Take five

Vincent van Oostrom

June 5, 1996

Abstract

This is a collection of five short notes on λ-calculus:

An Easy Expansion Exercise, written at NTT, Basic Research Laboratories, Atsugi, Japan.

FD à la Church and *FD à la Tait*, written at Puri Ulun Carik, Ubud, Bali, Indonesia.

Zooming-in on Omega and $\lambda xxx.x$ *&* xxx, written at TUM, München, Germany.

They were written as divertimenti for myself. In the notes some novel ways of applying known techniques to obtain known results are presented (except for the last note which actually contains a new result).

An Easy Expansion Exercise

Exercise 3.5.11.(vii) of [Bar84], displays two lambda terms $(\lambda x.bx(bc))c$ and $(\lambda x.xx)(bc)$ due to Plotkin, having a common one step beta reduct $bc(bc)$, but which do not have a common beta expand, i.e. are not *upward confluent*.

Conjecture 3.2.38 of [Oos94], displays two other lambda terms $(\lambda x.a(b(x)))(c(d))$ and $a((\lambda y.b(c(y)))d)$, having a common one step beta reduct $a(b(c(d)))$, but which were conjectured not to have a common beta expand. Here we prove this conjecture by elementary means.[1]

Suppose $s \equiv (\lambda x.a(b(x)))(c(d)) \twoheadleftarrow r \twoheadrightarrow a((\lambda y.b(c(y)))d) \equiv t$. By standardisation [Bar84, p. 296] the common expand can be reached by standard expansions: $s \twoheadleftarrow_s r \twoheadrightarrow_s t$, which we may assume to differ already in the first step from r. Let's display the leftmost of these two redexes: $r \equiv \ldots(\lambda z.M)N\ldots$, and call the standard rewrite contracting this redex σ and the other one τ. By definition of standardisation, the displayed part $\ldots(\lambda z$ is preserved by τ.

1. If τ ends in t, then since t contains only one λ, we must have that $z \equiv y$ and $r \equiv a((\lambda y.---$, which cannot rewrite to s,

2. If τ ends in s, then since s contains only one λ, we must have that $z \equiv x$ and $r \equiv (\lambda x.M)N$, with $M \twoheadrightarrow a(b(x))$ and $N \twoheadrightarrow c(d)$. Because of this last fact one can trace the descendants of N in the standard reduction $\sigma : r \equiv (\lambda x.M)N \rightarrow_s M[x := N] \twoheadrightarrow_s a((\lambda y.b(c(y)))d) \equiv t$. We know by confluence that any such descendant must reduce to $c(d)$. It is trivial to check that t does not contain subterms reducing to $c(d)$, so t does not contain descendants of N, hence we also have $M \equiv M[x := x] \twoheadrightarrow_s a((\lambda y.b(c(y)))d)$. Since $M \twoheadrightarrow a(b(x))$, this would contradict confluence.

The advantage of our terms over Plotkin's is that they are also elements of restricted lambda calculi such as linear and typed lambda calculus (the most complex type needed is $o \rightarrow o$). Moreover, the terms make clear that different ways of 'splitting' a term are one cause for failure of (local) upward confluence of beta.

As remarked by Paula Severi ([Sev95]) from this every term can be shown to be non-upward confluent, by using the K-combinator. We do not know whether upward confluence also fails in linear lambda calculus. As an interesting aside, Statman showed that for combinatory logic atoms and only those are upward confluent ([Sta93]).

References

[Bar84] H. P. Barendregt. *The Lambda Calculus, Its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. Elsevier Science Publishers B.V., Amsterdam, revised edition, 1984.

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

[Sev95] Paula Severi, 1995. Indirect personal communication via Femke van Raamsdonk.

[Sta93] Rick Statman. Some examples of non-existent combinators. *Theoretical Computer Science*, 121(1–2):441–448, December 1993.

[1] The same (even simpler) technique can be applied to Plotkin's example.

FD à la Church

We discuss some variations on a proof by translation into Church's λI-calculus, of the fact that all developments are finite in λ-calculus.

Introduction We assume the reader to be familiar with the basic syntactic theory of lambda calculus [Bar84] and only recapitulate main/less familiar no(ta)tions.

DEFINITION 1. The set $\underline{\Lambda}$ of *underlined* lambda terms is inductively defined by:

($\underline{\text{var}}$) $x \in \underline{\Lambda}$, for all (countable many) variables x,

($\underline{\text{app}}$) $M, N \in \underline{\Lambda} \implies MN \in \underline{\Lambda}$,

($\underline{\text{abs}}$) $M \in \underline{\Lambda} \implies \lambda x.M \in \underline{\Lambda}$,

($\underline{\text{beta}}$) $M, N \in \underline{\Lambda} \implies (\underline{\lambda} x.M)N \in \underline{\Lambda}$.

