\lambda x_1(\ldots$ and $\mathsf{A} \equiv \mu x.xx$.

(i) Note that

$$\begin{aligned} \mathrm{BT}(\mathsf{YK}) &= \bot. \\ \mathrm{LLT}(\mathsf{YK}) &= \mathsf{L}. \end{aligned}$$

(ii) $$\begin{aligned} \mathrm{BT}((\lambda x.xx)^\omega) &= \bot. \\ \mathrm{LLT}((\lambda x.xx)^\omega) &= \bot. \end{aligned}$$

(iii) $$\mathrm{BeT}((\lambda x.xx)^\omega) = \mathsf{A}.$$

The next lemma is easy to prove but very useful.

2.38. LEMMA (Partial conservativity). (i) *Let $M \in \Lambda^\infty$ and $N \in \Lambda$ in $\boldsymbol{\beta}$-normal form. Then*

$$M =_{\boldsymbol{\beta}^\infty} N \;\Rightarrow\; M \twoheadrightarrow_{\boldsymbol{\beta}} N.$$

(ii) *Let $M \in \Lambda^\infty$ and $N \in \Lambda$ in $\boldsymbol{\beta}\mathsf{f}$-normal form. Then*

$$M =_{\boldsymbol{\beta}\mathsf{f}^\infty} N \;\Rightarrow\; M \twoheadrightarrow_{\boldsymbol{\beta}\mathsf{f}} N.$$

PROOF. (i) If $M =_{\boldsymbol{\beta}^\infty} N$ with $N \in \Lambda_{\mathrm{NF}}$, then $M \twoheadrightarrow_{\boldsymbol{\beta}} N$, by Corollary 2.24(i), hence $M \twoheadrightarrow_\beta N$, by Proposition 2.28(i). Alternatively, note that $M =_{\boldsymbol{\beta}^\infty} N$ implies $M =_{\boldsymbol{\beta\Omega}^\infty} N$, hence applying CR^∞ for $\twoheadrightarrow_{\boldsymbol{\beta\Omega}}$ we get $M \twoheadrightarrow_{\boldsymbol{\beta\Omega}} N$, because $N \in \Lambda_{\mathrm{NF}}$; moreover one has $M \twoheadrightarrow_\beta N$, by Lemma 2.28(ii).

(ii) Similarly. ■

Note that the requirement that $N \in \Lambda_{\mathrm{NF}}$ cannot be dropped. E.g. $\mathsf{Y} =_{\boldsymbol{\beta}^\infty} \mathrm{BT}(\mathsf{Y})$, but $\mathsf{Y} \neq_{\boldsymbol{\beta}} \mathrm{BT}(\mathsf{Y})$.

2.39. DEFINITION. The set of Böhm trees is the following collection.

$$\mathcal{B} = \{M \in \Lambda^\infty(\bot) \mid \exists N \in \Lambda^\infty(\bot).\mathrm{BT}(N) \equiv M\}.$$

In Barendregt [1984] elements of this set are called *Böhm-like trees*; they may not be the BT of a finite λ-term.

2.40. DEFINITION. (i) $\mathcal{B}_\Lambda = \{M \in \mathcal{B} \mid \exists N \in \Lambda.\mathrm{BT}(N) \equiv M\}$.

(ii) $\mathcal{B}_{<\infty} = \{M \in \mathcal{B} \mid M \text{ is finite}\}$.

(iii) $\mathcal{B}_{\mathrm{nf}} = \{M \in \mathcal{B} \mid \exists N \in \Lambda_{\mathrm{NF}}.M \equiv N\}$.

(iv) $\mathcal{B}_{+\bot} = \{M \in \mathcal{B} \mid M \text{ contains a } \bot\}$.

(v) $\mathcal{B}_{-\bot} = \{M \in \mathcal{B} \mid M \text{ is } \bot\text{-free}\}$.

(vi) $\Lambda^\infty_\bullet(\bot) = \{M \in \Lambda^\infty(\bot) \mid \mathrm{BT}(M) \in \mathcal{B}_\bullet\}$,
where $\bullet$ is one of the symbols in $\{\Lambda, <\infty, \mathrm{nf}, +\bot, -\bot\}$

REMARK. $\Lambda^\infty_\Lambda(\bot) = \{M \in \Lambda^\infty \mid \mathrm{BT}(M) \text{ is r.e.}\}$. See Theorem 10.1.23 in Barendregt [1984].

2.41. LEMMA. (i) $\mathcal{B}_{\mathrm{nf}} \subseteq \mathcal{B}_{<\infty} \subseteq \mathcal{B}_\Lambda \subseteq \mathcal{B}$.

(ii) $\mathcal{B}_{-\bot} \cap \mathcal{B}_{<\infty} = \mathcal{B}_{\mathrm{nf}}$.

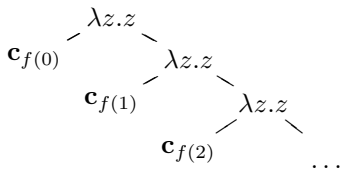
PROOF. Immediate. ■

In order to give examples of specific terms in or outside the given sets, we need the following notation.

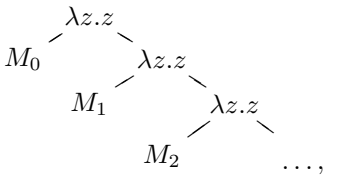
2.42. NOTATION. (i) For $A \subseteq \mathbb{N}$, its *partial* characteristic function χ_A is defined by

$$\begin{array}{rcll} \chi_A(n) & = & 1, & \text{if } n \in A, \\ & = & \uparrow, & \text{else } (\uparrow \text{ denoting 'undefined'}). \end{array}$$

(ii) Let $f : \mathbb{N} \to \mathbb{N}$. Then $\mathcal{G}_f \in \Lambda^\infty$ is defined by



(iii) Let $\psi : \mathbb{N} \hookrightarrow \mathbb{N}$ be a partial unary function. Then $\mathcal{G}_\psi \in \Lambda^\infty(\bot)$ is defined by



where

$$\begin{array}{rcll} M_k & = & \mathbf{c}_{\psi(k)}, & \text{if } \psi(k)\downarrow \text{ (here } \downarrow \text{ denotes 'defined')}, \\ & = & \bot, & \text{else} \end{array}$$

(iv) For $A \subseteq \mathbb{N}$, its *characteristic function* K_A is defined by

$$\begin{array}{rcll} K_A(n) & = & 1, & \text{if } n \in A, \\ & = & 0. & \text{else.} \end{array}$$

(v) $\mathcal{H} = \{n \in \mathbb{N} \mid \varphi_n(n)\downarrow\}$, where φ_e is the unary partial computable function with program e and $\overline{\mathcal{H}}$ is its complement.

2.43. EXAMPLE. The following examples show the general position of the defined subsets of $\mathcal{B}$.

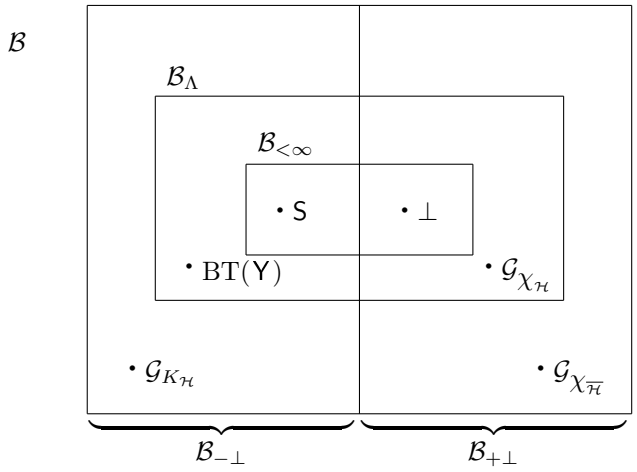


Figure 3: The collection of Böhm trees $\mathcal{B}$ and some subclasses

3. Berry sequentiality

One of the uses of Böhm trees is that they enable us to make a fundamental feature of β-reduction explicit, namely its sequential nature. This may be seen as a restriction in the expressivity of $\lambda\beta$-calculus, because it entails the classical fact that parallel functions like parallel-or are not definable in $\lambda\beta$-calculus. The basic theorem that states this sequentiality is Berry's Sequentiality Theorem (BST), that we will state below. Its main corollary of the non-definability of parallel-or is described in several places (Plotkin [1977], Barendregt [1984], Curien [1993], Berry [1978].) Barendregt [1984] also describes another consequence, the Perpendicular Lines Lemma. In the present paper we will employ BST to show that the operators and rules of Surjective Pairing are not λ-definable. In Barendregt [1974] this was first proved using an ad hoc underlining argument; the use of BST is more 'systematic'. We mention here that all these non-definability corollaries of BST, which we present as an infinitary statement regarding BTs, also are deduced in Endrullis and de Vrijer [2008] in $\lambda\beta$-calculus, so in a finitary framework.

