Contents

Chapter 2
Infinitary lambda calculus

2.0. Introduction.
The aim of this chapter, which is based on a paper in statu nascendi in cooperation with Henk Barendregt, 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 (BT's), but in the extended setting of infinitary λ-calculus (or λ^∞ for short) BT's are just a particular kind of infinite normal forms, and in the extended setting we can even apply a BT to another BT. In our Section on Preliminaries we will give a somewhat more detailed exposition of λ^∞ with β-reduction, $\lambda^\infty\beta$ 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 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, while the third family is a sophisticated tool for consistency studies as demonstrated in Berarducci-Intrigila [].

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

The third reason is found in the feature of *expressivity*. Infinite λ-terms can be non-recursive. 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 2.5, we will substantiate this.

The last reason, illustrated by Section 2.7 on Berry's Sequentiality Theorem (BST) and Section 2.9 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 xx 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.

2.1. Infinite lambda terms

We assume familiarity with ordinary untyped λ-calcuclus, see e.g. Barendregt [1984]. In particular the following notations will be used.

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) $\mathsf{I} \equiv \lambda x.x$, $\mathsf{K} \equiv \lambda xy.x$, $\mathsf{S} \equiv \lambda xyz.xz(yz)$, $\mathsf{Y} \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$, $\Omega \equiv (\lambda x.xx)(\lambda x.xx)$, $\mathsf{B} \equiv \lambda xyz.x(yz)$, $\Theta \equiv (\lambda xy.y(xxy))(\lambda xy.y(xxy))$, and the Church numerals are $\mathbf{c}_n \equiv \lambda fx.f^n x$. The set of lambda terms is denoted by Λ, that of normal forms (under β reduction) by Λ_{NF}. The set of closed λ-terms is denoted by $\Lambda^{\emptyset}$. The notation $[M, N] \equiv \lambda z.zMN$, with z a fresh variable, is used for pairing.

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 β-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* $\beta\mathsf{f}$ *are Church-Rosser.*

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

2.1.1. DEFINITION.

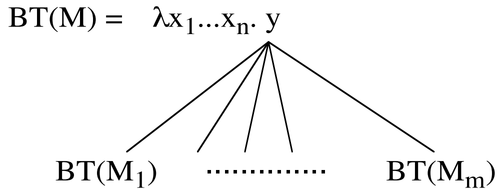
(i) Λ^∞ is the set of possibly infinite λ-terms, coinductively defined by

term::= x | term @ term | λx.term

(ii) $\Lambda^\infty(\perp)$ is defined similar, now allowing also occurrences of the constant $\perp$.

(iii) Certain elements of $\Lambda^\infty(\perp)$ are known as *Böhm trees of finite λ-terms* M $\in \Lambda$, notation BT(M), defined in Barendregt [84] by the following coinductive definition:

BT(M) = Ω if M is unsolvable (equivalently, has no head normal form);



if M has hnf $\lambda x_1 ... x_n .\ y M_1 ... M_m$.

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

2.2. β-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

```
 @
/ \
λx  ...
|
....
```

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, of which we will refrain here; instead we refer to a detailed treatment in Terese [03], 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 [84] 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 [03], 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 $\Lambda^\infty(\perp)$; we will not repeat the definitions as they are verbatim the same.

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

2.2.1. REMARK. The definition of $\Lambda^\infty(\perp)$ is very liberal, and admits also some pathological terms that are meaningless, but that will not bother us for our present purposes. Example ... below exhibits some of these 'garbage' terms.

2.2.2. NOTATION. (i) (*Applicative versus hnf notation.*)
For BT's the natural notation is the one suggested by our definition above, also employed in Barendregt [84]. We call this notation the 'head normal form notation' or hnf-notation. Sometimes this notation is less appropriate, namely when BT's are applied to each other, or in general to another term. Then the 'applicative' notation is more convenient, which extends the well-known term tree notation of finite terms where application (@) is a binary node, abstraction (λx) is a unary node, and variables and constants ($\perp$, **f**, ...) are terminal nodes. Examples of both notation formats are displayed in Figure 2.1.

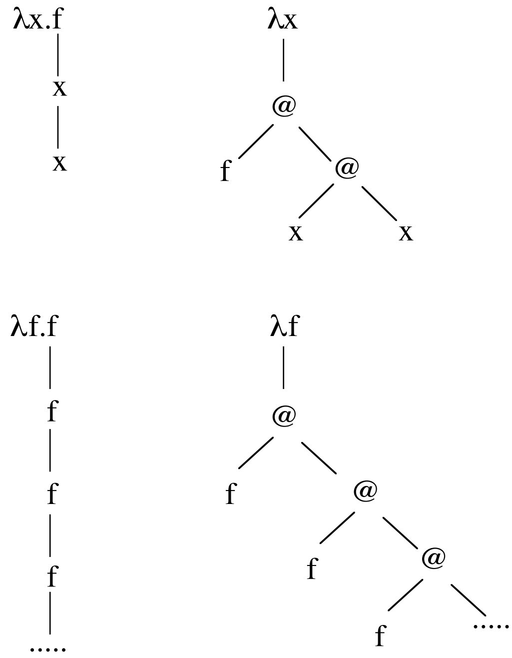


Figure 2.1. $\lambda x.f(xx)$ and BT(Y) in hnf and applicative notation.

(ii) A notation that we will sometimes use for $M \in \Lambda^\infty(\perp)$, is M^ω, defined by $M^\omega \equiv M(M^\omega)$. For instance $BT(Y) = \lambda f.f^\omega$. An interesting term that will play a role below, is I^ω. Note that this term contains infinitely many β-redexes, and reduces in one step to itself.

(iii) We generalize (ii), especially for use in the final section 2.9, to the well-known μ-notation; in this notation we have $M^\omega \equiv \mu x.Mx$. So $\mu x.xx \in \Lambda^\infty(\perp)$ is the binary tree consisting of appication nodes only.

2.2.3. REMARK. Whether a term such as $\mu x.xx$ is useless (i.e. equals $\perp$) 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 $M \equiv Y\omega$, where $\omega \equiv \lambda x.xx$. Then it is an easy exercise to show that M has no hnf, and thus $BT(M) \equiv \perp$. We could also take the BT after reducing M to its infinite normal form in $\Lambda^\infty(\perp)$: $M \twoheadrightarrow\!\!\!\!\rightarrow_\beta \mu x.xx$. (Note that this reduction is indeed strongly convergent: redex depth tends to infinity.) Now, residing in $\Lambda^\infty(\perp)$, we again have BT(μx.xx) ≡ ⊥, because μx.xx is a normal form, which is not a hnf, hence has no hnf.

Also in the semantics of Lévy-Longo trees (LLT's), this term and its infinite normal form μx.xx, both have LLT = ⊥.

However, in the Berarducci tree semantics, which gives a syntactic model of λ-calculus, these terms do have a non-trivial semantical value–namely μx.xx.

For the moment, we will only employ the coarsest of the three semantical views, namely that of BT's.

2.2.4. Next we introduce infinite β-reduction sequences. We will do this in an informal way, referring for a full detailed treatment to Terese [03], Kennaway et al.[], Ketema [], Klop & de Vrijer []. 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 ,... ∈ $\Lambda^\infty(\perp)$. We have single β-steps $M_\gamma \to_\beta M_{\gamma+1}$. For a limit ordinal λ, M_λ is the Cauchy limit of the earlier M_μ, $\mu < \lambda$, 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 $M_0 \equiv \Omega \equiv \omega\omega$ 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 ∈ $\Lambda^\infty(\perp)$ 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 disallowed, 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 defintion 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.

2.2.5. NOTATION. (*Infinitary β-reduction and conversion.)*
(i) Let $M,N \in \Lambda^\infty(\perp)$ and suppose that there is a transfinite strongly convergent reduction from M to N. Then we write $M \twoheadrightarrow\!\!\!\!\rightarrow_\beta N$.
(ii) $=_{\beta\infty}$ is the infinitary conversion relation corresponding to $\twoheadrightarrow\!\!\!\!\rightarrow_\beta$.

2.2.6. 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 S,I,Y above, consider $\delta \equiv \lambda ab.b(ab)$. Note that $\delta = SI$. The following is a well-known observation of C. Böhm: if Y is a 'reducing fpc', i.e. $Yx \twoheadrightarrow x(Yx)$ for a variable x, then $Y\delta$ is again a reducing fpc. Indeed, we have $Y\delta x \twoheadrightarrow \delta(Y\delta)x \twoheadrightarrow x(Y\delta x)$.

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

$$Y\delta x \twoheadrightarrow\!\!\!\!\rightarrow_\beta (\lambda f.f^\omega)\, \delta x \rightarrow_\beta \delta^\omega x \equiv \delta(\delta^\omega)x \rightarrow_\beta \rightarrow_\beta x(\delta^\omega x) \twoheadrightarrow\!\!\!\!\rightarrow_\beta x^\omega.$$

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

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 $\delta^\omega =_{\beta\infty} \lambda x.x^{\ \omega} =_{\beta\infty} Y$. 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, $\Lambda^\infty(\perp)$, we mention that one can find many more infinite fpc's, e.g., for every n the infinite term $(SS)^\omega S^{\sim n}I$ is a fpc. Here $S^{\sim n}$ denotes a string of n S's, with brackets associated to the left; thus for n = 3 we have $(SS)^\omega SSSI$. The simple verification is left to the reader, or can be found in Klop [2007], New fixed point combinators from old.

2.2.7. EXAMPLE. (*The equation BYS = BY and Scott's Induction Rule*)

In Scott [75] the question is discussed whether there are interesting equations that cannot be proved in (finite) λβ-calculus, but can be proved using Scott's Induction Rule (SIR). Scott mentions the equation BYS = BY and notes that with SIR this is easily proved, and conjectures that $BYS \neq_\beta BY$. We consider both points.

(i) That $BYS \neq_\beta BY$ follows immediately from the observation that applying an I to both sides of the equation in question, with result BYSI and BYI, we have $BYSI =_\beta \Theta$ and $BYI =_\beta Y$, respectively Turing's and Curry's fixed point combinator (see notations in Section 2.1). It is well-known that $\Theta \neq_\beta Y$; a non-trivial but easy exercise establishes this. It follows that $BYS \neq_\beta BY$.
Nota bene. Scott [75] refers in this discussion to Curry's fpc Y. What if we take another fpc in the equation BYS = BY? If Y is a fpc in the Böhm sequence $Y_0 \equiv Y$, $Y_1 \equiv \Theta =_\beta Y\delta$, $Y_2 \equiv Y\delta\delta$, $Y_3 \equiv Y\delta\delta\delta$,... then $BY_nS \neq_\beta BY_n$ follows similarly from the fact that $Y_n \neq_\beta Y_{n+1}$. In fact we even have $Y_n \neq_\beta Y_{n+k}$, for all $n,k \geq 0$. (A proof of this folklore result is in Klop [2007]). Much more difficult is it to prove $BYS \neq_\beta BY$ for an *arbitrary* fpc Y! Then the proof needs a deep result from Intrigila [], affirming a conjecture by Statman, stating that for no fpc Y we have $Y =_\beta Y\delta$. Assuming this, we have $BYS \neq_\beta BY$ just as before.