The rewrite relation on $\underline{\Lambda}$ is generated by the rule: $\underline{\beta}$: $(\underline{\lambda} y.P)Q \to P^{[y \mapsto Q]}$.

A *development* in λ-calculus is the projection (by forgetting the underlining) $|M_0| \to_\beta |M_1| \to_\beta \ldots$ of a rewrite $M_0 \to_{\underline{\beta}} M_1 \to_{\underline{\beta}} \ldots$ in $\underline{\Lambda}$. A term is said to be *strongly normalising* (SN) if there are no infinite rewrites starting from it.

THEOREM 2 (Finiteness of Developments (FD)). $\underline{\Lambda}$ *is* SN.

Although many proofs are known of this theorem, in the rest of this note we will be concerned with proving it. At several places in literature, it was remarked that FD can be proven by translating $\underline{\Lambda}$ into some strongly normalising lambda calculus, where the translation preserves rewriting and reflects SN. In [Kri90], [OR, Ghi], and [RS95] lambda calculus with intersection types, simply typed lambda calculus, and $\mathcal{SN}$ were used as respective target calculi. Here we use a memo(ry) calculus ($\underline{\Lambda}^m$) as target calculus.

DEFINITION 3. The set $\underline{\Lambda}^m$ of memo terms is defined by the same clauses as $\underline{\Lambda}$, but for ($\underline{\text{beta}}$):

($\underline{\text{beta}^m}$) $M, P, N \in \underline{\Lambda}^m \implies (\underline{\lambda} x.MP)N \in \underline{\Lambda}^m$, if x has at least one free occurrence in MP.

The embedding function $\iota : \underline{\Lambda} \to \underline{\Lambda}^m$ maps each $\underline{\Lambda}$-construct except ($\underline{\text{beta}}$) onto its $\underline{\Lambda}^m$ pendant,

$$\iota((\underline{\lambda} x.M)N) =^{\text{def}} (\underline{\lambda} x.\iota(M)x)\iota(N)$$

The projection function $\pi : \underline{\Lambda}^m \to \underline{\Lambda}$ maps each $\underline{\Lambda}^m$-construct onto the corresponding one for $\underline{\Lambda}$, but for ($\underline{\text{beta}^m}$) for which π is defined by:

$$\pi((\underline{\lambda} x.MP)N) =^{\text{def}} (\underline{\lambda} x.\pi(M))\pi(N)$$

$\underline{\Lambda}^m$ is closed under rewriting and $\pi \circ \iota = \mathsf{id}$. The embedding ι can be lifted to rewrite sequences.

LEMMA 4 (Lifting). *Let* $\pi(M) \to_{\underline{\beta}} N'$*, then there exists* N*, such that* $M \to_{\underline{\beta}} N$ *and* $\pi(N) = N'$.

Since no β-redexes can be created in $\underline{\Lambda}$, we need not worry in the proof of the lemma whether the 'memory' P prevents creation of redexes in $\underline{\Lambda}^m$ (cf. [Klo80]). The lemma reduces the question of strong normalisation of M to the same question of $\iota(M)$.

Church One observes that $\iota(M)$ is a term in (an underlined version of) Church's λI-calculus.

THEOREM 5 ((global) Conservation). *For any non-erasing orthogonal rewriting system, strong and weak normalisation coincide.*

PROOF This was first proven by Church for λI-calculus [Chu41, p. 26, 7 XXXI], and later generalised by Klop to combinatory reduction systems [Klo80, Thm. II.5.9.3]. A standard argument runs as follows. For orthogonal rewriting systems, developments satisfy the diamond property (see [Oos94] for notation):

$$\begin{tikzcd} M \arrow[r, "\circ", "U"'] \arrow[d, "\circ"', "V"] & N \arrow[d, "\circ"', "U\backslash V"] \\ P \arrow[r, "\circ", "V\backslash U"'] & Q \end{tikzcd}$$

where the common reduct Q can be reached from M either by first developing redexes in U and then the descendants of V after U, or by developing redexes in V followed by the descendants of U after V. If the system is non-erasing, we know moreover that $V \backslash U = \emptyset$ iff $U \subseteq V$. Let $M \mathrel{-\!\!\circ\!\!\to}_V N$, where N is a normal form, and consider a (possibly infinite) rewrite sequence $d\colon M \mathrel{-\!\!\circ\!\!\to}_U M_1 \mathrel{-\!\!\circ\!\!\to}_{U_1} M_2 \mathrel{-\!\!\circ\!\!\to}_{U_2} \ldots$, then we have

$$\begin{tikzcd} M \arrow[r, "\circ", "U"'] \arrow[d, "\circ"', "V"] & M_1 \arrow[r, "m"] \arrow[d, "\circ"', "V_1"] & M_{m+1} \equiv N \arrow[d, "\circ"', "V_{m+1} = \emptyset"] \\ N \arrow[r, "\circ", "\emptyset"'] & N \arrow[r, "m \; \circ", "\emptyset"'] & N \end{tikzcd}$$

for some m. This holds, since d must be a development of V-redexes, and all developments are finite. Repeated application yields the desired result that if a rewrite sequence $M \twoheadrightarrow N$ to normal form N exists, M must be SN. $\odot$

REMARK 6. The theorem corresponding to Theorem 5 as [Bar84, Thm. 11.3.7] corresponds to [Bar84, Thm. 11.3.4], reads as follows.