The theorem BST itself is a natural candidate for a treatment in infinitary λ-calculus, as was shown in Bethke et al. [2000]. The fact that BTs can be obtained as the result of an infinite reduction sequence, Corollary 2.34, enables us to perform a tracing argument that shows the origin of the $\bot$s in the output $\mathrm{BT}(M)$, as present in the input term M. (See Fig. 4 below.) The essence of the sequentiality is then intuitively very simple: in a reduction in $\lambda^\infty\boldsymbol{\beta\Omega}$-calculus, the ancestors of symbol occurrences can be traced back towards their origin in the initial term; a symbol either has one ancestor, or it was created (in our case by the first $\boldsymbol{\Omega}$-rule). So by tracing the symbols along the infinite reduction that computes the BT, we discover the 'causal relations' between the output $\bot$s and the input $\bot$s; and that is what BST is about. (This technique is also used in more application-oriented areas under the name of origin-tracking, e.g. for 'program slicing' for error detecting.) The precise details of the tracing procedure are intricate, and will not be considered below. A simpler proof of BST can also be found in Curien [1993]. Here we only hint at the infinitary tracing proof of BST, and concentrate on the two applications, to wit the non-definablity of SP and the derivation of the Perpendicular Lines lemma. Before doing so, we explain in an example what are the possible 'causal relations' between input and output $\bot$s.

3.1. Example. (i) Consider the finite BT-reduction sequence

$$M \equiv (\lambda xy.x\bot)\bot \to \lambda y.\bot\bot \to \lambda y.\bot \to \bot \equiv \mathrm{BT}(M).$$

Now an input z in the first $\bot$ has no output effect:

$$(\lambda xy.xz)\bot \to \lambda y.\bot z \to \lambda y.\bot \to \bot,$$

but with input z in the second $\bot$ we do have non-trivial output:

$$(\lambda xy.x\bot)z \to \lambda y.z\bot.$$

(ii) Let $\omega \equiv \lambda x.xx$. Consider the reduction

$$M \equiv (\lambda xy.x\omega\bot)\omega\bot \to (\lambda y.\omega\omega\bot)\bot \to (\lambda y.\bot\bot)\bot \to (\lambda y.\bot)\bot \to \bot.$$

In this example it is not possible to increase the output $\bot$; for refining both input $\bot$s in M to P, Q, respectively, gives

$$M \equiv (\lambda xy.x\omega P)\omega Q \to (\lambda y.\omega\omega P)Q \to (\lambda y.\bot P)Q \to (\lambda y.\bot)Q \to \bot.$$

(iii) In the two examples above there was only one output $\bot$. In the next example there are three output $\bot$s. Let

$$M \equiv (\lambda xz.zx(\omega\omega)\bot)\bot \twoheadrightarrow [\bot,\bot,\bot] \equiv \mathrm{BT}(M).$$

The second $\bot$ is independent of refinements of M; the first $\bot$ and third $\bot$ are descendants of the $\bot$'s in M in the order of appearance. This example already gives an intuition of why BST holds: the $\bot$s in the output that have no ancestor are the ones that are 'created' during the BT computation, while the output $\bot$s that have an ancestor are their descendants; in other words, they can be traced to input $\bot$s.

(iv) Finally consider

$$M \equiv (\lambda xy.y(xx))\omega \to \lambda y.y(\omega\omega) \to \lambda y.y\bot \equiv \mathrm{BT}(M).$$

Here the term without hnf $\omega\omega$ is created, and the $\bot$ in the final BT has no 'ancestor'. In the Fig. 4 the grey area denotes a spot where a creation is performed; there is no precise origin for the $\bot$ arising from that creation.

Before stating BST and turning to its applications we need to set up some notations.

3.2. Definition. Let $M \in \Lambda^\infty$.

(i) Let $\alpha \in \{0,1\}^*$ be a finite sequence of bits. We can use such sequences to denote positions of subterms of M. The corresponding subterm (occurrence) is denoted by $M|\alpha$. The notion is different from that with the same notation in Barendregt [1984], Definition 10.2.18(ii). In that book one first needs to determine the Böhm tree of M in order to evaluate $M|\alpha$. Moreover, due to the difference in notation for Böhm trees (applicative vs hnf, see Notation 2.9) α in Barendregt [1984] may be a sequence of elements of $\mathbb{N}$, not just of $\{0,1\}$ as in this paper.

(ii) The notion is illustrated for the term $M \equiv [a_1, a_2, [a_3]] \equiv \lambda z.za_1a_2(\lambda w.wa_3)$ of Notation 2.9(i).

$$\begin{array}{rcll} M|[\,] & = & M & \\ M|[0] & = & za_1a_2(\lambda w.wa_3) & \\ M|[1] & = & \uparrow, & \text{i.e. undefined,} \\ M|[00] & = & za_1a_2 & \\ M|[001] & = & a_2 & \\ M|[000] & = & za_1 & \end{array}$$

(iii) If $M|\alpha = \bot$ we write $\bot_\alpha \in M$ to denote the corresponding subterm occurrence of $\bot$ at position α. For example $\bot_{[01]}, \bot_{[001]} \in \lambda f.f\bot\bot$ denote the two subterm occurrences, as can be seen from the tree in Λ^∞.

3.3. DEFINITION. Let $M \twoheadrightarrow N$.

(i) Let $\bot_\alpha \in M$, $\bot_\beta \in N$ be subterm occurrences. We say that $\bot_\beta$ *traces back to* a $\bot_\alpha$ (w.r.t. the given reduction), notation $\bot_\alpha \rightsquigarrow \bot_\beta$, if coloring the different occurrences of $\bot$ in M with different colors and tracing the colors in the reduction $M \twoheadrightarrow_{\boldsymbol{\beta\Omega}} N$ yields the same color for $\bot_\beta$ as that for $\bot_\alpha$. During Ω-reduction steps, like $\bot M \to_\Omega \bot$, the right $\bot$ should have the same color as the left one.

(ii) $\bot_\beta \in N$ is said to be *created* (w.r.t. the given reduction), if it does not trace back to some $\bot_\alpha \in M$.

3.4. EXAMPLE. Let $z(\mathsf{I}\omega\omega)(\omega\bot_1)((\lambda x.yxx)\bot_2) \twoheadrightarrow z\bot_3\bot_4(y\bot_5\bot_6)$, with $\omega \equiv (\lambda x.xx)$. Then $\bot_3$ is created, $\bot_1 \rightsquigarrow \bot_4$, $\bot_2 \rightsquigarrow \bot_5$, and $\bot_2 \rightsquigarrow \bot_6$.

3.5. DEFINITION. (i) Let $M \in \Lambda^\infty$. Write M^s for a term that results from M by replacing occurrences of $\bot$ by arbitrary terms, whereby free variables may be captured. We view s as a 'liberal' substitution operator.

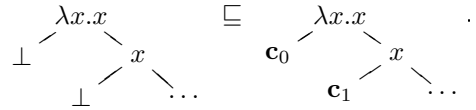
(ii) Let $M, N \in \Lambda^\infty$. We write $M \sqsubseteq N$ if $N \equiv M^s$ for some substitution operator.

Without proof we mention the following.

3.6. REMARK. For $M \in \Lambda^\infty(\bot)$ we have

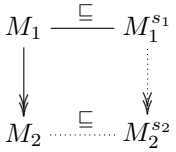
$$\mathrm{BT}(M) \sqsubseteq \mathrm{LLT}(M) \sqsubseteq \mathrm{BeT}(M).$$

3.7. EXAMPLE. (i) $x\Omega(\bot x)(x\bot)(\lambda x.\bot)\bot \sqsubseteq x\Omega(\mathsf{I}x)(x\mathsf{K})(\lambda x.\mathsf{S})\mathsf{Y}$.

(ii)


3.8. PROPOSITION. *Let $M_1 \twoheadrightarrow M_2$.*

(i) *For all substitutions s_1 there exists a substitution s_2 such that $M_1^{s_1} \twoheadrightarrow M_2^{s_2}$. This yields*



(ii) *If moreover $\bot_\alpha \rightsquigarrow \bot_\beta$, then we also may require in (i) that*

$$\begin{aligned} M_1^{s_1}|\alpha = \bot \quad &\Rightarrow \quad M_2^{s_2}|\beta = \bot; \\ M_1^{s_1}|\alpha = z \quad &\Rightarrow \quad M_2^{s_2}|\beta \neq \bot, \end{aligned}$$

where z is some fresh variable.

(iii) *If $\bot_{\alpha_1} \rightsquigarrow \bot_\beta$ and $\bot_{\alpha_2} \rightsquigarrow \bot_\beta$, then $\alpha_1 = \alpha_2$. In other words, every occurrence of $\bot \in M_2$ can be traced back to at most one $\alpha \in M_1$.*

PROOF. (i) By transfinite induction on α, the length of reduction establishing $M \to_{\boldsymbol{\beta\Omega}} N$. During an $\boldsymbol{\Omega}$-step like $\bot P \to \bot$, the substitution gets modified.

(ii) Like (i).

(iii) By (ii). ■

3.9. DEFINITION. Let $M \in \Lambda^\infty(\bot)$ and $M \twoheadrightarrow_{\boldsymbol{\beta\Omega}} N \equiv \mathrm{BT}(M)$. Then

(i) $\bot_\beta \in N$ is *dependent* of a $\bot_\alpha \in M$ if for all $M' \sqsupseteq M$ and all fresh variables z

$$\begin{aligned} M'|\alpha = \bot &\Rightarrow \mathrm{BT}(M')|\beta = \bot, \\ M'|\alpha = z &\Rightarrow \mathrm{BT}(M')|\beta \neq \bot. \end{aligned}$$

(ii) $\bot_\beta \in N$ is *constant* if $\forall M' \sqsupseteq M.\mathrm{BT}(M')|\beta = \bot$.

