Clocked Lambda Calculus [†]

Jörg Endrullis, Dimitri Hendriks, Jan Willem Klop, and Andrew Polonsky

{j.endrullis,r.d.a.hendriks,j.w.klop,a.polonsky}@vu.nl

VU University Amsterdam, Department of Computer Science

One of the best-known methods for discriminating λ-terms with respect to β-convertibility is due to Corrado Böhm. The idea is to compute the *infinitary* normal form of a λ-term M, the *Böhm Tree (BT) of M*. If λ-terms M, N have distinct BTs, then $M \neq_\beta N$, that is, M and N are not β-convertible. But what if their BTs coincide? For example, all fixed point combinators (fpcs) have the same BT, namely $\lambda x.x(x(x(\ldots)))$.

We introduce a *clocked λ-calculus*, an extension of the classical λ-calculus with a unary symbol τ used to witness the β-steps needed in the normalization to the BT. This extension is infinitary strongly normalizing, infinitary confluent, and the unique infinitary normal forms constitute enriched Böhm Trees, which we call *clocked Böhm Trees*. These are suitable for discriminating a rich class of λ-terms having the same BTs, including the well-known sequence of Böhm's fpcs.

We further increase the discrimination power in two directions. First, by a refinement of the calculus: the *atomic clocked λ-calculus*, where we employ symbols τ_p that also witness the (relative) positions p of the β-steps. Second, by employing a *localized* version of the (atomic) clocked BTs that has even more discriminating power.

We dedicate our paper to Corrado Böhm
in honour of his 90th birthday, in gratitude and admiration.

1. Introduction

We introduce new techniques for proving non-convertibility of λ-terms, and place our earlier work on clocked Böhm Trees (EHK10; EHKP12) in a new and more elegant setting, giving a first-class status to the clocks in a λ-calculus extended with an explicit unary constructor τ. The idea is that in the normalization to the Böhm Tree (BT), we leave behind an occurrence of τ at a position p to witness a β-step needed to head normalize the subterm at p. The calculus consists of the following two rules:

$$(\lambda x.M)N \to \tau(M[x:=N]) \qquad\qquad \tau(M)N \to \tau(MN)$$

[†] This research has been partially funded by the Netherlands Organisation for Scientific Research (NWO) under grant numbers 612.000.934, 639.021.020, and 612.001.002.

and we call it the *clocked λ-calculus*. It satisfies the desired infinitary properties of infinitary confluence, and infinitary strong normalization, and the unique infinitary normal forms are BTs (in fact, Lévy–Longo Trees) enriched with τ symbols witnessing the β-steps needed in the reduction to the BT. For a large class of λ-terms this yields a discrimination method, as follows: if the infinitary normal forms in the clocked λ-calculus cannot be converted by deleting finitely many τ symbols, then the terms are not β-convertible. This class of terms encompasses the so-called 'simple' terms, that is, terms that never duplicate redexes in the reduction to the BTs, see (EHK10; EHKP12).

We further enhance the discrimination power as follows. We extend the class of simple terms by allowing redex duplication, but requiring that of each redex only finitely many residuals are contracted. Moreover, we introduce a *sieve of time* that fine-tunes the method to a set of positions in the BT, then only requiring that the head reductions at these positions do not contract infinitely many residuals of a single redex.

We also introduce the *atomic clocked λ-calculus* where the τ's also record the position of the β-step they witness:

$$(\lambda x.M)N \to \tau_\epsilon(M[x := N]) \qquad\qquad \tau_p(M)N \to \tau_{\mathrm{L}p}(MN)$$

The need of refined discrimination techniques becomes apparent by studying fixed point combinators (fpcs), which are ideally suited to study Böhm Trees (BTs). Indeed, fpcs yield the simplest infinite BTs that there are: for every fpc Y, the Böhm Tree $\mathsf{BT}(Y) = \lambda x.x^\omega = \lambda x.x(x(x(\ldots)))$, an 'infinite normal form' in the infinitary λ-calculus λ^∞, see (KKSdV97; Ter03; EHK10; EHKP12). Thus they cannot be distinguished by their non-clocked BTs.

Fixed point combinators play an important role in the λ-calculus, namely in the construction of recursively defined terms. Terms M satisfying $M =_\beta C[M]$ where C is a context, can be defined by $M \equiv Y(\lambda x.C[x])$ where Y is an fpc. Fixed point combinators and weak fpcs, a generalization of fpcs, also play an important role in several typed λ-calculi (GW94), where a possibility of typing a wfpc is associated with the emergence of paradoxes.

Before setting up our technical framework, we discuss some related work.

Related Work

The idea of enriching BTs occurred as we recently noted, already in 1989 in a paper by Naoi and Inagaki (NI89) in the setting of first-order rewriting systems. Their definition bears a striking resemblance to our definition of clocked BTs and was meant to express a notion of complexity or efficiency for terms. It was done "by counting, for each node p in the limit, the number of rewriting steps required to obtain p". It was not used for discrimination purposes as we did in (EHK10; EHKP12), and again do in the present paper. The main purpose of (NI89) was to give a *continuous algebra semantics* to TRSs.

A second strand of related work, this time in the setting of λ-calculus, is by Aehlig and Joachimski in 2002 (AJ02). Just as we will do, they employed a 'waiting instruction' like our τ, but the actual setup is different from ours. More precisely, (AJ02) define a normalization function by guarded corecursion (guaranteeing a total, productive function (Coq94)), where the τ constructor serves as a guard, and is returned whenever the argument is not yet in (weak) head normal form. The purpose of (AJ02) was to give a *continuous normalization* strategy. We include τ in an extension of λ-calculus itself, and our main concerns are discrimination techniques. Beyond that, our extension of λ-calculus with τ resembles (AJ02) in the fact that our λ-calculus has the property SN^{∞}, infinitary normalization. Moreover, our calculus satisfies infinitary confluence, CR^{∞}. We remind that ordinary λ-calculus possesses neither of these two properties.

As a historical note, we mention that our τ constructor is very much reminiscent of the *hiaton* suggested by W. Wadge (see (Wad81)), signifying a delay step. It was also written as τ; that notation was suggested by D. Park in (Par83), inspired by the τ-step, or *silent move*, in process algebra, in particular Milner's CCS. Several studies (e.g. (Fau82; Mat85)) were employing this device called 'hiatonisation' in the semantics of programming languages and dataflow networks, with issues such as the well-known Brock–Ackerman anomaly, and Kahn's principle.

2. Preliminaries

To make this paper self-contained, and to fix notations, we recall the main concepts. For further reading on λ-calculus we refer to (Bar84) and (Bet03), and for Böhm, Berarducci and Lévy–Longo Trees to (Bar84; AO93; BKdV00; BK09).

Definition 1. We fix a countably infinite set $\mathcal{X}$ of variables $x, y, z, \ldots$. The set $\mathit{Ter}(\lambda)$ of *finite λ-terms* is inductively defined by the following grammar:

$$M ::= x \mid \lambda x.M \mid M \cdot M \qquad\qquad (x \in \mathcal{X})$$

We use $M, N, \ldots$ to range over the elements of $\mathit{Ter}(\lambda)$.

Usually we suppress the application symbol in a term $M \cdot N$ and write MN for short. We adopt the usual conventions for omitting brackets, i.e., we let application associate to the left, so that $N_1 N_2 \ldots N_k$ denotes $(\ldots(N_1 N_2) \ldots N_k)$, and we let abstraction associate to the right: $\lambda x_1 \ldots x_n.M$ stands for $(\lambda x_1.(\ldots(\lambda x_n.(M))))$.

Definition 2. Let $\Box$ be a fresh constant symbol, i.e., $\Box \notin \mathcal{X}$. Then a *context* is a term containing precisely one occurrence of $\Box$, that is, contexts are defined by

$$C ::= \Box \mid \lambda x.C \mid CM \mid MC \qquad\qquad (x \in \mathcal{X}, M \in \mathit{Ter}(\lambda))$$

We write $\mathit{Con}(\lambda)$ for the set of all finite contexts. For $M \in \mathit{Ter}(\lambda)$ and $C \in \mathit{Con}(\lambda)$ we write $C[M]$ to denote the term obtained from C by replacing the single occurrence of $\Box$ with the term M, that is:

$$\Box[M] = M \quad (\lambda x.C)[M] = \lambda x.C[M] \quad (CN)[M] = C[M]N \quad (NC)[M] = NC[M]$$

The set of finite and infinite terms is defined by interpreting the grammar from Definition 1 *coinductively*, that is, $\mathit{Ter}^{\infty}(\lambda)$ is the largest set X such that every element $M \in X$ is either a variable x, an abstraction $\lambda x.M'$ or an application $M_1 M_2$ with $M', M_1, M_2 \in X$. We will use $::=^{\mathrm{co}}$ to indicate that the grammar has to be interpreted coinductively. For a thorough treatment of coinductive definition and proof principles we refer to (SR12).

Definition 3. The set $\mathit{Ter}^{\infty}(\lambda)$ of (finite and) *infinite* λ*-terms* is defined by the grammar

$$M ::=^{\mathrm{co}} x \mid \lambda x.M \mid MM \qquad\qquad (x \in \mathcal{X})$$

Definition 4. The set of *infinite* contexts, which we denote by $\mathit{Con}^{\infty}(\lambda)$, is defined inductively by the grammar as in Definition 2 with the difference that $M \in \mathit{Ter}^{\infty}(\lambda)$. Context filling, $C[M]$ with $C \in \mathit{Con}^{\infty}(\lambda)$ and $M \in \mathit{Ter}^{\infty}(\lambda)$, is defined as before.

We note that infinite contexts are infinite λ-terms, but their single hole $\Box$ resides at finite depth.

Definition 5. A *position* is a sequence over $\{\lambda, \mathrm{L}, \mathrm{R}\}$. Let $M \in \mathit{Ter}^{\infty}(\lambda)$ and $p \in \{\lambda, \mathrm{L}, \mathrm{R}\}^*$. The *subterm* $M|_p$ *of* M *at position* p is defined as follows:

$$M|_{\epsilon} = M \qquad\qquad (MN)|_{\mathrm{L}p} = M|_p$$
$$(\lambda x.M)|_{\lambda p} = M|_p \qquad\qquad (MN)|_{\mathrm{R}p} = N|_p$$

We let $\mathit{Pos}(M) \subseteq \{\lambda, \mathrm{L}, \mathrm{R}\}^*$ denote the set of positions p such that $M|_p$ is defined.

The *root* of a term M is the outermost constructor of M. The symbol of M at position p, denoted by $M(p)$, is the root of the subterm $M|_p$. These notions are also employed for contexts.

We introduce some further notations. Let $\to_1$ and $\to_2$ be binary relations on terms. We write $\to_1 \cdot \to_2$ for the relational composition of $\to_1$ and $\to_2$, i.e., $M \to_1 \cdot \to_2 N$ iff $M \to_1 P \to_2 N$ for some term P. We write $\to^n$ for the n-fold composition of a relation $\to$, defined by $M \to^0 M$, and $M \to^{n+1} N$ iff $M \to \cdot \to^n N$. We use $\twoheadrightarrow$ to denote the reflexive-transitive closure of $\to$, $\twoheadrightarrow = \cup_{n\in\mathbb{N}} \to^n$. We let $\to^{=}$ denote the reflexive closure of $\to$, $\to^{=} = \to^0 \cup \to^1$.

Definition 6. The relation $\to_{\beta}$ on $\mathit{Ter}(\lambda)$ or $\mathit{Ter}^{\infty}(\lambda)$, called β*-reduction*, is the closure under contexts of the β-rule:

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

where $M[x := N]$ denotes the result of substituting N for all free occurrences of x in M.

We usually omit the subscript β in $\to_{\beta}$ and $\twoheadrightarrow_{\beta}$. For terms $M, N \in \mathit{Ter}^{\infty}(\lambda)$, we write $M \to_p N$ to indicate the witnessing position of the contracted redex, so $M \to_p N$ if there exists a context C such that $C(p) = \Box$, $M \equiv C[(\lambda x.P)Q]$ and $N = C[P[x := Q]]$.

