Descendants and Origins in Term Rewriting

Inge Bethke* inge@cs.vu.nl
Jan Willem Klop*‡† jwk@cs.vu.nl
Roel de Vrijer‡ rdv@cs.vu.nl

Abstract

In this paper we treat various aspects of a notion that is central in term rewriting, namely that of descendants or residuals. We address both first order term rewriting and λ-calculus, their finitary as well as their infinitary variants. A recurrent theme is the Parallel Moves Lemma. Next to the classical notion of descendant, we introduce an extended version, known as 'origin tracking'. Origin tracking has many applications. Here it is employed to give new proofs of three classical theorems: the Genericity Lemma in λ-calculus, the theorem of Huet and Lévy on needed reductions in first order term rewriting, and Berry's Sequentiality Theorem in (infinitary) λ-calculus.
Note: This article is based on a lecture given by Jan Willem Klop at RTA '98 held in Tsukuba, Japan.

Contents

*University of Nijmegen, Department of Computer Science, Toernooiveld 1, 6525 ED Nijmegen, The Netherlands
†CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
‡Vrije Universiteit, Department of Theoretical Computer Science, de Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

1 Introduction

This paper is an extended version of a talk by the second author given at the conference RTA '98 in Tsukuba. The purpose of the talk and the present written version was and is to present a tour through term rewriting centered around the notion that permeates all of the theory of rewriting, namely that of descendants or residuals. A priori it is quite understandable why this notion is so all-pervasive in rewriting: rewriting is about the way expressions change due to fixed rewrite rules; to get a grip on this dynamic change one naturally concentrates on what remains constant in this change—that is, what remains, step after step, of some expression part ('residuals'), or how a subsequent expression part 'descends from' its 'ancestor' part. It is therefore not surprising that several of the basic lemmas in rewrite theory are phrased in terms of this notion of descendants or residuals[1].

The paradigm of such a classical lemma is the Parallel Moves Lemma (PML), which roughly is half of the Church-Rosser Theorem, that is the cornerstone

[1] In the classical λ-calculus literature one usually reserves the term 'residual' for a descendant of a redex.

of λ-calculus and first-order (orthogonal) term rewriting. Therefore, in the tour along various rewriting systems made in this paper, we will at each 'stop' consider PML again and discuss its validity or failure.

Being a tour, the paper is rather loosely structured. The proofs of classical facts are only sketched, but references to complete proofs are given. We have included a few historical remarks, but without any claim of completeness. Descendants are studied in various settings. Apart from our primary concern with λ-calculus and first-order orthogonal term rewriting, we pay attention to the notion of descendant in the $\lambda\beta\eta$-calculus (Section 5), in orthogonal infinitary term rewriting systems (Section 9) and in infinitary λ-calculus (Section 10).

A major focus of this paper is a refined version of the descendant/ancestor relation, called origin tracking, which was introduced in Klop [Klo90]. Several variants of this notion have been studied, sometimes with applications that are similar to the ones described in this paper. We mention the work of Boudol [Bou85], Khasidashvili [Kha90, Kha93], Maranget [Mar92], Glauert & Khasidashvili [GK94] and van Oostrom [Oos97a].[2] A distinctive feature is that our presentation makes extensive use of Lévy labels (see Section 6). The method of origin tracking gives rise to perspicuous proofs of some well-known classical theorems. In Section 7 we prove in some detail the Genericity Lemma in λ-calculus, and in Section 8 the theorem of Huet and Lévy on needed reductions in first order term rewriting. In Section 11 we outline a new proof for Berry's Sequentiality Theorem in (infinitary) λ-calculus.

To sum up, our subject matter stretches from first-order (orthogonal) term rewriting to λ-calculus; and in another dimension it stretches from finitary rewriting to infinitary rewriting.

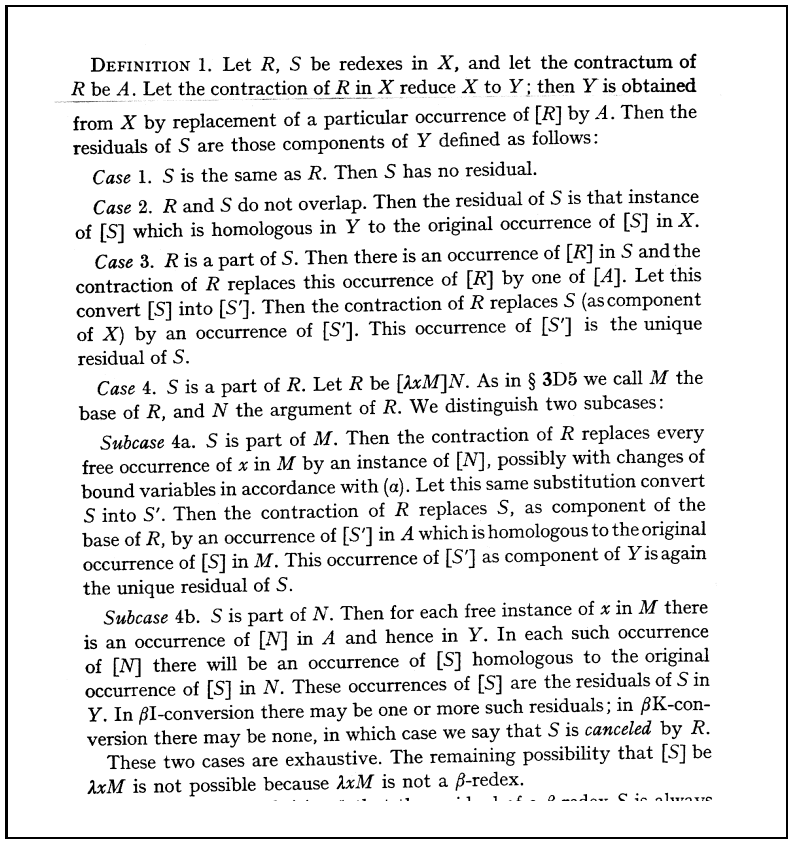
2 Early views on descendants

This paper does not intend to give a complete historical account of the origins of the residual notion in lambda-calculus and term rewriting, but we will shortly remember some of the prominent early contributions.

The notion of residual seems to originate with Church & Rosser [CR36], where it as used in the proof of the Church-Rosser theorem. There, and in Church [Chu41], one finds a lengthy verbal description of the notion of residual of a β-redex (after a sequence of α- and β-reductions). A detailed definition of residual for $\lambda\beta$-calculus in the same style as that of Church & Rosser is contained in Curry & Feys [CF58]; see Figure 1. This definition is clear but also verbose, using some intuitive descriptions ('homologous occurrence').

The first abstract treatment of rewriting is given by Newman [New42]. The paper contains the paramount result now known as Newman's Lemma and also proves the Finite Developments Theorem and the Church-Rosser Theorem for λ-calculus. The definition of residuals of a β-redex is given here by labeling bracket pairs with natural numbers and tracing these. Thus, also the idea of

[2] Further detailed studies involving residuals include Glauert & Khasidashvili [KG96, GK96, KG97], Kennaway et al. [KOV99], Khasidashvili & van Oostrom [KO95], van Oostrom [Oos96, Oos97b, Oos99].



DEFINITION 1. Let R, S be redexes in X, and let the contractum of R be A. Let the contraction of R in X reduce X to Y; then Y is obtained from X by replacement of a particular occurrence of $[R]$ by A. Then the residuals of S are those components of Y defined as follows:

Case 1. S is the same as R. Then S has no residual.

Case 2. R and S do not overlap. Then the residual of S is that instance of $[S]$ which is homologous in Y to the original occurrence of $[S]$ in X.

Case 3. R is a part of S. Then there is an occurrence of $[R]$ in S and the contraction of R replaces this occurrence of $[R]$ by one of $[A]$. Let this convert $[S]$ into $[S']$. Then the contraction of R replaces S (as component of X) by an occurrence of $[S']$. This occurrence of $[S']$ is the unique residual of S.

Case 4. S is a part of R. Let R be $[\lambda xM]N$. As in § 3D5 we call M the base of R, and N the argument of R. We distinguish two subcases:

Subcase 4a. S is part of M. Then the contraction of R replaces every free occurrence of x in M by an instance of $[N]$, possibly with changes of bound variables in accordance with (a). Let this same substitution convert S into S'. Then the contraction of R replaces S, as component of the base of R, by an occurrence of $[S']$ in A which is homologous to the original occurrence of $[S]$ in M. This occurrence of $[S']$ as component of Y is again the unique residual of S.

Subcase 4b. S is part of N. Then for each free instance of x in M there is an occurrence of $[N]$ in A and hence in Y. In each such occurrence of $[N]$ there will be an occurrence of $[S]$ homologous to the original occurrence of $[S]$ in N. These occurrences of $[S]$ are the residuals of S in Y. In βI-conversion there may be one or more such residuals; in βK-conversion there may be none, in which case we say that S is *canceled* by R.

These two cases are exhaustive. The remaining possibility that $[S]$ be λxM is not possible because λxM is not a β-redex.

Figure 1: Definition in Curry & Feys

defining residuals using labels originates with Newman [New42]. Later, Hindley [Hin69, Hin74] conducted an extensive axiomatic study of residuals. See Figure 2, displaying several assumptions about nesting of redexes ($\prec$) and residuals. Actually, several of these occur already in Newman [New42]. In recent years such studies have been taken up again by, among others, Plotkin, Gonthier, Lévy, Melliès and van Oostrom [Plo78, GLM92, Oos94, Mel97, Mel98].

The use of labels to trace subterms through a reduction was, in the form of 'underlining', an important ingredient in the early work of Barendregt. In [Bar71] he developed the technique of underlining into a sophisticated tool for the study of various systems of λ-calculus and Combinatory Logic.

With the appearance of the efficient inductive Church-Rosser proof for $\lambda\beta$-reduction discovered by Tait and Martin Löf (see e.g. Barendregt [Bar84]) detailed studies of the descendant relation seemed to be somewhat superseded (as remarked e.g. by Hindley in [Hin74]). This is not quite the case. We hope that, if anything, this paper shows that the descendant is alive.

Finally we mention an important contribution by O'Donnell [O'D77], presenting a deep analysis of orthogonal (first-order) term rewriting. He also investigated more general notions of residuals ('pseudo-residuals').

(C) *If a reduction ρ and a cell ξ are coinitial, then there exist reductions σ and τ such that $\rho + \sigma \simeq \xi + \tau$.* (See Figure 3.)

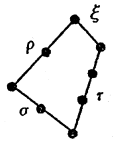


FIGURE 3

THEOREM 1. *Property* (C) *is implied by the conjunction of the following eight assumptions*:

(A1). $\xi \prec \eta \Rightarrow \eta \not\leftarrow \xi$.

(A2). $\xi \prec \eta \ \& \ \eta \prec \zeta \Rightarrow \xi \prec \zeta$.

(A3). *If $\xi \not\leftarrow \eta$, then ξ/η has no more than one member.*

(A4). $\xi/\xi = \varnothing$.

(A5). $\eta_1 \not\leftarrow \xi$ *and* $\eta_1 \not\leftarrow \eta_2 \Rightarrow \eta_1/\xi \not\leftarrow \eta_2/\xi$.

(A6). *If $\eta_i \prec \xi$ for $i = 1, \cdots, n$, then there exists k such that for all $j \neq k$, $\eta_j \not\leftarrow \eta_k$* and $\eta_j/\xi \not\leftarrow \eta_k/\xi$.

(A7). *If ξ and η are coinitial, then there exist an MCD ρ of ξ/η and an MCD σ of η/ξ, such that $\xi + \sigma \simeq \eta + \rho$.*

(A8). *If* (A7) *is true and ζ is any cell coinitial with ξ and η, then $\zeta/(\xi + \sigma) = \zeta/(\eta + \rho)$ in the following two cases*:

(i) $\zeta \not\leftarrow \xi$ *and* $\zeta \not\leftarrow \eta$,

(ii) $\eta \prec \xi$ *and* $\zeta \prec \xi$ *and* $\zeta \not\leftarrow \eta$ *and* $\zeta/\xi \not\leftarrow \eta/\xi$.

Figure 2: Hindley's axioms

3 Preliminaries

We briefly collect some preliminary notations and notions needed in the sequel. We assume familiarity with the λ-calculus and the notion of (first-order) term rewriting sytem (TRS). In general, we refer to Barendregt [Bar84], Dershowitz and Jouannaud [DJ90], Klop [Klo92], Baader and Nipkow [BN98].

3.1 Terms

The set of λ-terms is denoted $\mathrm{Ter}(\lambda)$. $M \equiv N$ denotes syntactic equality of terms M, N. Substitution of N for x in M is denoted by $M[x := N]$; here bound variables in M are assumed to be renamed when necessary to avoid capture of free variables in N. The notation $C[\ , \ldots, \ ]$ is used for a context with some holes; e.g. $(\lambda x.x[\ ])[\ ]y$. The result of substituting terms $N_1, \ldots, N_k$ for the holes, in the order from left to right, is denoted as $C[N_1, \ldots, N_k]$; in this case variables may be captured.

If S is a subterm (occurrence) in M, we write $S \subseteq M$. Likewise $s \in M$ when s is a symbol in M.

3.2 Reduction

We generally write $\to$ for a reduction or rewrite relation, possibly subscripted as in $\to_\beta$. Its transitive reflexive closure is denoted by $\twoheadrightarrow$ ($\twoheadrightarrow_\beta$ etc.), its reflexive closure by $\to^{=}$. The convertibility relation, i.e. the equivalence relation generated by $\to$, is denoted by $=$ ($=_\beta$ etc.).

We write $M \stackrel{R}{\to} N$ if M reduces to N by contracting the redex R. The reduction consisting of just that step is also denoted with $\{R\}$.

3.3 Unsolvables

A λ-term M is *solvable* if there are $N_1, \ldots, N_k$ such that $MN_1 \ldots N_k =_\beta I \equiv \lambda x.x$. Equivalently (see Barendregt [Bar84]): M has a head normal form. A *head normal form* is a λ-term of the form $\lambda x_1 \ldots x_n.yM_1 \ldots M_k$ for some variables $x_1, \ldots, x_n, y$ and λ-terms $M_1, \ldots, M_k$ $(n, k \geq 0)$. A term that is not solvable (so without head normal form) is called *unsolvable*. Unsolvables are closed under β-reduction, abstraction, substitution, and right application. A *weak head normal form* (see Abramsky & Ong [AO93]) is a term of the form $\lambda x.M$ or $yM_1 \ldots M_n$ $(n \geq 0)$. Note that a head normal form is also a weak head normal form, but not vice versa: consider for instance $\lambda y.(\lambda x.xx)(\lambda x.xx)$.

A *zero term* is a term that does not reduce to an abstraction term $\lambda x.P$. A *mute* term is a zero term that does not reduce to a variable, nor to an application MN where M is a zero term.

3.4 Term rewriting systems

We assume familiarity with the notion of (weakly) orthogonal first-order term rewriting system. We also assume some familiarity with the notion of higher-order rewriting, in the form of CRSs (Combinatory Reduction Systems) as in Klop, van Oostrom, van Raamsdonk [KOR93], or HRSs (Higher-order Rewrite Systems) as in Nipkow [Nip91]. SN stands for strong normalization, CR for Church-Rosser (see Klop [Klo92]).

3.5 The $\lambda\Omega$-calculus

The $\lambda\Omega$-calculus is λ-calculus equipped with a single constant Ω. Thus the set $\mathrm{Ter}(\lambda\Omega)$ of $\lambda\Omega$-terms is obtained by adding Ω to the formation rules of terms.

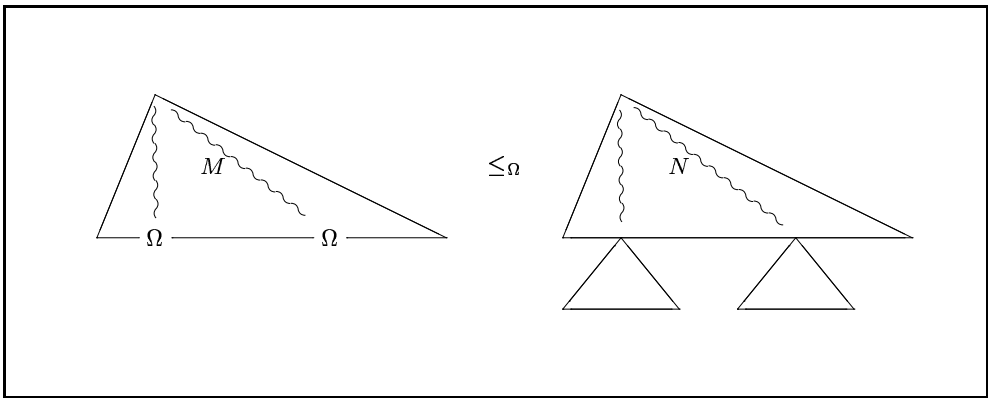


Figure 3: Refining Ω's

The $\lambda\Omega$-terms can be ordered partially with least element Ω in the following way: we say that a term N is *finer* than a term M and write $M \leq_\Omega N$, if N

originates from M by replacing some Ω's in M by some terms, or equivalently, if M originates from N by replacing, *cutting off*, some subterms of N by Ω. Figure 3 depicts this. In order to make this notion more precise, we shall identify $\lambda\Omega$-terms with their parse trees, i.e. rooted trees with binary application nodes @, unary abstraction nodes λx (where x is any variable), and leaves labeled either by a variable or by the constant Ω.

Finally, we define a $\lambda\Omega$-term M to be unsolvable if the λ-term $M[\Omega := z]$, that is, M with every occurrence of Ω replaced by some variable z, is unsolvable. Thus Ω is treated as an ordinary constant.

3.6 Redex patterns

A redex *pattern* is the 'fixed part' of the left-hand side of a reduction rule. So a pattern can be viewed as an incomplete term, or a context. In the $\lambda\beta$-calculus, with only the β-rule, there is, up to the choice of the bound variable, only one redex pattern: $(\lambda x.[\,])[\,]$. We will also represent it by the $\lambda\Omega$-term $(\lambda x.\Omega)\Omega$. Likewise, the redex patterns of a first-order term rewriting system can be represented either by a context or by replacing each variable in the left-hand side of a rewrite rule by the constant Ω.

4 Descendants in $\lambda\beta$-calculus

We will now give a more algebraic and less verbose definition of descendants in $\lambda\beta$-calculus. The definition is from Klop [Klo80]. We introduce *simply labeled λ-calculus* $\lambda_{\mathcal{A}}$ as follows:

Definition 4.1

1. $\mathcal{A} = \{a, b, c, \ldots\}$ is a set of labels.
2. $\mathrm{Ter}(\lambda_{\mathcal{A}})$, the set of $\lambda_{\mathcal{A}}$-terms, is given by the (quasi) BNF-definition $x^a \mid (AB)^a \mid (\lambda x.A)^a$. So, a $\lambda_{\mathcal{A}}$-term is an ordinary λ-term with each (occurrence of a) subterm superscribed with a label.
3. Labeled β-reduction, $\rightarrow_{\beta_{\mathcal{A}}}$, is defined as in Table 1:

$((\lambda x.M)^a N)^b$	$\rightarrow_{\beta_{\mathcal{A}}}$	$M[x := N]$
$x^a[x := N]$	$=$	N
$y^a[x := N]$	$=$	$y^a \quad (y \not\equiv x)$
$(AB)^a[x := N]$	$=$	$(A[x := N]B[x := N])^a$
$(\lambda y.A)^a[x := N]$	$=$	$(\lambda y.A[x := N])^a \quad (y \not\equiv x)$
$(\lambda x.A)^a[x := N]$	$=$	$(\lambda x.A)^a$

Table 1: Labeled β-reduction and substitution

A labeled λ-term A will sometimes be written as M^I where M is the λ-term obtained from A by erasing the labels and I is the *labeling map*, assigning a label to each subterm occurrence. Note that one can equivalently think of the labels being assigned to occurrences of symbols instead of subterms, a subterm being determined in a one-one way by its head symbol. Examples are the terms depicted as trees in Figure 4.

It is clear that, given a labeling I of M, a β-reduction step $\mathcal{R} = M \stackrel{R}{\to} N$ can be *lifted* to a labeled $\beta_{\mathcal{A}}$-step $\mathcal{R}^* = M^I \stackrel{R}{\to}_{\beta_{\mathcal{A}}} N^J$ for some labeling J of N (simply by contracting the 'same' redex R, but now also taking care of the labels).

Now in the following definition, we assume I be an *initial* labeling, that is: labels of distinct subterm occurrences are distinct.

Definition 4.2 Assume λ-terms M, N with $M \stackrel{R}{\to} N$, an initial labeling I of M, and let J be chosen such that the reduction step $M \stackrel{R}{\to} N$ lifts to $M^I \stackrel{R}{\to}_{\beta_{\mathcal{A}}} N^J$, as described above.

For symbol occurrences $s \in M$, $t \in N$ and subterm occurrences $S \subseteq M$, $T \subseteq N$ we define the relation $\blacktriangleright$ as follows:

$$s \blacktriangleright t \text{ iff } I(s) = J(t)$$

$$S \blacktriangleright T \text{ iff } I(S) = J(T)$$

and likewise for reductions $\twoheadrightarrow_{\beta_{\mathcal{A}}}$ of several steps. If $s \blacktriangleright t$, we say that s *descends to* t, or that t *is a descendant of* s, or that s *is an ancestor of* t. Likewise for S, T.

An example of a $\beta_{\mathcal{A}}$-reduction step is in Figure 4. Note that the subterms labeled with $37, 4, 7, 8$ have no descendants; in particular a redex has no residuals after its contraction[3]. Also, according to this definition, the function part ($\lambda x.M$ in the notation of Table 1) of the redex leaves no residuals, nor the variables substituted for. In the example, only the subterms labeled with $20, 2, 1, 0$ do have a residual after the displayed reduction step.

Remark 4.3 $\lambda_{\mathcal{A}}$ is an orthogonal CRS (HRS), and hence according to the general theory for higher-order rewriting (see Klop et al. [KOR93], Nipkow [Nip93]) $\lambda_{\mathcal{A}}$ is confluent.

4.1 Elementary diagrams

We now give a definition by example of the notion of *elementary diagram* (e.d.) for β-reduction.

Consider $M \equiv \omega_3(II)$ with $I \equiv \lambda x.x$, $\omega_3 \equiv \lambda x.xxx$ and 'diverging' reduction steps

$$M \to_\beta (II)(II)(II) \equiv M'$$

and

$$M \to_\beta (\lambda x.xxx)I \equiv M''.$$

[3]This is Hindley's assumption $\xi/\xi = \emptyset$ in Figure 2.

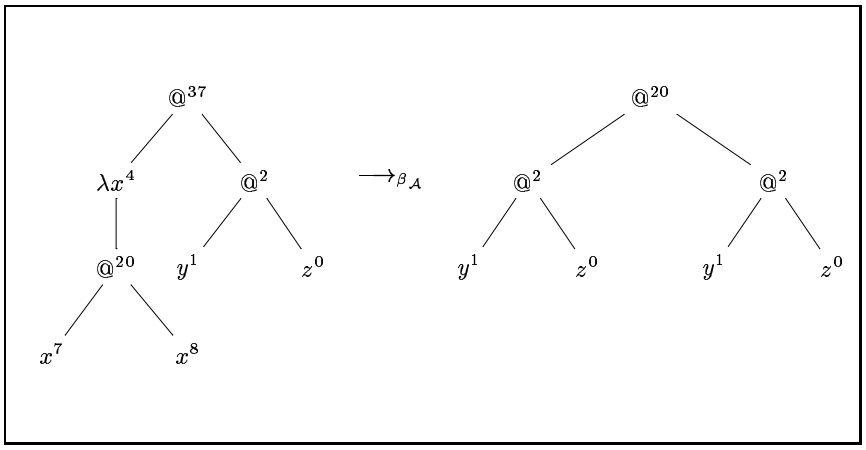


Figure 4: $\beta_{\mathcal{A}}$-reduction step

Clearly the canonical way of finding a common reduct M''' of M' and M'' is given by the converging reductions

$$M' \to_\beta \to_\beta \to_\beta III$$

(here the three residuals of redex II are contracted) and

$$M'' \to_\beta III$$

(here the one residual of the redex $\omega_3(II)$ is contracted).

The diagram spanned by M, M', M'', M''' as points and these reduction steps is an elementary diagram. It has the form of the first diagram in Figure 5. So in general on the lower and right side of an e.d. the residuals of the original

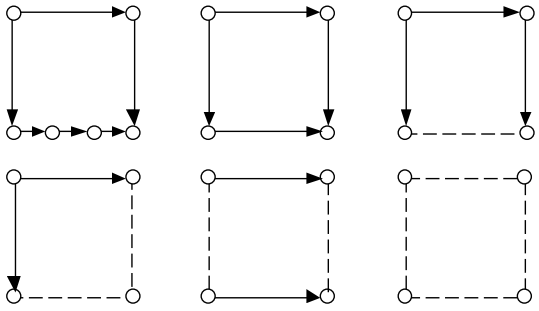


Figure 5: Elementary diagrams

redex at the opposite side (upper and left, respectively) are contracted.[4] In the

[4]Historically, the root of the elementary diagram construction is 'Property (D)' in Curry & Feys [CF58], here (slightly) paraphrased as follows:

> If R and S are two redexes in X, and the contraction of R followed by contractions of the residuals of S converts X to Y, then a contraction of S followed by contractions of the residuals of R also leads to Y.

example the lower edge splits into three steps, in general the lower or right edge may split into any number of steps.