(ii) On the other hand, BYS = BY (for an arbitrary fpc Y) can be proved using SIR. We will not invoke SIR, but instead use the framework of infinitary reductions, and our point is that this is convenient to work with. Indeed we have by a simple computation

$$BY \twoheadrightarrow_\beta \lambda ab.(ab)^\omega \text{ and also } BYS \twoheadrightarrow_\beta \lambda ab.(ab)^\omega.$$

So $BY =_{\beta\infty} BYS =_{\beta\infty} \lambda ab.(ab)^\omega$.

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

2.3. Basic properties of infinitary λ-calculus

We will briefly present some basic properties of the extended calculus, referring to Terese 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 [84], Chapter 1 of Terese [03], Klop [92].

Naturally, the question arises how these properties generalize to the infinitary calculus $\lambda^\infty\beta$. Notationnally the extension is easy, and we will consider the properties of infinitary confluence (CR^∞), SN^∞ and WN^∞ for strong and weak infinitary normalization, and UN^∞ for uniqueness of infinitary nomal forms. 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.

2.3.1. Definition. (i) The infinitary Church-Rosser (or confluence) property CR^∞ is:
$\forall M_0, M_1, M_2 \in \Lambda^\infty(\perp)\ \exists M_3 \in \Lambda^\infty(\perp)$:

$$M_0 \twoheadrightarrow_\beta M_1 \ \&\ M_0 \twoheadrightarrow_\beta M_2 \Rightarrow M_1 \twoheadrightarrow_\beta M_3 \ \&\ M_2 \twoheadrightarrow_\beta M_3$$

(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) The property of uniqueness of infinitary normal forms UN^∞ is:
$\forall M \in \Lambda^\infty(\bot)\ \forall N_1, N_2 \in \Lambda^\infty{}_{nf}$:

$$M \twoheadrightarrow\!\!\!\!\rightarrow_\beta N_1 \ \&\ M \twoheadrightarrow\!\!\!\!\rightarrow_\beta N_2 \Rightarrow N_1 \equiv N_2$$

(iii) PML^∞ is the property similar to CR^∞, but but with one of the coinitial reductions finite:
$\forall M_0, M_1, M_2 \in \Lambda^\infty(\bot)\ \exists M_3 \in \Lambda^\infty(\bot)$:

$$M_0 \twoheadrightarrow_\beta M_1 \ \&\ M_0 \twoheadrightarrow\!\!\!\!\rightarrow_\beta M_2 \Rightarrow M_1 \twoheadrightarrow\!\!\!\!\rightarrow_\beta M_3 \ \&\ M_2 \twoheadrightarrow\!\!\!\!\rightarrow_\beta M_3$$

(iv) A term $M \in \Lambda^\infty(\bot)$ has the infinitary Strong Normalization Property, M is SN^∞, if M admits no divergent reductions, in other words all reductions of M terminate eventually in a normal form.

(v) $M \in \Lambda^\infty(\bot)$ has the WN^∞ property if there exists $M \in \Lambda^\infty{}_{nf}$ such that $M \twoheadrightarrow\!\!\!\!\rightarrow_\beta N$.

2.3.2. EXAMPLE. (i) Every fpc Y is WN^∞, its normal form being $\lambda a.a^\omega$. For the fpc's $Y_0 \equiv Y$, $Y_1 \equiv \Theta =_\beta Y\delta$, $Y_n \equiv Y\delta^{\sim n}$ considered in Example 2.2.6, 2.2.7, we even have SN^∞.
(ii) A term which is WN^∞ but not SN^∞ is $KI\Omega$. This involves a term which is 'erasing', i.e. not a λI-term, so one may ask whether possibly Church's theorem, stating that for λI-terms M one has the equivalence M is SN $\Leftrightarrow$ M 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^∞ but not SN^∞, and it is a λI-term.

2.3.3. LEMMA. (*Failure of CR^∞ and PML^∞.*)
The properties PML^∞ *and a fortiori* CR^∞*, do not hold for infinitary* $\lambda\beta^\infty$*-calculus.*

PROOF. Consider YI. Then on the one hand $YI \rightarrow_\beta (\lambda x.I(xx))(\lambda x.I(xx)) \twoheadrightarrow\!\!\!\!\rightarrow_\beta I^\omega$, and on the other hand $YI \rightarrow_\beta (\lambda x.I(xx))(\lambda x.I(xx)) \twoheadrightarrow_\beta (\lambda x.xx)(\lambda x.xx) \equiv \Omega$. Both I^ω and Ω only reduce to themselves, so they have no common reduct and PML^∞ and hence also CR^∞ fail.

After these negative findings, we now turn to the positive state of affairs. There are two main ways of restoring confluence. Note that both I^ω and Ω in the proof above 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^∞. We will call this property the *normal form property w.r.t reduction*, $NF(\twoheadrightarrow_\beta)$; later we will also have the normal form property w.r.t. infinitary conversion, $NF(=_{\beta\infty})$. 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 presen in the η-reduction rule. For our purpose, we only mention that infinitary λ-calculus extended with the oracle **f**-rules $\lambda\beta\mathbf{f}^\infty$, is among this large class of higher-order rewrite systems.

2.3.4. LEMMA. (Ketema-Simonsen [2006])
Let $M_1 \twoheadrightarrow_{\beta\mathbf{f}} N$, *a normal form, and let* $M_1 \twoheadrightarrow_{\beta\mathbf{f}} M_2$. *Then* $M_2 \twoheadrightarrow_{\beta\mathbf{f}} N$.

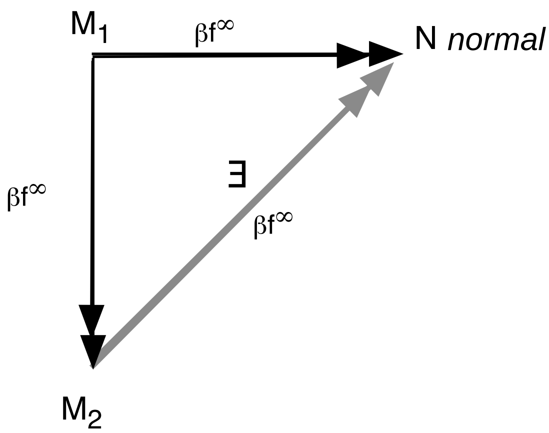


Figure 2.2.

This lemma has some useful consequences:

2.3.5. COROLLARY. *In* $\lambda^\infty\beta\mathbf{f}$*-calculus we have*
(i) the property of infinitary unique normal forms w.r.t reduction $\mathrm{UN}^\infty(\twoheadrightarrow_{\beta\mathbf{f}})$;
(ii) if $M \in \Lambda^\infty(\bot)$, *then* M *is* $\mathrm{WN}^\infty \Rightarrow$ M is CR^∞.

2.3.6. 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 length of length $\beta \le \omega$. In fact, the infinitary λ-calculus of Berarducci and Intrigila [xx] 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.

2.3.7. LEMMA. *(i) Let* R: $M_0 \twoheadrightarrow_\beta M_1$ *be an infinitary* $\lambda^\infty\beta$*-reduction of length* α, *a countable ordinal. Then there is an infinitary* $\lambda^\infty\beta$*-reduction of length* $\beta \le \omega$.
(R' is obtained from R by compression.)
(ii) Compression also holds for $\lambda^\infty\beta\phi$*-calculus, where the oracle rules for* ϕ *are added.*

PROOF. See Terese [03]. The compression is a straightforward application of 'dovetailing'.

2.3.8. EXAMPLE. $[Ya,Yb] \twoheadrightarrow_\beta [a^\omega, Yb] \twoheadrightarrow_\beta [a^\omega, b^\omega]$

2.3.9. 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 [03].

2.3.10. Infinitary conversion $=_{\beta\infty}$ **and** $=_{\beta\mathbf{f}\infty}$

2.10. 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.

2.4. Infinitary λ-calculus with Böhm reduction

We will now briefly focus on the extension of $\lambda^\infty\beta$ with Ω-reduction rules. Actually, as mentioned in the Introduction, the theory forks in three main directions. (See Terese [03], where also the notion of 'root stable term can be found.) We introduce the following three infinitary rewrite systems.

2.4.1. DEFINITION.
(i) (*For Böhm trees, BT's*) The $\lambda^\infty\beta\Omega_3$-calculus is the $\lambda^\infty\beta$-calculus extended with the three Ω-reduction rules given in Definition 2.3.
(ii) (*For Lévy-Longo trees, LLT's*) The $\lambda^\infty\beta\Omega_2$-calculus is the $\lambda^\infty\beta$-calculus extended with the two Ω-reduction rules:

$M \to_\Omega \bot$ if $M \neq \bot$ and M does not β-reduce to a weak hnf;

$\bot M \to_\Omega \bot$.

(iii) (*For Berarducci trees, BeT's*) The $\lambda^\infty\beta\Omega_1$-calculus is the $\lambda^\infty\beta$-calculus extended with the single Ω-reduction rule: $M \to_\Omega \bot$ if $M \neq \bot$ and M does not β-reduce to a root stable term.

Caveat: 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 very different definition of BT's. Whereas the first definition in 2.3 was in a coinductive fashion, the present alternative one is employing infinitary rewriting. We will only treat BT's, and refer just to $\lambda^\infty\beta\Omega$-calculus; the definitions of LLT's and BeT's are entirely analogous.

2.4.2. DEFINITION. Let $M \in \Lambda^\infty(\perp)$. Then BT(M) is the unique normal form with respect to infinitary reduction with Ω-rules, $\twoheadrightarrow_{\beta\Omega}$.

Indeed BT(M) is well-defined in this way, because:

2.4.3. LEMMA. (i) $\lambda^\infty\beta\Omega$-calculus has the properties WN^∞, CR^∞, and UN^∞.

2.4.4. COROLLARY. Let $M, N \in \Lambda^\infty(\perp)$. Then:

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

(ii) $BT(MN) =_{\beta\Omega\infty} BT(M).BT(N)$

(iii) $BT(BT(M)) \equiv BT(M)$

(iv) $M =_{\beta\Omega\infty} N \Leftrightarrow BT(M) \equiv BT(N)$.

2.4.5. REMARK. (i) If a priority is imposed between the Ω-rules and β-reduction, to the effect that the first have precedence over the latter, then the $\lambda^\infty\beta\Omega$-calculus is even SN^∞.
(ii) These definitions and facts generalize straightforward to the presence of the oracle φ-rules in Definition 2.1.0 (begin of section 2.1).

2.4.6. PROPOSITION.

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

$$\begin{array}{rlcl}
\text{(i)} & M \twoheadrightarrow_{\beta} N & \Rightarrow & M \twoheadrightarrow_{\beta} N. \\
\text{(ii)} & M \twoheadrightarrow_{\beta\Omega} N & \Rightarrow & M \twoheadrightarrow_{\beta} N. \\
\text{(iii)} & M \twoheadrightarrow_{\beta f\Omega} N & \Rightarrow & M \twoheadrightarrow_{\beta f} N.
\end{array}$$

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

(ii), (iii) Similarly. ■

2.34. 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 $M \twoheadrightarrow\!\!\!\!\rightarrow_{\boldsymbol{\beta}} N$, by Corollary 2.20(i), hence $M \twoheadrightarrow_{\boldsymbol{\beta}} N$, by Proposition 2.24(i). Alternatively, note that $M =_{\boldsymbol{\beta}^\infty} N$ implies $M =_{\boldsymbol{\beta\Omega}^\infty} N$, hence applying CR^∞ for $\twoheadrightarrow\!\!\!\!\rightarrow_{\boldsymbol{\beta\Omega}}$ we get $M \twoheadrightarrow\!\!\!\!\rightarrow_{\boldsymbol{\beta\Omega}} N$, because $N \in \Lambda_{\mathrm{NF}}$; moreover one has $M \twoheadrightarrow_{\boldsymbol{\beta}} N$, by Lemma 2.24(ii).