THEOREM 7 ((local) Conservation). *For any orthogonal rewriting system non-erasing rewrite steps reflect* SN.

This was recently shown to hold for the class of higher-order rewriting systems [Mel95].

The conservation theorem can be applied to show FD by observing that the rightmost innermost strategy reduces $\iota(M)$ to normal form, so $\iota(M)$ and hence M itself is strongly normalising.

Unfortunately, there seems to be a circularity in the argument, since FD is used in the proof of the Conservation Theorem. However, in the proof only FD for *non-erasing* rewriting systems is needed. In [Chu41, p. 20, 7 XXV] FD for λI-calculus is proven, by bounding the length of arbitrary developments by that of standard developments (i.e. developments which are standard in the sense of [Bar84]).[1,2] Here we present a method due to Hyland [Hyl73], which can be viewed at as avoiding the circularity by restricting developments $\mathrel{-\!\!\circ\!\!\to}$ in the proof above to parallel rewrite steps $\mathrel{+\!\!\!|\!\!\to}$. To show that parallel rewrite steps satisfy the diamond property, one needs that $\underline{\Lambda}$ satisifies the *disjointness property* (DP) meaning that all descendants of a redex are disjoint (not nested). Although the method was originally introduced for underlined λ-calculus, we present it here only for the, technically more convenient, case of underlined λI-calculus.

[1] For λ-calculus this does not hold, and a different notion of 'standard' is needed, cf. [RS95].

[2] Based on the above, one could say that Church had all the tools to prove FD for λ-calculus.

LEMMA 8. $\underline{\lambda I} \models \mathrm{DP}$.

DEFINITION 9. An *M-path* σ, is a sequence of positions in M, inductively defined by:

($\underline{\text{var}}$) ϵ is an x-path,

($\underline{\text{app}}$) if σ is an M-path, then $\epsilon, 0\sigma$ is an MN-path, where $i\tau$ denotes the path obtained from τ by prefixing each element with i. If σ is an N-path, then $\epsilon, 1\sigma$ is an MN-path.

($\underline{\text{abs}}$) if σ is an M-path, then $\epsilon, 0\sigma$ is an $\lambda x.M$-path.

($\underline{\text{beta}}$) if σ is an M-path not ending in x, then $\epsilon, 0, 00\sigma$ is an $(\underline{\lambda}x.M)N$-path. If σ is an M-path ending in x, and τ an N-path, then $\epsilon, 0, 00\sigma, 1\tau$ is an $(\underline{\lambda}x.M)N$-path. x is said to be the binding variable for N on that path.

The usual descendant relation on positions is pointwise extended to paths, with two exceptions. Positions which do not have a descendant are erased. Positions on a path, which are in the argument N of the contracted redex, only descend to the copy of N substituted for the binding variable for N on that path.

Some remarks about paths are in order. Our definition of paths can be viewed as a concrete version of persistent conclusion-to-conclusion paths as defined in [DR, Sec. 2.4]. Every path starts at the root, ends in some variable, and descends to exactly one path along any rewrite step. The positions on a path are in lexicographically increasing order. If some redex is a subterm of another one, there is some path through both of them.[3]

