

Approximation theory for distant Bang calculus

Kostia Chardonnet 

Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

<https://kostiachardonnet.github.io/>

Jules Chouquet 

Université d'Orléans, INSA CVL, LIFO, UR 4022, Orléans, France

<https://www.univ-orleans.fr/lifo/membres/chouquet/>

Axel Kerinec 

Université Paris Est Creteil, LACL, F-94010 Créteil, France <https://axelkrnc.github.io/>

Abstract

Approximation semantics capture the observable behaviour of λ -terms, with Böhm Trees and Taylor Expansion standing as two central paradigms. Although conceptually different, these notions are related via the Commutation Theorem, which links the Taylor expansion of a term to that of its Böhm tree. These notions are well understood in Call-by-Name λ -calculus and have been more recently introduced in Call-by-Value settings. Since these two evaluation strategies traditionally require separate theories, a natural next step is to seek a unified setting for approximation semantics. The Bang-calculus offers exactly such a framework, subsuming both CbN and CbV through linear-logic translations while providing robust rewriting properties. However, its approximation semantics is yet to be fully developed.

In this work, we develop the approximation semantics for dBang, the Bang-calculus with explicit substitutions and distant reductions. We define Böhm trees and Taylor expansion within dBang and establish their fundamental properties. Our results subsume and generalize Call-By-Name and Call-By-Value through their translations into Bang, offering a single framework that uniformly captures infinitary and resource-sensitive semantics across evaluation strategies.

2012 ACM Subject Classification Theory of computation → Linear logic; Theory of computation → Lambda calculus; Theory of computation → Operational semantics

Keywords and phrases Lambda-calculus, Böhm Trees, Taylor expansion of lambda-terms

Funding Kostia Chardonnet: This work is supported by the Plan France 2030 through the PEPR integrated project EPiQ ANR-22-PETQ-0007 and the HQI platform ANR-22-PNCQ-0002; and by the European project MSCA Staff Exchanges Qcomical HORIZON-MSCA-2023-SE-01. The project is also supported by the Maison du Quantique MaQuEst.

1 Introduction

One of the central approaches to studying the semantics of the λ -calculus is the theory of program approximations. The goal is to capture, in a finitary or infinitary manner, the computational behaviour of a program, offering a characterization of “meaningful” terms. In *Call-by-Name* (CbN), meaningful terms are the *solvable* ones: a term is solvable if, under some “testing context”, it reduces to a fully defined result, the identity. In *Call-by-Value* (CbV), we consider the *scrutable*¹ ones: we only require the reduction to reach a value. This second notion is strictly finer: every solvable term is scrutable, but not conversely. Among the various approximation techniques that have been proposed over the years, two of the most influential are *Böhm Trees* and *Taylor Expansions*.

Böhm trees were first introduced by Barendregt [14]. They assign to each λ -term a (possibly infinite) tree whose nodes describe successive approximations of the term’s head-normal form, or \perp if the term does not reduce to a *head normal form*. Böhm Trees thus make explicit the asymptotic

¹ Also called potentially valuable



© Kostia Chardonnet and Jules Chouquet and Axel Kerinec;
licensed under Creative Commons License CC-BY 4.0



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

behaviour of a program under CbN evaluation: they capture the stable information that survives arbitrary program contexts. Böhm trees were related to the notion of *solvability* of CbN terms by the fact that a term is solvable if and only if its Böhm tree is not \perp .

Much more recently, Ehrhard and Regnier developed the Taylor expansion of λ -terms [25]. Inspired by the differential λ -calculus [24] and relational semantics [31], Taylor expansions unfold a λ -term into an (infinite) formal sum of resource terms, thereby yielding a resource-sensitive notion of approximation. Whereas Böhm Trees approximate terms by progressively revealing their shape, Taylor expansions instead decompose terms into linearly-used resources.

Although conceptually very different, these two notions of approximation are intimately related. The Commutation Theorem [25, Corollary 35] states that normalizing the Taylor expansion of a term yields exactly the Taylor expansion of its Böhm tree. Taylor expansions can then be understood as a *resource-sensitive* version of Böhm trees. Originally proved for the CbN λ -calculus, this result provides a deep bridge between infinitary semantics and differential/resource semantics.

While Taylor expansions for the CbV calculus were already studied in [18, 21], the development of CbV Böhm Trees remained open until Kerinec’s PhD work [29, 28]. Those Böhm trees have the same commutation theorem with Taylor expansion than in the CbN case. They also respect the same relation with scrutability than as CbN Böhm trees do to solvability. This emphasizes that scrutability is the appropriate notion of meaningfulness in CbV. However, this result is obtained in an alternative version of the original Plotkin CbV. Indeed, the original calculus is known to have issues due to the β_v -reduction being “too weak”. Concretely, unlike in CbN, CbV reduction may get stuck: redexes can be blocked since their argument is in normal form but not a value. This phenomenon prevents a straightforward infinitary unfolding analogous to the CbN case.

The aforementioned CbV Böhm trees are defined in the λ_v^σ -calculus from [17], where the β_v -reduction is extended with permutation rules, the so-called σ -rules, originating from the translation of λ -terms to Proof-Nets [1].

Another way to solve the CbV issue, also coming from Proof nets, is using a distance-based CbV calculus *distant CbV* (*dCbV*)² [3, 2, 4]. In this system, substitutions may be frozen thanks to a new term of the language called *explicit substitutions*, written $M[N/x]$, which does *not* correspond to an (effective) substitution, but instead represents a substitution that is yet to be evaluated. The rewriting rules then act *at a distance* with regard to the explicit substitutions.

The divergence between CbN and CbV means that most semantic notions (solvability, Böhm Trees, Taylor expansions, denotational models, ...) have historically required two separate developments, one for each evaluation strategy. This duplication has motivated a substantial research effort toward a unified framework capable of expressing both paradigms at once.

A major step in this direction is Call-by-Push-Value (CbPV), introduced by Levy [32], which unifies *typed* CbN and CbV into a single calculus with a clean separation between values and computations. CbPV was later connected to Linear Logic [22], giving rise to the Bang-calculus [23], an untyped analogue of CbPV. Both CbN and CbV arise within the Bang-calculus via Girard’s translations of the intuitionistic arrow into linear logic [27], making it a natural setting in which to seek a uniform approximation theory. The Bang-calculus is then an extension of the λ -calculus with two new constructs: $!M$ (pronounced “bang M ”) which *freezes* the computation of M , and $\text{der}(M)$ which unfreezes it. Both CbN and CbV can then be translated into the Bang-calculus, which simulates their rewriting strategies within a single rewriting system.

However, the Bang-calculus exhibits the same issue as CbV: ill-formed redexes may block evaluation. One might expect that adapting the σ -rules to the Bang-calculus would solve this

² Also called Value Substitution Calculus

issue; however, while the CbV calculus with σ -rules is confluent, this is not the case for the Bang calculus [23, Sec.2.3]. Confluence can be recovered modulo an equivalence relation contained in the σ -equivalence generated by the σ -rules, but this forces us to work modulo said equivalence. Another solution to this problem is to consider the distance variant (dBang) [16], similar to the distant variant of CbV. This allows us to work without the σ -rules and the aforementioned equivalences. This new system has been shown to be confluent [16]. It has also been used in unifying multiple results for dCbN and dCbV. For instance, in [9] the authors show that the rewriting results (confluence, factorization) of dBang could carry over to the dCbN and dCbV setting using the translations into dBang³. From an approximation-theory point of view, the notion of solvability in both dCbN and dCbV has been captured through the notion of *meaningfulness* in dBang [30, 11], which they also characterize using an intersection type system. However, despite significant progress, the approximation theory of the dBang-calculus, its Böhm trees, its Taylor expansions, and their relationship, remains largely underdeveloped.

A recent work by Mazza & Dufour tried to close that gap [20]: they developed a generic notion of Böhm tree and Taylor expansion for a language called Proc, representing untyped proof structures. They showed how any language that can be embedded into Proc in a “nice way” inherits the notions of Böhm Trees and Taylor expansion from Proc, and their commutation Theorem. In particular, dBang admit such an embedding. However, no notion of meaningfulness exists in Proc and it is not clear how to relate the results from CbN and CbV with the notion developed in [20].

1.1 Contributions

We develop a theory of approximation of the distance Bang-calculus (dBang). To this end we recall the definitions and main results of dBang from [30, 10, 16] in Section 2. We start by developing the Taylor expansion of dBang (Section 2.3) where we introduce the resource calculus (Section 2.3.1) and define the approximation relation (Section 2.3.2) and establish a simulation result between dBang and its approximants in the Taylor expansion (Theorem 26). We next develop the Böhm approximants of dBang (Section 2.4) and prove a commutation theorem between the Taylor Expansion of Böhm Trees and the Taylor normal form (Theorem 51).

Finally, we establish the soundness of our definition with regard to the standard notion of Böhm Trees and Taylor Expansions in the CbN and CbV λ -calculus by translating these systems into dBang (Section 3), in particular, we show that the Böhm Trees of a term M in CbN (respectively CbV) are the same as its translation into dBang (Theorem 66). We show a similar result for Taylor expansion (Lemmas 57 and 59, and Corollaries 58 and 60).

1.2 Notations

For any reduction relation \rightarrow we define, we use the standard notations: \rightarrow^* , \rightarrow^k for, respectively, its reflexive transitive closure and its k -step iteration. We write $d_x(\tau)$ for the number of free occurrences of the variable x in the term τ (in any of the languages considered). We write $[m_1, \dots, m_k]$ for a finite multiset containing k occurrences of terms. When necessary, we use a subscript as $[m]_k$ or $[m, \dots, m]_k$ in order to make explicit the number of elements.

³ Note that while the translation of dCbN into dBang is the usual one mentioned before, the authors use a new translations for CbV which will be discussed in Section 3

2 The (distance) Bang-calculus

We begin by recalling the theory of dBang, first with some results from previous studies [30, 10, 16], and then develop the approximation theory via Taylor expansion and Böhm trees.

2.1 Definition of the calculus dBang

► **Definition 1.** (dBang: terms and contexts)

(Terms)	$M, N := x \mid MN \mid \lambda x M \mid !M \mid \mathbf{der}(M) \mid M[N/x]$
(List contexts)	$L := \square \mid L[M/x]$
(Surface contexts)	$S := \square \mid SM \mid MS \mid \lambda x S \mid \mathbf{der}(S) \mid S[M/x] \mid M[S/x]$
(Full contexts)	$F := \square \mid FM \mid MF \mid \lambda x F \mid \mathbf{der}(F) \mid F[M/x] \mid M[F/x] \mid !F$

The set of terms includes the standard constructs of the λ -calculus: variables x, y, z, \dots , ranging over a countably infinite set; application MN and lambda-abstraction $\lambda x M$. Furthermore, there are two new constructions: the *bang* (or *exponential*) $!M$, representing delayed evaluation of the subterm M , and *dereliction* $\mathbf{der}(M)$, which reactivates the evaluation of M if it has been delayed.

Finally, *explicit substitutions* $M[N/x]$ represent pending substitutions. Note that the lambda-abstraction and explicit substitution bind the variable x in M . We use *contexts*, i.e. terms with a subterm hole (\square) that can be filled by a given λ -term; we denote by $C\langle M \rangle$ the term obtained by replacing the hole in C by M . We have three types of contexts: *list contexts*, *surface contexts* and *full contexts*. List contexts are sequences of explicit substitutions and will be used for the reduction at a distance. Surface and full contexts determine whether reduction under a $!$ is allowed (in particular it is forbidden in *weak* calculi (CbN or CbV).

The dBang calculus has three reduction rules:

$$L\langle \lambda x M \rangle N \mapsto_! L\langle M[N/x] \rangle \quad M[L\langle !N \rangle /x] \mapsto_! L\langle M\{N/x\} \rangle \quad \mathbf{der}(L\langle !M \rangle) \mapsto_! L\langle M \rangle$$

Notice that requiring certain subterms to be of the form $!M$ grants them the status of *value* in the CbV sense.

We then give the following contextual extensions:

► **Definition 2.**

- $\rightarrow_{!s}$ is the closure under surface contexts S .
- $\rightarrow_!$ is the closure under full contexts F .

We also denote by \rightarrow_i the internal reductions (i.e. $\rightarrow_i = \rightarrow_! \setminus \rightarrow_{!s}$, these are the reductions occurring under a $!$ -construct).

► **Example 3.** $(\lambda xxy)[!z/y]!!((\lambda ww)!N) \rightarrow_{!s} (xy)[!!((\lambda ww)!N)/x][!z/y] \rightarrow_{!s} !((\lambda ww)!N)y[!z/y] \rightarrow_{!s} !((\lambda ww)!N)z \rightarrow_! !!Nz$. The last step could not be performed by a surface reduction.

► **Example 4.**

- $\Delta = \lambda x(x!x), \Omega = \Delta! \Delta$. We have $\Omega \rightarrow_{!s}^2 \Omega$.
- $Y_x^n = (\lambda yx!(y!y))!(\lambda yx!(y!y))$. We have $Y_x^n \rightarrow_{!s}^+ x!Y_x^n$
- $Y_x^v = (\lambda yx(y!y))!(\lambda yx(y!y))$. We have $Y_x^v \rightarrow_{!s}^+ xY_x^v$

The upper scripts in the two last items, as we shall see further, represent the fact that Y_x^n and Y_x^v correspond respectively to the CbN and CbV versions of the fixpoint of x .

► **Theorem 5** (Confluence [30, Theorem 1]).

- $\rightarrow_{!s}$ is confluent.
- $\rightarrow_!$ is confluent.

2.2 Meaningfulness in dBang

We then look at the notion of *meaningfulness* which will be related to approximation theory in Section 4. Intuitively, a meaningful term is a term which, under some *testing context*, reduces to a specific desired result. It generalizes the notions of *solvability* that exist in CbV and CbN, and has been studied in detail for dBang in [30].

► **Definition 6** (Testing Contexts). $(Tests) \quad T := \square \mid TM \mid (\lambda xT)M$

► **Definition 7.** A term M of dBang is said meaningful if there exists a testing context T and a term P such that $T\langle M \rangle \rightarrow_!^* !P$.

