Logical Methods in Computer Science
Volume 16, Issue 3, 2020, pp. 6:1–6:26
https://lmcs.episciences.org/

Submitted Sep. 11, 2018
Published Jul. 15, 2020

REVISITING CALL-BY-VALUE BÖHM TREES IN LIGHT OF THEIR TAYLOR EXPANSION

EMMA KERINEC, GIULIO MANZONETTO, AND MICHELE PAGANI

Université de Lyon, ENS de Lyon, Université Claude Bernard Lyon 1, LIP
e-mail address: emma.kerinec@ens-lyon.fr

LIPN, UMR 7030, Université Paris 13, Sorbonne Paris Cité, F-93430, Villetaneuse, France
e-mail address: giulio.manzonetto@univ-paris13.fr

CNRS, IRIF, Université de Paris, F-75205 Paris, France
e-mail address: pagani@irif.fr

Abstract. The call-by-value λ-calculus can be endowed with permutation rules, arising from linear logic proof-nets, having the advantage of unblocking some redexes that otherwise get stuck during the reduction. We show that such an extension allows to define a satisfying notion of Böhm(-like) tree and a theory of program approximation in the call-by-value setting. We prove that all λ-terms having the same Böhm tree are observationally equivalent, and characterize those Böhm-like trees arising as actual Böhm trees of λ-terms.

We also compare this approach with Ehrhard's theory of program approximation based on the Taylor expansion of λ-terms, translating each λ-term into a possibly infinite set of so-called resource terms. We provide sufficient and necessary conditions for a set of resource terms in order to be the Taylor expansion of a λ-term. Finally, we show that the normal form of the Taylor expansion of a λ-term can be computed by performing a normalized Taylor expansion of its Böhm tree. From this it follows that two λ-terms have the same Böhm tree if and only if the normal forms of their Taylor expansions coincide.

We are honoured to dedicate this article to Corrado Böhm, whose brilliant pioneering work has been an inspiration to us all.

Introduction

In 1968, Corrado Böhm published a separability theorem – known as *the Böhm Theorem* – which is nowadays universally recognized as a fundamental theorem in λ-calculus [Bö68]. Inspired by this result, Barendregt in 1977 proposed the definition of "Böhm tree of a λ-term" [Bar77], a notion which played for decades a prominent role in the theory of program approximation. The Böhm tree of a λ-term M represents the evaluation of M as a possibly infinite labelled tree coinductively, but effectively, constructed by collecting the stable amounts of information coming out of the computation. Equating all λ-terms

Key words and phrases: Lambda calculus, call-by-value, Böhm trees, differential linear logic, Taylor expansion, program approximation.

DOI:10.23638/LMCS-16(3:6)2020

having the same Böhm tree is a necessary, although non sufficient, step in the quest for fully abstract models of λ-calculus.

In 2003, Ehrhard and Regnier, motivated by insights from Linear Logic, introduced the notion of "Taylor expansion of a λ-term" as an alternative way of approximating λ-terms [ER03]. The Taylor expansion translates a λ-term M as a possibly infinite set[1] of multi-linear terms, each approximating a finite part of the behaviour of M. These terms populate a *resource calculus* [Tra09] where λ-calculus application is replaced by the application of a term to a bag of resources that cannot be erased, or duplicated and must be consumed during the reduction. The advantage of the Taylor expansion is that it exposes the amount of resources needed by a λ-term to produce (a finite part of) a value, a quantitative information that does not appear in its Böhm tree. The relationship between these two notions of program approximation has been investigated in [ER06], where the authors show that the Taylor expansion can actually be seen as a resource sensitive version of Böhm trees by demonstrating that the normal form of the Taylor expansion of M is actually equal to the Taylor expansion of its Böhm tree.

The notions of Böhm tree and Taylor expansion have been first developed in the setting of call-by-name (CbN) λ-calculus [Bar84]. However many modern functional programming languages, like OCaml, adopt a call-by-value (CbV) reduction strategy — a redex of shape $(\lambda x.M)N$ is only contracted when N is a *value*, namely a variable or a λ-abstraction. The call-by-value λ-calculus λ_v has been defined by Plotkin in 1975 [Plo75], but its theory of program approximation is still unsatisfactory and constitutes an ongoing line of research [Ehr12, CG14, MRP19]. For instance, it is unclear what should be the Böhm tree of a λ-term because of the possible presence of β-redexes that get stuck (waiting for a value) in the reduction. A paradigmatic example of this situation is the λ-term $M = (\lambda y.\Delta)(xx)\Delta$, where $\Delta = \lambda z.zz$ (see [PRDR99, AG17]). This term is a call-by-value normal form because the argument xx, which is not a value, blocks the evaluation (while one would expect M to behave as the divergent term $\Omega = \Delta\Delta$). A significant advance in reducing the number of stuck redexes has been made in [CG14] where Carraro and Guerrieri, inspired by Regnier's work in the call-by-name setting [Reg94], introduce permutations rules (σ) naturally arising from the translation of λ-terms into Linear Logic proof-nets. Using σ-rules, the λ-term M above rewrites in $(\lambda y.\Delta\Delta)(xx)$ which in its turn rewrites to itself, thus giving rise to an infinite reduction sequence, as desired. In [GPR17], Guerrieri *et al.* show that this extended calculus λ_v^σ still enjoys nice properties like confluence and standardization, and that adding the σ-rules preserves the operational semantics of Plotkin's CbV λ-calculus as well as the observational equivalence.

In the present paper we show that σ-rules actually open the way to provide a meaningful notion of call-by-value Böhm trees (Definition 2.8). Rather than giving a coinductive definition, which turns out to be more complicated than expected, we follow [AC98] and provide an appropriate notion of *approximants*, namely λ-terms possibly containing a constant $\bot$, that are in normal form w.r.t. the reduction rules of λ_v^σ (*i.e.*, the σ-rules and the restriction of (β) to values). In this context, $\bot$ represents the undefined value and this intuition is reflected in the definition of a preorder $\sqsubseteq$ between approximants which is generated by

[1] In its original definition, the Taylor expansion is a power series of multi-linear terms taking coefficients in the semiring of non-negative rational numbers. Following [MP11, Ehr12, BHP13], in this paper we abuse language and call "Taylor expansion" the *support* (underlying set) of the actual Taylor expansion. This is done for good reasons, as we are interested in the usual observational equivalences between λ-terms that overlook such coefficients.

$\perp \sqsubseteq V$, for all approximated values V. The next step is to associate with every λ-term M the set $\mathcal{A}(M)$ of its approximants and verify that they enjoy the following properties: (*i*) the "external shape" of an approximant of M is stable under reduction (Lemma 2.5); (*ii*) two interconvertible λ-terms share the same set of approximants (*cf.*, Lemma 2.6); (*iii*) the set of approximants of M is directed (Lemma 2.7). Once this preliminary work is accomplished, it is possible to define the Böhm tree of M as the supremum of $\mathcal{A}(M)$, the result being a possibly infinite labelled tree $\mathrm{BT}(M)$, as expected.

More generally, it is possible to define the notion of (CbV) "Böhm-like" trees as those labelled trees that can be obtained by arbitrary superpositions of (compatible) approximants. The Böhm-like trees corresponding to CbV Böhm trees of λ-terms have specific properties, that are due to the fact that λ-calculus constitutes a model of computation. Indeed, since every λ-term M is finite, $\mathrm{BT}(M)$ can only contain a finite number of free variables and, since M represents a program, the tree $\mathrm{BT}(M)$ must be computable. In Theorem 2.13 we demonstrate that these conditions are actually sufficient, thus providing a characterization.

To show that our notion of Böhm tree is actually meaningful, we prove that all λ-terms having the same Böhm tree are operationally indistinguishable (Theorem 4.16) and we investigate the relationship between Böhm trees and Taylor expansion in the call-by-value setting. Indeed, as explained by Ehrhard in [Ehr12], the CbV analogues of resource calculus and of Taylor expansion are unproblematic to define, because they are driven by solid intuitions coming from Linear Logic: rather than using the CbN translation $A \to B = !A \multimap B$ of intuitionistic arrow, it is enough to exploit Girard's so-called "boring" translation, which transforms $A \to B$ in $!(A \multimap B)$ and is suitable for CbV. Following [BHP13], we define a coherence relation $\frown$ between resource terms and prove that a set of such terms corresponds to the Taylor expansion of a λ-term if and only if it is an infinite clique having finite height. Subsequently, we focus on the dynamic aspects of the Taylor expansion by studying its *normal form*, that can always be calculated since the resource calculus enjoys strong normalization.

In [CG14], Carraro and Guerrieri propose to extend the CbV resource calculus with σ-rules to obtain a more refined normal form of the Taylor expansion $\mathscr{T}(M)$ of a λ-term M — this allows to mimic the σ-reductions occurring in M at the level of its resource approximants. Even with this shrewdness, it turns out that the normal form of $\mathscr{T}(M)$ is different from the normal form of $\mathscr{T}(\mathrm{BT}(M))$, the latter containing approximants that are not normal, but whose normal form is however empty (they disappear along the reduction). Although the result from [ER06] does not hold verbatim in CbV, we show that it is possible to define the *normalized Taylor expansion* $\mathscr{T}^\circ(-)$ *of a Böhm tree* and prove in Theorem 4.11 that the normal form of $\mathscr{T}(M)$ coincide with $\mathscr{T}^\circ(\mathrm{BT}(M))$, which is the main result of the paper. An interesting consequence, among others, is that all denotational models satisfying the Taylor expansion (*e.g.*, the one in [CG14]) equate all λ-terms having the same Böhm tree.

Related works. To our knowledge, in the literature no notion of CbV Böhm tree appears[2]. However, there have been attempts to develop syntactic bisimulation equivalences and theories of program approximation arising from denotational models. Lassen [Las05] coinductively defines a bisimulation equating all λ-terms having (recursively) the same "eager normal form", but he mentions that no obvious tree representations of the equivalence classes are at

[2] Even Paolini's separability result in [Pao01] for CbV λ-calculus does not rely on Böhm trees.

hand. In [RDRP04], Ronchi della Rocca and Paolini study a filter model of CbV λ-calculus and, in order to prove an Approximation Theorem, they need to define sets of *upper* and *lower* approximants of a λ-term. By admission of the authors [Roc18], these notions are not satisfactory because they correspond to an "over" (resp. "under") approximation of its behaviour.

We end this section by recalling that most of the results we prove in this paper are the CbV analogues of results well-known in CbN and contained in [Bar84, Ch. 10] (for Böhm trees), in [BHP13] (for Taylor expansion) and [ER06] (for the relationship between the two notions).

General notations. We denote by $\mathbb{N}$ the set of all natural numbers. Given a set X we denote by $\mathscr{P}(X)$ its powerset and by $\mathscr{P}_{\mathrm{f}}(X)$ the set of all finite subsets of X.

1. Call-By-Value λ-Calculus

The *call-by-value λ-calculus* λ_v, introduced by Plotkin in [Plo75], is a λ-calculus endowed with a reduction relation that allows the contraction of a redex $(\lambda x.M)N$ only when the argument N is a value, namely when N is a variable or an abstraction. In this section we briefly review its syntax and operational semantics. By extending its reduction with permutation rules σ, we obtain the calculus λ_v^σ introduced in [CG14], that will be our main subject of study.

1.1. Its syntax and operational semantics.

For the λ-calculus we mainly use the notions and notations from [Bar84]. We consider fixed a denumerable set $\mathbb{V}$ of variables.

Definition 1.1. The set Λ of *λ-terms* and the set Val of *values* are defined through the following grammars (where $x \in \mathbb{V}$):

$$\begin{array}{lrl} (\Lambda) & M,N,P,Q ::= & V \mid MN \\ (\mathrm{Val}) & U,V ::= & x \mid \lambda x.M \end{array}$$

As usual, we assume that application associates to the left and has higher precedence than λ-abstraction. For instance, $\lambda xyz.xyz = \lambda x.(\lambda y.(\lambda z.((xy)z)))$. Given $x_1,\dots,x_n \in \mathbb{V}$, we let $\lambda\vec{x}.M$ stand for $\lambda x_1 \dots \lambda x_n.M$. Finally, we write $MN^{\sim n}$ for $MN\cdots N$ (n times).

The set $\mathrm{FV}(M)$ of *free variables of M* and the *α-conversion* are defined as in [Bar84, §2.1]. A λ-term M is called *closed*, or *a combinator*, whenever $\mathrm{FV}(M) = \emptyset$. The set of all combinators is denoted by Λ^o. From now on, λ-terms are considered up to α-conversion, whence the symbol $=$ represents syntactic equality possibly up to renaming of bound variables.

Definition 1.2. Concerning specific combinators, we define:

$$\begin{array}{lllllllll} \mathtt{I} & = & \lambda x.x, & \Delta & = & \lambda x.xx, & \Omega & = & \Delta\Delta, \\ \mathtt{B} & = & \lambda fgx.f(gx), & \mathtt{K} & = & \lambda xy.x, & \mathtt{F} & = & \lambda xy.y, \\ \mathtt{Z} & = & \multicolumn{4}{l}{\lambda f.(\lambda y.f(\lambda z.yyz))(\lambda y.f(\lambda z.yyz)),} & \mathtt{K}^* & = & \mathtt{ZK}. \end{array}$$

where $\mathtt{I}$ is the identity, Ω is the paradigmatic looping combinator, $\mathtt{B}$ is the composition operator, $\mathtt{K}$ and $\mathtt{F}$ are the first and second projection (respectively), $\mathtt{Z}$ is Plotkin's recursion operator, and $\mathtt{K}^*$ is a λ-term producing an increasing amount of external abstractions.

Given $M, N \in \Lambda$ and $x \in \mathbb{V}$ we denote by $M[x := N]$ the λ-term obtained by substituting N for every free occurrence of x in M, subject to the usual proviso of renaming bound variables in M to avoid capture of free variables in N.

Remark 1.3. It is easy to check that the set Val is closed under substitution of values for free variables, namely $U, V \in \mathrm{Val}$ and $x \in \mathbb{V}$ entail $V[x := U] \in \mathrm{Val}$.

A context is a λ-term possibly containing occurrences of a distinguished algebraic variable, called *hole* and denoted by $(\!|-|\!)$. In the present paper we consider – without loss of generality for our purposes – contexts having a single occurrence of $(\!|-|\!)$.

Definition 1.4. A *(single-hole) context* $C(\!|-|\!)$ is generated by the simplified grammar:

$$C ::= (\!|-|\!) \mid CM \mid MC \mid \lambda x.C \qquad (\text{for } M \in \Lambda)$$

A context $C(\!|-|\!)$ is called a *head context* if it has shape $(\lambda x_1 \dots x_n.(\!|-|\!))V_1 \cdots V_m$ for $V_i \in \mathrm{Val}$.

Given $M \in \Lambda$, we write $C(\!|M|\!)$ for the λ-term obtained by replacing M for the hole $(\!|-|\!)$ in $C(\!|-|\!)$, possibly with capture of free variables.

We consider a CbV λ-calculus λ_v^σ endowed with the following notions of reductions. The β_v-reduction is the standard one, from [Plo75], while the σ-reductions have been introduced in [Reg94, CG14] and are inspired by the translation of λ-calculus into linear logic proof-nets.

Definition 1.5. The *β_v-reduction* $\to_{\beta_v}$ is the contextual closure of the following rule:

$$(\beta_v) \qquad (\lambda x.M)V \to M[x := V] \text{ whenever } V \in \mathrm{Val}$$

The *σ-reductions* $\to_{\sigma_1}$, $\to_{\sigma_3}$ are the contextual closures of the following rules (for $V \in \mathrm{Val}$):

$$\begin{array}{lll} (\sigma_1) & (\lambda x.M)NP \to (\lambda x.MP)N & \text{with } x \notin \mathrm{FV}(P) \\ (\sigma_3) & V((\lambda x.M)N) \to (\lambda x.VM)N & \text{with } x \notin \mathrm{FV}(V) \end{array}$$

We also set $\to_\sigma \, = \, \to_{\sigma_1} \cup \to_{\sigma_3}$ and $\to_{\mathsf{v}} \, = \, \to_{\beta_v} \cup \to_\sigma$.

The λ-term at the left side of the arrow in the rule (β_v) (resp. (σ_1), (σ_3)) is called β_v- (resp. σ_1-, σ_3-) *redex*, while the λ-term at the right side is the corresponding *contractum*. Notice that the condition for contracting a σ_1- (resp. σ_3-) redex can always be satisfied by performing appropriate α-conversions.