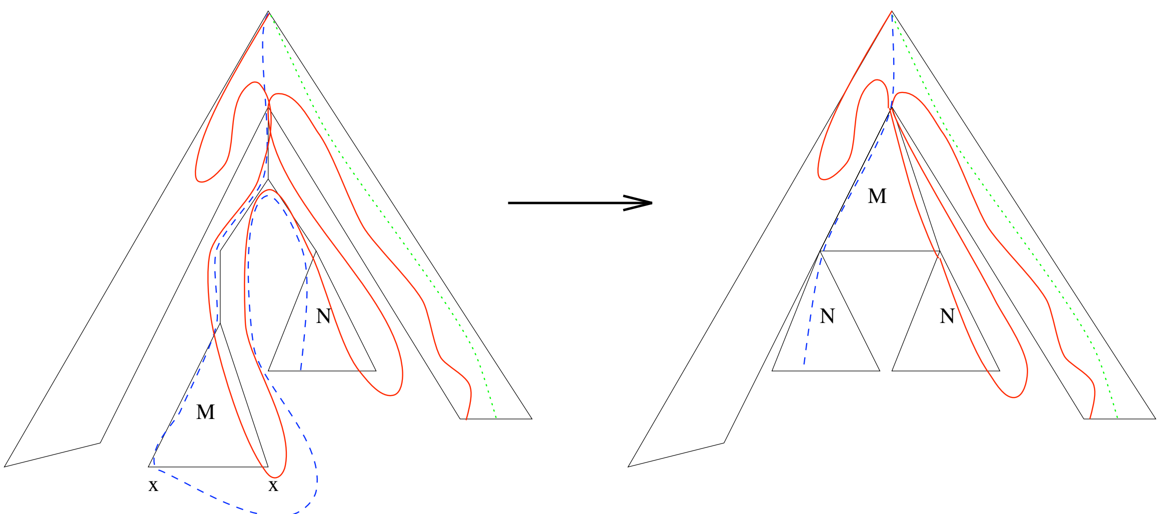

Figure 1: Transformation of paths

LEMMA 10. *The descendant relation induced by contraction of a redex is a bijective correspondence on paths.*[4]

[3]Non-erasingness is essential for this in the case of $\underline{\text{beta}}$.

[4]Bijectivity is lost for erasing calculi, causing technical inconveniences both in the statement of the (corresponding) lemma and its proof.

PROOF By a tedious case analysis (cf. [Klo80, Sec. I.4.3]) or by looking at Figure 1, showing how paths are transformed by contracting the redex $(\underline{\lambda}x.M)N$. Preservation of paths is obvious. Reflection follows from the observation that no path can visit more than one copy of N. This holds, since a path visiting one copy of N can only exit from it through a variable position which was bound outside the entire redex $(\underline{\lambda}x.M)N$. $\odot$

DP can now be proven as follows. Consider an arbitrary path in the initial term of a rewrite sequence and a redex on that path. By the pointwiseness of the descendant relation on paths, we have that at most one residual of the redex is on the residual of the path, anywhere along the rewrite. If two redexes in some term along the rewrite are nested, then there is a path through them, which means they cannot be residuals of one and the same redex.[5]

References

[Bar84] H. P. Barendregt. *The Lambda Calculus, Its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, Amsterdam, revised edition, 1984.

[Chu41] Alonzo Church. *The Calculi of Lambda-Conversion*. Princeton University Press, Princeton, New Jersey, 1941. Second Printing 1951.

[DR] V. Danos and L. Regnier. Proof-nets and the Hilbert space. In [GLR95, pp. 307–328].

[Ghi] Silvia Ghilezan. Application of typed lambda calculi in the untyped lambda calculus. In [NM94, pp. 129–139].

[GLR95] Jean-Yves Girard, Yves Lafont, and Laurent Regnier, editors. *Advances in Linear Logic*, volume 222 of *London Mathematical Society Lecture Notes Series*. Cambridge University Press, 1995.

[HMMN94] Jan Heering, Karl Meinke, Bernhard Möller, and Tobias Nipkow, editors. *Higher-Order Algebra, Logic, and Term Rewriting, First International Workshop, HOA '93, Amsterdam, The Netherlands, September 23–24 1993, Selected Papers*, volume 816 of *Lecture Notes in Computer Science*. Springer-Verlag, 1994.

[Hyl73] J. M. E. Hyland. A simple proof of the Church-Rosser theorem. Technical report, Oxford University, 1973. Typescript.

[Klo80] J. W. Klop. *Combinatory Reduction Systems*. PhD thesis, Rijksuniversiteit Utrecht, June 1980. Mathematical Centre Tracts 127.

[Kri90] J. L. Krivine. *Lambda-calcul, types et modèles*. Études et recherches en informatique. Masson, Paris, 1990.

[Mel95] Paul-André Melliès, 1995. Personal communication.

[NM94] A. Nerode and Yu. V. Matiyasevich, editors. *Third International Symposium, LFCS '94, St. Petersburg, Russia, July 1994, Proceedings*, volume 813 of *Lecture Notes in Computer Science*. Springer-Verlag, 1994.

[Oos94] Vincent van Oostrom. Developing developments. Technical Report: ISRL-94-4, NTT Basic Research Laboratories, December 1994. On `ftp://ftp.cs.vu.nl` as `/pub/oostrom/ISRL-94-4.ps.Z`.

[OR] Vincent van Oostrom and Femke van Raamsdonk. Comparing combinatory reduction systems and higher-order rewrite systems. In [HMMN94, pp. 276–304].

[RS95] F. van Raamsdonk and P. Severi. On normalisation. Report CS-R9545, CWI, June 1995.

[5] FD could be obtained directly from Lemma 10 by showing that the total number of redexes on all paths decreases for every reduction step.

FD à la Tait

The shortest proof of the fact that all developments in lambda calculus are finite is presented.

We assume the reader to be familiar with the basic syntactic theory of (untyped, underlined) lambda calculus [Bar84], and only recapitulate main/less familiar no(ta)tions.

