Taylor expansion for Call-By-Push-Value

Jules Chouquet, Christine Tasson

Taylor expansion for Call-By-Push-Value

Jules Chouquet
IRIF UMR 8243, Université Paris Diderot, Sorbonne Paris Cité, CNRS, France
Jules.Chouquet@irif.fr

Christine Tasson
IRIF UMR 8243, Université Paris Diderot, Sorbonne Paris Cité, CNRS, France
christine.tasson@irif.fr

Abstract

The connection between the Call-By-Push-Value lambda-calculus introduced by Levy and Linear Logic introduced by Girard has been widely explored through a denotational view reflecting the precise ruling of resources in this language. We take a further step in this direction and apply Taylor expansion introduced by Ehrhard and Regnier. We define a resource lambda-calculus in whose terms can be used to approximate terms of Call-By-Push-Value. We show that this approximation is coherent with reduction and with the translations of Call-By-Name and Call-By-Value strategies into Call-By-Push-Value. [1]

2012 ACM Subject Classification Theory of computation

Keywords and phrases Call-By-Push-Value; Quantitative semantics; Taylor expansion; Linear Logic

Digital Object Identifier 10.4230/LIPIcs...

Funding This work was supported by french ANR project Rapido (ANR number: ANR-14-CE25-0007).

1 Introduction

Linear Logic [15] has been introduced by Girard as a refinement of Intuitionistic Logic that take into account the use, reuse or erasing of formulas. In order to mark formulas that can be reused or erased, Girard introduced the *exponential* $!X$ and considered a *linear implication* $X \multimap Y$. Following the proof/program correspondence paradigm, Linear Logic can be used to type λ-calculus according to a chosen reduction strategy as Call-By-Name or Call-By-Value. Abstraction terms $\lambda x M$ usually typed by $X \Rightarrow Y$ will be typed as $!X \multimap Y$ when following a Call-By-Name evaluation strategy and by $!(X \multimap Y)$ when following a Call-By-Value strategy. Therefore, both evaluation strategies can be faithfully encoded in Linear Logic.

Levy followed a related goal when he introduced Call-By-Push-Value [21] : having a lambda calculus where both Call-By-Name and Call-By-Value can be taken into account. Since its introduction this calculus has been related to the Linear Logic approach [4, 12, 6, 22, 20]. We adopt this latest presentation which differentiates two kinds of types: positive and general types used for typing two kinds of terms: values and general terms respectively. The marker $!I$ is used to transform a general type I into a value type $!I$ which can be erased, used and duplicated. The idea behind ! is to stop the evaluation of the terms typed by $!I$ by placing them into thunks (*i.e.* putting them into boxes).

The purpose of this article is to push further the relations between Call-By-Push-Value and Linear Logic and to underline the resource consumption at play. For this we use syntactical Taylor expansion, that reflects Taylor expansion into semantics. Indeed, several semantics of Linear Logic and λ-calculus are interpreting types as topological vector spaces and terms

[1] The authors thank the ANR project Rapido, together with Lionel Vaux and Thomas Ehrhard for their useful advises and fertile discussions.

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

as smooth functions that enjoy Taylor expansion [5, 7, 8, 18]. Indeed, those functions can be written as power series whose coefficients are computed thanks to a derivative operator. The syntactical Taylor expansion enable the representation of terms as a combination of approximants named *resource terms*.

Taylor expansion has first been introduced by Ehrhard and Regnier while they presented the differential λ-calculus [9], they noticed that it was possible to give a syntactical version of Taylor formula, and that this object was defined on the multilinear fragment of differential λ-calculus. It consists in associating to a λ-term an infinite series of resource terms, that enjoy a linearity property, in the following sense: resource calculus is endowed with an operational semantics similar to λ-calculus, but with no duplication nor erasing of subterms during reduction. As, in analysis analytic maps are approximated by series of monomials, here λ-terms are approximated by series of resource terms. Taylor expansion gives a natural semantics, where the reduction rules of resource calculus aim to identify the terms having the same interpretation in a denotational model. In particular, the normal form of Taylor expansion (or *Taylor normal form*) is a pleasant notion of approximation of normal forms in various λ-calculi, and is strongly linked to the notion of Böhm trees, since Ehrhard and Regnier's seminal works [10]. This link has been extended in several direction, see *e.g.* Vaux [27] for algebraic λ-calculus, Kerinec, Manzonetto and Pagani [17] for Call-By-Value calculus, or Dal Lago and Leventis [19] for probabilistic λ-calculus. Let us also mention two other related approachs to approximation of λ-calculus with polyadic terms instead of resource terms [23, 24]. Taylor expansion has also been studied for the *Bang Calculus*, an untyped analogue of Call-By-Push-Value, by Guerrieri and Ehrhard [13] and then by Guerrieri and Manzonetto [16].