We write $M =_{\beta} N$ to denote that M is β-convertible with N, i.e., $=_{\beta}$ is the equivalence closure of $\to_{\beta}$. For syntactic equality (modulo renaming of bound variables) of λ-terms, we use $\equiv$.

Definition 7. A λ-term M is called a *normal form* if there exists no N with $M \to N$. We say that a term M *has a normal form* if it reduces to one, that is, if $M \twoheadrightarrow N$ for some normal form N. For λ-terms M having a normal form we write $\mathrm{nf}(M)$ to denote the unique normal form N with $M \twoheadrightarrow N$.[†]

Combinators are closed λ-terms, i.e., λ-terms without free variables. Some commonly used combinators are:

$$\mathsf{I} = \lambda x.x \qquad \mathsf{K} = \lambda xy.x \qquad \mathsf{S} = \lambda xyz.xz(yz) \qquad \mathsf{B} = \lambda xyz.x(yz)$$

Definition 8 (Fixed point combinators).

(i) A term Y is a *fixed point combinator (fpc)* if $Yx =_\beta x(Yx)$.
(ii) An fpc Y is *k-reducing* if $Yx \to^k x(Yx)$.
(iii) An fpc Y is *reducing* if Y is k-reducing for some $k \in \mathbb{N}$.
(iv) A term Z is a *weak fpc (wfpc)* if $Zx =_\beta x(Z'x)$ where Z' is again a wfpc.

Example 9. The well-known fpc's of Curry and Turing, Y_0 and Y_1, are defined as follows:

$$\begin{aligned} \mathsf{Y}_0 &\equiv \lambda f.\omega_f\omega_f & \qquad \mathsf{Y}_1 &\equiv \eta\eta \\ \omega_f &\equiv \lambda x.f(xx) & \qquad \eta &\equiv \lambda xf.f(xxf) \end{aligned}$$

Note that Turing's Y_1 is a reducing fpc, whereas Curry's Y_0 is not.

Example 10. Another example of a non-reducing fpc is Hurkens fpc:

$$\mathsf{Y}_{\mathrm{H}} = \lambda f.\alpha_f\alpha_f\omega \qquad \alpha_f = \lambda ab.f(bab) \qquad \omega = \lambda x.xx$$

Then $\mathsf{Y}_{\mathrm{H}}x \to \alpha_x\alpha_x\omega \to^2 x(\omega\alpha_x\omega) \to x(\alpha_x\alpha_x\omega) \leftarrow x(\mathsf{Y}_{\mathrm{H}}x)$. As we were informed by Herman Geuvers in personal communication, this fpc was derived by Tonny Hurkens in a study of Girard's paradox. This term was a 'looping combinator' (before erasing types), which is a well-typed wfpc. However, erasing the type information yielded the proper fpc Y_{H} above.

Example 11. Yet another example of a non-reducing fpc is Y_{MW}, found via mechanical search by McCune and Wos (MW91):

$$\mathsf{Y}_{\mathrm{MW}} = \lambda f.\alpha(\mathsf{B}(\mathsf{B}f))\alpha \qquad \alpha = \lambda ab.abab$$

Then $\mathsf{Y}_{\mathrm{MW}}x \twoheadrightarrow f(\alpha(\mathsf{B}(\mathsf{B}x))\alpha) \leftarrow x(\mathsf{Y}_{\mathrm{MW}}x)$.

Remark 12. The definition of weak fpc's in item (iv) of the above definition is essentially coinductive (SR12), that is, implicitly employing a 'largest set' semantics. In long form, the definition means the following: the set of weak fpc's is the largest set $W \subseteq \mathit{Ter}(\lambda)$ such that for every $Z \in W$ we have $Zx =_\beta x(Z'x)$ for some $Z' \in W$.

A wfpc is alternatively defined as a term having the same Böhm Tree as an fpc, namely $\lambda x.x^\omega \equiv \lambda x.x(x(x(\ldots)))$. In type systems, typable wfpcs are known as 'looping combinators'; see (CH94; GW94).

A third alternative definition of wfpcs is via infinitary λ-calculus: $Zx =_{\lambda^\infty} x(Zx)$.

[†] Uniqueness follows from confluence of the λ-calculus; see, e.g., (Bet03).

Example 13. Define by double recursion[‡], Z and Z' such that $Zx = x(Z'x)$ and $Z'x = x(Zx)$. Then Z, Z' are both wfpc's, and $Zx = x(x(Zx))$. So Z delivers its output twice as fast as an ordinary fpc, but the generator flipflops.

Example 14. An example of a weak fpc is the term $\mathsf{A}(\mathsf{BAB})$ where $\mathsf{A} \equiv \mathsf{BM}$ and $\mathsf{M} \equiv \lambda x.xx$. This example was found by Statman, in his study of terms composed only of symbols B and M. Here the generator changes in each 'production cycle'. We have the following reduction:

$$
\begin{array}{ll}
 & \mathsf{A}(\mathsf{BAB})x \\
\to^3 & \mathsf{M}(\mathsf{BAB}x) \\
\to & \mathsf{BAB}x(\mathsf{BAB}x) \\
\to^3 & \mathsf{A}(\mathsf{B}x)(\mathsf{BAB}x) \\
\to^3 & \mathsf{M}(\mathsf{B}x(\mathsf{BAB}x)) \\
\to & \mathsf{B}x(\mathsf{BAB}x)(\mathsf{B}x(\mathsf{BAB}x)) \\
\to^3 & x(\mathsf{BAB}x(\mathsf{B}x(\mathsf{BAB}x))) \\
\to^3 & x(\mathsf{A}(\mathsf{B}x)(\mathsf{B}x(\mathsf{BAB}x))) \\
\to^3 & x(\mathsf{M}(\mathsf{B}x(\mathsf{B}x(\mathsf{BAB}x)))) \\
\to & x(\mathsf{B}x(\mathsf{B}x(\mathsf{BAB}x))(\mathsf{B}x(\mathsf{B}x(\mathsf{BAB}x)))) \\
\to^3 & x(x(\mathsf{B}x(\mathsf{BAB}x)(\mathsf{B}x(\mathsf{B}x(\mathsf{BAB}x))))) \\
\to^3 & x(x(x(\mathsf{BAB}x(\mathsf{B}x(\mathsf{B}x(\mathsf{BAB}x)))))) \\
\to^3 & x(x(x(\mathsf{A}(\mathsf{B}x)(\mathsf{B}x(\mathsf{B}x(\mathsf{BAB}x)))))) \\
\to^3 & x(x(x(\mathsf{M}(\mathsf{B}x(\mathsf{B}x(\mathsf{B}x(\mathsf{BAB}x))))))) \\
\to & x(x(x(\mathsf{B}x(\mathsf{B}x(\mathsf{B}x(\mathsf{BAB}x)))(\mathsf{B}x(\mathsf{B}x(\mathsf{B}x(\mathsf{BAB}x))))))) \\
\to^3 & x(x(x(x(\mathsf{B}x(\mathsf{B}x(\mathsf{BAB}x))(\mathsf{B}x(\mathsf{B}x(\mathsf{B}x(\mathsf{BAB}x)))))))) \\
\to^3 & x(x(x(x(x(\mathsf{B}x(\mathsf{BAB}x)(\mathsf{B}x(\mathsf{B}x(\mathsf{B}x(\mathsf{BAB}x))))))))) \\
\to^3 & x(x(x(x(x(x(\mathsf{BAB}x(\mathsf{B}x(\mathsf{B}x(\mathsf{B}x(\mathsf{BAB}x)))))))))) \\
\to^3 & x(x(x(x(x(x(\mathsf{A}(\mathsf{B}x)(\mathsf{B}x(\mathsf{B}x(\mathsf{B}x(\mathsf{BAB}x)))))))))) \\
\to^3 & x(x(x(x(x(x(\mathsf{M}(\mathsf{B}x(\mathsf{B}x(\mathsf{B}x(\mathsf{B}x(\mathsf{BAB}x)))))))))))
\end{array}
$$

Definition 15. Let M, N be λ-terms, and n a natural number. We define $MN^{\sim n}$ and $M^n N$ as follows:

$$
\begin{array}{rclcrcl}
MN^{\sim 0} & = & M & \qquad\qquad & M^0 N & = & N \\
MN^{\sim n+1} & = & MNN^{\sim n} & & M^{n+1} N & = & M(M^n N)
\end{array}
$$

A context of the form $\Box N^{\sim n}$ is called a *vector*. For the vector notation, it is to be understood that term formation gets highest priority, i.e., $MNP^{\sim n} = (MN)P^{\sim n}$.

In the sequel we will consider extensions of the set of λ-terms, and of the λ-calculus. It is straightforward to extend notations and terminology correspondingly. For example, we write $\mathit{Ter}^{\infty}(\lambda\bot)$ for the set of infinite λ-terms with a special constant symbol $\bot$, i.e., defined by the 'cogrammar':

$$
M ::=^{\text{co}} x \mid \lambda x.M \mid MM \mid \bot \qquad\qquad (x \in \mathcal{X})
$$

Böhm Trees, Lévy–Longo Trees, and Berarducci Trees form particular subsets of the set $\mathit{Ter}^{\infty}(\lambda\bot)$, where $\bot$ stands for the different notions of 'undefined' in these semantics.

[‡] See (Klo07) for several proofs of the double fixed point theorem, including some of (Bar84; Smu85).

For completeness sake we repeat their classic definitions below, see Definition 20. We define some preliminary notions first.

Definition 16. We define a *metric* d on $\mathit{Ter}^\infty(\lambda\bot)$ by $\mathrm{d}(M, N) = 0$ whenever $M \equiv N$, and $\mathrm{d}(M, N) = 2^{-k}$ otherwise, where $k \in \mathbb{N}$ is the least length of all positions p such that $M(p) \not\equiv N(p)$.

Definition 17. Let R be a reduction relation on $\mathit{Ter}^\infty(\lambda)$. A *transfinite rewrite sequence* (of ordinal length α) is a sequence of rewrite steps $(M_\gamma \to_{R,p_\gamma} M_{\gamma+1})_{\gamma<\alpha}$ such that for every limit ordinal $\kappa < \alpha$ we have that if β approaches κ from below, then

(i) the distance $\mathrm{d}(M_\gamma, M_\kappa)$ tends to 0, and, moreover,
(ii) the depth of the rewrite action, i.e., the length of the positions p_γ, tends to infinity.

The sequence is called *strongly convergent* if α is a successor ordinal, or there exists a term M_α such that the conditions (i) and (ii) are fulfilled for every limit ordinal $\kappa \leq \alpha$. In this case we write $M_0 \mathrel{\to\!\!\!\!\twoheadrightarrow}_R M_\alpha$, or $M_0 \to^\alpha_R M_\alpha$ to explicitly indicate the length α of the sequence. The sequence is called *divergent* if it is not strongly convergent.

Let $M \in \mathit{Ter}^\infty(\lambda)$ be a term. The infinitary properties *strong normalization* SN^∞, *confluence* CR^∞, and *unique normalization* UN^∞ of R are defined as follows:

$\mathrm{SN}^\infty(M)$: all infinite rewrite sequences from M are strongly convergent;
$\mathrm{CR}^\infty(M) : \forall N_1, N_2\ (N_1 \mathrel{\twoheadleftarrow\!\!\!\!\leftarrow}_R M \mathrel{\to\!\!\!\!\twoheadrightarrow}_R N_2 \implies N_1 \mathrel{\to\!\!\!\!\twoheadrightarrow}_R \cdot \mathrel{\twoheadleftarrow\!\!\!\!\leftarrow}_R N_2)$;
$\mathrm{UN}^\infty(M) : \forall N_1, N_2\ (N_1 \mathrel{\twoheadleftarrow\!\!\!\!\leftarrow}_R M \mathrel{\to\!\!\!\!\twoheadrightarrow}_R N_2$ and N_1, N_2 normal forms $\implies N_1 \equiv N_2)$.

We write $\mathrm{SN}^\infty(R)$, $\mathrm{CR}^\infty(R)$ or $\mathrm{UN}^\infty(R)$ if the respective property holds for all terms.

Definition 18.