Notice that the surface reduction involved in meaningfulness is not restrictive: for any N , $N \rightarrow_!^* !P$, then there is some P' such that $N \rightarrow_{!s}^* !P'$ (this follows from a standardisation property, see Corollary 10 below). The notion of meaningfulness for dBang has been explored in [30], particularly with respect to the following results:

► **Theorem 8.** The following hold:

Meaningfulness meaningful terms and surface-normalizing terms can be characterized by an intersection type systems [30, Theorem 24].

Consistency the smallest theory that identifies all meaningless terms is consistent [30, Proposition 8].

Genericity if M is meaningless and $S\langle M \rangle$ is meaningful, then $S\langle N \rangle$ is meaningful for any N (i.e., meaningless subterms do not affect the operational meaning of a given term)[30, Corollary 11].

A natural property of surface reduction is that it determines the external shape of a term. In other words, allowing full reduction does not unlock external redexes. This is expressed as the following factorization proposition:

► **Proposition 9.** ([10], Corollary 21) Let $M \rightarrow_!^* N$. There is some P such that $M \rightarrow_{!s}^* P \rightarrow_i^* N$.

Analogously, a notable feature of internal reductions is that they do not modify the *external shape* of the term. We express this notion with multi-holes surface contexts, that let us reformulate the factorization property as follows:

► **Corollary 10** (Standardization). Let S^+ denote multi-holes surface contexts:

$$S^+ := \square \mid M \mid S^+S^+ \mid S^+[S^+/x] \mid \mathbf{der}(S^+) \mid \lambda xS^+$$

For any reduction $M \rightarrow_!^* P$, there are some terms N_i and a multi-hole surface context S such that $P = S\langle !N_1, \dots, !N_k \rangle$ and:

$$M \rightarrow_{!s}^* S\langle !N'_1, \dots, !N'_k \rangle \rightarrow_i^* S\langle !N_1, \dots, !N_k \rangle$$

Notice that k might be equal to 0, if the context has no hole, in that case the reduction occurs only at surface level.

2.3 Taylor expansion

Here we define the Taylor expansion of dBang. For that purpose, we first define a resource calculus for dBang: the language δ Bang. While we define this language to develop the approximants in dBang, we will also use it for both CbN and CbV resource approximants in Section 3⁴.

⁴ There is no necessity to define specific resource calculi, as δ Bang fits well as a target of usual Taylor expansion (Call-By-Name [26] and Call-By-Value [21]), with a straightforward adaptation to distant setting.

$$\begin{array}{c}
\frac{}{x \triangleleft! x} \quad \frac{m \triangleleft! M}{\lambda xm \triangleleft! \lambda xM} \quad \frac{m \triangleleft! M}{\mathbf{der}(m) \triangleleft! \mathbf{der}(M)} \quad \frac{m \triangleleft! M \quad n \triangleleft! N}{mn \triangleleft! MN} \\
\\
\frac{m \triangleleft! M \quad n \triangleleft! N}{m[n/x] \triangleleft! M[N/x]} \quad \frac{m_1 \triangleleft! M \quad \dots \quad m_k \triangleleft! M}{[m_1, \dots, m_k] \triangleleft! !M} \quad k \in \mathbb{N}
\end{array}$$

Figure 1 Resource approximation for dBang

2.3.1 dBang: resources

► **Definition 11** (Resource calculus δ Bang).

$$\begin{array}{ll}
(\text{terms}) & m, n := x \mid mn \mid \lambda xm \mid \mathbf{der}(m) \mid m[n/x] \mid [m_1, \dots, m_k] \\
(\text{lists}) & l := \square \mid l[m/x] \\
(\text{surface}) & s := \square \mid sm \mid ms \mid \lambda xs \mid \mathbf{der}(s) \mid s[m/x] \mid m[s/x] \\
(\text{full}) & f := \square \mid sm \mid mf \mid \lambda xf \mid \mathbf{der}(f) \mid f[m/x] \mid m[f/x] \mid [f, m_1, \dots, m_k] \\
(\text{tests}) & t := \square \mid tm \mid (\lambda xt)m
\end{array}$$

Terms $[m_1, \dots, m_k]$ for $k \in \mathbb{N}$, often called *bags* denote finite multisets of resource terms where $[]$ is the empty bag. Contexts are exactly as in dBang (Definition 1), plus the bag contexts instead of the exponential contexts. We write P_k for the sets of permutations of the set $\{1, \dots, k\}$, and we denote as $d_x(m)$ the number of free occurrences of the variable x in m . Resource substitution is (multi-)linear: when we write $m\{n_1/x_1, \dots, n_k/x_k\}$, it is always intended (in the resource setting) that x_i represents the i -th free occurrence of x in m . In that way, each term n_i is substituted exactly once in m .

We are now ready to define the reduction relation:

► **Definition 12.**

The reduction relation $\Rightarrow_\delta \subseteq \delta\text{Bang} \times \wp(\delta\text{Bang})$ is then defined as follows:

- $l\langle \lambda xm \rangle n \Rightarrow_\delta \{l\langle m[n/x] \rangle\}$
- $m[l\langle [n_1, \dots, n_k] \rangle / x] \Rightarrow_\delta \begin{cases} \bigcup_{\sigma \in P_k} l\langle m\{n_{\sigma(1)}/x_1, \dots, n_{\sigma(k)}/x_k\} \rangle & \text{if } k = d_x(m) \\ \emptyset & \text{otherwise} \end{cases}$
- $\mathbf{der}(l\langle [m_1, \dots, m_k] \rangle) \Rightarrow_\delta \{l\langle m_1 \rangle\}$ if $k = 1$ and \emptyset otherwise

We write $m \rightarrow_\delta n$ as soon as $m \Rightarrow_\delta X$ and $n \in X$ for some n (if $m \Rightarrow_\delta \emptyset$, we also abusively write $m \rightarrow_\delta \emptyset$, and we add the following equation: if $f\langle \emptyset \rangle = \emptyset$ for any full context f). We also define $\rightarrow_{\delta s}$ and \rightarrow_δ the contextual closures of \rightarrow_δ under surface and full contexts, respectively. Notice that none of these reductions is deterministic. Both $\rightarrow_{\delta s}$ and \rightarrow_δ are strongly normalizing, which is an immediate consequence of linearity: the size of bags of resource terms are decreasing following the reductions. Confluence of $\rightarrow_{\delta s}$ and \rightarrow_δ can be easily derived from standard results in resource calculus and from the proofs for $\rightarrow_{!s}$ and $\rightarrow_!$.

2.3.2 Approximation

In Figure 1, we define a relation $\triangleleft! \subseteq \delta\text{Bang} \times \text{dBang}$, where $m \triangleleft! M$ means that m is a multilinear resource approximation on M .

We extend this definition to list contexts as follows: $\square \triangleleft! \square, l[m/x] \triangleleft! L[M/x]$ if $l \triangleleft! L$ and $m \triangleleft! M$. The extension to surface and full contexts follows analogously.

The intended behaviour of context approximation is contained in the following result, which is shown by a standard induction on contexts:

► **Lemma 13.**

- If $m \triangleleft! L\langle N \rangle$, then there exist $l \triangleleft! L$ and $n \triangleleft! N$ such that $m = l\langle n \rangle$.
- If $m \triangleleft! S\langle N \rangle$, then there exist $s \triangleleft! S$ and $n \triangleleft! N$ such that $m = s\langle n \rangle$.

Notice that this property does not hold for full contexts; consequently, our definitions do not provide a convenient notion of approximation between full contexts. This is because full contexts require parallel treatment of terms within a bag, which - as we shall see later - is not possible with single-hole contexts.

► **Definition 14.** (*Taylor expansion*) For any $M \in \delta\text{Bang}$, we define its *Taylor expansion* as the set of its resource approximants:

$$\mathcal{T}(M) = \{m \in \delta\text{Bang} \mid m \triangleleft! M\}$$

By strong normalization of δBang ⁵, we can define the normal form $\mathbf{nf}(m)$ of a resource term m as the finite set made of its full reducts. We then define the *Taylor normal form* of δBang terms as $\text{TNF}(M) = \bigcup_{m \triangleleft! M} \mathbf{nf}(m)$. Notice that Taylor normal form is made of full normal terms, not only surface normal forms.

► **Remark 15.** [Clashes and normal forms] Note that we have not excluded so-called *clashes* from our calculus, as they do not pose a problem in our specific context. Thus, terms such $\mathbf{der}(\lambda xx)$ are considered as regular normal forms. This approach is similar to the one of Dufour and Mazza [20]. However, defining a clash-free fragment of δBang would be straightforward, as it would mainly not interfere with the technical developments in this paper. For example, adding reductions such as $\mathbf{der}(\lambda xm) \rightarrow \emptyset$ to resource calculus to prevent the appearance of clashes in Taylor expansion would be unproblematic).

► **Example 16.** Consider the terms given in Example 4.

- An approximant of $m \triangleleft! \Omega$ must be of shape $(\lambda xx[x]_k)[\lambda xx[x]_{k_1}, \dots, \lambda xx[x]_{k_l}]$. Now, if $k = l - 1$ (otherwise $m \mapsto! \emptyset$) then $m \rightarrow_{\delta_s} (\lambda xx[x]_{k_1})[\lambda xx[x]_{k_2}, \dots, \lambda xx[x]_{k_l}]$ ⁶, which is again an approximant of Ω . But we can observe that the cardinality of the bag reduces during this reduction; hence if we iterate this reduction, we eventually reach a term like $(\lambda x[x]_k)[] \rightarrow_{\delta_s} \emptyset$ (if an empty reduction has not occurred before). So, $\text{TNF}(\Omega) = \emptyset$.
- Similarly, if $m \triangleleft! Y_x^n$, we verify easily that $m \rightarrow_{\delta_s} x[n_1, \dots, n_k]$, with $n_i \triangleleft! Y_x^n$. In particular, $x[] \triangleleft! Y_x^n$ and is in normal form. Actually, $\text{TNF}(Y_x^n)$ can be characterized inductively : $x[] \in \text{TNF}(Y_x^n)$, and if $n_1, \dots, n_k \in \text{TNF}(Y_x^n)$, then $x[n_1, \dots, n_k] \in \text{TNF}(Y_x^n)$, for any k .
- The other fixpoint term, Y_x^v , behaves slightly differently: if $m \triangleleft! Y_x^v$, then we verify $m \rightarrow_{\delta_s}^* xn$, for some $n \triangleleft! Y_x^v$. But here, because of the argument not being in a bag, all approximants reduce (if not \emptyset) to some term $xxx\dots xn$, but are not in normal form; since such a reduction terminates, we observe that $\text{TNF}(Y_x^v) = \emptyset$.

2.3.3 Simulation

We systematize here what has been sketched in the Example 16, that reduction in δBang can be simulated in the approximants of δBang .

⁵ Recall that the resource reduction is size-decreasing.

⁶ We consider here one possible reduction, any element of the bag could be substituted to the inner head variable x , not necessarily $\lambda x[x]_{k_1}$, the argument is valid for all the reduction paths.

► **Lemma 17** (Substitution Lemma for Taylor expansion). *For any M, N of dBang, for any $m, n_1, \dots, n_{d_x(m)}$ of δBang , we have $m\{n_1/x_1 \dots n_{d_x(m)}/x_{d_x(m)}\} \triangleleft! M\{N/x\}$ if and only if $m \triangleleft! M$, and $n_i \triangleleft! N_i$ for all $i \leq d_x(m)$.*

Proof. By a routine induction on M . ◀

We can show, by a straightforward induction, with the help of the Substitution Lemma stated above, that the surface reduction acts exactly the same way in a dBang term and in its approximants:

► **Lemma 18.** *Let $m \triangleleft! M$ and $m \rightarrow_{\delta_s} n$ for some n . Then there is some $N \in \text{dBang}$ such that $M \rightarrow_{!s} N$ and $n \triangleleft! N$.*

However, this result is false when considering reduction in full contexts: let $m = [(\lambda xx)y, (\lambda xx)y]$. We have $m \triangleleft! ((\lambda xx)y)$ and $m \rightarrow_{\delta} n = [x[y/x], (\lambda xx)y]$, but n is not an approximant of any dBang term. We will need parallel reduction to achieve full simulation, this is done in Subsection 2.3.4.

► **Lemma 19.** *If $M \mapsto_! N$, then for any $m \triangleleft! M$, either $m \rightarrow_{\delta} \emptyset$, or there is some $n \triangleleft! N$ such that $m \rightarrow_{\delta} n$.*

Proof. By induction on the definition $\mapsto_!$:

- If $m \triangleleft! \text{der}(L\langle !N \rangle)$, then there exist $k \in \mathbb{N}, n_1, \dots, n_k \triangleleft! N$, and $l \triangleleft! L$ such that $m = \text{der}(l\langle [n_1, \dots, n_k] \rangle)$, and then $m \mapsto_! \emptyset$ if $k \neq 1$, otherwise $m \mapsto_! l\langle n_1 \rangle \triangleleft! L\langle N \rangle$.
- If $m \triangleleft! (L\langle \lambda xN \rangle P)$, then $m = l\langle \lambda xn \rangle p$ for some $l \triangleleft! L, n \triangleleft! N$ and $p \triangleleft! P$. Then, $m \rightarrow_{\delta} l\langle n[p/x] \rangle \triangleleft! L\langle N[P/x] \rangle$.
- If $m \triangleleft! N[L\langle !P \rangle / x]$ then there are $k \in \mathbb{N}, p_1, \dots, p_k \triangleleft! P, n \triangleleft! N, l \triangleleft! N$ such that $m = n[l\langle [p_1, \dots, p_k] / x \rangle]$. Then $m \rightarrow_{\delta} \emptyset$ if $d_x(m) \neq k$, and otherwise for any $\sigma \in P_k$, $m \rightarrow_{\delta} l\langle n\{p_{\sigma(1)}/x_1, \dots, p_{\sigma(k)}/x_k\} \rangle$. ◀

This simulation property can be extended to surface contexts:

► **Lemma 20.** *If $M \rightarrow_{!s} N$, then for any $m \triangleleft! M$, either $m \rightarrow_{\delta_s} \emptyset$ or there is some $n \triangleleft! N$ such that $m \rightarrow_{\delta_s} n$.*

Proof. Let $M = S\langle M' \rangle$ and $N = S\langle N' \rangle$ with $M \mapsto_! N$; and then $m = s\langle m' \rangle$ for $s \triangleleft! S$. By induction on surface contexts:

- $S = \square$. Then $M \mapsto_! N$, we apply Lemma 19.
- $S = \lambda xS'$. Then $M = \lambda xS'\langle M' \rangle$ and $N = \lambda xS'\langle N' \rangle$. By induction hypothesis, either $m' \rightarrow_{\delta_s} \emptyset$, and then $\lambda xs'\langle m' \rangle \rightarrow_{\delta_s} \emptyset$, either we have some $n' \triangleleft! N'$ such that $m' \rightarrow_{\delta_s} n'$, and then $\lambda xs\langle m' \rangle \rightarrow_{\delta} \lambda xs\langle n' \rangle$ by definition of resource surface reduction.

