Modular Construction of Fixed Point Combinators and Clocked Böhm Trees

Jörg Endrullis Dimitri Hendriks Jan Willem Klop

Vrije Universiteit Amsterdam, Department of Computer Science
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

joerg@few.vu.nl diem@cs.vu.nl jwk@cs.vu.nl

***Abstract*—Fixed point combinators (and their generalization: looping combinators) are classic notions belonging to the heart of λ-calculus and logic. We start with an exploration of the structure of fixed point combinators (fpc's), vastly generalizing the well-known fact that if Y is an fpc, $Y(\mathsf{SI})$ is again an fpc, generating the Böhm sequence of fpc's. Using the infinitary λ-calculus we devise infinitely many other generation schemes for fpc's. In this way we find schemes and building blocks to construct new fpc's in a modular way.**

Having created a plethora of new fixed point combinators, the task is to prove that they are indeed new. That is, we have to prove their β-inconvertibility. Known techniques via Böhm Trees do not apply, because all fpc's have the same Böhm Tree (BT). Therefore, we employ 'clocked BT's', with annotations that convey information of the tempo in which the data in the BT are produced. BT's are thus enriched with an intrinsic clock behaviour, leading to a refined discrimination method for λ-terms. The corresponding equality is strictly intermediate between $=_\beta$ and $=_{\mathsf{BT}}$, the equality in the classical models of λ-calculus. An analogous approach pertains to Lévy–Longo and Berarducci trees. Finally, we increase the discrimination power by a precision of the clock notion that we call 'atomic clock'.

> 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 [19].

I. Introduction

Böhm trees constitute a well-known method to discriminate λ-terms M, N: if $\mathsf{BT}(M)$ and $\mathsf{BT}(N)$ are not identical, then M and N are β-inconvertible, $M \neq_\beta N$. But how do we prove β-inconvertibility of λ-terms with the same BT? This question was raised in Scott [18] for the interesting equation $\mathsf{BY} = \mathsf{BYS}$ between terms that as Scott noted are presumably β-inconvertible, yet BT-equal ($=_{\mathsf{BT}}$). Scott used his Induction Rule to prove that $\mathsf{BY} = \mathsf{BYS}$; instead we will employ below the infinitary λ-calculus with the same effect, but with more convenience for calculations as a direct generalization of finitary λ-calculus. Often one can solve such a β-discrimination problem by finding a suitable invariant for all the β-reducts of M, N. Below we will do this by way of preparatory example for the fixed point combinators (fpc's) in the Böhm sequence. But a systematic method for this discrimination problem has been lacking, and such a method is one of the two contributions of this paper.

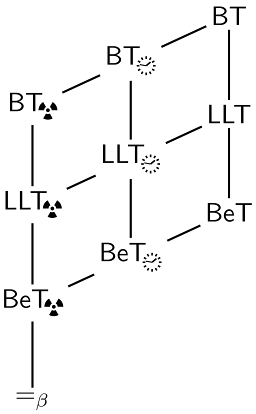

Fig. 1. Comparison of (atomic) clock semantics and unclocked semantics. Higher means more identifications.

Actually, the need for such a strategic method was forced upon us, by the other contribution, because Scott's equation $\mathsf{BY} = \mathsf{BYS}$ turned out to be the key unlocking a plethora of new fpc's. The new generation schemes are of the form: if Y is an fpc, then $YP_1 \dots P_n$ is an fpc, abbreviated as $Y \Rightarrow YP_1 \dots P_n$. So $\Box P_1 \dots P_n$ is an 'fpc-generating' vector, and can be considered as a building block to make new fpc's. But are they indeed new? A well-known example of a (singleton)-fpc-generating vector is $\Box\delta$, where $\delta = \mathsf{SI}$, giving rise when starting from Curry's fpc to the Böhm sequence of fpc's. Here another interesting equation is turning up, namely $Y = Y\delta$, for an arbitrary fpc Y, considered by Statman and Intrigila. In fact, it is implied by Scott's equation, for an arbitrary fpc Y:

$$\mathsf{B}Y = \mathsf{B}Y\mathsf{S} \implies \mathsf{B}Y\mathsf{I} = \mathsf{B}Y\mathsf{SI} \iff Y = Y\delta$$

The first equation $\mathsf{B}Y = \mathsf{B}Y\mathsf{S}$ will yield many new fcp's, built in a modular way; the last equation $Y = Y\delta$ addresses the question whether they are indeed new. Finding ad hoc invariant proofs for their novelty is too cumbersome. But fortunately, it turns out that although the new fpc's all have the same BT, namely $\lambda f.f^\omega$, they differ in the way this BT is formed, in the 'tempo of formation', where the ticks of the clock are head reduction steps. More generally, we can discern a clock-like behaviour of BT's, that enables us to discriminate the terms in question. However, this refined discrimination method does not work for all λ-terms; only for a class of 'simple' terms, that still is fairly extensive; it includes all fpc's that are constructed in the modular way that we present, thereby solving our novelty problem. In fact, we gain some more ground: though our discrimination method works best for pairs of simple terms, it can also fruitfully be applied to compare a simple term with a non-simple term, and with some more effort, we can even compare and discriminate two non-simple terms.

Even so, many pairs of fpc's cannot yet be discriminated, because they not only have the same BT, they also have the same clocked BT. Therefore, in a final grading up of the precision, we introduce 'atomic clocks', where the actual position of a head reduction step is administrated. All this pertains not only to the BT-semantics, but also to Lévy–Longo Trees (LLT) (or lazy trees), and Berarducci Trees (BeT) (or syntactic trees). Many problems stay open, in particular problems generalizing the equation of Statman and Intrigila, when arbitrary fpc's are considered — indeed, we have only scratched the surface.

In the first part of our paper, up to Section VI, we are concerned with constructing new fpc's from old, by some generating schemes. The underlying heuristics is given by employing infinitary λ-calculus. As related work we mention [11], where a heuristic procedure in finitary λ-calculus is given to construct fpc's in a uniform way.

The present paper is an extension and elaboration of the unpublished, unrefereed informal note [15] by one of the authors; that note contained no proofs nor the notion of atomic clocks.

II. Preliminaries

To make this paper moderately self-contained, and to fix notations, we lay out some ingredients. For λ-calculus we refer to [2] and [5]. For an introduction to Böhm, Berarducci and Lévy–Longo trees, we refer to [1–3,6].

Definition 1. λ-terms are defined by the grammar:

$$M ::= x \mid \lambda x.M \mid MM$$

We let $\mathit{Ter}(\lambda)$ denote the set of λ-terms, and use $M, N, \ldots$ to range over the elements of $\mathit{Ter}(\lambda)$. The relation $\to_\beta$ is the compatible closure (i.e., closure under term formation) of the β-rule:

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

where $M[N/x]$ denotes the result of substituting N for all free occurrences of x in M. Furthermore, we use $\twoheadrightarrow_\beta$ to denote the reflexive–transitive closure of $\to_\beta$, and $\to_\beta^{\equiv}$ for the reflexive closure. We write $M =_\beta N$ to denote that M is β-convertible with N, i.e., $=_\beta$ is the equivalence closure of $\to_\beta$. We write $\mathrel{\circ\!\!\!\!\rightarrow}_\beta$ for multi-steps [2], that is, complete developments of a set of redex occurrences in a term. For syntactic equality (modulo renaming of bound variables), we use $\equiv$. We will often omit the subscript β in $\to_\beta$, $\twoheadrightarrow_\beta$ and $\mathrel{\circ\!\!\!\!\rightarrow}_\beta$, but not so for $=_\beta$, in order to reserve $=$ for definitional equality.

A λ-term M is called *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. For λ-terms M having a normal form we write $\underline{M}$ for the unique normal form N with $M \twoheadrightarrow N$ (note that uniqueness follows from confluence of the λ-calculus).

Some commonly used combinators are:

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

Definition 2. A *position* is a sequence over $\{0, 1, 2\}$. The *subterm $M|_p$ of M at position p* is defined by:

$$\begin{aligned} M|_\epsilon &= M & (MN)|_{1p} &= M|_p \\ (\lambda x.M)|_{0p} &= M|_p & (MN)|_{2p} &= N|_p \end{aligned}$$

$\mathcal{P}os(M)$ is the set of positions p such that $M|_p$ is defined.

Definition 3.

(i) A term Y is an *fpc* if $Yx =_\beta x(Yx)$.
(ii) An fpc Y is *k-reducing* if $Yx \to^k x(Yx)$.
(iii) A term Z is a *weak fpc (wfpc)* if $Zx =_\beta x(Z'x)$ where Z' is a wfpc.

A wfpc is alternatively defined as a term having the same Böhm tree as an fpc, namely $\lambda x.x^\omega \equiv \lambda x.x(x(x(\ldots)))$. Weak fpc's are known in foundational studies of type systems as *looping combinators*; see, e.g., [8] and [10].

Example 4. 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.

As to 'double recursion', [15] collects several proofs of the double fixed point theorem, including some in [2,19].

Definition 5.