(i) A *head reduction step* $\to_h$ is a β-reduction step of the form:
$\lambda x_1 \dots x_n.(\lambda y.M)NN_1 \dots N_m \to \lambda x_1 \dots x_n.(M[y := N])N_1 \dots N_m$ with $n, m \geq 0$.
(ii) Accordingly, a *head normal form (hnf)* is a λ-term of the form:
$\lambda x_1. \dots \lambda x_n.yN_1 \dots N_m$ with $n, m \geq 0$ (where y may be one of the x_i $(1 \leq i \leq n)$).
(iii) A *weak head normal form (whnf)* is an hnf or an abstraction, that is, a whnf is a term of the form $xM_1 \dots M_m$ or $\lambda x.M$.
(iv) A term *has a (weak) hnf* if it reduces to one.
(v) We call a term *root-stable* if it does not reduce to a redex: $(\lambda x.M)N$. A term is called *root-active* if it does not reduce to a root-stable term.
(vi) A *term of order* 0 is a term that cannot be β-reduced to an abstraction term. A term M is *mute* (Ber96) if it is a term of order 0 which cannot be reduced to a variable or to an application M_1M_2 with M_1 a term of order 0. Equivalently: M has an infinite reduction with at the root infinitely many times a redex contraction.

Remark 19. We note that if M reduces to a hnf N, the number of head steps in any reduction from M to N is the same. This is the reason why the annotations in the clocked BTs introduced in Section 3 are canonical, and not subject to some reduction strategy.

Definition 20. Let $M \in \mathit{Ter}^\infty(\lambda\bot)$. Then we define the Böhm Tree $\mathsf{BT}(M)$, Lévy–

Longo Tree $\mathsf{LLT}(M)$, and Berarducci Tree $\mathsf{BeT}(M)$ coinductively by

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

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

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

3. Clocked Lambda Calculus

In previous work (EHK10; EHKP12), we introduced clocked Böhm Trees by annotating Böhm Trees. Here we give a first-class status to the clocks, and obtain the clocked BTs as the infinitary normal forms in an extended λ-calculus. We extend the λ-calculus with an explicit unary constructor τ in the spirit of (AJ02); cf. also (Wad81; Par83; NI89) (the latter though have no explicit constructor leading to the annotations as we define below). The idea is that in the normalization to the Böhm Tree, we leave behind an occurrence of τ at a position p to witness the β-step needed to head normalize the subterm at p.

Definition 21. The set $\mathit{Ter}^\infty(\lambda\tau)$ of (finite and infinite) terms of the clocked λ-calculus is coinductively defined by the following grammar

$$M ::=^{\text{co}} x \mid \lambda x.M \mid MM \mid \tau(M) \qquad\qquad (x \in \mathcal{X})$$

The set $\mathit{Con}^\infty(\lambda\tau)$ of infinite contexts is inductively defined by

$$C ::= \Box \mid \lambda x.C \mid CM \mid MC \mid \tau(C) \qquad\qquad (x \in \mathcal{X}, M \in \mathit{Ter}^\infty(\lambda\tau))$$

Next we define a rewrite relation $\to_{\text{⏲}}$ for obtaining clocked Lévy–Longo Trees (LLTs) as its infinitary normal forms. LLTs form a refinement of Böhm Trees, and likewise so for their clocked variants. The reason for focusing on LLTs will become clear in the sequel.

Definition 22. The relation $\to_{\text{⏲}}$ on $\mathit{Ter}^\infty(\lambda\tau)$ of the clocked λ-calculus is defined as the closure under contexts of the rules

$$(\lambda x.M)N \to \tau(M[x := N]) \qquad (\beta\tau)$$

$$\tau(M)N \to \tau(MN) \qquad (\tau\text{-app})$$

The τ symbol can be interpreted as follows: in the normalization of a term to its LLT every subterm $\tau^n(M)$ means that n β-steps were needed to normalize the original subterm to M, its weak head normal form (whnf, see Definition 18). Infinite stacks τ^ω then stand for 'undefined', i.e., the original subterm did not have a whnf.

Example 23. We compute the $\twoheadrightarrow_{\circledast}$-normal form of $\mathsf{Y}_0\mathsf{K}$. First we note that

$$\begin{aligned}\mathsf{Y}_0\mathsf{K} &\equiv (\lambda f.\omega_f\omega_f)\mathsf{K} \rightarrow_{\circledast} \tau(\omega_\mathsf{K}\omega_\mathsf{K})\\ \omega_\mathsf{K} &\equiv \lambda x.\mathsf{K}(xx) \rightarrow_{\circledast} \tau(\lambda xy.xx)\\ \omega_\mathsf{K}\omega_\mathsf{K} &\rightarrow_{\circledast} \tau(\lambda xy.xx)\omega_\mathsf{K} \rightarrow_{\circledast} \tau((\lambda xy.xx)\omega_\mathsf{K}) \rightarrow_{\circledast} \tau(\tau(\lambda y.\omega_\mathsf{K}\omega_\mathsf{K}))\end{aligned}$$

Hence we obtain

$$\mathsf{Y}_0\mathsf{K} \twoheadrightarrow_{\circledast} \tau^3(\lambda y.\tau^2(\lambda y.\tau^2(\lambda y.\tau^2(\lambda y.\tau^2(\ldots)))))$$

which can be recognized as the LLT $\lambda y.\lambda y.\lambda y.\ldots$ enriched with τ's. After the initial application of Y_0, every abstraction λy is produced by precisely two head reduction steps as witnessed by the preceding occurrence of τ^2.

Before we show that the normal forms of $\twoheadrightarrow_{\circledast}$ indeed constitute enriched LLTs, we collect some global infinitary properties of $\twoheadrightarrow_{\circledast}$.

Lemma 24. The relation $\twoheadrightarrow_{\circledast}$ has the properties UN^∞, SN^∞ and CR^∞.

Proof. UN^∞ follows from orthogonality of the rules defining $\rightarrow_{\circledast}$, see (KS09). SN^∞ is equivalent to the non-existence of root-active terms (KdV05). This follows from observing that any contraction of a root redex will introduce a τ at the root, hence every term admits at most one root step. Finally, CR^∞ immediately follows from UN^∞ and SN^∞. □

Definition 25. Let $M \in \mathit{Ter}^\infty(\lambda\tau)$. We define the *clocked Lévy–Longo Tree* $\mathsf{LLT}_{\circledast}(M)$ *of* M as the (unique) infinitary normal form of M with respect $\twoheadrightarrow_{\circledast}$.

Example 26. Consider the fpcs Y_0 of Curry and Y_1 of Turing, defined in Example 9. Figure 1 displays the clocked Lévy–Longo Trees of $\mathsf{Y}_0 f$ (left) and $\mathsf{Y}_1 f$ (right), computed as follows. We have $\mathsf{Y}_0 \equiv \lambda f.\omega_f\omega_f$ where $\omega_f \equiv \lambda x.f(xx)$, and

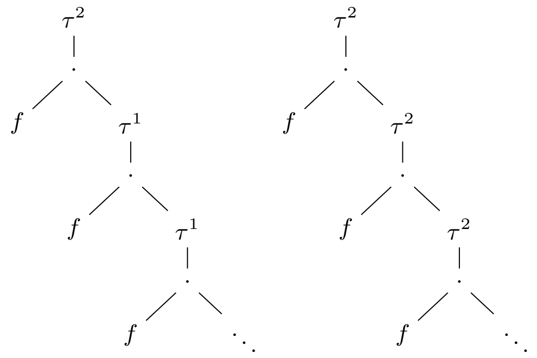


Fig. 1. *Clocked Lévy–Longo Trees of* $\mathsf{Y}_0 f$ *and* $\mathsf{Y}_1 f$.

$$\omega_f\omega_f \rightarrow \tau(f(\omega_f\omega_f))$$

Therefore we obtain

$$\begin{aligned}\mathsf{LLT}_{\clock}(\omega_f\omega_f) &= \tau(f\,\mathsf{LLT}_{\clock}(\omega_f\omega_f))\\ \mathsf{LLT}_{\clock}(\mathsf{Y}_0 f) &= \tau(\mathsf{LLT}_{\clock}(\omega_f\omega_f)) = \tau^2(f\,\mathsf{LLT}_{\clock}(\omega_f\omega_f))\end{aligned}$$

For $\mathsf{Y}_1 \equiv \eta\eta$ where $\eta \equiv \lambda x.\lambda f.f(xxf)$ we get:

$$\mathsf{Y}_1 f \equiv \eta\eta f \to_{\clock} \tau(\lambda f.f(\eta\eta f))f \to_{\clock} \tau((\lambda f.f(\eta\eta f))f) \to_{\clock} \tau(\tau(f(\eta\eta f)))$$

Hence, $\mathsf{LLT}_{\clock}(\mathsf{Y}_1 f) = \tau^2(f\,\mathsf{LLT}_{\clock}(\mathsf{Y}_1 f))$.

We let $\lfloor\cdot\rfloor : \mathit{Ter}^\infty(\lambda\tau) \to \mathit{Ter}^\infty(\lambda\bot)$ denote the map that replaces every outermost occurrence of a subterm of the form τ^ω by $\bot$ and removes all other occurrences of τ.

Lemma 27. Let $M \in \mathit{Ter}(\lambda)$. Then $\lfloor\mathsf{LLT}_{\clock}(M)\rfloor$ is the Lévy–Longo Tree $\mathsf{LLT}(M)$ of M.

Proof. By coinduction. We do case distinction on the weak head normal form of M.

(i)M has no whnf. Then $\lfloor\mathsf{LLT}_{\clock}(M)\rfloor = \lfloor\tau^\omega\rfloor = \bot = \mathsf{LLT}(M)$.

(ii)M converges to whnf $xM_1\cdots M_n$ after n steps of weak head reduction. By coinduction, $\lfloor\mathsf{LLT}_{\clock}(M_i)\rfloor = \mathsf{LLT}(M_i)$. Then

$$\begin{aligned}\lfloor\mathsf{LLT}_{\clock}(M)\rfloor = \lfloor\tau^n(xM_1\cdots M_n)\rfloor &= x\lfloor M_1\rfloor\cdots\lfloor M_n\rfloor\\ &= x\mathsf{LLT}(M_1)\cdots\mathsf{LLT}(M_n) = \mathsf{LLT}(M)\end{aligned}$$

(iii)M converges to whnf $\lambda x.N$ in n steps. By coinduction, $\lfloor\mathsf{LLT}_{\clock}(N)\rfloor = \mathsf{LLT}(N)$. Then

$$\begin{aligned}\lfloor\mathsf{LLT}_{\clock}(M)\rfloor &= \lfloor\tau^n(\lambda x.\mathsf{LLT}_{\clock}(N))\rfloor = \lambda x.\lfloor\mathsf{LLT}_{\clock}(N)\rfloor\\ &= \lambda x.\mathsf{LLT}(N) = \mathsf{LLT}(M)\end{aligned}$$

□

Remark 28. Let $\to_{\mathsf{BT}}$ be the extension of the relation $\to_{\clock}$ from Definition 22 by taking the closure under contexts of the rules $(\beta\tau)$, (τ-app) and

$$\lambda x.\tau(M) \to \tau(\lambda x.M) \qquad (\tau\lambda)$$

Then for every $M \in \mathit{Ter}^\infty(\lambda\tau)$, the infinitary normal form of M with respect to $\to_{\mathsf{BT}}$ is the clocked Böhm Tree of M. However, the rules are no longer orthogonal and infinitary confluence of $\to_{\mathsf{BT}}$ is just a syntactic accident. This becomes visible in Remark 46 where confluence is lost when τ's are annotated with positions.

A clocked version of Berarducci Trees can be obtained as the infinitary normal forms of the contextual closure of the rules

$$\begin{aligned}(\lambda x.M)N &\to \tau(M[x:=N])\\ \tau^n(\lambda x.M)N &\to \tau^n((\lambda x.M)N) \qquad (n \in \mathbb{N})\end{aligned}$$

Note that this system has infinitely many rules.