We propose, following that fertile discipline, a syntactical Taylor expansion for Λ_{pv}, which is the Linear Logic-oriented presentation of Call-By-Push-Value we use (and corresponds to Λ_{hp} in Ehrhard's paper [12]).

A first difficulty we have to tackle, is the fact that designing a convenient resource calculus, say Δ_{pv}, that respects Λ_{pv} dynamics is not trivial. In particular, in a redex, the argument is a value but is not necessary of exponential type. Then, the argument of a resource redex shall not be necessarily a multiset, while it is always the case in Call-By-Name and Call-By-Value resource calculi, as it ensures the reductions are linear. The semantical reason of that phenomenon is that in a quantitative model of Λ_{pv}, all values with a positive type are freely duplicable, thanks to the coalgebras morphisms associated to those types' interpretation. The solution we adopt is to give a syntactical account to those morphisms in the reduction rules, so as to Δ_{pv} stays consistent with Call-By-Push-Value operational and denotational semantics, while keeping the resource reduction linear.

We can then consider a Taylor expansion, as a function from Λ_{pv} to sets of terms in Δ_{pv}, that consists of *approximants*. Once this framework is set, we are able to show that the properties of Call-By-Push-Value, relative to the embeddings of various strategies of evaluation, can be transported at the resource level.

The principal result of the paper is the simulation of Λ_{pv} reductions in full Taylor expansion, where resource terms take coefficients in a commutative semiring. The key ingredients for this simulation to run are intrinsic to the properties of Δ_{pv}: the dynamics of reduction must reflect the reduction of Λ_{pv}, and the mechanisms of the calculus must enjoy combinatorial properties, so that the coefficients commute with the simulation. More precisely, it means that for $M, N \in \Lambda_{\mathsf{pv}}$ such that M reduces to N, if Taylor expansion of M is equal to $\sum_{i \in I} a_i m_i$, where a_i are coefficients taken in a semiring, and m_i are resource terms approximating M, then we have a notion of reduction such that $\sum_{i \in I} a_i m_i \Rightarrow \sum_{j \in I} a_j n_j$, and for each resource term n, its coefficient in the latter combination is the same as its coefficient in the Taylor expansion of N.

Contents of the paper

We first present (Section 2) Λ_{pv} as the starting point of our study, describing its operational semantics, provide examples of its expressive power, and give elements of its denotational semantics relative to coalgebras. We introduce and develop in Section 3 the resource calculus Δ_{pv} together with its operational semantics. Then, in Section 4, we define Taylor expansion for Λ_{pv}. First, in a qualitative way, with sets of approximants, where we show that it allows the simulation of Λ_{pv} reductions. We also describe how the embeddings of Call-By-Name and Call-By-Value into Call-By-Push-Value are transported at the resource level. Finally, we introduce quantitative Taylor expansion, with coefficients, and prove the commutation property between Taylor expansion and reduction that demonstrates that Taylor expansion is compatible with Λ_{pv} operational semantics.

Terminology and notations

We write $\mathbf{N}$ for the set of natural numbers, and $\mathfrak{S}_k$ for the group of permutations on $\{1, \ldots, k\}$. For a term m, and a variable x, we denote as $\mathsf{deg}_x(m)$ the number of free occurrences of x in m. These occurrences might be written $x_1, \ldots, x_{degx(m)}$, while all referring to x.

Finite multisets of elements of a set X are written $\overline{x} = [x_1, \ldots, x_k]$ for any $k \in \mathbf{N}$, and are functions from X to $\mathbf{N}$. We use the additive notation $\overline{x} + \overline{x}'$ for the multiset such that for all $y \in X, (\overline{x} + \overline{x}')(y) = \overline{x}(y) + \overline{x}'(y)$. The size of $\overline{x}$ is written $|\overline{x}|$ and is equal to $\sum_{y \in X} \overline{x}(y)$. We denote as $X^!$ the set of all finite multisets of elements of X. We might write $(x, \ldots, x)_k$ for tuples or $[x, \ldots, x]_k$ for multisets to denote k occurrences of the same element x.