(ii) Similarly. ■

2.5. 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: N → 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: N → 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 ≻ g, meaning that g can be computed with f as oracle, entirely in infinitary λ-calculus. As an intermediate and still finite λ-calculus we use λϕ, as introduced in Definition 2.1.0 (there the new constant is called f.) According to Kleene [63] we have that f ≻ g iff g can be computed in λϕ. Then we connect the finitary λϕ-calculus with the infinitary λ-calculus, $\lambda^\infty\beta$. The proof starts from the equivalence proved in Kleene [63].

We need the following preparation:

2.5.1. PROPOSITION. There is an $R \in \Lambda^\circ$ such that $Rx \twoheadrightarrow_\beta (x\mathbf{c}_0, x\mathbf{c}_1, x\mathbf{c}_2,\ldots)$. Here (-,-,-,...) is the construction for infinite sequences with the property that $(a_0, a_1, a_2,\ldots)\, \mathbf{c}_n \twoheadrightarrow_\beta a_n$.

PROOF. Exercise.

2.5.2. THEOREM. *The following are equivalent:*

(1) f ≻ g

(2) $\exists M \in \Lambda^\circ(\phi)\ \forall n{\geq}0\ M\mathbf{c}_n \twoheadrightarrow_{\beta f} \mathbf{c}_{g(n)}$

(3) $\exists T \in \Lambda^\circ\ T\mathcal{G}_f \twoheadrightarrow_\beta \mathcal{G}_g$

PROOF.

(2)⇒(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) ⇒ (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) ⇒ (3). Given is a λφ-term $M \in \Lambda^\circ(\phi)$ such that $M\mathbf{c}_n \twoheadrightarrow_{\beta f} \mathbf{c}_{g(n)}$.

Take $N \equiv \lambda f.\ M[\phi := f])$, so $N \in \Lambda^\circ$, and $N\phi \to_\beta M$. So $N\phi\mathbf{c}_n \twoheadrightarrow_{\beta f} \mathbf{c}_{g(n)}$.

In this reduction we replace all occurrences of φ by $\mathcal{G}_f$, and every step $\phi\mathbf{c}_n \to_f \mathbf{c}_{f(n)}$ by the finite reduction $\mathcal{G}_f\,\mathbf{c}_n \twoheadrightarrow_\beta \mathbf{c}_{f(n)}$. The result is a finite β-reduction starting with an infinite term: $N\mathcal{G}_f\,\mathbf{c}_n \twoheadrightarrow_\beta \mathbf{c}_{g(n)}$, for all n≥0. We now employ the term R from the Proposition above:
$R(N\mathcal{G}_f) \twoheadrightarrow_\beta (N\mathcal{G}_f\,\mathbf{c}_0, N\mathcal{G}_f\,\mathbf{c}_1, N\mathcal{G}_f\,\mathbf{c}_2,\ldots) \twoheadrightarrow_\beta (\mathbf{c}_{g(0)}, \mathbf{c}_{g(1)}, \mathbf{c}_{g(2)},\ldots) \equiv \mathcal{G}_g$.

So $R(N\ \mathcal{G}_f) \twoheadrightarrow_\beta \mathcal{G}_g$, so using the composition combinator B we have
$BRN\ \mathcal{G}_f \twoheadrightarrow_\beta \mathcal{G}_g$, hence $BRN \in \Lambda^\circ$ is the desired T.

(3) ⇒ (2) This part concerns the proof 'from infinitary to finitary'.

Given is a T such that $T \in \Lambda^\circ$ $T\mathcal{G}_f \twoheadrightarrow_\beta \mathcal{G}_g$. Now there is a finite 'generator' $G \in \Lambda^\circ(\phi)$ such that in $\lambda^\infty\beta\phi$ we have $G \twoheadrightarrow_\beta \mathcal{G}_f$. Namely: take $G \equiv R\mathbf{f}$, then

$R\phi \twoheadrightarrow_{\beta\phi} (\phi\mathbf{c}_0, \phi\mathbf{c}_1, \phi\mathbf{c}_2,\ldots) \twoheadrightarrow_f (\mathbf{c}_{f(0)}, \mathbf{c}_{f(1)}, \mathbf{c}_{f(2)},\ldots) \equiv \mathcal{G}_f$.

Now we have $T(R\phi) \twoheadrightarrow_{\beta\phi} T(\mathcal{G}_f) \twoheadrightarrow_{\beta\phi} \mathcal{G}_g$. This is a reduction sequence of length ω + ω. Applying Compression for the $\lambda^\infty\beta\phi$-calculus, we have $T(R\phi) \twoheadrightarrow_{\beta\phi} \mathcal{G}_g$ in ω or less (i.e. finitely many) steps. The case of finitely many steps is clearly impossible, since a finite term cannot in finitely many steps ($\twoheadrightarrow_{\beta\phi}$) reduce to an infinite term, being $\mathcal{G}_g$. So we have $T(R\phi) \twoheadrightarrow_{\beta\phi} \mathcal{G}_g$ in ω steps. Now we have $T(R\phi)\mathbf{c}_n \twoheadrightarrow_{\beta\phi} \mathcal{G}_g\ \mathbf{c}_n{>}\twoheadrightarrow_\beta{>}\mathbf{c}_{g(n)}$, which is a reduction of ω + k steps for some k ≥ 0. Again we apply Compression for the $\lambda^\infty\beta\phi$-calculus, yielding $T(R\phi)\mathbf{c}_n \twoheadrightarrow_{\beta f} \mathbf{c}_{g(n)}$, in ω or less steps. But now the case of ω steps is discarded, because no term can reduce in ω steps to a finite term, by Proposition xx. So we have $T(R\phi)\mathbf{c}_n \twoheadrightarrow \beta f\ c_{g(n)}$; but then we are done, and the desired λφ-term is BTRφ, and we have proved (2).
□

2.6. Böhm Trees and Böhm reduction

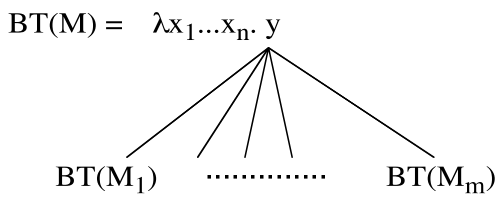
Lambda calculus, as a framework for computability, needs a way to handle the notion of 'undefined' (Barendregt [92]). It is wel-known that the naïve attempt to model 'defined' and 'undefined' by considering all normal forms as defined and all terms without normal form as undefined, fails as equating all terms without normal form yields inconsistency in the sense that all terms then are convertible. The right way is to consider all terms without *head normal form* as undefined, as demonstrated by Wadsworth. One also may consider Barendregt's equivalent notion of *unsolvable* term: a term is unsolvable iff it has no head normal form. This part of the theory belongs to the classics of λ-calculus and we refer for further exposition to Barendregt [84, 92].

The Böhm Tree BT(M) of a λ-term arises by bringing M into its head normal form $\lambda x_1 ... x_n.\ xM_1 ... M_k$ if it exists, and iterating this procedure on the subterms $M_1, ... M_k$, and so on *ad infinitum*. In this way a possibly infinite λΩ-term arises, with Ω's at places in the tree where the head normal form procedure failed. There are three ways to make this sketchy definition precise, as follows. Remarkably, they are very different.

2.6.1. Coinductive definition

The most common way to define BT(M) is as in Barendregt [84]:

BT(M) = Ω if M is unsolvable;



if M has hnf $\lambda x_1 ... x_n.\ yM_1 ... M_m$.

Note that the subterms $M_1, ..., M_m$ may be more complex than M itself—so this is not an *inductive* definition. In fact, the definition is *coinductive*.

2.6.2. Direct approximations and ideal completion

Another frequently used definition of BT(M) is the one in Curien [93], p.127, Def. 1.5.19: Define the *direct approximation* $\omega(M)$ of an Ω-term M by replacing the redex occurrences $(\lambda x.M)N$ by Ω, and apply 'as much as possible' the rules

$$\Omega M \to \Omega$$
$$\lambda x.\, \Omega \to \Omega$$

We will call these two rules the Ω-*normalisation rules*. It is easy to see that these rules are terminating (or SN) and confluent (or CR), hence the end result of applying them as much as possible indeed exists and is uniquely determined. Now the Böhm tree BT(M) of M is defined by

$$BT(M) = \bigcup\{\, \omega(N) \mid M \twoheadrightarrow_\beta N\}$$

Here $\bigcup$ denotes the supremum (lub) in the cpo $(Ter^\infty(\lambda\Omega), \leq_\Omega)$. In general the Böhm trees are infinite terms, in this setting perceived as elements of the ideal completion (see p.378 Curien [93]). All this is classical theory. Mention that the 'stable' part $\omega(N)$ is growing along a reduction, hence by CR every two $\omega(N)$'s have an upper bound (are consistent), hence the whole set has an upperbound hence the lub exists.

2.6.3. Infinitary rewriting

We will introduce BT's somewhat differently, adopting the framework of infinitary lambda calculus (λ^∞-calculus). We generate BT's by the following rules in Table 1.

Böhm Reduction	
$(\lambda x.Z(x))Z' \rightarrow Z(Z')$	(β)
$M \rightarrow \Omega$ if M Ω is unsolvable	(uns)
$\Omega M \rightarrow \Omega$	(Ω_l)
$\lambda x.\ \Omega \rightarrow \Omega$	(Ω_d)

Table 2.1.

typo M different from Omega

We use the abbreviations $I \equiv \lambda x.x$, $\omega \equiv \lambda x.xx$.

2.6.1. EXAMPLE. $(\lambda zy.y(z\omega\omega))I \rightarrow_\beta \lambda y.y(I\omega\omega) \rightarrow_{uns} \lambda y.y\Omega$.