Each reduction relation $\to_{\mathsf{R}}$ generates the corresponding *multistep relation* $\twoheadrightarrow_{\mathsf{R}}$ by taking its transitive and reflexive closure, and *conversion relation* $=_{\mathsf{R}}$ by taking its transitive, reflexive and symmetric closure. Moreover, we say that a λ-term M is in R-*normal form* (R-*nf*, for short) if there is no $N \in \Lambda$ such that $M \to_{\mathsf{R}} N$. We say that M *has an* R-*normal form* whenever $M \twoheadrightarrow_{\mathsf{R}} N$ for some N in R-nf, and in this case we denote N by $\mathrm{nf}_{\mathsf{R}}(M)$.

Example 1.6.

(1) $\mathtt{I}x \to_{\beta_v} x$, while $\mathtt{I}(xy)$ is a v-normal form.
(2) $\Omega \to_{\beta_v} \Omega$, whence Ω is a looping combinator in the CbV setting as well.
(3) $\mathtt{I}(\Delta(xx))$ is a β_v-nf, but contains a σ_3-redex, indeed $\mathtt{I}(\Delta(xx)) \to_{\sigma_3} (\lambda z.\mathtt{I}(zz))(xx)$.
(4) For all values V, we have $\mathtt{Z}V =_{\mathsf{v}} V(\lambda x.\mathtt{Z}Vx)$ with $x \notin \mathrm{FV}(V)$. So we get:
(5) $\mathtt{K}^* = \mathtt{ZK} =_{\mathsf{v}} \mathtt{K}(\lambda y.\mathtt{K}^*y) =_{\mathsf{v}} \lambda x_0x_1.\mathtt{K}^*x_1 =_{\mathsf{v}} \lambda x_0x_1x_2.\mathtt{K}^*x_2 =_{\mathsf{v}} \cdots =_{\mathsf{v}} \lambda x_0 \dots x_n.\mathtt{K}^*x_n$.

(6) Let $\Xi = \mathtt{Z}N$ for $N = \lambda f.(\lambda y_1.f\mathtt{I})(zz)$, then we have:

$$\begin{array}{rcll} \Xi & =_{\mathsf{v}} & N(\lambda w.\Xi w) & \text{by } \mathtt{Z}V =_{\mathsf{v}} V(\lambda w.\mathtt{Z}Vw) \\ & =_{\mathsf{v}} & (\lambda y_1.(\lambda w.\Xi w)\mathtt{I})(zz) & \text{by } (\beta_v) \\ & =_{\mathsf{v}} & (\lambda y_1.\Xi\mathtt{I})(zz) & \text{by } (\beta_v) \\ & =_{\mathsf{v}} & (\lambda y_1.((\lambda y_2.\Xi\mathtt{I})(zz))\mathtt{I})(zz) & \text{by } (\beta_v) \\ & =_{\mathsf{v}} & (\lambda y_1.((\lambda y_2.\Xi\mathtt{I}\mathtt{I})(zz)))(zz) & \text{by } (\sigma_1) \\ & =_{\mathsf{v}} & (\lambda y_1.((\lambda y_2.((\lambda y_3.\Xi\mathtt{I}\mathtt{I}\mathtt{I})(zz)))(zz)))(zz) & = \cdots \end{array}$$

(7) $\mathtt{ZB} =_{\mathsf{v}} \mathtt{B}(\lambda z.\mathtt{ZB}z) =_{\mathsf{v}} \lambda gx.(\lambda z.\mathtt{ZB}z)(gx) =_{\mathsf{v}} \lambda gx.(\lambda fy.(\lambda z.\mathtt{ZB}z)(fy))(gx) =_{\mathsf{v}} \cdots$

The next lemma was already used implicitly in [GPR17].

Lemma 1.7. *A λ-term M is in v-normal form if and only if M is a G-term generated by the following grammar (for $k \geq 0$):*

$$\begin{array}{rcl} G & ::= & H \mid R \\ H & ::= & x \mid \lambda x.G \mid xHG_1 \cdots G_k \\ R & ::= & (\lambda x.G)(yHG_1 \cdots G_k) \end{array}$$

Proof. ($\Rightarrow$) Assume that M is in v-nf and proceed by structural induction. Recall that every λ-term M can be uniquely written as $\lambda x_1 \ldots x_m.M'N_1 \cdots N_n$ where $m, n \geq 0$ and either $M' = x$ or $M' = (\lambda x.P)Q$. Moreover, the λ-terms $M', N_1, \ldots, N_n$ must be in v-nf's since M is v-nf. Now, if $m > 0$ then M is of the form $\lambda x.P$ with P in v-nf and the result follows from the induction hypothesis. Hence, we assume $m = 0$ and split into cases depending on M':

- $M' = x$ for some $x \in \mathbb{V}$. If $n = 0$ then we are done since x is an H-term. If $n > 0$ then $M = xN_1 \cdots N_n$ where all the N_i's are G-terms by induction hypothesis. Moreover, N_1 cannot be an R-term for otherwise M would have a σ_3-redex. Whence, N_1 must be an H-term and M is of the form $xHG_1 \cdots G_k$ for $k = n - 1$.
- $M' = (\lambda x.P)Q$ for some variable x and λ-terms P, Q in v-nf. In this case we must have $n = 0$ because M cannot have a σ_1-redex. By induction hypothesis, P, Q are G-terms, but Q cannot be an R-term or a value for otherwise M would have a σ_3- or a β_v-redex, respectively. We conclude that the only possibility for the shape of Q is $yHG_1 \cdots G_k$, whence M must be an R-term.

($\Leftarrow$) By induction on the grammar generating M. The only interesting cases are the following.

- $M = xHG_1 \cdots G_k$ could have a σ_3-redex if $H = (\lambda y.P)Q$, but this is impossible by definition of an H-term. As $H, G_1, \ldots, G_k$ are in v-nf by induction hypothesis, so must be M.
- $M = (\lambda x.G)(yHG_1 \cdots G_k)$ where $G, H, G_1, \ldots, G_k$ are in v-nf by induction hypothesis. In the previous item we established that $yHG_1 \cdots G_k$ is in v-nf. Thus, M could only have a β_v-redex if $yHG_1 \cdots G_k \in \mathrm{Val}$, but this is not the case by definition of Val. $\square$

Intuitively, in the grammar above, G stands for "general" normal form, R for "redex-like" normal form and H for "head" normal form. The following properties are well-established.

Proposition 1.8 (Properties of reductions [Plo75, CG14])**.**

(1) *The σ-reduction is confluent and strongly normalizing.*
(2) *The β_v- and v-reductions are confluent.*

Lambda terms are classified into valuables, potentially valuable and non-potentially valuable, depending on their capability of producing a value in a suitable environment.

Definition 1.9. A λ-term M is *valuable* if $M \twoheadrightarrow_{\beta_v} V$ for some $V \in \mathrm{Val}$. A λ-term M is *potentially valuable* if there exists a head context[3] $C(\!|-|\!) = (\lambda x_1 \dots x_n.(\!|-|\!))V_1 \cdots V_n$, where $\mathrm{FV}(M) = \{x_1, \dots, x_n\}$, such that $C(\!|M|\!)$ is valuable.

It is easy to check that M valuable entails M potentially valuable and that, for $M \in \Lambda^o$, the two notions coincide. As shown in [GPR17], a λ-term M is valuable (resp. potentially valuable) if and only if $M \twoheadrightarrow_{\mathsf{v}} V$ (resp. $C(\!|M|\!) \twoheadrightarrow_{\mathsf{v}} V$) for some $V \in \mathrm{Val}$. As a consequence, the calculus λ_v^σ can be used as a tool for studying the operational semantics of the original calculus λ_v.

In [Plo75], Plotkin defines an observational equivalence analogous to the following one.

Definition 1.10. The *observational equivalence* $\equiv$ is defined as follows (for $M, N \in \Lambda$):

$$M \equiv N$$
$$\iff$$
$$\forall C(\!|-|\!)\,.\,C(\!|M|\!), C(\!|N|\!) \in \Lambda^o\ [\ \exists V \in \mathrm{Val}\,.\,C(\!|M|\!) \twoheadrightarrow_{\beta_v} V \iff \exists U \in \mathrm{Val}\,.\,C(\!|N|\!) \twoheadrightarrow_{\beta_v} U\ ]$$

For example, we have $\mathtt{I} \equiv \lambda xy.xy$ and $\Xi \equiv \Omega$ (see Example 1.6(6)), while $\Omega \not\equiv \lambda x.\Omega$.

Remark 1.11. It is well known that, in order to check whether $M \equiv N$ holds, it is enough to consider head contexts (cf. [Ong97, Pao08]). In other words, $M \not\equiv N$ if and only if there exists a head context $C(\!|-|\!)$ such that $C(\!|M|\!)$ is valuable, while $C(\!|N|\!)$ is not.

2. Call-by-value Böhm Trees

In the call-by-name setting there are several equivalent ways of defining Böhm trees. The most famous definition is coinductive[4] [Las99], while the formal one in Barendregt's book exploits the notion of "effective Böhm-like trees" which is not easy to handle in practice. The definition given in Amadio and Curien's book [AC98, Def. 2.3.3] is formal, does not require coinductive techniques and, as it turns out, generalizes nicely to the CbV setting. The idea is to first define the set $\mathcal{A}(M)$ of approximants of a λ-term M, then show that it is directed w.r.t. some preorder $\sqsubseteq$ and, finally, define the Böhm tree of M as the supremum of $\mathcal{A}(M)$.

2.1. Böhm trees and approximants.

Let $\Lambda_\perp$ be the set of λ-terms possibly containing a constant $\perp$, representing the undefined *value*, and let $\sqsubseteq$ be the context-closed preorder on $\Lambda_\perp$ generated by setting, for all $x \in \mathbb{V}$ and $M \in \Lambda_\perp$:

$$\perp \sqsubseteq x, \qquad\qquad \perp \sqsubseteq \lambda x.M.$$

Notice that, by design, $\perp$ can only be used to approximate values, not λ-terms like Ω.

The reduction $\to_{\mathsf{v}}$ from Definition 1.5 generalizes to terms in $\Lambda_\perp$ in the obvious way, namely by considering a set $\mathrm{Val}_\perp$ of values generated by the grammar (for $M \in \Lambda_\perp$):

$$(\mathrm{Val}_\perp) \qquad U, V \ ::= \ \perp \mid x \mid \lambda x.M$$

For example, the β_v-reduction is extended by setting for all $M, V \in \Lambda_\perp$:

$$(\beta_v) \qquad (\lambda x.M)V \to M[x := V] \text{ whenever } V \in \mathrm{Val}_\perp$$

[3]Equivalently, M is potentially valuable if there is a substitution $\vartheta : \mathbb{V} \to \mathrm{Val}$ such that $\vartheta(M)$ is valuable.

[4]See also Definition 10.1.3 of [Bar84], marked by Barendregt as 'informal' because at the time the coinduction principle was not as well-understood as today.

Similarly, for the σ-rules. A $\perp$-*context* $C(\!|-|\!)$ is a context possibly containing some occurrences of $\perp$. We use for $\perp$-contexts the same notations introduced for contexts in Section 1.1.

Given $M, N \in \Lambda_\perp$ compatible[5] w.r.t. $\sqsubseteq$, we denote their least upper bound by $M \sqcup N$.

Definition 2.1.

(1) The set $\mathcal{A}$ of *approximants* contains the terms $A \in \Lambda_\perp$ generated by the grammar (for $k \geq 0$):

$$\begin{array}{rcl} A & ::= & B \mid C \\ B & ::= & x \mid \lambda x.A \mid \perp \mid xBA_1 \cdots A_k \\ C & ::= & (\lambda x.A)(yBA_1 \cdots A_k) \end{array}$$

(2) The *set of free variables* $\mathrm{FV}(-)$ is extended to approximants by setting $\mathrm{FV}(\perp) = \emptyset$.

(3) Given $M \in \Lambda$, *the set of approximants of* M is defined as follows:

$$\mathcal{A}(M) = \{A \in \mathcal{A} \mid \exists N \in \Lambda, M \twoheadrightarrow_{\mathsf{v}} N \text{ and } A \sqsubseteq N\}.$$

Example 2.2. (1) $\mathcal{A}(\mathtt{I}) = \{\perp, \lambda x.\perp, \lambda x.x\}$.

(2) $\mathcal{A}(\Omega) = \mathcal{A}(\Xi) = \emptyset$ and $\mathcal{A}(\lambda x.\Omega) = \{\perp\}$.

(3) $\mathcal{A}(\mathtt{I}(\Delta(xx))) = \{(\lambda z.(\lambda y.Y)(zZ))(xX) \mid Y \in \{y, \perp\} \ \wedge \ Z \in \{z, \perp\} \ \wedge \ X \in \{x, \perp\}\}$.
Notice that neither $(\lambda z.\perp)(xx)$ nor $(\lambda z.\perp)(x\perp)$ belong to this set, because $\perp \not\sqsubseteq \mathtt{I}(zz)$.

(4) $\mathcal{A}(\mathtt{Z}) = \bigcup_{n\in\mathbb{N}}\{\lambda f.f(\lambda z_0.f(\lambda z_1.f \cdots (\lambda z_n.f \perp Z_n) \cdots Z_1)Z_0) \mid \forall i\,.\, Z_i \in \{z_i, \perp\}\} \cup \{\perp\}$.

(5) $\mathcal{A}(\mathtt{K}^*) = \{\lambda x_1 \dots x_n.\perp \mid n \geq 0\}$.

(6) The set of approximants of $\mathtt{ZB}$ is particularly interesting to calculate:

$$\begin{array}{rl}\mathcal{A}(\mathtt{ZB}) = & \{\lambda f_0 x_0.(\cdots(\lambda f_{n-1}x_{n-1}.(\lambda f_n.\perp)(f_{n-1}X_{n-1}))\cdots)(f_0X_0) \mid n > 0, \forall i\,.\, X_i \in \{x_i, \perp\}\} \\ & \cup \{\perp, \lambda f_0.\perp\}.\end{array}$$

Lemma 2.3. *Every $M \in \mathcal{A}$ is in normal form with respect to the extended* v*-reduction.*

Proof. By a simple case analysis (analogous to the proof of Lemma 1.7). □

The following lemmas show that the "external shape" of an approximant is stable under v-reduction. For instance, if $A = (\lambda x.A_0)(yBA_1 \cdots A_k) \sqsubseteq M$ then all approximants $A' \in \mathcal{A}(M)$ have shape $(\lambda x.A'_0)(yB'A'_1 \cdots A'_k)$ for some $B', A'_0, \dots, A'_k \in \mathcal{A}$.

Lemma 2.4. *Let $C(\!|-|\!)$ be a (single-hole) $\perp$-context and $V \in \mathrm{Val}$. Then $C(\!|\perp|\!) \in \mathcal{A}$ and $C(\!|V|\!) \to_{\mathsf{v}} N$ entails that there exists a value V' such that $V \to_{\mathsf{v}} V'$ and $N = C(\!|V'|\!)$.*

Proof. Let $A = C(\!|\perp|\!) \in \mathcal{A}$. By Lemma 2.3, A cannot have any v-redex. Clearly, substituting V for an occurrence of $\perp$ in A does not create any new β_v-redex, so if $C(\!|V|\!) \to_{\beta_v} N$ then the contracted redex must occur in V. As V is a value, it can only v-reduce to a value V'.

It is slightly trickier to check by induction on $C(\!|-|\!)$ that such an operation does not introduce any σ-redex. The only interesting case is $C(\!|\perp|\!) = (\lambda x.A')(xC'(\!|\perp|\!)A_1 \cdots A_k)$ where $C'(\!|\perp|\!)$ is a B-term. Indeed, since $x \in \mathrm{Val}$, $xC'(\!|V|\!)$ would be a σ_3-redex for $C'(\!|V|\!) = (\lambda y.P)Q$ but this is impossible since $C'(\!|\perp|\!)$ is a B-term and B-terms cannot have this shape.

The case $C(\!|\perp|\!) = xC'(\!|\perp|\!)A_1 \cdots A_k$ is analogous. □

Lemma 2.5. *For $M \in \Lambda$ and $A \in \mathcal{A}$, $A \sqsubseteq M$ and $M \to_{\mathsf{v}} N$ entails $A \sqsubseteq N$.*

Proof. If $A \sqsubseteq M$ then M can be obtained from A by substituting each occurrence of $\perp$ for the appropriate subterm of M, and such subterm must be a value. Hence, the redex contracted in $M \to_{\mathsf{v}} N$ must occur in a subterm V of M corresponding to an occurrence[6] $C(\!|-|\!)$ of $\perp$ in

[5] Recall that M, N are *compatible* if there exists Z such that $M \sqsubseteq Z$ and $N \sqsubseteq Z$.