When there are no residuals, e.g. when the redex R is *erased*, as in a reduction step $(\lambda x.y)R \to y$, we use so-called *empty steps*, in order to keep the diagrams rectangular. Empty steps also result if the two original diverging steps are in fact identical, i.e. contract the same redex.

Figure 5 displays (essentially) all types of e.d.'s that exist. Empty steps are indicated by a dashed line. Note that also *improper* e.d.'s arise, when we start with an empty step at the left and/or the upper side.

4.2 Reduction diagrams

The elementary diagrams (that we suppose are scalable) will now be used to construct reduction diagrams spanned by two coinitial diverging finite reduction sequences. For $\lambda\beta$-calculus, this 'paving' or 'tiling' procedure will always terminate successfully (i.e. we do not have an infinite regress of ever smaller e.d.'s). See Figure 6 for a successfully completed reduction diagram.

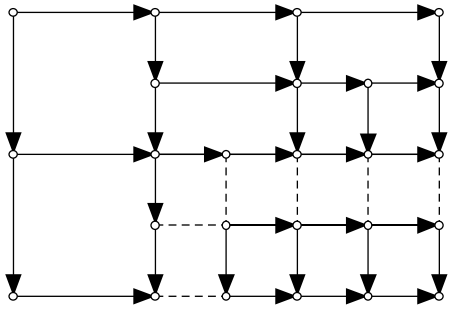


Figure 6: Completed reduction diagram

The insight that the construction of a reduction diagram will always succeed, is one route to the Church-Rosser Theorem, stating that any pair of (finite) reductions originating in the same term M can be continued in such a way that they meet again in a common reduct N:

$$M \twoheadrightarrow M_0 \;\&\; M \twoheadrightarrow M_1 \;\;\Rightarrow\;\; (\exists N)\,(M_0 \twoheadrightarrow N \;\&\; M_1 \twoheadrightarrow N).$$

4.3 The Parallel Moves Lemma (PML)

If we set out a single reduction step against a multiple step reduction and construct the corresponding reduction diagram, we have the situation of the classical Parallel Moves Lemma of Curry & Feys [CF58] (Figure 7). The reduction in the right side of the diagram consists of 'parallel' contraction of the residuals of the original redex contracted in the left vertical step; that is, the residuals after the horizontal upper reduction. This follows from the way the diagram is constructed: by simply tiling with elementary diagrams. The faint arrows in Figure 7 suggest how residuals of the contracted redex propagate.

So the steps at the right side of the diagram are the parallel moves. The word parallel should be understood here in the sense of 'at once'; it is not meant

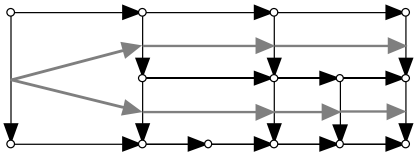


Figure 7: Parallel Moves Lemma

to imply that the redexes involved should be disjoint. Actually, in λ-calculus this will not always be the case.[5] In contrast: in orthogonal first-order term rewriting systems 'parallel' can be taken in the strict sense, since there residuals of disjoint redexes are always disjoint.

4.4 Projections

The diagram construction yields the notion of projection of reduction sequences over each other. Thus, if $\mathcal{R}, \mathcal{S}$ are coinitial reductions, constituting the left and upper side of reduction diagram $\mathcal{D}$ respectively, then the lower side is $\mathcal{S}/\mathcal{R}$ ('$\mathcal{S}$ projected over $\mathcal{R}$') and the right side is $\mathcal{R}/\mathcal{S}$ ('$\mathcal{R}$ projected over $\mathcal{S}$') (see Figure 8).

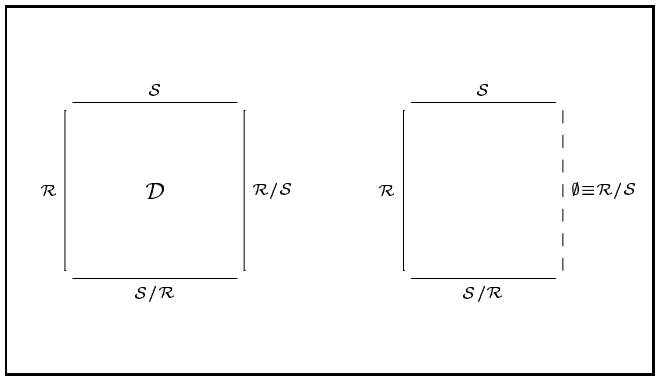


Figure 8: Projection and Lévy's partial order

4.5 Lévy-equivalence

Lévy [Lév78] has introduced an important notion of equivalence on reductions. By our use of empty steps, it may happen that $\mathcal{R}/\mathcal{S}$ or $\mathcal{S}/\mathcal{R}$ are the empty reduction $\emptyset$. If both are empty, we say that $\mathcal{R} \equiv_L \mathcal{S}$ ($\mathcal{R}$ and $\mathcal{S}$ are *Lévy-equivalent*). In a literal sense, $\mathcal{R}$ and $\mathcal{S}$ have cancelled each other out in the diagram, i.e. they perform somehow the same steps in a permuted way. Therefore $\equiv_L$ is also known as 'permutation equivalence'. We can also obtain a partial order on reductions (after Lévy): $\mathcal{R} \sqsubseteq_L \mathcal{S}$ if $\mathcal{R}/\mathcal{S} \equiv \emptyset$ (see Figure 8). Intuitively, $\mathcal{R} \sqsubseteq_L \mathcal{S}$ means that $\mathcal{R}$ does less or the same work as $\mathcal{S}$. So $\equiv_L$ is the symmetric closure of $\sqsubseteq_L$.

[5] An interesting observation for λ-calculus, due to S. Micali, is that if the upper side of the PML diagram is a development, then the right side *does* consists of disjoint redex contractions; see Klop [Klo80].

The projection operation "/", together with concatenation of reductions "$\cdot$" ($\mathcal{R} \cdot \mathcal{S}$ is $\mathcal{R}$ followed by $\mathcal{S}$) have the equational properties of Table 2. See also Figure 9.

$x \cdot \emptyset$	$=$	x	$x \cdot \emptyset$	$\to$	x
$\emptyset \cdot x$	$=$	x	$\emptyset \cdot x$	$\to$	x
$x/\emptyset$	$=$	x	$x/\emptyset$	$\to$	x
$\emptyset/x$	$=$	$\emptyset$	$\emptyset/x$	$\to$	$\emptyset$
x/x	$=$	$\emptyset$	x/x	$\to$	$\emptyset$
$(x \cdot y)/z$	$=$	$(x/z) \cdot (y/(z/x))$	$(x \cdot y)/z$	$\to$	$(x/z) \cdot (y/(z/x))$
$z/(x \cdot y)$	$=$	$(z/x)/y$	$z/(x \cdot y)$	$\to$	$(z/x)/y$

Table 2: Lévy-equivalence

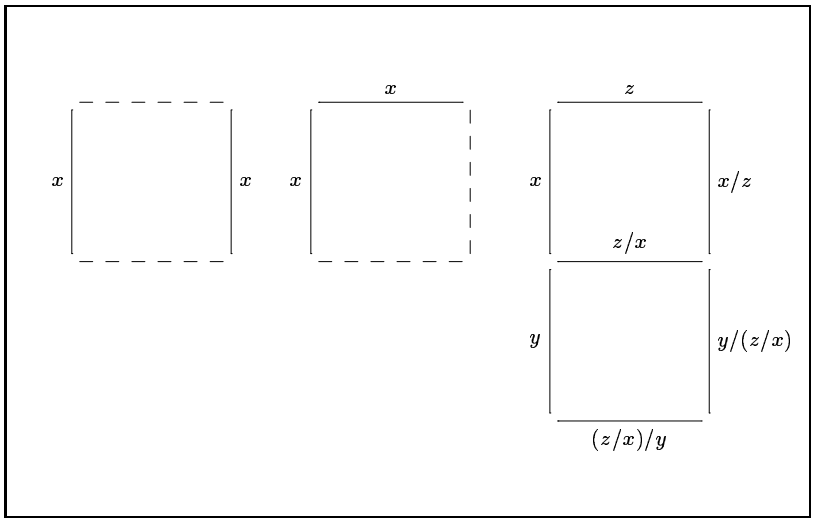


Figure 9: Projections

Reading these equations as rewrite rules, we have an abstract description of the process of construction of a projection. It is an instructive and non-trivial exercise to prove that this 'confluence TRS' is itself terminating and confluent. For the latter, an analysis of critical pairs suffices; most critical pairs are easily seen to be convergent, but one is non-trivial and converges only after several steps.

4.6 Redex creation and finite developments

Clearly, the descendants of a β-redex are again β-redexes. Vice versa, the ancestor of a β-redex, which always exists, does not need to be a β-redex. Such a redex, not being the descendant of a redex, is called *created*.

Lévy [Lév78] has analyzed how such creations happen. It turns out that there are the three situations responsible for redex creation depicted in Table 3 (see also Klop [Klo80]) where σ is the substitution $[x := \lambda y.A]$, σ' is $[x := D]$, and $C[\ ]$ and $\cdots[\ ]\cdots$ are arbitrary contexts.

I.	$\cdots[(\lambda x.C[xB])\lambda y.A]\cdots$	$\to_\beta$	$\cdots[C^\sigma[(\lambda y.A)B^\sigma]]\cdots$
II.	$\cdots[(\lambda x.x)(\lambda y.A)B]\cdots$	$\to_\beta$	$\cdots[(\lambda y.A)B]\cdots$
III.	$\cdots[(\lambda x.\lambda y.A)DB]\cdots$	$\to_\beta$	$\cdots[(\lambda y.A^{\sigma'})B]\cdots$

Table 3: Redex creation after Lévy

Remarkably, such redex creations are what makes infinite reductions possible: *in every infinite reduction in λ-calculus some created redex must be contracted.* This is actually a rephrasing of the Finite Developments (FD) Theorem. The usual formulation reads as follows: Let $\mathcal{R} : M_0 \to_\beta M_1 \to_\beta M_2 \to_\beta \cdots \to_\beta M_i \to_\beta \cdots$ be a reduction such that in each step, the contracted redex is a descendant of some redex in M_0. Then $\mathcal{R}$ is in fact a finite reduction; it is called a *development* of M_0. (If in the final term of $\mathcal{R}$ no descendant of a redex in M_0 is left, $\mathcal{R}$ is a *complete* development.) There are many proofs of FD. We refer to Barendregt [Bar84], Krivine [Kri93], van Oostrom & van Raamsdonk [OR94], de Vrijer [Vri85].

So, a development arises by forbidding contraction of redexes created as in I–III. It turns out that types II and III of redex creation are a somewhat more innocent way of creation. If we forbid only contraction of type I redexes in a reduction, we have by definition a *superdevelopment* (van Raamsdonk [Raa93]). Superdevelopments are also finite. It is interesting to note that, where developments correspond naturally with the notion of parallel reduction employed in the confluence proof of Tait-Martin Löf, superdevelopments correspond to the parallel reduction employed in a slight variant of that proof, by Aczel [Acz78]. See Appendix A.

4.7 Standardization and a duality

The next main theorem in $\lambda\beta$-calculus to be discussed is the Standardization Theorem. Again its formulation and proof crucially depend on the notion of descendant. Standardizing a reduction sequence can be compared to sorting a sequence of natural numbers in ascending order. In standardizing a reduction the redex contractions are permuted so that they occur in a left-to-right manner; the action in a standard reduction literally moves to the right, and an increasingly large left part of the term is fixed. More precisely: at every redex contraction in M we consider the λ of the contracted β-redex and mark all λ's to the left of it with *. These marks are inherited during the remaining reduction as if they were firmly glued to the λ's. Now the requirement for a standard reduction is that no marked redex, i.e. of the form $(\lambda^* x.A)B$, is contracted.

We discuss two proofs of the Standardization Theorem (from [Klo80]). Given a reduction $\mathcal{R} : M \equiv M_0 \to \cdots \to M_n$, we call the redex occurrence $R \subseteq M$ the *leftmost contracted redex* (notation $lmc(\mathcal{R})$) if:

1. R has a descendant in some M_i in $\mathcal{R}$ that is contracted in the step $M_i \to M_{i+1}$,
2. Of the redexes in M satisfying 1, the redex R is the leftmost one. (Leftmost in the textual left-to-right order; we compare just the position of 'the' λ of the redex.)

The algorithm to compute 'the' standard reduction $\mathcal{R}_s$ for a given reduction $\mathcal{R} : M_0 \to \cdots \to M_n$ is as follows. Define

$$\mathcal{R}_0 = \mathcal{R}$$

$$\mathcal{R}_{n+1} = \mathcal{R}_n / \{lmc(\mathcal{R}_n)\}$$

Note that since $\{lmc(\mathcal{R}_i)\}/\mathcal{R}_i = \emptyset$, the endpoint of each $\mathcal{R}_i$ is M_n.

Then the reduction

$$\mathcal{R}_s : M_0 \overset{lmc(\mathcal{R}_0)}{\to} M_1' \overset{lmc(\mathcal{R}_1)}{\to} M_2' \overset{lmc(\mathcal{R}_2)}{\to} \cdots$$

will terminate in M_n, and is indeed a standard reduction. The proof of termination in [Klo80] uses strong normalization of a labeled lambda calculus à la Lévy, see Section 6. It is 'the' standard reduction for $\mathcal{R}$, as it is Lévy-equivalent to $\mathcal{R}$ and actually the unique standard reduction in the Lévy-equivalence class of $\mathcal{R}$.

Figure 10 gives an example of the operation of this algorithm. Here $1, 2, 3$

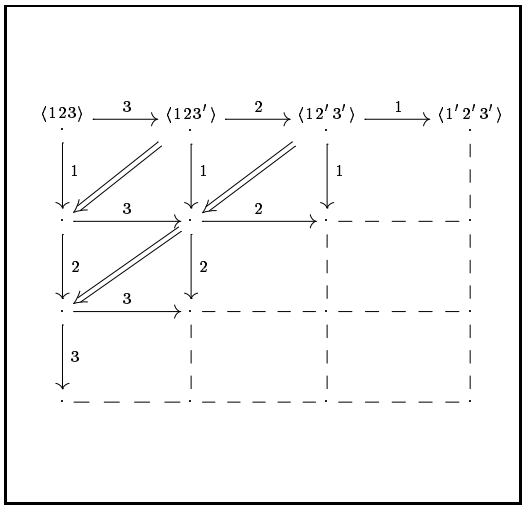


Figure 10: Standard reduction obtained by the 'lmc' procedure

are redexes and $1', 2', 3'$ are their respective contracta; $\langle \Box\Box\Box \rangle$ is some context with three holes, e.g. $\lambda z.z\Box\Box\Box$. The upper reduction is not standard; the left reduction down is. Note the sorting effect obtained in the standard reduction.

An example of a reduction that is not standard is

$$(\lambda x.xx)((\lambda x.x)y) \to (\lambda^* x.xx)y \to yy.$$

It is not standard, as the second step contracts a marked and thus forbidden redex. We call a two step reduction that is not standard, an *anti-standard pair*, and it is not hard to see that every reduction that is not standard must contain such an anti-standard pair. Thus, an alternative standardization algorithm suggests itself: swap such anti-standard pairs so that they become standard; an example is in Figure 11, where the reduction $\omega(II) \to \omega I \to II$ is 'swapped' to yield $\omega(II) \to II(II) \to I(II) \to II$ which is standard. Now one can prove

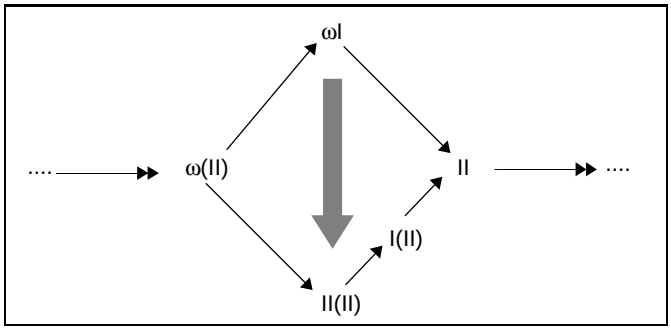


Figure 11: Swapping an anti-standard pair

(see Klop [Klo80]) that repeated swapping of anti-standard pairs in a given reduction $\mathcal{R}$ will terminate eventually in a standard reduction $\mathcal{R}_s$ for $\mathcal{R}$ (that coincides with the one found by the lmc algorithm above). The diagram in Figure 10 shows how three swaps starting from the horizontal reduction yield its standard reduction. We note that there is an interesting duality, expressed in Figure 12: an elementary diagram as depicted there, traversed from top left

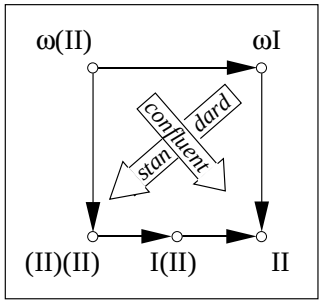


Figure 12: Duality between confluence and standardization

to bottom right tends to obtain confluence; traversed from top right to left bottom it tends to obtain standardization. This duality is also discussed in Melliès [Mel97].

5 Descendants in $\lambda\beta\eta$-calculus

Extending the $\lambda\beta$-calculus with the η-rule, $\lambda x.Mx \to M$ if x occurs not free in M, a complication arises due to overlap of the patterns of a β-redex and an η-redex, as in Figure 13. First, let us consider how the definition of descendant

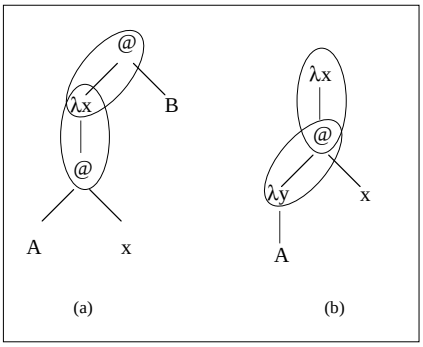


Figure 13: $\beta\eta$-overlap

(residual) of a redex can be adapted.

The definition of Curry-Feys [CF58] was as follows. Let R be the redex contracted in $M \to M'$ and $S \subseteq M$ the redex whose residuals we want to define. Four new cases arise, by overlapping[6] the patterns of the respective redexes:

1. $R \equiv (\underline{\lambda x.Ax}_S)B \to_\beta AB$,
2. $S \equiv (\underline{\lambda x.Ax}_R)B \to_\eta AB$,
3. $S \equiv (\lambda x.\underline{(\lambda y.A(y))x}_R) \to_\beta \lambda x.A(x)$,
4. $R \equiv \lambda x.\underline{(\lambda y.A(y))x}_S \to_\eta \lambda y.A(y)$.

In all these cases Curry-Feys define that S has no residuals after contraction of R. Indeed, this definition is entirely plausible. E.g. in case 1 contraction of R makes the symbols @ and λx of its pattern disappear. But thereby also the pattern of η-redex S is destroyed, since it uses the same symbol λx. Consequently, S has no residual, and analogously in the other cases.

Remark 5.1 Note that with the Curry-Feys definition of $\beta\eta$-residual there is the following phenomenon, that may seem curious. In case 2, if the η-redex $R \equiv \lambda x.Ax$ is actually $\lambda x.(\lambda y.A'y)x$, then η-reduction of R yields $S' \equiv (\lambda y.A'y)B$ which is still a β-redex, as before. So why would it not be a residual of the earlier one? Likewise (dually) in case 3.

As for the elementary diagrams, the consequence of this definition is the appearance of two new e.d.'s as in Figure 14, corresponding to the two ways the patterns of a β-redex and η-redex may overlap. Actually, these e.d.'s also show

[6] In Hindley [Hin77] the overlapping redexes R, S, in any of the situations 1–4 are suggestively called 'too close together'.

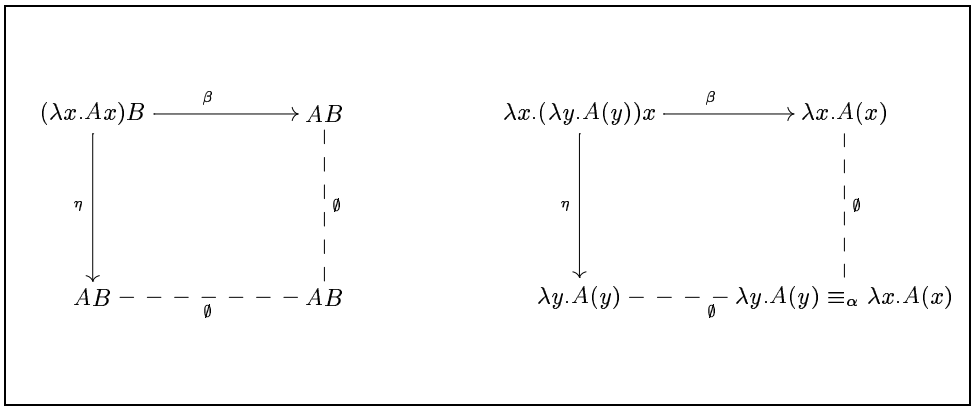


Figure 14: New elementary diagrams

that the critical pairs of the $\lambda\beta\eta$-calculus are trivial; in other words, that $\lambda\beta\eta$-calculus is *weakly orthogonal*. From this it follows by a recent general theorem for higher-order rewrite systems (HRSs or CRSs), of which $\lambda\beta\eta$-calculus is a typical example, that the $\lambda\beta\eta$-calculus is confluent (cf. van Oostrom & van Raamsdonk [OR94]). By contrast the CR proof of Curry & Feys is ad hoc and uses postponement of η-reductions. A simpler proof using commutation of β and η-reductions was given in [Klo80].

However, standard proofs of confluence for $\lambda\beta$ that rely on keeping track of residuals do not carry over easily to $\lambda\beta\eta$. This is because another classic result in λ-calculus, the Parallel Moves Lemma, fails for $\lambda\beta\eta$ with the definition of residuals due to Curry-Feys. Recall that PML reads:

> Given a one-step reduction $\{R\}$ against a reduction $\mathcal{R}$, construction of the diagram using tiling with e.d.'s yields a diagram $\mathcal{D}$ whose right side $\{R\}/\mathcal{R}$ consists of contractions of the residuals of R after $\mathcal{R}$.

The following counterexample is taken from [Klo80]. A similar counterexample was given independently by R. Hindley in unpublished notes.

Counterexample 5.2 Consider the reduction

$$\begin{array}{lllll}
\mathcal{R}: & (\lambda a.(\lambda b.ba)a)[\lambda z.(\underline{\lambda y.zIy})] & \equiv & M_0 & \to \\
& (\lambda b.b[\lambda_0 z.(\underline{\lambda_0 y.zIy})])[\lambda_1 z.(\underline{\lambda_1 y.zIy})] & \equiv & M_1 & \to \\
& [\lambda_1 z.(\underline{\lambda_1 y.zIy})][\lambda_0 z.(\underline{\lambda_0 y.zIy})] & \equiv & M_2 & \to \\
& \underline{\lambda_1 y.[\lambda_0 z.(\underline{\lambda_0 y.zIy})]Iy} & \equiv & M_3 & \to \\
& \underline{\underline{\lambda_1 y.(\underline{\lambda_0 y.IIy})y}} & \equiv & M_4 & \to \\
& \underline{\lambda_1 y.IIy} & \equiv & M_5 &
\end{array}$$

with $I \equiv \lambda x.x$. In the reduction sequence $\mathcal{R}$ (see also Figure 15) the labels 0, 1 are introduced to be able to indicate which redexes are contracted. The underlined redexes are the η-redex $R \equiv \lambda y.zIy$ in M_0 and its residuals. First R is doubled ($\lambda_0 y$ and $\lambda_1 y$) in M_1 and then one of these residuals is substituted in the other ($\lambda_0 y$ in $\lambda_1 y$) in M_3. In M_4 the symbol $\lambda_0 y$ turns out to belong to the pattern of a β-redex $(\lambda_0 y.IIy)y$. Contracting this redex destroys the other residual $\lambda_1 y$, according to case 3 in the definition of CF-residuals. So, the η-redex $M_5 \equiv \lambda_1 y.IIy$ is not a residual of the original η-redex R. But precisely that redex is contracted in $\{R\}/\mathcal{R}$. Hence the

Parallel Moves Lemma does not hold for the residual concept of Curry-Feys. (Note, however, that the final η-redex M_5 *is* a residual of the original η-redex in M_0 via the alternative reduction path $M_0 \to M_1 \to M_1' \to M_2' \to M_3' \to M_5$.[7])

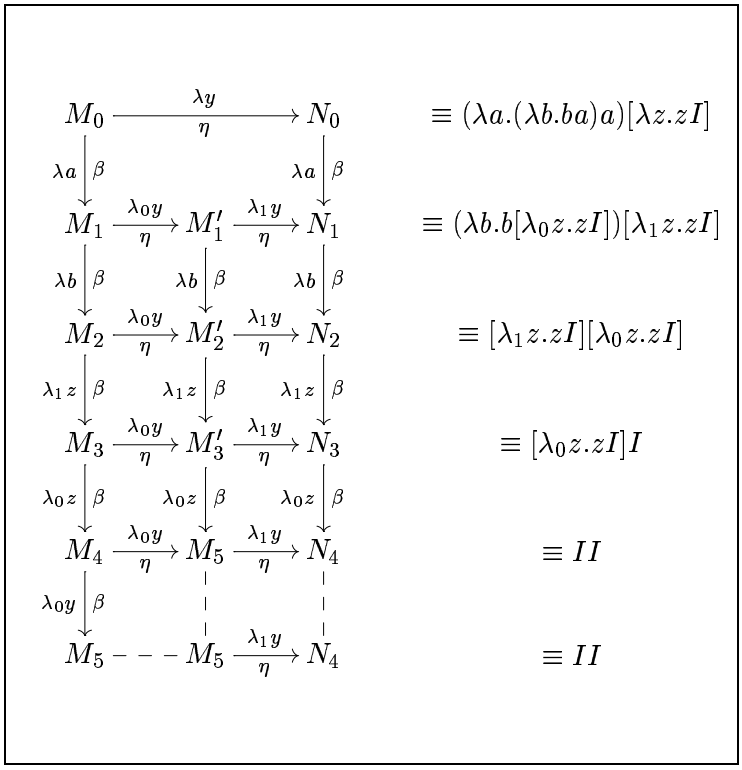


Figure 15: Counterexample to PML

Apparently, the notion of Curry-Feys residual is problematic in $\lambda\beta\eta$. This is a drawback in proving some of the classical theorems such as Standardization and Leftmost Normalization for $\lambda\beta\eta$. There are several ways to overcome this problem.