2.6.2. EXAMPLE. See Figure 2.3. The double lines signify that two different reduction steps are possible, e.g.:

$$(\lambda y.\ \Omega\Omega)\Omega \rightarrow_\beta \Omega\Omega \text{ but also } (\lambda y.\ \Omega\Omega)\Omega \rightarrow_{uns} \Omega\Omega.$$

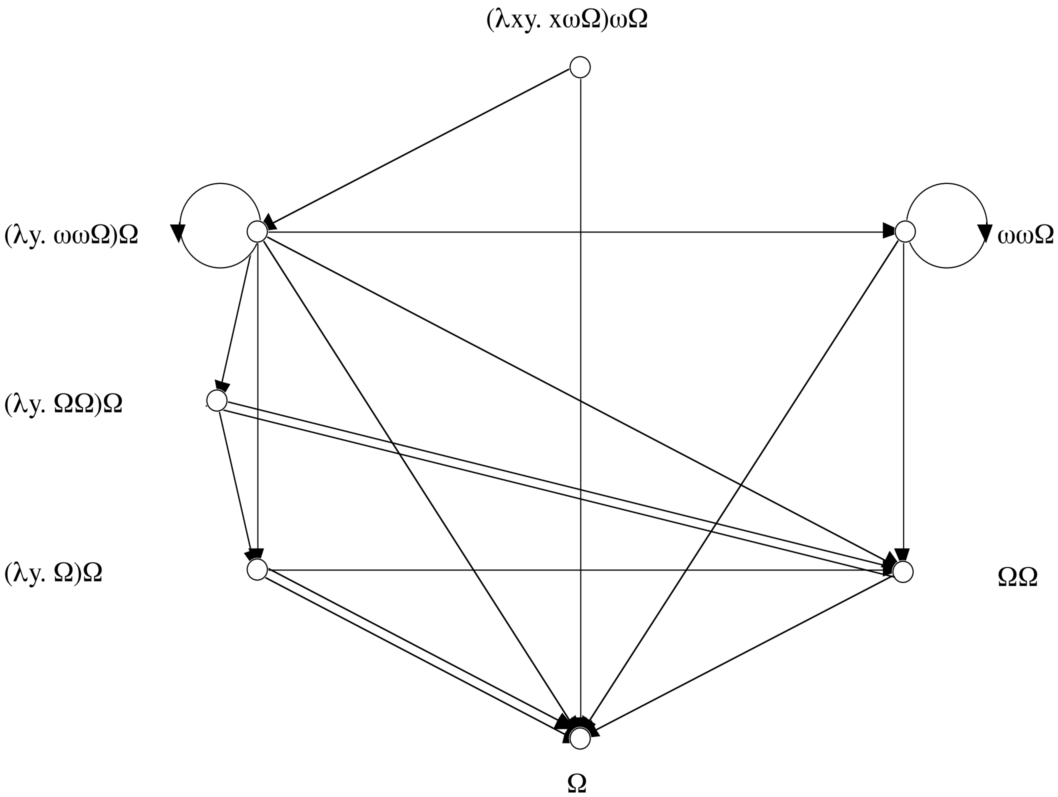


Figure 2.3. Böhm reduction graph of (λxy. xωΩ) ωΩ.

2.6.3. Remark.
1. The proviso $M \not\equiv \Omega$ in the rule *uns* serves to prevent the useless rewriting step $\Omega \to \Omega$.
2. The two Ω-normalisation rules (Ω_l) and (Ω_d) (Ω-left and Ω-down) are in fact superfluous as they are instances of the *uns* rule. Yet we include them, to facilitate comparison with the *labeled* Böhm reduction system in the sequel.
3. Overlap

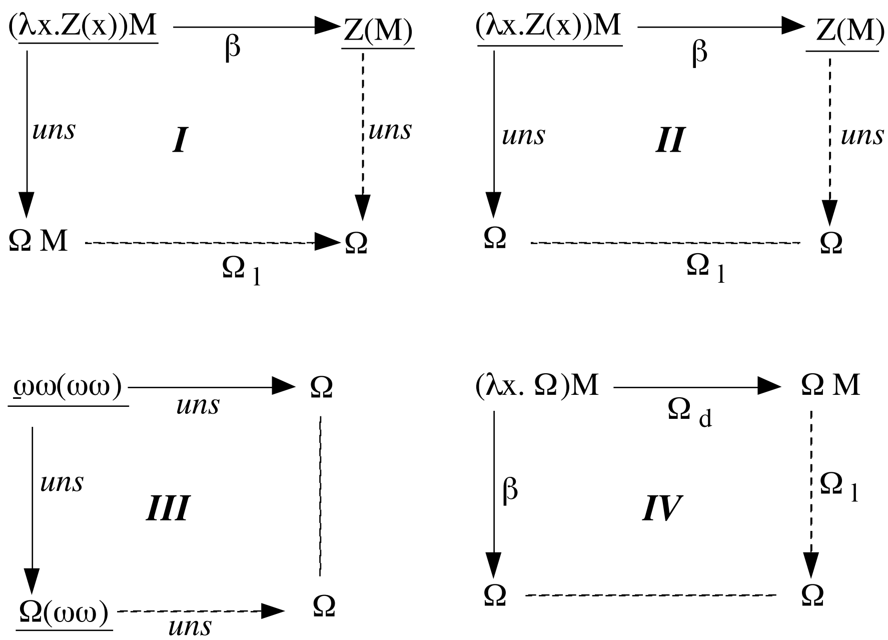


Figure 2.4. Overlaps in Böhm reduction rules. error top right?

2.6.4. THEOREM. *Böhm reduction is confluent.*

PROOF. A proof of this fact can be found in Barendregt [84], p.384-391, leading to Theorem 15.2.15 (i) stating that βΩ is CR, in the notation used there. That proof includes η-*reduction* which is not considered in this chapter. Barendregt's proof uses "Postponement of Ω-reductions". Another proof, using "Finite Developments", is in Barendregt, Bergstra, Klop, and Volken [1976).

In the sequel we will need the following standard facts. For the proof see e.g. Barendregt [84].

2.6.5. PROPOSITION.

1. $M \twoheadrightarrow_\beta N \Rightarrow BT(M) = BT(N)$
2. $M \leq_\Omega N \ \Rightarrow BT(M) \leq_\Omega BT(N)$

2.6.6. Remark. Now every reduction of this nature will deliver after a possibly transfinite sequence of α steps, where α is an ordinal, a possibly infinite tree: the

Böhm tree. We can view this fact as an *infinitary strong normalization* result, with the Böhm trees as possibly infinite normal forms. Also their unicity (i.e. independence of the actual reduction leading to them) is guaranteed, but we will not spell out the details here.

In fact one might want the Böhm tree to be reached as a limit after at most ω steps, instead of after some $\alpha > \omega$. In that case we have to impose a fairness assumption to the effect that no redex will be infinitely often 'neglected'. This is what we will do in the present note.

2.6.7. NOTATION.
1. Instead of BT(M) = BT(N) we will also write $M =_{BT} N$.
2. Let σ be an occurrence in BT(M) (written in the applicative notation, see Remark 4.5). Then BT(M)/σ is the *symbol* at that location, so one of @, λx, x, or Ω.
3. BT(M)//σ is the *subtree* at that location. Note that BT(M)/σ and BT(M)//σ coincide when one of them equals a variable x, or the constant Ω.

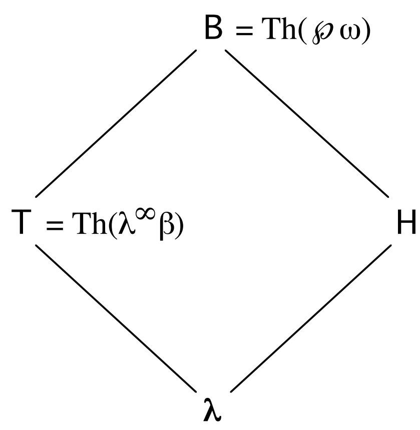


Figure 2.5. Lambda theories.

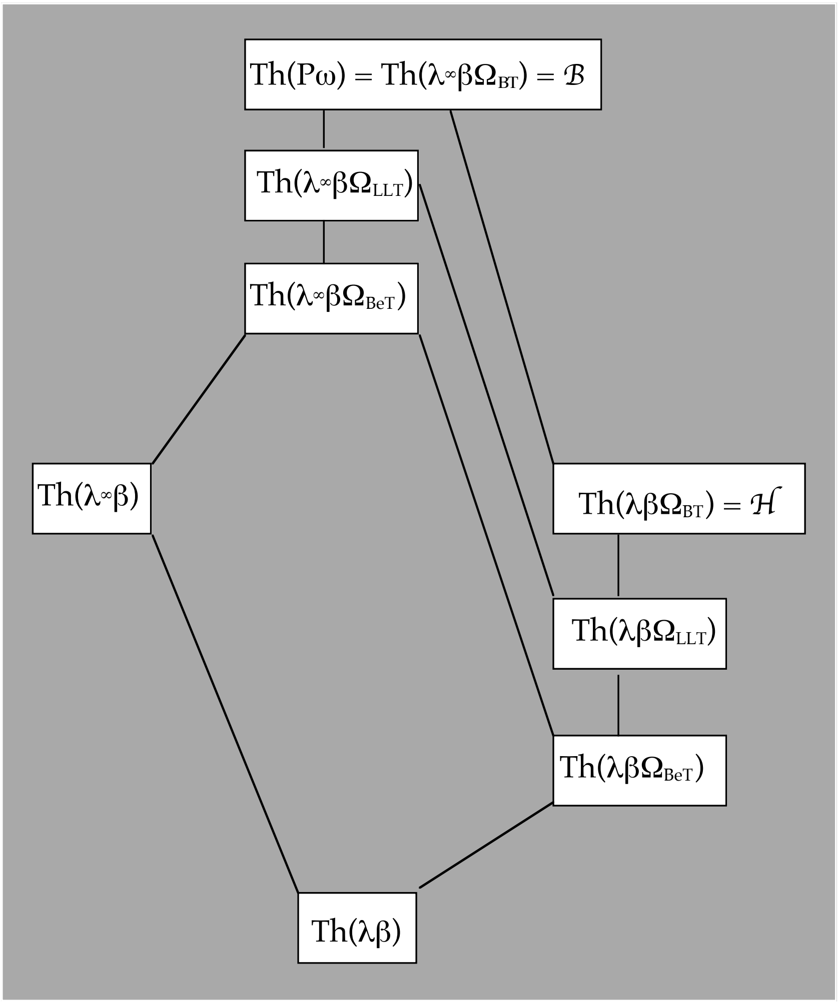


Figure 2.6. Lambda theories

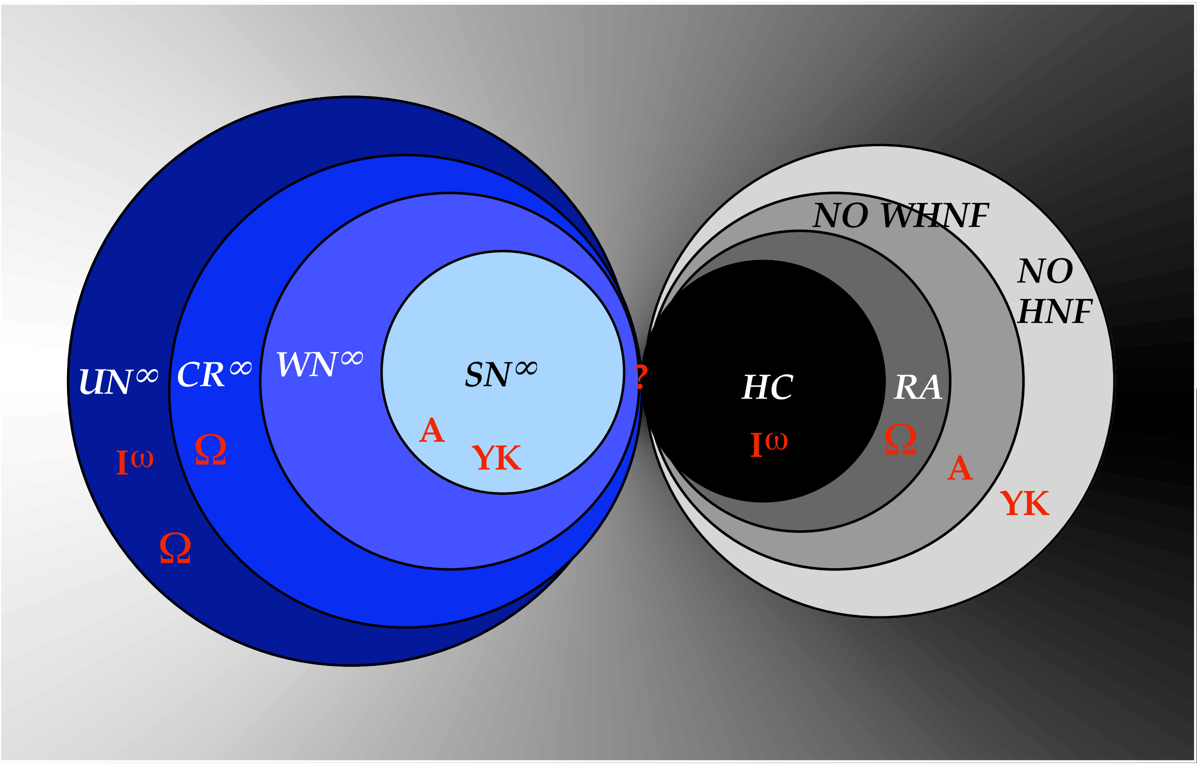


Figure 2.7. Attempt at classification

2.7. Berry's Sequentiality Theorem

We start with some experiments.