If σ is a linear combination of terms $\sum_{i \in I} a_i \cdot m_i$, we use the notation $\lambda x \sigma = \sum_{i \in I} a_i \cdot \lambda x m_i$, $\mathbf{der}(\sigma) = \sum_{i \in I} a_i \cdot \mathbf{der}(m_i)$, and $\sigma^! = \sum_{k \in \mathbf{N}} \sum_{i_1, \ldots, i_k \in I} a_{i_1} \ldots a_{i_k} \cdot [m_{i_1}, \ldots, m_{i_k}]$. In the same way, if $\tau = \sum_{j \in J} a_j \cdot n_j$, we write $(\sigma, \tau) = \sum_{i \in I} \sum_{j \in J} a_i a_j \cdot (m_i, n_j)$. $\langle \sigma \rangle \tau = \sum_{i \in I} \sum_{j \in J} a_i a_j \cdot \langle m_i \rangle n_j$. This notation corresponds to the linearity of syntactic constructors with respect to potentially infinite sums of terms that will appear in Taylor expansion.

2 Call-By-Push-Value

2.1 Syntax and operational semantics

We consider a presentation of Call-By-Push-Value coming from Ehrhard [12], and convenient for its study through Linear Logic semantics.

▶ **Definition 1** (Call-By-Push-Value calculus Λ_{pv}).

$$\Lambda_{\mathsf{pv}} : M ::= x \mid \lambda x M \mid \langle M \rangle M \mid \mathbf{case}(M, y \cdot M, z \cdot M) \mid \mathbf{fix}_x(M) \mid (M, M) \mid \pi_1(M) \mid \pi_2(M) \mid M^! \mid \mathbf{der}(M) \mid \iota_1(M) \mid \iota_2(M)$$

We distinguish a subset of Λ_{pv}, *the* values *:*

$$V ::= x \mid M^! \mid (V, V) \mid \iota_1(M) \mid \iota_2(M)$$

Positive types: $A, B ::= !I \mid A \otimes B \mid A \oplus B$
General types : $I, J ::= A \mid A \multimap I \mid \top$

$$\frac{}{\Gamma, x : A \vdash x : A} \qquad \frac{\Gamma \vdash M : I}{\Gamma \vdash M^! : !I} \qquad \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x M : A \multimap B} \qquad \frac{\Gamma \vdash M : A \multimap I \qquad \Delta \vdash N : A}{\Gamma, \Delta \vdash \langle M \rangle N : I}$$

$$\frac{\Gamma \vdash M : A \qquad \Delta \vdash N : B}{\Gamma, \Delta \vdash (M, N) : A \otimes B} \qquad \frac{\Gamma \vdash M : A_1 \otimes A_2}{\Gamma \vdash \pi_i(M) : A_i}\, i \in \{1,2\}$$

$$\frac{\Gamma \vdash M : A_i}{\Gamma \vdash \iota_i(M) : A_1 \oplus A_2}\, i \in \{1,2\} \qquad \frac{\Gamma \vdash m : !A}{\Gamma \vdash \mathbf{der}(m) : A}$$

$$\frac{\Gamma \vdash M_1 : A \oplus B \qquad \Delta \vdash M_2 : I \qquad \Theta \vdash M_3 : I}{\Gamma, \Delta, \Theta \vdash \mathbf{case}(M_1, y \cdot M_2, z \cdot M_3) : I} \qquad \frac{\Gamma, x : !I \vdash M : I}{\Gamma \vdash \mathbf{fix}_x(M) : I}$$

Figure 1 Typing rules for Λ_{pv}

The typing rules are given in Figure 1 and reduction rules are given below:

$$\langle \lambda x M \rangle V \to_{\mathsf{pv}} M[V/x] \qquad \mathbf{der}(M^!) \to_{\mathsf{pv}} M$$

$$\pi_i(V_1, V_2) \to_{\mathsf{pv}} V_i \qquad \mathbf{fix}_x(M) \to_{\mathsf{pv}} M[(\mathbf{fix}_x(M))^!/x]$$

$$\mathbf{case}(\iota_i(V), x_1 \cdot M_1, x_2 \cdot M_2) \to_{\mathsf{pv}} M_i[V/x_i]$$

We define evaluation contexts E, for all terms M, N.

$$E ::= [\,] \mid \langle M \rangle E \mid \langle E \rangle M \mid \pi_i(E) \mid \iota_i(E) \mid (M, E) \mid (E, M) \mid \mathbf{case}(E, x \cdot M, y \cdot N) \mid \mathbf{der}(E)$$

and we set as an additional reduction rule $E[M] \to_{\mathsf{pv}} E[N]$ for every M, N such that $M \to_{\mathsf{pv}} N$.