We will not prove the following Sequentiality theorem, see Bethke et al. [2000] for a proof in the infinitary context.

3.10. Proposition (Berry's Sequentiality Theorem). *Let $M \in \Lambda^\infty(\bot)$ and let $N \equiv \mathrm{BT}(M)$ be the $\boldsymbol{\beta\Omega}^\infty$-nf of M. Then we have the following.*

(i) *Every $\bot_\beta \in N$ is dependent of at most one $\bot_\alpha \in M$.*

(ii) *Those $\bot_\beta \in N$ that are not dependent of any $\bot_\alpha \in M$ are constant.* ■

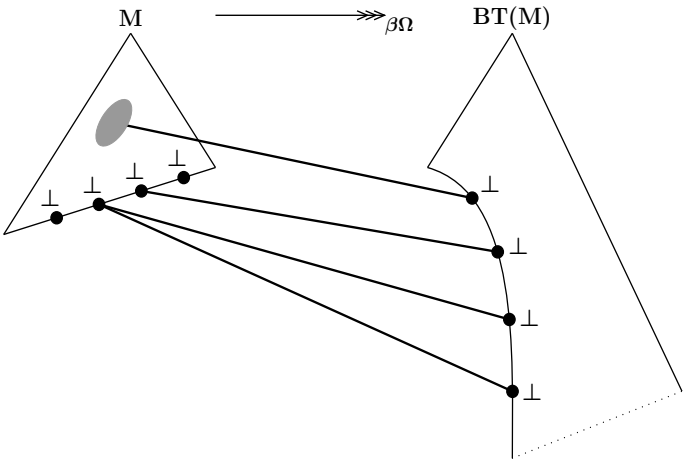


Figure 4: Three occurrences of $\bot$ in $\mathrm{BT}(M)$ trace back to some $\bot$ in M

Now we list some consequences of the Sequentiality Theorem. The following was first proved for finitary λ-calculus in Barendregt [1974], using the technique of underlining. For an alternative proof see de Vrijer [1987] or [2007].

3.11. Proposition. *There are no terms $\pi, \pi_1, \pi_2 \in \Lambda^\infty$ constituting a surjective pairing, i.e. such that*

$$\pi_i(\pi x_1 x_2) = x_i \;\;\&\;\; \pi(\pi_1 x)(\pi_2 x) = x.$$

Proof. Suppose π, π_1, π_2 do exist. Then

$$\pi(\pi_1 \bot)(\pi_2 \bot) = \bot.$$

The RHS $\bot$ is not constant in this situation, as $\pi(\pi_1 \mathsf{I})(\pi_2 \mathsf{I}) = \mathsf{I}$. By Sequentiality this $\bot$ must depend on say the first $\bot$ in the LHS. But then for all X one has

$$\bot = \pi(\pi_1 \bot)(\pi_2(\pi X Y)) = \pi(\pi_1 \bot)Y.$$

By taking the second projection one obtains $\pi_2 \bot = Y$. This implies $X = Y$ for all $X, Y \in \Lambda^\infty$, which is not so by UN^∞ for $\lambda^\infty\boldsymbol{\beta}$-reduction. ■

A second application is the generalization of the perpendicular lines lemma, see Barendregt [1984], Theorem 14.4.12 proved for $=_{\mathrm{BT}}$. It states that any λ-definable map $(\Lambda^\infty)^n \to \Lambda^\infty$ which is constant on n different perpendicular lines is constant everywhere.

3.12. THEOREM (Perpendicular lines lemma). *Let $F, M_{ij}, N_i \in \Lambda^\infty$ for $i,j \in \{1,\ldots,n\}$. Suppose for all $Z \in \Lambda^\infty$ one has (here $=$ stands for $=_{\beta\Omega^\infty}$)*

$$\begin{array}{llllllll} F & M_{11} & M_{12} & \ldots & M_{1(n-1)} & Z & = & N_1; \\ F & M_{21} & M_{22} & \ldots & Z & M_{2n} & = & N_2; \\ & & & & & & \ldots & \\ F & Z & M_{n2} & \ldots & M_{n(n-1)} & M_{nn} & = & N_n. \end{array}$$

Then for all $\vec{Z} = Z_1, \ldots, Z_n$ one has

$$F\vec{Z} = N_1 = \ldots = N_n.$$

PROOF. For notational simplicity we take $n = 3$. That is, let $F, M_{ij}, N_i \in \Lambda^\infty$ with $1 \leqslant i,j \leqslant 3$ be given such that for all $Z \in \Lambda^\infty$ one has

$$\begin{array}{llllll} F & M_{11} & M_{12} & Z & = & N_1 \qquad (1) \\ F & M_{21} & Z & N_{23} & = & N_2 \qquad (2) \\ F & Z & M_{32} & N_{33} & = & N_3. \qquad (3) \end{array}$$

We show that for all M_1, M_2, M_3 one has

$$FM_1M_2M_3 = F\bot\bot\bot.$$

Indeed, write $N \equiv \mathrm{BT}(F\bot\bot\bot)$. Then $F\bot\bot\bot \twoheadrightarrow N$. We have $\mathrm{BT}(FM_1M_2M_3) \sqsupseteq N$, since function application is monotonic w.r.t. $\sqsubseteq$. Suppose that for some M_1, M_2, M_3 the inequality is strict. Then some $\bot_\beta \in N$ is not constant in this situation. Hence this $\bot_\beta$ must depend on one of the three $\bot$s in $F\bot\bot\bot$, say the last one. Then

$$\begin{array}{lcll} N \mid \alpha & = & FM_1M_2\bot \mid \alpha, & \text{since } \bot_\beta \text{ depends on the third } \bot, \\ & = & N_1 \mid \alpha, & \text{by (1)}, \\ & = & FM_1M_2z \mid \alpha & \text{by (1)}, \\ & \neq & N \mid \alpha, & \text{since } \bot_\beta \text{ depends on the third } \bot. \end{array}$$

Therefore the assumption is false and we are done. ■

In Barendregt and Statman [1999] it is proved that the perpendicular lines lemma does not hold for β-equality.

4. Relative computability

In this section we will exploit the fact that infinite λ-terms can have arbitrary complexity. Coding a total number theoretic function $f{:}\mathbb{N} \to \mathbb{N}$ as an infinite λ-term $\mathcal{G}_f \in \Lambda^\infty$, we can use $\mathcal{G}_f$ itself as an oracle in the computation of another function $g : \mathbb{N} \to \mathbb{N}$, where the actual computation is performed by a finite λ-term and β-reduction. That is, we can capture the notion of relative computability, $f \rightsquigarrow g$, i.e. g can be computed with f as oracle, entirely in infinitary β-calculus. As an intermediate and still finite λ-calculus we use $\lambda\mathsf{f}$, as introduced in Definition 2.1. According to Kleene [1963] we have that $f \rightsquigarrow g$ iff g can be computed in $\lambda\mathsf{f}$. Then we connect the finitary $\lambda\mathsf{f}$-calculus with the infinitary $\lambda^\infty\beta$-calculus.

4.1. NOTATION. (i) Let $\mathbb{B}^k = \mathbb{N}^k \to \mathbb{N}$, with $\mathbb{B}^0 = \mathbb{N}$, and $\mathbb{B} = \bigcup_{k \in \mathbb{N}} \mathbb{B}^k$.
(ii) Let $[n_0, \ldots, n_{k-1}]$ be some coding of sequence numbers such that
(1) For all $k > 0$ the function $\lambda\!\!\lambda x_0 \ldots x_{k-1}.[x_0, \ldots, x_{k-1}] \in \mathbb{B}^k$ is computable;

(2) There is a computable $\lambda\!\!\lambda px.(x)_p \in \mathbb{B}^2$ projecting a sequence number onto its components, i.e. such that

$$\forall k, p \forall [n_0, \ldots, n_{k-1}].p < k \;\Rightarrow\; ([n_0, \ldots, n_{k-1}])_p = n_p.$$

(iii) For $g \in \mathbb{B}^k$ define $[g] \in \mathbb{B}^1$ by

$$\begin{array}{rcll} [g](n) & = & g((n)_0, \ldots, (n)_{k-1}), & \text{if } k > 0, \\ & = & g, & \text{if } k = 0. \end{array}$$

The computable functions form the least class of total functions that contain the initial functions (successor, constant zero function, and the projections $\lambda\!\!\lambda x_1 \ldots x_n.x_i$) and that is closed under composition, primitive recursion and minimalization.

4.2. Definition. Let $f, g \in \mathbb{B}$. We say that f *computes* g, notation $f \rightsquigarrow g$ iff g is computable in f (i.e. can be obtained by adding f to the initial functions and closing this collection under substitution, primitive recursion and minimalization).