2.7.1. EXAMPLE. Consider the following BT-evaluation:

$$(\lambda xy.\ x\Omega)\Omega\ \rightarrow$$
$$\lambda y.\ \Omega\Omega\ \rightarrow$$
$$\lambda y.\ \Omega\ \rightarrow$$
$$\Omega.$$

We want to view this constellation of initial term and final term, related by the BT-computation, as the input and output interface of a device as in the following figure. More specifically, the Ω's in the initial term are the input channels, and the final Ω is the output channel. The processing activity of the device is the BT-evaluation. Feeding data in the input channels means refining these Ω's in the sense of $\leq_\Omega$, after which we may or may not observe an output at the output channel Ω.

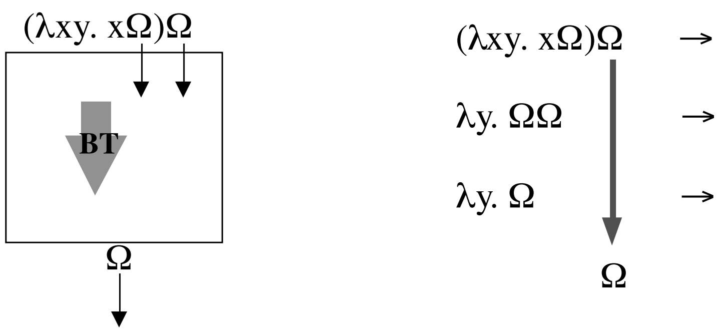


Figure 2.8. Input-output behaviour and causal dependence

In this example an arbitrary input Q in the first Ω has no output effect:
$(\lambda xy.\ xQ)\Omega\ \rightarrow \lambda y.\ \Omega Q\ \rightarrow \lambda y.\ \Omega\ \rightarrow \Omega.$
but with input z in the second Ω we do have output:
$(\lambda xy.\ x\Omega)z\ \rightarrow \lambda y.\ z\Omega$. So the output is $\lambda y.\ z\Omega$.

2.7.2. EXAMPLE.

$$(\lambda xy.\ x\omega\Omega)\omega\Omega\ \rightarrow$$
$$(\lambda y.\ \omega\omega\Omega)\Omega\ \rightarrow$$
$$(\lambda y.\ \Omega\Omega)\Omega\ \rightarrow$$

$(\lambda y.\ \Omega)\Omega \rightarrow$
Ω.

In this example there is no way to yield output at the output Ω; for, refine both Ω's to P, Q respectively:

$(\lambda xy.\ x\omega P)\omega Q \ \rightarrow$
$(\lambda y.\ \omega\omega P)Q \ \rightarrow$
$(\lambda y.\ \Omega P)Q \ \rightarrow$
$(\lambda y.\ \Omega)Q \rightarrow$
Ω.

2.7.3. EXAMPLE. In the two examples above there was only one output Ω. In the next example there are three output Ω's. Let $M \equiv \lambda x.\ x(\omega\omega)\Omega\Omega$. Then $BT(M) = \Omega\Omega\Omega$. The first Ω is independent of refinements of M; the second Ω and third Ω are caused by the Ω 's in M in the order of appearance. This example already gives an intuition of why BST holds: the Ω's in the output that have no cause are the ones that are *'created'* during the BT computation, while the output Ω's that have a cause are just the ones that in some sense are *descendants* of the original input Ω's. We will make this intuition precise in the sequel (not yet included) by giving a proof of BST based on 'origin tracking'.

2.7.4. EXAMPLE. In $(\lambda xy.y(xx)\omega \rightarrow \lambda y.y(\omega\omega)$ the unsolvable $\omega\omega$ is created.

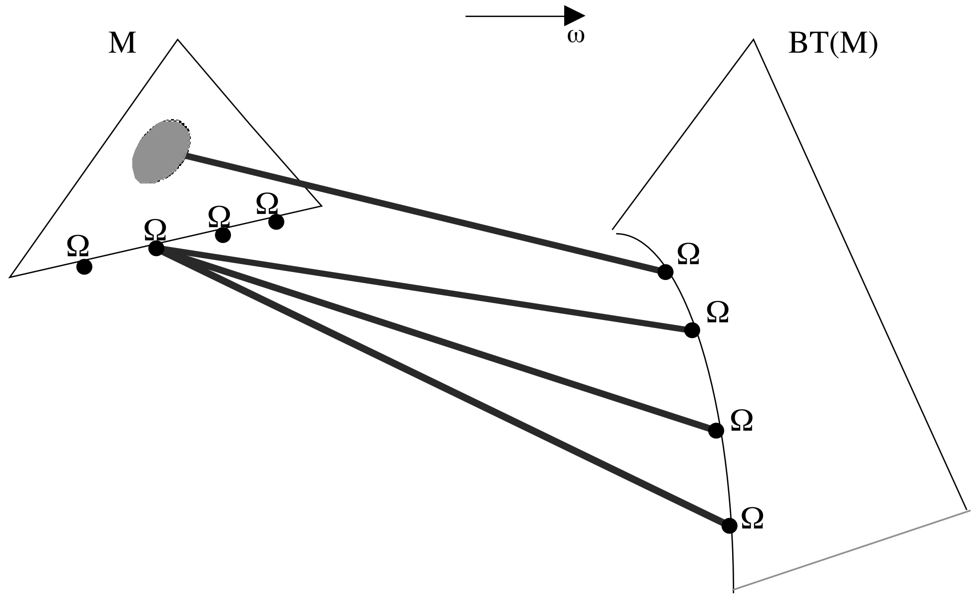


Figure 2.9. Causal dependence of output Ω's to input Ω's.

2.7.5. DEFINITION. 1. Let M ∈ Ter(λΩ). We write M ≡ C[Ω,...,Ω], where all n occurrences of Ω are displayed and C[,...,] is a context with n 'holes'. For convenience we will number the Ω's in M: $M \equiv C[\Omega_1,...,\Omega_n]$. Let Ω be an occurrence in BT(M) at position σ, so BT(M)/σ = Ω. Then this Ω is *insensitive* for refinements of the initial Ω's in M if

$$\forall M' \geq_\Omega M \;\; BT(M')/\sigma = \Omega.$$

We will say for short that this Ω is insensitive, without more.

2. The Ω is *caused by* Ω_i in M if it is properly increased when Ω_i in M is refined to a fresh variable z, and moreover is insensitive for increases at the other Ω's in M. So:

$BT(C[Q_1,...Q_{i-1}, \Omega, Q_{i+1}, ..., Q_n])/\sigma = \Omega$, and
$BT(C[Q_1,...Q_{i-1}, z, Q_{i+1}, ..., Q_n])/\sigma \neq \Omega$.

2.7.6. THEOREM. (*BST, Berry Sequentiality Theorem*). *Let* M ∈ Ter(λΩ). *Then an* Ω *in* BT(M) *is either insensitive, or caused by a unique* Ω *in* M.

2.7.7. REMARK. 1. In Fig.2.9 this situation is depicted. Note that an input Ω may be the cause of several (even infinitely many) output Ω's. But never will one output Ω be caused by more than one input Ω. The fuzzy area in Fig. 2.9 suggests that the Ω connected to it is somehow created by the material in the fuzzy area—but a precisely located cause cannot be given.
2. Actually, instead of a fresh variable in Case 2 of Definition 2.7.5 above any proper increase of Ω_i in M will do; proper meaning unequal to Ω, i.e. having a head normal form.
3. Note that the unicity of the 'index' i of the cause Ω_i is implied. For suppose there were two causes Ω_i and Ω_j with $i \neq j$. Then putting a variable z at Ω_i will not change the output at σ, since Ω_j is its cause; on the other hand it will increase the output at σ properly, since Ω_i is its cause.
4. Our treatment differs from the one in Barendregt [84]; there definitions are based on "refinement" modulo BT equality. Another point is that in Barendregt's proof of BST the notion of *stability* and the stability theorem is used; we would like to separate these issues.
5. Curien [93] contains a proof of BST in the framework of concrete data structures, see p.205.

2.8. Applications of BST

Before embarking on a proof of BST let us consider an important application: establishing nondefinability results.

We first consider a corollary of BST as given in Barendregt [84] (Thm. 14.4.12, p.376). We follow the proof given there. The theorem can (according to Barendregt [84]) be paraphrased as follows: if an n-ary context is constant modulo BT on n "perpendicular lines" it is constant everywhere. (See Fig. 2.10.)

2.8.1. THEOREM. (*Perpendicular lines, Barendregt*).

Let $M_{ij}, N_i \in \mathrm{Ter}^\infty(\lambda\Omega)$ *for* $i,j \in \{1,\dots,n\}$. *Let* $C[,\dots,]$ *be an n-ary context in* $\mathrm{Ter}^\infty(\lambda\Omega)$ *and suppose for all* $Z \in \mathrm{Ter}^\infty(\lambda\Omega)$:

$$C[M_{11}, M_{12},\dots, M_{1\,n-1}, Z] =_{BT} N_1$$
$$C[M_{21}, M_{22},\dots, z, M_{2n}] =_{BT} N_2$$

$$\begin{array}{ll} \ldots & \ldots \\ C[Z, M_{n2}, \ldots, M_{n\,n-1}, M_{nn}] & =_{BT} N_n \end{array}$$

Then for all $\mathbf{Z} = Z_1,\ldots,Z_n \in \mathrm{Ter}^\infty(\lambda\Omega)$ *we have*

$$C[\mathbf{Z}] =_{BT} N_1 =_{BT} \ldots =_{BT} N_n.$$

PROOF. We present the proof in Barendregt [84] in the present notation. The proof is given for n = 3, but the proof for general n ≥ 2 is analogous. (For n = 1 the theorem is trivial.) So assume the equations

(1) $C[M_{11}, M_{12}, Z] =_{BT} N_1$
(2) $C[M_{21}, Z, M_{23}] =_{BT} N_2$
(3) $C[Z, M_{32}, M_{33}] =_{BT} N_3$

Let $N \equiv C[\Omega, \Omega, \Omega] \equiv C[\Omega_1, \Omega_2, \Omega_3]$. Then $BT(N) \leq_\Omega BT(N_i)$, i = 1,2,3. Suppose one of these $\leq_\Omega$ is strict: $BT(N) <_\Omega BT(N_i)$. Then for some α:

(4) $BT(N)/\alpha = \Omega$ and
(5) $BT(N_i)/\alpha \neq \Omega$.