[6] An *occurrence* of a subterm N in a λ-term M is a (single-hole) context $C(\!|-|\!)$ such that $M = C(\!|N|\!)$.

A. So we have $C(\!|\bot|\!) = A$ and $C(\!|V|\!) \to_{\mathsf{v}} N'$ implies, by Lemma 2.4, that $N' = C(\!|V'|\!)$ for a V' such that $V \to_{\mathsf{v}} V'$. So we conclude that $A = C(\!|\bot|\!) \sqsubseteq N$, as desired. □

Lemma 2.6. *For $M, N \in \Lambda$, $M \to_{\mathsf{v}} N$ entails $\mathcal{A}(M) = \mathcal{A}(N)$.*

Proof. Straightforward from Definition 2.1(3) and Lemma 2.5. □

Proposition 2.7. *For all $M \in \Lambda$, the set $\mathcal{A}(M)$ is either empty or an ideal (i.e. non-empty, directed and downward closed) w.r.t. $\sqsubseteq$.*

Proof. Assume $\mathcal{A}(M)$ is non-empty. We check the remaining two conditions:

- To show that $\mathcal{A}(M)$ is directed, we need to prove that every $A_1, A_2 \in \mathcal{A}(M)$ have an upper bound $A_3 \in \mathcal{A}(M)$.

  We proceed by induction on A_1. In case $A_1 = \bot$ (resp. $A_2 = \bot$) simply take $A_3 = A_2$ (resp. $A_3 = A_1$). Let us assume that $A_1, A_2 \neq \bot$.

  Case $A_1 = x$, then $A_3 = A_2 = x$.

  Case $A_1 = xB_1A_1^1 \cdots A_k^1$. In this case we must have $M \twoheadrightarrow_{\mathsf{v}} N_1$ for $N_1 = xN_0' \cdots N_k'$ with $B_1 \sqsubseteq N_0'$ and $A_i^1 \sqsubseteq N_i'$ for all i such that $1 \leq i \leq k$. As $A_2 \in \mathcal{A}(M)$, there exists a λ-term N_2 such that $M \twoheadrightarrow_{\mathsf{v}} N_2$ and $A_2 \sqsubseteq N_2$. By Proposition 1.8(2) (confluence), N_1 and N_2 have a common reduct N. Since $A_1 \sqsubseteq N_1$, by Lemma 2.5 we get $A_1 \sqsubseteq N$ thus $N = xN_0 \cdots N_k$. By Lemma 2.5 again, $A_2 \sqsubseteq N$ whence $A_2 = xB_2A_1^2 \cdots A_k^2$ for some approximants $B_2 \sqsubseteq N_0$ and $A_i^2 \sqsubseteq N_i$ for $1 \leq i \leq k$. Now, by definition, $B_1, B_2 \in \mathcal{A}(N_0)$ and $A_i^1, A_i^2 \in \mathcal{A}(N_i)$ for $1 \leq i \leq k$. By induction hypothesis, there exists $B_3 \in \mathcal{A}(N_0), A_i^3 \in \mathcal{A}(N_i)$ such that $B_1 \sqsubseteq B_3 \sqsupseteq B_2$ and $A_i^1 \sqsubseteq A_i^3 \sqsupseteq A_i^2$ from which it follows that the upper bound $xB_3A_1^3 \cdots A_k^3$ of A_1, A_2 belongs to $\mathcal{A}(xN_0 \cdots N_k)$. By Lemma 2.6, we conclude that $xB_3A_1^3 \cdots A_k^3 \in \mathcal{A}(M)$, as desired.

  Case $A_1 = (\lambda x.A_1')(yB_1A_1^1 \cdots A_k^1)$. In this case we must have $M \twoheadrightarrow_{\mathsf{v}} N_1$ for $N_1 = (\lambda x.M')(yM_0 \cdots M_k)$ with $A_1' \sqsubseteq M'$, $B_1 \sqsubseteq M_0$ and $A_i^1 \sqsubseteq M_i$ for all i such that $1 \leq i \leq k$. Reasoning as above, $A_2 \in \mathcal{A}(M)$ implies there exists a λ-term N_2 such that $M \twoheadrightarrow_{\mathsf{v}} N_2$ and $A_2 \sqsubseteq N_2$. By Proposition 1.8(2), N_1 and N_2 have a common reduct N. Since $A_1 \sqsubseteq N_1$, by Lemma 2.5 we get $A_1 \sqsubseteq N$ thus $N = (\lambda x.N')(yN_0 \cdots N_k)$. By Lemma 2.5 again, $A_2 \sqsubseteq N$ whence $A_2 = (\lambda x.A_2')(yB_2A_1^2 \cdots A_k^2)$ where $A_2' \sqsubseteq N'$, $B_2 \sqsubseteq N_0$ and $A_i^2 \sqsubseteq N_i$ for $1 \leq i \leq k$. By induction hypothesis we get $A_3' \in \mathcal{A}(N')$ such that $A_1' \sqsubseteq A_3' \sqsupseteq A_2'$, $B_3 \in \mathcal{A}(N_0)$ such that $B_1 \sqsubseteq B_3 \sqsupseteq B_2$ and $A_i^3 \in \mathcal{A}(N_i)$ such that $A_i^1 \sqsubseteq A_i^3 \sqsupseteq A_i^2$ for $1 \leq i \leq k$. It follows that the upper bound $(\lambda x.A_3')(yB_3A_1^3 \cdots A_k^3)$ of A_1, A_2 belongs to $\mathcal{A}((\lambda x.N')(yN_0 \cdots N_k))$. By Lemma 2.6, we conclude that $(\lambda x.A_3')(yB_3A_1^3 \cdots A_k^3) \in \mathcal{A}(M)$.

  All other cases follow from Lemma 2.5, confluence of $\to_{\mathsf{v}}$ and the induction hypothesis.
- To prove that $\mathcal{A}(M)$ is downward closed, we need to show that for all $A_1, A_2 \in \mathcal{A}$, if $A_1 \sqsubseteq A_2 \in \mathcal{A}(M)$ then $A_1 \in \mathcal{A}(M)$, but this follows directly from its definition. □

As a consequence, whenever $\mathcal{A}(M) \neq \emptyset$, we can actually define the Böhm tree of a λ-term M as the supremum of its approximants in $\mathcal{A}(M)$.

Definition 2.8. (1) Let $M \in \Lambda$. The *(call-by-value) Böhm tree of M*, in symbols $\mathrm{BT}(M)$, is defined as follows (where we assume that $\bigsqcup \emptyset = \emptyset$):

$$\mathrm{BT}(M) = \bigsqcup \mathcal{A}(M)$$

Therefore, the resulting structure is a possibly infinite labelled tree T.

(2) More generally, every $\mathcal{X} \subseteq \mathcal{A}$ directed and downward closed determines a so-called *Böhm-like tree* $T = \bigsqcup \mathcal{X}$.

(3) Given a Böhm-like tree T, we set $\mathrm{FV}(T) = \mathrm{FV}(\mathcal{X}) = \bigcup_{A\in\mathcal{X}} \mathrm{FV}(A)$.

The difference between the Böhm tree of a λ-term M and a Böhm-like tree T is that the former must be "computable[7]" since it is λ-definable, while the latter can be arbitrary. In particular, any Böhm tree $\mathrm{BT}(M)$ is a Böhm-like tree but the converse does not hold.

Remark 2.9. (1) Notice that $\mathcal{A}(M) = \mathcal{A}(N)$ if and only if $\mathrm{BT}(M) = \mathrm{BT}(N)$.
(2) The supremum $\bigsqcup \mathcal{X}$ in Definition 2.8(2) (and a fortiori $\bigsqcup \mathcal{A}(M)$, in (1)) belongs to the larger set $\mathscr{X}$ generated by taking the grammar in Definition 2.1(1) coinductively, whose elements are ordered by $\sqsubseteq$ extended to infinite terms. However, $\mathscr{X}$ contains terms like

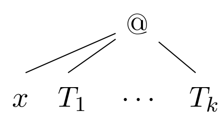
$$(\lambda y_1.(\lambda y_2.(\lambda y_3.\cdots)(yy))(yy))(yy) \in \mathscr{X}$$

that are not "Böhm-like" as they cannot be obtained as the supremum of a directed subset $\mathcal{X} \subseteq \mathcal{A}$.
(3) $\mathrm{FV}(\mathrm{BT}(M)) \subseteq \mathrm{FV}(M)$ and the inclusion can be strict: $\mathrm{FV}(\mathrm{BT}(\lambda x.\Omega y)) = \mathrm{FV}(\bot) = \emptyset$.

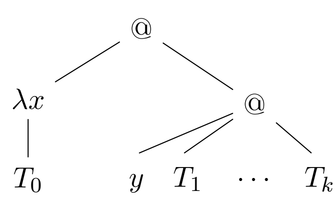
The Böhm-like trees defined above as the supremum of a set of approximants can be represented as actual trees. Indeed, any Böhm-like tree T can be depicted using the following "building blocks".

- If $T = \bot$ we actually draw a node labelled $\bot$.
- If $T = \lambda x.T'$ we use an *abstraction node* labelled "λx":

- If $T = xT_1 \cdots T_k$, we use an *application node* labelled by "@":



- If $T = (\lambda x.T_0)(yT_1 \cdots T_k)$ we combine the application and abstraction nodes as imagined:



Notice that the tree T_1 in the last two cases need to respect the shape of the corresponding approximant (Definition 2.1(1)) for otherwise T would not be the supremum of an ideal.

Example 2.10. Notable examples of Böhm trees of λ-terms are given in Figure 1. Interestingly, the λ-term Ξ from Example 1.6(6) satisfying

$$\Xi =_{\mathsf{v}} (\lambda y_1.((\lambda y_2.(\cdots(\lambda y_n.\Xi\mathtt{I}^{\sim n})(zz)\cdots))(zz)))(zz) \tag{2.1}$$

is such that $\mathrm{BT}(\Xi) = \bot$. Indeed, substituting $\bot$ for a $\lambda y_n.\Xi\mathtt{I}^{\sim n}$ in (2.1) never gives an approximant belonging to $\mathcal{A}$ (*cf.* the grammar of Definition 2.1(1)).

[7]The formal meaning of "computable" will be discussed in the rest of the section.

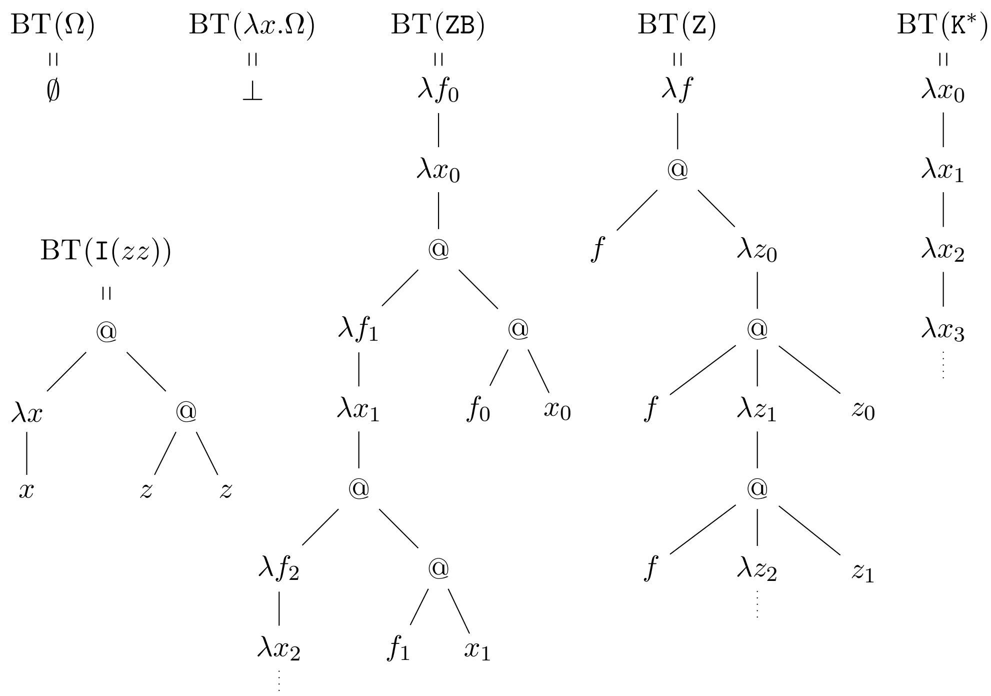


Figure 1: Examples of CbV Böhm trees.

Proposition 2.11. *For* $M, N \in \Lambda$*, if* $M =_{\mathsf{v}} N$ *then* $\mathrm{BT}(M) = \mathrm{BT}(N)$*.*

Proof. By Proposition 1.8(2) (*i.e.* confluence of $\to_{\mathsf{v}}$), $M =_{\mathsf{v}} N$ if and only if there exists a λ-term P such that $M \twoheadrightarrow_{\mathsf{v}} P$ and $N \twoheadrightarrow_{\mathsf{v}} P$. By an iterated application of Lemma 2.6 we get $\mathcal{A}(M) = \mathcal{A}(P) = \mathcal{A}(N)$, so we conclude $\mathrm{BT}(M) = \mathrm{BT}(N)$. □

Theorem 2.13 below provides a characterization of those Böhm-like trees arising as the Böhm tree of some λ-term, in the spirit of [Bar84, Thm. 10.1.23]. To achieve this result, it will be convenient to consider a tree as a set of sequences closed under prefix.

We denote by $\mathbb{N}^*$ the set of finite sequences of natural numbers. Given $n_1, \dots, n_k \in \mathbb{N}$, the corresponding sequence $\sigma \in \mathbb{N}^*$ of length k is represented by $\sigma = \langle n_1, \dots, n_k \rangle$. In particular, $\langle\rangle$ represents the empty sequence of length 0. Given $\sigma \in \mathbb{N}^*$ as above and $n \in \mathbb{N}$, we write $n :: \sigma$ for the sequence $\langle n, n_1, \dots, n_k \rangle$ and $\sigma; n$ for the sequence $\langle n_1, \dots, n_k, n \rangle$.

Given a tree T, the sequence $i :: \sigma$ possibly determines a subtree that can be found going through the $(i+1)$-th children of T (if it exists) and then following the path σ. Of course this is only the case if $i :: \sigma$ actually belongs to the domain of the tree. The following definition formalizes this intuitive idea in the particular case of syntax trees of approximants.

Definition 2.12. Let $\sigma \in \mathbb{N}^*, A \in \mathcal{A}$. The *subterm of A at σ*, written A_σ, is defined by:

$$A_{\langle\rangle} = A \qquad (\lambda x.A)_\sigma = \begin{cases} A_\tau & \text{if } \sigma = 0 :: \tau, \\ \uparrow & \text{otherwise,} \end{cases}$$

$$\bot_\sigma = \uparrow \qquad (xA_0 \cdots A_k)_\sigma = \begin{cases} (A_{i-1})_\tau & \text{if } 1 \leq i \leq k+1 \text{ and } \sigma = i :: \tau, \\ \uparrow & \text{otherwise,} \end{cases}$$

$$((\lambda x.A')(yA_0 \cdots A_k))_\sigma = \begin{cases} A'_\tau & \text{if } \sigma = 0 :: 0 :: \tau, \\ (A_{i-1})_\tau & \text{if } 1 \leq i \leq k+1 \text{ and } \sigma = 1 :: i :: \tau, \\ \uparrow & \text{otherwise.} \end{cases}$$

As a matter of notation, given an approximant A', a subset $\mathcal{X} \subseteq \mathcal{A}$ and a sequence $\sigma \in \mathbb{N}^*$, we write $\exists A_\sigma \simeq_{\mathcal{X}} A'$ whenever there exists $A \in \mathcal{X}$ such that A_σ is defined and $A_\sigma = A'$.

Theorem 2.13. *Let $\mathcal{X} \subseteq \mathcal{A}$ be a set of approximants. There exists $M \in \Lambda$ such that $\mathcal{A}(M) = \mathcal{X}$ if and only if the following three conditions hold:*
(1) *$\mathcal{X}$ is directed and downward closed w.r.t. $\sqsubseteq$,*
(2) *$\mathcal{X}$ is r.e. (after coding),*
(3) $\mathrm{FV}(\mathcal{X})$ *is finite.*

Proof sketch. ($\Rightarrow$) Let $M \in \Lambda$ be such that $\mathcal{X} = \mathcal{A}(M)$, then (1) is satisfied by Proposition 2.7 and (3) by Remark 2.9. Concerning (2), let us fix an effective bijective encoding $\# : \Lambda_\bot \to \mathbb{N}$. Then the set $\{\#A \mid A \in \mathcal{X}\}$ is r.e. because it is semi-decidable to determine if $M \twoheadrightarrow_{\mathsf{v}} N$ (just enumerate all v-reducts of M and check whether N is one of them), the set $\{\#A \mid A \in \mathcal{A}\}$ and the relation $\sqsubseteq$ restricted to $\mathcal{A} \times \Lambda$ are decidable.