4.3. Lemma. *For $g \in \mathbb{B}^k$ with $k > 0$ one has $g \rightsquigarrow [g]$ and $[g] \rightsquigarrow g$.*

Proof. Note that for $f \in \mathbb{B}^k$ with $k > 0$ one has for all $n_0, \ldots, n_{k-1}$

$$\begin{array}{rcl} [g](n) & = & g((n)_0, \ldots, (n)_{k-1}); \\ g(n_0, \ldots, n_{k-1}) & = & [g]([n_0, \ldots, n_{k-1}]). \blacksquare \end{array}$$

4.4. Definition. Let $g \in \mathbb{B}^1$. Then $\mathcal{G}_g \in \Lambda^\infty$ is defined as in Notation 2.42. If $g \in \mathbb{B}^k$ with $k \neq 1$, then $\mathcal{G}_g = \mathcal{G}_{[g]}$.

The following lemma states how one can transform $\mathcal{G}_g$ and a term G that λ-defines g in $\lambda^\infty\boldsymbol{\beta}$, in both directions into each other, by application of a finite term.

4.5. Lemma. *Let $g \in \mathbb{B}^1$.*

(i) *There exists an $S \in \Lambda^\emptyset$ such that $S\mathcal{G}_g\mathbf{c}_n \twoheadrightarrow_{\boldsymbol{\beta}} \mathbf{c}_{g(n)}$, for all $n \in \mathbb{N}$.*

(ii) *There exists a $T \in \Lambda^\emptyset$ such that for all $G \in \Lambda^\infty$*

$$[\forall n \in \mathbb{N}.\, G\mathbf{c}_n \twoheadrightarrow_{\boldsymbol{\beta}} \mathbf{c}_{g(n)}] \;\Rightarrow\; TG \twoheadrightarrow_{\boldsymbol{\beta}} \mathcal{G}_g.$$

(iii) *There exists a $T \in \Lambda^\emptyset$ such that for all $G \in \Lambda^\infty(\mathsf{f})$*

$$[\forall n \in \mathbb{N}.\, G\mathbf{c}_n \twoheadrightarrow_{\boldsymbol{\beta}\mathsf{f}} \mathbf{c}_{g(n)}] \;\Rightarrow\; TG \twoheadrightarrow_{\boldsymbol{\beta}\mathsf{f}} \mathcal{G}_g.$$

Proof. (i) Define $S \equiv \mathsf{Y}[\lambda sgn.\mathsf{zero}_?\, n(g\mathsf{K})(s(g(\mathsf{KI})(P^-n))$, where $\mathsf{zero}_?\,\mathbf{c}_0 =_{\boldsymbol{\beta}} \mathsf{K}$, $\mathsf{zero}_?\,\mathbf{c}_{n+1} =_{\boldsymbol{\beta}} \mathsf{KI}$ and $P^-\mathbf{c}_0 =_{\boldsymbol{\beta}} \mathbf{c}_0$, $P^-\mathbf{c}_{n+1} =_{\boldsymbol{\beta}} \mathbf{c}_n$. Then for all $G \in \Lambda^\infty$, $n \in \mathbb{N}$

$$\begin{array}{rcl} S\,G\,\mathbf{c}_0 & =_{\boldsymbol{\beta}} & G\mathsf{K}; \\ S\,G\,\mathbf{c}_{n+1} & =_{\boldsymbol{\beta}} & S\,(G(\mathsf{KI}))\,\mathbf{c}_n, \end{array}$$

a kind of primitive recursion for $\lambda^\infty\boldsymbol{\beta}$. It follows by induction on n that S works:

$$\begin{array}{rcllcll} S\mathcal{G}_g\mathbf{c}_0 & =_{\boldsymbol{\beta}} & \mathcal{G}_g\mathsf{K} & \equiv & [\mathbf{c}_{g(0)}, \mathbf{c}_{g(1)}, \ldots]\mathsf{K} & =_{\boldsymbol{\beta}} & \mathbf{c}_{g(0)} \\ S\mathcal{G}_g\mathbf{c}_{n+1} & =_{\boldsymbol{\beta}} & S(\mathcal{G}_g(\mathsf{KI}))\mathbf{c}_n & =_{\boldsymbol{\beta}} & S[\mathbf{c}_{g(1)}, \mathbf{c}_{g(2)}, \ldots]\mathbf{c}_n & =_{\boldsymbol{\beta}} & \mathbf{c}_{g(n)}. \end{array}$$

(ii) Define $H \equiv \mathsf{Y}(\lambda hgn.[fn, hg(\mathsf{suc}\ n)]$. Then for all $n \in \mathbb{N}$

$$HG\mathbf{c}_n \twoheadrightarrow_\beta [G\mathbf{c}_n, HG\mathbf{c}_{n+1}].$$

Take $T \equiv \lambda f.Hf\mathbf{c}_0$. Then,

$$\begin{array}{rcl} TG \ \to_\beta \ HG\mathbf{c}_0 & \twoheadrightarrow_\beta & [G\mathbf{c}_0, HG\mathbf{c}_1] \\ & \twoheadrightarrow_\beta & [G\mathbf{c}_0, G\mathbf{c}_1, HG\mathbf{c}_2] \\ & \twoheadrightarrow_\beta & [G\mathbf{c}_0, G\mathbf{c}_1, \ldots, HG\mathbf{c}_k] \\ & \twoheadrightarrow_{\boldsymbol{\beta}} & [G\mathbf{c}_0, G\mathbf{c}_1, \ldots] \\ & \twoheadrightarrow_{\boldsymbol{\beta}} & [\mathbf{c}_{g(0)}, \mathbf{c}_{g(1)}, \ldots] \\ & \equiv & \mathcal{G}_g. \end{array}$$

(iii) Similarly. ■

4.6. COROLLARY. *Let $g \in \mathbb{B}^1$. If g is computable, then $\mathcal{G}_g \in \Lambda^\infty_\Lambda(\bot)$.*

PROOF. By the λ-definability of computable functions there exists an $G \in \Lambda^\emptyset$ such that $G\mathbf{c}_n = \mathbf{c}_{g(n)}$, for all $n \in \mathbb{N}$. Hence by Lemma 4.5(ii) we have

$$TG \twoheadrightarrow_{\boldsymbol{\beta}} \mathcal{G}_g \in \Lambda^\infty_\Lambda(\bot). \ \blacksquare$$

Now we repeat Lemma 4.5 for functions of more variables.

4.7. LEMMA. (i) *For every $k \in \mathbb{N}$ there exists an $S_k \in \Lambda^\emptyset$ such that for all $g \in \mathbb{B}^k$*

$$\forall n_1, \ldots, n_k \in \mathbb{N}. S_k \, \mathcal{G}_g \, \mathbf{c}_{n_1} \ldots \mathbf{c}_{n_k} \twoheadrightarrow_{\boldsymbol{\beta}} \mathbf{c}_{g(n_1,\ldots,n_k)}.$$

(ii) *For every $k \in \mathbb{N}$ there exists a $T_k \in \Lambda^\emptyset$ such that for all $g \in \mathbb{B}^k, G \in \Lambda^\infty$*

$$[\forall n_1, \ldots, n_k \in \mathbb{N}. G\mathbf{c}_{n_1} \ldots \mathbf{c}_{n_k} \twoheadrightarrow_{\boldsymbol{\beta}} \mathbf{c}_{g(n_1,\ldots,n_k)}] \ \Rightarrow \ T_k \, G \twoheadrightarrow_{\boldsymbol{\beta}} \mathcal{G}_g.$$

(iii) *For every $k \in \mathbb{N}$ there exists a $T_k \in \Lambda^\emptyset(\mathsf{f})$ such that for all $g \in \mathbb{B}^k, G \in \Lambda^\infty(\mathsf{f})$*

$$[\forall n_1, \ldots, n_k \in \mathbb{N}. G\mathbf{c}_{n_1} \ldots \mathbf{c}_{n_k} \twoheadrightarrow_{\boldsymbol{\beta}\mathsf{f}} \mathbf{c}_{g(n_1,\ldots,n_k)}] \ \Rightarrow \ T_k \, G \twoheadrightarrow_{\boldsymbol{\beta}\mathsf{f}} \mathcal{G}_g.$$

PROOF. Similar to the proof of Lemma 4.5, using the λ-definability of the functions in the proof of Lemma 4.3. ■

4.8. COROLLARY. *Let g be defined from $g_1, \ldots, g_k$ by composition, primitive recursion or minimalisation. Then there exists a $T \in \Lambda^\emptyset$ such that*

$$T\mathcal{G}_{g_1} \ldots \mathcal{G}_{g_k} \twoheadrightarrow_{\boldsymbol{\beta}} \mathcal{G}_g.$$

PROOF. Without loss of generality we do this for $g(x,y) = g_1(g_2(x,y), g_3(x))$. Notice that for $G \equiv \lambda xy.S_2\mathcal{G}_{g_1}(S_2\mathcal{G}_{g_2}xy)(S\mathcal{G}_{g_3}x)$ one has by Lemma 4.7(i)

$$G\mathbf{c}_n\mathbf{c}_m \twoheadrightarrow_\beta S_2\mathcal{G}_{g_1}(S_2\mathcal{G}_{g_2}\mathbf{c}_n\mathbf{c}_m)(S\mathcal{G}_{g_3}\mathbf{c}_n) \twoheadrightarrow_{\boldsymbol{\beta}} S_2\mathcal{G}_{g_1}\mathbf{c}_{g_2(n,m)}\mathbf{c}_{g_3(n)} \twoheadrightarrow_{\boldsymbol{\beta}} \mathbf{c}_{g(n,m)}.$$

Therefore, by Lemma 4.7(ii), for the right $T \in \Lambda^\emptyset$ one has

$$T\mathcal{G}_{g_1}\mathcal{G}_{g_2}\mathcal{G}_{g_3} \twoheadrightarrow_{\boldsymbol{\beta}} T_2(\lambda xy.S_2\mathcal{G}_{g_1}(S_2\mathcal{G}_{g_2}xy)(S\mathcal{G}_{g_3}x)) \twoheadrightarrow_{\boldsymbol{\beta}} T_2G \twoheadrightarrow_{\boldsymbol{\beta}} \mathcal{G}_g. \ \blacksquare$$

Of the following equivalences (1) $\Longleftrightarrow$ (2) was proved in Kleene [1963].