All remainder cases are similar, since for any resource surface context s and term $n \triangleleft! N$, there exist S such that $s\langle n \rangle \triangleleft! S\langle N \rangle$ which enables the induction hypothesis. Again, this fails for full contexts. ◀

We can obtain a symmetric property with the same arguments. We could qualify, following Dufour and Mazza [20], the previous lemma as *push forward*, and the next lemmas as *pull back*.

► **Lemma 21.** *If $M \rightarrow_{!s} N$, then for any $n \triangleleft! N$, there is some $m \triangleleft! M$ such that $m \rightarrow_{\delta_s} n$.*

Proof. Let $M = S\langle M' \rangle, N = S\langle N' \rangle$ with $M' \mapsto_! N'$ then we can reason by induction on S .

- $S = \square$. Then $M \mapsto_! N$ and by a straightforward case analysis on $\mapsto_!$.

$$\begin{array}{c}
\frac{}{x \rightrightarrows_{\delta} x} \quad \frac{m_1 \rightrightarrows_{\delta} m'_1 \quad \dots \quad m_k \rightrightarrows_{\delta} m'_k}{[m_1, \dots, m_k] \rightrightarrows_{\delta} [m'_1, \dots, m'_k]} \quad k \in \mathbb{N} \\
\frac{m \rightrightarrows_{\delta} m'}{s(m) \rightrightarrows_{\delta} s(m')} * \quad \frac{[m_1, \dots, m_k] \rightrightarrows_{\delta} [m'_1, \dots, m'_k] \quad n \rightrightarrows_{\delta} n' \quad l \rightrightarrows_{\delta} l'}{n[l([m_1, \dots, m_k])] \rightrightarrows_{\delta} l'\langle n'\{m'_{\sigma(1)}/x_1, \dots, m'_{\sigma(k)}/x_k\} \rangle} ** \\
\frac{m \rightrightarrows_{\delta} m' \quad n \rightrightarrows_{\delta} n' \quad l \rightrightarrows_{\delta} l'}{l(\lambda xm)n \rightrightarrows_{\delta} l'\langle m'[n'/x] \rangle} \quad \frac{m_1 \rightrightarrows_{\delta} m'_1}{\text{der}([m_1, \dots, m_k]) \rightrightarrows_{\delta} m'_1} ***
\end{array}$$

* s is any surface resource context

** $\sigma \in P_k$, and if $k = d_x(n')$ (otherwise the reduction gives \emptyset).

*** $k = 1$. Otherwise, the reduction gives \emptyset .

■ **Figure 2** Parallel reduction for δ Bang

- $S = \lambda x S'$. Then $M = \lambda x S' \langle M' \rangle$, $N = \lambda x S' \langle N' \rangle$ and $m = \lambda x s' \langle m' \rangle$ with $s' \triangleleft! S'$. By induction hypothesis there exists $m' \triangleleft! M'$ such that $m' \rightarrow_{\delta} n'$ then let $m = \lambda x s' \langle m' \rangle$ we have that $m \rightarrow_{\delta_s} n$ by closure.

All the remainder cases are similar. ◀

More generally considering the following scheme:

$$\begin{array}{ccc}
M & \rightarrow_{!s} & N \\
\triangledown & & \triangledown \\
m & \rightarrow_{\delta_s} & n
\end{array}$$

We deduce from previous observations that, given three of four terms, we can always obtain a convenient fourth that completes the square.

► **Definition 22.** Let X and Y be sets, and $\mathcal{R} \subseteq X \times Y$ a relation. We write $X \overline{\mathcal{R}} Y$ whenever

- For any $x \in X$, there exists $y \in Y$ such that $(x, y) \in \mathcal{R}$.
- For any $y \in Y$, there exists $x \in X$ such that $(x, y) \in \mathcal{R}$.

Then, using Lemmas 20 and 21 we can obtain a simulation theorem for surface reduction:

► **Theorem 23** (Simulation).

Let M, N in dBang such that $M \rightarrow_{!s} N$. We have $\mathcal{T}(M) \Rightarrow_{\delta_s} \mathcal{T}(N)$.

2.3.4 Parallel reduction and full contexts

We define in Figure 2 a wider notion of resource reduction which allows us to consider full contexts. Intuitively, it needs to reduce at once every term occurring in a bag, in order to simulate internal reductions like $!M \rightarrow_{!} !M'$. In particular, it needs to be reflexive because e.g. $\square \triangleleft! !M$ and \square does not reduce to \emptyset , we have to consider that $\square \triangleleft! !M'$ is obtained from \square by reduction. This parallel notion of reduction, written $\rightrightarrows_{\delta} \subseteq \delta\text{Bang} \times \delta\text{Bang}$ in our setting, is a well-known non-deterministic extension of standard reduction which can be used to prove confluence property. In general, for a reduction \rightarrow , we have $\rightarrow \subseteq \rightrightarrows \subseteq \rightarrow^*$, with \rightrightarrows enjoying the diamond property. See e.g. Barendregt's proof of confluence for the λ -calculus [12]. We abusively write $l \rightrightarrows_{\delta} l'$ for contexts as soon as $l = \square[m_1/x_1] \dots [m_k/x_k]$, $l' = \square[m'_1/x_1] \dots [m'_k/x_k]$ and $m_i \rightrightarrows_{\delta} m'_i$ for any $i \leq k$.

XX:10 Approximation theory for distant Bang calculus

Since \Rightarrow_δ is size-decreasing, it enjoys weak normalization, but obviously not strong, as it is reflexive. It also enjoys the (one-step) diamond property, as it can be proved by standard techniques. We can now state our simulation results for full contexts:

► **Lemma 24.** *Let $M, N \in \text{dBang}$ with $M \rightarrow_! N$. For any $m \triangleleft_! M$, either $m \Rightarrow_\delta \emptyset$ or there is some $n \triangleleft_! N$ such that $m \Rightarrow_\delta n$*

Proof. We proceed by induction on the exponential depth where the reduction occurs. By lemma 20, and since $\rightarrow_{\delta s} \subseteq \Rightarrow_\delta$, the only case we need to show is the full context closure, where $M = F\langle M' \rangle = !F'\langle M' \rangle$ and $N = F\langle N' \rangle$.

Then, $m = [p_1, \dots, p_k]$ for some $k \in \mathbb{N}$, where $p_i \triangleleft_! F'\langle M' \rangle$.

By induction hypothesis, either some p_i reduces to \emptyset , and then also $m \Rightarrow_\delta \emptyset$, either for any $i \leq k$, there is some $p'_i \triangleleft_! F'\langle N' \rangle$ such that $p_i \Rightarrow_\delta p'_i$. Then, $m \Rightarrow_\delta [p'_1, \dots, p'_k] \triangleleft_! !F'\langle N' \rangle = N$. ◀

The symmetric counterpart of this result is obtained with a similar reasoning:

► **Lemma 25.** *Let $M, N \in \text{dBang}$ with $M \rightarrow_! N$. For any $n \triangleleft_! N$, there is some $m \triangleleft_! M$ such that $m \Rightarrow_\delta n$.*

The two previous lemmas give us the desired simulation result for full reduction:

► **Theorem 26.** *Let $M, N \in \text{dBang}$ such that $M \rightarrow_! N$. Then we have*

$$\mathcal{T}(M) \overline{\Xi}_\delta \mathcal{T}(M)$$

We saw that surface reduction acts similarly in dBang and in δ Bang, while parallel reduction is necessary to give a multilinear account to internal reductions (the definition of parallel reduction alone does not imply that they occur exclusively inside bags, but this is made mandatory through the use of invariant multi-hole surface contexts). The factorization properties established for dBang (Corollary 10) can easily be translated in the resource setting:

► **Proposition 27** (Factorization). *Let s^+ denote multi-holes surface contexts:*

$$s^+ := \square \mid m \mid s^+ s^+ \mid s^+ [s^+ / x] \mid \mathbf{der}(s^+) \mid \lambda x s^+$$

For any reduction $m \rightarrow_\delta^ p$, there are some bags \bar{n}_i and some multi-hole context s such that $p = s\langle \bar{n}_1, \dots, \bar{n}_k \rangle$ and:*

$$m \rightarrow_{\delta s}^* s\langle \bar{n}'_1, \dots, \bar{n}'_k \rangle \Rightarrow_\delta^* s\langle \bar{n}_1, \dots, \bar{n}_k \rangle$$

2.3.5 Taylor normal form

Following the reduction occurring on resource terms, we have defined previously Taylor normal form. We develop in this section some lemmas on those objects. They will be useful in order to prove the Commutation Theorem between Böhm trees and Taylor expansions.

► **Lemma 28.** *Given $M \rightarrow_!^* N$ then $\text{TNF}(M) = \text{TNF}(N)$.*

Proof. By Theorem 26, we immediately have $\text{TNF}(M) \subseteq \text{TNF}(N)$. Then, let $n \in \text{TNF}(N)$. By iteration of Lemma 25, there is some $m \in \mathcal{T}(M)$ such that $m \Rightarrow_\delta^* n$. Then n must be in $\text{TNF}(M)$. ◀

► **Lemma 29.** *Given $m \in \text{TNF}(M)$ then there exists M' such that $M \rightarrow_!^* M'$ and $m \triangleleft_! M'$.*

Proof. Consider $m_0 \triangleleft M$ such that $m_0 \rightarrow_{\delta}^* m$. We reason by induction over the exponential depth under which the reduction occurs. If this depth is 0, then $m_0 \rightarrow_{\delta_s}^* m$ and we apply (iteratively) Lemma 18 to conclude.

Otherwise, we use the factorization property : by Proposition 27, we have some n such that $m_0 \rightarrow_{\delta_s}^* n = s\langle \bar{n}_1, \dots, \bar{n}_k \rangle \Rightarrow_{\delta}^* s\langle \bar{n}'_1, \dots, \bar{n}'_k \rangle = m$. By Lemma 18, we have some $N = S(!N_1, \dots, !N_k)$ such that $M \rightarrow_{!s}^* N$ and $n \triangleleft N$ (thus $n_i \triangleleft N_i$). Then, for any $i \leq k$, $\bar{n}_i = [n_{i,1}, \dots, n_{i,l_i}]$ and $\bar{n}'_i = [n'_{i,1}, \dots, n'_{i,l_i}]$ (in normal form, since these are subterms of m) with $n_{i,j} \triangleleft N_i$ and $n_{i,j} \rightarrow_{\delta}^* n'_{i,j}$. This reduction occurs under an exponential (*i.e.* inside a multiset), we then can apply our induction hypothesis to assert that there is some N'_i such that $N_i \rightarrow_{!} N'_i$ and $n'_{i,j} \triangleleft N'_i$. We can conclude, by setting $M' = S(!N'_1, \dots, !N'_k)$. \blacktriangleleft

2.4 Böhm trees

In this section, we develop the Böhm approximation of dBang, before relating it to Taylor approximation. The result we have in sight here is the commutation theorem (Theorem 51), which states that Taylor expansion of the Böhm tree of a term is equal to its Taylor normal form (the first result of this kind is in Ehrhard and Regnier's seminal work [26] for CbN λ calculus; and the proof for CbV is more recent, see Kerinec, Manzonetto and Pagani [29]).

► **Definition 30** (dBang $_{\perp}$). Let dBang $_{\perp}$ be the set of dBang-terms extended with the symbol \perp . In the following we use subscripts as \perp_i in order to distinguish occurrences of \perp as a subterm.

Similarly, we extend the different types of contexts; and we extend reductions in the obvious way, \perp being considered as a regular term.

► **Definition 31.** The set of approximants is a strict subset of dBang $_{\perp}$ generated by the grammar:

$$\begin{aligned} A &:= \perp \mid B \mid \lambda x A \mid !A \mid A[A!/x] \\ B &:= x \mid A_{\lambda} A \mid \text{der}(A!) \\ A! &:= B \mid \lambda x A \mid A![A!/x] \\ A_{\lambda} &:= B \mid !A \mid A_{\lambda}[A!/x] \end{aligned}$$

► **Lemma 32.** Approximants are the normal forms of dBang $_{\perp}$.

The normal forms are preserved by substituting any term to a \perp , in the following sense:

► **Lemma 33.** Let A be an approximant, M any term of dBang $_{\perp}$, and \perp_i some occurrence of \perp in A . If $A[M/\perp_i] \rightarrow_{!} N$, then $N = A[M'/\perp_i]$ with $M \rightarrow_{!} M'$. i.e. we cannot create a redex when replacing a \perp by a term M in an approximant: the only redexes obtained this way are those already present in M .

Proof. This property is in fact a consequence of the syntactic structure of approximants. First, approximants contain no redexes. Secondly, there are no approximants containing subterms of shape $\text{der}(\perp), \perp A, A[\perp/x]$ that could hide a potential redex. \blacktriangleleft

► **Definition 34** (Order). Let $\sqsubseteq \subseteq \text{dBang}_{\perp} \times \text{dBang}_{\perp}$ be the least contextual closed preorder on dBang $_{\perp}$ generated by setting: $\forall M \in \text{dBang}_{\perp}$ and for all full context F , $F[\perp] \subseteq F[M]$.

In other words, $M \sqsubseteq N$ if and only if $M = N\overline{\{P/\perp\}}$. In the Böhm trees literature, we sometimes find approximation orders where only terms of shape $!P$ are replaced by a \perp (see e.g. Dufour and Mazza [20]). For technical reasons due to CbN and CbV embeddings into dBang, we need, for example, $\lambda x \perp \sqsubseteq \lambda x M$ to be a valid approximation.

Intuitively, the following lemma states that when a reduction occurs in a term, it cannot be seen by its approximations, that only represent a part of the skeleton of their normal form. In

XX:12 Approximation theory for distant Bang calculus

other words, when $A \sqsubseteq M$, then A represents a subtree of M which cannot be modified by some reduction.