2.2 An overview of denotational semantics and coalgebras

Let us give an overview of the denotational semantics of Call-By-Push-Value that justifies the introduction of the resource calculus below. This semantics is based on the semantics of Linear Logic that types the Call-By-Push-Value we are studying.

Let us describe briefly what is a model of Linear Logic (see [25] for a detailed presentation). It is given by a category $\mathcal{L}$ together with a **symmetric monoidal** structure $(\otimes, 1, \lambda, \rho, \alpha, \sigma)$ which is **closed**[2] and we write $X \multimap Y$ for the **object of linear morphisms**. It has a **cartesian** structure with cartesian product $\&$ and terminal object $\top$. The category $\mathcal{L}$ is equipped with a **comonad** $! : \mathcal{L} \to \mathcal{L}$ together with a counit $\mathbf{der}_X \in \mathcal{L}(!X, X)$ and a comultiplication $\mathbf{dig}_X \in \mathcal{L}(!X, !!X)$. This comonad comes with a symmetric monoidal structure[3] from $(\mathcal{L}, \&)$ to $(\mathcal{L}, \otimes)$, that is two natural isomorphisms $m^0 \in \mathcal{L}(1, !\top)$ and $m^2 \in \mathcal{L}(!X \otimes !Y, !(X \,\&\, Y))$.

By using isomorphisms m^0 and m^2; the functoriality of the comonad $!$ and the cartesian structure, we can build a structure of **comonoid** on any $!X$, which enable erasing and duplication of resources as we will see below.

$$\mathrm{erase}_{!X} \in \mathcal{L}(!X, 1) \qquad \mathrm{split}^2_{!X} \in \mathcal{L}(!X, !X \otimes !X)$$

A **coalgebra**[4] (P, h_P) is made of an object P and a morphism $h_P \in \mathcal{L}(P, !P)$ which

---

[2] Most model we consider are also $*$-autonomous: there is a $\bot$ such that X is isomorphic to $(X \multimap \bot) \multimap \bot$

[3] The two isomorphims m^0 and m^2 correspond to the so-called Seely isomorphisms.

[4] We want the semantics we use to interpret Call-By-Push-Value to be compatible with Taylor expansion. That is why, we have chosen to resolve the comonad using the Eilenberg-Moore resolution. The resulting category can be not well-pointed as for example the relational model described below. Another option, which is simpler and should be explored, is to use the Fam resolution [1].

is compatible with the comonad structure as $\mathbf{der}_P h_P = \mathbf{Id}$ and $\mathbf{dig}_P h_P = !h_P h_P$. Every coalgebra inherits the comonoid structure of $!P$, that is it is equipped with: $\mathrm{erase}_P \in \mathcal{L}(P, 1)$ and $\mathrm{split}^2_P \in \mathcal{L}(P, P \otimes P)$ defined as:

$$\mathrm{erase}_P : \; P \xrightarrow{h_P} !P \xrightarrow{w_P} 1 \qquad \mathrm{split}^2_P : \; P \xrightarrow{h_P} !P \xrightarrow{c_P} !P \otimes !P \xrightarrow{\mathbf{der}_P \otimes \mathbf{der}_P} P \otimes P.$$

Using similar computation, we can define $\mathrm{split}^k_P \in \mathcal{L}(P, \underbrace{P \otimes \cdots \otimes P}_{k})$.