One is to avoid the concept of residual altogether in developing the syntactic theory of $\lambda\beta\eta$. An example of this strategy can be found in the work of Takahashi [Tak95] who proves Leftmost Normalization using inductive proofs in the style of Tait and Martin Löf's well-known proof of confluence for $\lambda\beta$.

Another way is changing the CF-notion of residual, in order to get a notion that is better behaved. We list two approaches.

1. Klop, in [Klo80], remarks that tracing just the symbol λ in a reduction in $\lambda\beta\eta$ is easy and without problems, and defines:

   Definition 5.3 Let $\mathcal{R} = M_0 \to \cdots \to M_k$ be a $\beta\eta$-reduction, $R_0 \subseteq M_0$ a redex (β- or η), $R_k \subseteq M_k$ a redex, such that the head-λ of R_k descends (can be traced back) to that of R_0. Then, regardless of whether R_0, R_k are β- or η-redexes, R_k is a λ-*residual* of R_0.

[7] This shows a fundamental weakness of the residual notion of Curry-Feys: it is not independent of the order of reduction steps.

Note that in the $\lambda\beta$-calculus the notion of λ-residual coincides with the ordinary descendant notion. What is more, all residuals according to Curry-Feys are also λ-residuals. But in Figure 15 the final η-redex M_5 now is a λ-residual of the original η-redex in M_0.

Then Klop [Klo80] proves that PML for this revised notion of residual does hold[8]. Furthermore, using this lemma one can prove Standardization and Leftmost Normalization for $\lambda\beta\eta$. The proofs, however, are very laborious and tiresome. A further drawback is that FD, the theorem of Finite Developments, does not hold for λ-residuals. See Appendix B.

2. A case of overlap that is similar to that between β- and η-redexes has been studied by de Vrijer [Vri87, Vri89], in the context of λ-calculus with surjective pairing. It is between the reduction rules for projection, $\pi_0(\pi XY) \to X$, and surjectivity, $\pi(\pi_0 X)(\pi_1 X) \to X$. The notion of *cluster residual*[9] defined there can be easily adapted to the present case of $\lambda\beta\eta$. In the next section the approach with cluster residuals is briefly sketched.

5.1 Cluster residuals

We will again use labels $a, b, c, \ldots$ in order to trace symbols through a reduction, and indicate a redex by the pair of labels of the symbols that make up its pattern. Note that the definition of CF-residual boils down to the requirement that both symbols of the pattern, a λ and a @, trace back to the pattern of the ancestor. In contrast, in the notion of λ-residual this requirement is made only for the λ. The notion of cluster residual lies somewhere in between. It is an extension of CF-residuals with some, but not all, of the λ-residuals. Moreover, the symmetry in the treatment of the symbols λ and a @, lacking in the λ-residual approach, is restored.

We take a closer look at the critical reduction step, $M_4 \to M_5$, in the reduction $\mathcal{R}$ in Counterexample 5.2. It is depicted in Figure 16, the relevant symbols labeled with distinct labels a, b, c, d. The patterns of the involved redexes have been encircled. The residuals $(ab), (cd)$ of the original R (the underlined redexes) with a drawn line, the contracted redex (bc) by a dotted line. These three redexes form a cluster: the middle redex pattern shares a symbol with each of its neighbour redex patterns.

It has already been observed above, in Counterexample 5.2, that the redexes (ab) and (cd) in M_4 have no CF-residual in M_5. In order to get to $N_4 \equiv II$ we need to contract the 'created' η-redex (ad). Now we declare this redex to be a *cluster residual* of the *two* η-redexes $(ab), (cd)$. The general definition just follows this example.

Definition 5.4 CF-residuals are cluster residuals. Moreover, if in a term M we have a *cluster* of overlapping redexes $(ab), (bc), (cd)$, and $M \to N$ by contraction

[8] Albeit, in a slightly weakened form: the parallel steps at the right side of the diagram are λ-residuals of the original R, but not necessarily *all* residuals.

[9] It is called 'virtual residual' in [Vri89], but the term 'cluster residual', which we now propose, is more descriptive.

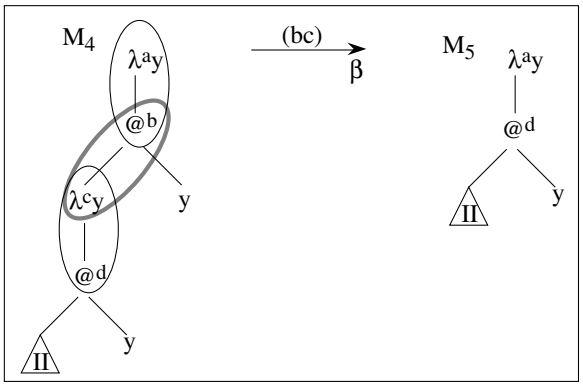


Figure 16: A redex cluster with cluster residual

of (bc), then (ad) in N is a cluster residual of $(ab), (cd)$.

Note that a cluster residual is of the same type (β or η) as its ancestors. An example of a β-residual is depicted in Figure 17. Obviously this definition

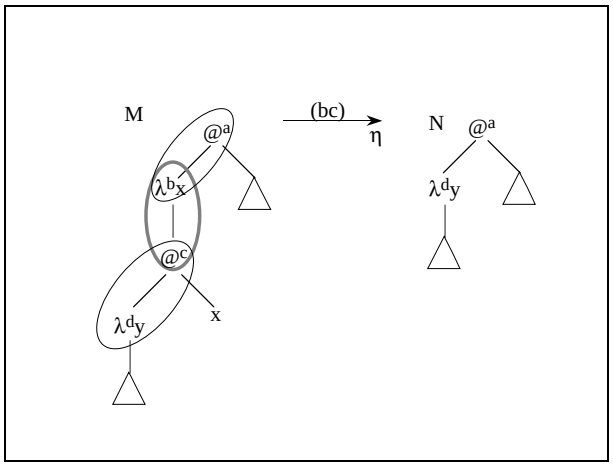


Figure 17: A redex cluster with β-cluster residual

only works if one traces sets of redexes, rather than individual redexes. This is a common procedure, but normally the residual relation is *distributive*: the residuals of a set of redexes S is the set of the residuals of the elements of S. If we also take cluster redexes into account this is not longer the case.

Along the lines of [Vri89] one can now show that with this extended notion of residual both PML and FD go through, and also standard proofs of CR that involve tracing of residuals. Moreover, the proof of standardization in terms of the *lmc*-procedure, sketched above for $\lambda\beta$, can also be easily generalized to $\lambda\beta\eta$.

It may be worth noting that the would-be residuals of Remark 5.1 are in fact cluster residuals.

6 Lévy's labeled λ-calculus

In the next two sections we will make use of a system of labeling λ-terms, due to Lévy [Lév75], that is a refinement of the simply labeled λ-calculus of Definition 4.1. We will refer to Lévy's labeled lambda calculus as λ_L-calculus or just λ_L. The Lévy-labels form a powerful tool, serving several purposes. The original purpose was to have a notation that not only enables one to trace residuals of redexes in the original term through a given reduction, as we did with $\lambda_{\mathcal{A}}$ in Section 4, but also to trace β-redexes that are created during that reduction. In a very general way, λ_L records the history of what happens in a β-reduction. Thereby it can be used to define the relation of *Lévy-equivalence* (or *permutation equivalence*) on reductions, which we discussed earlier in Section 4.5. Two reductions are Lévy-equivalent if they, put roughly, perform the same 'work', be it in a possibly different order. In some situations Lévy-labels are also a useful tool for proving termination (SN).

We will apply Lévy-labels in our definitions of origin tracking in Sections 7 and 8. In the latter section Lévy-labels will be adapted to the framework of first-order term rewriting systems.

The λ_L-calculus is defined as follows.

Definition 6.1

1. $L' = \{a, b, c, \ldots\}$ is a set of atomic labels.
2. L is the set of composite labels defined by
   (a) $L' \subseteq L$
   (b) $\alpha, \beta \in L \Rightarrow \alpha\beta \in L$
   (c) $\alpha \in L \Rightarrow \underline{\alpha} \in L$

So the labels are words over the set of atomic labels, with (nested) underlinings. An example of a label is: $\underline{aba\underline{c}}$. Note that there is in general no unique decomposition of labels that are formed with clause (b): there is only one label abc, composed of either a and bc, or ab and c. Further note the difference between the labels $\underline{d}\,\underline{e}$ and $\underline{de}$.

Now the set of λ_L-terms consists of λ-terms where every subterm has a label $\in L$. Often we will write a labeled term as M^I, where I is the function that maps the subterm occurrences to the set of labels. Multiple labels will be simplified as follows: $(M^\alpha)^\beta \mapsto M^{\alpha\beta}$.

Definition 6.2 The height $h(\alpha)$ of a label α is defined as follows:

1. $h(a) = 0$ for a $\in L'$
2. $h(\alpha\beta) = \max(h(\alpha), h(\beta))$
3. $h(\underline{\alpha}) = h(\alpha) + 1$.

So in the example $\alpha \equiv \underline{aba\underline{c}}$, we have $h(\alpha) = 2$.

$(\lambda x.M)^\alpha N \quad \to_{\beta_L} \quad M^{\underline{\alpha}}[x := N^{\underline{\alpha}}]$		
$x^\beta[x := N^{\underline{\alpha}}]$	$=$	$N^{\underline{\alpha}\beta}$
$y^\beta[x := N^{\underline{\alpha}}]$	$=$	$y^\beta \quad (y \not\equiv x)$
$(AB)^\beta[x := N^{\underline{\alpha}}]$	$=$	$(A[x := N^{\underline{\alpha}}]B[x := N^{\underline{\alpha}}])^\beta$
$(\lambda y.A)^\beta[x := N^{\underline{\alpha}}]$	$=$	$(\lambda y.A[x := N^{\underline{\alpha}}])^\beta \quad (y \not\equiv x)$
$(\lambda x.A)^\beta[x := N^{\underline{\alpha}}]$	$=$	$(\lambda x.A)^\beta$

Table 4: Lévy-labeled β-reduction and substitution

Definition 6.3 The reduction relation $\to_{\beta_L}$ and substitution in λ_L are defined as in Table 4. Here the label α is called the *degree* of the redex $(\lambda x.M)^\alpha N$.

Example 6.4 $((\lambda x.(x^a I)^b)^c(\lambda y.A)^d)^e \to_{\beta_L} ((\lambda y.A)^{\underline{dca}} I)^{\underline{bce}}$

Here we just mention some of the most salient facts of λ_L.

1. It is an orthogonal CRS, hence CR.
2. The simply labeled calculus $\lambda_{\mathcal{A}}$ can be obtained as a projection of λ_L. Namely, replacing each label in a reduction in λ_L by its first symbol results in a reduction in $\lambda_{\mathcal{A}}$.
3. Descendant redexes have the same degree as their ancestor redex.
4. Created redexes have a degree higher than that of the creator redex (that is the redex contracted in the creating reduction step).
5. Bounded reduction is SN: suppose reduction is only allowed if the height of the degree of the contracted redex is $\leq N$. Call the resulting rewrite system λ^N, the N-bounded fragment of λ_L; this still is an orthogonal CRS. Now λ^N satisfies SN for all N. Note that for $N = 0$ this is just FD.
6. Given a labeling I of M_0, a reduction $\sigma : \quad M_0 \to \cdots \to M_n$ can be uniquely lifted to $\lambda_L : M_0^I \to \cdots \to M_n^J$.
7. Reductions $\sigma : \ M_0 \to \cdots \to M_k$ and $\tau : N_0 \to \cdots \to N_l$ with $M_0 \equiv N_0$ and $M_k \equiv N_l$ are Lévy-equivalent if after lifting to $M_0^I \to \cdots \to M_k^J$ and $N_0^I \to \cdots \to N_l^{J'}$, respectively, we have $J = J'$.

7 Origin tracking in λ-calculus

In this section we will show how a refined notion of descendant, which we arrive at via Lévy-labels, can be applied to yield a perspicuous proof of a well-known, 'classical' lemma in λ-calculus, the Genericity Lemma. In Barendregt [Bar84] (p. 374, Prop. 14.3.24) it reads:

Lemma 7.1 *Let $U, N \in \mathrm{Ter}(\lambda)$, with U unsolvable and N a normal form. Then for every context $C[\,]$:*

$$C[U] =_\beta N \;\Rightarrow\; \forall Q \in \mathrm{Ter}(\lambda)\; C[Q] =_\beta N.$$

Barendregt [Bar84] gives an elegant high level abstract proof, referring to the tree topology. In this topology M is an isolated point (i.e. $\{M\}$ is an open set) iff M is a normal form; and M is a compactification point (i.e. the whole space $\mathrm{Ter}(\lambda)$ is the only neighbourhood of M) iff M is unsolvable. Using the fact that for every context $C[\,]$ the function $M \mapsto C[M]$ is continuous, the Genericity Lemma follows immediately.

An early proof, however only for CL (Combinatory Logic), is in Barendregt's Ph.D. thesis [Bar71]. There are several other proofs of the Genericity Lemma: Takahashi [Tak95], Kuper [Kup94], Kennaway et al. [KOV99] and others. Our interest here is primarily in the method employed.

The idea is, given a reduction $C[U] \twoheadrightarrow_\beta N$, to trace the symbols of N all the way back to $C[U]$. It will turn out that we will find a prefix of $C[U]$ as the origins of the symbols in N; this 'useful' prefix is followed by a lower part that is 'garbage', i.e. can be replaced by arbitrary terms $Q_1, Q_2, \ldots$ without altering the normal form N. It will moreover turn out that the useful prefix is independent of the actual reduction from $C[U]$ to N.

Let us first observe that the classical notion of descendant, as defined in Definition 4.2, does not yield these desiderata. Consider Figure 18. Here N

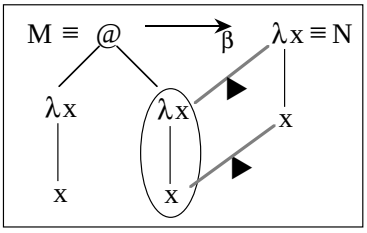


Figure 18: Failure of prefix property for ▶

traces back to the encircled part of M; but this is not a prefix of M (which by definition is an upward closed part of the term formation tree).

We now apply Lévy's λ_L: given $M \to N$ and symbol occurrences $s \in M, t \in N$ we define $s \rhd t$ (t is a *dynamic descendant* of s) as follows.

Definition 7.2 Give M an initial labeling I and lift the step $M \to N$ to the labeled step $M^I \to N^J$. Now s^a in M^I traces ($\rhd$) to all symbols t^α in N^J such that $a \in \alpha$ (a occurs in α). Then we project this relation down again to the original step $M \to N$.

We call the inverse $\lhd$ of the dynamic descendant relation the *origin* relation: if $t \lhd s$, then t is an *origin* of s.

Note that we have $p \blacktriangleright q \Rightarrow p \rhd q$, but not conversely. For an example see Figure 19.

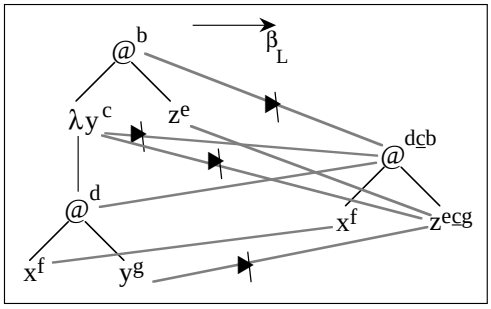


Figure 19: $((\lambda y.(x^f\, y^g)^d)^c z^e)^b \to_{\beta_L} (x^f\, z^{e \underline{c} g})^{d \underline{c} b}$

Remark 7.3 Khasidashvili [Kha90] uses a notion of descendant that also extends the classical notion $\blacktriangleright$. In his definition the contractum $M[x := N]$ of a redex $(\lambda x.M)N$ has as origins the redex itself, as well as the function part $\lambda x.M$ and its body M. These are also origins in our definition.

Proposition 7.4 *Let $M \to_\beta M' \equiv C'[N_1, \ldots, N_k]$, so $C'[\ , \ldots, \ ]$ is a prefix Π of M'. Then the original of Π in M with respect to $\rhd$ (notation $\lhd\Pi$) is again a prefix. See Figure 20.*[10]

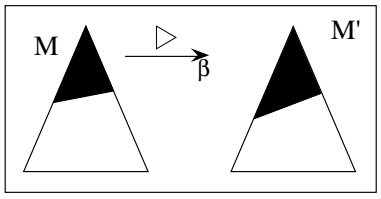


Figure 20: Prefix property of $\rhd$

Proof. Let M have an initial labeling. Consider an occurrence of atomic label a in M; let π be the path leading to it.

Now consider the position of the β-redex, given by its pattern (i.e. its @-node with left-successor the λ-node) relative to the path π. We have the following cases, of which we treat only 1 and 2. The third case is similar and left to the reader.

Case 1. @ is not on π, hence also λ is not on π.

Case 2. @ is on π, but λ is not.

Case 3. @ and λ are both on π.

See Figure 22.

Ad case 1. (See Figure 23)

[10]There is an analogy with the property "invert" studied by van Oostrom [Oos97b]. He noted that it is related to earlier results that were obtained in the context of the Automath project, especially van Daalen's "square bracket lemma", and to the folklore "Barendregt's lemma". See [NGV94].

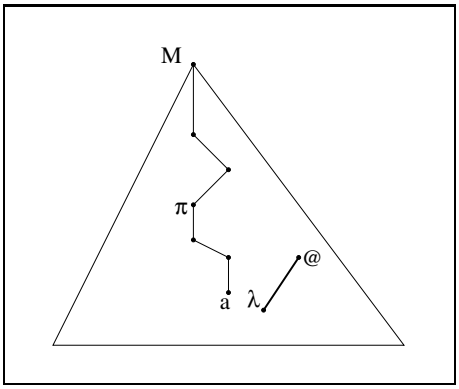


Figure 21: Relative position of label a and contracted redex

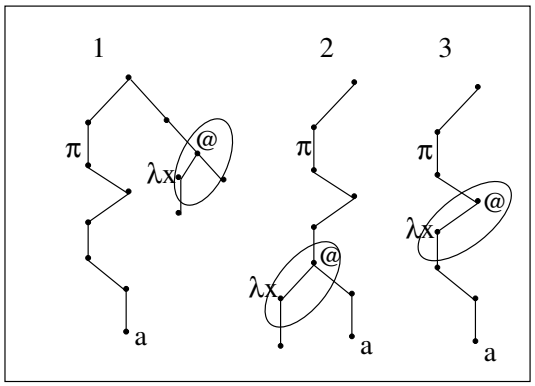


Figure 22: Case distinction

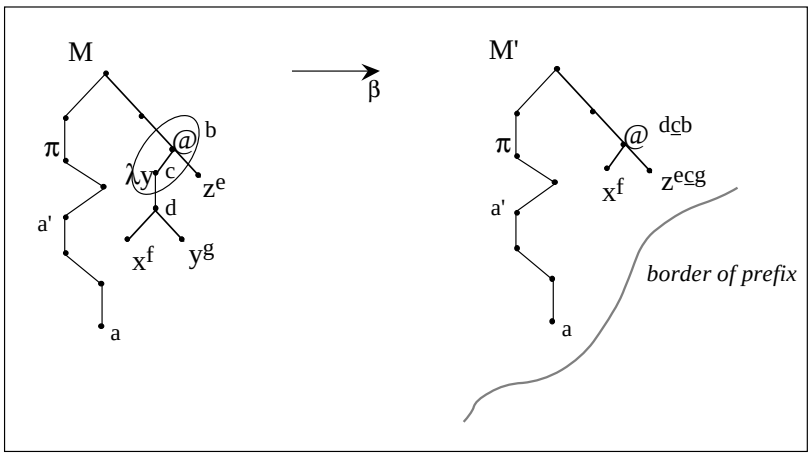


Figure 23: $((\lambda y.(x^f y^g)^d)^c z^e)^b \to (x^f z^{e\underline{c}g})^{d\underline{c}b}$

Consider a prefix Π of M', and let (the occurrence in M with) label a be in $\lhd\Pi$, the $\rhd$-original of prefix Π in M. We have to prove that all labels between a and the root of M (e.g. a') are also in $\lhd\Pi$. This is in this situation trivial, since the prefix border in M' (see Figure 23) must be below a.

Ad case 2. We only treat a typical case (see Figure 24).

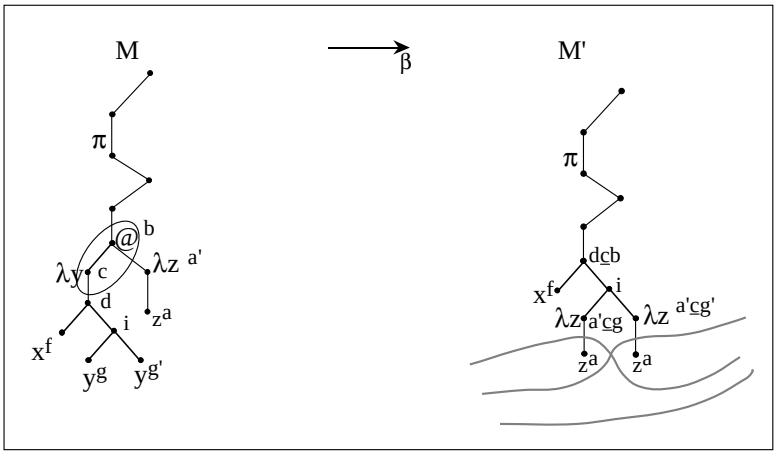


Figure 24: $((\lambda y.(x^f(y^g y^{g'})^i)^d)^c(\lambda z.z^a)^{a'})^b \to (x^f((\lambda z.z^a)^{a'\underline{c}g}(\lambda z.z^a)^{a'\underline{c}g'})^i)^{d\underline{c}b}$

Label a is in the $\rhd$-original of prefix Π in M. So Π must contain a copy of z^a, at least one. In the picture there are 3 possibilities, Π_1, Π_2, Π_3, containing respectively 1, 1, 2 occurrences of z^a. In all, it is clear that the labels $a', \ldots$ above a in M also are in $\lhd\Pi_i$ $(i = 1, 2, 3)$.

□

Proposition 7.5 *Let $M \twoheadrightarrow_\beta N$, where N is a normal form. Then the original $\lhd N$, a prefix of M, is independent of the actual reduction from M to N.*

Proof. If N is a normal form in λ, then each labeled version N^J is a normal form in λ_L. Orthogonality of λ_L implies uniqueness of normal forms. Hence if $M^I \twoheadrightarrow N^J$ (for I initial) on the one hand, and $M^I \twoheadrightarrow N^{J'}$ on the other hand, we have $N^J \equiv N^{J'}$, in particular $J \equiv J'$. Therefore $\lhd N$ in M is for both reductions the same as it only depends on the initial labeling I and the final labeling J (cf. the same situation in Section 8). □

Remark 7.6 Let $M \twoheadrightarrow_\beta N$, where N is not necessarily a normal form. Let Π be a prefix in N and $\lhd\Pi$ its ancestor prefix in M. Now $\lhd\Pi$ is dependent on the actual reduction $M \twoheadrightarrow_\beta N$.

An example is: $M \equiv (\lambda z.z)(\lambda y.y)x \twoheadrightarrow (\lambda a.a)x \equiv N$. Depending on whether the λz- or the λy-redex is contracted, we find different prefixes in M as original of the prefix indicated by underlining in N: $\underline{(\lambda a.a)x}$. Namely: with respect to the contraction of the λy-redex, the original is everything in M but the nodes λz and z; with respect to the contraction of the λz-redex the original is everything but the nodes λy and y.

Now on the basis of Proposition 7.5 we can define the *useful prefix* of a term having a normal form. For reasons that will become clear in Section 8 the useful prefix is also called *needed prefix*.

Definition 7.7 Given a term M with normal form N, the prefix $\lhd N$ is the *useful prefix* of M. The rest of M is called *garbage*.