So the Ω in BT(N) at α is not constant. By BST it must have a cause in N, say Ω_3. But then, because it is a refinement at non-causing Ω's:

(6) $BT(C[M_{11}, M_{12}, \Omega])/\alpha = \Omega$, and
(7) $BT(C[M_{11}, M_{12}, z])/\alpha \neq \Omega$. because of BST.
(8) $BT(C[M_{11}, M_{12}, \Omega]) = BT(N_1)$, by (1).
(9) $BT(C[M_{11}, M_{12}, z]) = BT(N_1)$, also by (1).
(10) $BT(N_1)/\alpha \neq \Omega$ by (7, 9).
(11) $BT(N_1)/\alpha = \Omega$, by (8, 6) a contradiction with (10). Conclusion:
(12) $BT(N) = BT(N_i)$, i = 1,2,3.

Next, suppose $C[\square, \square, \square]$ is not constant "modulo BT". So for some M_1, M_2, M_3, we have, putting $N' \equiv C[M_1, M_2, M_3]$,

$$BT(N') \equiv BT(C[M_1, M_2, M_3]) >_\Omega BT(N).$$

Then for some α,

(13) $BT(N)/\alpha = \Omega$ and
(14) $BT(N')/\alpha \neq \Omega$.

So the Ω at α in BT(N) is not constant, hence by BST it is caused by say the Ω_2 in $N \equiv C[\Omega_1, \Omega_2, \Omega_3]$. Then

(15) $BT(C[M_{21}, z, M_{23}]) = BT(N_2)$ by (2), and
(16) $BT(N) = BT(N_2)$ by (12) ,
(17) $BT(C[M_{21}, z, M_{23}])/\alpha \neq \Omega$ by BST, so
(18) $BT(N)/\alpha \neq \Omega$, by (15, 16, 17),
contradicting (13). ❑

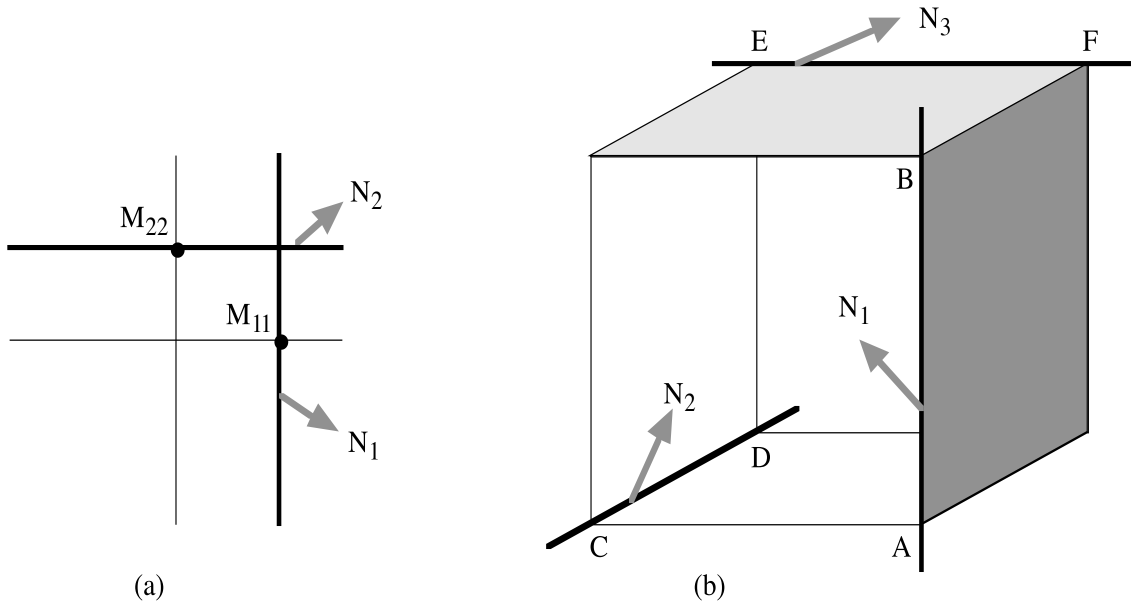


Figure 2.10. Perpendicular lines theorem.

2.8.2. EXAMPLE. Figure 2.10 depicts the cases for n = 2 and n = 3 of the Perpendicular Lines Theorem. In Fig. 2.10(a), for n = 2: $C[M_{11}, Z] = N_1$ and $C[Z, M_{22}] = N_2$. In Fig. 2.10(b), for n = 3:

$A = (M_{11}, M_{12}, M_{23})$, $B = (M_{11}, M_{12}, M_{33})$. The line through A, B is constant N_1.

$C = (M_{21}, M_{12}, M_{23})$, $D = (M_{21}, M_{32}, M_{23})$. The line through C, D is constant N_2.

$E = (M_{21}, M_{32}, M_{33})$, $F = (M_{11}, M_{32}, M_{33})$. The line through E, F is constant N_3.

An important corollary of BST is the *stability* theorem.

2.8.3. THEOREM. *Let* $C[\ ]$ *be a* $Ter^{\infty}(\lambda\Omega)$*-context. Let* $M, N \in Ter^{\infty}(\lambda\Omega)$ *be compatible:* $M \uparrow N$. *Then:* $BT(C[M \cap N]) = BT(C[M]) \cap BT(C[N])$.

PROOF. We have $M \cap N \leq_\Omega M, N$, hence $C[M \cap N] \leq_\Omega C[M], C[N]$, and by monotonicity $BT(C[M \cap N]) \leq_\Omega BT(C[M]),\ BT(C[N])$. So
$BT(C[M \cap N]) \leq_\Omega BT(C[M]) \cap BT(C[N])$.
So we need only to prove that this $\leq_\Omega$ is not strict. Suppose it is:

$BT(C[M \cap N]) <_\Omega BT(C[M]) \cap BT(C[N])$.
Then there is a position σ such that

$BT(C[M \cap N])/\sigma = \Omega$, and $(BT(C[M]) \cap BT(C[N]))/\sigma \neq \Omega$. Hence

$(BT(C[M])/\sigma \neq \Omega$ and $(BT(C[N])/\sigma \neq \Omega$. (*)

CLAIM. $(BT(C[M])$ *is not* σ*-constant*.

Proof of the claim. $M \uparrow N$, so for some Q: $Q \geq_\Omega M, N$, hence
$C[Q] \geq_\Omega C[M], C[N]$, and by monotonicity $BT(C[Q]) \geq_\Omega BT(C[M]),\ BT(C[N])$. So

$BT(C[Q]) \geq_\Omega BT(C[M]) \cap BT(C[N])$.

So $BT(C[Q])/\sigma \neq \Omega$. So indeed there is a refinement of $C[M \cap N]$, namely $C[Q]$, that does yield output at σ. ❑ $_{\text{Claim}}$

So by BST, the Ω at σ in $BT(C[M \cap N])$ is caused by an Ω in $C[M \cap N]$, say at position β:

$(C[M \cap N])/\beta = \Omega$.

Now there are three possibilities (see Figure 2.11.)

1. The Ω is cut off in the intersection $C[M \cap N])$ from M, or rather $C[M]$. In the Figure this is Ω_1. More precisely said: $C[M]/\beta \neq \Omega$. Then $C[N]$ is a refinement of $C[M \cap N]$ on other Ω's than the causing Ω at β, so $(BT(C[N])/\sigma = \Omega$, contradiction with (*).

2. The Ω is cut off in the intersection $C[M \cap N])$ from N, or rather $C[N]$. In the Figure this is Ω_4. More precisely said: $C[N]/\beta \neq \Omega$. Then $C[M]$ is a refinement of $C[M \cap N]$ on other Ω's than the causing Ω at β, so $(BT(C[M])/\sigma = \Omega$, contradiction with (*).

3. The Ω is present in both C[M and C[N. In the Figure, this is Ω_2 or Ω_3. So $C[M]/\beta = C[N]/\beta = \Omega$. Then $C[M]$, $C[N]$ both are a refinement of $C[M \cap N]$ on other Ω's than the causing Ω at β, so $(BT(C[M])/\sigma = \Omega$ and $(BT(C[N])/\sigma = \Omega$, again a contradiction. ❑

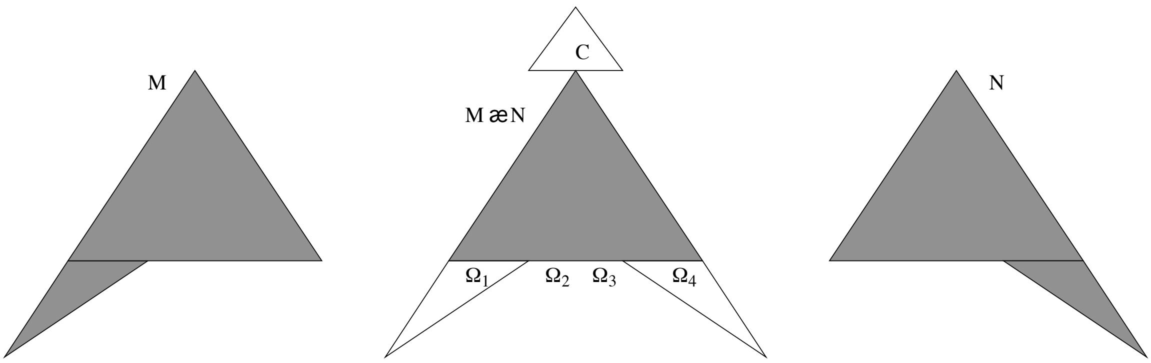


Figure 2.11. Intersection of C[M] and C[N].

2.8.4. Surjective pairing

Surjective Pairing (SP) is an extension of lambda calculus that has received considerable attention in the past. It consists of a pairing operator D and unpairing operators D_0 and D_1, satisfying the equations

$D_0(DXY) = X$

$D_1(DXY) = Y$

$D(D_0X)(D_1X) = X$

for all terms X, Y. There are the following important facts about SP:

1. SP *is not definable in λ-calculus*; this was first proved in Barendregt []. The proof was by means of an underlining technique that may be seen as a precursor of the tracing technique employed in the present paper. (In the setting of Barendregt [] only finite λ-terms are considered, though.)

2. *λ-calculus plus SP as rewrite rules is not confluent*; see Klop [80], Klop & de Vrijer [].

3. *SP is conservative over λ-calculus*; see De Vrijer [].

Here we revisit fact 1. Remarkably, the non-definability of SP is a straightforward corollary of BST (Theorem 2.7.6).

2.8.5. THEOREM. (*Barendregt*) *Surjective Pairing is not definable in λ-calculus.*

PROOF.
$D(D_0\Omega)(D_1\ \Omega) =_\beta \Omega$

so $D(D_0\Omega)(D_1\ \Omega) =_{BT} \Omega$.

The Ω in the right-hand side is not constant, since e.g. $D(D_0I)(D_1I) = I$. So it must be caused by say the second Ω in the left-hand side. So refining the first Ω to DXY still yields Ω as output. Now we have for all X, Y:

$D(D_0(DXY))(D_1\ \Omega) = \Omega$

$DX(D_1\ \Omega) = \Omega$

$D_1(DX(D_1\ \Omega)) = D_1\Omega$

$X = D_1\Omega$.

Hence for all X, Y: X = Y, a contradiction. It follows that SP is not λ-definable. ❑

2.8.6. Parallel-or.

Consider the equations as in the proof above:

$$\begin{array}{ll} PXT = T & (1) \\ PTX = T & (2) \end{array}$$

We would like to find λ-terms P and T (true) satisfying these equations; X is a metavariable. This problem is trivially solved: take P = λxy.T.