► **Lemma 35.** $A \sqsubseteq M$ and $M \rightarrow_!^* N$ then $A \sqsubseteq N$.

Proof. We show the lemma holds for one-step reductions, assuming $M \rightarrow_! N$. The closure is easily obtained by induction on the number of steps. By definition, $M = A\{P_1/\perp_1, \dots, P_k/\perp_k\}$, and by Lemma 33, there must be some $j \in \{1, \dots, k\}$ such that $N = A(\{P_i/\perp_i\}_{i \neq j} \{P'_j/\perp_j\})$ with $P_j \rightarrow_! P'_j$. Consequently, we have $A \sqsubseteq N$. ◀

► **Definition 36** (Set of approximants). *Given $M \in \text{dBang}$, the set of approximants of M is defined as follows: $\mathcal{A}(M) = \{A \mid M \rightarrow_!^* N, A \sqsubseteq N\}$.*

Note that $\mathcal{A}(M)$ is never empty, since it contains at least \perp .

► **Example 37.** Consider the terms given in Example 4.

- The only reducts of Ω being Ω itself, and since the syntax A of approximants contains no redex, we easily conclude that $\mathcal{A}(\Omega) = \{\perp\}$.
- The reducts of Y_x^n are of shape $x!x!x\dots!Y_x^n$, we conclude that $\mathcal{A}(Y_x^n)$ contains precisely the terms $\perp, x\perp, x!\perp, x!x\perp, x!x!\perp, \dots$.
- The reducts of Y_x^v follow the same observation, minus the exponentials, and $\mathcal{A}(Y_x^v)$ contains precisely $\perp, x\perp, xx\perp, xxx\perp, \dots$

► **Lemma 38.** $M \rightarrow_!^* N$ then $\mathcal{A}(M) = \mathcal{A}(N)$.

Proof. From Definition 36 we deduce $\mathcal{A}(N) \subseteq \mathcal{A}(M)$, and from Lemma 35 the other inclusion. ◀

► **Remark 39.** Clearly, if $A = F[\vec{\perp}] \in \mathcal{A}(M)$ (where F is a multi-hole context), then $M \rightarrow_!^* F[\vec{N}]$ for some \vec{N} . In particular, if A contains no \perp as a subterm, then M has a normal form, which is exactly A .

► **Definition 40** (Ideal). *Let $X \subseteq \text{dBang}_\perp$. We set X is an ideal when X is downwards closed for \sqsubseteq , and directed: for all $M, N \in X$ there exists some upper bound (with respect to \sqsubseteq) to $\{M, N\}$.*

► **Lemma 41.** *Let $M \in \text{dBang}$. $\mathcal{A}(M)$ is an ideal.*

Proof. The downwards closure is by definition of $\mathcal{A}(M)$. For directedness, let us assume $A_1, A_2 \in \mathcal{A}(M)$, we show that there exists $A_3 \in \mathcal{A}(M)$ such that $A_1 \sqsubseteq A_3$ and $A_2 \sqsubseteq A_3$, by induction on A_1 .

- If $A_1 = x$ then $M \rightarrow_!^* x$ and $A_2 = x$ or \perp (Remark 39), then we set $A_3 = x$.
- If $A_1 = \perp$, by definition of \sqsubseteq we have $A_1 \sqsubseteq A_2$, we set $A_3 = A_2$.
- If $A_1 = \lambda x A'_1$, by Lemma 33 we have $M \rightarrow_!^* \lambda x N$ and $A'_1 \in \mathcal{A}(N)$. Then, by Lemma 38 we have $A_2 \in \mathcal{A}(\lambda x N)$ so either $A_2 = \perp$ and we set $A_3 = A_2$ or $\lambda x A'_2$ so $A'_2 \in \mathcal{A}(N)$. Then, by induction hypothesis, there is some A'_3 such that $A'_1 \sqsubseteq A'_3$ and $A'_2 \sqsubseteq A'_3$. We then set $A_3 = \lambda x A'_3$.
- If $A_1 = !A'_1$ we reason as in the previous case.
- If $A_1 = \text{der}(A'_1)$. Then $M \rightarrow_!^* \text{der}(N)$ and $A' \in \mathcal{A}(N)$. By Lemma 38 we have $A_2 \in \mathcal{A}(\text{der}(N))$ so either $A_2 = \perp$ or $A_2 = \text{der}(A'_2)$ such that $A'_2 \in \text{der}(A'_1)$, and therefore we can use our induction hypothesis to obtain an upper bound A'_3 of A'_1 and A'_2 and set $A_3 = \text{der}(A'_3)$, which is indeed an upper bound of $\{A_1, A_2\}$ by contextual closure of the approximation order.

- If $A_1 = A'_1[A''_1/x]$, then $M \rightarrow^*_! N'[N''/x] = N$ with $A'_1 \in \mathcal{A}(N')$ and $A''_1 \in \mathcal{A}(N'')$. A''_1 is not of shape $L(\langle ! - \rangle)$, then neither is N'' . Again, $A_2 \in \mathcal{A}(N_1)$, then $A_2 = A'_2[A''_2/x]$ with $A'_2 \sqsubseteq N'$ and $A''_2 \sqsubseteq N''$. By induction hypothesis, we can find A'_3 an upper bound of $\{A'_1, A'_2\}$ and A''_3 an upper bound of $\{A''_1, A''_2\}$. We conclude by setting $A_3 = A'_3[A''_3/x]$.
- If $A_1 = A'_1 A''_1$, we reason similarly, but using the fact that A'_1 cannot be of shape $L(\langle \lambda x - \rangle)$. ◀

In order to define Böhm trees, we follow the long-established tradition for term rewriting systems, including CbN and CbV [6, 15, 13, 5, 7, 29], based on the ideal completion. This method construct the set of ideals of approximants (ordered by a direct partial order). Böhm trees are then identified with such ideals. The finite and infinite ideals represent respectively the finite and infinite trees. For a λ -term M , its Böhm tree is the ideal generated by its set of approximants; equivalently, it can be seen as the supremum of these approximants in the associated directed-complete domain. This domain admits a concrete presentation as a coinductive grammar extending that of approximants, where constructors may be unfolded infinitely often.

Given $M \in \text{dBang}$, $\mathcal{A}(M)$ has a supremum, noted $\cup \mathcal{A}(M)$, which is a potentially infinite tree.

► **Definition 42** (Böhm Tree). *The Böhm tree of a term M in dBang, is given by $\cup \mathcal{A}(M)$, and denoted $\text{BT}_!(M)$,*

Böhm trees satisfy the following properties, easily checked by an examination of the definitions (we already used these facts on approximants for previous results, they lift immediately to Böhm trees).

► **Proposition 43** (Some properties of Böhm trees).

- If M is in normal form, $\text{BT}_!(M) = M$
- $\text{BT}_!(M) = \perp$ if and only if $\mathcal{A}(M) = \{\perp\}$ ⁷, if and only if any reduct of M is a redex.
- If $M \rightarrow^* N$, $\text{BT}_!(M) = \text{BT}_!(N)$, in particular $\text{BT}_!(\text{der}(!M)) = \text{BT}_!(M)$, $\text{BT}_!(\langle \lambda x M \rangle N) = \text{BT}_!(M[N/x])$ and $\text{BT}_!(M[!N/x]) = \text{BT}_!(M\{N/x\})$
- $\text{BT}_!(\langle \lambda x M \rangle) = \langle \lambda x \text{BT}_!(M) \rangle$
- $\text{BT}_!(\langle !M \rangle) = \langle !(\text{BT}_!(M)) \rangle$
- If $\text{BT}_!(M) \neq L(\langle \lambda x - \rangle)$, then $\text{BT}_!(MN) = \text{BT}_!(M)\text{BT}_!(N)$
- If $\text{BT}_!(M) \neq L(\langle ! - \rangle)$, then $\text{BT}_!(\text{der}(M)) = \text{der}(\text{BT}_!(M))$ and $\text{BT}_!(N[M/x]) = \text{BT}_!(N)[\text{BT}_!(M)/x]$

In particular, we can infer from the facts above that if $\text{BT}_!(M)$ is an application, then it is equal to $(x)\text{BT}_!(N)$ for some N (hence $M \rightarrow^* L(x)N$).

► **Example 44.** From example 37, we can infer that $\text{BT}_!(\Omega) = \perp$, $\text{BT}_!(Y_x^n)$ is the infinite application $x!x!x!\dots$, and $\text{BT}_!(Y_x^v)$ is also an infinite application $xxxxxx\dots$. This is the intended behaviour of Böhm trees, as in this category of terms they represent, at the limit, the amount of result produced by a computation, even if the term itself has no normal form.

⁷ This distinguishes our Böhm trees from those of Mazza and Dufour [20], in which $\text{BT}_!(M) = \perp$ as soon as M has no surface normal form. Indeed, for technical reasons (relative to CbV embeddings), if $\text{BT}_!(M) = \perp$, we need to have $\text{BT}_!(\langle \lambda x M \rangle) = \langle \lambda x \perp \rangle$, and not \perp , because $\lambda x \perp$ embeds to $\langle !(\lambda x \perp) \rangle$, while \perp embeds to \perp , which loses the exponential and breaks commutation properties between Böhm trees and CbV embedding (Theorem 66). This distinction vanishes at the semantical level: as soon as we consider Taylor expansion of Böhm trees, the trees having no surface normal form are given an empty expansion (Definition 45).

2.5 The commutation between Böhm and Taylor approximation

We now combine the definitions and results presented so far and conclude this section with the Commutation Theorem. Let us first define the Taylor expansion of a Böhm tree.

► **Definition 45.** We extend the definition of $\mathcal{T}(M)$ to terms in dBang_\perp by setting $\mathcal{T}(\perp) = \emptyset$ (recall that $f[\emptyset] = \emptyset$ for any full context f). In other words, there is no δBang term m such that $m \triangleleft! \perp$.

One immediate consequence of this definition is that, for M in dBang_\perp , $\mathcal{T}(M) \neq \emptyset$ if M contains no \perp as a surface subterm. In other words, M expands to a non-empty set if and only if the \perp are under exponentials ! (in that case, these exponentials can be approximated by empty bags). We also observe that if $M \sqsubseteq N$, then $\mathcal{T}(M) \subseteq \mathcal{T}(N)$.

► **Definition 46** (Taylor expansion of Böhm trees). Given $M \in \text{dBang}$, $\mathcal{T}(\text{BT}_!(M)) = \cup_{a \in \mathcal{A}(M)} \mathcal{T}(a)$.

► **Example 47.** Following the terms of previous examples, we easily check that $\mathcal{T}(\text{BT}_!(\Omega)) = \emptyset$ (because $\mathcal{A}(\Omega) = \{\perp\}$). Recall that $\mathcal{A}(Y_x^n) = \{\perp, x\perp, x!\perp, \dots\}$. The first of these approximants having a non empty expansion is $x!\perp$, approximated by $x[]$, $\mathcal{T}(\text{BT}_!(Y_x^n)) = \{x[], x[x[]], \dots, x[\dots]\}$. And $\mathcal{T}(\text{BT}_!(Y_x^v)) = \emptyset$. Indeed, all approximants of Y_x^v are of shape $xxx\dots x\perp$, and have a surface \perp that expands to \emptyset .

So in these example, by checking examples 37 and 16, we observe immediately the identity of $\text{TNF}(M)$ and $\mathcal{T}(\text{BT}_!(M))$ which is proved in the remainder of this section.

Remark that, since $\mathcal{A}(M)$ is an ideal, $\mathcal{T}(\text{BT}_!(M))$ is a directed union. The purpose of Taylor expansion is to approach a term by finitary resource terms only, so we do not consider any infinite supremum of this union, and keep a set of terms, this approximation being inductive; while Böhm trees are coinductive and consist in infinite objects.

► **Lemma 48.** Let $A \sqsubseteq M$, then $\mathcal{T}(A) \subseteq \mathcal{T}(M)$.

Proof. By induction on A . If $A = \perp$ then $\mathcal{T}(A) = \emptyset \subseteq \mathcal{T}(M)$. If $A = x$ then $M = x$ and $\mathcal{T}(A) = \mathcal{T}(M) = \{x\}$. If $A = \lambda x A'$, then $M = \lambda x M'$ with $A' \sqsubseteq M'$. By induction hypothesis, $\mathcal{T}(A') \subseteq \mathcal{T}(M')$. Then, $\mathcal{T}(A) = \{\lambda x a' \mid a' \in \mathcal{T}(A')\} \subseteq \mathcal{T}(M) = \{\lambda x m' \mid m' \in \mathcal{T}(M')\}$. The other cases are treated similarly by routine induction. ◀

► **Lemma 49.** Let $A \in \mathcal{A}(M)$, then $\mathcal{T}(A) \subseteq \text{TNF}(M)$

Proof. We have some M' such that $M \rightarrow_!^* M'$ and $A \sqsubseteq M'$. We have that $\text{nf}(\mathcal{T}(M)) = \text{nf}(\mathcal{T}(M'))$ by Lemma 28. By Lemma 48 we have $\mathcal{T}(A) \subseteq \mathcal{T}(M')$. We conclude by observing that terms in $\mathcal{T}(A)$ are in normal form, and that normal terms in $\mathcal{T}(M')$ must also belong to $\text{nf}(\mathcal{T}(M'))$ as they are not affected by any reduction. ◀

► **Lemma 50.** let $m \triangleleft! M$ in normal form, then there exists an approximant A such that $A \sqsubseteq M$ and $m \triangleleft! A$ (remember that $m \triangleleft! M$ and $m \in \mathcal{T}(M)$ are the same thing).

Proof. By induction on m .

- If $m = x$, then $M = x$ and we set $A = x$.
- If $m = \lambda x n$, then $M = \lambda x N$ with $n \triangleleft! N$. Since n must be in normal form, we can apply the induction hypothesis to obtain some $A' \sqsubseteq N$ such that $n \triangleleft! A'$. We then set $A = \lambda x A'$.
- If $m = m_1 m_2$, then $M = M_1 M_2$ with $m_i \triangleleft! M_i$. Again, by induction hypothesis, we have $A_1 \sqsubseteq M_1$ and $A_2 \sqsubseteq M_2$ with $m_i \triangleleft! A_i$. It remains to show that $A_1 A_2$ belongs to the set of approximants described in Definition 31. Notice that m_1 cannot be of shape $l \langle \lambda x^- \rangle$, then since $m_1 \triangleleft! A_1$, A_1 cannot be a bottom or an abstraction. A simple examination of the syntax of approximants is enough to conclude that $A_1 A_2$ indeed belongs to it.