($\Leftarrow$) Assume that $\mathcal{X}$ is a set of approximants satisfying the conditions (1-3).

If $\mathcal{X} = \emptyset$ then we can simply take $M = \Omega$ since $\mathcal{A}(\Omega) = \emptyset$.

If $\mathcal{X}$ is non-empty then it is an ideal. Since $\mathcal{X}$ is r.e., if $A' \in \mathcal{A}$ and $\sigma \in \mathbb{N}^*$ are effectively given then the condition $\exists A_\sigma \simeq_{\mathcal{X}} A'$ is semi-decidable and a witness $\mathtt{A}$ can be computed. Let $\ulcorner \sigma \urcorner$ be the numeral associated with σ under an effective encoding and $\ulcorner A \urcorner$ be the *quote of A* as defined by Mogensen[8] in [Mog92], using a fresh variable $z_b \notin \mathrm{FV}(\mathcal{X})$ to represent the $\bot$. (Such variable always exists because $\mathrm{FV}(\mathcal{X})$ is finite.) The CbV λ-calculus being Turing-complete, as shown by Paolini in [Pao01], there exists a λ-term $P_{\mathcal{X}}$ satisfying:

$$P_{\mathcal{X}} \ulcorner \sigma \urcorner \ulcorner A' \urcorner =_{\mathsf{v}} \begin{cases} \ulcorner \mathtt{A} \urcorner & \text{if } \exists A_\sigma \simeq_{\mathcal{X}} A' \text{ holds,} \\ \text{not potentially valuable} & \text{otherwise.} \end{cases}$$

for some witness $\mathtt{A}$. Recall that there exists an *evaluator* $\mathtt{E} \in \Lambda^o$ such that $\mathtt{E}\ulcorner M \urcorner =_{\mathsf{v}} M$ for all λ-terms M. Using the λ-terms $\mathtt{E}, P_{\mathcal{X}}$ so-defined and the recursion operator $\mathtt{Z}$, it is possible

[8] This encoding is particularly convenient because it is effective, defined on open terms by exploiting the fact that $\mathrm{FV}(\ulcorner M \urcorner) = \mathrm{FV}(M)$ and works in the CbV setting as well (easy to check). See also [BDS13, §6.1] for a nice treatment.

to define a λ-term F (also depending on $\mathcal{X}$) satisfying the following recursive equations:

$$F\ulcorner\sigma\urcorner =_{\mathsf{v}} \begin{cases} x & \text{if } \exists A_\sigma \simeq_{\mathcal{X}} x, \\ \lambda x.F\ulcorner\sigma;0\urcorner & \text{if } \exists A_\sigma \simeq_{\mathcal{X}} \lambda x.A_1, \\ x(F\ulcorner\sigma;1\urcorner)\cdots(F\ulcorner\sigma;k+1\urcorner) & \text{if } \exists A_\sigma \simeq_{\mathcal{X}} xA_0\cdots A_k, \\ \big(\lambda x.F\ulcorner\sigma;0;0\urcorner\big)\big(y(F\ulcorner\sigma;1;1\urcorner)\cdots(F\ulcorner\sigma;1;k+1\urcorner)\big) & \text{if } \exists A_\sigma \simeq_{\mathcal{X}} (\lambda x.A)(yA_0\cdots A_k), \\ \text{not valuable} & \text{otherwise.} \end{cases}$$

The fact that $\mathcal{X}$ is directed guarantees that, for a given sequence σ, exactly one of the cases above is applicable. It is now easy to check that $\mathcal{A}(F\ulcorner\langle\rangle\urcorner) = \mathcal{X}$. □

3. Call-By-Value Taylor Expansion

The *(call-by-name) resource calculus* λ_r has been introduced by Tranquilli in his thesis [Tra09], and its promotion-free fragment is the target language of Ehrhard and Regnier's Taylor expansion [ER06]. Both the resource calculus and the notion of Taylor expansion have been adapted to the CbV setting by Ehrhard [Ehr12], using Girard's second translation of intuitionistic arrow in linear logic. Carraro and Guerrieri added to CbV λ_r the analogous of the σ-rules and studied the denotational and operational properties of the resulting language λ_r^σ in [CG14].

3.1. **Its syntax and operational semantics.** We briefly recall here the definition of the *call-by-value resource calculus* λ_r^σ from [CG14], and introduce some notations.

Definition 3.1. The sets Val^r of *resource values*, Λ^s of *simple terms* and Λ^r of *resource terms* are generated by the following grammars (for $k \geq 0$):

$$\begin{array}{llcll} (\mathrm{Val}^r) & u,v & ::= & x \mid \lambda x.t & \text{resource values} \\ (\Lambda^s) & s,t & ::= & st \mid [v_1,\dots,v_k] & \text{simple terms} \\ (\Lambda^r) & e & ::= & v \mid s & \text{resource terms} \end{array}$$

The notions of *α-conversion* and *free variable* are inherited from λ_v^σ. In particular, given $e \in \Lambda^r$, $\mathrm{FV}(e)$ denotes the set of free variables of e. The *size* of a resource term e is defined in the obvious way, while the *height* $\mathsf{ht}(e)$ *of* e is the height of its syntax tree:

$$\begin{array}{lcl} \mathsf{ht}(x) & = & 0, \\ \mathsf{ht}(\lambda x.t) & = & \mathsf{ht}(t)+1, \\ \mathsf{ht}(st) & = & \max\{\mathsf{ht}(s),\mathsf{ht}(t)\}+1, \\ \mathsf{ht}([v_1,\dots,v_k]) & = & \max\{\mathsf{ht}(v_i) \mid i \leq k\}+1. \end{array}$$

Resource values are analogous to the values of λ_v^σ, namely variables and λ-abstractions. Simple terms of shape $[v_1,\dots,v_n]$ are called *bags* and represent finite multisets of linear resources — this means that every v_i must be used exactly once along the reduction. Indeed, when a singleton bag $[\lambda x.t]$ is applied to a bag $[v_1,\dots,v_n]$ of resource values, each v_i is substituted for exactly one free occurrence of x in t. Such an occurrence is chosen non-deterministically, and all possibilities are taken into account — this is expressed by a set-theoretical union of resource terms (see Example 3.5 below). In case there is a mismatch between the cardinality of the bag and the number of occurrences of x in t, the reduction relation "raises an exception" and the result of the computation is the empty set $\emptyset$.

Whence, we need to introduce some notations concerning sets of resource terms.

$$\frac{t \to_{\mathsf{r}} \mathcal{T}}{\lambda x.t \to_{\mathsf{r}} \lambda x.\mathcal{T}} \qquad \frac{s \to_{\mathsf{r}} \mathcal{S}}{s\,t \to_{\mathsf{r}} \mathcal{S}\,t} \qquad \frac{t \to_{\mathsf{r}} \mathcal{T}}{s\,t \to_{\mathsf{r}} s\,\mathcal{T}} \qquad \frac{v_0 \to_{\mathsf{r}} \mathcal{V}_0}{[v_0, v_1, \dots, v_k] \to_{\mathsf{r}} [\mathcal{V}_0, v_1, \dots, v_k]}$$

$$\frac{e \to_{\mathsf{r}} \mathcal{E}_1 \quad e \notin \mathcal{E}_2}{\{e\} \cup \mathcal{E}_2 \to_{\mathsf{r}} \mathcal{E}_1 \cup \mathcal{E}_2}$$

Figure 2: Contextual rules for $\to_{\mathsf{r}} \subseteq \mathscr{P}_{\mathrm{f}}(\Lambda^r) \times \mathscr{P}_{\mathrm{f}}(\Lambda^r)$.

Notation 3.2. Sets of resource values, simple terms and resource terms are denoted by:

$$\mathcal{U}, \mathcal{V} \in \mathscr{P}(\mathrm{Val}^r), \quad \mathcal{S}, \mathcal{T} \in \mathscr{P}(\Lambda^s), \quad \mathcal{E} \in \mathscr{P}(\Lambda^r),$$

To simplify the subsequent definitions, given $\mathcal{S}, \mathcal{T} \in \mathscr{P}(\Lambda^s)$ and $\mathcal{V}_1, \dots, \mathcal{V}_k \in \mathscr{P}(\mathrm{Val}^r)$ we fix the following notations (as a syntactic sugar, not as actual syntax):

$$\begin{array}{rcl} \lambda x.\mathcal{T} & = & \{\lambda x.t \mid t \in \mathcal{T}\} \in \mathscr{P}(\mathrm{Val}^r), \\ \mathcal{S}\,\mathcal{T} & = & \{st \mid s \in \mathcal{S}, t \in \mathcal{T}\} \in \mathscr{P}(\Lambda^s), \\ [\mathcal{V}_1, \dots, \mathcal{V}_k] & = & \{[v_1, \dots, v_k] \mid v_1 \in \mathcal{V}_1, \dots, v_k \in \mathcal{V}_k\} \in \mathscr{P}(\Lambda^s). \end{array}$$

Indeed all constructors of λ_r^σ are multi-linear, so we get $\lambda x.\emptyset = \emptyset\mathcal{T} = \mathcal{S}\emptyset = [\emptyset, \mathcal{V}_1, \dots, \mathcal{V}_k] = \emptyset$.

These notations are used in a crucial way, *e.g.*, in Definition 3.4(2).

Definition 3.3. Let $e \in \Lambda^r$ and $x \in \mathbb{V}$.

(1) Define the *degree of x in e*, written $\deg_x(e)$, as the number of free occurrences of the variable x in the resource term e.
(2) Let $e \in \Lambda^r$, $v_1, \dots, v_n \in \mathrm{Val}^r$ and $x \in \mathbb{V}$. The *linear substitution of $v_1, \dots, v_n$ for x in e*, denoted by $e\langle x := [v_1, \dots, v_n]\rangle \in \mathscr{P}_{\mathrm{f}}(\Lambda^r)$, is defined as follows:

$$e\langle x := [v_1, \dots, v_n]\rangle = \begin{cases} \{e[x_1 := v_{\sigma(1)}, \dots, x_n := v_{\sigma(n)}] \mid \sigma \in \mathfrak{S}_n\}, & \text{if } \deg_x(e) = n, \\ \emptyset, & \text{otherwise.} \end{cases}$$

where $\mathfrak{S}_n$ is the group of permutations over $\{1, \dots, n\}$ and $x_1, \dots, x_n$ is an enumeration of the free occurrences of x in e, so that $e[x_i := v_{\sigma(i)}]$ denotes the resource term obtained from e by replacing the i-th free occurrence of x in e with the resource value $v_{\sigma(i)}$.

The definitions above open the way to introduce the following notions of reduction for λ_r^σ, mimicking the corresponding reductions of λ_v^σ (*cf.* Definition 1.5).

Definition 3.4. (1) The *β_r-reduction* is a relation $\to_{\beta_r} \subseteq \Lambda^r \times \mathscr{P}_{\mathrm{f}}(\Lambda^r)$ defined by the following rule (for $v_1, \dots, v_n \in \mathrm{Val}^r$):

$$(\beta_r) \quad [\lambda x.t][v_1, \dots, v_k] \to t\langle x := [v_1, \dots, v_n]\rangle.$$

Similarly, the *0-reduction* $\to_0 \subseteq \Lambda^r \times \mathscr{P}_{\mathrm{f}}(\Lambda^r)$ is defined by the rule:

$$(0) \quad [v_1, \dots, v_n]\,t \to \emptyset, \qquad \text{when } n \neq 1.$$

The *σ-reductions* $\to_{\sigma_1}, \to_{\sigma_3} \subseteq \Lambda^r \times \Lambda^r$ are defined by the rules:

$$\begin{array}{lll} (\sigma_1) & [\lambda x.t]s_1 s_2 \to [\lambda x.ts_2]s_1, & \text{if } x \notin \mathrm{FV}(s_1), \\ (\sigma_3) & [v]([\lambda x.t]s) \to [\lambda x.[v]t]s, & \text{if } x \notin \mathrm{FV}(v) \text{ and } v \in \mathrm{Val}^r. \end{array}$$

(2) The relation $\to_{\mathsf{r}} \subseteq \mathscr{P}_{\mathrm{f}}(\Lambda^r) \times \mathscr{P}_{\mathrm{f}}(\Lambda^r)$ is the contextual closure of the rules above, *i.e.* $\to_{\mathsf{r}}$ is the smallest relation including $(\beta_r), (0), (\sigma_1), (\sigma_3)$ and satisfying the rules in Figure 2.
(3) The transitive and reflexive closure of $\to_{\mathsf{r}}$ is denoted by $\twoheadrightarrow_{\mathsf{r}}$, as usual.

Example 3.5. We provide some examples of reductions:

(1) $[\lambda x.[x][x]][\lambda y.[y], z] \to_{\beta_r} \{[\lambda y.[y]][z], [z][\lambda y.[y]]\} \to_{\beta_r} \{[z], [z][\lambda y.[y]]\}$.
(2) $[\lambda x.[x, x]][\lambda y.[y], z] \to_{\beta_r} \{[\lambda y.[y], z], [z, \lambda y.[y]]\} = \{[\lambda y.[y], z]\}$.
(3) $[\lambda y.[\lambda x.[x, x][y]]]([z][w])[\mathtt{I}, w] \to_{\sigma_1} [\lambda y.[\lambda x.[x, x][y]][\mathtt{I}, w]]([z][w]) \to_{\beta_r} \{[\lambda y.[\mathtt{I}, w][y]]([z][w])\}$ $\to_0 \emptyset$. Note that (σ_1) is used to unblock an otherwise stuck β_r-redex.
(4) $[\mathtt{I}]([\lambda x.[\lambda y.[x][y]]][z][w]) \to_{\beta_r} \{[\lambda x.[\lambda y.[x][y]]][z][w]\} \to_{\beta_r} \{[\lambda y.[z][y]][w]\} \to_{\beta_r} \{[z][w]\}$.
(5) $[\mathtt{I}]([\lambda x.[\lambda y.[x][y]]][z][w]) \to_{\sigma_3} [\lambda x.[\mathtt{I}][\lambda y.[x][y]]][z][w] \to_{\sigma_1} [\lambda x.[\mathtt{I}][\lambda y.[x][y]][w]][z] \to_{\beta_r}$ $\{[\mathtt{I}][\lambda y.[z][y]][w]\} \to_{\beta_r} \{[\lambda y.[z][y]][w]\} \to_{\beta_r} \{[z][w]\}$.

Remark that (4) and (5) constitute two different reduction sequences originating from the same simple term.

As shown in [CG14], this notion of reduction enjoys the following properties.

Proposition 3.6. *The reduction $\to_{\mathsf{r}}$ is confluent and strongly normalizing.*

Note that strong normalization is straightforward to prove — indeed, a 0-reduction annihilates the whole term, σ-rules are strongly normalizing (*cf.* Proposition 1.8(1)) and contracting a β_r-redex in a resource term e produces a set of resource terms whose size is strictly smaller than e because no duplication is involved and a λ-abstraction is erased.

As a consequence of Proposition 3.6, the r-*normal form* of $\mathcal{E} \in \mathscr{P}_{\mathrm{f}}(\Lambda^r)$ always exists and is denoted by $\mathrm{nf}_{\mathsf{r}}(\mathcal{E})$, i.e. $\mathcal{E} \twoheadrightarrow_{\mathsf{r}} \mathrm{nf}_{\mathsf{r}}(\mathcal{E}) \in \mathscr{P}_{\mathrm{f}}(\Lambda^r)$ and there is no $\mathcal{E}'$ such that $\mathrm{nf}_{\mathsf{r}}(\mathcal{E}) \to_{\mathsf{r}} \mathcal{E}'$. Simple terms in r-nf are called "resource approximants" because their role is similar to the one played by finite approximants of Böhm trees, except that they approximate the normal form of the Taylor expansion. They admit the following syntactic characterization.

Definition 3.7. A *resource approximant* $a \in \Lambda^s$ is a simple term generated by the following grammar (for $k, n \geq 0$, where $[x^n]$ is the bag $[x, \dots, x]$ having n occurrences of x):

$$\begin{array}{lcl} a & ::= & b \mid c \\ b & ::= & [x^n] \mid [\lambda x.a_1, \dots, \lambda x.a_n] \mid [x] b a_1 \cdots a_k \\ c & ::= & [\lambda x.a]([y] b a_1 \cdots a_k) \end{array}$$

It is easy to check that resource approximants are r-normal forms.