(i) A *head reduction step* $\to_h$ is a β-reduction step of the form: $\lambda x_1 \ldots x_n.(\lambda y.M)NN_1 \ldots N_m \to \lambda x_1 \ldots x_n.(M[y/N])N_1 \ldots N_m$ with $n, m \geq 0$.
(ii) Accordingly, a *head normal form (hnf)* is a λ-term of the form $\lambda x_1. \ldots \lambda x_n.yN_1 \ldots N_m$ with $n, m \geq 0$.
(iii) A *weak head normal form (whnf)* is an hnf or an abstraction, that is, a whnf is a term of the form $xM_1 \ldots M_m$ 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.

Infinitary λ-calculus $\lambda^\infty\beta$: We will only use the infinitary λ-calculus $\lambda^\infty\beta$ for some simple calculations such as $(\lambda ab.(ab)^\omega)\mathsf{I} =_{\lambda^\infty\beta} \lambda b.(\mathsf{I}b)^\omega =_{\lambda^\infty\beta} \lambda b.b^\omega$. For a proper setup of $\lambda^\infty\beta$ we refer to [3,4,13,14]. In a nutshell, $\lambda^\infty\beta$ extends finitary λ-calculus by admitting infinite λ-terms, the set of which is called $\mathit{Ter}^\infty(\lambda)$, and infinite reductions (in [3,13] possibly transfinitely long, in [4] of length $\leq \omega$). Limits of infinite reduction sequences are obtained by a strengthening of Cauchy-convergence, stipulating that the depth of contracted redexes must tend to infinity. The $\lambda^\infty\beta$-calculus is not infinitary confluent (CR^∞), but still has unique infinite normal forms (UN^∞). Böhm Trees (BT's) without $\bot$ are infinite normal forms in $\lambda^\infty\beta$. But beware, the reverse does not hold, e.g. $\lambda x.(\lambda x.(\lambda x.\ldots))$ is an infinite normal form, but not a BT; it is in fact an LLT (Lévy–Longo Tree, and also a BeT (Berarducci Tree). The notions BT, LLT, BeT are defined e.g. in [3], and in [6]. These notions are also defined in Sections VI and VIII, via their clocked versions.

Definition 6. For terms A, B we define $AB^{\sim n}$ and A^nB:

$$AB^{\sim 0} = A \qquad A^0B = B$$
$$AB^{\sim n+1} = ABB^{\sim n} \qquad A^{n+1}B = A(A^nB)$$

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

III. The Böhm Sequence

There are several ways to make fpc's. For heuristics behind the construction of Curry's fpc $\mathsf{Y}_0 = \lambda f.\omega_f\omega_f$, with $\omega_f = \lambda x.f(xx)$, and Turing's fpc $\mathsf{Y}_1 = \eta\eta$ with $\eta = \lambda xf.f(xxf)$, see [2,15].

It is well-known, as observed by C. Böhm [2,7], that the class of fpc's coincides exactly with the class of fixed points of the peculiar term $\delta = \lambda ab.b(ab)$, convertible with SI. The notation δ is convenient for calculations and stems from [12]. This term also attracted the attention of R. Smullyan, in his beautiful fable about fpc's figuring as birds in an enchanted forest: "An extremely interesting bird is the owl O defined by the following condition: $Oxy = y(xy)$." [19]. We will return to the Owl in Remark 9 below.

Thus the term $Y\delta$ is an fpc whenever Y is. It follows that starting with Y_0, Curry's fpc, we have an infinite sequence of fpc's $\mathsf{Y}_0, \mathsf{Y}_0\delta, \mathsf{Y}_0\delta\delta, \ldots, \mathsf{Y}_0\delta^{\sim n}, \ldots$. We call this sequence the *Böhm sequence*. We will indicate $\mathsf{Y}_0\delta^{\sim n}$ by Y_n. Note that $\mathsf{Y}_0\delta =_\beta \eta\eta$, justifying the overloaded notation Y_1. Now the question is whether all these 'derived' fpc's are really new, in other words, whether the sequence is free of duplicates. This is *Exercise 6.8.9 in [2].

Note that we could also have started the sequence from another fpc than Curry's. Now for the sequence starting from an *arbitrary* fpc Y, it is actually an open problem whether that sequence of fpc's $Y, Y\delta, Y\delta\delta, \ldots, Y\delta^{\sim n}, \ldots$ is free of repetitions. All we know, applying Intrigila's theorem, Theorem 8 below, is that no two consecutive fpc's in this sequence are convertible. But let us first consider the Böhm sequence.

We show that the Böhm sequence contains no duplicates by determining the set of reducts of every Y_n. For $n \geq 1$ we take $\mathsf{Y}_n = \eta\eta\delta^{\sim(n-1)}$ ($=_\beta \mathsf{Y}_0\delta^{\sim n}$). For Y_3, the head reduction is displayed in Figure 2, but this is by no means the whole reduction graph. For future reference we note that the head reduction diagram suggests a 'clock behaviour'.

$$\begin{tikzcd}
\eta\eta\delta\delta x \arrow[r, "6", "h"'] & x(\eta\eta\delta\delta x) \arrow[l, bend left]
\end{tikzcd}$$

Fig. 2. Head reduction of Y_3x.

Theorem 7. *The Böhm sequence contains no duplicates.*

Proof: We define languages $\mathcal{L}_n \subseteq \mathit{Ter}(\lambda)$ where $\mathcal{L}_n$ is the set of $\twoheadrightarrow$-reducts of Y_n.

$$\begin{aligned}
&\mathcal{L}_0 ::= \lambda f.f^k(\omega_f\omega_f) && (k \geq 0)\\
&\mathcal{L}_1 ::= \eta\eta \mid \lambda f.f^k(\mathcal{L}_1 f) && (k > 0)\\
&\mathcal{L}_n ::= \mathcal{L}_{n-1}\delta \mid \lambda b.b^k(\mathcal{L}_n b) \mid \delta\mathcal{L}_n && (n > 1, k > 0)
\end{aligned}$$

Then we show that:

(i) $\mathcal{L}_n$ is closed under β-reduction; and
(ii) $\mathcal{L}_n$ and $\mathcal{L}_m$ are disjoint, for $n \neq m$.

This implies that $\mathsf{Y}_n \neq_\beta \mathsf{Y}_m$ for all $n \neq m$.

For $n \neq m$, $n > 1$, (ii) follows by counting the number of passive δ's. An occurrence of δ is passive if it occurs as $P\delta$ for some P. To see that $\mathcal{L}_0 \cap \mathcal{L}_1 = \varnothing$, note that if $M \in \mathcal{L}_1$ is an abstraction, then $M \equiv \lambda f.f^k(Pf)$ containing a subterm Pf which is never the case in $\mathcal{L}_0$.

We show (i): if $M \in \mathcal{L}_n$ and $M \to N$, then $N \in \mathcal{L}_n$. Using induction, we do not need to consider cases where the rewrite step is inside a variable of the grammar. We write $\mathcal{L}_n$ in terms as shorthand for a term $M \in \mathcal{L}_n$.

($\mathcal{L}_0$) We have $\lambda f.f^k(\omega_f\omega_f) \to \lambda f.f^{k+1}(\omega_f\omega_f) \in \mathcal{L}_0$.
($\mathcal{L}_1$) We have $\eta\eta \to \lambda f.f(\eta\eta f) \in \mathcal{L}_1$,
and $\lambda f.f^k(\lambda f.f^\ell(\mathcal{L}_1 f)f) \to \lambda f.f^{k+\ell}(\mathcal{L}_1 f) \in \mathcal{L}_1$.
($\mathcal{L}_n$) Case 1: $\lambda f.f^k(\mathcal{L}_1 f)\delta \to \delta^k(\mathcal{L}_1\delta) \in \mathcal{L}_n$ for $n = 2$, and $(\lambda b.b^k(\mathcal{L}_{n-1}b))\delta \to \delta^k(\mathcal{L}_{n-1}\delta) \in \mathcal{L}_n$ for $n > 2$.
Case 2: $\lambda b.b^k(\lambda c.c^\ell(\mathcal{L}_n c)b) \to \lambda b.b^{k+\ell}(\mathcal{L}_n b) \in \mathcal{L}_n$.
Case 3: $\delta\mathcal{L}_n \to \lambda b.b(\mathcal{L}_n b) \in \mathcal{L}_n$. ■

A very interesting theorem involving δ was proved by B. Intrigila, affirming a conjecture by R. Statman.

Theorem 8 (Intrigila [12])**.** *There is no 'double' fixed point combinator. That is, for no fpc Y we have $Y\delta =_\beta Y$.*

Remark 9 (Smullyan's Owl $\mathsf{SI} =_\beta \delta = \lambda ab.b(ab)$)**.** We collect some salient facts and questions.