The essence of the next three propositions is that in a reduction to normal form only the useful prefix is relevant. In whatever way the garbage part is changed, the normal form stays. We will be very sketchy. In Section 8 analogous results are proved, in much more detail, for the case of first order term rewriting systems. The reasoning here would be similar.

The essential technique is that of cutting off garbage (i.e., parts of the term that are below the needed prefix) with Ω's. It is essential that we cut off *only* garbage. That is, we only consider $M \geq_\Omega N$ where $M \equiv C[N_1, \ldots, N_k]$, $N \equiv C[\Omega, \ldots, \Omega]$ and $N_1, \ldots, N_k$ are in the garbage part of M.

Furthermore it is essential that the needed prefix of a term M is 'redex pattern closed' (i.e. when it contains the root of a redex in M, then it contains the entire redex, cf. Definition 8.19 and Proposition 8.20). The effect is that replacing the garbage by Ω's never cuts a redex in two.

Proposition 7.8 *[Cutting off a reduction] Let $M \equiv C[N_1, \ldots, N_k] \twoheadrightarrow N$, where $C[\ , \ldots, \ ]$ is the useful prefix of M and with N in normal form. Then we can 'cut off' this reduction to a reduction using only the useful prefix of M, disposing of the garbage: $C[\Omega, \ldots, \Omega] \twoheadrightarrow N$.*

Proof. The proof is similar to the proof of Propostion 8.25 in Section 8. □

Proposition 7.9 *Ω-refinement commutes with β-reduction. See Figure 25.*

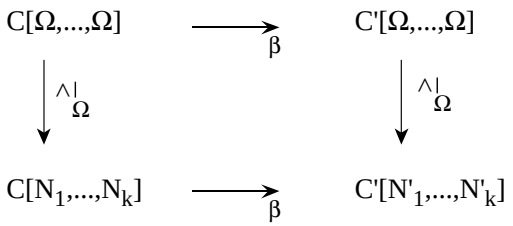


Figure 25: Ω-refinement commutes with β-reduction

(Note that the $\vec{\Omega}$ in $C'[\vec{\Omega}]$ that are refined to $C'[\vec{N'}]$ are the descendants of the $\vec{\Omega}$ in $C[\vec{\Omega}]$.)

Now we can prove the garbage property.

Proposition 7.10 *[Garbage property] Everything below the useful prefix found by tracing back the normal form to the original term, can be replaced by whatever terms without altering the normal form. See Figure 26.*

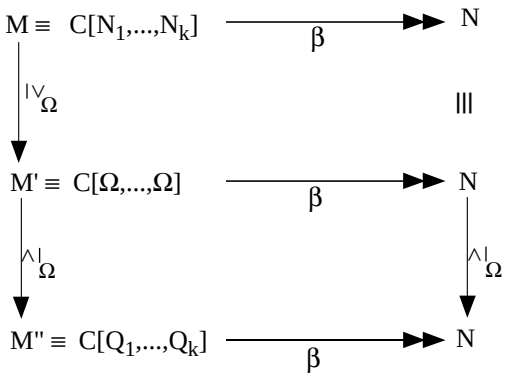


Figure 26: Proving the garbage property

Proof. Let $M \equiv C[N_1, \ldots, N_k] \twoheadrightarrow N$, where $C[\ ,\ldots,\ ]$ is the useful prefix of M and with N in normal form. By cutting off this reduction we get $C[\Omega, \ldots, \Omega] \twoheadrightarrow N$. Then we can, for arbitrary $Q_1, \ldots, Q_k$ by repeatedly applying Proposition 7.9, refine this reduction to $C[Q_1, \ldots, Q_k] \twoheadrightarrow N$. Note that the refining does not affect N, since N is Ω-free. See Figure 26. □

Before turning to the Genericity Lemma we need one more fact.

Proposition 7.11 *Let U be unsolvable, and $M \equiv C[U] \to_\beta M' \equiv C'[U']$ with $U \rhd U'$. Then U' is unsolvable.*

Proof. Easily obtained from the fact that an unsolvable stays so after internal reduction, after deletion of a λx at the root, and also after substitution. □

Theorem 7.12 *[Genericity Lemma] If $C[U] \twoheadrightarrow N$, with U unsolvable and N a normal form, then $C[Q] \twoheadrightarrow N$ for any term Q.*

Proof. Suppose $M \equiv C[U] \twoheadrightarrow_\beta N$, with U unsolvable, N a normal form. Trace N back to M; result $M \equiv D[N_1, \ldots, N_k]$, where D is the useful prefix and $N_1, \ldots, N_k$ is the garbage part.

> *Claim*: U must be in the garbage part $N_1, \ldots, N_k$.
>
> *Proof of the Claim*: Suppose not, then the root of U is in the prefix $D[\ldots]$. By the definition of useful prefix, this root then is connected (via $\rhd$) to some symbol in N. Now along this $\rhd$-trace, U stays unsolvable. But then N contains an unsolvable subterm, in contradiction with the assumption that N is a normal form.

Now the garbage property applies and we are done. □

8 Origin tracking in first-order rewriting

After the preceding exercise in origin tracking in the λ-calculus, we now turn to a similar enterprise in first-order term rewriting. The main device will be a labeling system inspired by the Lévy labels for λ-calculus. The theorem that we now address with this method is the classical one of Huet and Lévy concerning needed reduction.

8.1 The theorem of Huet and Lévy

We adopt the framework of orthogonal first-order term rewriting systems (for details we refer e.g. to [Klo92]). So we may assume confluence of the reduction relation and uniqueness of normal forms: each term can have at most one normal form.

The theorem of Huet and Lévy [HL91] concerns the notions of needed redex and needed reduction.

Definition 8.1 A redex in t is *needed* if in every reduction from t to its normal form, some descendant of that redex must be contracted. A reduction is needed if in each reduction step a needed redex is contracted.

Example 8.2 Consider the well-known orthogonal TRS for addition and multiplication on natural numbers generated by 0 and S. The reduction rules are given in Table 5. Now the redex $A(0,0)$ in the term $M(A(0,0),0)$ is not needed. It is erased in the

ρ_1	$A(x,0)$	$\to$	x
ρ_2	$A(x,S(y))$	$\to$	$S(A(x,y))$
ρ_3	$M(x,0)$	$\to$	0
ρ_4	$M(x,S(y))$	$\to$	$A(M(x,y),x)$

Table 5: The TRS for addition and multiplication

reduction to normal form consisting of the single reduction step

$$M(A(0,0),0) \to_{\rho_3} 0.$$

Theorem 8.3 *[Huet and Lévy]*

1. *Consider a term t having a normal form. If t is not a normal form itself, at least one of its redexes is a needed redex.*

2. *Repeatedly contracting needed redexes must lead eventually to the normal form, provided the original term has a normal form. In other words, needed reduction is a* normalizing *reduction strategy.*

3. *Needed reduction is not only normalizing, but even* hyper-normalizing*: there does not exist an infinite reduction of t containing infinitely many steps in which a needed redex is contracted. In other words, even the relaxed notion of needed reduction which allows between needed reduction steps any finite number of arbitrary reductions is normalizing.*

Without putting further restrictions on orthogonal term rewriting systems, this result is nice, but not necessarily very useful: we cannot always determine what the needed redexes are. The notion of needed redex is undecidable. However, Huet and Lévy [HL91] gave reasonable restrictions that ensure the decidability. Here we will not discuss this matter. What we are aiming at is a proof of this general theorem by means of origin tracking.

Remark 8.4 Note that if t has no normal form, every redex in t is trivially needed. Hence the present concept of neededness is not useful for terms without normal form, even though such terms may be very informative (e.g. computing an infinite stream of integers). Therefore Khasidashvili [Kha93] developed the notion of *essential* instead of needed, to cope with this situation. Middeldorp [Mid97] defined a variant of neededness called *root-neededness* that is also adequate for terms without normal form, and proved generalizations of Huet and Lévy's theorem for the setting of infinitary rewriting. In this section we stick to the theorem as stated, since the point we are presently discussing is about the method of proof, namely origin tracking.

8.2 The needed prefix

Consider a term t and a reduction $t \equiv t_0 \to t_1 \to \cdots \to t_n$ $(n \geq 0)$, where t_n is a normal form, and let us try to determine the part of t_0 that has been 'necessary' in manufacturing t_n. The idea is to look at each symbol s in t_n, and to determine what symbols $s', s'', \ldots$ in t_{n-1} were 'responsible' for the occurrence or appearance of s in t_n. Here 'responsible' is in a wider sense than the classical descendant-ancestor notion; also the symbols in the redex pattern are responsible for creating the situation after the redex contraction. The precise definition follows below.

The symbols s', s'' can also be viewed as the 'causes' or 'origins' of s. In turn, we trace back the symbols $s', s'', \ldots$ to the previous term t_{n-2}, and so on. In the end we arrive at a bunch of symbol occurrences in t_0 that are the original causes of the symbol s in t_n. Doing this for all symbols in t_n and taking all the 'origins' together, we have what we call the 'needed part' of t_0. Actually we will find that the situation can be summarized as follows.

> The origins in the original term $t \equiv t_0$ of all symbols in t_n will make up a prefix of t.

It will be called the *needed prefix* of t, since all redexes having their pattern in the needed prefix are in fact needed. Moreover, if t is not a normal form, the needed prefix will contain at least one redex (or rather a redex pattern). Finally, everything in the non-needed, dark part of t is *garbage*; it can be replaced by anything without affecting the normal form t_n. The situation is depicted in Figure 27.

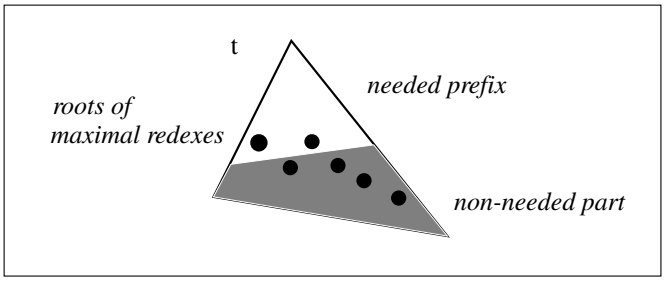


Figure 27: Needed prefix

8.3 Tracing back

How to define the tracing or tracking relation to find the s', s'' in t_{n-1} that are the *origins* of s in t_n? By way of example we have visualized in Figure 28(a) the trac(k)ing relation between symbols in the reduction step

$$S(F(G(0,1),H)) \to S(R(S(1),1)),$$

obtained from the reduction rule

$$\rho:\ F(G(x,y),H) \to R(S(y),y).$$

We now describe the intended tracing relation more precisely. Let $t(\vec{x})$ and

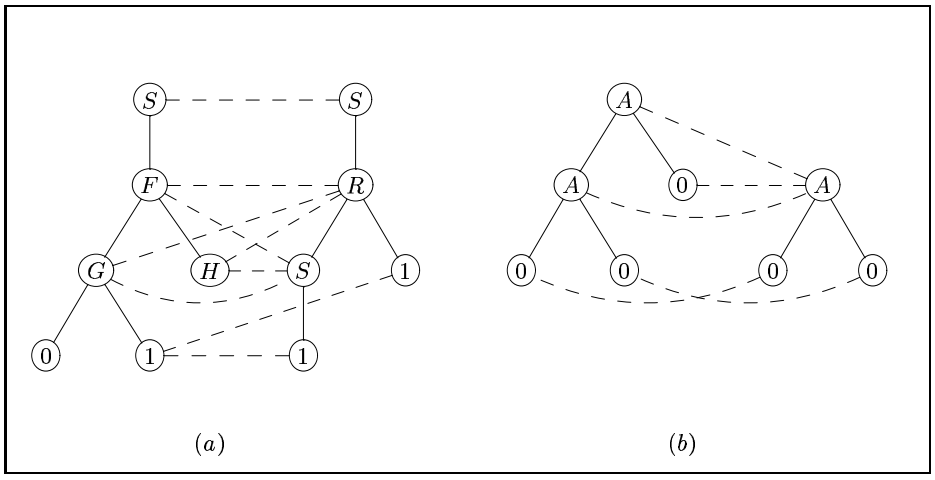


Figure 28: Tracing symbols

$s(\vec{x})$ be terms involving the variables $\vec{x} = x_1, \ldots, x_n$ and let $\rho:\ t(\vec{x}) \to s(\vec{x})$ be a rewrite rule. We call the context $t(\vec{\ })$ obtained by replacing in t the variables $\vec{x}$ by n holes the *redex pattern* of the rewrite rule ρ, and $s(\vec{\ })$ the *contractum pattern* of ρ.

Example 8.5 So in the rule $\rho:\ F(G(x,y),H) \to R(S(y),y)$ we have as redex pattern the context $F(G(\Box,\Box),H)$ and as contractum pattern the context $R(S(\Box),\Box)$.

Moreover, let $C[t(\vec{x}^\sigma)] \to C[s(\vec{x}^\sigma)]$, be a rewrite step generated by ρ. (Here $C[\,]$ is a context and σ is a substitution.) There are three cases for the position of a symbol to be traced:

Definition 8.6

1. A symbol in the context $C[\,]$ of the left-hand side of the rewrite step traces to the same symbol in the right-hand side $C[\,]$. So in Figure 28(a) the two top S's are connected.

2. A symbol in x_i^σ in the left-hand side of the rewrite step traces to the same symbol in all the copies of x_i^σ in the right-hand side. So in Figure 28(a) the 1 in the left-hand side is connected to both 1's in the right-hand side; the 0 is not connected to anything, it is erased.

3. A symbol in the redex pattern traces to all the symbols of the contractum pattern. In the figure each of the symbols F, G, H is connected to both R and the lower S in the right-hand side.

If symbol s in the left-hand side connects to t in the right-hand side, we say that s is the *origin* of t, and t is a *dynamic descendant* of s. Notation $s \rhd t$.

8.4 Collapsing rules

As often in term rewriting, collapsing reduction rules require special attention. A reduction rule $l \to r$ is called *collapsing* if the right-hand side r is a mere variable.

Example 8.7 The rule $A(x, 0) \to x$ from Table 5 is collapsing. And so is the rule for the combinator K from Combinatory Logic, $Kxy \to x$.

The definition above of the tracing relation does not provide for this situation, since in a collapsing rule there is no contractum pattern. It is the trivial or empty context $\Box$. So where to attach the traces leaving the symbols in the redex pattern? We extend Definition 8.6 with a fourth clause:

Definition 8.8

4. If $C[t(\vec{x}^\sigma)] \to C[x_i^\sigma]$ is a collapsing step, then all symbols in the redex pattern $t(\vec{\ })$ are connected with the top symbol (the root) of x_i^σ.

In Figure 28(b) we depicted the tracing relation between symbols in the reduction step that consists of contraction of the redex $A(A(0, 0), 0)$ with the collapsing reduction rule $A(x, 0) \to x$.

As we did for the λ-calculus in Section 7 we prefer to work with an algebraic characterization of the origin relation, in terms of a labeling system. It will be introduced in the next section.

Remark 8.9 The tracing definition (1-4) was suggested in Klop [Klo90]. A rather similar notion of trace has been defined by Boudol [Bou85]; it lies somewhere between the classical descendant notion and the present dynamic descendant notion. In Boudol's definition each symbol in the redex pattern traces to the top of the contractum. Note that this entails that the set of origins of a symbol may be empty, in contrast with the present definition. The same definition occurs in the work of Khasidashvili [Kha93]. Another study of origin tracking is by Bertot [Ber92], who uses 'origin functions'. Maranget, in his Ph.D. Thesis [Mar92] uses a labeling device, just as we do in the next section. It is somewhat different from ours, but also derived from Lévy labels for λ-calculus.

8.5 Labels

We present an algebraic formulation, in the syntax of term rewriting systems itself, of the origin relation. It is inspired by Lévy's labeled lambda calculus, already discussed extensively in Section 6. The actual labelings that we employ were introduced in Klop [Klo80].

The Lévy labels were defined in Definition 6.1. Recall that they are formed from atomic labels $a, b, c, \ldots$, using the operations of concatenation and underlining. That is, if α is a label, then $\underline{\alpha}$ is a label and if α, β are labels, then $\alpha\beta$ is one. E.g. $\underline{\underline{abc}}ad$ is a composite label. In our notation labels will be attached to symbols as superscripts, but their actual status is that of unary function symbols.

We are now going to decorate rewrite rules with labels. Consider the rule $F(G(x,y),H) \to R(S(y),y)$ above. For every label α, β, γ we will have the labeled version:

$$F^{\alpha}(G^{\beta}(x,y),H^{\gamma}) \to R^{\underline{\alpha\beta\gamma}}(S^{\underline{\alpha\beta\gamma}}(y),y).$$

So, every symbol in the redex pattern has some label. All these labels are swept together (say in order of appearance) and underlined. This new label is then attached to all symbols in the contractum pattern.

Now if R is an orthogonal TRS, then R^L will be the labeled TRS consisting of all labeled versions of the rewrite rules of R. We note that R^L is again an orthogonal TRS, because an overlap of the labeled rules will yield at once an overlap of the unlabeled rules after omitting the labels.

As before, we will use the notation t^I for a term t in R with labeling I. The labeling I can be perceived as a map from the symbol occurrences of t to the set of labels. So t^I is a term in R^L. Labelings will be denoted by $I, J, \ldots$.

We can now give a precise definition of the tracing relation between symbols in a rewrite step $t \to s$. It is the analogue of the previous Definition 7.2, that we used in the λ-calculus case.

Definition 8.10 Provide t with an *initial* labeling I, that is, a labeling where each symbol of t gets an atomic label such that different symbol occurrences get different labels. The result is the labeled term t^I. We then *lift* the rewrite step to the labeled TRS R^L, obtaining a labeled step $t^I \to s^J$.

Now we stipulate that a symbol p^a in t^I *traces* to all symbols q^{α} in s^J such that $a \in \alpha$. This tracing relation is then projected down again to the original unlabeled rewrite step $t \to s$. That is, if the symbols p in t, q in s correspond qua position to the labeled symbols p^a and q^{α}, then p traces to q. We also say that, or that q *traces back* to p, or p is an *origin* of q .

Notation 8.11 We use the notations $\rhd$ and $\lhd$ for the tracing relation and its inverse, the origin relation. That is, we write $p \rhd q$ or, equivalently, $q \lhd p$, if q traces back to p. If $t \to s$ and p, q are symbols in t, s respectively, then $\lhd q$ is the set of symbols in t to which q traces back. Likewise if Q is a set of symbols in s, $\lhd Q$ is the union of the origins in t of all $q \in Q$.

The following example demonstrates that for a non-collapsing rule the present algebraic definition of $\rhd$ by means of labels indeed yields the same notion as the one described in the more verbose way in Definition 8.6.

Example 8.12 Consider again the rule $\rho: F(G(x,y),H) \to R(S(y),y)$, yielding the rewrite step in Figure 28:

$$t \equiv S(F(G(0,1),H)) \to S(R(S(1),1)) \equiv s.$$

Provide the left-hand side t with an initial labeling:

$$S^a(F^b(G^c(0^d,1^e),H^f)).$$

We now apply the following labeled version of ρ:

$$F^b(G^c(x,y),H^f) \to R^{\underline{bcf}}(S^{\underline{bcf}}(y),y)$$

yielding the labeled step

$$t^I \equiv S^a(F^b(G^c(0^d,1^e),H^f)) \to S^a(R^{\underline{bcf}}(S^{\underline{bcf}}(1^e),1^e)) \equiv s^J.$$

Inspection of the labels clearly shows what pairs of symbols (p,q) in the corresponding unlabeled reduction step are in the relation $\rhd$ and it is easily checked that this is just the relation described in Definition 8.6.

So our algebraic approach works for non-collapsing rewrite rules. However, it is a nasty technical problem to extend the tracing definition by means of labels as above to the case of collapsing rules, while still retaining an orthogonal TRS. We are not aware of a solution that is both simple and natural. There are some tricks to eliminate the problem, however. The method we choose is to code the collapsing rules away. This is done by replacing e.g. the rule $A(x,0) \to x$ by $A(x,0) \to \varepsilon(x)$, where ε is a unary 'dummy' symbol. Then one has to add infinitely many new reduction rules, saturating' all left-hand sides with the symbol ε. The reader is referred to Appendix C.

Remark 8.13 Up to now we have only defined the tracing relation $\rhd$ for symbols in begin and end of a single reduction step $t \to s$. We would like to do this also for many-step reductions $t \twoheadrightarrow s$, or more explicitly, $t \equiv t_0 \to \cdots \to t_n \equiv s$. This is very simple: we extend $\rhd$ by transitivity in the obvious way. There is however another way as follows. Give t an initial labeling t^I and lift the reduction $t \twoheadrightarrow s$ to the labeled reduction $t^I \twoheadrightarrow s^J$. Now define as before that a symbol p in t traces to q in s if and only if its label (in t^I) is included in the label of q (in s^J).

So the difference is that in the former definition tracing is defined by repeated initialization of the labels: in each step the labels are 'refreshed' to an initial labeling. Fortunately we can without much effort prove that both ways yield the same. In other words, repeated initialization is superfluous. The proof is given in Appendix D.

This remark has an important consequence. Given a reduction $t \twoheadrightarrow s$, with s a normal form, let us trace back symbol q in s to its origins in t. Now the question is whether the set of origins depends on the actual reduction from t to s.

Having the second definition of $\rhd$ in mind (direct comparison of an initial labeling of t with the resulting labeling of s) we can now state that the set of

origins is independent from the actual intermediate reduction. To see this we first observe that if s is a normal form (in R), then each labeled version s^J is a normal form in R^L. Now orthogonality of R^L implies uniqueness of normal forms. Hence if $t^I \twoheadrightarrow s^J$ on the one hand, and $t^I \twoheadrightarrow s^{J'}$ on the other hand, we have $s^J \equiv s^{J'}$, in particular $J \equiv J'$. Therefore $\lhd q$ in t is for both reductions the same, as it only depends on the initial labeling I and the final labeling J.

By these observations it follows that the following is a sound definition.

Definition 8.14 Let $t \twoheadrightarrow s$, where s is a normal form. Then the set of positions in t that are the origins of s is called $\pi(t)$.

Remark 8.15

1. Note that traces can stop in forward direction. That is, if $t \to s$ and p is a symbol in t, the set $\rhd p$ (i.e. the set of p's dynamic descendants) may be empty. But in backward direction traces do not stop. Everything has an origin—symbols are not created out of the blue.
2. Note also that a redex has no ordinary descendants (as defined below, notation $\blacktriangleright$) after its contraction. Further note that the assumption of orthogonality yields that if $p \blacktriangleright q$ and p is a redex root, then so is q. For $p \rhd q$ the analogous statement does not hold.
3. Usually the notion of descendant is seen as a relation between subterm rather than symbol occurrences. But since subterms and their roots are in 1-1 correspondence, defining the relation on symbols, as we do here, amounts to the same thing.

8.6 Ordinary descendants and simple labels

With $\rhd$ and $\lhd$ we are able to follow symbols forwards and backwards through a reduction. But in a different way than according to the classical, standard descendant/ancestor relation. Let us compare the two approaches. Note that everything remains in perfect analogy with what we did before for λ-calculus, cf. Section 4, Definition 4.2.

Again we use a labeled system to define ordinary descendants. However, now only simple labels are allowed: $\epsilon, a, b, c, \ldots$. Here $a, b, c, \ldots$ are proper labels, they are single letters, and ϵ is the empty label.

Again we decorate rules with labels. Rule ρ gives now rise to all labeled versions

$$F^\alpha(G^\beta(x, y), H^\gamma) \to S(R(y), y).$$

Now the classical descendant relation $\blacktriangleright$ is defined analogously to the definition of $\rhd$ with this difference: $p \blacktriangleright q$ if and only if p and q have the same proper labeling.

We have $p \blacktriangleright q \Rightarrow p \rhd q$, but not vice versa. For occurrences of variables $x, y, z, \ldots$, and the symbol Ω that will be used later, the two notions are identical. In Tip [Tip95] the notion $\rhd$ is called *dynamic dependence tracking*.

8.7 The prefix property

Consider again the reduction $t_0 \to t_1 \to \cdots \to t_n$ with t_n in normal form. We will prove that $\lhd t_n$ in t_0, that is, all the origins of t_n traced back to t_0 constitute a prefix π of t_0. A *prefix* of a term is a set of occurrences that is upward closed. (So in the pictures, an upper 'half' of the triangle if terms are written as trees.) We will denote prefixes of terms by π, etc.

Notation 8.16 If p, q are symbol occurrences, then $p \leq q$ means that p is above q or q itself. We call $\leq$ the *prefix ordering.*

Now we have the following proposition. (See Figure 29.)

Proposition 8.17 *The prefix order $\leq$ and the tracing relation $\rhd$ commute, in the sense that:*

$$s' \leq s \rhd t \;\Rightarrow\; \exists t'\; s' \rhd t' \leq t.$$

Proof. The proposition is easily proved by distinghuishing some cases. Let r be the contracted redex. Here we consider only the cases where s is in the redex pattern.

- If s' is above s in the redex pattern, we can just take any t' in the contractum pattern above t (or t itself). Then $s' \lhd t'$.
- If s' is above the redex pattern, then actually it is in the context of r and we take the corresponding t' (at the same position).