Remark 29. We make the connection with the notations used in (EHK10; EHKP12). There we had annotated terms $[k]M$, and a constant symbol $\bot$. In the framework we introduce here, these correspond to terms $\tau^k(M)$ and τ^ω, respectively.

We now extend the notion of position as introduced in Definition 5 to $\mathit{Ter}^{\infty}(\lambda\tau)$.

Definition 30. A *position* is a sequence over $\{\lambda, \mathrm{L}, \mathrm{R}, \tau\}$. Let $M \in \mathit{Ter}^{\infty}(\lambda\tau)$ and $p \in \{\lambda, \mathrm{L}, \mathrm{R}, \tau\}^*$. The *subterm* $M|_p$ *of* M *at position* p is defined as follows:

$$\begin{array}{rclcrclcrcl} M|_\epsilon &=& M & \qquad & (MN)|_{\mathrm{L}p} &=& M|_p & \qquad & \tau(M)|_{\tau p} &=& M|_p \\ (\lambda x.M)|_{\lambda p} &=& M|_p & & (MN)|_{\mathrm{R}p} &=& N|_p & & & & \end{array}$$

We let $\mathit{Pos}(M) \subseteq \{\lambda, \mathrm{L}, \mathrm{R}, \tau\}^*$ denote the set of positions p such that $M|_p$ is defined.

We now define relations $\succeq_{\text{🕒}}$ and $=_{\text{🕒}\exists}$ on λ-terms via their clocked Lévy–Longo Trees.

Definition 31. We define $\to_\tau \subseteq \mathit{Ter}^{\infty}(\lambda\tau)^2$ as the closure under contexts of the rule

$$\tau(M) \to M$$

and use $=_\tau$ to denote the equivalence closure of $\to_\tau$. For $M, N \in \mathit{Ter}^{\infty}(\lambda\tau)$, we define

(i) $M \succeq_{\text{🕒}} N$, M *is globally improved by* N iff $\mathsf{LLT}_{\text{🕒}}(M) \twoheadrightarrow\!\!\!\twoheadrightarrow_\tau \mathsf{LLT}_{\text{🕒}}(N)$;
(ii) $M =_{\text{🕒}\exists} N$, M *eventually matches* N iff $\mathsf{LLT}_{\text{🕒}}(M) =_\tau \mathsf{LLT}_{\text{🕒}}(N)$.

For example, as can be deduced from the clocked LLTs of $\mathsf{Y}_0 f$ and $\mathsf{Y}_1 f$ in Figure 1, we have that $\mathsf{Y}_0 f$ globally improves $\mathsf{Y}_1 f$, in symbols $\mathsf{Y}_0 f \preceq_{\text{🕒}} \mathsf{Y}_1 f$.

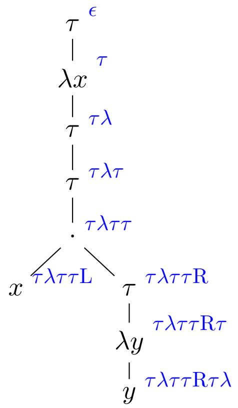
Definition 32. A position $p' \in \{\lambda, \mathrm{L}, \mathrm{R}, \tau\}^*$ is a τ*-extension* of $p \in \{\lambda, \mathrm{L}, \mathrm{R}\}^*$ if p is obtained from p' by dropping all occurrences of τ. Furthermore, let $M \in \mathit{Ter}^{\infty}(\lambda\tau)$ and $p \in \mathit{Pos}(\lfloor M \rfloor)$. Then we define $\sharp_\tau(M, p)$ as follows:

$$\begin{array}{rcll} \sharp_\tau(\tau^n(M), \epsilon) &=& n & \qquad\qquad \text{if } M(\epsilon) \neq \tau \\ \sharp_\tau(\tau(M), p) &=& \sharp_\tau(M, p) & \qquad\qquad \text{if } p \neq \epsilon \\ \sharp_\tau(\lambda x.M, \lambda p) &=& \sharp_\tau(M, p) & \\ \sharp_\tau(MN, \mathrm{L}p) &=& \sharp_\tau(M, p) & \\ \sharp_\tau(MN, \mathrm{R}p) &=& \sharp_\tau(N, p) & \end{array}$$

In other words, $\sharp_\tau(M, p)$ denotes the maximal $n \in \mathbb{N}$ such that $p'' = p'\tau^n \in \mathit{Pos}(M)$ and p'' is a τ-extension of p. Alternatively, $\sharp_\tau(M, p)$ is the number of τ-extensions p' of p such that $M(p') = \tau$.

Note that for terms $M, N \in \mathit{Ter}^{\infty}(\lambda\tau)$ with $\lfloor M \rfloor = \lfloor N \rfloor$, we have $M =_\tau N$ if and only if $\sharp_\tau(M, p) \neq \sharp_\tau(N, p)$ for at most finitely many positions $p \in \mathit{Pos}(\lfloor M \rfloor)$.

Example 33. Consider the term $M \equiv \tau(\lambda x.\tau(\tau(x\tau(\lambda y.y))))$ and its term tree depicted as follows, where the positions of M are displayed in blue:


τ
ϵ
λx
τ
τ
$\tau\lambda$
τ
$\tau\lambda\tau$
$\cdot$
$\tau\lambda\tau\tau$
x
$\tau\lambda\tau\tau\mathrm{L}$
τ
$\tau\lambda\tau\tau\mathrm{R}$
λy
$\tau\lambda\tau\tau\mathrm{R}\tau$
y
$\tau\lambda\tau\tau\mathrm{R}\tau\lambda$

Then $Pos(\lfloor M \rfloor) = \{\epsilon, \lambda, \lambda\mathrm{L}, \lambda\mathrm{R}, \lambda\mathrm{R}\lambda\}$, and $\sharp_\tau(M, \epsilon) = 1$, $\sharp_\tau(M, \lambda) = 2$, $\sharp_\tau(M, \lambda\mathrm{L}) = 0$, $\sharp_\tau(M, \lambda\mathrm{R}) = 1$, and $\sharp_\tau(M, \lambda\mathrm{R}\lambda) = 0$.

Lemma 34. Let $M \in Ter^\infty(\lambda\tau)$, and $p' \in Pos(M)$ be a τ-extension of $p \in \{\lambda, \mathrm{L}, \mathrm{R}\}^*$. Then $\lfloor M|_{p'} \rfloor = \lfloor M \rfloor|_p$.

We now adapt (EHKP12, Proposition 25) and (EHKP12, Theorem 26).

Proposition 35. Clocks are accelerated under reduction, that is, if $M \twoheadrightarrow N$, then the reduct N improves M globally, that is, $\mathsf{LLT}_{◷}(M) \mathrel{\twoheadrightarrow\!\!\!\!\twoheadrightarrow}_\tau \mathsf{LLT}_{◷}(N)$. Dually, clocks slow down under expansion (the reverse of reduction).

Proposition 35 yields the following method for discriminating λ-terms:

Theorem 36. Let M and N be λ-terms. If N cannot be improved globally by any reduct of M, then $M \neq_\beta N$.

Theorem 36 is often difficult to use as we have to prove something for all reducts of M. Nevertheless, it can be useful, see for example (EHKP12), where we apply the theorem to solve a question of Selinger and Plotkin (Plo07).

Fortunately, for a large class of λ-terms the clocks are invariant under reduction, that is, the clocked Lévy–Longo Trees coincide up to a finite number of τ's (i.e., modulo a finite number of insertion and removal of τ's). In (EHK10) we have shown that the clocks are invariant for 'simple' terms. For the application to Lévy–Longo Trees, here we adapt the definition from (EHK10) to weak head normal forms.

Definition 37 (Simple terms). A redex $(\lambda x.M)N$ is called:

(i) *linear* if x has at most one occurrence in M;
(ii) *call-by-value* if N is a normal form; and
(iii) *simple* if it is linear or call-by-value.

A λ-term M is *simple* if (a) it has no whnf, or the head reduction to whnf contracts only simple redexes and is of one of the following forms: (b) $M \twoheadrightarrow_h \lambda x.M'$ with M' a simple term, or (c) $M \twoheadrightarrow_h yM_1 \ldots M_m$ with $M_1, \ldots, M_m$ simple terms.

Proposition 38. Let N be a reduct of a simple term M. Then N eventually matches M (i.e., $\mathsf{LLT}_{\text{⏲}}(M) =_\tau \mathsf{LLT}_{\text{⏲}}(N)$).

The following is a reformulation of (EHK10, Corollary 32) for Lévy–Longo Trees:

Corollary 39. If simple terms M, N do not eventually match ($\mathsf{LLT}_{\text{⏲}}(M) \neq_\tau \mathsf{LLT}_{\text{⏲}}(N)$), then they are not β-convertible, that is, $M \neq_\beta N$.

Even if a term M is not simple, it frequently is possible to *simplify* M, that is, to reduce M to a simple term. This helps for distinguishing λ-terms M and N, since we can always consider β-equivalent terms $M' =_\beta M$ and $N' =_\beta N$ instead. However, there are also non-simplifiable fpcs, as given in the following example.

Example 40. Let $Y \equiv \lambda f.\alpha_f \alpha_f \mathsf{I}$ with $\alpha_f \equiv \lambda xy.yf(xx(yy))$. We then have

$$\begin{aligned} Y &\to \lambda f.(\lambda y.yf(\alpha_f\alpha_f(yy)))\mathsf{I} \to \lambda f.\mathsf{I}f(\alpha_f\alpha_f(\mathsf{II})) \to \lambda f.f(\alpha_f\alpha_f(\mathsf{II})) \\ &\to^4 \lambda f.f(f(\alpha_f\alpha_f(\mathsf{II}(\mathsf{II})))) \to^6 \lambda f.f(f(\alpha_f\alpha_f(\mathsf{II}(\mathsf{II})(\mathsf{II}(\mathsf{II}))))) \to^{10} \cdots \end{aligned}$$

It is not difficult to see that this fixed point combinator Y cannot be simplified.

4. Atomic Clocked Lambda Calculus

We generalize the method introduced in the previous section by not only recording whether head reduction steps have taken place, but also where they took place.

Definition 41. The set $\mathit{Ter}^\infty(\lambda\tau_\star)$ of (finite and infinite) terms of the atomic clocked λ-calculus is coinductively defined by the following grammar

$$M ::=^{\text{co}} x \mid \lambda x.M \mid MM \mid \tau_p(M) \qquad\qquad (x \in \mathcal{X},\ p \in \{\mathrm{L}\}^*)$$

The set $\mathit{Con}^\infty(\lambda\tau_p)$ of infinite contexts is inductively defined by

$$C ::= \Box \mid \lambda x.C \mid CM \mid MC \mid \tau_p(C) \qquad (x \in \mathcal{X}, M \in \mathit{Ter}^\infty(\lambda\tau),\ p \in \{\mathrm{L}\}^*)$$

We keep using the set $\{\lambda, \mathrm{L}, \mathrm{R}, \tau\}^*$ for the positions, ignoring the positions in the subscripts of τ. Accordingly, the notion of τ-extension remains unchanged.

Definition 42. We define the rewrite relation $\to_{\text{☢}}$ on $\mathit{Ter}^\infty(\lambda\tau_\star)$ of the atomic clocked λ-calculus as the closure under contexts $C \in \mathit{Con}^\infty(\lambda\tau_p)$ of the following rules:

$$\begin{aligned} (\lambda x.M)N &\to \tau_\epsilon(M[x := N]) \\ \tau_p(M)N &\to \tau_{\mathrm{L}p}(MN) \end{aligned}$$

We overload the notation $\to_\tau$ and also use it for the rewrite relation that removes symbols τ_p. Moreover, we reuse the terminology from Section 3.

Definition 43. We define $\to_\tau \subseteq \mathit{Ter}^\infty(\lambda\tau_\star)^2$ as the closure under contexts of the rule

$$\tau_p(M) \to M$$

and use $=_\tau$ to denote the equivalence closure of $\to_\tau$. We define

(i) $M \succeq_{\text{☢}} N$, *M is globally improved by N* iff $\mathsf{LLT}_{\text{☢}}(M) \twoheadrightarrow\!\!\!\!\to_\tau \mathsf{LLT}_{\text{☢}}(N)$;

(ii) $M =_{\text{☢}\exists} N$, *M eventually matches N* iff $\mathsf{LLT}_{\text{☢}}(M) =_{\tau} \mathsf{LLT}_{\text{☢}}(N)$.