Notice that the structure of comonad of ! induces a coalgebras structure on $!X$. Moreover, every construction of positive type preserves the coalgebra structure. To define the coalgebraic structure of $P \otimes Q$ where P and Q are both coalgebras, let us first define the morphisms $\mu^0 \in \mathcal{L}(1, !1)$ and $\mu^2 \in \mathcal{L}(!X \otimes !Y, !(X \otimes Y))$ as

$$\mu^0 : 1 \xrightarrow{m^0} !\top \xrightarrow{\mathbf{dig}_\top} !!\top \xrightarrow{!(m^0)^{-1}} !1$$

$$\mu^2 : !X \otimes !Y \xrightarrow{m^2} !(X \,\&\, Y) \xrightarrow{\mathbf{dig}_{X \& Y}} !!(X \,\&\, Y) \xrightarrow{!(m^2)^{-1}} !(!X \otimes !Y) \xrightarrow{!(\mathbf{der}_X \otimes \mathbf{der}_Y)} !(X \otimes Y).$$

Then, we can define $h_{P \otimes Q} : P \otimes Q \xrightarrow{h_P \otimes h_Q} !P \otimes !Q \xrightarrow{\mu^2} !(P \otimes Q)$. The coalgebraic structure of the coproduct is entirely defined by the morphisms for $i \in \{1, 2\}$: $P_i \xrightarrow{h_{P_i}} !P_i \xrightarrow{!\mathrm{in}_i} !(P_1 \oplus P_2)$ if the category has coproducts.

Thus, we can deduce that every positive type is interpreted as a coalgebra.

Example

The **relational** model is closely related to the Taylor expansion of the λ-calculus. Indeed, every λ-term is interpreted as the set of the interpretation of the resource terms that appear in its Taylor expansion. We can state that Taylor expansion is the syntactical counterpart of the relational model.

Let us describe some of these constructions on the **relational** model of linear logic. The category **Rel** is made of sets and relations. The tensor product is given by the set cartesian product and its unit is the singleton set whose unique element is denoted $*$. The product is given by disjoint union and the terminal object is the emptyset. **Rel** can be equipped with the comonad of finite multisets. The comonadic structure of $!X$ is

$$\mathbf{der}_X = \{([a], a) | a \in X\} \qquad \mathbf{dig}_X = \{(\overline{m}, [\overline{m}_1, \dots, \overline{m}_k]) | \overline{m}_1 + \cdots + \overline{m}_k = \overline{m}\}.$$

The comonoidal structure of $!X$ is

$$\mathrm{erase}_{!X} = \{([], *)\} \qquad \mathrm{split}^2_{!X} = \{(\overline{m}, (\overline{m}_1, \overline{m}_2)) | \overline{m}_1 + \overline{m}_2 = \overline{m}\}.$$

A positive type is a finite combination of $\oplus, \otimes, !$. For instance if $P = (!X_1 \oplus !X_2) \otimes (!Y \otimes !Z)$, then P is a coalgebra (see Figure 2):

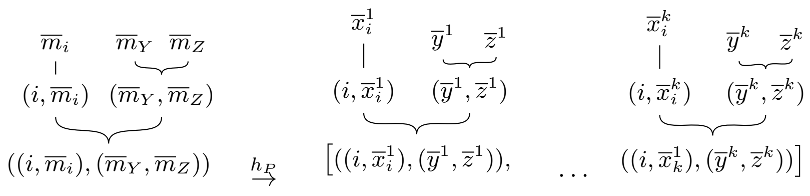
$$h_P = \{(((i, \overline{m}_i)) \;,\; (\overline{m}_Y, \overline{m}_Z)), [((i, \overline{x}^1_i), (\overline{y}^1, \overline{z}^1)), \dots, ((i, \overline{x}^k_i), (\overline{y}^k, \overline{z}^k))] | \\ \overline{m}_i = \overline{x}^1_i + \cdots + \overline{x}^k_i, \overline{m}_Y = \overline{y}_1 + \cdots + \overline{y}_k, \overline{m}_Z = \overline{z}_1 + \cdots + \overline{z}_k\},$$

and is equipped with the comonoidal structure:

$$\mathrm{erase}_P = \{(((i, []), ([], []), *)\}$$

$$\mathrm{split}^2_P = \{((i, \overline{m}_i), (\overline{m}_Y, \overline{m}_Z)) \;,\; ((i, (\overline{m}^1_i + \overline{m}^2_i)), ((\overline{m}^1_Y + \overline{m}^2_Y), (\overline{m}^1_Z + \overline{m}^2_Z))) | \\ \overline{m}^1_i + \overline{m}^2_i = \overline{m}_i, \overline{m}^1_Y + \overline{m}^2_Y = \overline{m}_Y, m^1_Z + \overline{m}^2_Z = \overline{m}_Z\}\}.$$

Remark that the structural morphisms are the same as those of $!X$ but at the leaves of the tree structure describing the formula P.



where $\sum_{j=1}^{k} \overline{x}_i^j = \overline{m}_i$, $\sum_{j=1}^{k} \overline{y}^j = \overline{m}_Y$, and $\sum_{j=1}^{k} \overline{z}^j = \overline{m}_Z$.