4.9. THEOREM. *Let $f, g \in \mathbb{B}$. Then the following are equivalent.*

(1) $f \rightsquigarrow g$;

(2) $\exists G \in \Lambda^\emptyset(\mathsf{f}) \forall n \in \mathbb{N} \, . \, G\mathbf{c}_n \twoheadrightarrow_{\boldsymbol{\beta}\mathsf{f}} \mathbf{c}_{g(n)}$;

(3) $\exists H \in \Lambda^\emptyset \, . \, H\mathcal{G}_f \twoheadrightarrow_{\boldsymbol{\beta}} \mathcal{G}_g$.

PROOF. We show (3) $\Rightarrow$ (2) $\Rightarrow$ (1) $\Rightarrow$ (3). We give an extra proof of (1) $\Rightarrow$ (2) $\Rightarrow$ (3), to isolate the equivalence between the 'finite' statements (1) and (2) and to shed more light on the transition between the finite (2) and the infinite system (3).

(3)$\Rightarrow$(2) Assume $H\mathcal{G}_f \twoheadrightarrow_{\boldsymbol{\beta}} \mathcal{G}_g$, with $H \in \Lambda^{\emptyset}$. Now $\mathsf{f}\mathbf{c}_n \to_{\mathsf{f}} \mathbf{c}_{f(n)}$, so

$$T\mathsf{f} \twoheadrightarrow_{\boldsymbol{\beta}\mathsf{f}} \mathcal{G}_f,$$

by Lemma 4.5(iii). Hence by assumption $H(T\mathsf{f}) \twoheadrightarrow_{\boldsymbol{\beta}\mathsf{f}} \mathcal{G}_g$. Then for all $n \in \mathbb{N}$

$$\begin{array}{lll} S(H(T\mathsf{f}))\mathbf{c}_n & \twoheadrightarrow_{\boldsymbol{\beta}\mathsf{f}} & \mathbf{c}_{g(n)}, \quad \text{by Lemma 4.5(i),} \\ S(H(T\mathsf{f}))\mathbf{c}_n & \twoheadrightarrow_{\beta\mathsf{f}} & \mathbf{c}_{g(n)}, \quad \text{by Lemma 2.38(ii) or 2.28(ii).} \end{array}$$

Therefore we can take $G \equiv S(H(T\mathsf{f}))$.

(2)$\Rightarrow$(1) The relation $P \to_{\beta\mathsf{f}} Q$ is (after coding) computable in f. This makes $P \twoheadrightarrow_{\beta\mathsf{f}} Q$ and $P =_{\beta\mathsf{f}} Q$ r.e. in f. It follows that

$$\{[n,m] \mid g(n) = m\} = \{[n,m] \mid G\mathbf{c}_n =_{\beta\mathsf{f}} \mathbf{c}_m\}$$

is r.e. in f. Therefore g is computable in f.

(1)$\Rightarrow$(3) Assuming $f \rightsquigarrow g$, we show by induction on the generation of g from f that there exists an $H \in \Lambda^{\emptyset}$ such that $H\mathcal{G}_f \twoheadrightarrow_{\boldsymbol{\beta}} \mathcal{G}_g$. If $g = f$ we can take $H = \mathsf{I}$. If g is an ordinary initial function, then $\mathcal{G}_g \in \Lambda^{\infty}_{\Lambda}(\bot)$, by Corollary 4.6, and we can take $H = \lambda x.M$ for some $M \in \Lambda^{\emptyset}$ such that $M \twoheadrightarrow_{\boldsymbol{\beta}} \mathcal{G}_g$. Now suppose g results from composition, primitive recursion or minimalization from previously obtained functions $g_1, \ldots, g_k$ from f. Then taking $H \equiv (\lambda x.T(H_1x)\ldots(H_kx))$, with T as in Corollary 4.8 one has

$$\begin{array}{lll} H\mathcal{G}_f & \equiv & (\lambda x.T(H_1x)\ldots(H_kx))\mathcal{G}_f \\ & \to_{\beta} & T(H_1\mathcal{G}_f)\ldots(H_k\mathcal{G}_f) \\ & \twoheadrightarrow_{\boldsymbol{\beta}} & T\mathcal{G}_{g_1}\ldots\mathcal{G}_{g_k}, \quad \text{by the induction hypothesis,} \\ & \twoheadrightarrow_{\boldsymbol{\beta}} & \mathcal{G}_g, \quad \text{by Corollary 4.8.(}\blacksquare\text{)} \end{array}$$

(1) $\Rightarrow$ (2) We claim that if $f \rightsquigarrow g$, then g can be λ-defined in $\lambda\beta\mathsf{f}$ by some $G \in \Lambda^{\emptyset}(\mathsf{f})$, i.e. $G\mathbf{c}_n =_{\beta\mathsf{f}} \mathbf{c}_{g(n)}$. This is done by induction of the generation of g from f according to the μ-recursive schemes. For $g = f$ this follows by taking $G = \mathsf{f}$. For the other initial functions λ-definability is trivial. Closure of $\lambda\mathsf{f}$-definability under the schemata of composition, primitive recursion and minimalisation is proved as for the ordinary recursive functions, see e.g. Barendregt [1984], §6.3.

(2) $\Rightarrow$ (3) Given is a $\lambda\mathsf{f}$-term $G \in \Lambda^{\emptyset}(\mathsf{f})$ such that $G\mathbf{c}_n \twoheadrightarrow_{\beta\mathsf{f}} \mathbf{c}_{g(n)}$. Taking $M \equiv \lambda f.G[\mathsf{f} := f])$, we have $M \in \Lambda^{\emptyset}$ and $M\mathsf{f} \to_{\beta} G$. So for all $n \in \mathbb{N}$ we have

$$M\mathsf{f}\mathbf{c}_n \twoheadrightarrow_{\boldsymbol{\beta}\mathsf{f}} \mathbf{c}_{g(n)} \tag{1}$$

By Lemma 4.5(i) we have $S\mathcal{G}_f\mathbf{c}_n \twoheadrightarrow_{\boldsymbol{\beta}} \mathbf{c}_{f(n)}$ and hence by Lemma 2.38(i)

$$S\mathcal{G}_f\mathbf{c}_n \twoheadrightarrow_{\beta} \mathbf{c}_{f(n)}.$$

In reduction (1) we replace all occurrences of f by $(S\mathcal{G}_f)$, and steps $\mathsf{f}\mathbf{c}_n \to_{\mathsf{f}} \mathbf{c}_{f(n)}$ by the finite reduction $S\mathcal{G}_f\mathbf{c}_n \twoheadrightarrow_{\beta} \mathbf{c}_{f(n)}$. The result is a finite β-reduction starting with an infinite term: $M(S\mathcal{G}_f)\mathbf{c}_n \twoheadrightarrow_{\beta} \mathbf{c}_{g(n)}$, for all $n \geq 0$. Hence by Lemma 4.5(ii) we have $T(M(S\mathcal{G}_f)) \twoheadrightarrow_{\boldsymbol{\beta}} \mathcal{G}_g$. So we can take $H \equiv (\lambda g.T(M(Sg)))$. ■

5. Non-left linear reduction

In Section 3 we have discussed the non-definability of Surjective Pairing, as defined by π, π_0,π_1 and the equations $\pi_0xy = x, \pi_1xy = y, \pi(\pi_0x)(\pi_1x) = x$. It turns out

that not only definability is problematic for these reduction rules, but also the (finitary) confluence property for the extension of $\lambda\boldsymbol{\beta}$-calculus with these rules. Turning these equations into the reduction rules $\pi_0 xy \to x, \pi_1 xy \to y, \pi(\pi_0 x)(\pi_1 x) \to x$ yields a non left-linear system, due to the repetition of the variable x in the lefthand-side of the third rule. The question remained whether this trio of reduction rules, which we will also refer to as SP, can be added to the $\lambda\boldsymbol{\beta}$-calculus such that the resulting system is CR. In Klop [80] it was shown that the addition yields non-confluence, thus solving a problem in the list of open problems in Böhm [1975], p. 367. The 'correctness proof' of these CR-counterexamples in Klop [1980] was rather elaborate, requiring standardization and postponement arguments. But it was also suggested there that an excursion to the realm of infinite terms could convey the essence of the counterexample in a more succinct way; see also Barendregt [1984] Section 15.3. In the present section we will elaborate this suggestion in detail.