Definition 44. Let $M \in \mathit{Ter}^{\infty}(\lambda\tau_{\star})$ and $p \in \mathit{Pos}(\lfloor M \rfloor)$. We define $\sharp_{\tau}(M, p)$ as follows:

$$\begin{aligned}
\sharp_{\tau}(\tau_{q_1}(\dots\tau_{q_n}(M)\dots), \epsilon) &= \langle q_1, \dots, q_n \rangle && \text{if for all } q \in \{\mathrm{L}\}^* \text{ we have } M(\epsilon) \neq \tau_q \\
\sharp_{\tau}(\tau_q(M), p) &= \sharp_{\tau}(M, p) && \text{if } p \neq \epsilon \\
\sharp_{\tau}(\lambda x.M, \lambda p) &= \sharp_{\tau}(M, p) \\
\sharp_{\tau}(MN, \mathrm{L}p) &= \sharp_{\tau}(M, p) \\
\sharp_{\tau}(MN, \mathrm{R}p) &= \sharp_{\tau}(N, p)
\end{aligned}$$

It is straightforward to adapt Proposition 35, Theorem 36, Proposition 38 and Corollary 39 from the previous section to the refined setting of atomic clocks.

Atomic clocks do improve discrimination power, as can be seen in the following example.

Example 45. In (EHKP12, Examples 35, 36) we computed the (non-atomic) clocked BTs of the fpcs $\mathsf{Y}_n \equiv \mathsf{Y}_0\delta^{\sim n}$ with $\delta \equiv \lambda ab.b(ab)$ from the Böhm sequence and the fpcs $\mathsf{U}_n \equiv \mathsf{BY}_0\mathsf{S}^{\sim n}\mathsf{I}$ of the Scott sequence. This showed that both sequences do not contain any duplicates. In the framework of Section 3, for $n \geq 2$ they are rendered as $\mathsf{LLT}_{\text{⏲}}(\mathsf{Y}_n) = \tau^{2n}(x\,\mathsf{LLT}_{\text{⏲}}(\mathsf{Y}_n x))$, and $\mathsf{LLT}_{\text{⏲}}(\mathsf{U}_n x) = \tau^{3n-2}(x\,\mathsf{LLT}_{\text{⏲}}(\mathsf{U}_n x))$. From these clocks it follows that $\mathsf{Y}_n \neq_{\beta} \mathsf{U}_n$ for all $n > 2$. We now discriminate Y_2 from U_2 by their atomic clocked LLTs, computed as follows. We first reduce both terms to simple terms:

$$\begin{aligned}
\mathsf{Y}_2 x &\equiv \mathsf{Y}_0\delta\delta x \twoheadrightarrow \eta\eta\delta x && \text{where } \eta \equiv \lambda ab.b(aab) \\
\mathsf{U}_2 x &\equiv \mathsf{BY}_0\mathsf{SSI}x \twoheadrightarrow \theta\theta\mathsf{I}x && \text{where } \theta \equiv \lambda abc.bc(aabc)
\end{aligned}$$

Then we compute the atomic clocked LLTs of these simple reducts, as follows:

$$\begin{aligned}
\eta\eta\delta x &\to_{\text{☢}} \tau_{\epsilon}((\lambda b.b(\eta\eta b)))\delta x \\
&\to_{\text{☢}} \tau_{\mathrm{L}}((\lambda b.b(\eta\eta b))\delta)x \\
&\to_{\text{☢}} \tau_{\mathrm{L}}(\tau_{\epsilon}(\delta(\eta\eta\delta)))x \\
&\to_{\text{☢}} \tau_{\mathrm{L}}(\tau_{\epsilon}(\tau_{\epsilon}(\lambda b.b(\eta\eta\delta b))))x \\
&\to_{\text{☢}}^{3} \tau_{\mathrm{LL}}(\tau_{\mathrm{L}}(\tau_{\mathrm{L}}(\lambda b.b(\eta\eta\delta b)x))) \\
&\to_{\text{☢}} \tau_{\mathrm{LL}}(\tau_{\mathrm{L}}(\tau_{\mathrm{L}}(\tau_{\epsilon}(x(\eta\eta\delta x)))))
\end{aligned}
\qquad
\begin{aligned}
\theta\theta\mathsf{I}x &\to_{\text{☢}} \tau_{\epsilon}(\lambda bc.bc(\theta\theta bc))\mathsf{I}x \\
&\to_{\text{☢}} \tau_{\mathrm{L}}((\lambda bc.bc(\theta\theta bc))\mathsf{I})x \\
&\to_{\text{☢}} \tau_{\mathrm{L}}(\tau_{\epsilon}(\lambda c.\mathsf{I}c(\theta\theta\mathsf{I}c)))x \\
&\to_{\text{☢}}^{2} \tau_{\mathrm{LL}}(\tau_{\mathrm{L}}((\lambda c.\mathsf{I}c(\theta\theta\mathsf{I}c))x)) \\
&\to_{\text{☢}} \tau_{\mathrm{LL}}(\tau_{\mathrm{L}}(\tau_{\epsilon}(\mathsf{I}x(\theta\theta\mathsf{I}x)))) \\
&\to_{\text{☢}} \tau_{\mathrm{LL}}(\tau_{\mathrm{L}}(\tau_{\epsilon}(\tau_{\epsilon}(x)(\theta\theta\mathsf{I}x)))) \\
&\to_{\text{☢}} \tau_{\mathrm{LL}}(\tau_{\mathrm{L}}(\tau_{\epsilon}(\tau_{\mathrm{L}}(x(\theta\theta\mathsf{I}x)))))
\end{aligned}$$

Thus the atomic clocked LLTs of these terms can be expressed by the equations:

$$\begin{aligned}
\mathsf{LLT}_{\text{⏲}}(\eta\eta\delta x) &= T_1 && \text{where} && T_1 = \tau_{\mathrm{LL}}(\tau_{\mathrm{L}}(\tau_{\mathrm{L}}(\tau_{\epsilon}(x\,T_1)))) \\
\mathsf{LLT}_{\text{⏲}}(\theta\theta\mathsf{I}x) &= T_2 && \text{where} && T_2 = \tau_{\mathrm{LL}}(\tau_{\mathrm{L}}(\tau_{\epsilon}(\tau_{\mathrm{L}}(x\,T_2))))
\end{aligned}$$

Note that their atomic clocks are distinct indeed, while both terms have the same (non-atomic) clocked LLT $T \equiv \tau^4(xT)$. Hence the method from the previous section is not applicable. However, the atomic clocks do allow us to discriminate the terms. Hence $\mathsf{Y}_2 \neq_{\beta} \mathsf{U}_2$ (by Corollary 39 which generalizes to the setting of atomic BT's).

Remark 46. If, instead of Lévy–Longo Trees, we want a calculus for obtaining Böhm

Trees, we have to let the τs move over the abstractions (part of the hnfs that Böhm Trees are built from), that is, we then add the following rule to the system of Definition 42:

$$\lambda x.\tau_p(M) \to \tau_{\lambda p}(\lambda x.M)$$

However, we find that the critical pair arising from $M \equiv (\lambda x.\tau_p(P))Q$ is not joinable:

$$\tau_\epsilon(\tau_p(P[x := Q])) \leftarrow M \to \tau_{\lambda p}(\lambda x.P)Q \to \tau_{\mathrm{L}\lambda p}((\lambda x.P)Q) \to \tau_{\mathrm{L}\lambda p}(\tau_\epsilon(P[x := Q]))$$

Confluence can be restored by imposing the 'head-first' strategy as defined in the next section.

5. Localized Clocks

In this section, we increase the power of our discrimination method. We extend the class of simple terms in two directions. First, we allow redex duplication, but require that of each redex only finitely many residuals are contracted. Second, we localize the method to a set of positions in the Lévy–Longo Tree; we then only require that the head reductions at these positions do not contract infinitely many residuals of a single redex. To keep the presentation simple, we present this section using the non-atomic clocked λ-calculus. We emphasize that everything in this section generalizes to the atomic clocked λ-calculus.

We define a 'head-first, then arguments' evaluation strategy for $\to_{\text{🕓}}$:

Definition 47. A redex occurrence at the root of R in a term $C[R\,M_1\,M_2 \ldots M_n]$ is said to *precede* all other redex occurrences in R and all redex occurrences in $M_1, \ldots, M_n$.

A *head-first redex* is a redex occurrence that is not preceded by another redex occurrence. A rewrite sequence adheres to the *head-first strategy* if it only contracts head-first redexes. The *top-down strategy* contracts of all head-first redexes at a minimal depth, the leftmost one.

In other words, the head-first strategy forbids the contraction of a redex at position p if there is a redex at a position $q \sqsubset p$ or at a position $q\mathrm{L}^n$ with $q\mathrm{R} \sqsubset p$ and $n \geq 1$. Note that the top-down strategy is deterministic. Correspondingly, for terms M, we refer to the unique top-down reduction starting from M as *the top-down reduction for M*.

We briefly introduce a tracing residuals via underlining (BKdV00; Ter03). To keep the presentation simple, we only trace redexes in a term M to their residuals in $\mathsf{LLT}_{\text{🕓}}(M)$; this suffices for our purposes.

Definition 48. We define the set $\mathit{Ter}^\infty(\underline{\lambda\tau})$ by the following grammar:

$$M ::=^{\mathrm{co}} x \mid \lambda x.M \mid \underline{\lambda} x.M \mid MM \mid \tau(M) \mid \underline{\tau}(M) \qquad\qquad (x \in \mathcal{X})$$

For positions, we ignore the underlining and keep using $\{\lambda, \mathrm{L}, \mathrm{R}, \tau\}^*$. Let $\to_{\underline{\text{🕓}}}$ be the closure under contexts $\mathit{Con}^\infty(\underline{\lambda\tau})$ of the rules $(\beta\tau)$, (τ-app) and

$$(\underline{\lambda} x.M)N \to \underline{\tau}(M[x := N]) \qquad (\underline{\beta\tau})$$

$$\underline{\tau}(x)y \to \underline{\tau}(xy) \qquad (\underline{\tau}\text{-app})$$

We use $\mathsf{LLT}_{\underline{\text{🕓}}}(M)$ to denote the infinitary normal form of M with respect to $\twoheadrightarrow\!\!\!\!\to_{\underline{\text{🕓}}}$.

Let $M \in \mathit{Ter}^{\infty}(\lambda\tau)$ and $p \in \mathit{Pos}(M)$ the position of a redex χ in M. We define $\underline{M}$ as the term obtained from M by underlining the symbol λ at position $p\mathrm{L}$. The underlined occurrences of τ in $\mathsf{LLT}_{\underline{\text{⏲}}}(\underline{M})$ are called the *witnesses* of χ. Let $q \in \mathit{Pos}(\mathsf{LLT}(M))$ be a position in the (τ-free) Lévy–Longo Tree of M. We say that χ *contributes to* q if there is a witness of χ at some τ-extension q' of q.

Example 49. Consider the term $M \equiv \mathsf{S}xy(\mathsf{I}\mathsf{I})$ and the redex $\chi \equiv \mathsf{I}\mathsf{I}$ at position R. Let $\underline{M} \equiv \mathsf{S}xyZ$, where $Z \equiv (\underline{\lambda}x.x)\mathsf{I}$. Then we have

$$\begin{aligned}
\underline{M} &\to_{\underline{\text{⏲}}} \tau(\lambda yz.xz(yz))yZ \\
&\to_{\underline{\text{⏲}}} \tau((\lambda yz.xz(yz))y)Z \\
&\to_{\underline{\text{⏲}}} \tau(\tau(\lambda z.xz(yz)))Z \\
&\to_{\underline{\text{⏲}}} \tau(\tau(\lambda z.xz(yz))Z) \\
&\to_{\underline{\text{⏲}}} \tau(\tau((\lambda z.xz(yz))Z)) \\
&\to_{\underline{\text{⏲}}} \tau(\tau(\tau(xZ(yZ)))) \\
&\to_{\underline{\text{⏲}}} \tau(\tau(\tau(x\underline{\tau}(\mathsf{I})(yZ)))) \\
&\to_{\underline{\text{⏲}}} \tau(\tau(\tau(x\underline{\tau}(\mathsf{I})(y\underline{\tau}(\mathsf{I})))))
\end{aligned}$$

Now observe that the witnesses of χ are at positions $\tau\tau\tau\mathrm{LR}$ and $\tau\tau\tau\mathrm{RR}$, and hence χ contributes to the positions LR and RR of the Lévy–Longo Tree of M.