Example 3.8. The following are examples of resource approximants:

(1) $[\lambda x.[x], \lambda x.[x, x], \lambda x.[x, x, x]]$ and $[\lambda x.[x][x, x], \lambda x.[x][x, x, x]]$ belong to the Taylor expansion of some λ-term (as we will see in Example 3.11).
(2) $[\lambda x.[x, x, x], \lambda x.[y, y, y]]$ does not, as will be shown in Proposition 3.18.

3.2. **Characterizing the Taylor Expansion of a λ-Term.** We recall the definition of the Taylor expansion of a λ-term in the CbV setting, following [Ehr12, CG14]. Such a Taylor expansion translates a λ-term M into an infinite set[9] of simple terms. Subsequently, we characterize those sets of resource terms arising as a Taylor expansion of some $M \in \Lambda$.

Definition 3.9. The *Taylor expansion* $\mathscr{T}(M) \subseteq \Lambda^s$ of a λ-term M is an infinite set of simple terms defined by induction as follows:

$$\begin{array}{lcl} \mathscr{T}(x) & = & \{[x^n] \mid n \geq 0\}, \text{ where } [x^n] = [x, \dots, x] \ (n \text{ times}), \\ \mathscr{T}(\lambda x.N) & = & \{[\lambda x.t_1, \dots, \lambda x.t_n] \mid n \geq 0, \forall i \leq n,\ t_i \in \mathscr{T}(N)\}, \\ \mathscr{T}(PQ) & = & \{st \mid s \in \mathscr{T}(P),\ t \in \mathscr{T}(Q)\}. \end{array}$$

[9] This set can be thought of as the *support* of the actual Taylor expansion, which is an infinite formal linear combination of simple terms taking coefficients in the semiring of non-negative rational numbers.

From the definition above, we get the following easy properties.

Remark 3.10.
(1) $[] \in \mathscr{T}(V)$ if and only if $V \in \mathrm{Val}$.
(2) Every occurrence of a $\beta_r\sigma$-redex in $t \in \mathscr{T}(M)$ arises from some v-redex in M.
(3) By exploiting Notation 3.2, we can rewrite the Taylor expansion of an application or an abstraction as follows:

$$\begin{array}{rcl}\mathscr{T}(PQ) & = & \mathscr{T}(P)\mathscr{T}(Q), \\ \mathscr{T}(\lambda x.N) & = & \bigcup_{n\in\mathbb{N}}\{[\underbrace{\lambda x.\mathscr{T}(N),\ldots,\lambda x.\mathscr{T}(N)}_{n \text{ times}}]\}.\end{array}$$

Example 3.11. We calculate the Taylor expansion of some λ-terms.
(1) $\mathscr{T}(\mathtt{I}) = \{[\lambda x.[x^{n_1}],\ldots,\lambda x.[x^{n_k}]] \mid k \geq 0, \forall i \leq k, n_i \geq 0\}$,
(2) $\mathscr{T}(\Delta) = \{[\lambda x.[x^{n_1}][x^{m_1}],\ldots,\lambda x.[x^{n_k}][x^{m_k}]] \mid k \geq 0, \forall i \leq k, m_i, n_i \geq 0\}$,
(3) $\mathscr{T}(\Delta\mathtt{I}) = \{st \mid s \in \mathscr{T}(\Delta), t \in \mathscr{T}(\mathtt{I})\}$,
(4) $\mathscr{T}(\Omega) = \{st \mid s,t \in \mathscr{T}(\Delta)\}$,
(5) $\mathscr{T}(\lambda z.yyz) = \{[\lambda z.[y^{\ell_1}][y^{m_1}][z^{n_1}],\ldots,\lambda z.[y^{\ell_k}][y^{m_k}][z^{n_k}]] \mid k \geq 0, \forall i \leq k, \ell_i, m_i, n_i \geq 0\}$,
(6) $\mathscr{T}(\lambda y.f(\lambda z.yyz)) = \{[\lambda y.[f^{n_1}]t_1,\ldots,\lambda y.[f^{n_k}]t_k] \mid k \geq 0, \forall i \leq k, n_i \geq 0, t_i \in \mathscr{T}(\lambda z.yyz)\}$,
(7) $\mathscr{T}(\mathtt{Z}) = \{[\lambda f.s_1t_1,\ldots,\lambda f.s_kt_k] \mid k \geq 0, \forall i \leq k, s_i, t_i \in \mathscr{T}(\lambda y.f(\lambda z.yyz))\}$.

These examples naturally brings to formulate the next remark and lemma.

Remark 3.12. An element t belonging to the Taylor expansion of a λ-term M in v-nf might not be in r-nf, due to the possible presence of 0-redexes. For an example, consider $[\lambda x.[x,x][x,x], \lambda x.[x][x,x,x]] \in \mathscr{T}(\Delta)$. Notice that, since the reduction does not modify the cardinality of a bag, a more refined definition of Taylor expansion eliminating all 0-redexes is possible by substituting the application case with the following:

$$\mathscr{T}(VM_0\cdots M_k) = \{[v]t_0\cdots t_k \mid [v] \in \mathscr{T}(V), \forall i\,.\,(0 \leq i \leq k)\ t_i \in \mathscr{T}(M_i)\}$$

We prefer to keep Ehrhard's original notion because it has a simpler inductive definition.

The following statement concerning the Taylor expansion of λ-terms in v-nf does hold.

Lemma 3.13. *For $M \in \Lambda$, the following are equivalent:*
(1) *M is in v-normal form,*
(2) *every $t \in \mathscr{T}(M)$ is in $\beta_r\sigma$-normal form.*

Proof. ($1 \Rightarrow 2$) Using Lemma 1.7, we proceed by induction on the normal structure of M.

If $M = x$ then $t \in \mathscr{T}(M)$ entails $t = [x,\ldots,x]$ which is in v-nf.

If $M = \lambda x.G$ then $t \in \mathscr{T}(M)$ implies that $t = [\lambda x.t_1,\ldots,\lambda x.t_n]$ where $t_i \in \mathscr{T}(G)$ for all $i \leq n$. By the induction hypothesis each t_i is in $\beta_r\sigma$-nf, hence, so is t.

If $M = xHG_1\cdots G_k$ then $t \in \mathscr{T}(M)$ entails $t = [x^n]st_1\cdots t_k$ for some $n \geq 0$, $s \in \mathscr{T}(H)$ and $t_i \in \mathscr{T}(G_i)$ $(1 \leq i \leq k)$. By induction hypothesis $s, t_1,\ldots,t_k$ are in $\beta_r\sigma$-nf, so t is in β_r-nf. Concerning σ-rules, t could have a σ_3-redex in case $s = [\lambda x.s']t'$ but this is impossible since $s \in \mathscr{T}(H)$ and H cannot have shape $(\lambda x.P)Q$.

If $M = (\lambda x.G)(yHG_1\cdots G_k)$ and $t \in \mathscr{T}(M)$ then $t = [\lambda x.s_1,\ldots,\lambda x.s_n]t'$ for some $n \geq 0$, $s_i \in \mathscr{T}(G)$, $1 \leq i \leq n$, and $t' \in \mathscr{T}(yHG_1\cdots G_k)$. By induction hypothesis, the resource terms $s_1,\ldots,s_n$ and t' are in $\beta_r\sigma$-nf. In principle, when $n = 1$, the simple term t might have the shape either of a β_r-redex or of a σ_3-redex. Both cases are impossible since $t' \in \mathscr{T}(yHG_1\cdots G_k)$ entails $t' = [y^m]st_1\cdots t_k$ which is neither a resource value nor a simple term of shape $[\lambda z.s]s'$. We conclude that t is in $\beta_r\sigma$-nf.

(2 $\Rightarrow$ 1) We prove the contrapositive. Assume that M is not in v-nf, then either M itself is a β_{v}- or σ-redex, or it contains one as a subterm. Let us analyze first the former case.

(β_{v}) If $M = (\lambda x.N)V$ for $V \in \mathrm{Val}$ then, by Remark 3.10(1), the β_r-redex $[\lambda x.s][]$ belongs to $\mathscr{T}(M)$ for every $s \in \mathscr{T}(N)$.

(σ_1) If $M = (\lambda x.N)PQ$ then for all $s \in \mathscr{T}(N), t_1 \in \mathscr{T}(P), t_2 \in \mathscr{T}(Q)$ we have $[\lambda x.s]t_1t_2 \in \mathscr{T}(M)$ and this simple term is a σ_1-redex.

(σ_3) If $M = V((\lambda x.P)Q)$ for $V \in \mathrm{Val}$ then for all $[v] \in \mathscr{T}(V), s \in \mathscr{T}(P)$ and $t' \in \mathscr{T}(Q)$ we have $[v]([\lambda x.s]t') \in \mathscr{T}(M)$ and this resource term is a σ_2-redex.

Otherwise $M = C(\!|M'|\!)$ where C is a context and M' is a v-redex having one of the shapes above; in this case there is $t \in \mathscr{T}(M)$ containing a $\beta_r\sigma$-redex $t' \in \mathscr{T}(M')$ as a subterm. $\square$

The rest of the section is devoted to provide a characterization of all sets of simple terms that arise as the Taylor expansion of some λ-term M.

Definition 3.14. (1) The *height of a non-empty set* $\mathcal{E} \subseteq \Lambda^r$, written $\mathsf{ht}(\mathcal{E})$, is the maximal height of its elements, if it exists, and in this case we say that $\mathcal{E}$ has *finite* height. Otherwise, we define $\mathsf{ht}(\mathcal{E}) = \aleph_0$ and we say that $\mathcal{E}$ has *infinite* height.

(2) Define a *coherence relation* $\coh \subseteq \Lambda^r \times \Lambda^r$ as the smallest relation satisfying:

$$\frac{}{x \coh x} \qquad \frac{s \coh t}{\lambda x.s \coh \lambda x.t} \qquad \frac{v_i \coh v_j \quad (\forall i, j \leq n)}{[v_1, \dots, v_k] \coh [v_{k+1}, \dots, v_n]} \qquad \frac{s_1 \coh s_2 \quad t_1 \coh t_2}{s_1t_1 \coh s_2t_2}$$

(3) A subset $\mathcal{E} \subseteq \Lambda^r$ is a *clique* whenever $e \coh e'$ holds for all $e, e' \in \mathcal{E}$.

(4) A clique $\mathcal{E}$ is *maximal* if, for every $e \in \Lambda^r$, $\mathcal{E} \cup \{e\}$ is a clique entails $e \in \mathcal{E}$.

The coherence relation above is inspired by Ehrhard's work in the call-by-name setting [ER08]. Note that $\coh$ is symmetric, but neither reflexive as $[x, y] \not\coh [x, y]$ nor transitive since $[x] \coh [] \coh [y]$ but $[x] \not\coh [y]$.

Example 3.15. Notice that all sets in Example 3.11 are maximal cliques of finite height. For instance, $\mathsf{ht}(\mathscr{T}(\mathtt{I})) = 3$ and by following the rules in Definition 3.14(2) we have $u \coh t$ for all $t \in \mathscr{T}(\mathtt{I})$ if and only if either $u = []$ or $u = [x^n]$ for some $n \in \mathbb{N}$ if and only if $u \in \mathscr{T}(\mathtt{I})$. Therefore $\mathscr{T}(\mathtt{I})$ is maximal.

The rest of the section is devoted to proving that these two properties actually characterize those sets that the Taylor expansions of λ-terms (Proposition 3.18).

We may now characterize resource approximants (Definition 3.7).

Lemma 3.16. *Let $t \in \Lambda^s$ be such that $t \coh t$. Then t is in r-nf iff t is a resource approximant.*

Proof. Notice that $t \coh t$ guarantees that all terms in each bag occurring in t have similar shape. The proof of the absence of β_r- and σ- redexes, is analogous to the one of Lemma 1.7. The bags occurring in $[x]ba_1 \cdots a_k$ and $[\lambda x.a]([y]ba_1 \cdots a_k)$ must be singleton multisets, for otherwise we would have some 0-redexes. $\square$

This lemma follows easily from Definition 3.14(1) and Remark 3.10(3).

Lemma 3.17. *For $N, P, Q \in \Lambda$, we have:*

(1) $\mathsf{ht}(\mathscr{T}(\lambda x.N)) = \mathsf{ht}(\mathscr{T}(N)) + 2$.

(2) $\mathsf{ht}(\mathscr{T}(PQ)) = \mathsf{ht}(\mathscr{T}(P) \cup \mathscr{T}(Q)) + 1$,

Proof. (1) Indeed, we have:

$$\begin{aligned}
\mathsf{ht}(\mathscr{T}(\lambda x.N)) &= \max\{\mathsf{ht}([\lambda x.t_1,\dots,\lambda x.t_n]) \mid n \geq 0, \forall i \leq n,\ t_i \in \mathscr{T}(N)\},\\
&= \max\{\max\{\mathsf{ht}(\lambda x.t_1),\dots,\mathsf{ht}(\lambda x.t_n)\} + 1 \mid n \geq 0, \forall i \leq n,\ t_i \in \mathscr{T}(N)\},\\
&= \max\{\max\{\mathsf{ht}(t_1),\dots,\mathsf{ht}(t_n)\} + 2 \mid n \geq 0, \forall i \leq n,\ t_i \in \mathscr{T}(N)\},\\
&= \max\{\mathsf{ht}(t) + 2 \mid t \in \mathscr{T}(N)\} = \mathsf{ht}(\mathscr{T}(N)) + 2.
\end{aligned}$$

(2) This case is analogous but simpler, and we omit it. □

The next proposition gives a characterization of those sets of simple terms corresponding to the Taylor expansion of some λ-terms and constitutes the main result of the section.

Proposition 3.18. *For $\mathcal{E} \subseteq \Lambda^s$, the following are equivalent:*

(1) *$\mathcal{E}$ is a maximal clique having finite height,*
(2) *There exists $M \in \Lambda$ such that $\mathcal{E} = \mathscr{T}(M)$.*

Proof. ($1 \Rightarrow 2$) As $\mathcal{E}$ maximal entails $\mathcal{E} \neq \emptyset$, we can proceed by induction on $h = \mathsf{ht}(\mathcal{E})$.

The case $h = 0$ is vacuous because no simple term has height 0.

If $h = 1$ then $t \in \mathcal{E}$ implies $t = [x_1,\dots,x_n]$ since variables are the only resource terms of height 0. Now, $t \coh t$ holds since $\mathcal{E}$ is a clique so the x_i's must be pairwise coherent with each other, but $x_i \coh x_j$ holds if and only if $x_i = x_j$ whence $t = [x_i,\dots,x_i]$ for some index i. From this, and the fact that $\mathcal{E}$ is maximal, we conclude $\mathcal{E} = \mathscr{T}(x_i)$.

Assume $h > 1$ and split into cases depending on the form of $t \in \mathcal{E}$.

- Case $t = [\lambda x.s_1,\dots,\lambda x.s_k]$. Since $\mathsf{ht}(\mathcal{E}) > 1$ we can assume wlog that $t \neq []$, namely $k > 0$. Moreover, since $\mathcal{E}$ is a clique, all $t' \in \mathcal{E}$ must have shape $t' = [\lambda x.s_{k+1},\dots,\lambda x.s_n]$ for some n with $s_i \coh s_j$ for all $i,j \leq n$. It follows that the set $\mathcal{S} = \{s \mid [\lambda x.s] \in \mathcal{E}\}$ is a maximal clique, because $\mathcal{E}$ is maximal, and has height $h-2$ since $\mathsf{ht}([\lambda x.s]) = \mathsf{ht}(s) + 2$. Moreover, $\mathcal{E} = \{[\lambda x.s_1,\dots,\lambda x.s_k] \mid k \geq 0, \forall i \leq k\ .\ s_i \in \mathcal{S}\}$. By induction hypothesis there exists $N \in \Lambda$ such that $\mathcal{S} = \mathscr{T}(N)$, so we get $\mathcal{E} = \mathscr{T}(\lambda x.N)$.
- Otherwise, if $t = s_1 s_2$ then all $t' \in \mathcal{E}$ must be of the form $t' = s_1' s_2'$ with $s_1 \coh s_1'$ and $s_2 \coh s_2'$. So, the set $\mathcal{E}$ can be written as $\mathcal{E} = \mathcal{S}_1\mathcal{S}_2$ where $\mathcal{S}_1 = \{t \mid t s_2 \in \mathcal{E}\}$ and $\mathcal{S}_2 = \{t \mid s_1 t \in \mathcal{E}\}$. As $\mathcal{E}$ is a maximal clique, the sets $\mathcal{S}_1, \mathcal{S}_2$ are independent from the choice of s_2, s_1 (resp.), and they are maximal cliques themselves. Moreover, $\mathsf{ht}(\mathcal{E}) = \mathsf{ht}(\mathcal{S}_1 \cup \mathcal{S}_2) + 1$, whence the heights of $\mathcal{S}_1, \mathcal{S}_2$ are strictly smaller than h. By the induction hypothesis, there exists $P, Q \in \Lambda$ such that $\mathcal{S}_1 = \mathscr{T}(P)$ and $\mathcal{S}_2 = \mathscr{T}(Q)$, from which it follows $\mathcal{E} = \mathscr{T}(PQ)$.