We will discuss the following four versions of a non-left linear rule, to be added to λ-calculus, in increasing order of difficulty.

5.1. DEFINITION (J. Staples). The notion of reduction δ_S is defined on $\Lambda(\delta, \varepsilon)$ by the rule

$$\delta xx \to_{\delta_S} \varepsilon.$$

5.2. PROPOSITION. *The notion of reduction $\boldsymbol{\beta}\delta_S$ is not CR. By a fixed point construction there are terms $C, A \in \Lambda(\delta, \varepsilon)$ such that*

$$\begin{aligned} Cx &\twoheadrightarrow_{\boldsymbol{\beta}} \delta x(Cx), \\ A &\twoheadrightarrow_{\boldsymbol{\beta}} CA. \end{aligned}$$

Then $C\epsilon =_{\boldsymbol{\beta}\delta_S} \epsilon$, but these terms have no common reduct.

PROOF. We have the (more-step) $\beta\delta$-reductions

$$\begin{array}{ccccccc} A & \twoheadrightarrow_{\boldsymbol{\beta}} & CA & \twoheadrightarrow_{\boldsymbol{\beta}} & \delta A(CA) & \twoheadrightarrow_{\boldsymbol{\beta}} & \delta(CA)(CA) \to_{\delta_S} \varepsilon \\ & & \downarrow & & & & \\ & & C\varepsilon & & & & \end{array}$$

The three terms $CA, C\varepsilon$ and ε form a counterexample against the CR property. In this case it is easily proved that $C\varepsilon \not\twoheadrightarrow \varepsilon$, i.e. $C\varepsilon$ and ε have no common reduct, as is left to the reader. ■

As a preparation to the other more complicated versions, we look at the infinite normal forms of the three terms just mentioned in this proof.

In fact these are $\bot$-free Böhm trees, since there are no terms without a head normal form in the reducts of the terms in consideration. The BT's, see Definition 2.31 for the notion of BT for terms in $\Lambda(\delta, \epsilon)$, turn out to be infinite regular trees. Employing the μ-notation as in Example 2.10 they are as follows.

$$\begin{aligned} \mathrm{BT}(CA) &\equiv \mu x.\delta xx \equiv \Delta, \\ \mathrm{BT}(C\varepsilon) &\equiv \mu x.\delta\varepsilon x, \\ \mathrm{BT}(\varepsilon) &\equiv \varepsilon. \end{aligned}$$

A slightly more difficult extension is the following.

5.3. DEFINITION. (Klop [1980]) The notion of reduction δ_K is defined on $\Lambda(\delta, \epsilon)$ by

$$\delta xx \to_{\delta_K} \varepsilon x.$$

5.4. PROPOSITION. *The notion of reduction $\boldsymbol{\beta}\delta_K$ is not CR.*

PROOF. Defining the same terms $C, A \in \Lambda(\delta, \epsilon)$ as in Proposition 5.2 we have the following.

$$\begin{array}{ccccccc} A & \twoheadrightarrow & CA & \twoheadrightarrow & \delta A(CA) & \twoheadrightarrow & \delta(CA)(CA) \xrightarrow[\delta_K]{} \varepsilon(CA). \\ & & \downarrow & & & & \\ & & C(\varepsilon(CA)) & & & & \end{array}$$

Now it is a bit more laborious to show that $\varepsilon(CA) \not\twoheadrightarrow C(\varepsilon(CA))$, which was done in Klop [1980] using finitary arguments. The infinitary argumentation employs the BT's of the three relevant terms CA, $\varepsilon(CA)$ and $C(\varepsilon(CA))$. They are Δ, $\varepsilon\Delta$, and $\mu x.\delta(\varepsilon\Delta)x$, respectively. The treatment will be analogous to the more complicated version introduced next, and will therefore not be given here separately. ■

5.5. REMARK. For Propositions 5.2 and 5.4 the situation is:

$$M \to_\delta M' \;\Rightarrow\; \mathrm{BT}(M) \to_\delta^{\leqslant\omega} \mathrm{BT}(M').$$

As a notational reminder $\to_\delta^{\leqslant\omega}$ stands for a δ-reduction of length $\leqslant \omega$. For the next counterexamples the situation is more complex and we need a definition.

5.6. DEFINITION. (i) An occurrence of δ is called *balanced* if it is the head of a δ-redex δMM, with $M \in \Lambda^\infty(\delta, \epsilon)$.

(ii) Analogously, for the case of Surjective Pairing below, an occurrence of π is called *balanced* if it is the head of a π-redex $\pi(\pi_0 M)(\pi_1 M)$, with $M \in \Lambda^\infty(\pi, \pi_0, \pi_1)$.

A slightly more complex variant of δ-reduction comes close to Surjective Pairing.

5.7. DEFINITION. (J.R. Hindley) The notion of reduction δ_H is defined on $\Lambda(\delta)$ by

$$\delta xx \to_{\delta_H} x.$$

The reason that δ_H is more complex than the versions in Definitions 5.1 and 5.3 lies in the possibility that new redexes can be created by application of the δ_H-rule, which is now a collapsing rule (i.e. the RHS is a single variable), e.g. $\delta_H \mathsf{III} \to_{\delta_H} \mathsf{II}$. For Surjective Pairing the same holds.

5.8. PROPOSITION. *The notion of reduction $\boldsymbol{\beta}\delta_H$ is not CR. By a fixed point construction there are terms C, A such that*

$$\begin{array}{rcl} Cx & \twoheadrightarrow & \varepsilon(\delta x(Cx)) \\ A & \twoheadrightarrow & CA, \end{array}$$

PROOF. We have reductions that are almost similar to the ones for $\boldsymbol{\beta}\delta_K$.

$$\begin{array}{ccccccc} A & \twoheadrightarrow & CA & \twoheadrightarrow & \varepsilon(\delta A(CA)) & \twoheadrightarrow & \varepsilon(\delta(CA)(CA)) \to \varepsilon(CA) \\ & & \downarrow & & & & \\ & & C(\varepsilon(CA)) & & & & \end{array}$$

The BTs of the relevant trio of terms CA, $\varepsilon(CA)$, $C(\varepsilon(CA))$ are respectively the trees $\mu x.\varepsilon(\delta xx) \equiv T$, εT and $\mu x.\varepsilon(\delta(\varepsilon T)x)$. The corresponding cyclic graphs are drawn in the lower plane in Figure 5.

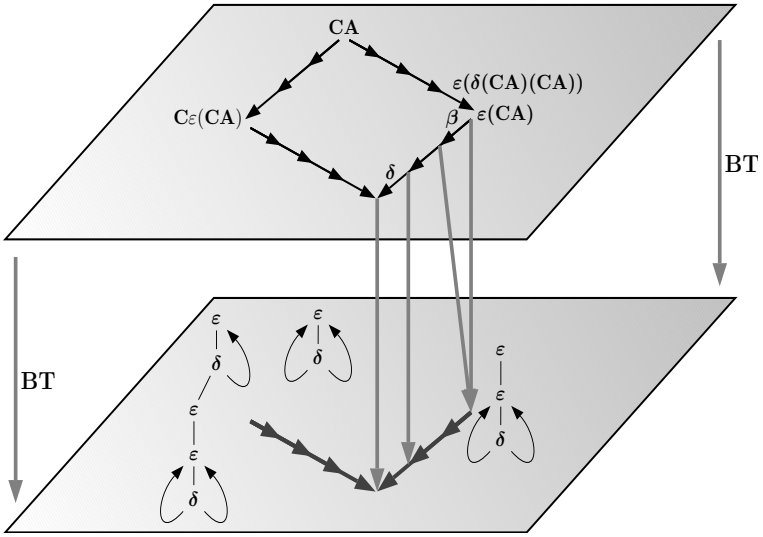


Figure 5: Projection by BT

First note that there was an *unbalancing effect* leading to BT($C(\epsilon(CA))$ (the leftmost cyclic graph in Fig. 5) whose top δ is unbalanced.

Now we will prove that indeed we have $(\varepsilon(CA)) \not\downarrow C(\varepsilon(CA))$, by an excursion to the infinitary setting, depicted in Figure 5. The upper plane is that of finite terms, projected to the lower plane of infinite terms via the operation BT of taking the Böhm tree. The question whether in the finitary plane the terms $C(\varepsilon(CA))$ and $\varepsilon(CA)$ have a common $\beta\delta$-reduct, translates in the infinitary plane to the question whether the infinite terms BT($C(\varepsilon(CA))$) and BT($\varepsilon(CA)$), rendered as cyclic term graphs in the figure, are convergent by means of steps resulting from projections of β- and δ-steps. Here there is a bonus: the projection of a β-step trivializes, because it follows from $M \to_\beta M'$ that BT(M) $\equiv$ BT(M').