(i) If Z is a wfpc, both δZ and $Z\delta$ are wfpc's [19].
(ii) Call an applicative combination of δ's a δ-term. In spite of δ's simplicity, not all δ-terms are strongly normalizing (SN). An example of a δ-term with infinite reduction is $\delta\delta(\delta\delta)$ (Johannes Waldman, Hans Zantema, personal communication, 2007).
(iii) Let M be a non-trivial δ-term, i.e., not a single δ. Then M is SN iff M contains exactly one occurrence of $\delta\delta$. Furthermore, if δ-terms M, M' are SN, then they are convertible iff M, M' have the same length [15].
(iv) Convertibility is decidable for δ-terms [20].
(v) Call $\Delta = \delta^\omega$, so $\Delta \equiv \delta\Delta$. Then, the infinite λ-term Δ is an fpc: $\Delta x \equiv \delta\Delta x \twoheadrightarrow x(\Delta x)$. Δ can be normalized again: $\Delta \to_\omega \lambda f.f^\omega$. There are many more infinitary fpc's, e.g., for every n, the infinite term $(\mathsf{SS})^\omega\mathsf{S}^{\sim n}\mathsf{I}$ is one, as will be clear from the sequel.
(vi) $\mathsf{BT}(\delta\delta(\delta\delta)) \equiv \bot$, $\delta\delta(\delta\delta)$ has no hnf. Its Berarducci tree is not trivial. Zantema remarked that δ-terms, even

infinite ones, such as $\Delta\Delta$, are “top-terminating” (Zantema restricted himself to the applicative rule for δ only — we expect that his observation remains valid for the $\lambda\beta$-version).

(vii) Is Intrigila's theorem also valid for wfpc's: for no wfpc Z we have $Z\delta =_\beta Z$?

IV. The Scott Sequence

In [18, p. 360] the equation $\mathsf{B}\mathsf{Y}_0 = \mathsf{B}\mathsf{Y}_0\mathsf{S}$ is mentioned as an interesting example of an equation not provable in $\lambda\beta$[1], while easily provable with Scott's Induction Rule. Scott mentions that he expects that using ‘methods of Böhm’ the non-convertibility in $\lambda\beta$ can be established, but that he did not attempt a proof. On the other hand, with the induction rule the equality is easily established. We will not consider Scott's Induction Rule, but we will be working in the infinitary lambda calculus, $\lambda^\infty\beta$. It is readily verified that in $\lambda^\infty\beta$ we have:

$$\mathsf{B}\mathsf{Y}_0 =_{\lambda^\infty\beta} \mathsf{B}\mathsf{Y}_0\mathsf{S} =_{\lambda^\infty\beta} \lambda ab.(ab)^\omega$$

Proposition 10. $\mathsf{B}\mathsf{Y}_0 \neq_\beta \mathsf{B}\mathsf{Y}_0\mathsf{S}$

Proof: Postfixing the combinator I yields $\mathsf{B}\mathsf{Y}_0\mathsf{I}$ and $\mathsf{B}\mathsf{Y}_0\mathsf{S}\mathsf{I}$. Now $\mathsf{B}\mathsf{Y}_0\mathsf{I} =_\beta \mathsf{Y}_0$ and $\mathsf{B}\mathsf{Y}_0\mathsf{S}\mathsf{I} =_\beta \mathsf{Y}_0(\mathsf{S}\mathsf{I}) = \mathsf{Y}_1$. Because $\mathsf{Y}_0 \neq \mathsf{Y}_1$ (Theorem 7), the result follows. ■

In the same way we can strengthen this non-equation to all fpc's Y, using Theorem 8.

Remark 11.

(i) The idea of postfixing an I is suggested by the BT $\lambda ab.(ab)^\omega$ of $\mathsf{B}Y$ and $\mathsf{B}Y\mathsf{S}$. Namely, in $\lambda^\infty\beta$ we calculate: $(\lambda ab.(ab)^\omega)\mathsf{I} = \lambda b.(\mathsf{I}b)^\omega = \lambda b.b^\omega$ which is the BT of any fpc.

(ii) Interestingly, Scott's equation $\mathsf{B}Y = \mathsf{B}Y\mathsf{S}$ implies the equation of Statman and Intrigila, $Y = Y\delta$ as one readily verifies, as in the proof of Proposition 10.

Actually, the comparison between the terms $\mathsf{B}Y$ and $\mathsf{B}Y\mathsf{S}$ has more in store for us than just providing an example that the extension from finitary lambda calculus $\lambda\beta$ to infinitary lambda calculus $\lambda^\infty\beta$ is not conservative. The BT-equality of $\mathsf{B}Y$ and $\mathsf{B}Y\mathsf{S}$ suggests looking at the whole sequence $\mathsf{B}Y, \mathsf{B}YS, \mathsf{B}Y\mathsf{S}\mathsf{S}, \ldots, \mathsf{B}Y\mathsf{S}^{\sim n}, \ldots$. All these terms have the BT $\lambda ab.(ab)^\omega$, and hence they are not fpc's. But postfixing an I turns them into fpc's.

Definition 12. The *Scott sequence* is defined by:

$$\mathsf{B}\mathsf{Y}_0\mathsf{I},\ \mathsf{B}\mathsf{Y}_0\mathsf{S}\mathsf{I},\ \mathsf{B}\mathsf{Y}_0\mathsf{S}\mathsf{S}\mathsf{I},\ \ldots,\ \mathsf{B}\mathsf{Y}_0\mathsf{S}^{\sim n}\mathsf{I}, \ldots$$

We write $\mathsf{U}_n = \mathsf{B}\mathsf{Y}_0\mathsf{S}^{\sim n}\mathsf{I}$ for the n-th term in this sequence.

The Scott sequence concurs with the Böhm sequence of fpc's only for the first two elements, and then splits off with different fpc's. But there is a second surprise. In showing that U_n is an fpc, we find as a bonus the fpc-generating vector $\Box(\mathsf{S}\mathsf{S})\mathsf{S}^{\sim n}\mathsf{I}$ (which does preserve reducingness).

Theorem 13. *Let Y be a k-reducing fpc and $n \geq 0$. Then:*

(i) $\mathsf{B}Y\mathsf{S}^{\sim n}\mathsf{I}$ *is a (non-reducing) fpc;*

(ii) $Y(\mathsf{S}\mathsf{S})\mathsf{S}^{\sim n}\mathsf{I}$ *is a $(k + 3n + 7)$-reducing fpc.*

The proof of Theorem 13 is easy: see the next example.

Example 14. Let Y be a k-reducing fpc. Then:

$$\begin{aligned} \mathsf{B}Y\mathsf{S}\mathsf{S}\mathsf{S}\mathsf{I}x \twoheadrightarrow_h\ & Y(\mathsf{S}\mathsf{S})\mathsf{S}\mathsf{I}x \rightarrow_h^k \mathsf{S}\mathsf{S}(Y(\mathsf{S}\mathsf{S}))\mathsf{S}\mathsf{I}x \\ \rightarrow_h^3\ & \mathsf{S}\mathsf{S}(Y(\mathsf{S}\mathsf{S})\mathsf{S})\mathsf{I}x \rightarrow_h^3 \mathsf{S}\mathsf{I}(Y(\mathsf{S}\mathsf{S})\mathsf{S}\mathsf{I})x \\ \rightarrow_h^3\ & \mathsf{I}x(Y(\mathsf{S}\mathsf{S})\mathsf{S}\mathsf{I}x) \rightarrow_h^1 x(Y(\mathsf{S}\mathsf{S})\mathsf{S}\mathsf{I}x) \end{aligned}$$

This shows that $\mathsf{B}Y\mathsf{S}^{\sim 3}\mathsf{I}$ is a non-reducing fpc, and at the same time that $Y(\mathsf{S}\mathsf{S})\mathsf{S}\mathsf{I}x$ is reducing.

Remark 15. Another ‘fpc-generating vector’ is obtained as follows. Start with the equation $Mab = ab(Mab)$; solutions all have the BT seen above, $\lambda ab.(ab)^\omega$. For every M satisfying this equation, we have that $M\mathsf{I}$ is an fpc. For: $M\mathsf{I}x = \mathsf{I}x(M\mathsf{I}x) = x(M\mathsf{I}x)$. Now we can solve the equation in different ways.

(i) $Mab = Y(ab)$, so $M = \lambda ab.Y(ab) = (\lambda yab.y(ab))Y = \mathsf{B}Y$, as found before.

(ii) $Mab = ab(Mab) = \mathsf{S}a(Ma)b$, which is obtained by solving $Ma = \mathsf{S}a(Ma)$, leading to $Ma = Y(\mathsf{S}a) = \mathsf{B}Y\mathsf{S}a$, so $M = \mathsf{B}Y\mathsf{S}$. Also this solution was considered before.

(iii) $M = \lambda ab.ab(Mab) = (\lambda mab.ab(mab))M$, yielding $M = Y\varepsilon$ with $\varepsilon = \lambda abc.bc(abc)$. So if Y is an fpc, then $Y\varepsilon\,\mathsf{I}$ is again an fpc.

V. Generalized Generation Schemes

The schemes mentioned in Theorem 13 and Remark 15 (iii) for generating new fixed points from old, are by no means the only ones. There are in fact infinitely many of such schemes. They can be obtained analogously to the ones that we extracted above from the equation $\mathsf{B}Y = \mathsf{B}Y\mathsf{S} = \lambda ab.(ab)^\omega$, or the equation $Mab = ab(Mab)$. We only treat the case for $n = 3$: consider the equation $Nabc = abc(Nabc)$. Then every solution N is again a ‘pre-fpc’, namely $N\mathsf{I}\mathsf{I}$ is an fpc: $N\mathsf{I}\mathsf{I}x =_\beta \mathsf{I}\mathsf{I}x(N\mathsf{I}\mathsf{I}x) =_\beta x(N\mathsf{I}\mathsf{I}x)$.