Definition 50. For a Lévy–Longo Tree $T \in \mathit{Ter}^{\infty}(\lambda\bot)$, we write $\mathcal{P}\mathit{os}^{\star}(T)$ for the set of positions that are neither $\bot$, nor the left child of an application. (In other words, the elements of $\mathcal{P}\mathit{os}^{\star}(T)$ are precisely the positions of maximal weak head normal forms.)

Next, we vastly extend the discrimination methods for simple terms (Proposition 38 and Corollary 39). First, we fine-tune the notion of 'invariance under reduction' by considering sets of positions $P \subseteq \mathcal{P}\mathit{os}^{\star}(\mathsf{LLT}(M))$. Second, we allow the contraction of non-simple redexes if only finitely many descendants of the copied redex are contracted.

For the purpose of refining the comparison of clocks to positions $P \subseteq \mathcal{P}\mathit{os}^{\star}(\mathsf{LLT}(M)))$, we define a function reset_P that 'resets' the clocks for all positions *not* belonging to P.

Definition 51. Let $P \subseteq \{\lambda, \mathrm{L}, \mathrm{R}\}^*$. We define $\mathit{reset}_P(\cdot) : \mathit{Ter}^{\infty}(\lambda\tau) \to \mathit{Ter}^{\infty}(\lambda\tau)$ as follows. For $T \in \mathit{Ter}^{\infty}(\lambda\tau)$ we let $\mathit{reset}_P(T) = \mathit{reset}^{\epsilon}_P(T)$ where:

$$\begin{aligned}
\mathit{reset}^{p}_P(\lambda x.T) &= \lambda x.\mathit{reset}^{p\lambda}_P(T) \\
\mathit{reset}^{p}_P(T_1 T_2) &= \mathit{reset}^{p\mathrm{L}}_P(T_1)\, \mathit{reset}^{p\mathrm{R}}_P(T_2) \\
\mathit{reset}^{p}_P(\tau(T)) &= \begin{cases} T & \text{if } T = \tau^{\omega} \\ \tau(\mathit{reset}^{p}_P(T)) & \text{if } T \neq \tau^{\omega} \text{ and } p \in P \\ \mathit{reset}^{p}_P(T) & \text{if } T \neq \tau^{\omega} \text{ and } p \notin P \end{cases}
\end{aligned}$$

So the term $\mathit{reset}_P(T)$ is obtained from T by removing all occurrences of τ that are neither (i) at a position p' which is a τ-extension of some $p \in P$, nor (ii) part of an infinite τ-stack.

We now define relations $\succeq^{P}_{\circlearrowleft}$ and $=^{P}_{\circlearrowleft\exists}$, for comparing the clocks at positions $P \subseteq \{\lambda, \mathrm{L}, \mathrm{R}\}^*$. These can be viewed as 'localized' versions of $\succeq_{\circlearrowleft}$ and $=_{\circlearrowleft\exists}$ (see Definition 31).

Definition 52. For $M, N \in \mathit{Ter}^{\infty}(\lambda\tau)$ and $P \subseteq \mathcal{P}os^{\star}(\mathsf{LLT}(M))$, we define:

(i) $M \succeq^{P}_{\circlearrowleft} N$, *$M$ is globally improved by N on P* if and only if

$$\mathit{reset}_P(\mathsf{LLT}_{\circlearrowleft}(M)) \twoheadrightarrow\!\!\!\!\twoheadrightarrow_{\tau} \mathit{reset}_P(\mathsf{LLT}_{\circlearrowleft}(N))\,,$$

see Figure 2

(ii) $M \succeq^{P}_{\circlearrowleft\exists} N$, *$M$ is eventually improved by N on P* if and only if

$$\mathit{reset}_P(\mathsf{LLT}_{\circlearrowleft}(M)) =_{\tau} \cdot \twoheadrightarrow\!\!\!\!\twoheadrightarrow_{\tau} \mathit{reset}_P(\mathsf{LLT}_{\circlearrowleft}(N))\,;$$

(iii) $M =^{P}_{\circlearrowleft\exists} N$, *$M$ eventually matches N on P* if and only if

$$\mathit{reset}_P(\mathsf{LLT}_{\circlearrowleft}(M)) =_{\tau} \mathit{reset}_P(\mathsf{LLT}_{\circlearrowleft}(N))\,,$$

see Figure 3.

Whenever we suppress P it is to be understood that $P = \mathcal{P}os^{\star}(\mathsf{LLT}(M))$.

These properties can be equivalently formulated as follows:

(i) $M \succeq^{P}_{\circlearrowleft} N$ iff $M =_{\mathsf{LLT}} N$ and $\sharp_{\tau}(\mathsf{LLT}_{\circlearrowleft}(M), p) \geq \sharp_{\tau}(\mathsf{LLT}_{\circlearrowleft}(N), p)$ for all $p \in P$;

(ii) $M \succeq^{P}_{\circlearrowleft\exists} N$ iff $M =_{\mathsf{LLT}} N$ and $\sharp_{\tau}(\mathsf{LLT}_{\circlearrowleft}(M), p) \geq \sharp_{\tau}(\mathsf{LLT}_{\circlearrowleft}(N), p)$ for almost all $p \in P$.

(iii) $M =^{P}_{\circlearrowleft\exists} N$ iff $M =_{\mathsf{LLT}} N$ and $\sharp_{\tau}(\mathsf{LLT}_{\circlearrowleft}(M), p) = \sharp_{\tau}(\mathsf{LLT}_{\circlearrowleft}(N), p)$ for almost all $p \in P$.

where we write $M =_{\mathsf{LLT}} N$ as a shorthand for $\mathsf{LLT}(M) \equiv \mathsf{LLT}(N)$.

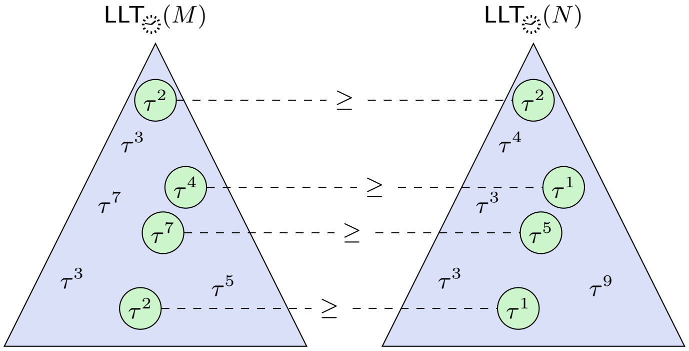


Fig. 2. *M is globally improved by N on P; the positions corresponding to $P \subseteq \mathcal{P}os^{\star}(\mathsf{LLT}(M))$ are encircled.*

The following is a straightforward generalization of Proposition 35.

Proposition 53. Clocks are accelerated under reduction, that is, if $M \twoheadrightarrow N$, then the reduct N globally improves M on P. Dually, clocks slow down under expansion (the reverse of reduction).

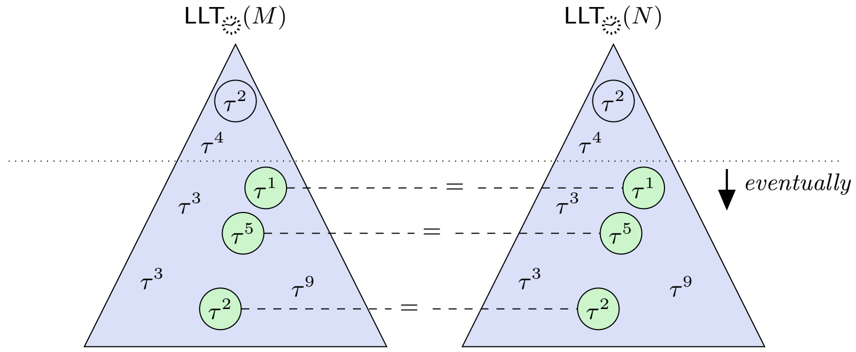


Fig. 3. *M eventually matches N on P; the positions corresponding to $P \subseteq \mathcal{P}os^\star(\mathsf{LLT}(M))$ are encircled.*

We generalize the notion of simple terms to 'P-safe' terms as follows; see Definition 47 for the notion of top-down reduction. In the following definition, by 'P is prefix-closed' we refer to the closure with respect to the superset $\mathcal{P}os^\star(\mathsf{LLT}(M))$, i.e., whenever $p \in P$, $p' \in \mathcal{P}os^\star(\mathsf{LLT}(M))$ and $p' \sqsubseteq p$, then $p' \in P$.

Definition 54. Let $M \in \mathit{Ter}^\infty(\lambda)$ and $P \subseteq \mathcal{P}os^\star(\mathsf{LLT}(M))$. Then we say M is:

(i) *P-bounded* if no term in the top-down reduction $\twoheadrightarrow\!\!\!\!\twoheadrightarrow_{\circlearrowright}$ of M to normal form contains a redex contributing to infinitely many $p \in P$;
(ii) *P-safe* if every $\twoheadrightarrow_\beta$ reduct of M is P-bounded;
(iii) *strongly P-safe* if P is prefix-closed and M is P-bounded.

In order to understand the notion of P-bounded, as defined in item (i) of Definition 54, one can think of it as follows: Suppose that, in the reduction to the infinite normal form, we give every created redex a unique name (and let the residuals carry the same name), and we assign the same name to the τ that is created when the redex is contracted. Then M is P-bounded, if each name occurs only finitely often at τ-extensions of $p \in P$.

We use the property 'strongly P-safe' as a simple sufficient criterion for being P-safe. The following lemma justifies the naming:

Lemma 55. Let M be a λ-term and $P \subseteq \mathcal{P}os^\star(\mathsf{LLT}(M))$. If M is strongly P-safe then M is P-safe.

Proof. Let M be strongly P-safe, that is, P is prefix-closed and M is P-bounded. We use $\multimap\!\!\!\rightarrow$ to denote a complete development of a set of redexes (Ter03); note that $\to\; \subseteq\; \multimap\!\!\!\rightarrow$. It suffices to show that the property of being strongly P-safe is preserved under single steps: $\gamma : M \to_{\beta\tau} N$. To this end, let $\sigma_M : M \to_{\circlearrowright}^{\leq\omega} \mathsf{LLT}_{\circlearrowright}(M)$ be the top-down reduction of M to clocked Lévy–Longo Tree normal form. Let σ_N be the projection σ_M over γ, that is, $\sigma_N = \sigma_M/\gamma$ (Ter03). Then σ_N is the top-down reduction of N to clocked Lévy–Longo Tree normal form: $\sigma_N : N \to_{\circlearrowright}^{\leq\omega} \mathsf{LLT}_{\circlearrowright}(N) \equiv \mathsf{LLT}_{\circlearrowright}(M)$. As a consequence of $\sigma_N = \sigma_M/\gamma$ and the fact that no redex contracted in σ_M can ever get nested inside another redex, we have that $(*)$ the steps of σ_N form a subsequence of σ_M.