How does a δ-step translate? Intuitively, as a possibly infinite sequence of δ-steps on infinite trees, so $\longrightarrow^{\leqslant\omega}{}_\delta$. Possibly infinite, because a δ-redex in the upper plane may have infinitely many descendants after the BT-projection. But it is immediately clear from inspection of BT($C(\varepsilon(CA))$) and BT($\varepsilon(CA)$) that such steps do not have an effect, for two reasons, which are best seen in the cyclic graph of BT($C(\varepsilon(CA))$). It contains two δ's, the lower balanced, the upper unbalanced. Contracting a balanced δ keeps the tree the same, due to the cyclicity: the contractum is identical to the contracted δ-redex. Contracting an unbalanced δ is not even possible, by definition of δ-reduction. Hence BT($C(\varepsilon(CA))$) cannot be altered, and therefore it cannot be confluent with BT($\varepsilon(CA)$).

Now let us consider the translation of a δ-step in more detail. In order to tackle this problem, we will introduce a new constant γ that describes 'sharing', with the new rules $\delta xx \to \gamma x$ and $\gamma x \to \mathsf{I}x$ where $\mathsf{I} \equiv \lambda x.x$. We will call these rules $(\delta\gamma)$ and $(\gamma\mathsf{I})$ respectively, to be read as 'δ to γ' and 'γ to I'. The δ-step $\delta MM \to M$ is now splitted in three:

$$\delta MM \to_{\delta\gamma} \gamma M \to_{\gamma\mathsf{I}} \mathsf{I}M \to_{\boldsymbol{\beta}} M.$$

The new rules $(\delta\gamma)$ and $(\gamma\mathsf{I})$ are extended to infinite terms in the obvious way.

Example. Let $\Delta \equiv \mu x.\delta xx$ be the infinite binary tree of δ's as above. Then

$$\Delta \equiv \delta\Delta\Delta \to_{\gamma\delta} \gamma\Delta \to_{\gamma\delta} \gamma^2\Delta \to^{\omega}_{\gamma\delta} \gamma^\omega \equiv \mu x.\gamma x.$$

(Note that this is a strongly convergent reduction.)

We now have the situation as in Figure 6, corresponding to the following.

$$\begin{array}{lrcl}
(1) & M_0 \to_{\delta\gamma} M_1 & \Rightarrow & BT(M_0) \to_{\delta\gamma}^{\leqslant\omega} BT(M_1); \\
(2) & M_1 \to_{\gamma\mathsf{I}} M_2 & \Rightarrow & BT(M_1) \to_{\gamma\mathsf{I}}^{\leqslant\omega} P; \\
(3) & M_0 \to_{\delta} M_3 & \Rightarrow & BT(M_0) \to_{\delta\gamma}^{\leqslant\omega} \to_{\gamma\mathsf{I}}^{\leqslant\omega} \twoheadrightarrow_{\boldsymbol{\beta\Omega}} \mathrm{BT}(M_3).
\end{array}$$

$$\begin{array}{ccc}
M_0 \equiv C[\delta MM] & \longrightarrow & \mathrm{BT}(M_0) \\
{\scriptstyle \delta\gamma}\downarrow & & {\scriptstyle \delta\gamma}\downarrow {\scriptstyle \leqslant\omega} \\
M_1 \equiv C[\gamma M] & \longrightarrow & \mathrm{BT}(M_1) \\
{\scriptstyle \gamma\mathsf{I}}\downarrow & & {\scriptstyle \gamma\mathsf{I}}\downarrow {\scriptstyle \leqslant\omega} \\
M_2 \equiv C[\mathsf{I}M] & \longrightarrow & P \\
{\scriptstyle \boldsymbol{\beta}}\downarrow & & {\scriptstyle \boldsymbol{\beta\Omega}}\downarrow {\scriptstyle \leqslant\omega} \\
M_3 \equiv C[M] & \longrightarrow & \mathrm{BT}(M_3)
\end{array}$$

Figure 6: Fine-structure of δ-parallel moves

As to (1): a δ-redex in M_0 is preserved as (possibly infinitely many) δ-redexes in $\mathrm{BT}(M_0)$. That this is so, is best seen by evaluating the BT not in an arbitrary way, but using Knuth-Gross 'steps'. A Knuth-Gross 'step' starting from a finite term M consists of the complete development of all β-redexes in M simultaneously. In other words, we apply the Knuth-Gross reduction strategy to compute the BT. The point is that in this way, in each Knuth-Gross 'step', δ-redexes are preserved. See Barendregt [1984], Def. 13.2.7 for the precise definition of the Knuth-Gross strategy. That δ-redexes are indeed preserved, after a Knuth-Gross 'step', is an easy exercise. That this remains so in the limit, $\mathrm{BT}(M_0)$, is obvious.

As to (2): the intermediate tree P is not yet a BT. This is so because subterms (subtrees) without hnf may have arisen, necessitating further normalisation by replacing these by $\bot$, to obtain a BT.

Now we can conclude. Consider the infinite terms εT and $\mu x.\varepsilon(\delta(\varepsilon T)x)$, with $T \equiv \mu x.\varepsilon(\delta x x)$, to be made confluent in the infinite plane, where we have to employ 'macro steps' steps like:

$$\to_{\delta\gamma}^{\leqslant\omega} \to_{\gamma\mathsf{I}}^{\leqslant\omega} \twoheadrightarrow_{\boldsymbol{\beta\Omega}} .$$

However, we will not come far in this way; the only change that can be effectuated is the (total or partial) transformation of T into $\mu x.\varepsilon(\gamma x) \equiv G$. But doing so, the unbalanced δ displayed in $\mu x.\varepsilon(\delta(\varepsilon T)x)$ cannot be balanced, and will therefore prohibit a confluence with εT. ■

The most complicated extension is λ-calculus plus Surjective Pairing as in the introduction of this section.

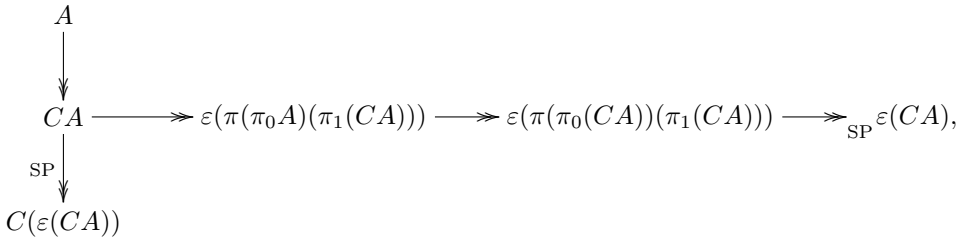
5.9. THEOREM. *The notion of reduction* $\boldsymbol{\beta}\,\mathrm{SP}$ *on* $\Lambda(\pi, \pi_1, \pi_2)$

$$\pi_i(\pi M_1 M_2) \to_{\mathrm{SP}} M_i \quad \pi(\pi_1 M)(\pi_2 M) \to_{\mathrm{SP}} M.$$

is not CR. By a fixed point construction there are terms $C, A \in \Lambda(\pi, \pi_1, \pi_2)$ *such that*

$$\begin{array}{rcl}
Cx & \twoheadrightarrow_{\boldsymbol{\beta}} & \varepsilon(\pi(\pi_0 x)(\pi_1(Cx))), \\
A & \twoheadrightarrow_{\boldsymbol{\beta}} & CA.
\end{array}$$

Then



while $\varepsilon(CA)$ and $C(\varepsilon(CA))$ have no common reduct.

PROOF. Again we compute the BT's of the three relevant terms.

$$\begin{array}{rcl} BT(CA)) & \equiv & \mu x.\varepsilon(\pi((\pi_0 x)(\pi_1 x))) \equiv S. \\ BT(\varepsilon(CA)) & \equiv & \varepsilon S. \\ BT(C(\varepsilon(CA))) & \equiv & \mu x.\varepsilon(\pi(\pi_0(\varepsilon S)(\pi_1 x))). \end{array}$$

The remainder of the infinitary proof using these BTs is entirely analogous to the treatment of the previous δ_H-version, requiring only a notational adaptation, which is left to the reader. ■

6. Concluding remarks and questions

In this paper we have endeavoured to give some examples of applications of rewriting with infinite λ-terms, or infinitary λ-calculus. Several questions remain, of which we specifically mention the following.

- It would be interesting to investigate the precise relation of Scott's Induction Rule (SIR), that we encountered in Example 2.15, to the present infinitary setting. Is it true that infinitary λ-conversion $=_{\beta^\infty}$, includes all consequences of SIR?
- Above, we introduced the μ-notation as a convenient notation for regular infinite λ-trees; this amounts just to cyclic graphs of λ-terms. Mixing the μ-terms with λ-calculus, allowing β-reduction under the μ, provides for faster evaluation. It would be interesting to pursue studies of term graph rewriting against the background of infinitary λ-calculus, as a continuation of work by Kennaway et al. [1995a], and Ariola and Klop [1994], [1996], [1996], [1997], where this theme was studied with reference to infinitary first order rewriting.
- It will be interesting to extend the result in Section 4 on relative computability from total functions to partial functions.