(i) $Nabc = Y(abc)$, which yields $N = (\lambda yabc.y(abc)))Y = (\lambda yabc.\mathsf{B}\mathsf{B}\mathsf{B}yabc)Y$. We obtain $N = \mathsf{B}\mathsf{B}\mathsf{B}Y$.

(ii) $N = Y\xi$ with $\xi = \lambda nabc.abc(nabc)$, yielding the fpc-generating vector $\Box\xi\mathsf{I}\mathsf{I}$.

(iii) $Nabc = abc(Nabc) = \mathsf{S}(ab)(Nab)c$. So we take $Nab = \mathsf{S}(ab)(Nab)$, which yields $Nab = Y(\mathsf{S}(ab)) = \mathsf{B}\mathsf{B}\mathsf{B}Y(\mathsf{B}\mathsf{S})ab$. So $N = \mathsf{B}\mathsf{B}\mathsf{B}Y(\mathsf{B}\mathsf{S})$, and thus we find the equation $\mathsf{B}\mathsf{B}\mathsf{B}Y = \mathsf{B}\mathsf{B}\mathsf{B}Y(\mathsf{B}\mathsf{S})$, in analogy with the equation $\mathsf{B}Y = \mathsf{B}Y\mathsf{S}$ above.

Also this equation spawns lots of fpc's as well as fpc-generating vectors. Let's abbreviate $\mathsf{B}\mathsf{S}$ by A. First one forms the sequence

$$\mathsf{B}\mathsf{B}\mathsf{B}Y,\ \mathsf{B}\mathsf{B}\mathsf{B}Y\mathsf{A},\ \mathsf{B}\mathsf{B}\mathsf{B}Y\mathsf{A}\mathsf{A},\ \mathsf{B}\mathsf{B}\mathsf{B}Y\mathsf{A}\mathsf{A}\mathsf{A}, \ldots$$

[1] This equation is also discussed in [9].

These terms all have the BT $\lambda abc.abc(abc)^{\omega}$. They are not yet fpc's , but only 'pre-fpc's'. But after postfixing this time $\ldots \mathsf{II}$ we do again obtain a sequence of fpc's:

$$\mathsf{BBB}Y\mathsf{II},\ \mathsf{BBB}Y\mathsf{AII},\ \mathsf{BBB}Y\mathsf{AAII}, \ldots$$

Again the first two coincide with $\mathsf{Y}_0, \mathsf{Y}_1$, but the series deviates not only from the Böhm sequence but also from the Scott sequence above. As above, the proof that a term in this sequence is indeed an fpc, yields an fpc-generating vector. Thus we find e.g. the following new fpc-generating schemes, which we render in a self-explaining notation:

(i) $Y \Rightarrow Y(\mathsf{SS})\mathsf{I}$
(ii) $Y \Rightarrow Y(\mathsf{AAA})\mathsf{A}^{\sim n}\mathsf{II}$
(iii) $Y \Rightarrow Y(\mathsf{AII})$

(Note: scheme (iii) came up out of the general search; one may recognize that it is not a new scheme, because the term AII is actually the Owl δ). We can derive many more of these schemes by proceeding with solving the general equation $Na_1a_2...a_n = a_1a_2...a_n(Na_1a_2...a_n)$, bearing in mind the following proposition.

Proposition 16. *If N is a term satisfying $Na_1a_2 \ldots a_n = a_1a_2 \ldots a_n(Na_1a_2 \ldots a_n)$, then $N\mathsf{I}^{\sim(n-1)}$ is an fpc.*

We finally mention an fpc-generating scheme with 'dummy parameters'.

(iv) $Y \Rightarrow YQP_1 \ldots P_n$ where $P_1, \ldots, P_n$ are arbitrary (dummy) terms, and $Q = \lambda yp_1...p_nx.x(yp_1 \ldots p_nx)$.

VI. Clock Behaviour of Lambda Terms

As we have seen, there is vast space of fpc's and there are many ways to derive new fpc's. The question is whether all these fpc's are indeed new. So we have to prove that they are not β-convertible.

For the Böhm sequence we did this by an ad hoc argument based on a syntactic invariant; and this method works fine to establish lots of non-equations between the alleged 'new' fpc's that we constructed above. Still, the question remains whether there are not more 'strategic' ways of proving such inequalities.

In this section we propose a more strategic way to discriminate terms with respect to β-conversion. The idea is to extract from a λ-term more than just its BT, but also how the BT was formed; one could say, in what tempo, or in what rhythm. A BT is formed from static pieces of information, but these are rendered in a clock-wise fashion, where the ticks of the internal clock are head reduction steps.

In the sequel we write $[k]M$ for the term M where the root is *annotated with* $k \in \mathbb{N}$. Here, term formation binds stronger than annotation $[k]$. For example $[k]MN$ stands for the term $[k](MN)$ (that is, annotating the (non-displayed) application symbol in-between M and N, in contrast to $([k]M)N$). Moreover, for an annotated term M we use $\lfloor M \rfloor$ to denote the term obtaind from M by dropping all annotations (including annotations of subterms).

Definition 17 (Clocked Böhm trees)**.** Let M be a λ-term. The *clocked Böhm tree* $\mathsf{BT}_{\circlearrowright}(M)$ *of* M is an annotated potentially infinite term defined as follows. If M has no hnf, then define $\mathsf{BT}_{\circlearrowright}(M)$ as $\bot$. Otherwise, there is a head reduction $M \to_h^k \lambda x_1. \ldots \lambda x_n.yM_1 \ldots M_m$ to hnf. Then we define $\mathsf{BT}_{\circlearrowright}(M)$ as the term $[k]\lambda x_1. \ldots \lambda x_n.y\mathsf{BT}_{\circlearrowright}(M_1) \ldots \mathsf{BT}_{\circlearrowright}(M_m)$.

The (non-clocked) Böhm tree of a λ-term M can be obtained by dropping the annotations: $\mathsf{BT}(M) = \lfloor \mathsf{BT}_{\circlearrowright}(M) \rfloor$.

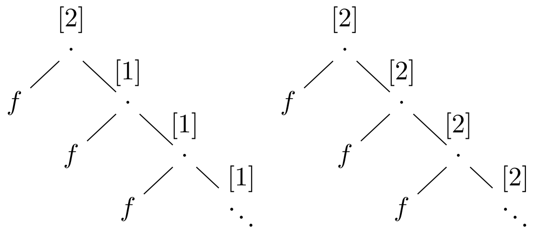

Fig. 3. Clocked Böhm trees of $\mathsf{Y}_0 f$ and $\mathsf{Y}_1 f$.

Let us consider the fpc's Y_0 of Curry and Y_1 of Turing. We have $\mathsf{Y}_0 \equiv \lambda f.\omega_f\omega_f$ where $\omega_f \equiv \lambda x.fxx$, and

$$\omega_f\omega_f \to_h^1 f(\omega_f\omega_f)$$

Therefore we obtain $\mathsf{BT}_{\circlearrowright}(\mathsf{Y}_0 f) = [2]f\mathsf{BT}_{\circlearrowright}(\omega_f\omega_f)$, and $\mathsf{BT}_{\circlearrowright}(\omega_f\omega_f) = [1]f\mathsf{BT}_{\circlearrowright}(\omega_f\omega_f)$.

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

$$\mathsf{Y}_1 f \equiv \eta\eta f \to_h^2 f(\eta\eta f)$$

Hence, $\mathsf{BT}_{\circlearrowright}(\mathsf{Y}_1 f) = [2]f\mathsf{BT}_{\circlearrowright}(\mathsf{Y}_1 f)$. Figure 3 displays the clocked Böhm trees of $\mathsf{Y}_0 f$ (left) and $\mathsf{Y}_1 f$ (right).

The following definition captures the well-known Böhm equality of λ-terms.

Definition 18. λ-terms M and N are BT-*equal*, denoted by $M =_{\mathsf{BT}} N$, if $\mathsf{BT}(M) \equiv \mathsf{BT}(N)$.

If M and N are not BT-equal, then $M \neq_\beta N$. More generally, if for some F, $\mathsf{BT}(MF) \not\equiv \mathsf{BT}(NF)$, then $M \neq_\beta N$. This method is known as *Böhm-out technique* [2].

Below, we refine this approach by comparing the clocked Böhm trees $\mathsf{BT}_{\circlearrowright}(M)$ and $\mathsf{BT}_{\circlearrowright}(N)$ instead of the ordinary (non-clocked) Böhm trees. In general, $\mathsf{BT}_{\circlearrowright}(M) \not\equiv \mathsf{BT}_{\circlearrowright}(N)$ does not always imply that $M \neq_\beta N$. Nevertheless, for a large class of λ-terms, called 'simple' below, this implication will turn out to be true.

We lift relations over natural numbers to relations over clocked Böhm trees.

Definition 19. Let T_1 and T_2 be clocked Böhm trees, $p \in \mathcal{Pos}(T_1) \cap \mathcal{Pos}(T_2)$ and $R \subseteq \mathbb{N} \times \mathbb{N}$.