Definition 1. The set $\underline{\Lambda}$ of *underlined* lambda terms is inductively defined by:

- ($\underline{\text{var}}$) $x \in \underline{\Lambda}$, for all (countable many) variables x,
- ($\underline{\text{app}}$) $M, N \in \underline{\Lambda} \implies MN \in \underline{\Lambda}$,
- ($\underline{\text{abs}}$) $M \in \underline{\Lambda} \implies \lambda x.M \in \underline{\Lambda}$,
- ($\underline{\text{beta}}$) $M, N \in \underline{\Lambda} \implies (\underline{\lambda} x.M)N \in \underline{\Lambda}$.

The rewrite relation on $\underline{\Lambda}$ is generated by the rule: $\underline{\boldsymbol{\beta}}$: $(\underline{\lambda} y.P)Q \to P^{[y\mapsto Q]}$.

A *development* in Λ (untyped lambda calculus) is the projection (by forgetting the underlining) $|M_0| \to_\beta |M_1| \to_\beta \ldots$ of a rewrite $M_0 \to_{\underline{\beta}} M_1 \to_{\underline{\beta}} \ldots$ in $\underline{\Lambda}$. A term is said to be *strongly normalising* (SN) if there are no infinite rewrites starting from it.

Theorem 2 (Finiteness of Developments (FD)). $\underline{\Lambda}$ *is* SN.

Proof We prove *for all* $\underline{\Lambda}$*-terms* M *and substitutions* σ *mapping the free variables of* M *to* $\underline{\Lambda}$*-terms in* SN*,* $M^\sigma \in$ SN, by induction on the derivation of $M \in \underline{\Lambda}$, from which the theorem follows taking the identity for σ.

- ($\underline{\text{var}}$) $x^\sigma \equiv \sigma(x) \in$ SN, by assumption,
- ($\underline{\text{app}}$) $(MN)^\sigma \equiv M^\sigma N^\sigma \in$ SN by induction hypothesis for M and N (note that M^σ cannot rewrite to $\underline{\lambda} x.M'$, since $\underline{\Lambda}$ is closed under $\underline{\beta}$),
- ($\underline{\text{abs}}$) $(\lambda x.M)^\sigma \equiv \lambda x.M^\sigma \in$ SN by induction hypothesis for M,
- ($\underline{\text{beta}}$) $((\underline{\lambda} x.M)N)^\sigma \equiv (\underline{\lambda} x.M^\sigma)N^\sigma$. M^σ and N^σ are SN by the induction hypothesis, so an infinite rewrite must be of the form $(\underline{\lambda} x.M^\sigma)N^\sigma \twoheadrightarrow_{\underline{\beta}} (\underline{\lambda} x.M')N' \to_{\underline{\beta}} M'^{[x\mapsto N']} \to^{\infty}_{\underline{\beta}} \ldots$, but $M'^{[x\mapsto N']} \twoheadleftarrow_{\underline{\beta}} M^{\sigma[x\mapsto N^\sigma]} \in$ SN by induction hypothesis. $\odot$

The proof is based on the following standard (and easy to prove) properties of $\underline{\Lambda}$, making it a calculus. (1) $\underline{\Lambda}$ is closed under rewriting. (2) $\underline{\Lambda}$ is closed under substitution. (3) if $M \twoheadrightarrow_{\underline{\beta}} M'$ and $N \twoheadrightarrow_{\underline{\beta}} N'$, then $M^{[x\mapsto N]} \twoheadrightarrow_{\underline{\beta}} M'^{[x\mapsto N']}$, for any variable x.

There are several ways to arrive at this proof. (1) The proof can be viewed upon as a specialisation of Tait's computability proof of SN for simply typed lambda calculus (λ^o) [Tai67], exploiting the observation that induction-loading on the predicate SN is not necessary here since no redexes can be created. (2) Removing the indirection via the set $\mathcal{SN}$ in the proof by Van Raamsdonk and Severi [RS95] gives rise to it. (3) It can be found by looking for a proof which has the same relationship to De Vrijer's proof of FD [Vri87], as Tait's proof has to De Vrijer's proof of SN for λ^o [Vri87].

References

[Bar84] H. P. Barendregt. *The Lambda Calculus, Its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, Amsterdam, revised edition, 1984.

[RS95] F. van Raamsdonk and P. Severi. On normalisation. Report CS-R9545, CWI, June 1995.

[Tai67] W. Tait, W. Intensional interpretation of functional of finite type *I*. *the Journal of Symbolic Logic*, 32:198–212, 1967.

[Vri87] R. C. de Vrijer. *Surjective Pairing and Strong Normalization: Two Themes in Lambda Calculus*. PhD thesis, Universiteit van Amsterdam, January 1987.