- If $m = \mathbf{der}(n)$, then we again obtain $n \triangleleft! N$ and some $A' \sqsubseteq N$ with $n \triangleleft! A$. Since n cannot be of shape $l\langle[-]\rangle$. (since m is normal), we can again check Definition 31 to conclude that $A = \mathbf{der}(A)'$ is an approximant, and that $m \triangleleft! A$.
- If $m = n[p/x]$, we reason as for the application case, but using the case that p cannot be of shape $l\langle[-]\rangle$.
- If $m = [n_1, \dots, n_k]$, then $M = !N$ with $n_i \triangleleft! N$ for all $i \in \{1, \dots, k\}$. Then by induction hypothesis, there is $A_i \sqsubseteq N$ with $n_i \triangleleft! A_i$. We then take $A = !A_i$, which works for any i . ◀

We now have all the necessary ingredients for the Commutation Theorem.

► **Theorem 51.** Let $M \in \mathbf{dBang}$. $\mathcal{T}(\mathbf{BT}_!(M)) = \mathbf{TNF}(M)$.

Proof. We proceed by double inclusion.

- Take $m \in \mathcal{T}(\mathbf{BT}_!(M))$, then there exists some $A_0 \in \mathcal{A}(M)$ such that $m \in \mathcal{T}(A_0)$, by Definition 46. There is M_0 such that $M \rightarrow_!^* M_0$ and $A_0 \sqsubseteq M_0$. We can therefore apply Lemma 49 to conclude that $m \in \mathbf{TNF}(M_0)$, which is equal to $\mathbf{TNF}(M)$ by Lemma 28.
- Assume $m \in \mathbf{TNF}(M)$. By Lemma 29 there exists M_0 such that $M \rightarrow_!^* M_0$ and $m \in \mathcal{T}(M_0)$. By Lemma 50, there is $A \sqsubseteq M_0$ such that $m \in \mathcal{T}(A)$. By definition, $A \in \mathcal{A}(M)$, so we conclude that $m \in \mathcal{T}(\mathbf{BT}_!(M))$. ◀

3 Translations

This section is about the approximation theory of dCBN dCBV, in particular about how the embeddings into dBang can profit from the result of previous Section. These embeddings, some variants and their properties have been well studied in the literature about dBang [16, 30, 8, 9, 10, 11], we start by recalling some definitions and main results.

The syntax of terms in both dCBV and dCBN is the same:

$$M, N ::= x \mid \lambda x M \mid MN \mid M[N/x]$$

While surfaces contexts (in dCBN and dCBV respectively) are defined as:

- $S_n ::= \square \mid S_n M \mid \lambda x S_n \mid S_n[N/x]$
- $S_v ::= \square \mid S_v M \mid M S_v \mid S_v[M/x] \mid M[S_v/x]$

Furthermore, the values V are defined as either a variable x or a λ -abstraction $\lambda x M$. As in dBang, we define *list contexts* as $L := \square \mid L[M/x]$. The reduction rules are defined as follows:

- In dCBN: $L\langle\lambda x M\rangle N \rightarrow_n L\langle M[N/x]\rangle$ and $M[N/x] \rightarrow_n M\{N/x\}$
- In dCBV: $L\langle\lambda x M\rangle N \rightarrow_v L\langle M[N/x]\rangle$ and $M[L\langle V\rangle/x] \rightarrow_v L\langle M\{N/x\}\rangle$

The surface reduction of dCBV (resp. dCBN) is the surface closure of rewrite rules above.

We are now ready to define the translations of dCBN (noted $(\cdot)^n$) and dCBV (noted $(\cdot)^v$) into dBang. It is worth noting that there exist multiple translations of dCBV into dBang (or the original Bang calculus without explicit substitutions). The first one [23], inspired by Girard *second translation*, does not preserve normal forms (xy translates to $\mathbf{der}(!x)!y$). Another translation was then proposed [16], which fixes this problem by simplifying the created redexes by the translations on the fly. This is the translation that we use here. It is worth noticing that this translation does not satisfy reverse simulation from dBang to dCBV: $((\lambda xx)(\lambda yy))z^n = \mathbf{der}((\lambda x!x)!(\lambda y!y))!z \rightarrow_! \mathbf{der}(!(\lambda y!y))!z$, while the last term does not correspond to a valid translation. This issue has been addressed by a third translation [10] (which adds some $!$ and $\mathbf{der}()$ in the translation). However, reverse simulation

is not necessary in our case, and we will keep with the translation proposed in [16], although we are convinced that the same developments could be carried out with the translation from [10]. This choice is motivated by the fact that the translation we consider fits better with the Linear Logic discipline from which stems Taylor expansion, and also because when considering the study of meaningfulness, we can rely on results that have been proved for this translation. Moreover, this translation enjoys a weaker property than strict reverse simulation, that we call embedding (see Lemma 65) and which is sufficient for our study (Böhm trees and Taylor expansion concern mostly iterated reductions \rightarrow^* , and one-step, thus reverse simulation is then not mandatory for our results).

$$\begin{array}{ll}
x^n = x & x^v = !x \\
(\lambda x M)^n = \lambda x M^n & (\lambda x M)^v = !(\lambda x M^v) \\
(M N)^n = M^n !N^n & (M N)^v = \begin{cases} L(P) N^v & \text{if } M^v = L(!P) \\ \mathbf{der}(M^v) N^v & \text{otherwise} \end{cases} \\
(M[N/x])^n = M^n [!N^n / x] & (M[N/x])^v = M^v [N^v / x]
\end{array}$$

We abusively extend the translations to list contexts: let $\circ \in \{n, v\}$; if $L = \square[M_1/x_1] \dots [M_k/x_k]$, we write $L^\circ = \square[M_1^\circ/x_1] \dots [M_k^\circ/x_k]$.

We are now ready to define meaningfulness in both systems:

► **Definition 52** (dCBV and dCBN meaningfulness [30]). *Given a testing context*

$$T := \square \mid T N \mid (\lambda x T) N$$

we say that a term is $M \in \text{dCBV}$ is dCBV-meaningful (resp. $M \in \text{dCBN}$ and is dCBN-meaningful) if $T \langle M \rangle \rightarrow_v^ V$ for some value V (resp. $T \langle M \rangle \rightarrow_n^* \lambda x x$).*

It has been shown that dCBN dCBV can be simulated through their encoding in dBang, and that the meaningfulness of dCBV and dCBN coincide with the one of dBang (Definition 7):

► **Theorem 53.**

1. If $M \rightarrow_n N$ (resp. $M \rightarrow_v N$) then $M^n \mapsto_! N^n$ (resp. $M^v \mapsto_! N^v$) [16, Lemma 4.6].
2. M is dCBN-meaningful iff M^n is meaningful [30, Theorem 25].
3. M is dCBV-meaningful iff M^v is meaningful [30, Theorem 30].

We will now study their Böhm Trees and Taylor expansion in regard of the translation of dCBV and dCBN. This will be useful for a characterization of meaningfulness in those calculi.

3.1 Taylor and Böhm approximation for dCBN and dCBV

3.1.1 Taylor expansion for dCBN and dCBV

► **Definition 54** (Resource approximations of dCBN).

We define an approximation \triangleleft_n relation between δBang ⁸ and dCBN. Notice that despite the approximants being defined in δBang , there is no dereliction needed in the case of dCBN.

- $x \triangleleft_n x$
- $\lambda x m \triangleleft_n \lambda x M$ if $m \triangleleft_n M$.

⁸ We do not need to define a specific resource calculus for dCBN nor dCBV, since δBang semantics precisely subsumes both approximation theories.

- $m[n_1, \dots, n_k] \triangleleft_n MN$ if $m \triangleleft_n M$ and $n_i \triangleleft_n N$ for any $i \leq k$
- $m[[n_1, \dots, n_k]/x] \triangleleft_n M[N/x]$ if $m \triangleleft_n M$ and $n_i \triangleleft_n N$ for any $i \leq k$

Taylor expansion is again defined as sets of approximations: $\mathcal{T}^n(M) = \{m \in \delta\text{Bang} \mid m \triangleleft_n M\}$.

► Remark 55. dCBN approximants can be described as:

$$m, n := x \mid m [n_1, \dots, n_k] \mid \lambda xm \mid m[[n_1, \dots, n_k]/x]$$

► Definition 56 (Resource approximation for dCBV). *We define the relation $m \triangleleft_v M$ for $m \in \delta\text{Bang}$ and $M \in \text{dCBV}$.*

- $[x, \dots, x]_k \triangleleft_v x$ for any $k \in \mathbb{N}$.
- $[\lambda xm_1, \dots, \lambda xm_k] \triangleleft_v \lambda xM$ if $m_i \triangleleft_v M$ for any $i \leq k$.
- $\text{der}(m)n \triangleleft_v MN$ if $m \triangleleft_v M$, $n \triangleleft_v N$ and $M \notin V$
- $mn \triangleleft_v VN$ if $[m] \triangleleft_v V$ and $n \triangleleft_v N$
- $m[n/x] \triangleleft_v M[N/x]$ if $m \triangleleft_v M$ and $n \triangleleft_v N$.

Taylor expansion is defined as $\mathcal{T}^v(M) = \{m \in \delta\text{Bang} \mid m \triangleleft_v M\}$. Notice that, so as Taylor expansion commutes with this embedding, Taylor approximation also suppresses derelictions redexes, so as the expansion also preserves normal forms.

Both in dCBV and dCBN we also define $\text{TNF}(M)$ as the set containing the normal forms of resource approximants of M .

► Lemma 57. Let $M \in \text{dCBN}$. $\mathcal{T}^n(M) = \mathcal{T}(M^n)$.

Proof. Considering a resource term $m \in \delta\text{Bang}$, we can show that $m \triangleleft_n M$ (see Definition 54) if and only if $m \triangleleft! M^n$ (see Figure 1), by an immediate induction on M .

- x is the only approximation of $x = x^n$.
- $m \triangleleft_n \lambda xN \in \text{dCBN}$ if and only if $m = \lambda xn$ with $n \triangleleft_n N$, if and only if (induction hypothesis) $n \triangleleft! N^n$, and then if and only if $m \triangleleft! \lambda xN^n = (\lambda xN)^n$
- $m \triangleleft_n NP \in \text{dCBN}$ if and only if $m = n[p_1, \dots, p_k]$ with $n \triangleleft_n N$, $p_i \triangleleft! P$ for any $i \leq k$ if and only if (induction hypothesis) $n \triangleleft! N^n$ and $p_i \triangleleft! P^n$, and then if and only if $m \triangleleft! (NP)^n$.
- Case $M = N[P/x]$ is similar to the previous one. ◀

► Corollary 58. Let $M \in \text{dCBN}$. $\text{TNF}(M) = \emptyset \leftrightarrow \text{TNF}(M^n) = \emptyset$

dCBV translation enjoys the same property. Notice that the translation of application, as well as its Taylor expansion, is described by case on its first component such that the translation (resp. the expansion) of a term in normal form does not lead to any reducible pattern.

► Lemma 59. Let $M \in \text{dCBV}$. $\mathcal{T}^v(M) = \mathcal{T}(M^v)$.

Proof. By induction on M :

- For any $k \in \mathbb{N}$, $[x]_k \triangleleft_v x$ and $[x]_k \triangleleft! x^v = !x$.
- For any $k \in \mathbb{N}$, $[\lambda xm_1, \dots, \lambda xm_k] \triangleleft_v \lambda xM$ iff $m_i \triangleleft_v M$ for all i iff (induction hypothesis) $m_i \triangleleft! M^v$ iff $[\lambda xm_1, \dots, \lambda xm_k] \triangleleft! (\lambda xM)^v = !(\lambda xM^v)$.
- $m \triangleleft_v NP$.
 - Either N is an application. Then $m \triangleleft_v NP$ iff $m = \text{der}(n)p$ with $n \triangleleft_v N$ and $p \triangleleft_v P$ iff (induction hypothesis) $n \triangleleft! N^v$ and $p \triangleleft! P^v$ iff $\text{der}(n)p \triangleleft! (NP)^v$.
 - Either $N = !N'$, and then $m \triangleleft_v NP$ iff $m = n'p$ with $n' \triangleleft_v N'$ and $p \triangleleft_v P$ iff $n' \triangleleft! N'^v$, and $p \triangleleft! P^v$ iff $n'p \triangleleft! (NP)^v$. ◀

► Corollary 60. $\text{TNF}(M) = \emptyset \leftrightarrow \text{TNF}(M^v) = \emptyset$.

3.1.2 Böhm trees for dCBN and dCBV

► **Definition 61.** *The syntax of dCBN approximants is given by:*

$$A_n ::= N_\lambda \mid \lambda x A_n \quad N_\lambda ::= x \mid \perp \mid N_\lambda A_n$$

Notice that this syntax coincides with inductive head normal forms, as it is standard in Böhm trees literature [12].

► **Definition 62.** *The syntax of dCBV approximants is given by:*

$$\begin{aligned} A_v &::= A_\lambda \mid \lambda x A_v \mid A_v[A_{\lambda x}/x] \\ A_\lambda &::= x \mid \perp \mid A_\lambda A_v \mid A_\lambda[A_{\lambda x}/x] \\ A_{\lambda x} &::= A_\lambda A_v \mid A_{\lambda x}[A_{\lambda x}/x] \end{aligned}$$

We then define Böhm trees in the usual way, setting $\text{BT}_\circ(M) = \cup\{A_\circ \mid M \rightarrow_\circ^* N, A_\circ \sqsubseteq N\}$, for $\circ \in \{n, v\}$, (and where $M \sqsubseteq N$ means again that M is obtained by replacing subterms of N by \perp). We do not provide a proof for their well-definedness, as these Böhm trees have been thoroughly established in CbN [12] and CbV [29]. We leave it to the reader to verify that the proof of Lemma 41 can be adapted to dCBV and dCBN, incorporating the distant setting into these standard results.

In the following, we extend translations $(\cdot)^n$ and $(\cdot)^v$ to approximants by setting $\perp^n = \perp^v = \perp$, and study the commutation between Böhm approximation and the embedding (Theorem 66), that will allow us to transport our Commutation Theorem from dBang (Theorem 51) into dCBV and dCBN.