For a contradiction, we assume that a term N' in the reduction σ_N contains a redex R

such that there is an infinite set S_N of steps $\to_{\text{⏲}}$ in σ_N that contract a residual of R and contribute to a position $p \in P$. Thus we have a prefix σ'_N of σ_N with $\sigma'_N : N \to_{\text{⏲}}^{*} N'$, and a corresponding prefix σ'_M of σ_M such that $\sigma'_M : M \to_{\text{⏲}}^{*} M'$ with $\gamma/\sigma'_M : M' \mathrel{-\!\!\!\circ\!\!\!\to} N'$ contracting the residuals of γ. By $(*)$ we can find every step of S_N back in σ_M; thus S_N in σ_N traces back to set of steps S_M in σ_M. It follows that R is not a residual of a redex in M', thus is created by γ/σ'_M, for otherwise M was not strongly P-safe. We trace every step of S_M back along σ_M to the point of its creation. As M is strongly P-safe (and by the pigeonhole principle), these steps trace back to an infinite number of distinct redex creations ζ. Note that redexes that contribute to a position p can only be created by contraction of redexes that contribute to a position $p' \sqsubseteq p$. Thus the redex creations in ζ are part of steps $\to_{\text{⏲}}$ contributing to $p' \in P$ as a consequence of S_M belonging to steps $\to_{\text{⏲}}$ that contribute to $p \in P$, and P being prefix-closed. Since $\gamma/\sigma'_M : M' \mathrel{-\!\!\!\circ\!\!\!\to} N'$ contracts only residuals of γ and creates R, it follows that every redex creation in ζ is due to a residual of the step γ. However, the contraction of an infinite number of residuals of γ in steps $\to_{\text{⏲}}$ contributing to positions $p \in P$ contradicts M being strongly P-safe. □

Example 56. We consider the λ-term $M = NN$ with $N = \lambda x.((\lambda y.a(ya(xx)))\mathsf{I})$. Then

$$\sigma'_M : M = NN \to_h (\lambda y.a(ya(NN)))\mathsf{I} \to_h a(\mathsf{I}a(NN)) \quad \text{and} \quad \varphi : \mathsf{I}a(NN) \to_h a(NN)$$

and $\mathsf{LLT}_{\text{⏲}}(M) = \tau^2(a(\tau^1(a(\mathsf{LLT}_{\text{⏲}}(M)))))$. Let $P = \{\, 2(22)^n \mid n \in \mathbb{N} \,\}$, that is, the positions $p \in \mathit{Pos}(\mathsf{LLT}(M))$ of the subterms with clock 1, that is, $\sharp_\tau(\mathsf{LLT}_{\text{⏲}}(M), p) = 1$. Note that the only redexes that contribute to positions $p \in P$ are the $\mathsf{I}a$-redex that are always created by the immediately preceding step. Thus M is P-bounded, but it admits a reduct that is not P-bounded: $M \twoheadrightarrow M' = N'N'$ where $N' = \lambda x.a(\mathsf{I}a(xx))$; here

$$\sigma'_M : M' = N'N' \to_h a(\mathsf{I}a(NN)) \qquad \text{and} \qquad \varphi : \mathsf{I}a(N'N') \to_h a(N'N')$$

and $\mathsf{LLT}_{\text{⏲}}(M') = \tau^1(a(\tau^1(a(\mathsf{LLT}_{\text{⏲}}(M')))))$. Now the steps $\to_{\text{⏲}}$ repeatedly contract redexes $\mathsf{I}a$ that are, except for the first, residuals of the redex $\mathsf{I}a$ in the second N' in $N'N' = M'$.

This illustrates that the property 'P-bounded' is not preserved under reduction, and thus does not imply P-safety. Moreover, it demonstrates that the condition of P being prefix-closed is crucial in the definition of 'strongly P-safe'.

The property 'strongly P-safe' (and thus 'P-safe') is a generalization of simple terms.

Lemma 57. Let M be a simple λ-term. Then M is strongly P-safe for every prefix-closed $P \subseteq \mathcal{P}os^\star(\mathsf{LLT}(M))$.

Proof. Follows immediately from the fact that simple terms do not duplicate redexes throughout the top-down reduction to clocked Lévy–Longo Tree normal form. □

For P-safe terms, the clock on positions P is invariant under reduction:

Lemma 58. Let M be a λ-term, $P \subseteq \mathcal{P}os^\star(\mathsf{LLT}(M))$ such that M is P-safe. If $M \twoheadrightarrow_\beta N$ then M eventually matches N on P, that is, $\mathit{reset}_P(\mathsf{LLT}_{\text{⏲}}(M)) =_\tau \mathit{reset}_P(\mathsf{LLT}_{\text{⏲}}(N))$.

Proof. By induction it suffices to consider the case $\gamma : M \to_{\text{⏲}} N$. Consider the top-down rewrite sequences $\sigma : M \to_{\text{⏲}}^{\leq\omega} \mathsf{LLT}_{\text{⏲}}(M)$ and $\sigma' : N \to_{\text{⏲}}^{\leq\omega} \mathsf{LLT}_{\text{⏲}}(N)$. Then σ' is the subsequence of σ where precisely those steps are selected that are not residuals of γ. Since M is P-safe only a finite number of the residuals of γ are part of steps $\to_{\text{⏲}}$ in σ contributing to $p \in P$. Thus a finite number of $=_\tau$ steps suffices to equalize the clocks at all positions $p \in P$. □

Example 59. We continue Example 56 to illustrate that the property P-bounded is not sufficient for Lemma 58. We have $M \twoheadrightarrow M'' = N''N''$ where $N'' = \lambda x.a(a(xx))$, and

$$\sigma'_M : M'' = N''N'' \to_h a(a(M''))$$

Thus $\mathsf{LLT}_{\text{⏲}}(M'') = \tau^1(a(\tau^0(a(\mathsf{BT}_{\text{⏲}}(M')))))$. Now although M is P-bounded and $M \twoheadrightarrow M''$, we do not have $\mathit{reset}_P(\mathsf{LLT}_{\text{⏲}}(M)) =_\tau \mathit{reset}_P(\mathsf{LLT}_{\text{⏲}}(N))$ since from $\mathsf{LLT}_{\text{⏲}}(M)$ to $\mathsf{LLT}_{\text{⏲}}(M'')$ all positions in P have changed from τ^1 to τ^0.

As immediate consequence, we obtain the following discrimination methods:

Proposition 60. Let M and N be λ-terms, $P \subseteq \mathcal{P}os^\star(\mathsf{LLT}(M))$ such that M is P-safe. If M does *not* eventually improve N on P (not $M \preceq^P_{\text{⏲}\exists} N$), then $M \neq_\beta N$.

Theorem 61. Let M and N be P-safe λ-terms where $P \subseteq \mathcal{P}os^\star(\mathsf{LLT}(M))$. If M and N do *not* match eventually on P (not $M =^P_{\text{⏲}\exists} N$), then $M \neq_\beta N$.

We give an example that shows that the extension of the method can handle duplication of redexes.

Example 62. Let $Y \equiv \lambda f.\alpha_f\alpha_f\mathsf{I}(\mathsf{II})$ with $\alpha_f \equiv \lambda xyz.zzf(xxy(yy))$. We then have the following top-down reduction:

$$\begin{aligned}&Yf \to_{\text{⏲}} \tau(T)\\ &\quad T \equiv \alpha_f\alpha_f\mathsf{I}(\mathsf{II}) \to^3_{\text{⏲}} \tau^3(\mathsf{II}(\mathsf{II})fT) \to^4_{\text{⏲}} \tau^4(\mathsf{I}(\mathsf{II})fT) \to^3_{\text{⏲}} \tau^5(\mathsf{II}fT) \to^3_{\text{⏲}} \tau^6(\mathsf{I}fT) \to^2_{\text{⏲}} \tau^7(fT)\end{aligned}$$

Thus $\mathsf{LLT}_{\text{⏲}}(Yf) \equiv \tau^8(f\,\tau^7(f\,\tau^7(f\,\tau^7(f\ \ldots))))$. The term Yx is not simple, and cannot be simplified. Nevertheless, the term is P-safe for $P = \mathcal{P}os^\star(\mathsf{LLT}(Yf)) = \{\mathrm{R}^n \mid n \in \mathbb{N}\}$ since (i) P is prefix-closed and (ii) in the top-down reduction displayed above, there is no redex contributing to an infinite number of positions $p \in P$. For (ii) note that the only redex duplicated in the cyclic part of the reduction of T is II and all residuals of this redex are contracted before the end of the cycle (within the next 12 steps).

Thus we can apply either Proposition 60 or Theorem 61 to conclude that Yf is not β-convertible to $\mathsf{Y}_0 f$ and $\mathsf{Y}_1 f$ (see Figure 1) which are also P-safe by Lemma 57.

The following example illustrates the use of localized clocks.

Example 63. Recall $\mathsf{Y}_1 \equiv \eta\eta$ where $\eta \equiv \lambda xf.f(xxf)$. We consider the terms M and N defined by

$$\begin{aligned}M &\equiv \alpha_M\alpha_M\mathsf{I}\mathsf{Y}_1 &\qquad \alpha_M &\equiv \lambda xyz.ya(xxyz)z\\ N &\equiv \alpha_N\alpha_N\mathsf{I}\mathsf{Y}_1 &\qquad \alpha_N &\equiv \lambda xy.y\lambda z.a(xxyz)z\end{aligned}$$

We have the following head reductions

$$M \to_{h,\mathrm{LL}} \to_{h,\mathrm{L}} \to_{h,\epsilon} \mathsf{I} aM\mathsf{Y}_1 \to_{h,\mathrm{LL}} aM\mathsf{Y}_1$$
$$N \to_{h,\mathrm{LL}} \to_{h,\mathrm{L}} \mathsf{I}(\lambda z.a(\alpha_N\alpha_N\mathsf{I}z)z)\mathsf{Y}_1 \to_{h,\mathrm{L}} (\lambda z.a(\alpha_N\alpha_N\mathsf{I}z)z)\mathsf{Y}_1 \to_{h,\epsilon} aN\mathsf{Y}_1$$

thus $M \to^{*}_{\text{☢}} \tau_{\mathrm{LL}}(\tau_{\mathrm{L}}(\tau_\epsilon(\tau_{\mathrm{LL}}(aM\mathsf{Y}_1))))$ and $N \to^{*}_{\text{☢}} \tau_{\mathrm{LL}}(\tau_{\mathrm{L}}(\tau_{\mathrm{L}}(\tau_\epsilon(aN\mathsf{Y}_1))))$. Note that the non-atomic clocked Lévy–Longo Trees of M and N coincide: $\mathsf{LLT}_{\text{◷}}(M) \equiv \mathsf{LLT}_{\text{◷}}(N) \equiv T$ where $T \equiv \tau^4(a\,T\,\mathsf{LLT}_{\text{◷}}(\mathsf{Y}_1))$.

The terms M and N cannot be simplified as they infinitely often duplicate Y_1, and the redexes in Y_1 contribute to infinitely many positions of $\mathsf{LLT}(M)$ and $\mathsf{LLT}(N)$, respectively. As a consequence, we need to choose a set of positions $P \subseteq \mathsf{LLT}(M)$ to which Y_1 does not contribute: $P = \{\, (\mathrm{LR})^n \mid n \in \mathbb{N}\,\}$. This set is prefix-closed (in $\mathcal{P}os^\star(\mathsf{LLT}(M))$) and in the reductions displayed above no residual of a duplicated redex is contracted. Thus the terms M and N are strongly P-safe and thus P-safe by Lemma 55. We have that

$$\begin{aligned} &\mathit{reset}_P(\mathsf{LLT}_{\text{☢}}(M)) \equiv T_M && T_M \equiv \tau_{\mathrm{LL}}(\tau_{\mathrm{L}}(\tau_\epsilon(\tau_{\mathrm{LL}}(aT_M\mathsf{LLT}(\mathsf{Y}_1))))) \\ &\mathit{reset}_P(\mathsf{LLT}_{\text{☢}}(N)) \equiv T_N && T_N \equiv \tau_{\mathrm{LL}}(\tau_{\mathrm{L}}(\tau_{\mathrm{L}}(\tau_\epsilon(aT_N\mathsf{LLT}_{\text{☢}}(\mathsf{Y}_1))))) \end{aligned}$$

Hence M and N do not eventually match on P, and hence $M \neq_\beta N$ by Theorem 61 (for atomic Lévy–Longo Trees).