However, the next version of parallel-or

$$\begin{array}{l} PXT = T \\ PTX = T \\ PFF = F \end{array}$$

is not definable in λ-calculus.

Proof. By Theorem 2.8.1 we would have BT(PXY) = BT(T) for all X, Y. So BT(PFF) = BT(F) = BT(T), contradiction. ❑

2.8.7. *Remark.* 1. In Curien [93]p.195 parallel or is rendered as follows:

por(x,y) = T iff x = T or y = T
por(x,y) = F iff x = y = F

2. In Mitchell [96] p.114 parallel or is rendered as follows:

$$\begin{array}{lll} \text{porMN} \twoheadrightarrow T & \text{if } M \twoheadrightarrow T \text{ or } N \twoheadrightarrow T \\ & \text{if } M \twoheadrightarrow F \text{ and } N \twoheadrightarrow F \\ & \text{no normal form otherwise.} \end{array}$$

3. In Barendregt [84], referring to Plotkin [77], parallel or is a term F such that FMN = I if M or N is solvable, and unsolvable else.

2.8.8. Remark. In Theorem 2.8.1 the terms in the right-hand sides N_i may not contain the meta-variable Z. For, consider

$$KZT =_{BT} Z$$
$$KTZ =_{BT} T$$

This is indeed of the form prescribed by the would-be theorem:

$$C[T, Z] =_{BT} M_1[Z]$$

$$C[Z, T] =_{BT} M_2$$

But then the conclusion is: for all Z, Z' $C[Z, Z'] =_{BT} M_1 =_{BT} M_2$, in particular for all Z, $Z =_{BT} T$, a contradiction.

(See also CURIEN P.330, 332, 333 FOR PIF)

2.8.9. todo. *The Berry-Kleene function*
Berry's function (also called Gustave's function, or Berry-Kleene function (Curien [93, p.202.)

2.9. 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 $\pi_0 xy = x$; $\pi_1 xy = y$; $\pi(\pi_0 x)(\pi_0 x) = x$. It turns out that not only definability is problematic for these reduction rules, but also the (finitary) confluence property for the extension of $\lambda\beta$-calculus with these rules. Turning these equations into the reduction rules $\pi_0 xy \to x$; $\pi_1 xy \to y$; $\pi(\pi_0 x)(\pi_0 x) \to x$ yields a non left-linear system, due to the repetition of the variable x in the lefthandside 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 $\lambda\beta$-calculus such that the resulting system is CR. In Klop [80] it was shown that the addition yields nonconfluence, 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.

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

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

2.9.2. Proposition. *The reduction relation* $\beta\delta_S$ *is not CR. By a fixed point construction there are terms* $C, A \in \Lambda(\delta,\varepsilon)$ *such that*

$$Cx \twoheadrightarrow_\beta \delta x(Cx),$$
$$A \twoheadrightarrow_\beta CA.$$

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

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

$$\begin{array}{l} A \twoheadrightarrow CA \twoheadrightarrow_\beta \delta A(CA) \twoheadrightarrow_\beta \delta(CA)(CA) \to_{\delta_S} \varepsilon \\ \qquad\;\; \downarrow \\ \qquad\;\; C\varepsilon \end{array}$$

The three terms CA, Cε and ε form a counterexample against the CR property. In this case it is easily proved that Cε 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 ⊥-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 2.28 for the notion of BT for terms in $\Lambda(\delta,\varepsilon)$, turn out to be infinite regular trees. Employing the μ-notation as in Example 2.8 they are as follows.

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

A slightly more difficult extension is the following.

2.9.3. DEFINITION. (*Klop [1980]*) The reduction relation δ_K is defined on $\Lambda(\delta,\varepsilon)$ by

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

2.9.4. PROPOSITION. *The reduction relation* $\beta\delta_K$ *is not CR.*

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

$$\begin{array}{lllll} A \twoheadrightarrow & CA & \twoheadrightarrow \delta A(CA) & \twoheadrightarrow \delta(CA)(CA) & \to_{\delta_K} \varepsilon(CA) \\ & \big\downarrow & & & \\ & C(\varepsilon(CA)) & & & \end{array}$$

Now it is a bit more laborious to show that ε(CA) and C(ε(CA)) have no common reduct, which was done in Klop [1980] using finitary arguments. The infinitary argumentation employs the BT's of the three relevant terms CA, ε(CA) and C(ε(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. □

2.9.5. REMARK. For Propositions 2.9.2 and 2.9.4 the situation is:

$$M \to_\delta M' \Rightarrow BT(M) \to_\delta^{\leq\omega} BT(M').$$

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

2.9.6. DEFINITION. (i) An occurrence of δ is called *balanced* if it is the head of a δ-redex δMM, with $M \in \Lambda^\infty(\delta,\varepsilon)$.
(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 $\pi(\pi_0 x)\,(\pi_1 x)$, with $M \in \Lambda^\infty(\pi,\pi_0,\pi_1)$.

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

2.9.7. DEFINITION. (*J.R. Hindley*) The reduction relation δ_H is defined on $\Lambda(\delta,\varepsilon)$ 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. $\delta_H III \to_{\delta_H} II$. For Surjective Pairing the same holds.

2.9.8. PROPOSITION. *The reduction relation* $\beta\delta_H$ *is not CR. By a fixed point construction there are terms C,A such that*

$$Cx \twoheadrightarrow \varepsilon(\delta x(Cx))$$
$$A \twoheadrightarrow CA.$$

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

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

The BTs of the relevant trio of terms CA, ε(CA), C(ε(CA)) are respectively the trees μx. ε(δxx) ≡ T, εT and μx. ε(δ(εT)x). The corresponding cyclic graphs are drawn in the lower plane in Figure 2.12.

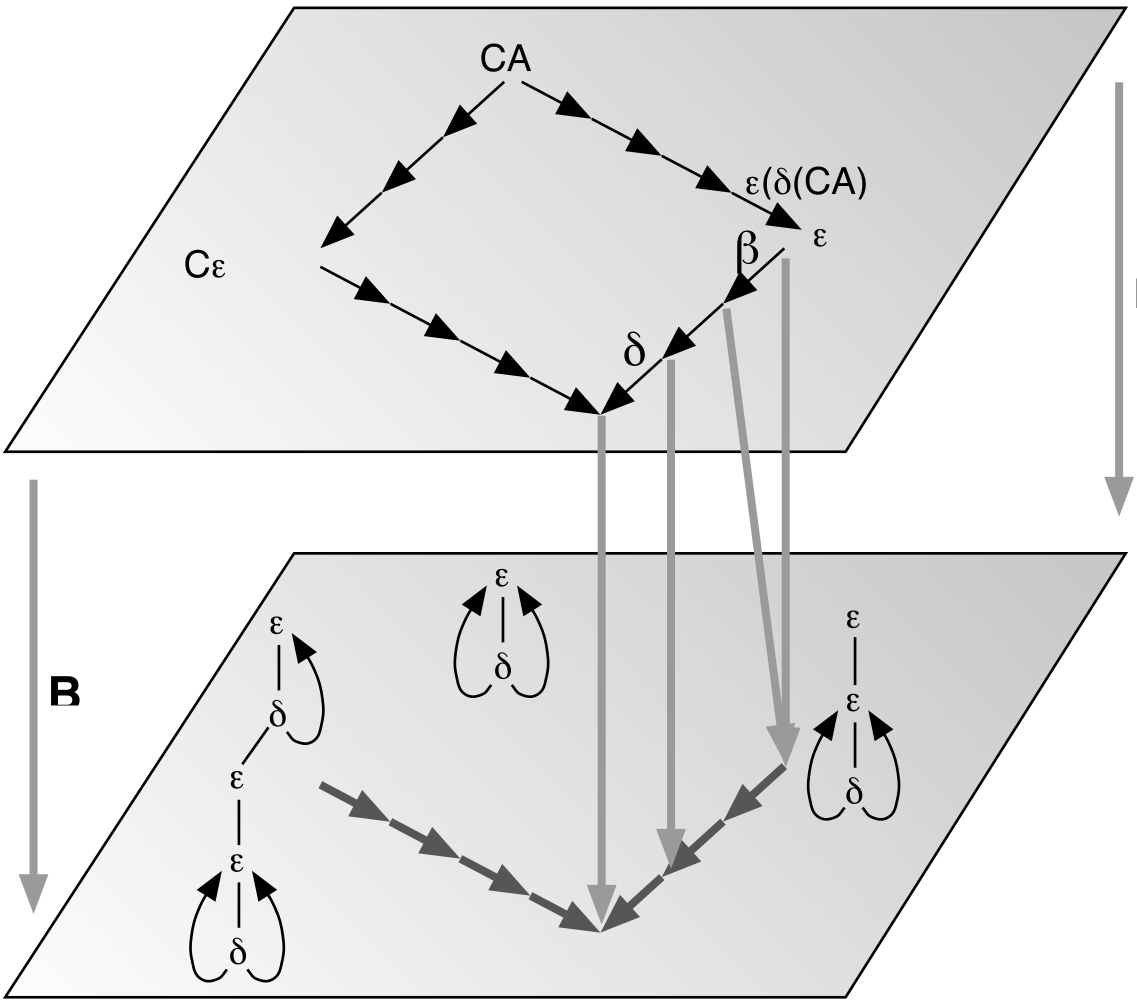


Figure 2.12: Projection by BT

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

Now we will prove that indeed (ε(CA)) and C(ε(CA)) have no common reduct, by an excursion to the infinitary setting, depicted in Figure 2.12. 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 $\to_\delta^{\leq\omega}$. 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 $\gamma x \to Ix$ where $I \equiv \lambda x.x$. We will call these rules $(\delta\gamma)$ and (γI) respectively, to be read as 'δ to γ' and 'γ to I'. The δ-step $\delta MM \to M$ is now splitted in three:

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

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

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

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

(Note that this is a strongly convergent reduction.)

We now have the situation as in Figure 2.13, corresponding to the following.

(1) $M_0 \to_{\delta\gamma} M_1 \;\Rightarrow\; BT(M_0) \to_{\delta\gamma}{}^{\leq\omega} BT(M_1)$;
(2) $M_1 \to_{\gamma I} M_2 \;\Rightarrow\; BT(M_1) \to_{\gamma I}{}^{\leq\omega} P$;
(3) $M_0 \to_\delta M_3 \;\Rightarrow\; BT(M_0) \to_{\delta\gamma}{}^{\leq\omega} \to_{\gamma I}{}^{\leq\omega} \twoheadrightarrow_{\beta\Omega} BT(M_3)$.

$$\begin{tikzcd}
M_0 \equiv C[\delta M M] \arrow[r] \arrow[d, "\delta\gamma"'] & \mathrm{BT}(M_0) \arrow[d, "\delta\gamma"', "\leqslant\omega"] \\
M_1 \equiv C[\gamma M] \arrow[r] \arrow[d, "\gamma\mathsf{I}"'] & \mathrm{BT}(M_1) \arrow[d, "\gamma\mathsf{I}"', "\leqslant\omega"] \\
M_2 \equiv C[\mathsf{I} M] \arrow[r] \arrow[d, "\boldsymbol{\beta}"'] & P \arrow[d, "\boldsymbol{\beta\Omega}"', "\leqslant\omega"] \\
M_3 \equiv C[M] \arrow[r] & \mathrm{BT}(M_3)
\end{tikzcd}$$

Figure 6: Fine-structure of δ-parallel moves

Figure 2.13: Fine-structure of δ-parallel moves

As to (1): a δ-redex in M_0 is preserved as (possibly infinitely many) δ-redexes in $BT(M_0)$: 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, $BT(M_0)$, 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 ⊥, to obtain a BT.