Zooming-in on Omega

$\Omega \equiv (\lambda x.xx)\lambda y.yy$ is the smallest term in the lambda calculus with β-reduction which is not normalising. Recently it was observed by Sørensen that Ω is also generic in the sense that it can be textually embedded in any non-strongly normalising term. We present a small variation on his proof using a *zoom-in* (or *constricting*) strategy.

Familiarity with the basic syntactic theory and no(ta)tions of the lambda calculus in [Bar84] is assumed. We write $N \trianglelefteq M$ to denote that N can be textually embedded in M, e.g. $\lambda x.xx \trianglelefteq \lambda z.(\lambda x.zx(\lambda y.x))$, but not $\lambda x.xx \trianglelefteq (\lambda x.x)\lambda x.x$ since we assume the Variable Convention. If $xx \not\trianglelefteq M$ the term $\lambda x.M$, the redex $(\lambda x.M)N$, and a step contracting that redex are called *linear*.

Definition 1 *Let $\infty(M)$. A* minimal *step contracts the leftmost redex of a minimal $\infty(N) \subset M$. A* zoom-in *strategy ([Mel95, Sec. 6.1]) always contracts a minimal step such that the rewrite sequence looks like $M \equiv C[N] \to C[M'] \equiv C[C'[N']] \to C[C'[M'']] \equiv C[C'[C''[N'']]] \to \ldots$.*

Observe that in this definition $\infty(M')$ holds since the leftmost redex in N must be contracted in any infinite rewrite sequence starting from N by the minimality assumption, hence the zoom-in strategy is well-defined. This is an instance of a so-called *constricting* strategy (cf. [Gra95, Rem. 3.3.7]).

Lemma 2 ([Sør95]) *If $\infty(M)$ then $\Omega \trianglelefteq M$.*

Proof It suffices to show that terms in which all non-linear subterms are nested, e.g. $\lambda x.x(\lambda y.yxy)$, are strongly normalising. So consider a *totally nested* term M such that if R and R' are non-linear subterms of M, then $R \subset R'$ or $R' \subset R$. Totally nested terms are closed under rewriting since duplication of non-linear subterms cannot happen by totality and creation of non-linear subterms, e.g. $\lambda y.(\lambda x.xx)y \to \lambda y.yy$, can only happen towards the root by contraction of a non-linear redex preserving totality. In particular, consider a step $C[N] \to C[M']$ in a zoom-in sequence. By the minimality assumption N cannot be of the form $\lambda z.S$ or $z\vec{S}$, so must look like $(\lambda x.P)Q\vec{S}$. Creation of non-linear subterms can only happen if $\lambda x.P$ is non-linear and Q contains variables bound outside N. Since these binders are outside N they stay put afterwards by the definition of zoom-in strategy, motivating measuring the step by the pair $(n, \|N\|)$, where n is the number of non-linear subterms occurring in N. Ordered lexicographically this measure decreases in every step contradicting infinity of the zoom-in strategy and hence of M. $\square$

References

[Bar84] H. P. Barendregt. *The Lambda Calculus, Its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, revised edition, 1984.

[Gra95] Bernhard Gramlich. *Termination and Confluence Properties of Structured Rewrite Systems*. Dissertation, Universität Kaiserslautern, Preliminary version of November 9, 1995.

[Mel95] Paul-André Melliès. *Description Abstraite des Systèmes de Réécriture*. Thèse de doctorat, Université Paris VII, Preliminary version of September 28, 1995.

[Sør95] Morten Heine Sørensen. Embeddings and infinite reduction paths in untyped lambda-calculus. In the 2nd International Workshop on Termination, La Bresse, France, 1995.

I never did, I never did, I never *did* want
"Hold-my-hand";
I never did, I never did, I never *did* think much of
"Not up there, dear!"
It's no good saying it. They don't understand.

A. A. Milne, *When We Were Very Young*. 1924.

$\lambda xxx.x$ & xxx

We toy a bit with the weak Church-Rosser (WCR) property of β-expansion in the λ-calculus. $\lambda xxx.x$ and xxx are shown to be smallest λ-terms (textually, in shorthand notation) allowing violation of the *blocked* and *balanced* conditions, hence the smallest non-WCR λ-terms.

Familiarity with the basic syntactic theory and no(ta)tions of rewriting in general and the λ-calculus in particular is assumed ([Bar84, DJ91]). In $C[M[N]_P]_r$, r specifies the position of $M[N]$ and P the set of positions of N (disjoint and below r), where $FV(N) \subseteq FV(M[N])$. The pair rP is called a *pattern (occurrence)*, P a *cut-set* and the step β-expanding rP is $C[M[N]]\ {}_{rP}\!\leftarrow C[(\lambda x.M)N]$.