Acknowledgements. We cordially thank the two anonymous referees for many suggestions pertaining to detailed improvements a well as instigating major changes in the first version. We are grateful to Jeroen Ketema, for many detailed suggestions and raising useful critical points, and to Roel de Vrijer and Vincent van Oostrom for useful discussions. Our thanks also to Marianne Klop-Leicher for help in the typographical rendering of this paper, using LyX, and to Jörg Endrullis for programming some of the figures using pstricks.

References

Abramsky, S. and C.-H. L. Ong [1993]. Full abstraction in the lazy lambda calculus, *Inform. and Comput.* **105**(2), pp. 159–267.

Abramsky, S., Dov M. Gabbay and T.S.E. Maibaum (eds.) [1992]. *Handbook of Logic in Computer Science, Volume 2: Background: Computational Structures*, Oxford Science Publications.

Ariola, Z.M. and J.W. Klop [1994]. Cyclic lambda graph rewriting, *Proc. of the 9th Annual Symposium on Logic in Computer Science (LICS'94)*, pp. 416–425.

Ariola, Z.M. and J.W. Klop [1996]. Equational term graph rewriting, *Fundamenta Informaticae* **26**(3/4), pp. 207–240. Extended version as University of Oregon Technical Report CIS-TR-95-16.

Ariola, Z.M. and J.W. Klop [1997]. Lambda calculus with explicit recursion, *Information and Computation* **139**(2), pp. 154–233.

Barendregt, H. P. [1974]. Pairing without conventional restraints, *Z. Math. Logik Grundlagen Math.* **20**, pp. 289–306.

Barendregt, H. P. [1977]. *The type-free lambda calculus*, Elsevier, pp. 1092–1132. In Barwise et al. [1977].

Barendregt, H. P. [1984]. *The Lambda Calculus: its Syntax and Semantics*, revised edition, North-Holland, Amsterdam.

Barendregt, H. P. and R. Statman [1999]. Applications of Plotkin terms: partitions and morphisms for closed terms, *J. Funct. Programming* **9**(5), pp. 565–575.

Barendsen, E. [2003]. *Term graph rewriting*, Cambridge University Press, chapter 13 in Terese [2003], pp. 712–743.

Barwise, J. with the cooperation of H. J. Keisler, K. Kunen, Y. N. Moschovakis and A. S. Troelstra (eds.) [1977]. *Handbook of mathematical logic*, Studies in Logic and the Foundations of Mathematics, Vol. 90, North-Holland Publishing Co., Amsterdam.

Berarducci, A. [1996]. Infinite lambda-calculus and non-sensible models, *in:* A. Ursini and P. Aglianò (eds.), *Logic and Algebra (Pontignano 1994)*, Lecture Notes in Pure and Applied Mathematics Series 180, Marcel Dekker Inc., pp. 339–378.

Berarducci, A. and B. Intrigila [1996]. Church-Rosser λ-theories, infinite λ-terms and consistency problems, *Logic: from foundations to applications (Staffordshire, 1993)*, Oxford Sci. Publ., Oxford Univ. Press, New York, pp. 33–58.

Berry, G. [1978]. Séquentialité de l'évaluation formelle des λ-expressions, *Transformations de programmes (Proc. Third Internat. Sympos. Programming, Paris, 1978)*, Dunod, Paris, pp. 67–80.

Bethke, I., J. W. Klop and R. de Vrijer [2000]. Descendants and origins in term rewriting, *Inform. and Comput.* **159**(1-2), pp. 59–124. RTA-98 (Tsukuba).

Böhm, C. [1963]. The CUCH as a Formal and Description Language, *in:* Richard Goodman (ed.), *Annual Review in Automatic Programming, Vol. 3*, Pergamon Press, Oxford, pp. 179–197.

Böhm, C. (ed.) [1975]. *λ-calculus and Computer Science Theory*, LNCS 37, Springer.

Curien, Pierre-Louis [1993]. *Categorical combinators, sequential algorithms, and functional programming*, Progress in Theoretical Computer Science, second edition, Birkhäuser Boston Inc., Boston, MA.

Endrullis, J. and R. de Vrijer [2008]. Reduction under substitution, *in:* A. Voronkov (ed.), *Proceedings of RTA'2008*, LNCS 5117/2008, Springer, pp. 425–440.

Intrigila, B. [1997]. Non-existent Statman's double fixed point operator does not exist, *Information and Computation* **137**, pp. 35–40.

Kahrs, S. [2007]. Infinitary rewriting: meta-theory and convergence, *Acta Inform.* **44**(2), pp. 91–121.

Kennaway, J. R., J. W. Klop, M. R. Sleep and F.-J. de Vries [1995a]. Infinitary lambda calculus and Böhm models, *Proc. Conference on Rewriting Techniques and Applications*, LNCS 914, Springer, pp. 257–270.

Kennaway, J. R., J. W. Klop, M. R. Sleep and F.-J. de Vries [1997]. Infinitary lambda calculus, *Theoret. Comput. Sci.* **175**(1), pp. 93–125. Non-standard logics and logical aspects of computer science (Kanazawa, 1994).

Kennaway, J.R., J.W. Klop, M.R. Sleep and F.-J. de Vries [1994]. On the adequacy of graph rewriting for simulating term rewriting, *ACM Transactions on Programming Languages and Systems* **16**(3), pp. 493–523.

Kennaway, J.R., P. Severi, M.R. Sleep and F.-J de Vries [2005]. Infinitary rewriting: from syntax to semantics, *in:* A. Middeldorp, V. van Oostrom, F. van Raamsdonk and R. de Vrijer (eds.), *Processes, Terms and Cycles: Steps on the Road to Infinity*, LNCS 3838, Springer, pp. 148–173.

Kennaway, R. and F.-J. de Vries [2003]. *Infinitary rewriting*, Cambridge University Press, chapter 12 in Terese [2003], pp. 668–711.

Kennaway, R., J. W. Klop, R. Sleep and F.-J. de Vries [1995b]. Transfinite reductions in orthogonal term rewriting systems, *Inform. and Comput.* **119**(1), pp. 18–38.

Ketema, J. and J. G. Simonsen [2005]. Infinitary combinatory reduction systems, *in:* J. Giesl (ed.), *Proceedings of the 16th International Conference on Rewriting Techniques and Applications (RTA 2005)*, number 3467 in *Lecture Notes in Computer Science*, Springer-Verlag, pp. 438–452.

Ketema, J. and J. G. Simonsen [2006]. Infinitary combinatory reduction systems, *Technical Report Technical Report D-558*, DIKU, University of Copenhagen.

Kleene, S. C. [1963]. Recursive functionals and quantifiers of finite types. II, *Trans. Amer. Math. Soc.* **108**, pp. 106–142.

Klop, J.W. [1980]. *Combinatory Reduction Systems*, Dissertation, Utrecht University. Appeared as Mathematical Centre Tracts 127, Kruislaan 413, 1098 SJ Amsterdam. Available at `<web.mac.com/janwillemklop/iWeb/Site/Bibliography.html>`.

Klop, J.W. [1992]. Term rewriting systems, Abramsky et al. [1992], Oxford University Press, pp. 1–116.

Klop, J.W. [2007]. New fixed point combinators from old, *in:* E. Barendsen, V. Capretta, H. Geuvers and M. Niqui (eds.), *Reflections on Type Theory, λ-Calculus, and the Mind. Essays dedicated to Henk Barendregt on the occasion of his 60th birthday*, Radboud University Nijmegen, pp. 197–211. Available at <www.cs.ru.nl/barendregt60>.

Klop, J.W. and R. de Vrijer [2005]. Infinitary normalization, *in:* S. N. Artemov, H. Barringer, A. S. d'Avila Garcez, L. C. Lamb and J. Woods (eds.), *We will show them: Essays in honour of Dov Gabbay*, Vol. 2, College Publications, pp. 169–192.

Longo, G. [1983]. Set-theoretical models of λ-calculus: theories, expansions, isomorphisms, *Ann. Pure Appl. Logic* **24**(2), pp. 153–188.

Plotkin, G. [1977]. LCF considered as a programming language, *Theoretical Computer Science* **5**, pp. 225–255.

Scott, D. S. [1975a]. Open problem, *in:* C. Böhm (ed.), *λ-Calculus and Computer Science Theory (Proc. Sympos., Rome, 1975)*, LNCS 37, Springer, Berlin, p. 368.

Scott, D. S. [1975b]. Some philosophical issues concerning theories of combinators, *in:* C. Böhm (ed.), *λ-Calculus and computer science theory (Proc. Sympos., Rome, 1975)*, LNCS 37, Springer, Berlin, pp. 346–366.

Terese [2003]. *Term Rewriting Systems*, Cambridge University Press.

de Vrijer, R. [1987]. *Surjective pairing and strong normalization: two themes in lambda calculus*, Dissertation, University of Amsterdam.

de Vrijer, R. [1999]. Conditional linearization, *Indagationes Mathematicae, N.S.* **10**(1), pp. 145–159.

de Vrijer, R. [2007]. Barendregt's lemma, *in:* E. Barendsen, V. Capretta, H. Geuvers and M. Niqui (eds.), *Reflections on Type Theory, λ-Calculus, and the Mind. Essays dedicated to Henk Barendregt on the occasion of his 60th birthday*, Radboud University Nijmegen, pp. 275–284. Available at <www.cs.ru.nl/barendregt60>.