Now we can conclude. Consider the infinite terms εT and μx. ε(δ(εT)x), with T ≡ μx. ε(δxx), to be made confluent in the infinite plane, where we have to employ 'macro steps' steps like:

$$\rightarrow_{\delta\gamma}{}^{\leq\omega} \rightarrow_{\gamma I}{}^{\leq\omega} \twoheadrightarrow_{\beta\Omega} .$$

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.

2.9.10. THEOREM. *The reduction relation* βSP *on* $\Lambda(\pi,\pi_1,\pi_2)$

$\pi_0 xy \to x;\ \pi_1 xy \to y;\ \pi(\pi_0 x)(\pi_0 x) \to x$

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

$$Cx \twoheadrightarrow_\beta \varepsilon(\pi((\pi_0 x)(\pi_1(Cx))),$$

$$A \twoheadrightarrow_\beta CA.$$

Then

$$\begin{array}{llll} A \twoheadrightarrow & CA & \twoheadrightarrow \varepsilon(\pi(\pi_0 A)(\pi_1(CA))) \twoheadrightarrow \varepsilon(\pi(\pi_0\,(CA))(\pi_1(CA))) \to_{SP} \varepsilon(CA) \\ & \downarrow_{SP} & \\ & C(\varepsilon(CA)) & \end{array}$$

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

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

$$BT(CA)) \equiv \mu x.\ \varepsilon(\pi((\pi_0 x)(\pi_1 x))) \equiv S$$
$$BT(\varepsilon(CA)) \equiv \varepsilon S$$
$$BT(C(\varepsilon(CA))) \equiv \mu x.\ \varepsilon(\pi(\pi_0(\varepsilon S)(\pi_1 x)))$$

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. □

2.10. Exercises

2.10.1. **EXERCISE.** Barendregt [84], Section 18.3, p.486 introduces the λ-calculus model $\mathfrak{B}$, where BT's can also be applied to one another directly, just as in λ^∞-calculus. The way used for the construction of $\mathfrak{B}$ is via approximations of the BT's. The construction is due to Barendregt. In some sense the model $\mathfrak{B}$ is a precursor of λ^∞-calculus. Investigate more precisely the relationship between $\mathfrak{B}$ and the model obtained by λ^∞-calculus.

2.10.2. **EXERCISE.** Prove the Compression property for $\lambda^\infty\beta\Omega$-calculus.

2.10.3. **EXERCISE.** (J.R. Kennaway, F.J. de Vries, P. Severi, M.R. Sleep)
Prove that in $\lambda^\infty\beta$ all root active terms are convertible to each other, in particular to Ω and I^ω.
Solution. In their paper in Springer LNCS 3838.

2.10.4. **EXERCISE.** (H.P. Barendregt).
(i) In λ-calculus there are pure n-cycles for every $n \geq 1$. Prove this.
(ii) **NOTE.** In CL this is different; there are no pure reduction cycles. In fact CL has the property, at least when it is based on the familiar combinators S,K,I, that a reduction graph containing a cycle, must be infinite. Or otherwise said: In CL, a finite reduction graph contains no cycles.
See also Exercise 1.4.16 in Chapter 1.

2.10.5. **EXERCISE.** Prove BST for CL.

2.10.6. **EXERCISE.** *Let* U *be unsolvable. Let* C[ ] *be a context in* $Ter^\infty(\lambda\Omega)$. *Then:*
C[U] *is unsolvable* $\Leftrightarrow$ C[Ω] *is unsolvable.*

Solution. ($\Rightarrow$) BT(C[U]) = BT(C[BT(U)]) = BT(C[Ω]) = Ω.

($\Leftarrow$) BT(C[Ω]) = BT(C[BT(U)]) = BT(C[U]) = Ω.

2.10.7. **EXERCISE.** Investigate the precise relation of Scott's Induction Rule (SIR), that we encountered in Remark xx, to the present infinitary setting. Is it true that infinitary λ-conversion $\twoheadrightarrow_\beta$, includes all consequences of SIR?

2.10.8. **EXERCISE.** 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, causes a dramatic speed-up in evaluation. It would be interesting to pursue studies of term graph rewriting against the back-ground of infinitary λ-calculus, as a continuation of work by Kennaway et al [], and Ariola-Klop [], where this theme was studied with reference to infinitary first order rewriting.

2.10.9. **EXERCISE.** Extend the result in Section 2.5 on relative computability from total functions to partial functions.

Todo note. Unsolvables. Fibres, stable and unstable.
Todo note. Three semantics (the eight-fold path), BT, LT, BeT, theorem of de Vries on

corresponding deductive systems.

2.10.10. **EXERCISE.** The correctness of the classification attempt in Figure 2.7 is not clear. Jeroen Ketema (personal communication June 9, 2008) gave the following comments, in dutch, and drew a different Venn diagram, see below.

"Ik heb er even naar gekeken, en volgens mij is de situatie zoals in het aangehechte plaatje (waarbij ik even de oneindig tekens heb weggelaten).

(1) UN^∞ valt samen met het universum van de (oneindige) lambda termen, omdat oneindige lambda calculus Church-Rosser modulo hypercollapsing subtermen is.
(2) HC valt samen met RA, omdat de beta regel een collapsing regel is.

(3) HC is disjunct van SN^∞ en WN^∞, omdat HC termen per definitie geen normaalvorm kan hebben.

(4) HC is niet disjunct van CR^∞ - neem b.v. I^ω - maar valt ook niet samen met CR^∞ - door het standaard tegenvoorbeeld met twee verschillende ongebonden variabelen.

(5) SN^∞ en WN^∞ zijn duidelijk, CR^∞ moet deze bevatten, omdat oneindige lambda calculus Church-Rosser modulo hypercollapsing subtermen is.

(6) "geen WHNF" en "geen HNF" zijn niet disjunct van SN^∞ en WN^∞ - door b.v. jullie term A - maar vallen ook niet samen met WN^∞ of SN^∞, omdat HC bevat is in beide.
(A is $\mu x.\ xx$, the infinite binary tree consisting of only application nodes.)
(7) De situatie is bijna identiek voor iedere orthogonale, fully-extended iCRS. Het verschil is dat in dit meer algemene geval er termen in RA kunnen zitten die niet in HC zitten HC en "geen WHNF" en "geen HNF" zijn natuurlijk specifiek voor de lambda calculus."

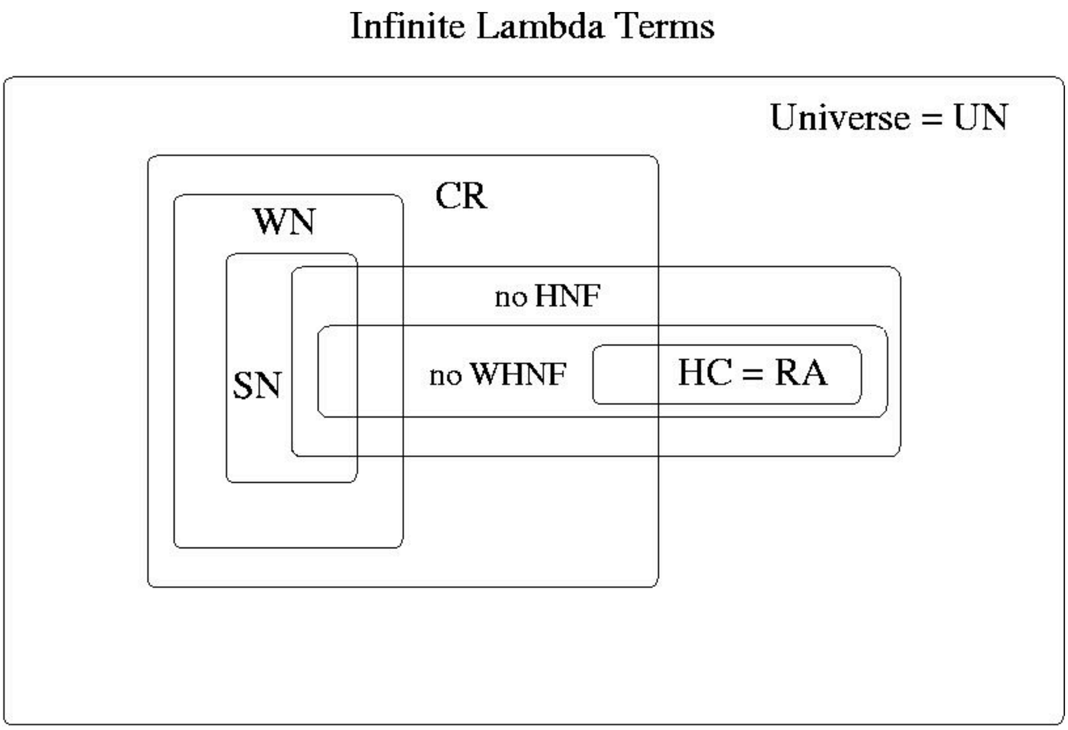


Figure 2.14. Classification of properties (by Jeroen Ketema)

Appendix E: Undefinability of Parallel-or

We would like to define a λ-term P (parallel-or) together with λ-terms T (true) and F (false) satisfying for all $\lambda\Omega$-terms X:

$$PXT =_\beta T \tag{1}$$

$$PTX =_\beta T \tag{2}$$

$$PFF =_\beta F. \tag{3}$$

For T, F we can take as in Barendregt [Bar84] $\lambda xy.x$ and $\lambda xy.y$, respectively. Now we can prove using BST and basic properties of Böhm trees that such a λ-term P does not exist.

Consider $BT(P\Omega\Omega)$. Since $P\Omega\Omega \leqslant_\Omega PxT$ (for some arbitrary variable x), we have by monotonicity of BT's and (1) that $BT(P\Omega\Omega) \leqslant_\Omega T$. Likewise, using (3), we have $BT(P\Omega\Omega) \leqslant_\Omega F$. Since Ω is the only minorant of both T and F, we have

$$BT(P\Omega\Omega) \equiv \Omega. \tag{$*$}$$

Now we can apply BST and conclude that the Ω in the right-hand side of $(*)$ is in one of three cases:

Case 1. The Ω has no origin in $P\Omega\Omega$.

Case 2. The Ω has as origin the first Ω in $P\Omega\Omega$.

Case 3. The Ω has as origin the second Ω in $P\Omega\Omega$.

Ad case 1. According to BST, the Ω in the right-hand side then is insensitive for increases at the two input Ω's in $P\Omega\Omega$. However, refining to PFF yields as BT output F, by Eq. (3); and this is a proper refinement of Ω. So this case does not apply.

Ad case 2. Now BST states that the right-hand side Ω is insensitive for increases of the second Ω in $P\Omega\Omega$. However, refining to $P\Omega T$ and using (1) we have as BT output T, a proper refinement of Ω in the right-hand side. So also this case is impossible.

Ad case 3. Now BST and (2) yield the impossibility.

We conclude that there is no $\lambda\Omega$-term P with the desired behavior (1)–(3). A fortiori there is no such λ-term.