The following lemmas, 63 and 64, are proved by a routine induction on the syntax of terms.

► **Lemma 63.**

1. *M is a dCBN approximant if and only if M^n is a dBang approximant.*
2. *M is a dCBV approximant if and only if M^v is a dBang approximant.*

► **Lemma 64** (Substitution).

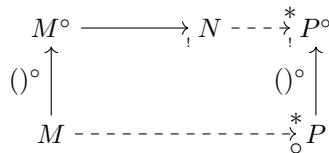
1. *Let $M, N \in \text{dCBN}$. $M^n\{N^n/x\} = M\{N/x\}^n$.*
2. *Let $M, V \in \text{dCBV}$. $M^v\{V^v/x\} = M\{V/x\}^v$.*

Now we can prove the weaker form of reverse simulation mentioned in the preamble of this section. Notice that this result coincides precisely to the notion of embedding studied by Dufour and Mazza [20].

► **Lemma 65** (Embedding).

1. *Let $M \in \text{dCBN}$. If $M^n \rightarrow_! N$, then there is some $P \in \text{dCBN}$ such that $M \rightarrow_n^* P$ and $N \rightarrow_!^* P^n$.*
2. *Let $M \in \text{dCBV}$. If $M^v \rightarrow_! N$, then there is some $P \in \text{dCBV}$ such that $M \rightarrow_v^* P$ and $N \rightarrow_!^* P^v$.*

Proof. For $\circ \in \{v, n\}$, the statements of the lemma can be depicted as follows, where the dashed lines and the term P are the one to be established:



First, notice that the translations $(\cdot)^v$ and $(\cdot)^n$ generate only redexes of shape $L\langle\lambda xM\rangle N$ and $M[L\langle!N\rangle/x]$ ⁹.

(1) By induction on the reduction $M^n \rightarrow_! N$.

- $M^n = L^n\langle\lambda xN_1^n\rangle!N_2^n$ and $N = L^n\langle N_1^n[!N_2^n/x]\rangle$. We have $N = (L\langle N_1[N_2/x]\rangle)^n$ by definition of $(\cdot)^n$. Since $M = L\langle\lambda xN_1\rangle N_2$, we have $M \rightarrow_! L\langle N_1[N_2/x]\rangle$, and we are done, setting $P = L\langle N_1[N_2/x]\rangle$.
- $M^n = N_1^n[L^n\langle!N_2^n\rangle/x]$ and $N = L^n\langle N_1^n\{N_2^n/x\}\rangle$. By Lemma 64, $N = (L\langle N_1\{N_2/x\}\rangle)^n$, and again we are done, setting $P = L\langle N_1\{N_2/x\}\rangle$, since $M = N_1[L\langle N_2\rangle/x] \rightarrow_! L\langle N_1\{N_2/x\}\rangle$.
- The reduction is contextual:
 - $M^n = N_1^n!N_2^n$ and $N = N'_1!N'_2$ with $N_1 \rightarrow_n N'_1$. By induction hypothesis, there is some P_1 such that $N_1 \rightarrow_n^* P_1$ and $N'_1 \rightarrow_!^* P_1^n$. Then we set $P = P_1 N_2$, and we have indeed $M = N_1 N_1 \rightarrow_n^* P$ and $N = N'_1!N'_2 \rightarrow_!^* P_1^n!N_2^n = (P_1 N_2)^n = P^n$.
 - $M^n = N_1^n!N_2^n$ and $N = N_1^n!N'_2$ with $N_2 \rightarrow_! N'_2$. By induction hypothesis, we have some P_2 such that $N_2 \rightarrow_n^* P_2$ and $N'_2 \rightarrow_!^* P_2^n$. We then set $P = N_1 P_2$.
 - $M^n = N_1^n[!N_2^n/x]$, $N = N_1^n[!N'_2/x]$ with $N_2 \rightarrow_! N'_2$. We reason as in the previous case.
 - $M^n = \lambda xN_0^n$, $N = \lambda xN'_0$ with $N_0^n \rightarrow_! N'_0$. By induction hypothesis, there is P_0 such that $N_0 \rightarrow_n^* P_0$ and $N'_0 \rightarrow_!^* P_0^n$. We then set $P = \lambda xP_0$.

(2) By induction on the reduction $M^v \rightarrow_! N$. The first two cases are similar to before except the position of the exponential.

- $M^v = L^v\langle\lambda xN_1^v\rangle N_2^v$ and $N = L^v\langle N_1^v[N_2^v/x]\rangle$. We have $N = (L\langle N_1[N_2/x]\rangle)^v$ by definition of $(\cdot)^v$, and $M = L\langle\lambda xN_1\rangle N_2$. We set $P = L\langle N_1[N_2/x]\rangle$, satisfying $M \rightarrow_v P$ and $N \rightarrow_!^0 P^v$.
- $M^v = N_1^v[L^v\langle!N_2\rangle/x]$ and $N = L^v\langle N_1^v\{N_2^v/x\}\rangle$. By Lemma 64 we have $N = (L\langle N_1\{N_2/x\}\rangle)^v$. We then set $P = L\langle N_1\{N_2/x\}\rangle$.
- The reduction is contextual. We only detail the case where the reduction occurs in the left member of an application and under a dereliction; the other cases follow from an application of the induction hypothesis as before. The second of these two case is important, as it is the responsible for the only configuration where P must be distinct from N .
 - $M^v = \mathbf{der}(N_1^v)N_2^v$ (in that case $M = N_1 N_2$ with N_1 not being of shape $L\langle V \rangle$ for any value V) and $N = \mathbf{der}(N'_1)N_2^v$. By induction hypothesis, there is some P_1 such that $N_1 \rightarrow_v^* P_1$ and $N'_1 \rightarrow_!^* P_1^v$. We then have two possibilities:
 - * $P_1^v \neq L\langle!\neg\rangle$ (P_1 is not a value). Then, $(P_1 N_2)^v = \mathbf{der}(P_1^v)N_2^v$. We then set $P = P_1 N_2$, and we have $M = N_1 N_2 \rightarrow_v^* P_1 N_2$ and $N \rightarrow_!^* \mathbf{der}(P_1^v)N_2^v = P^v$.
 - * $P_1^v = L^v\langle!Q^v\rangle$. Then, $(P_1 N_2)^v = L^v\langle Q^v \rangle N_2^v$. We have $N \rightarrow_!^* \mathbf{der}(L^v\langle!Q^v\rangle)N_2^v \rightarrow_! L^v\langle Q^v \rangle N_2^v$ by a single reduction step¹⁰. We then set $P = L\langle Q \rangle N_2$. It verifies $N \rightarrow_!^* P^v$ as we just saw. We also have $M \rightarrow_v^* P$, because $M = N_1 N_2$ and $N_1 \rightarrow_v^* P_1$. Since $P_1^v = L^v\langle!Q^v\rangle$, it follows that Q is a value (either a variable or an abstraction) and that $P_1 = L\langle Q \rangle$, by definition of $(\cdot)^v$. ◀

► Theorem 66.

1. Let $M \in \text{dCBN}$. $(\text{BT}_n(M))^n = \text{BT}_!(M^n)$.
2. Let $M \in \text{dCBV}$. Then $(\text{BT}_v(M))^v = \text{BT}_!(M^v)$.

⁹ Redexes like $\mathbf{der}(L\langle!N\rangle)$ can however appear during reductions, from translations $(\cdot)^v$, but not in the translation itself.

¹⁰ These steps are called *administrative* in Arrial, Guerrieri and Kessner's work [9].

XX:20 Approximation theory for distant Bang calculus

Proof. For this proof we will benefit from the properties of Böhm trees stated in Proposition 43.

(1) We proceed by coinduction on $\text{BT}_n(M)$.

- If $\text{BT}_n(M) = \perp$, then $\mathcal{A}(M) = \{\perp\}$. We need to show that $\mathcal{A}(M^n) = \{\perp\}$. Consider $A \in \mathcal{A}(M^n)$, we have $M^n \rightarrow_!^* N$ with $A \sqsubseteq N$. By Lemma 65, we have some P such that $M \rightarrow_!^* P$ and $N \rightarrow_!^* P^n$. By Lemma 35, $A \sqsubseteq P^n$. Now, observe that P must be some dCBN redex, otherwise $\mathcal{A}(M)$ would contain other approximations than \perp . Then, by simulation (Theorem 53), P^n also is a redex, and since the syntax of approximants contains no redex, necessarily $A = \perp$. We conclude that $\text{BT}_!(M^n) = \perp$.
- If $\text{BT}_n(M) = x$, then $\text{BT}_n(M)^n = x = \text{BT}_!(x) = \text{BT}_!(x^n)$.
- If $\text{BT}_n(M) = \lambda x \text{BT}_n(N)$, then $(\text{BT}_n(M))^n = (\lambda x \text{BT}_n(N))^n = \lambda x (\text{BT}_n(N))^n$ (by definition of $(\cdot)^n$). By coinduction hypothesis, $(\text{BT}_n(N))^n = \text{BT}_!(N^n)$. Then, $(\text{BT}_n(M))^n = \lambda x \text{BT}_!(N^n) = \text{BT}_!((\lambda x N^n)) = \text{BT}_!(M^n)$.
- If $\text{BT}_n(M)$ is an application, then it is equal to some $(x) \text{BT}_n(N)$. Then, we have $(\text{BT}_n(M))^n = (x)! (\text{BT}_n(N))^n = (x)! \text{BT}_!(N^n)$, by coinduction hypothesis, which is equal to $\text{BT}_!((x)! N^n) = \text{BT}_!((x) N^n) = \text{BT}_!(M^n)$.
- $\text{BT}_n(M)$ cannot contain any explicit substitution, as they always correspond to redexes in dCBN.

(2) We proceed by coinduction on $\text{BT}_v(M)$.

- If $\text{BT}_v(M) = \perp$, we reason as above, using this time the second item of Lemma 65.
- If $\text{BT}_v(M) = x$, then $(\text{BT}_v(M))^v = x^v = !x = \text{BT}_!(!x) = \text{BT}_!(x^v)$.
- If $\text{BT}_v(M) = \lambda x \text{BT}_v(N)$, then $(\text{BT}_v(M))^v = !(\lambda x (\text{BT}_v(N))^v)$. By coinduction hypothesis, it is equal to $!(\lambda x \text{BT}_!(N^v)) = \text{BT}_!(!(\lambda x N^v)) = \text{BT}_!((\lambda x N^v)) = \text{BT}_!(M^v)$.
- If $\text{BT}_v(M)$ is an application, then it must be equal to some $x([\text{BT}_v(N_i)/y_i])_{1 \leq i \leq k} \text{BT}_v(N_0^v)$ (because in this case M reduces to some $L(x)N_0$). Then $(\text{BT}_v(M))^v = x([\text{BT}_v(N_i)^v/y_i]_{1 \leq i \leq k} \text{BT}_v(N_0))^v$. By coinduction hypothesis, $(\text{BT}_v(N_j))^v = \text{BT}_!(N_j^v)$ for $j \in \{0, \dots, k\}$. Then $(\text{BT}_v(M))^v = x([\text{BT}_!(N_i^v)/y_i]_{1 \leq i \leq k} \text{BT}_!(N_0^v))$. Again, for $i \in \{1, \dots, k\}$, $\text{BT}_!(N_i^v)$ cannot be an exponential, since those explicit substitution must not be reducible. Then, $(\text{BT}_v(M))^v = \text{BT}_! \left(x ([N_i^v/x]_{i \in \{1, \dots, k\}} N_0^v) \right) = \text{BT}_!(M^v)$.
- If $\text{BT}_v(M) = \text{BT}_v(N_1)[\text{BT}_v(N_2)/x]$, then again $\text{BT}_v(N_2)$ cannot be a value, hence $(\text{BT}_v(M))^v = \text{BT}_!(N_1^v)[\text{BT}_!(N_2^v)/x]$ (by coinduction hypothesis) $= \text{BT}_!(N_1^v [N_2^v/x]) = \text{BT}_!((N_1 [N_2/x])^v)$.

◀

Thanks to the compatibility of Böhm trees and Taylor expansion with the translations of dCBN and dCBV into dBang we can apply our commutation result for dBang to both calculi. Although these results have been well established (to our knowledge, only for non-distant CbN and CbV), this application illustrates that the subsuming power of dBang has been brought in the approximation theory, which was our purpose.

► Proposition 67.

1. Let $M \in \text{dCBV}$. $\mathcal{T}^v(\text{BT}_v(M)) = \text{nf}(\mathcal{T}^v(M))$.
2. Let $M \in \text{dCBN}$. $\mathcal{T}^v(\text{BT}_n(M)) = \text{nf}(\mathcal{T}^n(M))$.

Proof. Let $\circ \in \{n, v\}$. We have the following equalities:

$$\begin{aligned}
 \text{nf}(\mathcal{T}^\circ(M)) &= \text{nf}(\mathcal{T}(M^\circ)) && \text{By lemmas 59 and 57} \\
 &= \mathcal{T}(\text{BT}(M^\circ)) && \text{By Theorem 51} \\
 &= \mathcal{T}((\text{BT}_\circ(M))^\circ) && \text{By Theorem 66} \\
 &= \mathcal{T}_\circ(\text{BT}_\circ(M)) && \text{By lemmas 59 and 57}
 \end{aligned}$$

◀

4 Meaningfulness and Taylor expansion

The aim of this section is to establish a link between Taylor expansion and meaningfulness. In Call-By-Name, it is known, since Ehrhard and Regnier's seminal work [26], that a term is solvable if and only if its Taylor normal form is not empty. For Call-By-Value, more recent advances [19] have demonstrated analogous results with scrutable terms.

In dBang, we would like to be able to establish such a link between Taylor normal forms and meaningfulness. The first part of this result is achieved by Theorem 68 (recall that $m \triangleleft! M$ means $m \in \mathcal{T}(M)$). The second part presents significant challenges, which this section aims to address.

► **Theorem 68.** *Let $M \in \text{dBang}$. If M is meaningful, then $\text{TNF}(M) \neq \emptyset$.*

Proof. We show the contrapositive of the statement: assume that $\text{TNF}(M) = \emptyset$. By definition, for every $m \in \mathcal{T}(M)$, we have $m \rightarrow_{\delta}^* \emptyset$. Consider now any resource testing context t . We can easily establish that $t\langle m \rangle \rightarrow_{\delta}^* \emptyset$; since $\emptyset m = \emptyset$, $(\lambda x \emptyset)m = \emptyset$, and then by induction.