Definition 1 (Independence) *Let rP and sQ be patterns. If $r \parallel s$, then $rP \parallel sQ$ (*disjointness*). If $p \le s$ for some $p \in P$, then $rP \le sQ$ (*nesting*). If $r \le s$ and for all $p \in P$, $q \in Q$, $p \not< q$ (*blocked*) and the sets $\{q' | qq' \in P\}$ are the same (*balanced*), then $rP \trianglerighteq sQ$ (*encompassment*). Two patterns in a λ-term and their corresponding β-expansion steps are* independent *if they are disjoint or one nests or encompasses the other.* Dependence *is non-independence, denoted by $\not\simeq$.*

Note that the three cases of independence roughly correspond to the three possible relative positions of two distinct β-contraction steps. Disjoint β-redexes give rise to disjoint patterns and a β-redex inside the argument N (body M) of a redex $(\lambda x.M)N$ results in nested (encompassed) patterns. Balancedness is vacuously true in the case of *affine* β-expansion steps, i.e. if the cut-set contains at most one element. If the cut-set is empty blockedness holds as well, hence the corresponding *ex nihilum* step is independent of any other step.

Lemma 2 (WCR) *Independent β-expansion steps are WCR.*

Proof By cases for patterns rP and sQ in $C[K[L]_P]_r \equiv R \equiv D[M[N]_Q]_s$.

1. If $rP \parallel sQ$, $R \equiv E[K[L], M[N]]$ in which case $E[(\lambda x.K)L, (\lambda y.M)N]$ is a common expand.
2. If $rP \le sQ$, $R \equiv C[K[L[M[N]]]]$ and $C[(\lambda x.K)L[(\lambda y.M)N]]$ is a common expand.
3. If $rP \trianglerighteq sQ$, then define $P_{\parallel} = \{p \in P | s \le p \& \nexists q \in Q. q \le p\}$ and $P_{\ge} = \{p \in P | \exists q \in Q. q \le p\}$. Then $R \equiv C[K[E[x]_{P_{\parallel}}[F[x]_{P_{\ge}}]][L]]$, where $E[x := L] \equiv M$ using blockedness and $F[x := L] \equiv N$ using balancedness, and $C[(\lambda x.K[(\lambda y.E[x])F[x]])L]$ is a common expand. □

The diagrams resulting from the constructions of the common expand of independent steps are usually referred to as *elementary* diagrams.

Example 3 (Accidents Syntaxiques) *Let $\mathbf{I} \equiv \lambda x.x$.*

1. $(\lambda x.\mathbf{I}(\mathbf{I}(x)))(\mathbf{II}) \to_{\epsilon\{11\}} \mathbf{I}(\mathbf{I}(\mathbf{II}))\ {}_{1\{111\}}\!\leftarrow \mathbf{I}((\lambda y.\mathbf{I}(\mathbf{I}y))\mathbf{I})$. *Since $1 \not\le \epsilon$ and $111 > 11$ neither step encompasses the other and $\epsilon\{11\} \not\simeq 1\{111\}$, but $\mathbf{I}((\lambda x.\mathbf{I}(\mathbf{I}x))(\mathbf{II}))$ is a common expand.*
2. $(\lambda y.yy\mathbf{I})\mathbf{I} \to_{\epsilon\{00,01\}} \mathbf{III}\ {}_{\epsilon\{01,1\}}\!\leftarrow (\lambda y.\mathbf{I}yy)\mathbf{I}$. *Since $\{p' | 00p' \in \{01,1\}\} \ne \{p' | 01p' \in \{01,1\}\}$ and $\{p' | 01p' \in \{00,01\}\} \ne \{p' | 1p' \in \{00,01\}\}$ neither step encompasses the other and $\epsilon\{00,01\} \not\simeq \epsilon\{01,1\}$. Nevertheless $(\lambda y.\mathbf{I}yy)((\lambda y.yy\mathbf{I})\mathbf{I})$ is a common expand.*
3. *Write $\langle M, P\rangle$ (M with* memory *P) for $(\lambda x.M)P$ where $x \notin FV(M)$ and consider a divergence consisting of two ex nihilum steps $\langle L, M\rangle \to_{\epsilon\emptyset} L\ {}_{\epsilon\emptyset}\!\leftarrow \langle L, N\rangle$ for arbitrary M and N. Since the patterns are equal they encompass each other. Their common expands as computed in the lemma, $\langle\langle L, M\rangle, N\rangle$ and $\langle\langle L, N\rangle, M\rangle$, differ.*
4. *$\mathbf{I}L \to_{\epsilon\{\epsilon\}} L\ {}_{\epsilon\emptyset}\!\leftarrow \langle L, M\rangle$. We have $\epsilon\{\epsilon\} \le \epsilon\emptyset$, $\epsilon\{\epsilon\} \trianglerighteq \epsilon\emptyset$ as well as $\epsilon\emptyset \trianglerighteq \epsilon\{\epsilon\}$. The computed common expands are $\mathbf{I}\langle L, M\rangle$ and $\langle \mathbf{I}L, M\rangle$.*

That the weak Church-Rosser property in the first two items holds in spite of dependence of the initial steps is a bit lucky, as is shown by the next lemma.