We use $T_1 \ R_p \ T_2$ to denote that either both $T_1|_p$ and $T_2|_p$ are not annotated, or both are annotated and $T_1|_p \equiv [k_1]T_1'$ and $T_2|_p \equiv [k_2]T_2'$ with $k_1 \ R \ k_2$.

We write $T_1 \ R \ T_2$ if $\lfloor T_1 \rfloor \equiv \lfloor T_2 \rfloor$ and $T_1 \ R_p \ T_2$ for every $p \in \mathcal{Pos}(T_1)$.

We write $T_1 \; R_{\exists} \; T_2$, and say that R *holds eventually*, if $\lfloor T_1 \rfloor \equiv \lfloor T_2 \rfloor$ and there exists a depth level $\ell \in \mathbb{N}$ such that $T_1 \; R_p \; T_2$ for all positions $p \in \mathcal{P}os(T_1)$ with $|p| \geq \ell$.

Definition 20. For λ-terms M and N we say:

(i) M *improves* N *globally* if $\mathsf{BT}_{\circledcirc}(M) \leq \mathsf{BT}_{\circledcirc}(N)$,
(ii) M *improves* N *eventually* if $\mathsf{BT}_{\circledcirc}(M) \leq_{\exists} \mathsf{BT}_{\circledcirc}(N)$,
(iii) M *matches* N *eventually* if $\mathsf{BT}_{\circledcirc}(M) =_{\exists} \mathsf{BT}_{\circledcirc}(N)$.

The following proposition suggests that the ordering $>_{\circledcirc}$ on λ-terms defined by $M >_{\circledcirc} N$ if and only if $\mathsf{BT}_{\circledcirc}(M) \geq \mathsf{BT}_{\circledcirc}(N)$ is a 'semi-model' of β-reduction [17]. We leave this for future research.

Proposition 21. *Clocks are accelerated under reduction, that is, if $M \twoheadrightarrow N$, then the reduct N improves M globally* ($\mathsf{BT}_{\circledcirc}(M) \geq \mathsf{BT}_{\circledcirc}(N)$). *Dually, clocks slow down under expansion (the reverse of reduction).*

Proof: We proceed by an elementary diagram construction. Whenever we have co-initial steps $M \to_h M_1$ and $M \mathrel{\circ\!\!\!\rightarrow} M_2$, then by orthogonal projection [21] there exist joining steps $M_1 \mathrel{\circ\!\!\!\rightarrow} M'$ and $M_2 \to_h^{\equiv} M'$. Note that the head step $M \to_h M_1$ cannot be duplicated, only erased in case of an overlap. This leads to the elementary diagram displayed in Figure 4.

$$\begin{tikzcd}
M \arrow[r, "h"'] \arrow[d, "\circ"] & M_1 \arrow[d, "\circ"] \\
M_2 \arrow[r, "\equiv", "h"'] & M'
\end{tikzcd}$$

Fig. 4. Elementary diagram.

We have $\twoheadrightarrow \; \subseteq \; \mathrel{\circ\!\!\!\rightarrow}^*$. By induction on the length of the rewrite sequence $\mathrel{\circ\!\!\!\rightarrow}^*$ it suffices to show that $M \mathrel{\circ\!\!\!\rightarrow} N$ implies $\mathsf{BT}_{\circledcirc}(M) \geq \mathsf{BT}_{\circledcirc}(N)$. Let $M \mathrel{\circ\!\!\!\rightarrow} N$. If M has no hnf, then the same holds for N, and hence $\mathsf{BT}_{\circledcirc}(M) = \bot = \mathsf{BT}_{\circledcirc}(N)$. Assume that there exists a head rewrite sequence $M \to_h^k H \equiv \lambda x_1. \ldots \lambda x_n.y M_1 \ldots M_m$ to hnf. We have $\mathsf{BT}_{\circledcirc}(M) \equiv [k] \lambda x_1. \ldots \lambda x_n.y \mathsf{BT}_{\circledcirc}(M_1) \ldots \mathsf{BT}_{\circledcirc}(M_m)$.

Using the elementary diagram above (k times), we can project $M \mathrel{\circ\!\!\!\rightarrow} N$ over $M \to_h^k H$, and obtain $H \mathrel{\circ\!\!\!\rightarrow} H'$, $N \to_h^{\ell} H' \equiv \lambda x_1. \ldots \lambda x_n.y M'_1 \ldots M'_m$ with $\ell \leq k$. Then $\mathsf{BT}_{\circledcirc}(N) \equiv [\ell] \lambda x_1. \ldots \lambda x_n.y \mathsf{BT}_{\circledcirc}(M'_1) \ldots \mathsf{BT}_{\circledcirc}(M'_m)$ and $\ell \leq k$. Since $H \mathrel{\circ\!\!\!\rightarrow} H'$ and H is in hnf, we get $M_i \mathrel{\circ\!\!\!\rightarrow} M'_i$ for every $i = 1, \ldots, m$. Co-recursively applying the same argument to $M_i \mathrel{\circ\!\!\!\rightarrow} M'_i$ yields $\mathsf{BT}_{\circledcirc}(M) \geq \mathsf{BT}_{\circledcirc}(N)$. ■

While $\mathsf{BT}_{\circledcirc}(M) \not\equiv \mathsf{BT}_{\circledcirc}(N)$ does not imply $M \neq_{\beta} N$, the following theorem allows us to use clocked Böhm trees for discriminating λ-terms:

Theorem 22. *Let M and N be λ-terms. If N cannot be improved globally by any reduct of M, then $M \neq_{\beta} N$.*

Proof: If $M =_{\beta} N$, then $M \twoheadrightarrow M' \twoheadleftarrow N$ for some M'. Hence $\mathsf{BT}_{\circledcirc}(M') \leq \mathsf{BT}_{\circledcirc}(N)$ by Proposition 21. ■

Note that for distinguishing M and N we can always consider β-equivalent terms $M' =_{\beta} M$ and $N' =_{\beta} N$ instead. For Theorem 22 we have to show $\neg(\mathsf{BT}_{\circledcirc}(M') \leq \mathsf{BT}_{\circledcirc}(N))$ for all reducts M' of M. This condition is in general difficult to prove. However, the theorem is of use if one of the terms has a manageable set of reducts, and this term happens to have slower clocks.

For a large class of λ-terms it turns out that clocks are invariant under reduction. We call these terms 'simple'.

Definition 23. 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.

The definition of simple redexes generalizes the well-known notions of call-by-value and linear redexes. Next, we define simple *terms*. Intuitively, we call a term M 'simple' if every reduction admitted by M only contracts simple redexes. The following definition further generalises this intuition by considering only reductions computing the Böhm tree:

Definition 24 (Simple terms)**.** A λ-term M is *simple* if either M has no hnf, or the head reduction to hnf $M \to_h^k \lambda x_1. \ldots \lambda x_n.y M_1 \ldots M_m$ contracts only simple redexes, and $M_1, \ldots, M_m$ are simple terms.

All the fpc's in this paper are either simple or have simple reducts. The clock of simple λ-terms is invariant under reduction, that is, when ignoring finite prefixes of the clocked Böhm trees (by reducing a term we can always make the clock values in a finite prefix equal to 0).

Proposition 25. *Let N be a reduct of a simple term M. Then N matches M eventually* ($\mathsf{BT}_{\circledcirc}(M) =_{\exists} \mathsf{BT}_{\circledcirc}(N)$).

Proof: The proof is a straightforward extension of the proof of Proposition 21 with the observation that for simple terms M, rewriting $M \to_h^k H \equiv \lambda x_1. \ldots \lambda x_n.y M_1 \ldots M_m$ to hnf does not duplicate redexes. Hence, the elementary diagrams are now of the form displayed in Figure 5.

$$\begin{tikzcd}
M \arrow[r, "h"'] \arrow[d] & M_1 \arrow[d, "\varnothing"] \\
M_2 \arrow[r, "\varnothing"] & M'
\end{tikzcd}
\qquad
\begin{tikzcd}
M \arrow[r, "h"'] \arrow[d] & M_1 \arrow[d] \\
M_2 \arrow[r, "h"'] & M'
\end{tikzcd}$$

Fig. 5. Elementary diagrams for simple M.

That is, whenever we have co-initial steps $M \to_h M_1$ and $M \to M_2$ and M is a simple term, then either the steps cancel each other out $M_1 \equiv M_2$ (if both are the same step), or they can be joined by single steps $M_1 \to_h M' \leftarrow M_2$.

As a consequence, when projecting $M \to_h M'$ over a rewrite sequence $M \to^n N$, then either $M' \to^n M'' \leftarrow_h N$ or there has been cancellation and $M' \to^{n-1} M'' \equiv N$. Every cancellation decreases the number of steps $M' \to^{n-1} M''$, and hence there can only be finitely many cancellations. This implies the claim that $\mathsf{BT}_{\circledcirc}(M)$ is equal to $\mathsf{BT}_{\circledcirc}(N)$ modulo a finite prefix, that is, $\mathsf{BT}_{\circledcirc}(M) =_{\exists} \mathsf{BT}_{\circledcirc}(N)$. ■

Reduction accelerates clocks, and for simple terms the clock is invariant under reduction, see Proposition 25. Hence if a term M has a simple reduct N, then N has the fastest clock reachable from M modulo a finite prefix. This justifies the following convention.