Figure 2 Action of the coalgebra morphism h_P on a positive type

3 Resource calculus for Call-By-Push-Value

We introduce a typed resource calculus, able to simulate the operational semantics of Λ_{pv}. The conditional construction is considered through tests of equality, and there is no explicit fixpoint. The main difference with other resource calculi, like Call-By-Name or Call-By-Value, is that redexes of shape $\langle \lambda x m\rangle \overline{n}$ are not enough to entail Λ_{pv} reduction. Indeed, the notion of value is too wide to be entirely captured in multisets of approximants: $\langle \lambda x M\rangle(V_1, V_2)$ is a redex in Λ_{pv}, then we must be able to reduce terms like $\langle \lambda x m\rangle(v_1, v_2)$ in the resource setting, while keeping it sensitive to resource consumption. We proceed so with the introduction of a splitting operator, which allows us to duplicate a value using the structure of its positive type.

▶ **Definition 2** (Call-By-Push-Value resource calculus Δ_{pv})**.** *The syntax of types is the same as the syntax of Λ_{pv}.*

$$\begin{aligned} \Delta_{\mathsf{pv}} : m ::= {} & x \mid 1 \mid 2 \mid \lambda x m \mid \langle m\rangle m \mid (m = m)\cdot m \mid (m, m) \mid \pi_1(m) \mid \pi_2(m) \\ & \mid [m, \dots, m] \mid \mathbf{der}(m) \end{aligned}$$

We distinguish the values of the calculus:

$$v ::= x \mid 1 \mid 2 \mid [m, \dots, m] \mid (v, v)$$

$$\frac{}{\Gamma, x : A \vdash x : A} \qquad \frac{\Gamma \vdash m_i : I,\ i \in \{1, \dots, k\}}{\Gamma \vdash [m_1, \dots, m_k] : !I} \qquad \frac{\Gamma, x : A \vdash m : B}{\Gamma \vdash \lambda x m : A \multimap B}$$

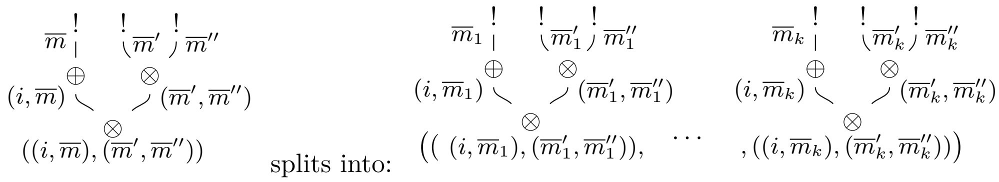
$$\frac{\Gamma \vdash m : A \multimap I \qquad \Delta \vdash n : A}{\Gamma, \Delta \vdash \langle m\rangle n : I} \qquad \frac{\Gamma \vdash m : !A}{\Gamma \vdash \mathbf{der}(m) : A}$$

$$\frac{\Gamma \vdash m : A \qquad \Delta \vdash n : B}{\Gamma, \Delta \vdash (m, n) : A \otimes B} \qquad \frac{\Gamma \vdash m : A_1 \otimes A_2}{\Gamma \vdash \pi_i(m) : A_i}\ i \in \{1, 2\} \qquad \frac{\Gamma \vdash m : A_i}{\Gamma \vdash (i, m) : A_1 \oplus A_2}\ i \in \{1, 2\}$$

$$\frac{\Gamma \vdash m_1 : A_1 \oplus A_2 \qquad \Delta \vdash m_2 : A_i \qquad \Theta \vdash m_3 : I}{\Gamma, \Delta, \Theta \vdash (m_1 = (i, m_2)) \cdot m_3 : I}$$

Figure 3 Typing rules for Δ_{pv}

In order to set the operational semantics of the resource calculus just defined, we introduce a new construction $\mathbf{split}^k$. Its operational semantics is the duplication of ground values such as integers or variables and the split of the leaves of tree structure induced by pairs and injections, as exemplified in Figure 4. This **splitting operator** is the syntactical counterpart



where $\sum_{i=1}^{k} \overline{m}_i = \overline{m}$, $\sum_{i=1}^{k} \overline{m}'_i = \overline{m}'$, and $\sum_{i=1}^{k} \overline{m}''_i = \overline{m}''$.