For M to be meaningful, there must exist a testing context T such that $T\langle M \rangle \rightarrow_{!s}^k !P$ for some P and $k \in \mathbb{N}$. We have $\emptyset \triangleleft! !P$. By iteratively applying Lemma 21, there is some term $s \triangleleft! T\langle M \rangle$ such that $s \rightarrow_{\delta_s}^k \emptyset$. By Lemma 13, and because testing contexts are included in surface contexts, we have some $t \triangleleft! T$, $m \triangleleft! M$ such that $s = t\langle m \rangle$.

This leads to a contradiction: $t\langle m \rangle \rightarrow_{\delta_s}^k \emptyset$, yet we have shown that $t\langle m \rangle \rightarrow_{\delta_s} \emptyset$ for any t . ◀

This result is encouraging for our study of Taylor expansion in dBang framework, as it applies to dCBN and dCBV through our previous simulation results (Corollary 58 and 60): Theorem 68 applies to both settings.

However, the converse of Theorem 68 (non-empty Taylor normal form implies meaningfulness) happens to be false in general. As mentioned in [30], some elementary terms, such as xx , are meaningful, whereas xy is not. Their intersection type system can distinguish between these terms, but it is unlikely that such a distinction can be made at the syntactic level using Taylor expansion.

Another approach would be to restrict ourselves to a (clash-free) fragment of dBang excluding patterns that do not make sense from a dCBV nor a dCBN discipline (such as xx), but again we can exhibit terms such as $(x!x)(x!x)$ ¹¹ that are meaningless but cannot reasonably be assigned an empty Taylor normal form.

We prove in the remaining of this section that the result holds independently for the two sublanguages of dBang consisting of terms translated from $(\cdot)^v$ and $(\cdot)^n$.

Our proof employs techniques adapted from CbN[26] and CbV [19], providing an initial characterization of the relationship between meaningfulness and Taylor expression in a distant setting. Although it is frustrating that we cannot prove the equivalence once for dBang and to apply it directly to its fragments; this limitation also raises an open question which we find to be of interest: is there a significant, bigger fragment of dBang for which the equivalence can be proven generically? Would this fragment cover terms not coming from a dCBV or dCBN translations? This is, for now, an open question.

We consider two strict subsets of dBang: dBang_V and dBang_N corresponding to terms obtained by translating from dCBN and dCBV, respectively. These fragments also have the advantage of excluding *clashes* - problematic dBang terms such as $\text{der}(\lambda x M)$ - which are often omitted from the analysis [16, 23] (see Remark 15).

► **Definition 69.**

$$\text{dBang}_N: M_n := x \mid \lambda x M_n \mid M_n!M_n \mid M_n[!M_n/x]$$

¹¹ Recall that $(xM)^v = !x(M^v)$ and $(Mx)^n = M^n!x$.

$$\begin{aligned} \text{dBang}_V : M_v &:= !x \mid !(\lambda x M_v) \mid L_v \langle \lambda x M_v \rangle M_v \mid L_v \langle x \rangle M_v \mid \mathbf{der}(M_v) M_v \mid M_v[M_v/x] \\ L_v &:= \square \mid L_v[M_v/x] \end{aligned}$$

A simple inspection of the definitions yields the following property :

► **Lemma 70.** *For any $M \in \text{dCBV}$, $M^v \in \text{dBang}_V$, and for any $M \in \text{dCBN}$, $M^n \in \text{dBang}_N$.*

Note that the converse holds for dBang_N , but not for dBang_V . For example $\mathbf{der}(!x)M \in \text{dBang}_V$, but no term of dCBV translates to this term due of the side condition of $(-)^v$ which ensures the preservation of normal forms. However, we cannot exclude these patterns from dBang_V as they can be obtained from some reduction as shown in Section 3.

We aim to ensure that our fragments are closed under reduction. Otherwise, a term in dBang_V , for example, could reduce in a term in dBang for which meaningfulness cannot be guaranteed (such as xx or clashes like $\mathbf{der}(\lambda x M)$). The following lemma can be proven by a standard induction.

► **Lemma 71.** *dBang_V and dBang_N are closed under $\rightarrow_!$.*

4.1 Meaningfulness and Taylor expansion for dCBN

The case of dCBN is relatively easy to handle, as we can adapt to the distant case the following properties; which correspond to well-known features of λ calculus:

- resource terms in normal form correspond to head normal forms.
- terms with head normal forms are meaningful.

► **Lemma 72.** *The normal forms of dBang_N are of shape $\lambda x_1 \dots x_k(x)!N_1 \dots !N_l$, where $k, l \in \mathbb{N}$.*

Proof. The following observations suffice:

- terms in dBang_N containing explicit substitutions are reducible.
- if the leftmost subterm of an application is not a variable, the entire term is reducible. ◀

Naturally, full normal form require the N_i to be in normal form too, but as we shall see, this is not relevant for studying meaningfulness, as these terms will be erased by an appropriate testing context. Previous observations can be brought at a resource level.

► **Lemma 73.** *Let $m \triangleleft_n M$ with $M \in \text{dBang}_N$. If $\mathbf{nf}(m) \neq \emptyset$, it is of shape $\lambda x_1 \dots x_k(x)\bar{n}_1 \dots \bar{n}_l$.*

We are now able to state the theorem establishing the classical link between Taylor expansion and meaningfulness in the case of dBang_N .

► **Theorem 74.** *Let $M \in \text{dBang}_N$. If $\text{TNF}(M) \neq \emptyset$, then M is meaningful.*

Proof. Consider some $p \in \text{TNF}(M)$, assumed non-empty. Then there is some $m \triangleleft_! M$ such that $m \rightarrow_{\delta}^* p$.

By Lemma 73, $p = \lambda x_1 \dots x_k(x)\bar{n}_1 \dots \bar{n}_l$ for some $k, l \in \mathbb{N}$. Proposition 27 allows us to focus on surface reduction: there are some $m' = \lambda x_1 \dots x_k(x)\bar{n}'_1, \dots, \bar{n}'_l$ such that $m \rightarrow_{\delta_s}^* m' \rightarrow_{\delta}^* p$ (the second part of the reduction acting inside the bags).

Then, by iteratively applying Lemma 18, we obtain $M' \in \text{dBang}$ such that $m' \triangleleft_! M'$ and $M \rightarrow_{\delta}^* M'$. By the definition of the approximation relation $\triangleleft_!$, we have $M' = \lambda x_1 \dots x_k(x)!N'_1 \dots !N'_l$ where $\bar{n}'_i \triangleleft_! !N'_i$.

We now define the appropriate testing context $T = ((\lambda x \square)!(\lambda y_1 \dots y_l z_0))!z_1 \dots !z_k$ where the z_i are chosen distinct from the x_j and y_j .

We observe that $T \langle M' \rangle \rightarrow_{!s} (\lambda x_1 \dots x_k(\lambda y_1 \dots y_l z_0)!z_1 \dots !z_k)!N'_1 \dots !N'_k \rightarrow_{!s}^2 !z_0$. We conclude as follows: since $T \langle M' \rangle \rightarrow_{!s}^2 !z_0$, $M \rightarrow_{!s}^* M'$ and T is a surface context, we have $T \langle M \rangle \rightarrow_{!s}^* !z_0$ by the contextual closure of $\rightarrow_{!s}$. Therefore M is meaningful. ◀

► **Corollary 75.** *For any M in dCBN, $\text{TNF}(M) \neq \emptyset$ if and only if M is meaningful.*

Proof. Recall that, by Corollary 58 and Theorem 53, $\text{TNF}(M) \neq \emptyset$ if and only if $\text{TNF}(M^n) \neq \emptyset$, and M is meaningful if and only if M^n is meaningful.

(\rightarrow) Assume M is meaningful, then M^n is also meaningful, and by Theorem 68, $\text{TNF}(M^n) \neq \emptyset$. It follows that $\text{TNF}(M) \neq \emptyset$.

(\leftarrow) If $\text{TNF}(M) \neq \emptyset$, then $\text{TNF}(M^n) \neq \emptyset$. By Lemma 70, $M^n \in \text{dBang}_N$, and Theorem 74 implies that M^n must be meaningful. Therefore, M is also meaningful. ◀

4.2 Meaningfulness and Taylor expansion for dCBV

We will proceed similarly to the dCBN case, although the structure of the approximants is more complex. We first characterize normal forms of dBang_V and their counterparts in resource calculus. The following definition and lemma are derived by a standard induction over the syntax of dBang_V .

► **Definition 76.** *The normal forms of dBang_V are described by the following syntax:*

$$\begin{aligned} B &:= B_! \mid !\lambda x B \mid !x \mid L\langle B \rangle \\ B_! &:= L\langle x \rangle B \mid \mathbf{der}(B_!) B \mid L\langle B_! \rangle \\ L &:= \square \mid L[B_!/x] \end{aligned}$$

► **Lemma 77.** *Let $m \triangleleft_! M \in \text{dBang}_V$. If $\mathbf{nf}(m) \neq \emptyset$ then it is of the following form:*

$$\begin{aligned} b &:= b_! \mid [\lambda x b, \dots, \lambda x b] \mid [x, \dots, x] \mid l\langle b \rangle \\ b_! &:= l\langle x \rangle b \mid \mathbf{der}(b_!) b \mid l\langle b_! \rangle \\ l &:= \square \mid l[b_!/x] \end{aligned}$$

Here we adapt the proof technique used by Carraro and Guerrieri (our Lemmas 78 and 79 correspond to Lemmas 26 and 27 in [19]) for Call-By-Value, applying it to dBang_V . We consider a family of terms which are suitable for providing an appropriate testing context for any term of dBang_V with non-empty Taylor normal form, in which it eventually reduces to a value:

$$\circ_0 = \lambda x_0 x_0^{12}$$

$$\circ_{k+1} = \lambda x_{k+1} !\circ_k$$

In what follows, the variables in \circ_i are always taken fresh, so as they do not interfere with variables in the terms where the \circ_i are substituted. In particular, we use the fact that for any $i > 1$ and any M , $\circ_k !M \rightarrow_!^2 !\circ_{k-1}$.

We establish the testing context by proving the two following lemmas through mutual induction.

► **Lemma 78.** *Let $\{x_1, \dots, x_n\}$ be a set of variables and $M \in B$ (Definition 76) with $\mathbf{fv}(M) \subseteq \{x_1, \dots, x_n\}$. There exists $c \in \mathbb{N}$ such that for any $k_1, \dots, k_n \geq c$ we have $M\{\circ_{k_1}/x_1 \dots \circ_{k_n}/x_n\} \rightarrow_!^* !P$ for some P .*

Proof. By induction on the syntax B :

- If M is of the form $!x$ or $!\lambda x B$, then we are done since $M\sigma \rightarrow_!^0 !P$ for some P and any substitution σ .
- If $M \in B_!$, we apply Lemma 79, which guarantees the existence of a substitution σ such that $M\sigma \rightarrow_!^* !\circ_j$ for some j .
- If $M = N[P_1/x_1] \dots [P_k/x_k]$, then $N \in B$ and $P_i \in B_!$ for all i . Let $\{y_1, \dots, y_l\} = \mathbf{fv}(N) \cup \bigcup_{i \leq k} \mathbf{fv}(P_k)$. By induction hypothesis, we have c such that for any $k_1, \dots, k_l \geq c$, $N\{\circ_{k_1}/y_1, \dots, \circ_{k_l}/y_l\} \rightarrow_!$

¹² The definition of \circ_0 is arbitrary, as in practice we will consider only terms \circ_k with $k > 1$ in the following proofs.

$!P$ for some P . By Lemma 79, for $i \leq k$, there exist c_i and n_i such that for any $k_{i,1}, \dots, k_{i,l} \geq c_i$, and for some $j_i \geq n_i$, $P_i\{\circ_{k_{i,1}}/y_1, \dots, \circ_{k_{i,l}}/y_l\} \rightarrow_!^* !\circ_{j_i}$.

Consider then $m_i = \max\{k_i, k_{i,1}, \dots, k_{i,l}\}$ for any $i \leq l$. We then have some P' and some r_i with $N[P_1/x_1] \dots [P_k/x_k]\{\circ_{m_1}/y_1, \dots, \circ_{m_l}/y_l\} \rightarrow_!^* !P'![\circ_{r_1}/y_1] \dots [\circ_{r_l}/y_l] \rightarrow_!^* !P'\{\circ_{r_1}/y_1\} \dots \{\circ_{r_l}/y_l\}$, which is again a value as required. \blacktriangleleft

► **Lemma 79.** Let $\{x_1, \dots, x_n\}$ be a set of variables and $M \in B_!$ (Definition 76) with $\mathbf{fv}(M) \subseteq \{x_1, \dots, x_n\}$. There exist $k, c \in \mathbb{N}$ such that for any $k_1, \dots, k_n \geq c$, there is some $j \geq k$ with $M\{\circ_{k_1}/x_1 \dots \circ_{k_n}/x_n\} \rightarrow_!^* !\circ_j$.

Proof. By induction on $B_!$:

- $M = L\langle x \rangle N = (x[N_1/y_1] \dots [N_m/y_m])N$ with $N \in B$, $N_i \in B_!$ for all $x \leq m$, and $\{x_1, \dots, x_n\} = \mathbf{fv}(M)$. By Lemma 78, there exists k such that $N\{\circ_{k_1}/x_1, \dots, \circ_{k_n}/x_n\} \rightarrow_!^* !P$ for any $k_i \geq k$ and some P .

By induction hypothesis, we have for each $i \leq m$, some l_i and c_i such that for any $l_{i,1}, \dots, l_{i,n} \geq l_i$, $N_i\{\circ_{l_{i,1}}/y_1, \dots, \circ_{l_{i,n}}/y_n\} \rightarrow_!^* !\circ_{j_i}$ for some $j_i \geq c_i$.

Let n_x be the index of x in $\{x_1, \dots, x_n\}$ (of course, $x \in \mathbf{fv}(M)$).

We then set $r_i = \max\{k_i, l_{i,1}, \dots, l_{i,n}\}$ for each $i \neq n_x$; and we consider r_{n_x} an arbitrary integer greater or equal to $\max\{k_{n_x}, l_{n_x,1}, \dots, l_{n_x,n}\}$.