Lemma 4 (Non-WCR) *Dependent β-expansion steps from a β-normal form are not WCR.*

Proof The proof assumes familiarity with the notions in [Lév78]. We show that the final steps of two permutation equivalent rewrite sequences are independent (or identical), from which the lemma follows since all coinitial reductions to normal form are permutation equivalent. For a contradiction let $M \to_x; \twoheadrightarrow_\sigma; \to_u N$ and $M \to_y; \twoheadrightarrow_\tau; \to_v N$, where $x;\sigma$ and $y;\tau$ may be assumed standard and also $x <_{lex} y$ may be assumed. By $x <_{lex} y$ and standardness of $y;\tau$, $x/(y;\tau) = x'$ (x' is x after $y;\tau$). By permutation equivalence $x'/v = 0$, so $v \leq x'$ and there exists a unique $v' \leq x$ such that $v'/(y;\tau) = v$.

1. If $v' < x$, then by standardness of $x;\sigma$, $v'/(x;\sigma) = v''$. By permutation equivalence $v''/u = 0$, so $u \leq v''$ and one easily concludes $v = u$.

2. Suppose $v' = x$ and contributes to u. Consider the initial labelling of the two rewrite sequences, then using the first rewrite sequence the label of v occurs with one over/underlining in the final result since v is contracted as the last step, but using the second sequence it occurs with at least two over/underlinings in the final result since v contributed to u. quod non.

3. Suppose $v' = x$ and does not contribute to u. If $u \leq_{lex} v'$ then one can proceed as in the first case. Let $v' : C[(\lambda x.K)L] \to_\beta C[K[L]]$. The separation of L from the rest can be maintained along σ by standardness, $\sigma : C[K[L]] \twoheadrightarrow C[K'[\vec{L'}]]$, where $\sigma_K : K[x] \twoheadrightarrow K'[\vec{x}]$ and $\sigma_i : L \twoheadrightarrow L'_i$. We distinguish cases according to where u is contracted.

   (a) Let u be inside K'. If no L has descendants after u, σ can be lifted to $\sigma' : C[(\lambda x.K)L] \twoheadrightarrow C[(\lambda x.K')L]$ via σ_K. If some L_i has descendants after u, all descendants of L must be in normal form L'_i, so σ can be lifted to $\sigma' : C[(\lambda x.K)L] \twoheadrightarrow C[(\lambda x.K'[\vec{x}])L']$ via $\sigma_i;\sigma_K$.

   (b) Let u be inside L'_i. We lift σ to $\sigma' : C[(\lambda x.K)L] \twoheadrightarrow C[(\lambda x.K'[\vec{x}])L'_i]$ via σ_i.

   (c) no other cases are possible by standardness and since v' did not contribute to u.

   In each of the three cases, $v';\sigma;u$ is permutation equivalent to $\sigma';v'';U$ for $v'' = v'/\sigma'$ and $U/v'' = u$. Since $v''/U = v$ (v is external!) and $U/v'' = u$, we have constructed an independence diagram. □

Example 5 (Non-Accidents) *1. Let $(\lambda x.a(b(x)))(cd) \to_{\epsilon\{11\}} a(b(cd))\ {}_{1\{111\}}\!\leftarrow a((\lambda y.b(cy))d)$ ([Oos94, Conj. 3.2.38]). One computes as above $\epsilon\{11\} \not\leftrightarrows 1\{111\}$ (blocks are disturbed) and we conclude non-WCR from the lemma.*

2. Consider the divergence $(\lambda x.bx(bc))c \to_{\epsilon\{01\}} bc(bc)\ {}_{\epsilon\{0,1\}}\!\leftarrow (\lambda x.xx)(bc)$ due to Plotkin ([Bar84, Exe. 3.5.11.vii]). One computes as above that $\epsilon\{01\} \not\leftrightarrows \epsilon\{0,1\}$ (either a block or balance are disturbed) and we conclude non-WCR.

To disturb a block one easily computes that at least 4 symbols nested inside each other are needed and $\lambda xxx.x$ is the shortest term achieving this. To disturb balance at least 5 symbols are needed, and xxx has them. It is left to the reader to construct the non-WCR divergences.

References

[Bar84] H. P. Barendregt. *The Lambda Calculus, Its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, revised edition, 1984.

[DJ91] Nachum Dershowitz and Jean-Pierre Jouannaud. Notations for rewriting. *Bulletin of the European Association for Theoretical Computer Science*, 43:162–172, February 1991.

[Lév78] Jean-Jacques Lévy. *Réductions correctes et optimales dans le λ-calcul.* Thèse de doctorat d'etat, Université Paris VII, 1978.

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