6. Statman's Conjecture

R. Statman has conjectured that there is no fpc Y such that $Y =_\beta Y\delta$ where $\delta \equiv \lambda ab.b(ab)$. In an equivalent phrasing, there is no solution for the unknown Y in the following system of equations:

$$Y =_\beta \delta Y$$
$$Y =_\beta Y\delta$$

Note that $Y = \delta Y$ if and only if Y is an fpc, i.e., all fpcs are fixed points of δ. B. Intrigila gave a confirmation of this conjecture in (Int97), employing a syntactic analysis of the standard reductions to a hypothetical common reduct. The proof employs an induction on n on the number of x's produced in the common reduct. (This refers to both Y and $Y\delta$ having BT $\lambda x.x^\omega$; a more precise statement is below.) The proof in (Int97) seems to have a gap however for the base case of the above induction, as the present authors noticed in communication with B. Intrigila. As yet, this gap has not been closed.

An Analysis of Intrigila's Proof

If Y is a fixed point combinator, we have $Yx \twoheadrightarrow x(C[x])$ for some multi-hole context C with $C[x] =_\beta Yx$. A multi-hole context is a λ-term with multiple (0 or more) occurrences of $\square$, and context filling $C[M]$ replaces all occurrences of $\square$ with M. In the remainder of this section we fix a variable x that is fresh for all multi-hole contexts C used here.

Definition 64. A multi-hole context C is a *fixed point context (fpcx)* if $C[x] =_\beta x(C[x])$.

Obviously every fpc Y gives rise to an fpcx $Y\Box$. Moreover, for every fpcx C we have:

(i) $C[x] \twoheadrightarrow x(C'[x]) \twoheadleftarrow x(C[x])$ for some fpcx C', and
(ii) there exists a head reduction $C[x] \twoheadrightarrow_h x(C''[x])$ for some fpcx C'' with $C''[x] =_\beta C[x]$.

In (Int97), Intrigila suggests the following generalization of Statman's question to fpcxs:

Conjecture 65. There exists no fpcx C such that $\lambda x.C[x] =_\beta C[\delta]$.

To see why this is a generalization, note that $Y =_\beta \lambda x.Yx$ for every fpc Y. The advantage of working with fpcxs in place of fpcs is that if $C[x] \twoheadrightarrow_\beta x(C'[x])$ then C' is again an fpcx.

The following is a compressed rendering of the proof of (Int97).

We define the weight of a fixed point context C as follows:

$$w(C) = \min\{\, n \mid \lambda x.C[x] \twoheadrightarrow \lambda x.x^n H \twoheadleftarrow C[\delta],\ H \text{ not of the form } x\Box \,\}$$

Assume there exists an fpcx C with $\lambda x.C[x] = C[\delta]$. Then let C be such a context with minimal weight $w(C)$. Then there exist standard reductions

$$\sigma_1 : \lambda x.C[x] \twoheadrightarrow \lambda x.x^{w(C)}H \qquad\qquad \sigma_2 : C[\delta] \twoheadrightarrow \lambda x.x^{w(C)}H$$

with H not of the form $x\Box$.

We have $C[x] \twoheadrightarrow_h x(C'[x])$ for some fpcx C'. As a consequence, the standard reduction σ_2 starts with the same steps where x is replaced by δ:

$$\sigma_2 : C[\delta] \twoheadrightarrow_h \delta(C'[\delta]) \rightarrow_h \lambda x.x(C'[\delta]x) \twoheadrightarrow \lambda x.x^{w(C)}H$$

Throughout $C[\delta] \twoheadrightarrow_h \delta(C'[\delta])$ no abstraction is created at the root, but the final term of σ_2 has an abstraction at the root. Thus there must be an additional head step that creates the abstraction: $\sigma_3 : \delta(C'[\delta]) \rightarrow_h \lambda x.x(C'[\delta]x)$. Hence $w(C) > 0$. Let $H' \equiv x^{w(C)-1}(H)$. Then $C'[x] \twoheadrightarrow H' \twoheadleftarrow C'[\delta]x$ by σ_2 and σ_3. If

$$\lambda x.H' \twoheadleftarrow C'[\delta], \tag{$*$}$$

then $w(C') \leq w(C) - 1$ contradicts the choice of C.

If we have $(*)$, then we are finished. Unfortunately, in (Int97), the proof of $(*)$ is left to the reader, see (Int97, Claim 3). For the case $w(C) \geq 2$ the argument is indeed trivial. However, for the base case $w(C) = 1$ we were not able to prove $(*)$.

Statman's Conjecture in a Wider Perspective

Statman's $Y =_\beta Y\delta$ problem, aptly paraphrased by Statman and Intrigila as:

> *Does there exist a double fixed point combinator?*

is in our opinion far more important than a mere syntactic puzzle. We have the impression that it refers to deep structures in λ-calculus which may be only partially understood yet. The $Y =_\beta Y\delta$ problem, or its variations below, may require new techniques to discriminate λ-terms. As Intrigila remarked in (Int97) in a closing sentence:

> *There are at present hardly any techniques to prove such non-equations.*

Our present clocked λ-calculus endeavors to contribute in this respect.

Let us give a reason why $Y \neq_\beta Y\delta$ for any fpc Y is made more plausible. We can prove this non-equation for all fpcs Y that we have seen, for example those in the Böhm sequence and those in the Scott sequence, see Example 45. In fact, we can prove $Y \neq_\beta Y\delta$ for all simple or simplifiable fpcs Y, also for some non-simplifiable fpcs, see Example 62.

Statman's conjecture can be seen as part of a much more encompassing conjecture, as follows. Here we call a context C an *fpc generating context* if $C[Y]$ is an fpc for every fpc Y, see (EHKP12). We consider the following fpc generating contexts

$$\Box\delta \qquad\qquad\qquad \Box(\mathsf{SS})\mathsf{S}^{\sim k}\mathsf{I} \quad (k \in \mathbb{N})$$

Conjecture 66. There are no non-trivial identifications between the fpcs thus obtained. More precisely, we have that $C[Y] \neq_\beta D[Y]$ for all fpcs Y and contexts $C = C_1[C_2[\ldots[C_n[\Box]]\ldots]]$, and $D = D_1[D_2[\ldots[D_m[\Box]]\ldots]]$ such that $C \not\equiv D$, where $C_1, \ldots, C_n$ and $D_1, \ldots, D_m$ are fpc generating contexts displayed above.

There are several interesting further variations on Statman's conjecture:

(i) $Z \neq_\beta Z\delta$ for wpcs Z;
(ii) $Y =_\beta Y'$ iff $Y\delta = Y'\delta$ for fpcs Y, Y'.

Finally we quote R. Smullyan:

> *The theory of sage birds (technically called* fixed point combinators*) is a fascinating and basic part of combinatory logic; we have only scratched the surface.*
>
> R. Smullyan (Smu85).

7. Concluding Remarks

In future work we intend to extend the current clock and discrimination techniques to the setting of simply typed λ-calculus, as in Plotkin's PCF (Plo77). Such an extension is even more interesting with respect to our interest in fpcs, as PCF has fpcs built-in as primitives.

A second extension is to extend pure lambda calculus with the μ-operator, with the reduction rule $\mu x.M \to M[x := \mu x.M]$. Although the μ-operator and its reduction rule are directly definable in λ-calculus, the interplay between λ and μ is quite interesting, as is the employment of μ in rendering fpcs. It is possible to define a clocked $\lambda\mu$-calculus, in analogy to the clocked calculus of the present paper. A combination of μ and simple types is also in a preliminary way studied in the wake of this paper, but its elaboration will only be in forthcoming work.

A third extension is to consider the `letrec` constructor, yielding the existence of solutions to arbitrary systems of equations.

References

K. Aehlig and F. Joachimski. On Continuous Normalization. In *Proc. Workshop on Computer Science Logic (CSL 2002)*, volume 2471 of *LNCS*, pages 59–73. Springer, 2002.

S. Abramsky and C.-H.L. Ong. Full Abstraction in the Lazy Lambda Calculus. *Information and Computation*, 105(2):159–267, 1993.

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.

A. Berarducci. Infinite λ-Calculus and Non-Sensible Models. In *Logic and Algebra (Pontignano, 1994)*, pages 339–377. Dekker, New York, 1996.

I. Bethke. Lambda Calculus, 2003. Chapter 10 in (Ter03).

H.P. Barendregt and J.W. Klop. Applications of Infinitary Lambda Calculus. *Information and Computation*, 207(5):559–582, 2009.

I. Bethke, J.W. Klop, and R.C. de Vrijer. Descendants and Origins in Term Rewriting. *Information and Computation*, 159(1–2):59–124, 2000.

Th. Coquand and H. Herbelin. *A*-Translation and Looping Combinators in Pure Type Systems. *Journal of Functional Programming*, 4(1):77–88, 1994.

Th. Coquand. Infinite Objects in Type Theory. In H. Barendregt and T. Nipkow, editors, *TYPES*, volume 806, pages 62–78. Springer–Verlag, Berlin, 1994.

J. Endrullis, D. Hendriks, and J.W. Klop. Modular Construction of Fixed Point Combinators and Clocked Böhm Trees. In *Proc. Symp. on Logic in Computer Science (LICS 2010)*, pages 111–119, 2010.

J. Endrullis, D. Hendriks, J.W. Klop, and A. Polonsky. Discriminating Lambda-Terms using Clocked Böhm Trees. *Logical Methods in Computer Science*, 2012. In print.

A.A. Faustini. *The Equivalence of an Operational and a Denotational Semantics for Pure Dataflow*. PhD thesis, University of Warwick, 1982.

H. Geuvers and B. Werner. On the Church–Rosser Property for Expressive Type Systems and its Consequences for their Metatheoretic Study. In *Proc. Symp. on Logic in Computer Science (LICS 1994)*, pages 320–329, 1994.

B. Intrigila. Non-Existent Statman's Double Fixed Point Combinator Does Not Exist, Indeed. *Information and Computation*, 137(1):35–40, 1997.

J.W. Klop and R.C. de Vrijer. Infinitary Normalization. In *We Will Show Them: Essays in Honour of Dov Gabbay*, volume 2, pages 169–192. College Publ., 2005. Techn. report: `http://www.cwi.nl/ftp/CWIreports/SEN/SEN-R0516.pdf`.

R. Kennaway, J.W. Klop, M.R. Sleep, and F.-J. de Vries. Infinitary Lambda Calculus. *Theoretic Computer Science*, 175(1):93–125, 1997.

J.W. Klop. New Fixed Point Combinators from Old. In *Reflections on Type Theory, λ-Calculus, and the Mind. Essays Dedicated to Henk Barendregt on the Occasion of his 60th Birthday*, pages 197–210. 2007. Online version: `http://www.cs.ru.nl/barendregt60`.

J. Ketema and J.G. Simonsen. Infinitary Combinatory Reduction Systems: Confluence. *Logical Methods in Computer Science*, 5(4):1–29, 2009.

S.G. Matthews. *Metric Domains for Completeness*. PhD thesis, University of Warwick, 1985.

W. McCune and L. Wos. The Absence and the Presence of Fixed Point Combinators. *Theoretic Compututer Science*, 87(1):221–228, 1991.

T. Naoi and Y. Inagaki. Algebraic Semantics and Complexity of Term Rewriting Systems. In *Proc. Conf. on Rewriting Techniques and Applications (RTA 1989)*, volume 355 of *Lecture Notes in Computer Science*, pages 311–325. Springer, 1989.

D. Park. The Fairness Problem and Nondeterministic Computing Networks. *Foundations of Computer Science IV, Distributed Systems: Part 2*, (159):133–161, 1983.

G.D. Plotkin. Lcf considered as a programming language. *Theoretical Computer Science*, 5(3):223–255, 1977.

G.D. Plotkin, 2007. Personal communication at the symposium for H. Barendregt's 60th birthday.

R. Smullyan. *To Mock a Mockingbird, and Other Logic Puzzles: Including an Amazing Adventure in Combinatory Logic*. Alfred A. Knopf, New York, 1985.

D. Sangiorgi and J.J.M.M. Rutten. *Advanced Topics in Bisimulation and Coinduction*, volume 52 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2012.

Terese. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.

W.W. Wadge. An Extensional Treatment of Dataflow Deadlock. *Theoretical Computer Science*, 13:3–15, 1981.