Convention 26. The *(minimal) clock* of a λ-term M with a simple reduct N is $\mathsf{BT}_{\circlearrowright}(N)$, the clocked BT of N.

For simple terms we obtain the following theorem:

Theorem 27. *Let M and N be λ-terms such that M is simple. If M does not improve eventually on N, then $M \neq_\beta N$.*

Proof: Assume $M =_\beta N$. Then $M \twoheadrightarrow M' \twoheadleftarrow N$ for some M'. We have $\mathsf{BT}_{\circlearrowright}(M) =_\exists \mathsf{BT}_{\circlearrowright}(M')$ by Proposition 25, and $\mathsf{BT}_{\circlearrowright}(M') \leq \mathsf{BT}_{\circlearrowright}(N)$ by Proposition 21. Hence we obtain $\mathsf{BT}_{\circlearrowright}(M) \leq_\exists \mathsf{BT}_{\circlearrowright}(N)$. ■

Theorem 27 significantly reduces the proof obligation in comparison to Theorem 22. We only consider the clocked BT's of M and N, instead of all reducts M' of M.

Note that Theorem 27 can also be employed for discriminating non-simple λ-terms if one of the terms has a simple reduct. For the case that both M and N are simple, there is no need to look for reducts since the eventual clocks are invariant under reduction, see Proposition 25:

Corollary 28. *If simple terms M, N do not match each other eventually, then they are β-inconvertible.*

Proof: Assume $M = N$, then $M \twoheadrightarrow O \twoheadleftarrow N$ for a common reduct O. Then $\mathsf{BT}_{\circlearrowright}(M) =_\exists \mathsf{BT}_{\circlearrowright}(O) =_\exists \mathsf{BT}_{\circlearrowright}(N)$ by Proposition 25. Hence $\mathsf{BT}_{\circlearrowright}(M) =_\exists \mathsf{BT}_{\circlearrowright}(N)$ which contradicts $\neg(\mathsf{BT}_{\circlearrowright}(M) =_\exists \mathsf{BT}_{\circlearrowright}(N))$. ■

Remark 29. The reason for the qualifier 'eventually' in the notions above, in other words, working modulo a finite prefix of the BT, is that by some preliminary reduction we can always make the clock values in any finite prefix equal to 0. So we are interested exclusively in the 'tail behaviour', or the behaviour 'at infinity', and not in the initial behaviour of the development to the BT.

To give a concrete example: Y_0 and Y_1, the fpc's of Curry and Turing, can be reduced to reducts M_0, M_1 respectively, that have an initial segment of arbitrary length n of their BT's with clock labels 0 (just reduce first to $\lambda f.f^n(\ldots)$). However, the infinite remainders of their BT's, their tails as it were, will reveal the difference in clock values, witnessing the fact that Y_0 eventually improves Y_1. And this situation is stable under reduction; indeed, for any two reducts M_0, M_1 as above, the first eventually improves the second.

Remark 30. Take λ-terms M, N with finite BT's. Then the clock comparison of M and N amounts to the comparison of their non-clocked BT's. In case their BT's are equal and $\bot$-free, then $M =_\beta N$. If their BT's coincide but are not $\bot$-free, then we can fine-tune the analysis by comparing their clocked Lévy–Longo or Berarducci trees (a $\bot$ in a BT may give rise to an infinite Berarducci tree), see Section VIII.

Example 31. Let $n \geq 2$. We compute the clocks of the fpc's Y_n of the Böhm sequence. We first reduce $\mathsf{Y}_n = \mathsf{Y}_0\delta^{\sim n}$ with $\mathsf{Y}_0 = \lambda f.\omega_f\omega_f$ and $\omega_f = \lambda x.f(xx)$ to a simple term:

$$\mathsf{Y}_n x \to \omega_\delta\omega_\delta\delta^{\sim n-1}x \twoheadrightarrow \eta\eta\delta^{\sim n-1}x$$

where $\eta = \lambda ab.b(aab)$. We compute the clock:

$$\begin{aligned} \eta\eta\delta^{\sim n-1}x &\to_h^2 \delta(\eta\eta\delta)\delta^{\sim n-2}x \\ &\to_h^{2(n-2)} \delta(\eta\eta\delta^{\sim n-1})x \\ &\to_h^2 x(\eta\eta\delta^{\sim n-1}x) \end{aligned}$$

We find $\mathsf{BT}_{\circlearrowright}(\eta\eta\delta^{\sim n-1}x) = [2n](x\mathsf{BT}_{\circlearrowright}(\eta\eta\delta^{\sim n-1}x))$. Hence, for $n \geq 2$ the clock of Y_n is $2n$.

By Theorem 27, Example 31 and Figure 3 we obtain an alternative proof for Theorem 7: the Böhm sequence contains no duplicates.

Example 32. Let $n \geq 2$. We compute the clocks of the fpc's $\mathsf{U}_n = \mathsf{B}\mathsf{Y}_0\mathsf{S}^{\sim n}\mathsf{I}$ of the Scott sequence. We first reduce U_n to a simple term:

$$\begin{aligned} \mathsf{U}_n x &\twoheadrightarrow \mathsf{Y}_0(\mathsf{SS})\mathsf{S}^{\sim(n-2)}\mathsf{I}x \\ &\to \omega_{\mathsf{SS}}\,\omega_{\mathsf{SS}}\mathsf{S}^{\sim(n-2)}\mathsf{I}x \\ &\twoheadrightarrow \underline{\omega_{\mathsf{SS}}}\,\underline{\omega_{\mathsf{SS}}}\mathsf{S}^{\sim(n-2)}\mathsf{I}x \end{aligned}$$

where $\underline{\omega_{\mathsf{SS}}} \equiv \lambda abc.bc(aabc)$. We abbreviate $\theta = \underline{\omega_{\mathsf{SS}}}$. Then we compute the clocks for $n = 2$, $n = 3$, and $n > 3$:

$$\begin{aligned} \theta\theta\mathsf{I}x &\to_h^3 \mathsf{I}x(\theta\theta\mathsf{I}x) \to_h^1 x(\theta\theta\mathsf{I}x) \\ \theta\theta\mathsf{SI}x &\to_h^3 \mathsf{SI}(\theta\theta\mathsf{SI})x \to_h^4 x(\theta\theta\mathsf{SI}x) \\ \theta\theta\mathsf{S}^{\sim(n-2)}\mathsf{I}x &\to_h^3 \mathsf{SS}(\theta\theta\mathsf{SS})\mathsf{S}^{\sim(n-4)}\mathsf{I}x \\ &\to_h^{3(n-4)} \mathsf{SS}(\theta\theta\mathsf{SSS}^{\sim(n-4)})\mathsf{I}x \\ &\to_h^3 \mathsf{SI}(\theta\theta\mathsf{S}^{\sim(n-2)}\mathsf{I})x \\ &\to_h^4 x(\theta\theta\mathsf{S}^{\sim(n-2)}\mathsf{I}x) \end{aligned}$$

respectively. For all three cases, we find:
$\mathsf{BT}_{\circlearrowright}(\theta\theta\mathsf{S}^{\sim(n-2)}\mathsf{I}x) = [3n-2](x\mathsf{BT}_{\circlearrowright}(\theta\theta\mathsf{S}^{\sim(n-2)}\mathsf{I}x))$.

Using Theorem 27 we infer from Example 32 and Figure 3 (recall that $\mathsf{U}_0 =_\beta \mathsf{Y}_0$ and $\mathsf{U}_1 =_\beta \mathsf{Y}_1$):

Corollary 33. *The Scott sequence contains no duplicates.*

Plotkin [16] asked: Is there an fpc Y such that

$$A_Y \equiv Y(\lambda z.fzz) =_\beta Y(\lambda x.Y(\lambda y.fxy)) \equiv B_Y$$

or in other notation: $\mu z.fzz =_\beta \mu x.\mu y.fxy$, with the definition $\mu x.M(x) = Y(\lambda x.M(x))$. The terms A_Y and B_Y have the same Böhm tree, namely the solution of $T = fTT$.

The terms A_Y and B_Y are not simple. An extension of our clock method can be given which restricts the clock comparison to single paths in the clocked Böhm tree along which there is no duplication of redexes. We leave this extension to future work. Using this extension would allow us to settle the question in the negative for all simple fpc's.