($2 \Rightarrow 1$) We proceed by induction on the structure of M.

If $M = x$ then $t, t' \in \mathscr{T}(M)$ entails $t = [x^k]$ and $t' = [x^n]$ for some $k, n \geq 0$, whence $\mathscr{T}(x)$ is a clique of height 1. It is moreover maximal because it contains $[x^i]$ for all $i \geq 0$.

If $M = \lambda x.N$ then $t, t' \in \mathscr{T}(M)$ entails $t = [\lambda x.t_1,\dots,\lambda x.t_k]$ and $t' = [\lambda x.t_{k+1},\dots,\lambda x.t_n]$ with $t_i \in \mathscr{T}(N)$ for all $i \leq n$. By induction hypothesis $\mathscr{T}(N)$ is a maximal clique of finite height $h \in \mathbb{N}$, in particular $t_i \coh t_j$ for all $i,j \leq n$ which entails $t \coh t'$. The maximality of $\mathscr{T}(M)$ follows from that of $\mathscr{T}(N)$ and, by Lemma 3.17(1), $\mathsf{ht}(\mathscr{T}(M))$ has finite height $h+2$.

If $M = PQ$ then $t, t' \in \mathscr{T}(M)$ entails $t = s_1 t_1$ and $t' = s_2 t_2$ for $s_1, s_2 \in \mathscr{T}(P)$ and $t_1, t_2 \in \mathscr{T}(Q)$. By induction hypothesis, $s_1 \coh s_2$ and $t_1 \coh t_2$ hold and thus $t \coh t'$. Also in this case, the maximality of $\mathscr{T}(M)$ follows from the same property of $\mathscr{T}(P), \mathscr{T}(Q)$. Finally, by induction hypothesis, $\mathsf{ht}(\mathscr{T}(P)) = h_1$ and $\mathsf{ht}(\mathscr{T}(Q)) = h_2$ for $h_1, h_2 \in \mathbb{N}$ then $\mathsf{ht}(\mathscr{T}(M)) = \max\{h_1, h_2\} + 1$ by Lemma 3.17(2), and this concludes the proof. □

4. Computing the Normal Form of the Taylor expansion, and Beyond

The Taylor expansion, as defined in Section 3.2, is a static operation translating a λ-term into an infinite set of simple terms. However, we have seen in Proposition 3.6 that the reduction $\to_{\mathsf{r}}$ is confluent and strongly normalizing. Whence, it is possible to define the normal form of an arbitrary set of resource terms as follows.

Definition 4.1. The r-*normal form* is extended element-wise to any subset $\mathcal{E} \subseteq \Lambda^r$ by setting $\mathrm{NF}(\mathcal{E}) = \bigcup_{e\in\mathcal{E}} \mathrm{nf}_{\mathsf{r}}(e)$.

In particular, $\mathrm{NF}(\Lambda^s)$ (resp. $\mathrm{NF}(\Lambda^r)$, $\mathrm{NF}(\mathrm{Val}^r)$) represents the set of all simple terms (resp. resource terms, resource values) in r-nf generated by the grammar in Definition 3.7. Moreover, $\mathrm{NF}(\mathscr{T}(M))$ is a well-defined subset of $\mathrm{NF}(\Lambda^s)$ for every $M \in \Lambda$ (it can possibly be the empty set, thought).

Example 4.2. We calculate the r-normal form of the Taylor expansions from Example 3.11:
(1) $\mathrm{NF}(\mathscr{T}(\mathtt{I})) = \mathscr{T}(\mathtt{I}) = \{[\lambda x.[x^{n_1}], \dots, \lambda x.[x^{n_k}]] \mid k \geq 0, \forall i \leq k, n_i \geq 0\}$,
(2) $\mathrm{NF}(\mathscr{T}(\Delta)) = \{[\lambda x.[x][x^{m_1}], \dots, \lambda x.[x][x^{m_k}]] \mid k \geq 0, \forall i \leq k, m_i \geq 0\}$,
(3) $\mathrm{NF}(\mathscr{T}(\Delta\mathtt{I})) = \mathrm{NF}(\mathscr{T}(\mathtt{I}))$,
(4) $\mathrm{NF}(\mathscr{T}(\Omega)) = \emptyset$, from this it follows:
(5) $\mathrm{NF}(\mathscr{T}(\lambda x.\Omega)) = \{[]\}$, moreover, for $\mathtt{A} = (\lambda z.(\lambda y.y)(zz))(xx)$, we obtain:
(6) $\mathrm{NF}(\mathscr{T}(\mathtt{A})) = \{[\lambda z.[[\lambda y.[y^{\ell_1}]]([z][z^{m_1}])]([x][x^{n_1}]), \dots, \lambda z.[[\lambda y.[y^{\ell_k}]]([z][z^{m_k}])]([x][x^{n_k}])] \mid k \geq 0, \forall i \leq k, \ell_i, m_i, n_i \geq 0\}$.

On the one hand, it is not difficult to calculate the normal forms of the Taylor expansions of $\mathtt{I}, \Delta$ and $\mathtt{A}$. (As shown in Lemma 3.13, it is enough to perform some 0-reductions.) Similarly, it is not difficult to check that $\mathrm{NF}(\mathscr{T}(\Omega))$ is empty, once realized that no term $t \in \mathscr{T}(\Omega)$ can survive through the reduction. On the other hand, it is more complicated to compute the normal forms of $\mathscr{T}(\mathtt{Z})$, and hence $\mathscr{T}(\mathtt{ZB})$, without having a result connecting such normal forms with the v-reductions of the corresponding λ-terms. The rest of the section is devoted to study such a relationship. We start with some technical lemmas.

Lemma 4.3 (Substitution Lemma)**.** *Let $M \in \Lambda$, $V \in \mathrm{Val}$ and $x \in \mathbb{V}$. Then we have:*

$$\mathscr{T}(M[x := V]) = \bigcup_{t\in\mathscr{T}(M)} \bigcup_{[v_1,\dots,v_n]\in\mathscr{T}(V)} t\langle x := [v_1, \dots, v_n]\rangle.$$

Proof. Straightforward induction on the structure of M. □

Lemma 4.4. *Let $M, N \in \Lambda$ be such that $M \to_{\mathsf{v}} N$. Then:*
(1) *for all $t \in \mathscr{T}(M)$, there exists $\mathcal{T} \subseteq \mathscr{T}(N)$ such that $t \twoheadrightarrow_{\mathsf{r}} \mathcal{T}$,*
(2) *for all $t' \in \mathscr{T}(N)$ such that $t' \not\to_0 \emptyset$, there exist $t \in \mathscr{T}(M)$ and $\mathcal{T} \in \mathscr{P}_{\mathrm{f}}(\Lambda^s)$ satisfying $t \twoheadrightarrow_{\mathsf{r}} \{t'\} \cup \mathcal{T}$. Moreover such a t is unique.*

Proof. We check that both (1) and (2) hold by induction on a derivation of $M \to_{\mathsf{v}} N$, splitting into cases depending on the kind of redex is reduced.

(β_v): If $M = (\lambda x.Q)V$ and $N = Q[x := V]$ then items (1) and (2) follow by Lemma 4.3.
(σ_1): If $M = (\lambda x.M')PQ$ and $N = (\lambda x.M'Q)P$ then

$$\begin{array}{rcl}\mathscr{T}(M) &=& \{[\lambda x.t_1, \dots, \lambda x.t_n]s_1s_2 \mid n \geq 0, t_i \in \mathscr{T}(M'), s_1 \in \mathscr{T}(P), s_2 \in \mathscr{T}(Q)\},\\ \mathscr{T}(N) &=& \{[\lambda x.t'_1s'_1, \dots, \lambda x.t'_ns'_n]s \mid n \geq 0, t'_i \in \mathscr{T}(M'), s'_i \in \mathscr{T}(Q), s \in \mathscr{T}(P)\}.\end{array}$$

For $n \neq 1$, we have $[\lambda x.t_1, \dots, \lambda x.t_n]s_1 s_2 \to_0 \emptyset \subseteq \mathscr{T}(N)$. For $n = 1$, we get $[\lambda x.t_1]s_1 s_2 \to_{\sigma_1} [\lambda x.t_1 s_2]s_1$ for $t_1 \in \mathscr{T}(M')$, $s_1 \in \mathscr{T}(P)$ and $s_2 \in \mathscr{T}(Q)$, whence $[\lambda x.t_1 s_2]s_1 \in \mathscr{T}(N)$ and (1) holds. Concerning (2), note that $[\lambda x.t'_1 s'_1, \dots, \lambda x.t'_n s'_n]s \not\to_0 \emptyset$ entails $n = 1$. Moreover, $\mathscr{T}(M) \ni [\lambda x.t'_1]ss'_1 \to_{\sigma_1} [\lambda x.t'_1 s'_1]s$ since $t'_1 \in \mathscr{T}(M')$, $s'_1 \in \mathscr{T}(Q)$, $s \in \mathscr{T}(P)$.

(σ_3): If $M = V((\lambda x.P)Q)$ for $V \in \mathrm{Val}$ and $N = (\lambda x.VP)Q$ then

$$\mathscr{T}(M) = \{[v_1, \dots, v_n]([\lambda x.s_1, \dots, \lambda x.s_m]s) \mid \begin{array}{l} n \geq 0, [v_1, \dots, v_n] \in \mathscr{T}(V), \\ i \leq m,\ s_i \in \mathscr{T}(P), s \in \mathscr{T}(Q)\}, \end{array}$$

$$\mathscr{T}(N) = \{[\lambda x.[v_{11}, \dots, v_{1k_1}]s_1, \dots, \lambda x.[v_{n1}, \dots, v_{nk_n}]s_n]s \mid \begin{array}{l} n \geq 0,\ i \leq n, \\ s_i \in \mathscr{T}(P), s \in \mathscr{T}(Q), \\ [v_{i1}, \dots, v_{ik_i}] \in \mathscr{T}(V)\}, \end{array}$$

For $m \neq 1$ $n \neq 1$, we have $[v_1, \dots, v_n]([\lambda x.s_1, \dots, \lambda x.s_m]s) \to_0 \emptyset \subseteq \mathscr{T}(N)$. For $m = n = 1$, we get $[v_1]([\lambda x.s_1]s) \to_{\sigma_3} [\lambda x.[v_1]s_1]s \in \mathscr{T}(N)$, so (1) holds. Similarly, we have that $[\lambda x.[v_{11}, \dots, v_{1k_1}]s_1, \dots, \lambda x.[v_{n1}, \dots, v_{nk_n}]s_n]s \to_0 \emptyset$ whenever $n \neq 1$ or $k_i \neq 1$. For $n = k_1 = 1$, we get $\mathscr{T}(M) \ni [v_{11}]([\lambda x.s_1]s) \to_{\sigma_3} [\lambda x.[v_{11}]s_1]s$ which proves (2).

In the cases above it is easy to check that t is actually unique. The contextual cases follow straightforwardly from the induction hypothesis. □

As a consequence, we obtain the analogue of Proposition 2.11 for Taylor expansions.

Corollary 4.5. *For $M, N \in \Lambda$, $M =_{\mathsf{v}} N$ entails* $\mathrm{NF}(\mathscr{T}(M)) = \mathrm{NF}(\mathscr{T}(N))$.

Proof. It is enough to prove $\mathrm{NF}(\mathscr{T}(M)) = \mathrm{NF}(\mathscr{T}(N))$ for M and N such that $M \to_{\mathsf{v}} N$, indeed the general result follows by confluence of v-reduction. We show the two inclusions.

($\subseteq$) Consider $t \in \mathrm{NF}(\mathscr{T}(M))$, then there exists $t_0 \in \mathscr{T}(M)$ and $\mathcal{T} \in \mathscr{P}_{\mathrm{f}}(\Lambda^s)$ such that $t_0 \twoheadrightarrow_{\mathsf{r}} \{t\} \cup \mathcal{T}$. Since $\lambda^\sigma_{\mathsf{r}}$ is strongly normalizing (Proposition 3.6), we assume wlog $\mathcal{T}$ in r-nf. By Lemma 4.4(1), we have $t_0 \twoheadrightarrow_{\mathsf{r}} \mathcal{T}_0 \subseteq \mathscr{T}(N)$ so by confluence of $\to_r$ we get $\mathcal{T}_0 \twoheadrightarrow_{\mathsf{r}} \{t\} \cup \mathcal{T}$ which entails $t \in \mathrm{NF}(\mathscr{T}(N))$ because t is in r-nf.

($\supseteq$) If $t \in \mathrm{NF}(\mathscr{T}(N))$ then there are $s \in \mathscr{T}(N)$ and $\mathcal{T} \in \mathscr{P}_{\mathrm{f}}(\Lambda^s)$ such that $s \twoheadrightarrow_{\mathsf{r}} \{t\} \cup \mathcal{T}$. By Lemma 4.4(2), there exists $s_0 \in \mathscr{T}(M)$ and $\mathcal{S} \in \mathscr{P}_{\mathrm{f}}(\Lambda^s)$ satisfying $s_0 \twoheadrightarrow_{\mathsf{r}} \{s\} \cup \mathcal{S}$. Composing the two reductions we get $s_0 \twoheadrightarrow_{\mathsf{r}} \{t\} \cup \mathcal{S} \cup \mathcal{T}$, thus $t \in \mathrm{NF}(\mathscr{T}(M))$ as well. □

We now prove a Context Lemma for Taylor expansions in the spirit of [Bar84, Cor. 14.3.20] (namely, the Context Lemma for CbN Böhm trees). For the sake of simplicity, in the next lemma we consider head contexts but the same reasoning works for arbitrary contexts.

Lemma 4.6 (Context Lemma for Taylor expansions)**.** *Let $M, N \in \Lambda$. If* $\mathrm{NF}(\mathscr{T}(M)) = \mathrm{NF}(\mathscr{T}(N))$ *then, for all head contexts $C(\!|-|\!)$, we have* $\mathrm{NF}(\mathscr{T}(C(\!|M|\!))) = \mathrm{NF}(\mathscr{T}(C(\!|N|\!)))$.

Proof. Consider $C(\!|-|\!) = (\lambda x_1 \dots x_n.(\!|-|\!))V_1 \cdots V_k$ for $n, k \geq 0$. Let us take $t \in \mathrm{NF}(\mathscr{T}(C(\!|M|\!)))$ and prove that t belongs to $\mathrm{NF}(\mathscr{T}(C(\!|N|\!)))$, the other inclusion being symmetrical. Then there exists $t_0 \in \mathscr{T}(C(\!|M|\!))$ and $\mathcal{T} \in \mathscr{P}_{\mathrm{f}}(\mathrm{NF}(\Lambda^s))$ such that $t_0 \twoheadrightarrow_{\mathsf{r}} \{t\} \cup \mathcal{T}$. By definition of $C(\!|-|\!)$ and $\mathscr{T}(-)$, t_0 must have the following shape:

$$t_0 = [\lambda x_1.[\cdots[\lambda x_n.s]\cdots]][v_{11}, \dots, v_{1n_1}]\cdots[v_{k1}, \dots, v_{kn_k}]$$

where $s \in \mathscr{T}(M)$, $[v_{i1}, \dots, v_{in_i}] \in \mathscr{T}(V_i)$ where $1 \leq i \leq k$ and $n_i = \deg_{x_i}(s)$ for otherwise $t_0 \to_0 \emptyset$, which is impossible. By confluence and strong normalization of $\to_{\mathsf{r}}$ (Proposition 3.6), the reduction $t_0 \twoheadrightarrow_{\mathsf{r}} \{t\} \cup \mathcal{T}$ factorizes as $t_0 \twoheadrightarrow_{\mathsf{r}} \mathcal{T}_0 \twoheadrightarrow_{\mathsf{r}} \{t\} \cup \mathcal{T}$ where

$$\mathcal{T}_0 = [\lambda x_1.[\cdots[\lambda x_n.\mathrm{nf}_{\mathsf{r}}(s)]\cdots]][v_{11}, \dots, v_{1n_1}]\cdots[v_{k1}, \dots, v_{kn_k}]$$

and $\mathrm{nf}_{\mathsf{r}}(s) \in \mathscr{P}_{\mathrm{f}}(\mathrm{NF}(\mathscr{T}(M)))$. By hypothesis $\mathrm{nf}_{\mathsf{r}}(s) \in \mathscr{P}_{\mathrm{f}}(\mathrm{NF}(\mathscr{T}(N)))$, therefore there are $\mathcal{S}_1 \in \mathscr{P}_{\mathrm{f}}(\mathscr{T}(N))$ such that $\mathcal{S}_1 \twoheadrightarrow_{\mathsf{r}} \mathrm{nf}_{\mathsf{r}}(s) \cup \mathcal{S}'$, for some $\mathcal{S}'$, and $\mathcal{S}_0 \subseteq \mathscr{T}(C(\!|N|\!))$ of shape

$$\mathcal{S}_0 = [\lambda x_1.[\cdots[\lambda x_n.\mathcal{S}_1]\cdots]][v_{11},\dots,v_{1n_1}]\cdots[v_{k1},\dots,v_{kn_k}]$$

so we conclude, for some $\mathcal{S}''$, that $\mathcal{S}_0 \twoheadrightarrow_{\mathsf{r}} \mathcal{T}_0 \cup \mathcal{S}'' \twoheadrightarrow_{\mathsf{r}} \{t\} \cup \mathcal{T} \cup \mathrm{nf}_{\mathsf{r}}(\mathcal{S}'') \subseteq \mathrm{NF}(\mathscr{T}(C(\!|N|\!)))$. □

4.1. **Taylor expanding Böhm trees.** The Taylor expansion can be extended to elements of $\Lambda_\perp$ by adding $\mathscr{T}(\perp) = \{[]\}$ to the rules of Definition 3.9. However, the resulting translation of an approximant A produces a set of resource terms that are not necessarily in r-normal form because of the presence of (0)-redexes (as already discussed in Remark 3.12). Luckily, it is possible to slightly modify such a definition by performing an "on the flight" normalization and obtain directly the normalized Taylor expansion of a Böhm tree.