Figure 4 Splitting a value, the tree of its positive type labelled by resource components.

of the semantical morphism associated to each coalgebra P interpreting a positive type: $\mathrm{split}^k_P \in \mathcal{L}(P, \underbrace{P \otimes \cdots \otimes P}_{k})$ (see Section 2.2).

▶ **Definition 3** (Split). $\mathbf{split}^k(m)$ *is defined as a set of* k*-tuples of values of same shape than* m*. It is defined when* m *is a value itself.*

- $\mathbf{split}^k(\overline{m}) = \{(\overline{m}_1, \dots, \overline{m}_k) \mid \sum_{i=1}^{k} \overline{m}_i = \overline{m}\}$
- $\mathbf{split}^k(x) = \{(x, \dots, x)_k\}$
- $\mathbf{split}^k(i) = \{(i, \dots, i)_k\}$ *for* $i \in \{1, 2\}$.
- $\mathbf{split}^k((m, n)) = \{((m_1, n_1), \dots, (m_k, n_k)) \mid (m_1, \dots, m_k) \in \mathbf{split}^k(m), (n_1, \dots, n_k) \in \mathbf{split}^k(n)\}$.

We define now the reduction rules associated to Δ_{pv}, by adding the distinguished term 0 to the calculus.

- $\langle \lambda x m \rangle n \to_{\mathsf{rpv}} m[n_1/x_1, \dots, n_k/x_k]$ for $\mathsf{deg}_x(m) = k$ and all $(n_1, \dots, n'_k) \in \mathbf{split}^k(n)$.
- $(v = (i, v')) \cdot n \to_{\mathsf{rpv}} n$ if $v = (i, v')$. $(v = (i, v')) \cdot n \to_{\mathsf{rpv}} 0$ otherwise.
- $\mathbf{der}([m_1, \dots, m_k]) \to_{\mathsf{rpv}} m_1$ if $k = 1$, and $\mathbf{der}([m_1, \dots, m_k]) \to_{\mathsf{rpv}} 0$ otherwise.
- $\pi_i((m_1, m_2)) \to_{\mathsf{rpv}} m_i$

We define evaluation contexts e, for all terms t, u of Δ_{pv} :

$$e ::= [] \mid \langle e \rangle m \mid \langle m \rangle e \mid \lambda x e \mid (e, m) \mid (m, e) \mid (e = m) \cdot n \mid (m = e) \cdot n \mid \mathbf{der}(e)$$

and set the additional rule $e[m] \to_{\mathsf{rpv}} e[n]$ if $m \to_{\mathsf{rpv}} n$ by one of the above rules, with $e[0] = 0$ for all context e.

We cannot define a reduction for tests of equality that produces non values-terms, because we would lost confluence: for example, if we allow to reduce $m(\pi_1(m_1, m_2) = m_1) \cdot n$, then m reduces to 0, and it reduces as well to $(m_1 = m_1) \cdot n$, which reduces to n.

▶ **Proposition 4** (Subject Reduction). *For any terms* m, n *and general type* I*, if* $m : I$ *and* $m \to_{\mathsf{rpv}} n$*, then* $n : I$*.*

Proof. By induction on m.

- If $m = (\pi_i(m_1, m_2))$ and if $n = m_i$, then there exist A_1, A_2 such that $m_i : A_i$, and we have $m : A_i$ and $n : A_i$.
- If $m = \mathbf{der}([n])$, then there is a type J such that $n : J$, and we have $[n] : !J$ and $m : J$.
- If $m = (v_1 = (i, v_2)) \cdot n$, then if $n : J$ for some type J, then $m : J$.
- If $m = \langle \lambda x m' \rangle v$ and $n = m'[v_1/x_1, \dots, v_k/x_k]$ for $k = \mathsf{deg}_x(m')$ and $(v_1, \dots, v_k) \in \mathbf{split}^k(v)$, then $x : A, v : A, m' : J, \lambda x m' : A \multimap J$, for some types A, J. Then $m : J$, in order to conclude $n : J$, it remains to ensure that for all $i \in \{1, \dots, k\}$, $v_i : A$ which is

done easily by an induction on v, and that it implies $m'[v_1/x_1, \ldots, v_k/x_k] : A$. That last point follows from a standard argument.