□

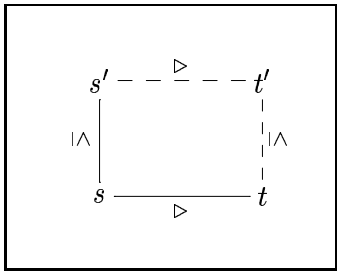


Figure 29: A commuting diagram yielding the prefix property

From the proposition we have at once that prefixes are preserved in tracing back: if $t \to s$, and π is a prefix of s, then $\lhd\pi$ is a prefix of t. Moreover, by transitivity we have the same if $t \twoheadrightarrow s$. In particular, when s is a normal form and for π we take the whole term s, this yields the prefix property: $\pi(s)$ is a prefix of t.

Remark 8.18 Actually, we have some immediate generalizations of the preservation of prefixes under $\lhd$. We make use of the following terminology.

A set of occurrences is *convex*, if with each two points it also contains all points in between in the sense of the prefix ordering. A convex set of occurrences in t can be characterized as a union $\pi_1 \cup \ldots \cup \pi_n$, with $\pi_1, \ldots, \pi_n$ prefixes of disjoint subterms $s_1, \ldots, s_n$ of t. Note that a prefix of t is any convex set containing the root of t.

A convex set is called a *slice*, if it is a prefix of a single subterm, i.e. a convex set with only one maximal occurrence (in the prefix order). Note that a singleton set of occurrences, a single point in a term, is also a slice.

The notion of prefix, slice and convex set have been depicted in that order in Figure 30(a, b, c). Now, first, if C is convex then $\lhd C$ is again convex. Secondly (and

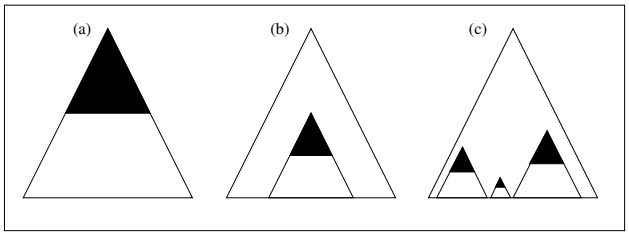


Figure 30: Prefix, slice and convex set of occurrences

this is not the same!) also slices trace back to slices under $\lhd$. This fact is used in 'program slicing', for the analysis of dependencies within programs and error recovery, by Field, Tip [FT94, Tip95] and others.

In the sequel we will need the following property of prefixes.

Definition 8.19 A prefix t' of a term t is *redex-pattern closed*, or for short, has the *rpc* property, if it contains redex patterns only in their entirety, and not 'half' of a redex pattern. In other words, if the prefix t' contains the root of a redex r in t, it must contain the whole redex pattern p of r. See Figure 31, where r' is the intersection of redex r and prefix t'.

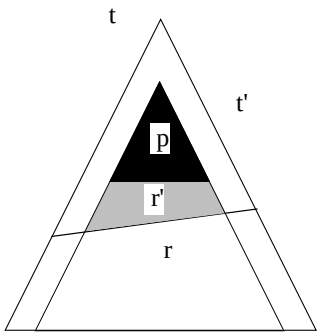


Figure 31: The rpc-property

Now we can strengthen our previous result that prefixes are preserved in tracing back.

Proposition 8.20 *All prefixes that we find by tracing back the normal form t_n, have the rpc property.*

Proof. That t_n itself, being its own prefix, has the rpc property, is trivial as it contains no redexes at all. The further proof is again by a simple analysis of cases, and will be omitted here. □

We have the following nice situation:

> All terms in the reduction graph of t_0 are partitioned in a 'white' prefix above and a 'dark' remainder. The white prefixes are made up from the origins of all the symbols in t_n. If t, s are reducts of t_0 and $t \twoheadrightarrow s$, then their white prefixes $\pi(t)$ and $\pi(s)$ are related by $\lhd\pi(s) = \pi(t)$.

This is illustrated in Figure 32, where the dark and white area's are indicated for some reducts of the original term t_0, including its normal form t_n.

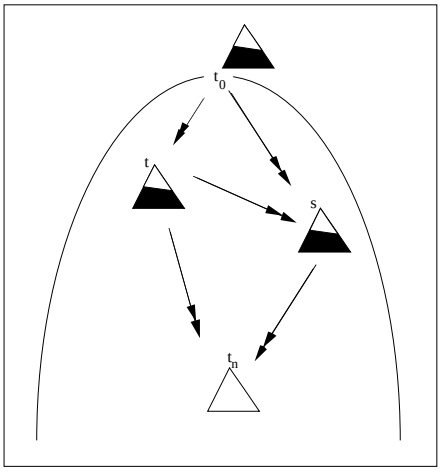


Figure 32: Reduction graph of t_0

In the sequel we will employ Ω-terms, i.e. terms where the constant Ω may occur. We will use Ω as demarcation of prefixes, by appending them at the cut-points of the prefix. More precisely, if the prefix $\pi(t)$ rendered as a multi-hole context is C, then $\pi(t)$ is identified with $C[\Omega, \ldots, \Omega]$, the result of placing Ω's at the open places.

Definition 8.21 Let t, s be reducts of t_0. Then $t \to_\Omega s$ if s results from t by replacing a subterm t' of t in the garbage part by Ω. (Here, in order to avoid vacuous reduction steps, it is assumed that not already $t' \equiv \Omega$.)

So, in particular we have that $t \twoheadrightarrow_\Omega \pi(t)$. Note that $\to_\Omega$ is SN and CR.

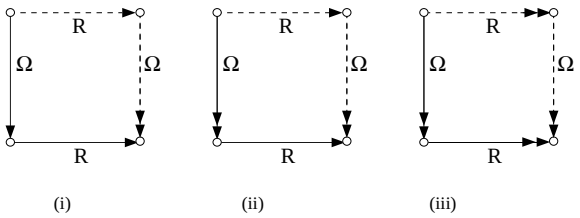


Figure 33: Ω-postponement

Proposition 8.22 *[Ω-postponement.] In a reduction involving $\to_R$-steps and $\to_\Omega$-steps, the $\to_\Omega$-steps can be postponed.*

Proof. See Figure 33. Diagram (i) is a routine check. Further (i) ⇒ (ii) ⇒ (iii). Then the result follows by simply permuting Ω- and R-parts of a reduction. □

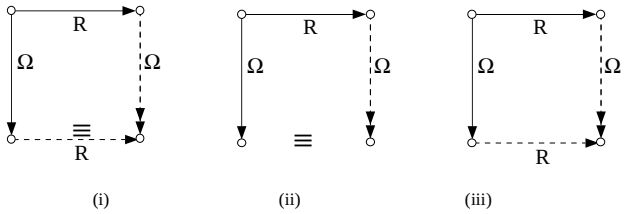


Figure 34: Commutation of Ω- and R-reduction

Proposition 8.23 *Ω- and R-reduction commute in the sense of Figure 34(i).*

Proof. We recall the rpc property (Proposition 8.20). It entails that either $R \subseteq \Omega$ or $\Omega \subseteq R$, where R, Ω denote the relevant R-redex and Ω-redex. So we have either diagram (ii) or diagram (iii) in Figure 34. □

Proposition 8.24 *Let $t \to s$. Then for some s' we have $\pi(t) \stackrel{\equiv}{\to}_R s' \twoheadrightarrow_\Omega \pi(s)$. See Figure 35.*

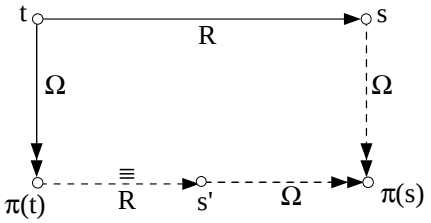


Figure 35: Projecting $\to_R$ over π

Proof. By employing Proposition 8.23 and CR for $\to_\Omega$ we obtain Figure 36. The Ω-reduction from $\pi(s)$ must be empty, since $\pi(s)$ is an Ω-normal form. (It is left to the reader to verify that all Ω-steps involved here are indeed 'garbage-collecting' steps.) Note that, as a matter of fact, $\pi(s') \equiv \pi(s)$, again since $\pi(s)$ is an Ω-normal form. □

Proposition 8.25 *Let $t \equiv t_0 \to \cdots \to t_n$, where t_n is in normal form. Then $\pi(t_0) \twoheadrightarrow t_n$.*

Proof. First, we have the upper part of Figure 37. Then by Ω-postponement we obtain the lower part: $\pi(t_0) \twoheadrightarrow_R s \twoheadrightarrow_\Omega t_n$ for some s. Since t_n is in normal form, it contains no Ω's, and thus the reduction $s \twoheadrightarrow_\Omega t_n$ is empty. So $\pi(t_0) \twoheadrightarrow_R t_n$. □

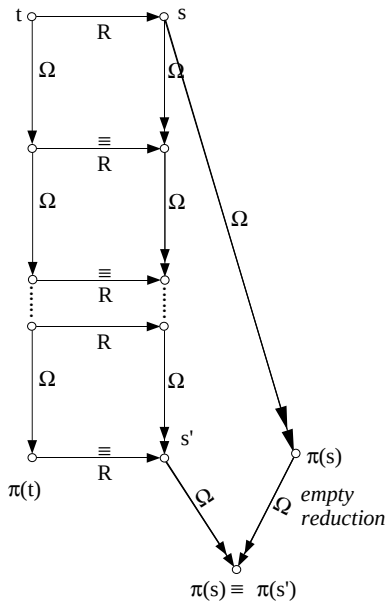


Figure 36: Proof of Proposition 8.24

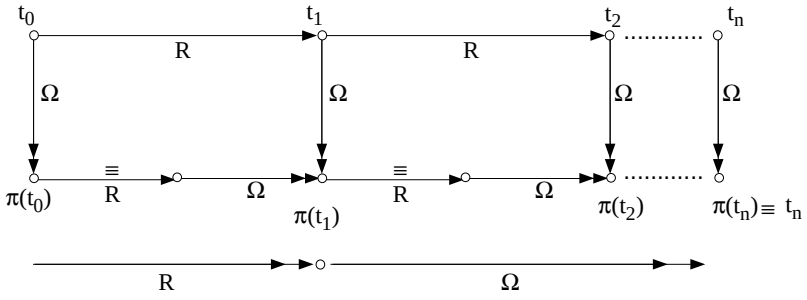


Figure 37: Projecting a reduction to the prefix

Remark 8.26 The way Proposition 8.25 is proved via Proposition 8.24 derives from an analysis of abstract rewriting in de Vrijer [Vri87, Vri89].

We have several consequences of Proposition 8.25.

Corollary 8.27 *Let t_0 be not a normal form. Then $\pi(t_0)$ contains a redex pattern.*

Proof. Suppose not. Then $\pi(t_0)$ is a normal form, so by Proposition 8.25 we have

$$\pi(t_0) \equiv t_n \qquad (i)$$

So $\pi(t_0)$ is Ω-free. But then the reduction $t_0 \twoheadrightarrow_\Omega \pi(t_0)$ is empty, i.e.

$$\pi(t_0) \equiv t_0 \qquad (ii)$$

From (i) and (ii) follows $t_0 \equiv t_n$. This contradicts the assumption that t_0 is not in normal form. □

Corollary 8.28 *The subterms 'under' the (white) prefix $\pi(t_0)$ (i.e. the dark part), are indeed garbage.*

More precisely: Let the prefix $\pi(t_0)$ be the Ω-term $C[\Omega, \ldots, \Omega]$, so $t_0 \equiv C[s_0, \ldots, s_n]$. Then for arbitrary $q_1, \ldots, q_n$ we have $C[q_0, \ldots, q_n] \twoheadrightarrow t_n$.

Proof. This follows readily from $\pi(t_0) \twoheadrightarrow t_n$ and the simple fact that Ω-refinement commutes with R-reduction. (Which actually is nothing more than substitutivity of R-reduction: if $t \to s$, then $t[\vec{x} := \vec{q}] \to s[\vec{x} := \vec{q}]$.) □

Corollary 8.29 *A redex r in the dark part is not needed.*

Proof. Let $\pi(t_0) \equiv C[\Omega_1, \ldots, \Omega_m]$, $t_0 \equiv C[s_0, \ldots, s_m]$, and $r \subseteq s_i$ for some i. To show that r is not needed we have to establish a reduction from t_0 to t_n in which no descendant of r is contracted. This is easy: just take the reduction $\pi(t_0) \twoheadrightarrow t_n$ found above in Proposition 8.25 and substitute $s_0, \ldots, s_m$ for $\Omega_1 \ldots, \Omega_m$. Clearly in this reduction all descendants of r stay 'at rest', actually they will all be erased on the way to t_n, but none of them is contracted. □

The proof of the following lemma is routine and omitted.

Lemma 8.30 *Let $t \to_{r^*} s$ be a reduction step in which redex r^* is contracted. Let $r \neq r^*$ be a different redex occurrence in t and let p be the head symbol of r (or any symbol in the pattern of r). Then for all symbol occurrences q in s we have: $p \rhd q \Leftrightarrow p \blacktriangleright q$.*

Corollary 8.31 *Any redex r in the prefix $\pi(t_0)$ is needed.*

Proof. Suppose not, then there is a reduction $t_0 \to t'_1 \to t'_2 \to \cdots \to t'_m \equiv t_n$ such that in some t'_k $(k \leq n)$ all $\blacktriangleright$-descendants must have vanished, not by contraction, but by erasure. Now consider the root symbol of r, call it p. Then for some symbol q in t'_k we must have $p \rhd q$; otherwise p was not in the prefix $\pi(t_0)$. Now, observing that in the considered reduction no descendant of r is contracted, Lemma 8.30 yields $p \blacktriangleright q$. But this contradicts the assumption that the redex headed by p has no $\blacktriangleright$-descendant in t'_k.

(More precisely: Observe that in the reduction we are considering no descendant of r is contracted. Lemma 8.30 now yields $p \blacktriangleright q$. But this contradicts the assumption that the redex headed by p has no $\blacktriangleright$-descendant in t'_k.)

□

8.8 Needed reduction is (hyper)normalizing

We will now show that repeated contraction of needed redexes must terminate, in the normal form, even when between needed contractions we contract some non-needed redexes. To this end we assign a norm $\|t\|$ to each term t in the reduction graph of t_0. First we define the norm $|\alpha|$ of a label: this is just the number of its symbols, counting an underlining as one symbol. Now $\|t\|$ is the sum of the $|\alpha|$ for every α in the prefix $\pi(t)$. Now (i) for a needed contraction $t \to s$ we have $\|t\| < \|s\|$, while (ii) for a non-needed contraction we have $\|t\| \leq \|s\|$. The proper increase in (i) is due to the fact that the labels attached in the contractum pattern are underlined. That in case (ii) no decrease is possible, is due to the fact that the non-needed redexes are in the dark part below the white prefix—so they cannot erase symbols and labels in the white prefix. From (i) and (ii) we immediately have termination as announced, since the norms are bounded by $\|t_n\|$, the norm of the normal form.

Remark 8.32 It is worthwhile to remark that we also have as an immediate corollary that parallel outermost reductions are normalizing as first proved in O'Donnell [O'D77]. This is seen by first noting that there must be a needed outermost redex, since neededness is preserved upward. If redex r is needed, and r' is a redex containing r as a subterm, then r' is needed. So one of the outermost redexes must be needed. Parallel outermost reduction therefore must be normalizing by the termination theorem just mentioned.

Remark 8.33 Actually, we can give a bound on the degrees of needed redexes and thereby obtain an alternative termination proof of needed reduction as follows. Here the *degree* of a redex $t(\vec{x}^{\sigma})$ is the concatenation of all labels in the pattern $t(\vec{\ })$, in the order of appearance. This definition is from Klop [Klo80], but is a straightforward generalization from Lévy's similar notion for the labeled lambda calculus as in Section 6.

Take an arbitrary reduction from t_0 to its normal form t_n. Assume that the set of degrees of needed redexes contracted in this reduction is $\{d_0, d_1, \ldots, d_m\}$. Then, for every reduct t' of t_0 we have that if a redex in t' is needed, it has as degree one of the $d_0, d_1, \ldots d_m$. (The converse does not hold.) Lévy [Lév75] proved that in labeled lambda calculus, bounded labeled reduction is terminating—or rephrased, that in every infinite reduction labels must grow unboundedly. This also holds in the present setting of first-order orthogonal rewriting. As a corollary we again have immediately the termination of needed reduction.

A precise treatment of this matter, the termination of needed reduction via termination of bounded reduction, is given in the Ph.D. Thesis of Maranget [Mar92]. He uses the method of recursive path orderings to prove termination of bounded reduction.

9 First-order infinitary rewriting

In this section we explain the development of infinite term rewriting as reported in Kennaway, Klop, Sleep & de Vries [KKSV95a], Klop & de Vrijer [KV91]. A complete formal treatment, including full proofs, can be found in [KKSV95a]. This work was stimulated by earlier studies of infinite rewriting by Dershowitz, Kaplan & Plaisted [DKP89] and Farmer & Watro [FW89].

As we will see, the crucial step in setting up a satisfactory framework for infinitary rewriting, namely establishing the notion of strong convergence, is induced by the very need to have a good concept of descendant.

Remark 9.1 There is ample motivation for a theoretical study of infinite rewriting, in view of the facility that several lazy functional programming languages such as Miranda ([Tur85]), Haskell ([Hud88]), Clean ([PvE93] have, enabling them to deal with (potentially) infinite terms, representing e.g. the list of all primes. Another motivation is the correspondence between infinite rewriting and rewriting of term graphs: a theory for infinite rewriting provides a foundation for a theory of term graph rewriting, since a cyclic term graph yields after unwinding an infinite term. Indeed, this correspondence has been the starting point for the work of Farmer & Watro [FW89].

Our starting point is an ordinary TRS (Σ, R), where Σ is the signature and R is the set of rewrite rules. In fact, we will suppose that our TRSs are orthogonal, just as in the previous section. Now it is obvious that the rules of the TRS (Σ, R) just as well apply to infinite terms as to the usual finite ones. First, let us explain the notion of infinite term that we have in mind. Let $\mathrm{Ter}(\Sigma)$ be the set of finite Σ-terms. Then $\mathrm{Ter}(\Sigma)$ can be equipped with the usual distance function d such that for $t, s \in \mathrm{Ter}(\Sigma)$, we have $d(t,s) = 2^{-n}$ if the n-th level of the terms s, t (viewed as labeled trees) is the first level where a difference appears, in case s and t are not identical; furthermore, $d(t,t) = 0$. It is well-known that this construction yields $(\mathrm{Ter}(\Sigma), d)$ as a metric space. Now infinite terms are obtained by taking the completion of this metric space, and they are represented by infinite trees. We will refer to the complete metric space arising in this way as $(\mathrm{Ter}^\infty(\Sigma), d)$, where $\mathrm{Ter}^\infty(\Sigma)$ is the set of finite and infinite terms over Σ.

A natural consequence of this construction is the emergence of the notion of Cauchy convergence as a possible basis for infinite reductions which have a limit: we say that $t_0 \to t_1 \to t_2 \to \cdots$ is an infinite reduction sequence with limit t, if t is the limit of the sequence $t_0, t_1, \ldots$ in the usual sense of Cauchy convergence. See Figure 38 for an example, based on a rewrite rule $F(x) \to P(x, F(S(x)))$ in the presence of a constant 0 in Σ. In fact, this notion of converging reduction sequence is the starting point for Dershowitz e.a. [DKP89]. In the sequel we will however adopt a stronger notion of converging reduction sequence which turns out to have better properties. First, let us argue that it makes sense to consider not only reduction sequences of length ω, but

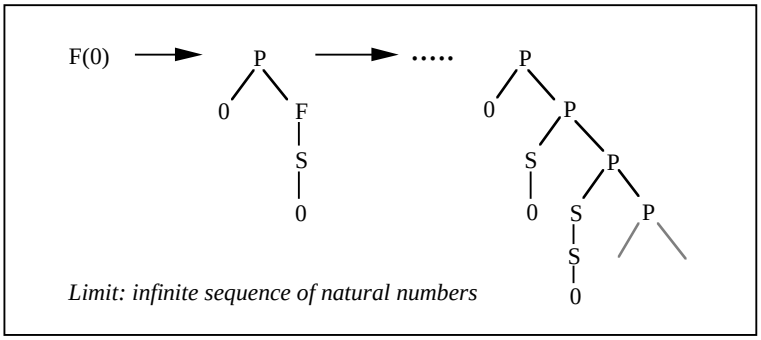


Figure 38: Generating the sequence of natural numbers

even reduction sequences of length α for arbitrary ordinals α. Given a notion of convergence, and limits, we may iterate reduction sequences beyond length ω and consider e.g. $t_0 \to t_1 \to t_2 \to \cdots \to t_n \cdots s_0 \to s_1 \to s_2 \to s_3 \to \cdots r$ where $\lim_{n\to\infty} t_n = s_0$ and $\lim_{n\to\infty} s_n = r$. See Figure 39 for such a reduction sequence of length $\omega + \omega$, which may arise by evaluating first the left part of the term at hand, and next the right part. Of course, in this example a 'fair' evaluation is possible in only ω many reduction steps, but we do not want to impose fairness requirements at the start of the theory development even though we may (and will) consider it to be a desirable feature that reductions of length α could be 'compressed' to reductions of length not exceeding ω steps, yielding the same 'result'.

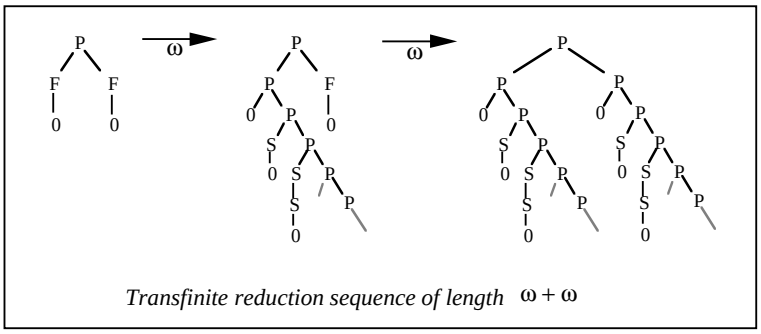


Figure 39: A transfinite reduction sequence

We will give a formal definition now.

Definition 9.2 Let (Σ, R) be a TRS. *A (Cauchy-) convergent R-reduction sequence of length* α (an ordinal) is a sequence $\langle t_\beta \mid \beta \leq \alpha \rangle$ of terms in $\mathrm{Ter}^\infty(\Sigma)$, such that

1. $t_\beta \to_R t_{\beta+1}$ for all $\beta < \alpha$,
2. $t_\lambda = \lim_{\beta<\lambda} t_\beta$ for every limit ordinal $\lambda \leq \alpha$.

Here 2. means: $\forall n \exists \mu < \lambda \forall \nu (\ \mu \leq \nu \leq \lambda \Rightarrow d(t_\nu, t_\lambda) \leq 2^{-n}\ )$.

Notation 9.3 If $\langle t_\beta \mid \beta \leq \alpha \rangle$ is a Cauchy-convergent reduction sequence we write $t_0 \to^c_\alpha t_\alpha$ ('c' for 'Cauchy').

The notion of normal form as a final result has to be considered next. We simply generalize the old finitary notion of normal form to the present infinitary setting thus: a (possibly infinite) term is a normal form when it contains no redexes. The only difference with the finitary case is that here a redex may be itself an infinite term. But note that a redex is still so by virtue of a finite prefix, called as before the redex pattern—this is so because our rewrite rules are orthogonal and hence contain no repeated variables[11]. So, in Figure 40 we have, with as

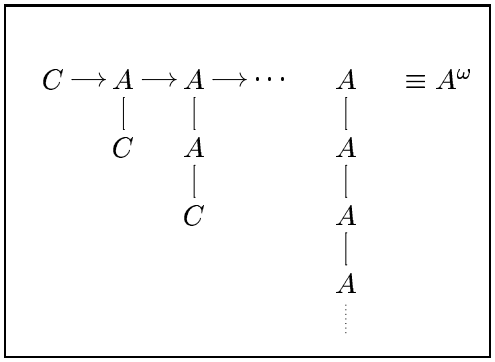


Figure 40: Limit an ω-normal form but not an infinitary normal form

TRS $\{C \to A(C), A(x) \to x\}$, a (Cauchy-) converging reduction sequence with as limit the infinite term $A(A(A(A\cdots$, abbreviated as A^ω; this limit is not a normal form in our sense but it is an ω-normal form, as A^ω only reduces to itself: $A^\omega \to A^\omega$. (Note that this step can be performed in infinitely many different ways, since every A in A^ω is the root of a redex.) Normal forms in our sense are shown in Figures 38, 39 as the rightmost terms (if no other reduction rules are present than the one mentioned above). Henceforth we will often drop the word 'infinite' or 'infinitary'. Thus a term, or a normal form, may be finite or infinite. Note that the concept 'normal form', in contrast to that of 'ω-normal form', only depends on the left-hand sides of the reduction rules in the TRS (Σ, R), which makes the former notion more amenable for analysis.

The notion of Cauchy-converging reduction sequence that was considered so far, is not quite satisfactory. We would like to have the compression property:

$$t_0 \to^c_\alpha t_\alpha \Rightarrow t_0 \to^c_{\leq\omega} t_\alpha.$$

That is, given a reduction $t_0 \to^c_\alpha t_a$, of length α, the result t_α can already be found in at most ω many steps. ('At most', since it may happen that a transfinite reduction sequence can be compressed to finite length, but not to length ω.) Unfortunately, $\to^c_\alpha$ lacks this property:

[11] This choice of 'normal form' deviates from that in Dershowitz e.a. [DKP89]: there a (possibly infinite) term t is said to be an ω-normal form if either t contains no redexes, or the only possible reduction of t is to itself: $t \to t$, in one step.