Definition 4.7. (1) Let $A \in \mathcal{A}$. The *normalized Taylor expansion of* A, in symbols $\mathscr{T}^\circ(A)$, is defined by structural induction following the grammar of Definition 2.1(1):

$$\begin{array}{rcl}
\mathscr{T}^\circ(x) & = & \{[x^n] \mid n \geq 0\},\\
\mathscr{T}^\circ(\lambda x.A') & = & \{[\lambda x.t_1,\dots,\lambda x.t_n] \mid n \geq 0, \forall i \leq k \;.\; t_i \in \mathscr{T}^\circ(A')\},\\
\mathscr{T}^\circ(\perp) & = & \{[]\},\\
\mathscr{T}^\circ(xBA_1\cdots A_k) & = & \{[x]t_0\cdots t_n \mid t_0 \in \mathscr{T}^\circ(B), \forall 1 \leq i \leq k \;.\; t_i \in \mathscr{T}^\circ(A_i)\},\\
\mathscr{T}^\circ((\lambda x.A')(yBA_1\cdots A_k)) & = & \{[\lambda x.s]t \mid s \in \mathscr{T}^\circ(A'),\; t \in \mathscr{T}^\circ(yBA_1\cdots A_k)\}.
\end{array}$$

(2) The *normalized Taylor expansion of* $\mathrm{BT}(M)$, written $\mathscr{T}^\circ(\mathrm{BT}(M))$, is defined by setting:

$$\mathscr{T}^\circ(\mathrm{BT}(M)) = \bigcup_{A \in \mathcal{A}(M)} \mathscr{T}^\circ(A)$$

Example 4.8. (1) Recall from Example 2.2(1) that $\mathcal{A}(\mathtt{I}) = \{\perp, \lambda x.\perp, \lambda x.x\}$, therefore

$$\mathscr{T}^\circ(\mathcal{A}(\mathtt{I})) = \{[]\} \cup \{[(\lambda x.[])^k] \mid k \geq 0\} \cup \{[\lambda x.[x^{n_1}],\dots,\lambda x.[x^{n_k}]] \mid k, n_1,\dots,n_k \geq 0\}$$

By Example 4.2(1) this is equal to $\mathrm{NF}(\mathscr{T}(\mathtt{I}))$.

(2) Since $\mathcal{A}(\Omega) = \emptyset$ we have $\mathscr{T}^\circ(\mathcal{A}(\Omega)) = \emptyset = \mathrm{NF}(\mathscr{T}(\Omega))$.

(3) Also, $\mathcal{A}(\Delta) = \{\perp, \lambda x.\perp, \lambda x.xx\}$, so that

$$\mathscr{T}^\circ(\mathcal{A}(\Delta)) = \{[]\} \cup \{[(\lambda x.[])^k] \mid k \geq 0\} \cup \{[\lambda x.[x][x^{n_1}],\dots,\lambda x.[x][x^{n_k}]] \mid k, n_1,\dots,n_k \geq 0\}$$

By Example 4.2(2) this is equal to $\mathrm{NF}(\mathscr{T}(\Delta))$.

(4) Finally, Examples 2.2(3) and 4.2(3) and the above item (1) give us $\mathscr{T}^\circ(\mathcal{A}(\Delta\mathtt{I})) = \mathscr{T}^\circ(\mathcal{A}(\mathtt{I})) = \mathrm{NF}(\mathscr{T}(\mathtt{I})) = \mathrm{NF}(\mathscr{T}(\Delta\mathtt{I}))$.

The rest of the section is devoted to generalizing the above example, proving that the normal form of the Taylor expansion of any λ-term M is equal to the normalized Taylor expansion of the Böhm tree of M (Theorem 4.11). On the one side, this link is extremely useful to compute $\mathrm{NF}(\mathscr{T}(M))$ because the Böhm trees have the advantage of hiding the explicit amounts of resources that can become verbose and difficult to handle. On the other side, this allow to transfer results from the Taylor expansions to Böhm trees, Lemma 4.15 being a paradigmatic example.

Lemma 4.9. *Let $M \in \Lambda$.*

(1) *If $t \in \mathscr{T}(M)$ and $t \to_{\mathsf{r}} \{t_1\} \cup \mathcal{T}_1$, then there exists $N \in \Lambda$ and $\mathcal{T}_2 \in \mathscr{P}_{\mathrm{f}}(\Lambda^s)$, such that $M \to_{\mathsf{v}} N$ and $\{t_1\} \cup \mathcal{T}_1 \twoheadrightarrow_{\mathsf{r}} \mathcal{T}_2 \subseteq \mathscr{T}(N)$.*
(2) *If $t \in \mathrm{NF}(\mathscr{T}(M))$ then there exists M' such that $M \twoheadrightarrow_{\mathsf{v}} M'$ and $t \in \mathscr{T}(M')$.*
(3) *If $t, s \in \mathscr{T}(M)$ then $\mathrm{NF}(t) \cap \mathrm{NF}(s) \neq \emptyset$ entails $t = s$.*
(4) *If $t \in \mathscr{T}(M) \cap \mathrm{NF}(\Lambda^s)$ then there exists $A \in \mathcal{A}$ such that $A \sqsubseteq M$ and $t \in \mathscr{T}^{\circ}(A)$.*

Proof. (1) Note that $t \to_{\mathsf{r}} \{t_1\} \cup \mathcal{T}_1$ by contracting an r-redex arising from an occurrence of a v-redex in M, so $M \to_{\mathsf{v}} N$ where N is obtained by contracting such a redex occurrence. By Lemma 4.4(1) and confluence of $\to_{\mathsf{r}}$, there exists $\mathcal{T}_2 \subseteq \mathscr{T}(N)$ such that $t \twoheadrightarrow_{\mathsf{r}} \{t_1\} \cup \mathcal{T}_1 \twoheadrightarrow_{\mathsf{r}} \mathcal{T}_2$.

(2) Assume that $t \in \mathrm{NF}(\mathscr{T}(M))$, then there are $t_0 \in \mathscr{T}(M)$ and $\mathcal{T} \in \mathscr{P}_{\mathrm{f}}(\Lambda^s)$ such that $t_0 \twoheadrightarrow_{\mathsf{r}} \{t\} \cup \mathcal{T}$. Since $\to_{\mathsf{r}}$ is strongly normalizing, we can assume $\mathcal{T} \subseteq \mathrm{NF}(\Lambda^s)$ and choose such a reduction to have maximal length n. We proceed by induction on n to show that the λ-term M' exists. If $n = 0$ then t_0 is in r-nf so just take $t_0 = t$, $\mathcal{T} = \emptyset$ and $M = M'$. Otherwise $n > 0$ and $t_0 \to_{\mathsf{r}} \{t_1\} \cup \mathcal{T}_1 \twoheadrightarrow_{\mathsf{r}} \{t\} \cup \mathcal{T}$ where the second reduction is strictly shorter. By (1) and confluence there exists N such that $M \to_{\mathsf{v}} N$ and $\{t_1\} \cup \mathcal{T}_1 \twoheadrightarrow_{\mathsf{r}} \mathcal{T}_2 \twoheadrightarrow_{\mathsf{r}} \{t\} \cup \mathcal{T}$ for some $\mathcal{T}_2 \subseteq \mathscr{T}(N)$. So, there are $t_2 \in \mathcal{T}_2$ and $\mathcal{T}' \in \mathscr{P}_{\mathrm{f}}(\Lambda^s)$ such that $t_2 \twoheadrightarrow_{\mathsf{r}} \{t\} \cup \mathcal{T}' \subseteq \mathrm{NF}(\mathscr{T}(M))$ so we conclude by applying the induction hypothesis to this reduction shorter than n.

(3) Assume $t_0 \in \mathrm{NF}(t) \subseteq \mathrm{NF}(\mathscr{T}(M))$. By (2) there is a reduction $M \to_{\mathsf{v}} M_1 \to_{\mathsf{v}} \cdots \to_{\mathsf{v}} M_k$ such that $t_0 \in \mathscr{T}(M_k)$. By an iterated application of Lemma 4.4(2), we get that t is the unique element in $\mathscr{T}(M)$ generating t_0. Therefore, $t_0 \in \mathrm{NF}(s)$ entails $s = t$.

(4) By structural induction on the normal structure of t (characterized in Lemma 3.16: notice that $t \subset t$ by Proposition 3.18).

If $t = []$ then $M \in \mathrm{Val}$ and there are two subcases: either $M = x$, or $M = \lambda x.M'$ so we simply take $A = \bot$. Similarly, if $t = [x, \dots, x]$ ($n > 0$ occurrences) then $M = A = x$.

If $t = [\lambda x.a_1, \dots, \lambda x.a_n]$ with $n > 0$ then $M = \lambda x.M'$ and $a_i \in \mathscr{T}(M')$ for $i \leq n$. By induction hypothesis, there are approximants $A_i \sqsubseteq M'$ such that $a_i \in \mathscr{T}^{\circ}(A_i)$. Then we set $A = \lambda x.A'$ for $A' = A_1 \sqcup \cdots \sqcup A_n$ which exists because the A_i's are pairwise compatible.

If $t = [x]ba_1 \cdots a_k$ then $M = xM_0 \cdots M_k$ and $b \in \mathscr{T}(M_0)$ and $a_j \in \mathscr{T}(M_i)$ for $1 \leq j \leq k$. By induction hypothesis, there are $A_0, \dots, A_k$ such that $A_i \sqsubseteq M_i$ for all i ($0 \leq i \leq k$), $b \in \mathscr{T}^{\circ}(A_0)$ and $a_j \in \mathscr{T}^{\circ}(A_j)$ for $1 \leq j \leq k$. Moreover $b \in \mathscr{T}^{\circ}(A_0)$ entails that A_0 is a B-term from the grammar in Lemma 1.7, therefore we may take $A = xA_0 \cdots A_k \in \mathcal{A}$.

Finally, if $t = [\lambda x.a]([y]ba_1 \cdots a_k)$ then $M = (\lambda x.M')(yM_0 \cdots M_k)$ with $a \in \mathscr{T}(M')$ and $[y]ba_1 \cdots a_k \in \mathscr{T}(yM_0 \cdots M_k)$. Reasoning as in the previous case, we get $yA_0 \cdots A_k \in \mathcal{A}$ such that $[y]ba_1 \cdots a_k \in \mathscr{T}^{\circ}(yA_0 \cdots A_k)$. Moreover, by induction hypothesis, there is $A' \sqsubseteq M'$ such that $a \in \mathscr{T}^{\circ}(A')$. We conclude by taking $A = (\lambda x.A')(yA_0 \cdots A_k)$. $\square$

Lemma 4.10. *Let $M \in \Lambda$ and $A \in \mathcal{A}$.*

(1) *If $A \sqsubseteq M$ then $\mathscr{T}^{\circ}(A) \subseteq \mathrm{NF}(\mathscr{T}(M))$.*
(2) *If $\mathscr{T}^{\circ}(A) \subseteq \mathscr{T}^{\circ}(\mathrm{BT}(M))$ then $A \in \mathcal{A}(M)$.*

Proof. (1) If $A = \bot$ then $M \in \mathrm{Val}$ and $\mathscr{T}^{\circ}(\bot) = \{[]\} \subseteq \mathscr{T}(M) \cap \mathrm{NF}(\Lambda^r)$.

Otherwise, it follows by induction on A exploiting the fact that all simple terms in $\mathscr{T}^{\circ}(A)$ belong to $\mathscr{T}(M)$ and are already in r-nf.

(2) We proceed by structural induction on A, the case $A = \bot$ being trivial.

- If $A = x$ then $\mathscr{T}^{\circ}(x) = \{[x^n] \mid n \geq 0\} \subseteq \mathscr{T}^{\circ}(\mathrm{BT}(M))$ entails $M \twoheadrightarrow_{\mathsf{v}} x$ and we are done.
- If $A = \lambda x.A'$ then $\mathscr{T}^{\circ}(\lambda x.A') = \{[\lambda x.t_1, \dots, \lambda x.t_n] \mid n \geq 0, \forall i \leq k \ . \ t_i \in \mathscr{T}^{\circ}(A')\}$. So, $\mathscr{T}^{\circ}(\lambda x.A') \subseteq \mathscr{T}^{\circ}(\mathrm{BT}(M))$ implies that $M \twoheadrightarrow_{\mathsf{v}} \lambda x.M'$ for some M' such that $\mathscr{T}^{\circ}(A') \subseteq$

$\mathscr{T}^\circ(\mathrm{BT}(M'))$. By induction hypothesis, we get $A' \in \mathcal{A}(M')$ and $\lambda x.A' \in \mathcal{A}(\lambda x.M')$. By Lemma 2.6 we obtain $\lambda x.A' \in \mathcal{A}(M)$ as desired.

- If $A = \bot$, then $\mathscr{T}^\circ(A) = \{[]\} \subseteq \mathscr{T}^\circ(\mathrm{BT}(M))$ entails $M \twoheadrightarrow_{\mathsf{v}} V$ for some value V, therefore we get $\bot \in \mathcal{A}(V)$ and we conclude by Lemma 2.6.
- If $A = xBA'_1 \cdots A'_k$ then $\mathscr{T}^\circ(A) = \{[x]t_0 \cdots t_n \mid t_0 \in \mathscr{T}^\circ(B), \forall 1 \le i \le k \ .\ t_i \in \mathscr{T}^\circ(A'_i)\}$. In this case, we must have $M \twoheadrightarrow_{\mathsf{v}} xM_0 \cdots M_k$ with $\mathscr{T}^\circ(B) \subseteq \mathscr{T}^\circ(\mathrm{BT}(M_0))$ and $\mathscr{T}^\circ(A'_i) \subseteq \mathscr{T}^\circ(\mathrm{BT}(M_i))$ for $1 \le i \le k$. By induction hypothesis $B \in \mathcal{A}(M_0)$ and $A'_i \in \mathcal{A}(M_i)$ $\forall i \in \{1, \dots, k\}$, thus $xBA'_1 \cdots A'_k \in \mathcal{A}(xM_0 \cdots M_k) = \mathcal{A}(M)$ by Lemma 2.6.
- If $A = (\lambda x.A')(yBA'_1 \cdots A'_k)$, then $\mathscr{T}^\circ(A) = \{[\lambda x.s]t \mid s \in \mathscr{T}^\circ(A'),\ t \in \mathscr{T}^\circ(yBA'_1 \cdots A'_k)\}$. In this case we get $M \twoheadrightarrow_{\mathsf{v}} (\lambda x.M')(yM_0 \cdots M_k)$ with $\mathscr{T}^\circ(A') \subseteq \mathscr{T}^\circ(\mathrm{BT}(M'))$ and $\mathscr{T}^\circ(yBA'_1 \cdots A'_k) \subseteq \mathscr{T}^\circ(\mathrm{BT}(yM_0 \cdots M_k))$. By applying the induction hypothesis, we obtain $A \in \mathcal{A}((\lambda x.M')(yM_0 \cdots M_k))$ and once again we conclude by Lemma 2.6. □

The following constitutes the main result of the section, relating Böhm trees and Taylor expansion in the spirit of [ER06].