We find that $M\{\circ_{r_1}/x_1, \dots, \circ_{r_m}/x_m\} \rightarrow_!^* \circ_{n_x}![\circ_{r'_1}/x_1] \dots [\circ_{r'_m}/x_1]!P'$, with $r'_i \geq r_i$ for all $i \leq m$. The reduction then yields $\circ_{n_x}!P'$, which reduces immediately to $!\circ_{r_{n_x}-1}$. This concludes the case, as the reduction holds for any $r_{n_x} \geq \max\{k_{n_x}, l_{n_x,1}, \dots, l_{n_x,n}\}$.

- $M = \mathbf{der}(N)N'$ with $N \in B_!$, $N' \in B$, and $\{x_1, \dots, x_m\} = \mathbf{fv}(M)$. By induction hypothesis, we have some n, n', c such that for any $n_1, \dots, n_m \geq n$ and $N\{\circ_{n_1}/x_1, \dots, \circ_{n_m}/x_m\} \rightarrow_!^* !\circ_j$ for all $j \geq c$, and for any $n'_1, \dots, n'_m \geq n'$, $N'\{\circ_{n'_1}/x_1, \dots, \circ_{n'_m}/x_m\} \rightarrow_!^* !P$ for some P . Then, consider $k_i = \max\{n_i, n'_i\}$ for any $i \leq m$, we have that $M\{\circ_{k_1}/x_1, \dots, \circ_{k_m}/x_m\} \rightarrow_!^* \mathbf{der}(!\circ_j)!P' \rightarrow_! \circ_j!P' \rightarrow_!^* !\circ_{j-1}$ (notice that we need here to take $j > 1$, which is allowed by our hypothesis).
- $M = N[P_1/x_1] \dots [P_k/x_k]$. This case is similar to the third case of Lemma 78: the explicit substitutions are removed after an application of the induction hypothesis. \blacktriangleleft

We can now state the central theorem of this section.

► **Theorem 80.** Let $M \in \mathbf{dBang}_V$. If $\mathbf{TNF}(M) \neq \emptyset$, then M is meaningful.

Proof. Consider some $p \in \mathbf{TNF}(M)$, assumed non-empty. There is some $m \triangleleft_! M$ such that $m \rightarrow_\delta^* p$. By Lemma 77, p belongs to the syntax b . Proposition 27 ensures that there is some $p' \in b$ such that $m \rightarrow_{\delta_s} p'$ (as in Theorem 74, we focus on internal reduction).

By iteratively applying Lemma 21, we obtain P' such that $M \rightarrow_!^* P'$ and $p' \triangleleft_! P'$. By the definition of $\triangleleft_!$, we also have that $P' \in B$.

Let $\{x_1, \dots, x_k\} = \mathbf{fv}(P')$. Lemma 78 implies that there are some terms N_1, \dots, N_k such that $P'\{N_1/x_1, \dots, N_k/x_k\} \rightarrow_!^* !Q$ for some term Q .

We define the testing context $C = (\lambda x_1 \dots \lambda x_k \square)!N_1 \dots !N_k$, which satisfies $C\langle P' \rangle \rightarrow_!^* !Q$. We can conclude that M is meaningful by the contextuality of reduction, since $C\langle M \rangle \rightarrow_!^* C\langle P' \rangle \rightarrow_!^* !Q$. \blacktriangleleft

► **Corollary 81.** Let $M \in \mathbf{dCBV}$. M is meaningful if and only if $\mathbf{TNF}(M) \neq \emptyset$.

Proof. Recall that, by Corollary 60 and Theorem 53, $\text{TNF}(M) \neq \emptyset$ if and only if $\text{TNF}(M^v) \neq \emptyset$, and M is meaningful if and only if M^v is meaningful.

(\rightarrow) Assume M is meaningful, then M^v is meaningful, and by Theorem 68, $\text{TNF}(M^v) \neq \emptyset$. It follows that $\text{TNF}(M) \neq \emptyset$.

(\leftarrow) If $\text{TNF}(M) \neq \emptyset$, then $\text{TNF}(M^v) \neq \emptyset$. By Lemma 70, $M^v \in \text{dBang}_V$, and Theorem 4.2 implies that M^v is meaningful. Therefore, M is also meaningful. \blacktriangleleft

5 Conclusion and Discussions

In this work we developed a theory of approximation in the distant bang-calculus, analogous to the ones existing in the CbN and CbV λ -calculi. To do so we define the Böhm Trees and Taylor Expansions in this setting. Furthermore, we showed how they relate to the notion of meaningfulness of dBang [30] and to each other. These results are part of a wider effort to generalize the theory of the CbN and CbV λ -calculi, and indeed, the Böhm Trees and Taylor Expansions we developed generalized those in [29, 28, 14, 25].

We already mentioned in Introduction that in Dufour and Mazza's work [20], there is no notion of meaningfulness, as their calculus Proc is very close to proof structure, and then its structure is not inductive. (in particular, Böhm trees do not have an actual tree structure since of course, approximations cannot be described through an inductive syntax). We observe some differences between our technical approaches, e.g. we consider bags and not lists (our Taylor expansion is not *rigid*); and as already discussed, our Böhm trees are not empty even in the case of terms having no surface normal forms.

For future studies, these reflections suggest that working with proof structures can provide new interesting results. Indeed, Dufour and Mazza's work showed that the inductive tree structure is not mandatory to define Böhm-like approximations and to relate it to Taylor expansion; pursuing this work with the aim of defining an analogous notion of meaningfulness in order to rely on our results with proof structures should be of interest. Also, we mentioned in Section 4 an open question about the possibility to characterize a significant fragment of dBang for which meaningful terms coincide with terms having a non-empty Taylor normal form; this line of work should be explored to develop the general understanding of dBang.

References

- 1 *Advances in Linear Logic*. London Mathematical Society Lecture Note Series. Cambridge University Press, 1995.
- 2 Beniamino Accattoli. An Abstract Factorization Theorem for Explicit Substitutions. In Ashish Tiwari, editor, *23rd International Conference on Rewriting Techniques and Applications (RTA'12)*, volume 15 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 6–21, Dagstuhl, Germany, 2012. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: <https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.RTA.2012.6>, doi:10.4230/LIPIcs.RTA.2012.6.
- 3 Beniamino Accattoli and Delia Kesner. The structural λ -calculus. In Anuj Dawar and Helmut Veith, editors, *Computer Science Logic*, pages 381–395, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
- 4 Beniamino Accattoli and Delia Kesner. Preservation of strong normalisation modulo permutations for the structural lambda-calculus. *Logical Methods in Computer Science*, Volume 8, Issue 1, March 2012. URL: [http://dx.doi.org/10.2168/LMCS-8\(1:28\)2012](http://dx.doi.org/10.2168/LMCS-8(1:28)2012), doi:10.2168/lmcs-8(1:28)2012.
- 5 Roberto Amadio and Pierre-Louis Curien. *Domains and Lambda Calculi*. 1998.
- 6 Zena Ariola. Relating graph and term rewriting via böhm models. *Applicable Algebra in Engineering, Communication and Computing*, 7, 02 1970. doi:10.1007/BF00825406.

- 7 Zena M. Ariola and Stefan Blom. Skew confluence and the lambda calculus with letrec. *Annals of Pure and Applied Logic*, 117(1):95–168, 2002. URL: <https://www.sciencedirect.com/science/article/pii/S016800720100104X>. doi:10.1016/S0168-0072(01)00104-X.
- 8 Victor Arrial, Giulio Guerrieri, and Delia Kesner. Quantitative inhabitation for different lambda calculi in a unifying framework. *Proc. ACM Program. Lang.*, 7(POPL), January 2023. doi:10.1145/3571244.
- 9 Victor Arrial, Giulio Guerrieri, and Delia Kesner. The benefits of diligence. In Christoph Benzmüller, Marijn J.H. Heule, and Renate A. Schmidt, editors, *Automated Reasoning*, pages 338–359, Cham, 2024. Springer Nature Switzerland.
- 10 Victor Arrial, Giulio Guerrieri, and Delia Kesner. The benefits of diligence. In Christoph Benzmüller, Marijn J. H. Heule, and Renate A. Schmidt, editors, *Automated Reasoning - 12th International Joint Conference, IJCAR 2024, Nancy, France, July 3-6, 2024, Proceedings, Part II*, volume 14740 of *Lecture Notes in Computer Science*, pages 338–359. Springer, 2024. doi:10.1007/978-3-031-63501-4__18.
- 11 Victor Arrial, Giulio Guerrieri, and Delia Kesner. Genericity through stratification. In *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS ’24, New York, NY, USA, 2024. Association for Computing Machinery. doi:10.1145/3661814.3662113.
- 12 Henk Barendregt. *The Lambda Calculus: Its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, Amsterdam, 1984.
- 13 Henk (Hendrik) Barendregt, Marko Eekelen, John Glauert, Richard Kennaway, Marinus Plasmeijer, and Michael Sleep. Term graph rewriting. pages 141–158, 01 1987.
- 14 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. URL: <https://www.sciencedirect.com/science/article/pii/S0049237X08711297>. doi:10.1016/S0049-237X(08)71129-7.
- 15 Gérard Boudol. Computational semantics of terms rewriting systems. Research Report RR-0192, INRIA, 1983. URL: <https://inria.hal.science/inria-00076366>.
- 16 Antonio Bucciarelli, Delia Kesner, Alejandro Ríos, and Andrés Viso. The bang calculus revisited. *Information and Computation*, 293:105047, 2023. URL: <https://www.sciencedirect.com/science/article/pii/S0890540123000500>. doi:10.1016/j.ic.2023.105047.
- 17 Albero 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 - 17th International Conference, FOSSACS 2014, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2014, Grenoble, France, April 5-13, 2014, Proceedings*, volume 8412 of *Lecture Notes in Computer Science*, pages 103–118. Springer, 2014. doi:10.1007/978-3-642-54830-7__7.
- 18 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*, pages 103–118, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.
- 19 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 - 17th International Conference, FOSSACS 2014, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2014, Grenoble, France, April 5-13, 2014, Proceedings*, volume 8412 of *Lecture Notes in Computer Science*, pages 103–118. Springer, 2014. doi:10.1007/978-3-642-54830-7__7.
- 20 Aloÿs Dufour and Damiano Mazza. Böhm and Taylor for All! In Jakob Rehof, editor, *9th International Conference on Formal Structures for Computation and Deduction (FSCD 2024)*, volume 299 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 29:1–29:20, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: <https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.FSCD.2024.29>. doi:10.4230/LIPIcs.FSCD.2024.29.
- 21 Thomas Ehrhard. Collapsing non-idempotent intersection types. In Patrick Cégielski and Arnaud Durand, editors, *Computer Science Logic (CSL’12) - 26th International Workshop/21st Annual Conference of the EACSL*, volume 16 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 259–273, Dagstuhl, Germany, 2012. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: <https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.CSL.2012.259>. doi:10.4230/LIPIcs.CSL.2012.259.

- 22 Thomas Ehrhard. Call-by-push-value from a linear logic point of view. In Peter Thiemann, editor, *Programming Languages and Systems*, pages 202–228, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.
- 23 Thomas Ehrhard and Giulio Guerrieri. The bang calculus: an untyped lambda-calculus generalizing call-by-name and call-by-value. In *Proceedings of the 18th International Symposium on Principles and Practice of Declarative Programming*, PPDP ’16, page 174–187, New York, NY, USA, 2016. Association for Computing Machinery. [doi:10.1145/2967973.2968608](https://doi.org/10.1145/2967973.2968608).
- 24 Thomas Ehrhard and Laurent Regnier. The differential lambda-calculus. *Theoretical Computer Science*, 309(1):1–41, 2003. URL: <https://www.sciencedirect.com/science/article/pii/S030439750300392X>, [doi:10.1016/S0304-3975\(03\)00392-X](https://doi.org/10.1016/S0304-3975(03)00392-X).
- 25 Thomas Ehrhard and Laurent Regnier. Uniformity and the taylor expansion of ordinary lambda-terms. *Theoretical Computer Science*, 403(2):347–372, 2008. URL: <https://www.sciencedirect.com/science/article/pii/S0304397508004064>, [doi:10.1016/j.tcs.2008.06.001](https://doi.org/10.1016/j.tcs.2008.06.001).
- 26 Thomas Ehrhard and Laurent Regnier. Uniformity and the taylor expansion of ordinary lambda-terms. *Theor. Comput. Sci.*, 403(2-3):347–372, 2008. URL: <https://doi.org/10.1016/j.tcs.2008.06.001>, [doi:10.1016/J.TCS.2008.06.001](https://doi.org/10.1016/J.TCS.2008.06.001).
- 27 Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–101, 1987. URL: <https://www.sciencedirect.com/science/article/pii/0304397587900454>, [doi:10.1016/0304-3975\(87\)90045-4](https://doi.org/10.1016/0304-3975(87)90045-4).
- 28 Axel Kerinec. *A story of lambda-calculus and approximation*. Theses, Université Paris-Nord - Paris XIII, June 2023. URL: <https://theses.hal.science/tel-04624826>.
- 29 Axel Kerinec, Giulio Manzonetto, and Michele Pagani. Revisiting call-by-value böhm trees in light of their taylor expansion. *Logical Methods in Computer Science*, Volume 16, Issue 3, Jul 2020. URL: <https://lmcs.episciences.org/4817>, [doi:10.23638/LMCS-16\(3:6\)2020](https://doi.org/10.23638/LMCS-16(3:6)2020).
- 30 Delia Kesner, Victor Arrial, and Giulio Guerrieri. Meaningfulness and Genericity in a Subsuming Framework. In Jakob Rehof, editor, *9th International Conference on Formal Structures for Computation and Deduction (FSCD 2024)*, volume 299 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 1:1–1:24, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: <https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.FSCD.2024.1>, [doi:10.4230/LIPIcs.FSCD.2024.1](https://doi.org/10.4230/LIPIcs.FSCD.2024.1).
- 31 Jim Laird, Giulio Manzonetto, Guy McCusker, and Michele Pagani. Weighted relational models of typed lambda-calculi. In *Proceedings of the 2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS ’13, page 301–310, USA, 2013. IEEE Computer Society. [doi:10.1109/LICS.2013.36](https://doi.org/10.1109/LICS.2013.36).
- 32 Paul Blain Levy. Call-by-push-value: A subsuming paradigm. In Jean-Yves Girard, editor, *Typed Lambda Calculi and Applications*, pages 228–243, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.