Counterexample 9.4 Consider the orthogonal TRS with rules

$$\{A(x) \to A(B(x)), B(x) \to E(x)\}.$$

Then $A(x) \to_\omega A(B^\omega) \to A(E(B^\omega))$, so $A(x) \to_{\omega+1} A(E(B^\omega))$. However, we do not have $A(x) \to_{\leq\omega} A(E(B^\omega))$, as can easily be verified.

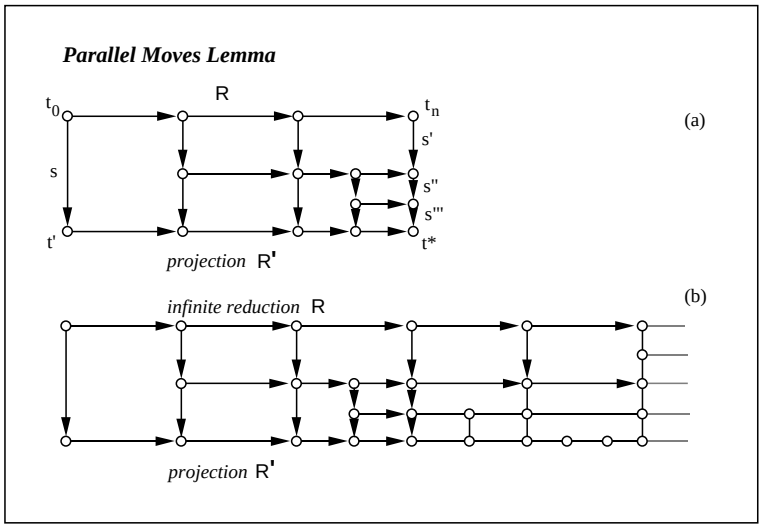


Figure 41: Projecting an infinite reduction

Another obstacle to a satisfactory theory development for $\to^c_\alpha$ is that the Parallel Moves Lemma resists a generalization to the present transfinite case. We recall the Parallel Moves Lemma in Figure 41(a): setting out a finite reduction $\mathcal{R} : t_0 \twoheadrightarrow t_n$ against a one step reduction $t_0 \to_s t'$ (where s is the contracted redex), one can complete the reduction diagram in a canonical way, thereby obtaining as the right-hand side of the diagram a reduction $t_n \twoheadrightarrow t^*$ which consists entirely of contractions of all the descendants of s along $\mathcal{R}$. Furthermore, the reduction $\mathcal{R}' : t' \twoheadrightarrow t^*$ arising as the lower side of this reduction diagram, is called the *projection* of $\mathcal{R}$ over the reduction step $t_0 \to_s t'$. Notation: $\mathcal{R}' = \mathcal{R}/(t_0 \to_s t')$.

We would like to have a generalization of the Parallel Moves Lemma where $\mathcal{R}$ is allowed to be infinite, and converging to a limit. In this way we would have a good stepping stone towards establishing infinitary confluence properties. However, it is not clear at all how such a generalization can be established. The problem is shown in Figure 42. First note that we can without problem generalize the notion of 'projection' to infinite reductions, as in Figure 41(b): there $\mathcal{R}'$ is the projection of the infinite $\mathcal{R}$ over the displayed reduction step. This merely requires an iteration of the finitary Parallel Moves Lemma, no infinitary version is needed. Now consider the two rule TRS $\{A(x,y) \to A(y,x), C \to D\}$. Let $\mathcal{R}$ be the infinite reduction $A(C,C) \to A(C,C) \to A(C,C) \to \cdots$, in fact a reduction cycle of length 1. Note that R is Cauchy converging, with limit $A(C,C)$. The projection $\mathcal{R}'$ of R over the step $A(C,C) \to A(D,C)$, however, is no longer

Cauchy-converging. For, this is $A(D,C) \to A(C,D) \to A(D,C) \to \cdots$, a 'two cycle'. So, the class of infinite converging reduction sequences is not closed under projection. This means that in order to get some decent properties of infinitary reduction in this sense, one has to impose further restrictions; Dershowitz e.a. [DKP89] chooses to impose these restrictions on the terms, thus ruling out e.g. terms as $A(C,C)$ because they are not 'top-terminating'. Another road, the one taken here, is to strengthen the concept of converging reduction sequence. This option is also chosen in Farmer & Watro [FW89].

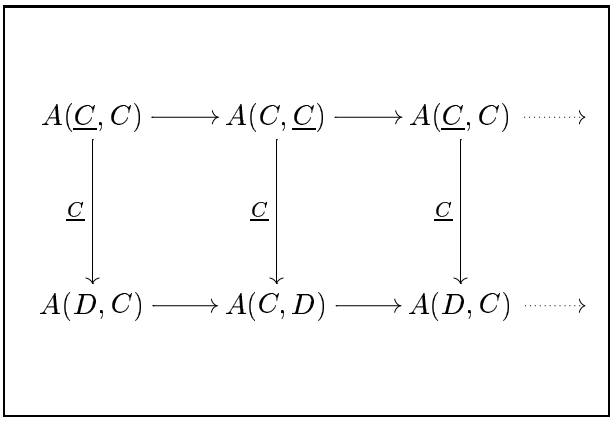


Figure 42: Cauchy converging reduction with divergent projection

As the last example shows, there is a difficulty in that we lose the notion of descendants which is so clear and helpful in finite reductions. Indeed, after the infinite reduction $A(C,C) \to A(C,C) \to A(C,C) \to \cdots$, with Cauchy limit $A(C,C)$, what is the descendant of the original underlined redex C in the limit $A(C,C)$? There is no likely candidate.

We will now describe the stronger notion of converging reduction sequence that does preserve the notion of descendants in limits. If we have a converging reduction sequence $t_0 \to_{s_0} t_1 \to_{s_1} \cdots t$, where s_i is the redex contracted in the step $t_i \to t_{i+1}$ and t is the limit, we now moreover require that

$$\lim_{i \to \infty} \mathit{depth}(s_i) = \infty. \qquad (*)$$

Here $\mathit{depth}(s_i)$, the depth of redex s_i, is the distance of the root of t_i to the root of the subterm s_i. If the converging reduction sequence satisfies this additional requirement $(*)$, it is called *strongly convergent* (see also Figure 43). The difference between the previous notion of (Cauchy-) converging reduction sequence and the present one, is suggested by Figure 44. The circles in that figure indicate the root nodes of the contracted redexes; the shaded part is that prefix part of the term that does not change anymore in the sequel of the reduction. The point of the additional requirement $(*)$ is that this growing non-changing prefix is required really to be non-changing, in the sense that no activity (redex contractions) in it may occur at all, even when this activity would by accident yield the same prefix.

Note that there is now an obvious definition of descendants in the limit terms; the precise formulation is left to the reader.

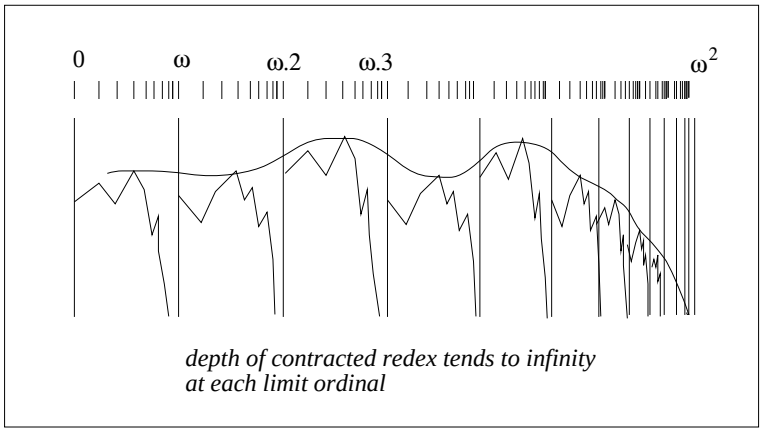


Figure 43: Depth of redex contractions in strongly convergent reduction sequence

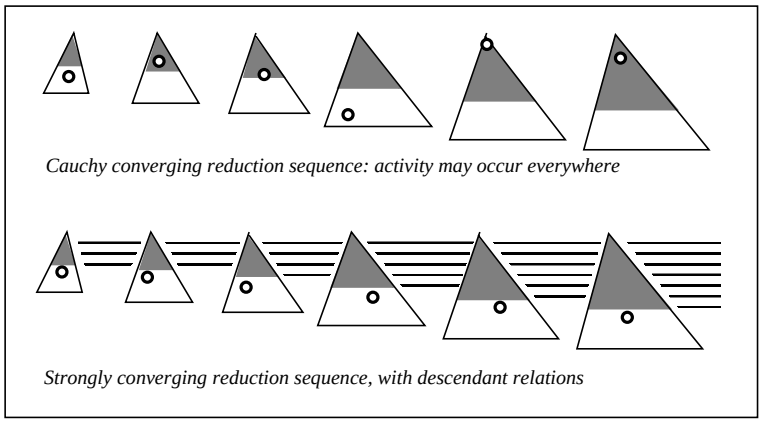


Figure 44: Cauchy convergence and strong convergence

In fact, we define strongly converging reductions of length α for every ordinal α, by imposing the additional condition (*) whenever a limit ordinal $\lambda \leq \alpha$ is encountered. See Figure 43. (It will turn out however that only countable ordinals will occur.) More formally:

Definition 9.5 Let (Σ, R) be a TRS. A *strongly convergent R-reduction sequence of length* α is a sequence $\langle t_\beta \mid \beta \leq \alpha \rangle$ of terms in $\mathrm{Ter}^\infty(\Sigma)$, together with a sequence $\langle s_\beta \mid \beta \leq \alpha \rangle$ of redex occurrences s_β in t_β, such that

1. $t_\beta \to_{s_\beta} t_{\beta+1}$ for all $\beta \leq \alpha$,
2. for every limit ordinal $\lambda \leq \alpha$:

$$\forall n \exists \mu < \lambda \forall \nu (\ \mu \leq \nu \leq \lambda \Rightarrow d(t_\nu, t_\lambda) \leq 2^{-n} \ \&\ \mathit{depth}(s_\nu) \geq n\ ).$$

Often we will suppress explicit mention of the contracted redexes s_β. If $\langle t_\beta \mid \beta \leq \alpha \rangle$ is a strongly convergent reduction sequence we write $t_0 \to_\alpha t_\alpha$.

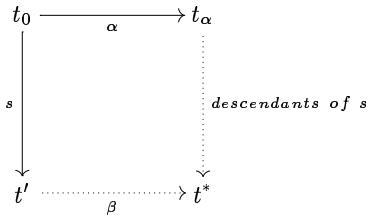
Henceforth all our infinitary reductions will be strongly convergent. Now we can state the benefits of this notion; for the full proofs we refer to Kennaway e.a. [KKSV95a, KKSV97].

Compression Lemma 9.6 *In every orthogonal TRS:*

$$t \to_\alpha t' \Rightarrow t \to_{\leq\omega} t'.$$

(Note that Counterexample 9.4 to compression for Cauchy converging reductions was not strongly converging.)

Infinitary Parallel Moves Lemma 9.7. *In every orthogonal TRS:*



That is, whenever $t_0 \to_\alpha t_\alpha$ and $t_0 \to_s t'$, where s is the contracted redex (occurrence), there are infinitary reductions $t' \to_\beta t^*$ and $t_\alpha \to_\gamma t^*$. The latter reduction consists of contractions of all descendants of s along the reduction $t_0 \to_\alpha t_\alpha$.

Actually, by the Compression Lemma we can find $\beta, \gamma \leq \omega$.

Remark 9.8

1. In every TRS (even with uncountably many symbols and rules), all transfinite reductions have countable length.
2. All countable ordinals can indeed occur as length of a strongly convergent reduction.
3. For ordinary Cauchy convergent reductions this is not so: the rewrite rule $C \to C$ yields arbitrarily long convergent reductions $C \to^c_\alpha C$. However, these are not strongly convergent.

The infinitary Parallel Moves Lemma is "half of the infinitary confluence property". The question arises whether full infinitary confluence holds. That is, given $t_0 \to_\alpha t_1$, $t_0 \to_\beta t_2$, is there a t_3 such that $t_1 \to_\gamma t_3$, $t_2 \to_\delta t_3$ for some γ, δ? Using the Compression Lemma and the Parallel Moves Lemma all that remains to prove is: given $t_0 \to_\omega t_1$, $t_0 \to t_2$, is there a t_3 such that $t_1 \to_{\leq\omega} t_3$, $t_2 \to_{\leq\omega} t_3$? Surprisingly, the answer is negative: full infinitary confluence for orthogonal rewriting does not hold. The counterexample is in Figure 45, consisting of an orthogonal TRS with three rules, two of which are collapsing rules.

Indeed, in Figure 45(a) we have $C \to_\omega A^\omega$, $C \to_\omega B^\omega$ but A^ω, B^ω have no common reduct as they only reduce to themselves. Note that these reductions are indeed strongly convergent. (Figure 45(b) contains a rearrangement of these reductions.)

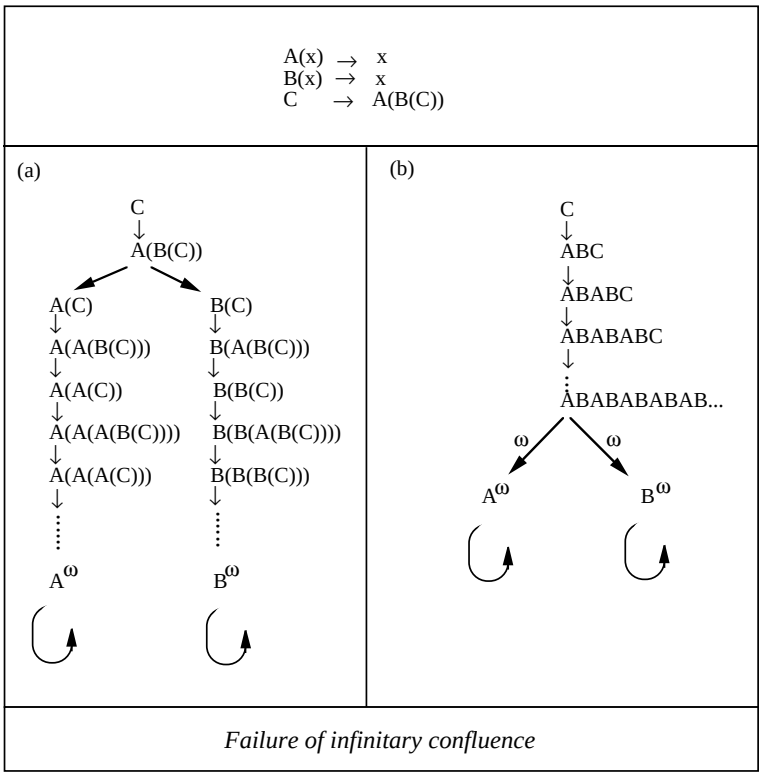


Figure 45: Counterexamples to infinitary confluence

However, we do have unicity of (possibly infinite) normal forms.

Theorem 9.9 *For all orthogonal TRSs: Let $t \to_\alpha t'$, $t \to_\beta t''$ where t', t'' are (possibly infinite) normal forms. Then $t' \equiv t''$.*

Here $\equiv$ denotes syntactical equality. Note that in the ABC counterexample in Figure 45 the terms A^ω and B^ω are not normal forms.

We will now investigate the extent to which infinitary orthogonal rewriting lacks full confluence. It will turn out that non-confluence is only marginal, and that terms which display the bad behaviour are included in a very restricted class. The following definition is inspired by the corresponding notion in λ-calculus; see Section 3.3 or, for more details, Barendregt [Bar84].

Definition 9.10

1. The term t is in *head normal form* if $t \equiv C[t_1, \ldots, t_n]$ where $C[\ , \ldots, \ ]$ is a non-empty context (prefix) such that no reduction of t can affect the prefix $C[\ , \ldots, \ ]$. More precisely, if $t \twoheadrightarrow s$ then $s \equiv C[s_1, \ldots, s_n]$ for

some s_i $(i = 1, \ldots, n)$, and every redex of s is included in one of the s_i $(i = 1, \ldots, n)$.

2. t has a head normal form if $t \twoheadrightarrow s$ and s is in head normal form.[12]

Definition 9.11 If t is a term of the TRS R, then the family of t is the set of subterms of reducts of t, i.e. $\{s \mid t \twoheadrightarrow C[s]$ for some context $C[\,]\}$.

Theorem 9.12 *For all orthogonal TRSs: Let t have no term without head normal form in its family. Then t is infinitary confluent.*

Actually, this theorem can be much improved. Consider again the ABC example in Figure 45. Rearranging the reductions $C \to_\omega A^\omega$, $C \to_\omega B^\omega$ as in Figure 45(b) into reductions $C \to_\omega (AB)^\omega \to_\omega A^\omega$ and $C \to_\omega (AB)^\omega \to_\omega B^\omega$ makes it more perspicuous what is going on: $(AB)^\omega$ is an infinite 'tower' built from two different collapsing contexts $A(\,)$, $B(\,)$, and this infinite tower can be collapsed in different ways.

Remark 9.13

1. The ABC example (Figure 45) is not merely a pathological example; the same phenomenon (and therefore failure of infinitary confluence) occurs in Combinatory Logic, as in Figure 46, where an infinite tower built from the two different collapsing contexts $K\Box K$ and $K\Box S$ is able to collapse in two different ways. (Note that analogous to the situation in Figure 45, the middle term, built alternatingly from $K\Box K$ and $K\Box S$, can be obtained after ω steps from a finite term which can easily be found by a fixed point construction.)
2. Also for λ-calculus one can now easily construct a counterexample to infinitary confluence.

Remarkably, it turns out that the collapsing phenomenon is the only cause of failure of infinitary confluence. (The full proof is in Kennaway e.a. [KKSV95a].) Thus we have:

Theorem 9.14

1. *Let the orthogonal TRS R have no collapsing rewrite rules $t(x_1, \ldots, x_n) \to x_i$. Then R is infinitarily confluent.*
2. *If R is an orthogonal TRS with as only collapsing rule: $I(x) \to x$, then R is infinitary confluent.*

Call an infinite term $C_1[C_2[\cdots C_n[\cdots]\cdots]]$, built from infinitely many non-empty collapsing contexts $C_i[\,]$, a *hereditarily collapsing (hc) term.* (A context $C[\,]$ is collapsing if $C[\,]$ contains one hole $\Box$ and $C[\,] \twoheadrightarrow \Box$.) Also a term reducing to a hc term is called a hc term. E.g. C from the ABC example in Figure 45 is a hc term. Clearly, hc terms do not have a head normal form.

[12] Actually, this definition is equivalent to one of Dershowitz e.a. [DKP89]; there a term t is called 'top-terminating' if there is no infinite reduction $t \to t' \to t'' \to \cdots$ in which infinitely many times a redex contraction at the root takes place. So: t is top-terminating iff t has a head normal form.

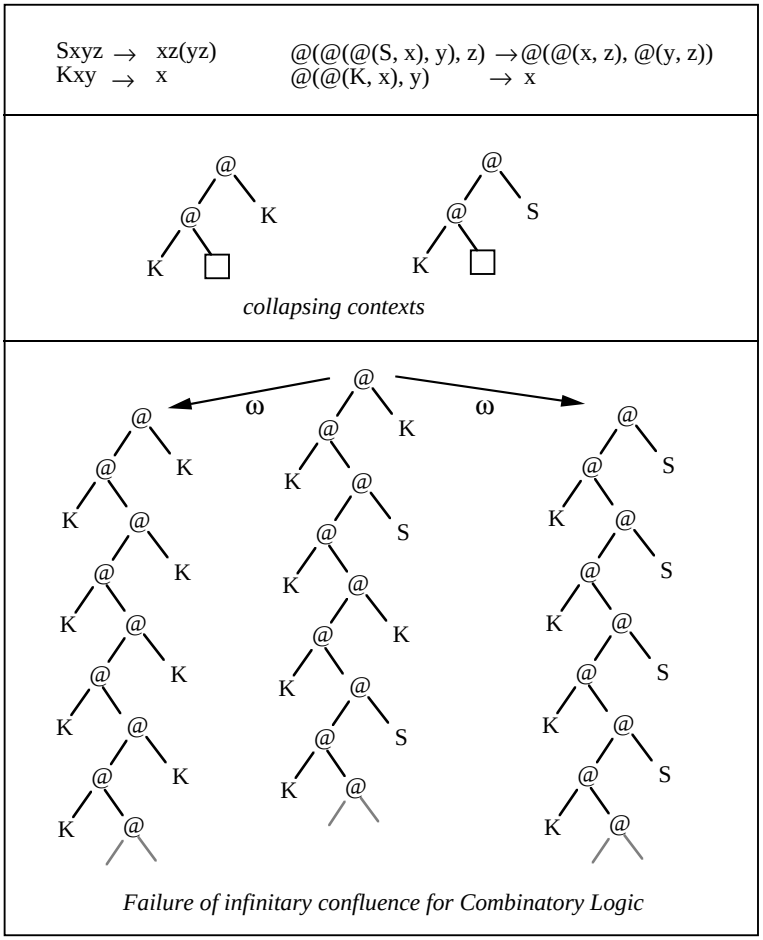


Figure 46: Failure of infinitary confluence for CL

Theorem 9.15 *Let t be a term in an orthogonal TRS, which has not a hc term in its family. Then t is infinitary confluent.*

This theorem can be sharpened somewhat, as follows. Let us introduce a new symbol $\bullet$ to denote hc terms, with the rewrite rule:

$$t \to_\bullet \bullet \quad \text{if } t \text{ is a hc term.}$$

We call $\bullet$ the 'black hole', because of its infinite collapsing behaviour. Of course this rule is not 'constructive', i.e. the reduction relation $\to_\bullet$ may be undecidable (as it is in CL, Combinatory Logic). However, we now have that orthogonal reduction extended with $\to_\bullet$ is infinitary confluent.

10 Infinitary λ-calculus

After our exploration of infinitary rewriting for the first-order case we now turn to the same endeavour for λ-calculus. In part, infinitary λ-calculus is already

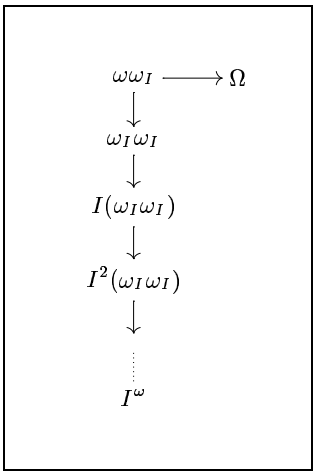


Figure 47: Fixed point of I: failure of PML in infinitary λ-calculus

well-known in so far as Böhm Trees can be perceived as a kind of "infinitary normal forms", but before [KKSV95b] (since then superseded by [KKSV97]) this intuitive idea was not yet made precise. The basic idea to set the scene for infinitary λ-calculus is analogous to the first-order case. In particular, the requirement of strong convergence is essential here again.

But there are some striking differences too. One of these is that PML does not hold anymore, as the simple counterexample in Figure 47 demonstrates.

> We use the abbreveations $I \equiv \lambda x.x$, $\omega \equiv \lambda x.xx$, $\Omega \equiv (\lambda x.xx)(\lambda x.xx)$, $\omega_I \equiv \lambda x.I(xx)$. So $\omega\omega_I =_\beta YI$, where $Y \equiv \lambda f.[\lambda x.f(xx)][\lambda x.f(xx)]$, Curry's fixed point combinator. The limit $I^\omega \equiv I(I(I(\ldots$ is depicted as an infinite term tree in Figure 49.

Both Ω and I^ω only reduce to themselves. Note that the infinite reduction $\omega\omega_I \to \cdots I^\omega$ is strongly convergent indeed, i.e. the contracted redex depth tends to ∞.

A fortiori, CR fails—but this we knew already from the first-order case, as the counterexamples there can be transposed easily to infinitary λ-calculus (e.g. the example in Figure 46).

Many basic concepts easily generalize from finitary λ-calculus to the infinitary case: normal form, β-reduction, substitution, α-conversion, etc. The Finite Developments Theorem of course does not generalize, since an infinite λ-term may possess infinitely many redexes. But a satisfactory analogous fact does hold: the end result of all strongly convergent complete developments of some possibly infinite set of redexes in an infinitary λ-term M is unique. The complication here is that a set of redexes in M cannot always be completely developed in a strongly convergent way. (E.g. take the redexes in I^ω.)

We did not yet stipulate what an infinite λ-term actually is. The first thought is that it is a possibly infinite unary-binary tree built from the binary @ (application), the unary λx (abstraction), and variables $x, y, z, \ldots$ and possibly constants, notably Ω (to denote 'undefined'), together with the 'usual' metric.

Remark 10.1 A different notation for Böhm Trees is employed in Barendregt [Bare84],

where the nodes of such a tree are of the form either x, or Ω, or:

$$\lambda x_1 \cdots x_n.y \quad \begin{array}{ccc} \diagup & & \diagdown \\ \Box & \cdots & \Box \end{array}$$

For obvious reasons, we call this the 'head normal form notation', or briefly *hnf notation* (refer to Section 3.3). This notation is suitable for terms in normal form, among which Böhm Trees; but the notation does not lend itself for representing terms containing β-redexes.