For Turing's fpc Y_1 this is seen by computing the clocked BT's of A_{Y_1} and B_{Y_1}. Recall $\mathsf{Y}_1 \equiv \eta\eta$ with $\eta \equiv \lambda xf.f(xxf)$.

$$\begin{aligned} A_{\mathsf{Y}_1} &\equiv \eta\eta(\lambda z.fzz) \to_h^2 (\lambda z.fzz)A_{\mathsf{Y}_1} \to_h^1 fA_{\mathsf{Y}_1}A_{\mathsf{Y}_1} \\ B_{\mathsf{Y}_1} &\equiv \eta\eta(\lambda x.\eta\eta(\lambda y.fxy)) \to_h^2 (\lambda x.\eta\eta(\lambda y.fxy))B_{\mathsf{Y}_1} \\ &\to_h^1 \eta\eta(\lambda y.fB_{\mathsf{Y}_1}y) \to_h^2 (\lambda y.fB_{\mathsf{Y}_1}y)(\eta\eta(\lambda y.fB_{\mathsf{Y}_1}y)) \\ &\to_h^1 fB_{\mathsf{Y}_1}(\eta\eta(\lambda y.fB_{\mathsf{Y}_1}y)) \end{aligned}$$

Note that for B_{Y_1} developing the left branch takes six steps, whereas the right only needs three. The clocked BT's for A_{Y_1} and B_{Y_1} are depicted in Figure 6 using hnf-notation (see [2] or [3]).

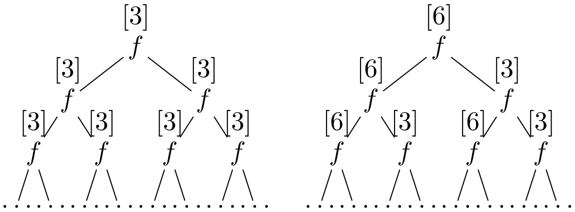

Fig. 6. Clocked BT's for A_{Y_1} and B_{Y_1}.

We conjecture that for no fpc Y, $A_Y =_\beta B_Y$; maybe this requires an extension of the proof in [12].

VII. Atomic Clocks

We have introduced clocked Böhm trees for discriminating λ-terms. In this section, we refine the clocks to measure not only the *number* of head steps, but, in addition, the *position* of each of these steps. We call these clocks 'atomic'.

Let us consider a motivating example. We discriminate Y_2 and U_2. First, we reduce both terms to simple reducts:

$$\begin{aligned} \mathsf{Y}_2x &\equiv \mathsf{Y}_0\delta\delta x \twoheadrightarrow \eta\eta\delta x && \text{where } \eta = \lambda ab.b(aab) \\ \mathsf{U}_2x &\equiv \mathsf{Y}_0(\mathsf{SS})\mathsf{I}x \twoheadrightarrow \theta\theta\mathsf{I}x && \text{where } \theta = \lambda abc.bc(aabc) \end{aligned}$$

We compute the atomic clocks of these simple reducts:

$$\begin{aligned} \eta\eta\delta x &\to_{h,11} (\lambda b.b(\eta\eta b))\delta x \to_{h,1} \delta(\eta\eta\delta)x \\ &\to_{h,1} (\lambda b.b(\eta\eta\delta b))x \to_{h,\epsilon} x(\eta\eta\delta x) \\ \theta\theta\mathsf{I}x &\to_{h,11} (\lambda bc.bc(\theta\theta bc))\mathsf{I}x \to_{h,1} (\lambda c.\mathsf{I}c(\theta\theta\mathsf{I}c))x \\ &\to_{h,\epsilon} \mathsf{I}x(\theta\theta\mathsf{I}x) \to_{h,1} x(\theta\theta\mathsf{I}x) \end{aligned}$$

Both terms have the clocked Böhm tree $T \equiv [4](xT)$. Thus the method from the previous section is not applicable.

However, with atomic clocks we have:

$$\begin{aligned} \mathsf{BT}_{\text{☢}}(\eta\eta\delta x) &= [11, 1, 1, \epsilon](x\mathsf{BT}_{\text{☢}}(\eta\eta\delta x)) \\ \mathsf{BT}_{\text{☢}}(\theta\theta\mathsf{I}x) &= [11, 1, \epsilon, 1](x\mathsf{BT}_{\text{☢}}(\theta\theta\mathsf{I}x)) \end{aligned}$$

which allows us to discriminate the terms. Hence $\mathsf{Y}_2 \neq_\beta \mathsf{U}_2$ (by Corollary 28 which generalises to the setting of atomic BT's). Note that the (non-atomic) clocked BT's can be obtained by taking the length of the lists of positions.

For lists $\vec{p}, \vec{q}$ of positions, we write $\vec{p} \cdot \vec{q}$ for concatenating $\vec{p}$ to $\vec{q}$. We write $\to_{h,\langle p_1,\ldots,p_n\rangle}$ for the rewrite sequence $\to_{h,p_1} \cdots \to_{h,p_n}$ consisting of steps at position p_1,…,p_n.

Definition 34 (Atomic clock Böhm trees)**.** Let $M \in \mathit{Ter}(\lambda)$. The *atomic clock Böhm tree* $\mathsf{BT}_{\text{☢}}(M)$ *of* M is an annotated infinite term defined as follows. If M has no hnf, then define $\mathsf{BT}_{\text{☢}}(M)$ as $\perp$. Otherwise, there is a head reduction

$$M \to_{h,p_1} \cdots \to_{h,p_k} \lambda x_1.\ldots\lambda x_n.yM_1\ldots M_m$$

of length k to hnf. Then we define $\mathsf{BT}_{\text{☢}}(M)$ as the term $[\langle p_1,\ldots,p_k\rangle]\lambda x_1.\ldots\lambda x_n.y\mathsf{BT}_{\text{☢}}(M_1)\ldots\mathsf{BT}_{\text{☢}}(M_m)$.

The theory developed for (non-atomic) BT's generalises as follows to the atomic trees. For lists of positions $\vec{p}, \vec{q}$ we define $\vec{p} \geq \vec{q}$ whenever $\vec{q}$ is a subsequence of $\vec{p}$, and $\vec{p} > \vec{q}$ if additionally $\vec{p} \neq \vec{q}$. Here $\langle a_1,\ldots,a_n\rangle$ is a subsequence of $\langle b_1,\ldots,b_m\rangle$ if there exist indexes $i_1 < i_2 < \ldots < i_n$ such that $\langle a_1,\ldots,a_n\rangle = \langle b_{i_1},\ldots,b_{i_n}\rangle$.

Using this notation for comparing the atomic annotations (lists of positions), Proposition 21, Theorem 22, Proposition 25, Theorem 27, and Corollary 28 remain valid (using basically the same proofs).

Proposition 35. *Let* $\mathsf{G}_n = \square(\mathsf{SS})\mathsf{S}^{\sim n}\mathsf{I}$ *the fpc-generating vectors from Theorem 13. For* $n_1,\ldots,n_k \in \mathbb{N}$ *we define* $\mathsf{Y}^{\langle n_1,\ldots,n_k\rangle} = \mathsf{Y}_0\mathsf{G}_{n_1}\ldots\mathsf{G}_{n_k}$. *We prove that all these fpc's are inconvertible, that is,* $\vec{n} \neq \vec{m}$ *implies* $\mathsf{Y}^{\vec{n}} \neq_\beta \mathsf{Y}^{\vec{m}}$.

This proposition cannot be proved using (non-atomic) clocks, as for example: $\mathsf{BT}_{\text{⏲}}(\mathsf{Y}^{\langle n,\,m\rangle}) = \mathsf{BT}_{\text{⏲}}(\mathsf{Y}^{\langle m,\,n\rangle})$. We introduce some auxiliary notations. Let $\mathsf{G}'_n = \square\mathsf{S}^{\sim n}\mathsf{I}$, and define $\underline{\mathsf{G}_n} = \square\underline{(\mathsf{SS})}\mathsf{S}^{\sim n}\mathsf{I}$ where $\underline{\mathsf{SS}} = \lambda abc.bc(abc)$. For $\vec{p} = \langle 1^{m_1},\ldots,1^{m_k}\rangle$ a list of positions define $\vec{p} \times n = \vec{p} \cdot (\langle 1^{m_1-1},\ldots,1^{m_k-1}\rangle \times (n-1))$ for $n > 1$ and $\vec{p} \times 1 = \vec{p}$.

Proof: Let $n_1,\ldots,n_k \in \mathbb{N}$ and $\mathsf{Y} = \theta\theta\mathsf{G}'_{n_1}\underline{\mathsf{G}_{n_2}}\ldots\underline{\mathsf{G}_{n_k}}$ where $\theta = \lambda abc.bc(aabc)$, then $\mathsf{Y}^{\langle n_1,\ldots,n_k\rangle}x \twoheadrightarrow \mathsf{Y}x$ (for $k \geq 1$) where $\mathsf{Y}x$ is a simple term. Apart from the initial and final steps, the rewrite sequence $\mathsf{Y}x \twoheadrightarrow_h x(\mathsf{Y}x)$ is composed of k subsequences of the form:

$$\begin{aligned} \mathsf{SI}(\theta\theta\ldots)\underline{\mathsf{G}_n}V_m &\equiv \mathsf{SI}(\theta\theta\ldots)\underline{(\mathsf{SS})}\mathsf{S}^{\sim n}\mathsf{I}V_m \\ &\to_{h,1^{n+m+3}\times 3} \mathsf{I}\underline{(\mathsf{SS})}(\theta\theta\ldots\underline{(\mathsf{SS})})\mathsf{S}^{\sim n}\mathsf{I}V_m \\ &\to_{h,1^{n+m+2}} \underline{\mathsf{SS}}(\theta\theta\ldots\underline{(\mathsf{SS})})\mathsf{S}^{\sim n}\mathsf{I}V_m \\ &\to_{h,1^{n+m+1}\times 3} \mathsf{SS}(\theta\theta\ldots\underline{(\mathsf{SS})})\mathsf{S}^{\sim n-2}\mathsf{I}V_m \\ &\to_{h,(1^{n+m}\times 3)\times(n-1)} \mathsf{SI}(\theta\theta\ldots\underline{\mathsf{G}_n})V_m \end{aligned}$$

for every G_n with $n \geq 2$, and vector V_m of length m.