Theorem 4.11. *For all $M \in \Lambda$, we have $\mathscr{T}^\circ(\mathrm{BT}(M)) = \mathrm{NF}(\mathscr{T}(M))$.*

Proof. ($\subseteq$) Take $t \in \mathscr{T}^\circ(\mathrm{BT}(M))$, then there exists an approximant $A' \in \mathcal{A}(M)$ such that $t \in \mathscr{T}^\circ(A')$. As $A' \in \mathcal{A}(M)$, there is $M' \in \Lambda$ such that $M \twoheadrightarrow_{\mathsf{v}} M'$ and $A' \sqsubseteq M'$. We can therefore apply Lemma 4.10(1) to conclude that $t \in \mathrm{NF}(\mathscr{T}(M'))$, which is equal to $\mathrm{NF}(\mathscr{T}(M))$ by Lemma 4.5.

($\supseteq$) Assume $t \in \mathrm{NF}(\mathscr{T}(M))$. By Lemma 4.9(2) there exists $M' \in \Lambda$ such that $M \twoheadrightarrow_{\mathsf{v}} M'$ and $t \in \mathscr{T}(M')$. By Lemma 4.9(4), there is $A \sqsubseteq M'$ such that $t \in \mathscr{T}^\circ(A)$. By the conditions above we have $A \in \mathcal{A}(M)$, so we conclude that $t \in \mathscr{T}^\circ(\mathrm{BT}(M))$. □

4.2. **Consequences of the main theorem.** The rest of the section is devoted to present some interesting consequences of Theorem 4.11.

Corollary 4.12. *For $M, N \in \Lambda$, the following are equivalent:*
(1) $\mathrm{BT}(M) = \mathrm{BT}(N)$,
(2) $\mathrm{NF}(\mathscr{T}(M)) = \mathrm{NF}(\mathscr{T}(N))$.

Proof. ($1 \Rightarrow 2$) If M, N have the same Böhm tree, we can apply Theorem 4.11 to get

$$\mathrm{NF}(\mathscr{T}(M)) = \mathscr{T}^\circ(\mathrm{BT}(M)) = \mathscr{T}^\circ(\mathrm{BT}(N)) = \mathrm{NF}(\mathscr{T}(N)).$$

($1 \Leftarrow 2$) We assume $\mathrm{NF}(\mathscr{T}(M)) = \mathrm{NF}(\mathscr{T}(N))$ and start showing $\mathcal{A}(M) \subseteq \mathcal{A}(N)$. Take any $A \in \mathcal{A}(M)$, by definition we have $\mathscr{T}^\circ(A) \subseteq \mathscr{T}^\circ(\mathrm{BT}(M))$, so Lemma 4.10(2) entails $A \in \mathrm{BT}(N)$. The converse inclusion being symmetrical, we get $\mathcal{A}(M) = \mathcal{A}(N)$ which in its turn entails $\mathrm{BT}(M) = \mathrm{BT}(N)$ by Remark 2.9. □

Carraro and Guerrieri showed in [CG14] that the relational model $\mathscr{U}$ of CbV λ-calculus and resource calculus introduced by Ehrhard in [Ehr12] satisfies the σ-rules, so it is actually a model of both λ_v^σ and λ_r^σ. They also prove that $\mathscr{U}$ satisfies the Taylor expansion in the following technical sense (where $[\![-]\!]$ represents the interpretation function in $\mathscr{U}$):

$$[\![M]\!] = \bigcup_{t \in \mathscr{T}(M)} [\![t]\!] \tag{4.1}$$

As a consequence, we get that the theory of the model $\mathscr{U}$ is included in the theory equating all λ-terms having the same Böhm trees. We conjecture that the two theories coincide.

Theorem 4.13. *For $M, N \in \Lambda$, we have:*

$$\mathrm{BT}(M) = \mathrm{BT}(N) \;\Rightarrow\; [\![M]\!] = [\![N]\!].$$

Proof. Indeed, we have the following chain of equalities:

$$\begin{array}{rll} [\![M]\!] = & \bigcup_{t \in \mathscr{T}(M)} [\![t]\!], & \text{by (4.1)}, \\ = & \bigcup_{t \in \mathrm{NF}(\mathscr{T}(M))} [\![t]\!], & \text{as } [\![t]\!] = \bigcup_{s \in \mathrm{nf}_{\mathsf{r}}(t)} [\![s]\!], \\ = & \bigcup_{t \in \mathscr{T}^{\circ}(\mathrm{BT}(M))} [\![t]\!], & \text{by Theorem 4.11}, \\ = & \bigcup_{t \in \mathscr{T}^{\circ}(\mathrm{BT}(N))} [\![t]\!], & \text{as } \mathrm{BT}(M) = \mathrm{BT}(N), \\ = & \bigcup_{t \in \mathrm{NF}(\mathscr{T}(N))} [\![t]\!], & \text{by Theorem 4.11}, \\ = & \bigcup_{t \in \mathscr{T}(N)} [\![t]\!], & \text{as } [\![t]\!] = \bigcup_{s \in \mathrm{nf}_{\mathsf{r}}(t)} [\![s]\!], \\ = & [\![N]\!], & \text{by (4.1)}. \end{array}$$

This concludes the proof. □

In the paper [CG14], the authors also prove that $[\![M]\!] \neq \emptyset$ exactly when M is potentially valuable (Definition 1.9). From this result, we obtain easily the lemma below.

Theorem 4.14. *For $M \in \Lambda$, the following are equivalent:*
(1) *M is potentially valuable,*
(2) *$\mathrm{BT}(M) \neq \bot$.*

Proof. It is easy to check that all resource approximants t have non-empty interpretation in $\mathscr{U}$, i.e. $t \in \mathrm{NF}(\Lambda^s)$ entails $[\![t]\!] \neq \emptyset$. Therefore we have the following chain of equivalences:

$$\begin{array}{rcll} M \text{ potentially valuable} & \iff & [\![M]\!] \neq \emptyset, & \text{by [CG14, Thm. 24]}, \\ & \iff & \exists t \in \mathrm{NF}(\mathscr{T}(M)), [\![t]\!] \neq \emptyset & \text{by (4.1)}, \\ & \iff & \exists s \in \mathscr{T}^{\circ}(\mathrm{BT}(M)), [\![s]\!] \neq \emptyset & \text{by Theorem 4.11}, \\ & \iff & \exists A \in \mathcal{A}(M), A \neq \bot & \end{array}$$

This is equivalent to say that $\mathrm{BT}(M) \neq \bot$. □

After this short, but fruitful, semantical digression we conclude proving that all λ-terms having the same Böhm tree are indistinguishable from an observational point of view. As in the CbN setting, also in CbV this result follows from the Context Lemma for Böhm trees. The classical proof of this lemma in CbN is obtained by developing an interesting, but complicated, theory of syntactic continuity (see [Bar84, §14.3] and [AC98, §2.4]). Here we bypass this problem completely, and obtain such a result as a corollary of the Context Lemma for Taylor expansions by applying Theorem 4.11.

Lemma 4.15 (Context Lemma for Böhm trees)**.** *Let $M, N \in \Lambda$. If $\mathrm{BT}(M) = \mathrm{BT}(N)$ then, for all head contexts $C(\!|-|\!)$, we have $\mathrm{BT}(C(\!|M|\!)) = \mathrm{BT}(C(\!|N|\!))$.*

Proof. It follows from the Context Lemma for Taylor expansions (Lemma 4.6) by applying Theorem 4.11 and Corollary 4.12. □

As mentioned in the discussion before Lemma 4.6, both the statement and the proof generalize to arbitrary contexts. Thanks to Remark 1.11, we only need head contexts in order to prove the following theorem stating that the Böhm tree model defined in this paper is adequate for Plotkin's CbV λ-calculus.

Theorem 4.16. *Let $M, N \in \Lambda$. If $\mathrm{BT}(M) = \mathrm{BT}(N)$ then $M \equiv N$.*

Proof. Assume, by the way of contradiction, that $\mathrm{BT}(M) = \mathrm{BT}(N)$ but $M \not\equiv N$. Then, there exists a head context $C(\!|-|\!)$ such that $C(\!|M|\!), C(\!|N|\!) \in \Lambda^o$ and, say, $C(\!|M|\!)$ is valuable while $C(\!|N|\!)$ is not. Since they are closed λ-terms, this is equivalent to say that $C(\!|M|\!)$ is potentially valuable while $C(\!|N|\!)$ is not. By Theorem 4.14, $\mathrm{BT}(C(\!|M|\!)) \neq \bot$ and $\mathrm{BT}(C(\!|N|\!)) = \bot$. As a consequence, we obtain $\mathrm{BT}(C(\!|M|\!)) \neq \mathrm{BT}(C(\!|N|\!))$ thus contradicting the Context Lemma for Böhm trees (Lemma 4.15). □

Notice that the converse implication does not hold — for instance it is easy to check that $\Delta(yy) \equiv yy(yy)$ holds, but the two λ-terms have different Böhm trees.

5. Conclusions

Inspired by the work of Ehrhard [Ehr12], Carraro and Guerrieri [CG14], we proposed a notion of Böhm tree for Plotkin's call-by-value λ-calculus λ_v, having a strong mathematical background rooted in Linear Logic. We proved that CbV Böhm trees provide a syntactic model of λ_v which is adequate (in the sense expressed by Theorem 4.16) but not fully abstract — there are operationally indistinguishable λ-terms having different Böhm trees. The situation looks similar in call-by-name where one needs to consider Böhm trees up to possibly infinite η-expansions to capture the λ-theory $\mathcal{H}^*$ and obtain a fully abstract model [Bar84, Cor. 19.2.10]. Developing a notion of extensionality for CbV Böhm trees is certainly interesting, as it might help to describe the equational theory of some extensional denotational model, and a necessary step towards full abstraction. Contrary to what happens in call-by-name, this will not be enough to achieve full abstraction as shown by the counterexample $\Delta(yy) \equiv yy(yy)$ but $\mathrm{BT}(\Delta(yy)) \neq \mathrm{BT}(yy(yy))$, where extensionality plays no role. The second and third authors, together with Ronchi Della Rocca, recently introduced in [MRP19] a new class of adequate models of λ_v^σ and showed that they validate not only $=_{\mathsf{v}}$ but also some I-reductions (in the sense of λI-calculus [Bar84, Ch. 9]) preserving the operational semantics of λ-terms. Finding a precise characterization of those I-redexes that can be safely contracted in the construction of a CbV Böhm tree is a crucial open problem that can lead to full abstraction.

Acknowledgment

The authors wish to acknowledge fruitful discussions with Giulio Guerrieri, Luca Paolini and Simona Ronchi della Rocca.

References

[AC98] Roberto Amadio and Pierre-Louis Curien. *Domains and Lambda Calculi*. Cambridge tracts in theoretical computer science. Cambridge University Press, 1998.

[AG17] Beniamino Accattoli and Giulio Guerrieri. Implementing open call-by-value (extended version). *CoRR*, abs/1701.08186, 2017.

[Bar77] Henk P. Barendregt. The type free lambda calculus. In Jon Barwise, editor, *Handbook of Mathematical Logic*, volume 90 of *Studies in Logic and the Foundations of Mathematics*, pages 1091 – 1132. Elsevier, 1977.

[Bar84] Henk P. Barendregt. *The lambda-calculus, its syntax and semantics*. Number 103 in Studies in Logic and the Foundations of Mathematics. North-Holland, second edition, 1984.

[BDS13] Henk P. Barendregt, Wil Dekkers, and Richard Statman. *Lambda Calculus with Types*. Perspectives in logic. Cambridge University Press, 2013.

[BHP13] Pierre Boudes, Fanny He, and Michele Pagani. A characterization of the Taylor expansion of lambda-terms. In Simona Ronchi Della Rocca, editor, *Computer Science Logic 2013 (CSL 2013), CSL 2013, September 2-5, 2013, Torino, Italy*, volume 23 of *LIPIcs*, pages 101–115. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013.

[Bö68] Corrado Böhm. Alcune proprietà delle forme β-η-normali nel λ-K-calcolo. *INAC*, 696:1–19, 1968.

[CG14] Alberto Carraro and Giulio Guerrieri. A Semantical and Operational Account of Call-by-Value Solvability. In Anca Muscholl, editor, *Foundations of Software Science and Computation Structures*, volume 8412 of *Lecture Notes in Computer Science*, pages 103–118. Springer-Verlag, 2014.

[Ehr12] Thomas Ehrhard. Collapsing non-idempotent intersection types. In P. Cégielski and A. Durand, editors, *Computer Science Logic (CSL'12), 21st Annual Conference of the EACSL, CSL 2012*, volume 16 of *LIPIcs*, pages 259–273. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012.

[ER03] Thomas Ehrhard and Laurent Regnier. The differential lambda-calculus. *Theor. Comput. Sci.*, 309(1-3):1–41, 2003.

[ER06] Thomas Ehrhard and Laurent Regnier. Böhm trees, Krivine's machine and the Taylor expansion of lambda-terms. In Arnold Beckmann, Ulrich Berger, Benedikt Löwe, and John V. Tucker, editors, *Logical Approaches to Computational Barriers, Second Conference on Computability in Europe, CiE 2006, Swansea, UK, June 30-July 5, 2006, Proceedings*, volume 3988 of *Lecture Notes in Computer Science*, pages 186–197. Springer, 2006.

[ER08] Thomas Ehrhard and Laurent Regnier. Uniformity and the Taylor expansion of ordinary lambda-terms. *Theor. Comput. Sci.*, 403(2-3):347–372, 2008.

[GPR17] Giulio Guerrieri, Luca Paolini, and Simona Ronchi Della Rocca. Standardization and conservativity of a refined call-by-value lambda-calculus. *Logical Methods in Computer Science*, 13(4), 2017.

[Las99] Søren B. Lassen. Bisimulation in untyped lambda calculus: Böhm trees and bisimulation up to context. *Electr. Notes Theor. Comput. Sci.*, 20:346–374, 1999.

[Las05] Søren B. Lassen. Eager normal form bisimulation. In *20th IEEE Symposium on Logic in Computer Science (LICS 2005), 26-29 June 2005, Chicago, IL, USA, Proceedings*, pages 345–354. IEEE Computer Society, 2005.

[Mog92] Torben Æ. Mogensen. Efficient self-interpretations in lambda calculus. *J. Funct. Program.*, 2(3):345–363, 1992.

[MP11] Giulio Manzonetto and Michele Pagani. Böhm's theorem for resource lambda calculus through Taylor expansion. In C.-H. Luke Ong, editor, *Typed Lambda Calculi and Applications - 10th International Conference, TLCA 2011, Novi Sad, Serbia, June 1-3, 2011. Proceedings*, volume 6690 of *Lecture Notes in Computer Science*, pages 153–168. Springer, 2011.

[MRP19] Giulio Manzonetto, Simona Ronchi Della Rocca, and Michele Pagani. New semantical insights into call-by-value λ-calculus. *Fundam. Inform.*, 170(1-3):241–265, 2019.

[Ong97] Luke Ong. Lambda calculus, 1997. Lecture Notes.

[Pao01] Luca Paolini. Call-by-value separability and computability. In Antonio Restivo, Simona Ronchi Della Rocca, and Luca Roversi, editors, *Theoretical Computer Science, 7th Italian Conference, ICTCS 2001*, volume 2202 of *Lecture Notes in Computer Science*, pages 74–89. Springer, 2001.

[Pao08] Luca Paolini. Parametric λ-theories. *Theoretical Computer Science*, 398(1):51 – 62, 2008.

[Plo75] Gordon D. Plotkin. Call-by-name, call-by-value and the lambda-calculus. *Theor. Comput. Sci.*, 1(2):125–159, 1975.

[PRDR99] Luca Paolini and Simona Ronchi Della Rocca. Call-by-value solvability. *ITA*, 33(6):507–534, 1999.

[RDRP04] Simona Ronchi Della Rocca and Luca Paolini. *The Parametric λ-Calculus: a Metamodel for Computation.* EATCS Series. Springer, Berlin, 2004.

[Reg94] Laurent Regnier. Une équivalence sur les lambda-termes. *Theoretical Computer Science*, 126:281–292, 1994.

[Roc18] Simona Ronchi Della Rocca. Personal communication, 2018.

[Tra09] P. Tranquilli. *Nets Between Determinism and Nondeterminism.* PhD thesis, Univ. Paris 7 and Univ. Roma 3, 2009.