Figure 48 gives as examples the finite term $\lambda x.y(xx)$ and the infinite Böhm tree of the Y-combinator of Curry, $BT(Y)$, written in both notations. We will henceforth employ only the 'applicative' notation.

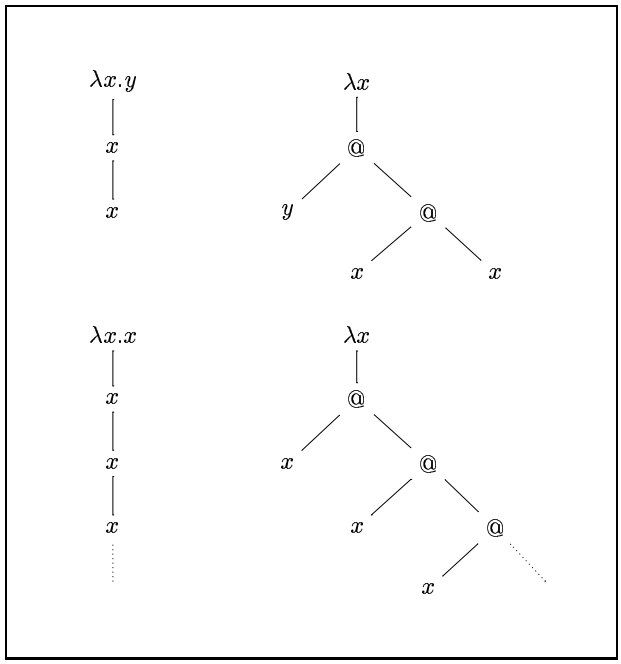


Figure 48: $\lambda x.y(xx)$ and $BT(Y)$ in hnf and applicative notation

Now, continuing with the definition of infinite λ-terms, there is an interesting ramification presenting itself. In Figure 49 we have displayed the term I^ω, encountered before, and its 'mirror image' ${}^\omega I$, possessing an infinite left branch of @-nodes.

This ${}^\omega I$ is an anomalous object; e.g. it is a normal form, but it is also unsolvable (in the obvious generalization of that concept to the infinitary case). We can exclude such unwanted terms, in a way that has some unexpected extra benefits.

Trees composed of @- and λx-nodes have 3 dimensions in which they can grow, depicted in Figure 50: down, left, right (dlr).

We now define 8 notions of depth of an occurrence in a λ-term, indexed by the tuples listed in Table 6.

E.g. the 110-depth counts only d- and l-steps, disregarding the r-steps. So the displayed occurrence of x in the term $\lambda xy.((xy)(\lambda x.x))$ (see Figure 51)

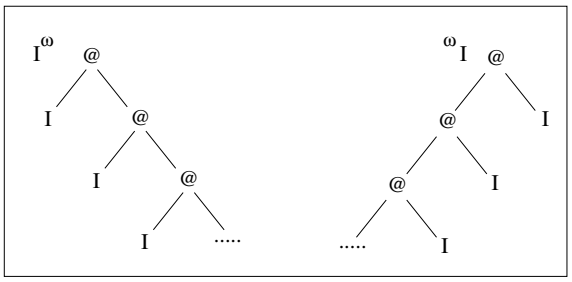


Figure 49: Trees of the terms I^ω and ${}^\omega I$

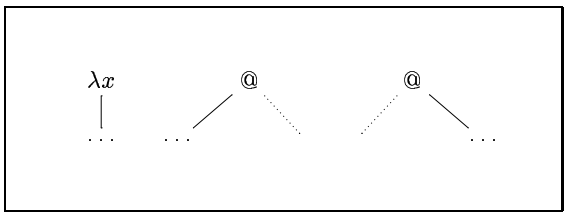


Figure 50: Directions down, left, right

d	l	r
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Table 6: The 8 possible dlr-tuples

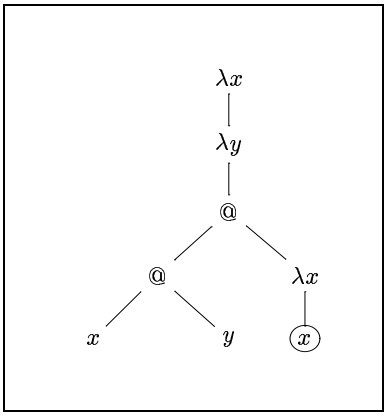


Figure 51: Depth in a λ-term

has 110-depth 3. Accordingly, we define the usual notion of distance between $M, N \in \mathrm{Ter}(\lambda)$ ($\mathrm{Ter}(\lambda)$ is the set of λ-terms): it is 2^{-n} where n is the minimal depth such that M, N differ at an occurrence of depth n. If $M \equiv N$, their distance is 0. Now by parametrizing the depth d as before, e.g. 110-depth etc., we have 8 metric spaces $(\mathrm{Ter}(\lambda), d_{abc})$ leading after completion to 8 complete metric spaces $(\mathrm{Ter}^{\infty}(\lambda), d_{abc})$ of finite and infinite λ-trees. They can be equipped with generalizations of the finitary notions of substitution, α-conversion, β-reduction (now infinitary!) etc. Let us call these λ-calculi λ_{abc}.

One of the calculi, λ_{000}, is trivial as an infinitary calculus: it is the finite λ-calculus. Four others, $\lambda_{010}, \lambda_{011}, \lambda_{100}$ and λ_{110}, turn out to be uninteresting (they lack some basic properties, such as substitutivity of the reduction relation). Three remain: $\lambda_{001}, \lambda_{101}$ and λ_{111}.

Note that I^{ω} is an object (a term) in all three of $\lambda_{001}, \lambda_{101}, \lambda_{111}$; but ${}^{\omega}I \notin \mathrm{Ter}(\lambda_{001}), \mathrm{Ter}(\lambda_{101})$. And the term consisting of an infinite string of abstractions $\lambda x_0.(\lambda x_1.(\lambda x_2.\ldots$ [13] is absent in λ_{001}, but is present in λ_{101} and λ_{111}. Also any term which "would have" an infinite dl-branch (a 'spine' in the sense of Barendregt et al. [BKKS87]) is absent in λ_{001}. (See the Remark 10.2 below).

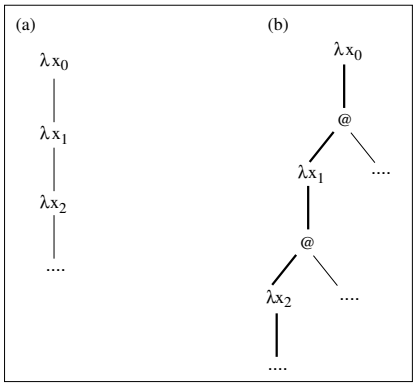


Figure 52: Example of an infinite d-branch and an infinite dl-branch

It turns out that the three infinitary calculi $\lambda_{001}, \lambda_{101}, \lambda_{111}$ are the natural home resorts for two well-known concepts and one recently emerged:

- λ_{001} contains the Böhm trees $BT(M)$,
- λ_{101} contains the Lévy-Longo (or lazy) trees $LLT(M)$,
- λ_{111} contains the Berarducci trees $BeT(M)$.

Böhm trees are well-known (see Barendregt [Bar84]). For LL-Trees, in lazy λ-calculus, see Abramsky & Ong [AO93]. For Berarducci trees, see Berarducci [BI96]. The latter arose in studies of consistent extensions of λ-calculus. For a general introduction to these three models we refer to Kennaway et al. [KKSV93] and Kennaway et al. [KKSV95a].

[13] Note that due to α-conversion there is only one such term, not continuum many as otherwise would be the case.

	BT	LLT	BeT
β-reduction	•	•	•
$M \to_{\text{uns}} \Omega$ if M has no hnf/has no whnf/is mute	•	•	•
Ω_l-rule $\Omega M \to \Omega$	•	•	
Ω_d-rule $\lambda x.\Omega \to \Omega$	•		

Table 7: Infinitary λ-calculi compared

The three models (BT, LLT, BeT) employ different notions of undefined. In the BT-model λ_{001}, terms without head normal form (i.e. the unsolvable terms) are equated to Ω. In the LLT-model λ_{101}, terms without weak head normal form are equated to Ω. In the BeT-model λ_{111} mute terms are equated to Ω. In all three calculi we obtain the BT's, LLT's, BeT's in a uniform way as infinitary normal forms with respect to the notions of reduction as in Table 7; each consists of β-reduction, the 'unsolvable rule', and 0,1 or 2 Ω-simplification rules according to the dlr-parametrization discussed so far.

All three of these notions of reduction are infinitarily confluent, so the corresponding trees (BT's etc.) are unique.

Viewing Böhm trees as normal forms obtained by a possibly infinite reduction is obviously a view that is totally different from the more usual alternative definitions using coinduction or direct approximations and ideal completion. The main difference is that now we can obtain information by inspection of the reduction sequence yielding the Böhm tree, as indeed we will do now.

Remark 10.2

1. The term consisting of an infinite list of abstractions $Z \equiv \lambda x_0.(\lambda x_1.(\lambda x_2.\ldots$ can be obtained as the fixed point YK where $K \equiv \lambda xy.x$. In λ_{001} we have $Z = \Omega$, but not so in λ_{101}. Indeed, every term Z such that $Z \equiv Z_0 \twoheadrightarrow \lambda x_0.Z_1, Z_1 \twoheadrightarrow \lambda x_1.Z_2, Z_2 \twoheadrightarrow \lambda x_2.Z_3, \ldots$, is unsolvable. Likewise, every term that would 'generate' in a similar manner an infinite dl-path (from the root) is unsolvable.

2. With respect to the partial order $\geq_\Omega$ (called Ω-refinement) we have for all M:

$$BT(M) \leq_\Omega LLT(M) \leq_\Omega BeT(M).$$

   So Berarducci Trees contain most information, Böhm Trees the least.

3. As to the domains of the three models:

$$\text{Ter}(\lambda_{001}) \subset \text{Ter}(\lambda_{101}) \subset \text{Ter}(\lambda_{111}).$$

4. An example of a nontrivial Berarducci Tree that trivializes (i.e. $= \Omega$) in both λ_{001} and λ_{101}:

$$\Omega_3 I, \text{ where } \Omega_3 \equiv \omega_3\omega_3 \equiv (\lambda x.xxx)(\lambda x.xxx).$$

   The Berarducci Tree of this term is depicted in Figure 53. It is a so-called 'easy' term, i.e. one that can consistently be equated to any desired λ-term.

5. For the terms in Figure 49 we have $BT(I^\omega) = LLT(I^\omega) = BeT(I^\omega) = \Omega$; $BT({}^\omega I) = LLT({}^\omega I) = \Omega$. $BeT({}^\omega I)$ is the nontrivial infinite tree displayed there.

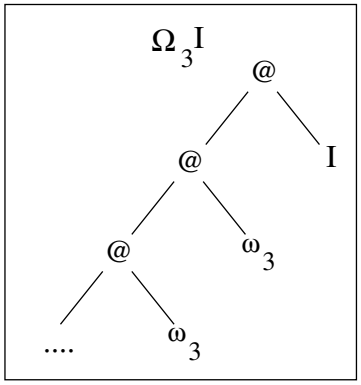


Figure 53: Example of a Berarducci tree

11 Origin tracking in infinitary λ-calculus

In the previous section we have set up a framework for infinitary λ-calculus that enables us to compute Böhm Trees (and Lévy-Longo Trees, and Berarducci Trees) by infinitary rewriting. In this section we will apply this rewrite system to obtain a perspicuous proof of an important theorem due to G. Berry that establishes the inherently sequential nature of evaluation in λ-calculus. (Other proofs can be found in Berry [Ber78],[Ber79], Barendregt [Bar84], Curien [Cur93].) We will restrict ourselves to the case of Böhm trees, but we expect that the same analysis can also be applied to the other two kinds of trees (LLT and BeT).

Analogous to the sections 7 and 8 we will start from the normal form $BT(M)$, and then trace back the origin of an Ω in $BT(M)$ all the way to M. The difference is that now we are dealing with infinite terms (trees) and infinite reductions. Let us first consider Berry's Sequentiality Theorem (BST). It states that, given a $\lambda\Omega$-term M as 'input', the Ω's in the Böhm tree of M, $BT(M)$, the 'output', are causally related in a very specific way to the Ω's in the input. Namely either

1. an output Ω is not causally related to any of the input Ω's, or
2. an output Ω is caused by precisely one of the input Ω's.

In Figure 54, 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. Case 1 means that no refinement of the input Ω's will cause a proper refinement of the considered output Ω; case 2 means that a proper refinement of the considered output Ω can only be realized by a proper refinement of the one input Ω that is the cause, the origin, of the considered output Ω. More precisely stated:

Theorem 11.1 *[Berry [Ber78]] Let $M \in \mathrm{Ter}^\infty(\lambda\Omega)$ and let Ω occur in $BT(M)$ at position p (notation $BT(M) \mid_p = \Omega$). Then one of the following two cases holds:*

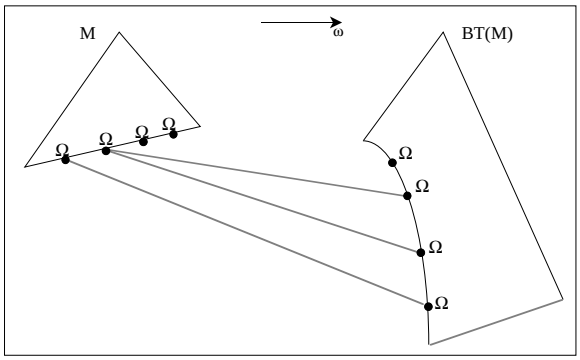


Figure 54: Causal dependence of Ω's in Böhm tree from those in original term

1. *The Ω is independent of the Ω's in M. This means that no refinement of M is able to give more information at position p, i.e.*

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

2. *There is an Ω in M which causes the Ω at position p in $BT(M)$. This means that Ω at p is insensitive for increases at any of the other Ω's in M and, moreover, that Ω at p will be properly increased when the Ω in M is refined to a fresh variable z, i.e. there exists some context C such that $M = C[\Omega]$ and for all one-hole contexts $C' \geq_\Omega C$ and every fresh variable z,*

$$BT(C'[\Omega]) \mid_p = \Omega \text{ and } BT(C'[z]) \mid_p \neq \Omega.$$

Berry's Sequentiality Theorem can be used to prove the non-definability of 'parallel-or' and other parallel functions. See Appendix E, where the non-definability (in λ-calculus) of 'parallel-or' is proved.

As said, the idea is to trace back a given Ω in $BT(M)$ for $M \in \mathrm{Ter}(\lambda\Omega)$ to its origin in M. Now there are two cases. Either the Ω under consideration traces back to a unique Ω-occurrence in M, or Ω has no origin at all. This will happen if the Ω, or rather the unsolvable $\lambda\Omega$-term that gave rise to the Ω, is *created* along the way (see Section 4.6).

Let us consider the rewrite system that is used to trace back the $\Omega \in BT(M)$. In first approximation this is the system in Table 8. Note that we do not have $\Omega \to_{uns} \Omega$, since Ω is not an unsolvable $\lambda\Omega$-term; see the definitions in Section 3.3 and 3.5.

In order to define the tracing relation that we need, we now lift this rewrite system to a labeled version, as in Table 9.

So we have in fact partially labeled $\lambda\Omega$-terms (or in other words, one of the labels is the empty label). As in the simply labeled λ-calculus (Definition 4.1, Table 1) the labels are simple letters $a, b, c, \ldots$. Now the tracing relation is given as before in the simply labeled λ-calculus, by identity of labels. That is, in the rule Ω_l the Ω in the righthand side traces back to the displayed Ω_l the Ω in the lefthand side—and not to the whole term. Likewise for the Ω_d-rule.

$(\lambda x.M)N \to_\beta M[x := N]$

$M \to_{uns} \Omega$, if M is unsolvable

$\Omega M \to_l \Omega$

$\lambda x.\Omega \to_d \Omega$

Table 8: Böhm reduction

β	$((\lambda x.Z)^a Z')^b \to_\beta Z[x := Z']$
uns	$M \to_{uns} \Omega$, if M is an unsolvable $\lambda\Omega$-term
Ω_l	$(\Omega^a M)^b \to \Omega^a$
Ω_d	$(\lambda x.\Omega^a)^b \to_d \Omega^a$

Table 9: Labeled Böhm reduction

Example 11.2 Figure 55 gives an example of the tracing relation that results from the following labeled Böhm reduction.

$$\begin{array}{lll} ((\lambda^a x.\lambda^b y.(x^c \Omega^d)^e)\Omega^g)^h & \to_\beta & \lambda^b y.(\Omega^g \Omega^d)^e \\ & \to_l & \lambda^b y.\Omega^g \\ & \to_d & \Omega^g \end{array}$$

Note that origins, if they exist, are unique. (But in contrast to the situation in Section 7, not everything has an origin.) Moreover, they are independent of the actual reduction sequence. For finite reductions this follows from the fact that labeled Böhm reduction is confluent; the proof is not much harder than the confluence proof in Barendregt [Bar84] for the (unlabeled) $\lambda\Omega$-calculus.

It still may not be clear how we can trace back an Ω in $BT(M)$ to M, since an infinite reduction $M \to_\omega BT(M)$ is involved here. Figure 56 clarifies the working of this procedure. We have the infinite reduction $M \equiv M_0 \to M_1 \to \cdots \to M_n \to \cdots M_\omega \equiv BT(M)$. Let Ω occur in M_ω at position p. We wish to trace back Ω to the original term M_0. Let n be the depth of Ω in M_ω. Then from some M_k onwards, the redexes contracted are deeper than n. So in the tail of the reduction sequence from M_k to M_ω, the prefix up to n of M_k, $M_{k+1}, \ldots, M_\omega$ is 'at rest'. So we can take the ancestor of Ω in M_k via the trivial descendant relation.

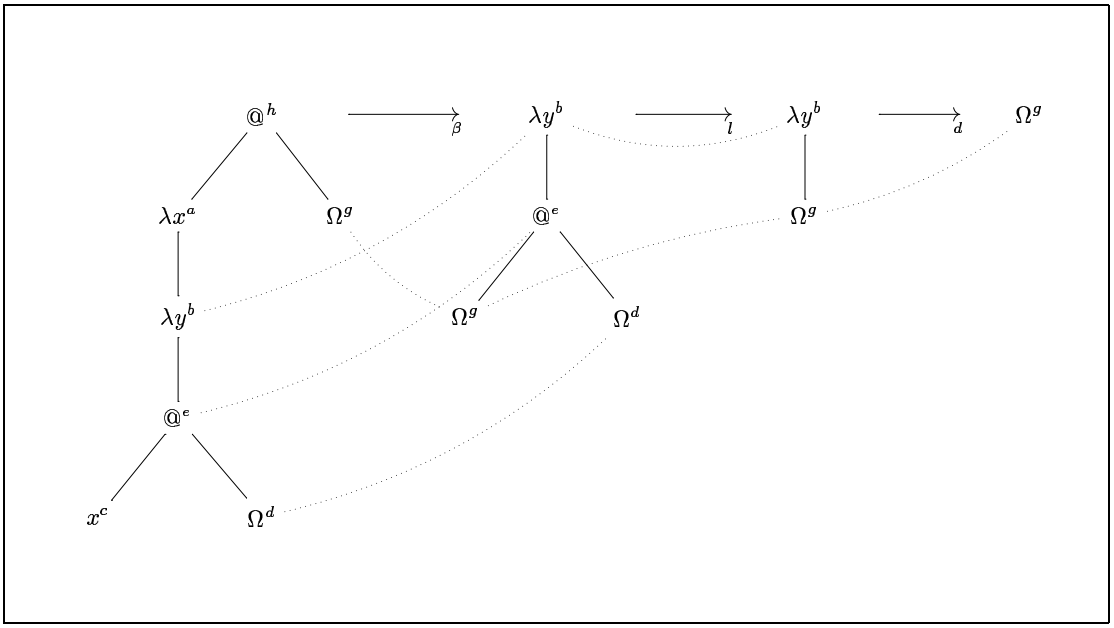


Figure 55: Origin tracking

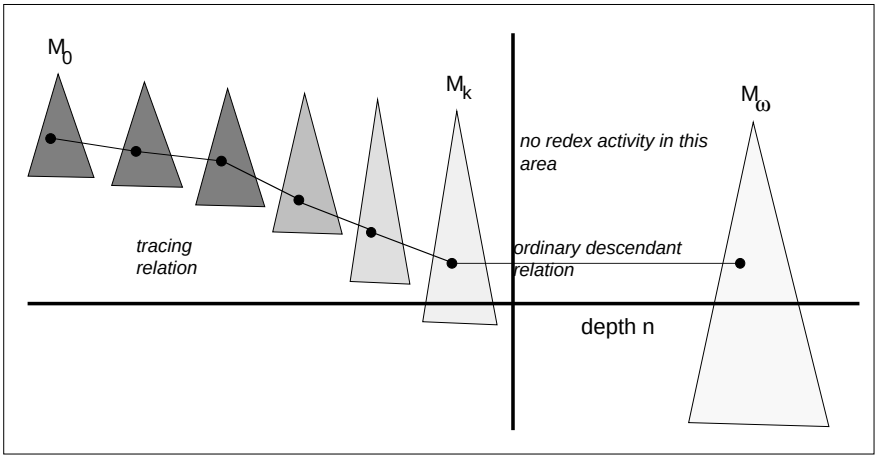


Figure 56: Tracing back along infinite reduction sequence

From M_k to M_0 the symbol Ω can now be traced back by the definition of tracing given above. In this way we see that an Ω in $BT(M)$ traces back to a unique symbol Ω in M—or it has no origin at all.

Finally, we can with little effort establish the properties concerning refining of Ω's that BST asserts, by following the infinite reduction from M to $BT(M)$ step by step, and apply the fact that β and $\geq_\Omega$ commute. (See Proposition 7.9.)

Remark 11.3 The above proof sketch applies, mutatis mutandis, just as well to LLT's and BeT's. In that case we need not both rules Ω_l and Ω_d, but as already displayed in Table 7, only Ω_l for LLT's and none of the rules Ω_l, Ω_d for BeT's.

Acknowledgements. We thank Henk Barendregt, Stefan Blom, Gérard Boudol, Vincent van Oostrom, Gordon Plotkin and Femke van Raamsdonk for help and useful discussions. Vincent van Oostrom was particularly helpful in pointing out the norm used in the proof that needed reduction is hypernormalizing and in pointing out some errors.

12 Appendices

Appendix A: Parallel reduction à la Aczel

We compare the notions of parallel reduction as it usually employed in proofs of CR due to Tait and Martin-Löf, and the amended notion that was proposed by Aczel [Acz78]. For an extensive discussion see van Raamsdonk [Raa96].

We use the notation $\multimap$ for parallel reduction. In the style of Tait and Martin-Löf, it is defined by the inductive clauses in Table 10. It characterizes complete developments, in the sense that $M \multimap N$ if and only if there is a complete development from M to N.

$$M \multimap M$$

$$\frac{M \multimap M'}{\lambda x.M \multimap \lambda x.M'}$$

$$\frac{M \multimap M' \qquad N \multimap N'}{MN \multimap M'N'}$$

$$\frac{M \multimap M' \qquad N \multimap N'}{(\lambda x.M)N \multimap M'[x := N']}$$

Table 10: Parallel reduction à la Tait & Martin-Löf

In Aczel [Acz78] the last clause is replaced by:

$$\frac{M \multimap \lambda x.M' \qquad N \multimap N'}{MN \multimap M'[x := N']}$$

Now there is a complete β-*super*development form M to N if and only if $M \multimap N$ according to Aczel's definition.

Example 12.1 In the first definition, due to Tait and Martin-Löf, we do not have $IIII \multimap I$ (with $I \equiv \lambda x.x$); in Aczel's definition we do.

Likewise $(\lambda xyz.xyz)abc \multimap abc$ and even $II(\lambda xyz.xyz)abc \multimap abc$.

Appendix B: Failure of FD for λ-residuals

We give the counterexample to Finite Developments for the notion of λ-residual in $\lambda\beta\eta$ from Klop [Klo80]. See Definition 5.3.

The following is an infinite reduction in which all the contracted redexes are λ-residuals of redexes in M_0.

$$\begin{array}{rcl} M_0 & \equiv & (\lambda_0 x.xx)(\lambda_1 z.(\lambda_2 y.yy)z) \\ & \to_{\lambda_0} & (\lambda_1 z.(\lambda_2 y.yy)z)(\lambda_1 z.(\lambda_2 y.yy)z) \\ & \to_{\lambda_1} & (\lambda_2 y.yy)(\lambda_1 z.(\lambda_2 y.yy)z) \\ & \to \to & (\lambda_2 y.yy)(\lambda_1 z.(\lambda_2 y.yy)z) \\ & \to & \cdots \end{array}$$

Note that FD does hold for (ordinary) CF-residuals in $\lambda\beta\eta$. See e.g. Barendregt, Bergstra, Klop and Volken [BBKV76], Chapter II, using the method of decreasing weights (also used for FD in $\lambda\beta$ in Barendregt [Bar84]). FD also holds for cluster residuals (de Vrijer [Vri89]).