For every G_n there is exactly one occurrence of four consecutive steps at 'decreasing' positions 1^{n+m+2}, 1^{n+m+1}, 1^{n+m}, 1^{n+m-1} (btw, this also holds for $n < 2$). Hence, from the distance between these occurrences we can reconstruct the vector $\langle n_1,\ldots,n_k\rangle$. This shows that $\vec{n} \neq \vec{m}$ implies that $\mathsf{BT}_{\text{☢}}(\mathsf{Y}^{\vec{n}}) =_\exists \mathsf{BT}_{\text{☢}}(\mathsf{Y}^{\vec{m}})$ is false, and hence we conclude $\mathsf{Y}^{\vec{n}} \neq_\beta \mathsf{Y}^{\vec{m}}$ by Corollary 28. ■

VIII. Clocked Lévy–Longo and Berarducci Trees

In fact, there are three main semantics for the λ-calculus: BT, LLT, and BeT (see [1,3,4,6,13]). The notions from the previous section generalize directly to LLT and BeT semantics.

Definition 36 (Clocked Lévy–Longo trees)**.** Let M be a λ-term. The *clocked Lévy–Longo tree* $\mathsf{LLT}_{\circledcirc}(M)$ *of* M is an annotated potentially infinite term defined as follows. If M has no whnf, then define $\mathsf{LLT}_{\circledcirc}(M)$ as $\bot$. Otherwise, there exists a head rewrite sequence $M \to_h^k \lambda x.N$ or $M \to_h^k xM_1 \ldots M_m$ to whnf. In this case, we define $\mathsf{LLT}_{\circledcirc}(M)$ as $[k]\lambda x.\mathsf{LLT}_{\circledcirc}(N)$ or $[k]x\mathsf{LLT}_{\circledcirc}(M_1) \ldots \mathsf{LLT}_{\circledcirc}(M_m)$, respectively.

Definition 37 (Clocked Berarducci trees)**.** Let M be a λ-term. The *clocked Berarducci tree* $\mathsf{BeT}_{\circledcirc}(M)$ *of* M is an annotated potentially infinite term defined as follows. If M is root-active, let $\mathsf{BeT}_{\circledcirc}(M) \equiv \bot$. If $M \to_h^k N$ rewrites to a root-stable term $N \equiv x$, $N \equiv \lambda x.P$ or $N \equiv PQ$, then define $\mathsf{BeT}_{\circledcirc}(M)$ as $[k]x$, $[k]\lambda x.\mathsf{BeT}_{\circledcirc}(P)$ or $[k]\mathsf{BeT}_{\circledcirc}(P)\mathsf{BeT}_{\circledcirc}(Q)$, respectively.

Example 38. We consider the terms $M = PP$ with $P = \lambda x.\lambda y.xx$ and $N = QQ$ with $Q = \lambda x.\lambda y.\lambda z.xx$. We have:

$$\mathsf{LLT}_{\circledcirc}(M) \equiv [1]\lambda y.\mathsf{LLT}_{\circledcirc}(M)$$
$$\mathsf{LLT}_{\circledcirc}(N) \equiv [1]\lambda y.[0]\lambda z.\mathsf{LLT}_{\circledcirc}(N)$$

Thus, in $\mathsf{LLT}_{\circledcirc}(M)$ every λ requires one head reduction step whereas in $\mathsf{LLT}_{\circledcirc}(N)$ every second λ is obtained for 'free' (that is, in 0 steps).

We remark that M and N cannot be distinguished in the Böhm tree semantics since $\mathsf{BT}_{\circledcirc}(M) \equiv \mathsf{BT}_{\circledcirc}(N) \equiv \bot$. In the Böhm tree semantics, a term is meaningful only if it has a hnf. The Lévy–Longo semantics weakens this condition to whnf's, and thereby allows more terms to be distinguished. The Berarducci tree semantics is a further weakening where only root-active terms are discarded as meaningless.

IX. Concluding remarks

We conclude with an encompassing conjecture.

Conjecture 39. *Building fpc's with fcp-generating vectors is a free construction, that is, there are no non-trivial identifications.*

A first step is found in Intrigila's theorem $Y\delta \neq_\beta Y$, for any fpc Y. A second step is that the Böhm sequence is duplicate-free. A third step is found in our proof that the Scott sequence is duplicate-free, and Proposition 35, which states that there are no identifications when starting the construction with Y_0.

Other parts of the conjecture are as follows. Let Y, Y' be fpc's and $B_1 \ldots B_n, C_1 \ldots C_k$ be fpc-generating vectors.

(i) $Y\delta =_\beta Y'\delta$ iff $Y =_\beta Y'$.

(ii) $YB_1 \ldots B_n =_\beta Y'B_1 \ldots B_n$ iff $Y = Y'$.

(iii) $YB_1 \ldots B_n \neq_\beta YC_1 \ldots C_k$ if $B_1 \ldots B_n \not\equiv C_1 \ldots C_k$.

For general fpc's Y, Y' these conjectures may be beyond current techniques, but for the well-known fpc's of Curry and Turing, and the fpc-generating vectors introduced here, including their versions for $n > 3$, these problems are tractable.

References

[1] S. Abramsky and C.-H. L. Ong, "Full Abstraction in the Lazy Lambda Calculus," *Inf. Comput.*, vol. 105, no. 2, pp. 159–267, 1993.

[2] H. P. Barendregt, *The Lambda Calculus. Its Syntax and Semantics*, revised ed., ser. Studies in Logic and The Foundations of Mathematics. North-Holland, 1984, vol. 103.

[3] H. P. Barendregt and J. W. Klop, "Applications of Infinitary Lambda Calculus," *Inf. Comput.*, vol. 207, no. 5, pp. 559–582, 2009.

[4] A. Berarducci and B. Intrigila, "Church–Rosser λ-theories, Infinite λ-calculus and Consistency Problems," *Logic: From Foundations to Applications*, pp. 33–58, 1996.

[5] I. Bethke, "Lamda Calculus," Chapter 10 in [21].

[6] I. Bethke, J. W. Klop, and R. C. de Vrijer, "Descendants and Origins in Term Rewriting," *Inf. Comput.*, vol. 159, no. 1–2, pp. 59–124, 2000.

[7] C. Böhm, "The CUCH as a Formal and Description Language," *Annual Review in Automatic Programming*, vol. 3, pp. 179–197, 1963.

[8] T. Coquand and H. Herbelin, "*A*-Translation and Looping Combinators in Pure Type Systems," *J. Funct. Program.*, vol. 4, no. 1, pp. 77–88, 1994.

[9] M. Dezani-Ciancaglini, P. Severi, and F. J. de Vries, "Infinitary Lambda Calculus and Discrimination of Berarducci Trees," *Theor. Comput. Sci.*, vol. 2, no. 298, pp. 275–302, 2003.

[10] H. Geuvers and B. Werner, "On the Church–Rosser Property for Expressive Type Systems and its Consequences for their Metatheoretic Study," in *LICS*, 1994, pp. 320–329.

[11] M. Goldberg, "Constructing Fixed-Point Combinators Using Application Survival," Dept. of Computer Science, University of Aarhus, Tech. Rep. BRICS RS-95-35, 1995.

[12] B. Intrigila, "Non-Existent Statman's Double Fixed Point Combinator Does Not Exist, Indeed," *Inf. Comput.*, vol. 137, no. 1, pp. 35–40, 1997.

[13] R. Kennaway and F.-J. de Vries, "Infinitary Rewriting," Chapter 12 in [21].

[14] R. Kennaway, J. W. Klop, M. R. Sleep, and F.-J. de Vries, "Infinitary Lambda Calculus," *Theor. Comput. Sci.*, vol. 175, no. 1, pp. 93–125, 1997.

[15] 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*, 2007, pp. 197–210, online version: `http://www.cs.ru.nl/barendregt60`.

[16] G. Plotkin, 2007, personal communication at the symposium for H. Barendregt's 60th birthday.

[17] G. Plotkin, "A Semantics for Type Checking," in *TACS*, ser. LNCS, vol. 526. Springer, 1991, pp. 1–17.

[18] D. S. Scott, "Some Philosophical Issues Concerning Theories of Combinators," in *Lambda Calculus and Computer Science Theory*, ser. LNCS, C. Böhm, Ed., vol. 37, 1975, pp. 346–366.

[19] R. Smullyan, *To Mock a Mockingbird, and Other Logic Puzzles*. Alfred A. Knopf, New York, 1985.

[20] R. Statman, "The Word problem for Smullyan's Lark Combinator is Decidable," *Journal of Symbolic Computation*, vol. 7, pp. 103–112, 1989.

[21] Terese, *Term Rewriting Systems*, ser. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2003, vol. 55.