Appendix C: Collapsing reductions

In this section we will treat the case of collapsing reductions and verify in detail that the main properties of the needed prefix, Corollaries 8.27-8.31, carry over to the presence of collapsing reductions.

Definition 12.2 Let R be an orthogonal TRS. To R we associate a TRS R_ε as follows. We extend the signature of R with the unary function symbol ε. We will use the collapsing rule $\varepsilon(x) \to x$, and call it the ε-rule. If t reduces to s by applying the ε-rule, we say that t is an ε-expansion of s.

Now let $r : t \to s$ be a rule from R. Then R_ε will contain all rules (in the extended signature) of the form $t^* \to \varepsilon(s)$, where t^* is an ε-expansion of t obtained by ε- expanding some internal subterms q of t to $\varepsilon(q)$. (A subterm (occurrence) q of t is internal if it is not a variable nor the whole t.) Call the collection of all such rules: r^ε.

Now R_ε has as rules the union of r^ε for all rules in R.

Example 12.3 Let r be the collapsing rule $A(x,0) \to x$. Then r^ε consists of the rules $A(x,0) \to \varepsilon(x)$, $A(x,\varepsilon(0)) \to \varepsilon(x)$, $A(x,\varepsilon(\varepsilon(0))) \to \varepsilon(x)$ etc.

Proposition 12.4 *Let R be an orthogonal TRS. Then R_ε is an orthogonal TRS with only non-collapsing rules.*

Proof. That R_ε is non-collapsing is clear. Left-linearity of R_ε is also clear. The rules of R_ε are also non-overlapping. For suppose there is an overlap, then removal of ε's would yield an overlap between rules of R. Making this argument more precise is routine. □

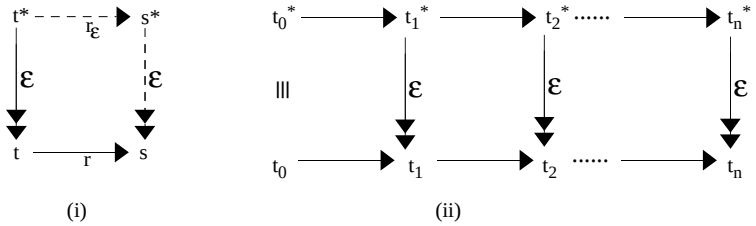


Figure 57: Lifting of reductions

Proposition 12.5 *[Lifting of reductions]*

1. *Let $t \to_r s$ be a reduction step in R according to rule r. Let t^* be an ε-expansion of t. Then for some rule $r_\varepsilon \in r^\varepsilon$ and ε-expansion s^* of s we have $t^* \to_{r_\varepsilon} s^*$. (See Figure 57(i).)*

2. *A reduction $t_0 \to t_1 \to \cdots \to t_n$ can be "lifted" to a reduction $t_0 \equiv t_0^* \to t_1^* \to ... \to t_n^*$ in R_ε, as in Figure 57(ii).*

Proof. Straightforward from the definitions. □

We can now state the definition of $\rhd$ also when collapsing rules are present. Note that this formalizes the verbal description that was given in four clauses in Definitions 8.6 and 8.8.

Definition 12.6 Let R be an orthogonal TRS, possibly with collapsing rules. Let $t_0 \to t_1 \to \cdots \to t_n$ be a reduction in R. We define the relation $\rhd \subseteq Symb(t_0) \times Symb(t_n)$ as follows. (Here $Symb(t)$ denotes the set of symbol occurrences in t.)

1. Lift the reduction to $t_0 \equiv t_0^* \to t_1^* \to ... \to t_n^*$ in R_ε.

2. Consider the relation $\rhd \subseteq Symb(t_0) \times Symb(t_n^*)$ as defined above for the non-collapsing case.

3. "Project back" $\rhd$ to $Symb(t_0) \times Symb(t_n)$, by forgetting ε's. More precisely:

   Let $t \in Ter(R_\varepsilon)$ and let $t_e \in Ter(R)$ be the ε-normal form of t. Then to each occurrence of $p \in t$ there corresponds in the obvious way an occurrence p_e of t_e.

Now let $t, s \in Ter(R_\varepsilon)$, with $\rhd \subseteq Symb(t) \times Symb(s)$, and define $\rhd_e \subseteq Symb(t_e) \times Symb(s_e)$ by:

$$p \rhd q \Rightarrow p_e \rhd_e q_e.$$

Then $\rhd_e$ is the relation that we aimed to define.

It is now crucial that Proposition 8.20 generalizes to the case of collapsing rules.

Proposition 12.7 *All prefixes obtained by tracing back the normal form have the rpc-property.*

Proof. We assume the proposition proved for the non-collapsing case, as in Section 8.7.

Consider a reduction $\mathcal{R} : t_0 \to \cdots \to t_n$, where t_n is in normal form, in R. Lift this reduction to $\mathcal{R}^* : t_0 \to \cdots \to t_n^*$ in R_ε. Let $\pi^*(t_0)$ be the prefix obtained by tracing back the R_ε-normal form t_n^* to t_0 via $\mathcal{R}^*$. As before, $\pi(t_0)$ is the prefix obtained by tracing back t_n to t_0 via $\mathcal{R}$.

Note that $\pi(t_0) \equiv \pi^*(t_0)$, by the trace definition for the collapsing case. Now suppose that $\pi(t_0)$ would not have the rpc property. Then there is an R-redex r whose pattern crosses the border of $\pi(t_0)$. But an R-redex is also an ε-redex; so the rpc-property would fail for R_ε, contrary to our initial assumption. □

This proposition entails that the commutation of Ω- and R-reduction (which rests on the rpc-property) generalizes to the collapsing case. Hence also Proposition 8.24; hence also $\pi(t_0) \twoheadrightarrow t_n$; and hence Corollary 8.27: the prefix $\pi(t_0)$ contains a redex pattern. Also the other three corollaries of Proposition 8.24 go through.

Appendix D: Transitivity of the descendant relation

In this Appendix we elaborate the claim made in Remark 8.13. We start with a simple observation about substitutivity of labeled reduction.

Definition 12.8

1. A *label substitution* is a map from atomic labels to the set of labels, extended to the set of labels in the obvious (homomorphic) way.
2. If t^I is a labeled term and σ a label substitution, $t^{\sigma(I)}$ is the labeled term obtained by substituting for every atomic label a the label $\sigma(a)$.
3. if t^I is a labeled term, p a symbol (occurrence) in t, and α its label, we simply write $p^\alpha \in t^I$.

Proposition 12.9 *[Substitutivity of labels]*

1. *Let $t^I \to s^J$ be a labeled step. Let $\sigma(I)$ and $\sigma(J)$ be obtained from I, J by label substitution σ. Then $t^{\sigma(I)} \to s^{\sigma(J)}$.*

2. *Let $t^I \to s^J$ be a labeled step; and let $t^{I_0} \to s^{J_0}$ be an initially labeled step. Then I, J are label substitutions of I_0, J_0: i.e., $I = \sigma(I_0)$, $J = \sigma(J_0)$ for some label substitution σ.*

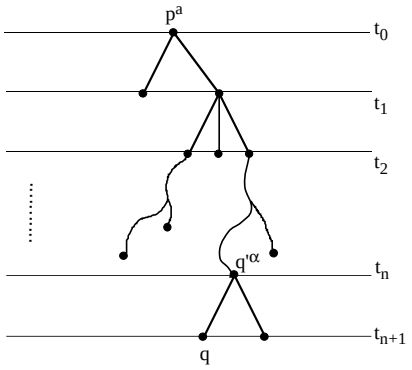


Figure 58: A tree of descendants

Proposition 12.10 *Let $t \equiv t_0 \to t_1 \cdots t_n \equiv s$ be a reduction. Let $\rhd$ be defined by transitivity as suggested in Remark 8.13 and let $\rhd'$ be defined directly without re-initialisation of labels after each step. Then $\rhd = \rhd'$.*

Proof. Induction on n, the length of the reduction.

Basis. If $n = 1$, the statement follows by definition.

Induction step. Consider a reduction $t \equiv t_0 \to t_1 \cdots t_{n+1} \equiv s$. (See Figure 58.)

1. ($\rhd \subseteq \rhd'$.) Take $p \in t_0$, $q \in t_{n+1}$ such that $p \rhd q$. To prove $p \rhd' q$. Give t_0 an initial labeling $I = I_0$, and lift the reduction to

$$(t_0)^{I_0} \to (t_1)^{I_1} \to \cdots \to (t_n)^{I_n} \to (t_{n+1})^{I_{n+1}}.$$

   Let p have label a. We must prove that the label α of q in t_{n+1} contains a. By definition of $\rhd$, there is some q' in t_n such that $p \rhd q' \rhd q$. By induction hypothesis $p \rhd' q'$, so (definition of $\rhd'$) q' in $(t_n)^{I_n}$ has some label α containing a: $\alpha = ---a---$. Now consider $q' \rhd q$. Re-initialising the labels in t_n we have in the step $t_n \to t_{n+1}$: $q'^c \rhd q^{(\cdots c \cdots)}$. By the preceding proposition on label substitutivity we therefore have in $(t_n)^{I_n}$ and $(t_{n+1})^{I_{n+1}}$, respectively, the labeled symbols q'^{α} and $q^{\cdots\alpha\cdots} \equiv q^{\cdots---a---\cdots}$. So the label of q indeed contains a, and therefore $p \rhd' q$.

2. ($\rhd' \subseteq \rhd$.) In the labeled reduction

$$(t_0)^{I_0} \to (t_1)^{I_1} \cdots (t_n)^{I_n} \to (t_{n+1})^{I_{n+1}},$$

   let $p^a \in (t_0)^{I_0}$ and $q^{---a---} \in (t_{n+1})^{I_{n+1}}$, so $p \rhd' q$. To prove: $p \rhd q$. Consider the ancestors of q in t_n with respect to $\rhd$. Clearly

at least one of these ancestors, say q', must have a label containing a. (By (ii) of Proposition 12.9 on label substitutivity.) So $p \rhd q'$ by induction hypothesis, and $p \rhd q' \rhd q$, yielding $p \rhd q$.

□

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:

$$PXT =_\beta T \quad (1)$$
$$PTX =_\beta T \quad (2)$$
$$PFF =_\beta F \quad (3)$$

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

$$BT(P\Omega\Omega) \equiv \Omega \qquad (*)$$

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 equation (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 behaviour (1)-(3). A fortiori there is no such λ-term.

References

[AO93] S. Abramsky and C.-H. L. Ong. Full abstraction in the lazy lambda calculus. *Information and Computation*, 105:159–267, 1993.

[Acz78] P. Aczel. A general Church-Rosser theorem. Technical report, University of Manchester, july 1978.

[BN98] F. Baader and T. Nipkow. *Term rewriting and all that.* Cambridge University Press, Cambridge, 1998.

[Bar71] H.P. Barendregt. *Some extensional term models for combinatory logics and λ-calculi.* PhD thesis, University of Utrecht, 1971.

[Bar84] H.P. Barendregt. *The Lambda Calculus, its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics.* North-Holland Publishing Company, revised edition, 1984. (Second printing 1985).

[BBKV76] H.P. Barendregt, J.A. Bergstra, J.W. Klop, and H. Volken. Degrees, reductions and representability in the lambda calculus. Preprint 22, Utrecht University, Department of Mathematics, 1976.

[BKKS87] H. P. Barendregt, J. R. Kennaway, J. W. Klop, and M. R. Sleep. Needed reduction and spine strategies for the lambda calculus. *Inform. and Comput.*, 75(3):191–231, 1987.

[BI96] A. Berarducci and B. Intrigila. Church-Rosser λ-theories, infinite λ-terms and consistency problems. In C. Steinhorn W. Hodges, M. Hyland and J. Truss, editors, *Logic : from foundations to applications; European logic colloquium 93, Keele, Staffordshire, England*, pages 33–58, Oxford, 1996. Oxford University Press.

[Ber78] G. Berry. Séquentialité de l'évaluation formelle des lambda-expressions. In B. Robinet, editor, *Transformations de Programmes; 3^e Colloque International sur la Programmation, Paris, France*, pages 66–80, Paris, 1978. Dunod.

[Ber79] G. Berry. *Modèles complètement adéquats et stables des λ-calculs typés.* PhD thesis, University of Paris 7, 1979.

[Ber92] Y. Bertot. Origin functions in λ-calculus and term rewriting systems. In J.-C. Raoult, editor, *17th Collquium on Trees in Algebra and Programming, CAAP'92, Rennes, France, February 26–28,1992*, volume 581 of *Lecture Notes in Computer Science*, pages 49–65. Springer-Verlag, 1992.

[Bou85] G. Boudol. Computational semantics of term rewriting systems. In Maurice Nivat and John C. Reynolds, editors, *Algebraic methods in semantics (Fontainebleau, 1982)*, pages 169–236. Cambridge Univ. Press, Cambridge, 1985.

[Chu41] Alonzo Church. *The Calculi of Lambda Conversion.* Princeton University Press, Princeton, 1941.

[CR36] A. Church and J.B. Rosser. Some properties of conversion. *Transactions of the American Mathematical Society*, 39:472–482, 1936.

[Cur93] P.-L. Curien. *Categorical combinators, sequential algorithms and functional programming.* Birkhäuser, Boston, second edition, 1993.

[CF58] H.B. Curry and R. Feys. *Combinatory Logic*, volume I. North-Holland, Amsterdam, 1958.

[DJ90] N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Formal Methods and Semantics, Handbook of Theoretical Computer Science, Volume B*, chapter 6, pages 243–320. MIT Press, 1990.

[DKP89] N. Derschowitz, S. Kaplan, and D.A. Plaisted. Infinite normal forms. In *ICALP'89*, volume 372 of *Lecture Notes in Computer Science*, pages 249–262, 1989.

[FW89] W.M. Farmer and R.J. Watro. Redex capturing in term graph rewriting. Technical report, M89-59, MITRE, 1989.

[FT94] J. Field and F. Tip. Dynamic dependence in term rewriting systems and its application to program slicing. In M. Hermenegildo and J. Penjam, editors, *Proceedings of the Sixth International Symposium on Programming Language Implementation and Logic Programming, PLILP'94*, volume 844 of *Lecture Notes in Computer Science*, pages 415–431. Springer-Verlag, 1994.

[GK94] J. Glauert and Z. Khasidashvili. Relative normalization in orthogonal expression reduction systems. In N. Dershowitz and N. Lindenstrauss, editors, *Workshop on Conditional (and Typed) Term Rewriting Systems, CTRS'94*, volume 968 of *Lecture Notes in Computer Science*, pages 144–165. Springer Verlag, 1994.

[GK96] J. Glauert and Z. Khasidashvili. Relative normalization in deterministic residual structures. In H. Kirchner, editor, *Proceedings of the 19th International Colloquium on Trees in Algebra and Programming, CAAP'96*, volume 1059 of *Lecture Notes in Computer Science*, pages 180–195. Springer Verlag, 1996.

[GLM92] Georges Gonthier, Jean-Jacques Lévy, and Paul-André Melliès. An abstract standardisation theorem. In *Proceedings of 7th Annual Symposium on Logic in Computer Science, LICS'92*, pages 72–81, Santa Cruz, California, 1992. IEEE Computer Society Press.

[Hin69] J.R. Hindley. An abstract form of the Church-Rosser theorem, I. *Journal of Symbolic Logic*, 34(4):545–560, 1969.

[Hin74] J.R. Hindley. An abstract Church-Rosser theorem, II; applications. *Journal of Symbolic Logic*, 39(1):1–22, 1974.

[Hin77] R. Hindley. The equivalence of complete reductions. *Transactions of the American Mathematical Society*, 229:227–248, 1977.

[HL91] G. Huet and J.-J. Lévy. Computations in orthogonal rewriting systems. In J.-J. Lassez and G. Plotkin, editors, *Computational Logic: Essays in honor of Alan Robinson*, pages 395–443. The MIT Press, Cambridge MA, 1991.

[Hud88] P. Hudak. Report on the functional programming language Haskell. Draft Proposed Standard, 1988.

[KKSV93] J.R. Kennaway, J.W. Klop, M.R. Sleep, and F.J. de Vries. An infinitary Church-Rosser property for non-collapsing orthogonal term rewriting systems. In M.R. Sleep, M.J. Plasmeijer, and M.C.J.D. van Eekelen, editors, *Term Graph Rewriting - theory and practice*, pages 47–59. Wiley, 1993.

[KKSV95a] J.R. Kennaway, J.W. Klop, M.R. Sleep, and F.J. de Vries. Infinitary lambda calculus and Böhm models. In *Proceedings of the Conference on Rewriting Techniques and Applications, RTA95*, volume 914 of *Lecture Notes in Computer Science*, pages 257–270, Kaiserslautern, 1995. Springer-Verlag.

[KKSV95b] J.R. Kennaway, J.W. Klop, M.R. Sleep, and F.J. de Vries. Transfinite reductions in orthogonal term rewriting systems. *Information and Computation*, 119(1):18–38, 1995.

[KKSV97] J.R. Kennaway, J.W. Klop, M.R. Sleep, and F.J. de Vries. Infinitary lambda calculus. *Theoretical Computer Science*, 175(1):93–125, 1997.

[KOV99] J.R. Kennaway, V. van Oostrom, and F.J. de Vries. Meaningless terms in rewriting. *Journal of Functional and Logic Programming*, 1999(1), February 1999. *http://www.cs.tu-berlin.de/journal/jflp/*.

[Kha90] Zurab O. Khasidashvili. β-reductions and β-developments of λ-terms with the least number of steps. In *COLOG-88 (Tallinn, 1988)*, pages 105–111. Springer, Berlin, 1990.

[Kha93] Zurab Khasidashvili. Optimal normalization in orthogonal term rewriting systems. In *Rewriting techniques and applications (Montreal, 1993)*, pages 243–258. Springer, Berlin, 1993.

[KG96] Z. Khasidashvili and J. Glauert. Discrete normalization in deterministic residual structures. In M. Hanus and M. Rodríguez-Artalejo, editors, *Proceedings of the 5th International Conference on Algebraic and Logic Programming, ALP'96*, volume 1139 of *Lecture Notes in Computer Science*, pages 135–149. Springer Verlag, 1996.

[KG97] Z. Khasidashvili and J. Glauert. The geometry of orthogonal reduction spaces. In P. Degano, R. Gorrieri, and A. Marchetti-Spaccamela, editors, *Proceedings of the 24th International Colloquium on Automata, Languages, and Programming, ICALP'97*, volume 1256 of *Lecture Notes in Computer Science*, pages 649–659. Springer Verlag, 1997.

[KO95] Z. Khasidashvili and V. van Oostrom. Context-sensitive conditional expression reduction systems. In *Electronic Notes in Theoretical Computer Science, SEGRAGRA '95, Joint COMPUGRAPH/SEMAGRAPH Workshop on Graph Rewriting and Computation, Volterra*, pages 141–150. 1995.

[Klo80] J.W. Klop. *Combinatory Reduction Systems*. Mathematical Centre Tracts Nr. 127. CWI, Amsterdam, 1980. PhD Thesis.

[Klo90] J. W. Klop. Term rewriting systems. Report CS-R9073, Centre for Mathematics and Computer Science, December 1990.

[Klo92] J.W. Klop. Term rewriting systems. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science, Volume II*. Oxford University Press, 1992.

[KOR93] J.W. Klop, V. van Oostrom, and F. van Raamsdonk. Combinatory reduction systems, introduction and survey. *Theoretical Computer Science*, 121:279–308, 1993.

[KV91] J.W. Klop and R.C. de Vrijer. Extended rewriting systems. In S. Kaplan and M. Okada, editors, *Proceedings of the Workshop on Conditional an Typed Rewriting Systems*, volume 516 of *Lecture Notes in Computer Science*, pages 26–50, Montreal, 1991. Springer.

[Kri93] J.L. Krivine. *Lambda-calculus, types and models.* Masson, Paris, 1993.

[Kup94] J. Kuper. *Partiality in Logic and Computation.* PhD thesis, University of Twente, 1994.

[Lév75] J.-J. Lévy. An algebraic interpretation of λ-calculus and a labelled λ-calculus. In C. Böhm, editor, *λ-calculus and Computer Science*, volume 37 of *Lecture Notes in Computer Science*, pages 147–165. Springer-Verlag, Berlin, 1975.

[Lév78] J.-J. Lévy. *Réductions correctes et optimales dans le lambda-calcul.* PhD thesis, Université de Paris VII, 1978.

[Mar92] Luc Maranget. *La stratégie paresseuse.* PhD thesis, L'Université Paris VII, 1992.

[Mel97] Paul-André Melliès. Axiomatic rewriting theory III: A factorisation theorem in rewriting theory. In *Proceedings of the 7th Conference on Category Theory and Computer Science, CTCS'97*, volume 1290 of *Lecture Notes in Computer Science*, pages 46–68, Santa Margherita Ligure, 1997. Springer Verlag.

[Mel98] Paul-André Melliès. Axiomatic rewriting theory IV: A stability theorem in rewriting theory. In *Proceedings of the 14th Annual Symposium on Logic in Computer Science, LICS'98*, pages 287–298, Indianapolis, 1998. IEEE Computer Society Press.

[Mid97] A. Middeldorp. Call by need computations to root-stable form. In *Proceedings of the 24th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL'97*, pages 94–105. 1997.

[New42] M.H.A. Newman. On theories with a combinatorial definition of 'equivalence'. *Annals of Mathematics*, 43(2):223–243, 1942.

[NGV94] R.P. Nederpelt, J.H. Geuvers, and R.C. de Vrijer. *Selected Papers on Automath*, volume 133 of *Studies in Logic and the Foundations of Mathematics.* North-Holland, Amsterdam, 1994.

[Nip91] T. Nipkow. Higher-order critical pairs. In *Proceedings of the sixth annual IEEE Symposium on Logic in Computer Science*, pages 342–349. IEEE Computer Society Press, 1991.

[Nip93] T. Nipkow. Orthogonal Higher-Order Rewrite Systems are Confluent. In *Proceedings of the International Conference on Typed Lambda Calculi and Application*, pages 306–317, 1993.

[O'D77] M.J. O'Donnell. *Computing in systems described by equations.*, volume 58 of *Lecture Notes in Computer Science.* Springer-Verlag, 1977.

[Oos94] Vincent van Oostrom. *Confluence for Abstract and Higher-Order Rewriting.* PhD thesis, Vrije Universiteit, Amsterdam, March 1994.

[Oos96] V. van Oostrom. Take five. Technical report, Vrije Universiteit, Amsterdam, IR-406, 1996.

[Oos97a] V. van Oostrom. Developing developments. *Theoretical Computer Science*, 175(1):159–181, 1997.

[Oos97b] V. van Oostrom. Finite family developments. In H. Comon, editor, *Rewriting Techniques and Applications. 8th International Conference, RTA 97*, volume 1232 of *Lecture Notes in Computer Science*, pages 308–322. Springer, 1997.

[Oos99] V. van Oostrom. Normalisation in weakly orthogonal rewriting. In P. Narendran and M. Rusinowitch, editors, *10th International Conference on Rewriting Techniques and Applications, RTA'99, Trento, June 1999*, volume 1631 of *Lecture Notes in Computer Science*, pages 60–74. Springer, 1999.

[OR94] V. van Oostrom and F. van Raamsdonk. Weak orthogonality implies confluence: The higher-order case. In A. Nerode and Yu.V. Matiyasevich, editors, *Proceedings of the third International Symposium on Logical Foundations of Computer Science*, volume 813 of *Lecture Notes in Computer Science*, pages 379–392. Springer, 1994.

[PvE93] M.J. Plasmeijer and M.C.J.D. van Eekelen. *Functional Programming and Parallel Graph Rewriting*. Addison Wesley, 1993.

[Plo78] G. Plotkin. Church-Rosser categories. Manuscript, 1978.

[Raa93] F. van Raamsdonk. Confluence and superdevelopments. In C. Kirchner, editor, *Proceedings of the 5th International Conference on Rewriting Techniques and Applications (RTA 93)*, volume 690 of *Lecture Notes in Computer Science*, pages 168–182. Springer-Verlag, 1993.

[Raa96] F. van Raamsdonk. *Confluence and Normalisation for higher-order rewriting*. PhD thesis, Vrije Universiteit Amsterdam, 1996.

[Tak95] M. Takahashi. Parallel reductions in λ-calculus. *Information and Computation*, 118:120–127, 1995.

[Tip95] F. Tip. *Generation of program analysis tools*. PhD thesis, Universiteit van Amsterdam, 1995.

[Tur85] D.A. Turner. Miranda: a non-strict functional language with polymorphic types. In J.-P. Jouannaud, editor, *Proceedings of the ACM Conference on Functional Programming Languages and Computer Architecture*, volume 201 of *Lecture Notes in Computer Science*, pages 1–16. Springer-Verlag, 1985.

[Vri85] R.C. de Vrijer. A direct proof of the finite development theorem. *Journal of Symbolic Logic*, 50(2):339–343, 1985.

[Vri87] R.C. de Vrijer. *Surjective pairing and strong normalization: two themes in lambda calculus*. PhD thesis, University of Amsterdam, 1987.

[Vri89] R.C. de Vrijer. Extending the lambda calculus with surjective pairing is conservative. In *Proceedings of the 4th IEEE Symposium on Logic in Computer Science*, pages 204–215, Asilomar, California, 1989.