- If $m = e[m']$ and $n = e[n']$ for $n \to_{\mathsf{rpv}} n'$, we conclude by induction hypothesis. ◀

We define for all $k \in \mathbf{N}$, all variable x and $m \in \Delta_{\mathsf{pv}}$, a set of terms $\mathbf{fix}_x^k(m)$ as follows, with $\mathbf{fix}_x^0(m) = \{m[[]/x_1, \ldots, []/x_{\mathsf{deg}_x(m)}]\}$:

$$\mathbf{fix}_x^{k+1}(m) = \left\{m\left[\overline{m}_1/x_1, \ldots, \overline{m}_{\mathsf{deg}_x(m)}/x_{\mathsf{deg}_x(m)}\right] \mid \forall i \leq \mathsf{deg}_x(m) : \overline{m}_i \in (\mathbf{fix}_x^k(m))^!\right\}.$$

4 Taylor expansion

Taylor expansion consists in taking infinitely many approximants of a given object. As analytic maps can be understood as infinite series of polynomials that approximate it, Λ_{pv} terms can be considered through all resource terms that are also multilinear (in the computational sense) approximants. We first introduce a qualitative version, with sets, through which we show a first simulation property (Proposition 9), and we prove that the embeddings of Call-By-Name and Call-By-Value behave well at the resource level (Property 2). Then, we introduce coefficients so as to consider full quantitative Taylor expansion. Lemma 10 ensures that it does not lead to divergence issues through a finiteness property of antireduction. Finally, we prove the full simulation of Λ_{pv} reduction in Taylor expansion, showing that coefficients commute with reduction, in Theorem 17.

4.1 Definition and Simulation

▶ **Definition 5** (Support of Taylor expansion)**.** *We define the sets of resource terms corresponding to the support of Taylor expansion of Λ_{pv}:*

$$\begin{array}{ll}
\mathcal{T}_{\mathsf{pv}}(x) = \{x\} & \mathcal{T}_{\mathsf{pv}}\langle M\rangle N = \{\langle m\rangle n \mid m \in \mathcal{T}_{\mathsf{pv}}(M), n \in \mathcal{T}_{\mathsf{pv}}(N)\} \\
\mathcal{T}_{\mathsf{pv}}(\iota_i(M)) = \{(i, m) \mid m \in \mathcal{T}_{\mathsf{pv}}(M)\} & \mathcal{T}_{\mathsf{pv}}(\mathbf{der}(M)) = \{\mathbf{der}(m) \mid m \in \mathcal{T}_{\mathsf{pv}}(M)\} \\
\mathcal{T}_{\mathsf{pv}}(M^!) = \mathcal{T}_{\mathsf{pv}}(M)^! & \mathcal{T}_{\mathsf{pv}}((M, N)) = \{(m, n) \mid m \in \mathcal{T}_{\mathsf{pv}}(M), n \in \mathcal{T}_{\mathsf{pv}}(N)\} \\
\mathcal{T}_{\mathsf{pv}}(\pi_i(M)) = \{\pi_i(m) \mid m \in \mathcal{T}_{\mathsf{pv}}(M)\} & \mathcal{T}_{\mathsf{pv}}(\mathbf{fix}_x(M)) = \{\mathbf{fix}_x^k(m) \mid m \in \mathcal{T}_{\mathsf{pv}}(M), k \in \mathbf{N}\} \\
\mathcal{T}_{\mathsf{pv}}(\lambda x M) = \{\lambda x m \mid m \in \mathcal{T}_{\mathsf{pv}}(M)\} & \mathcal{T}_{\mathsf{pv}}(\mathbf{case}(M, z_1 {\cdot} N_1, z_2 {\cdot} N_2)) = \{(m = (i, m')) {\cdot} n_i[m'/z_i] \\
 & \quad \mid i \in \{1, 2\}, m \in \mathcal{T}_{\mathsf{pv}}(M), n_i \in \mathcal{T}_{\mathsf{pv}}(N_i), m' \in \ \Delta_{\mathsf{pv}}\}
\end{array}$$

▷ Property 1. Let $M \in \Lambda_{\mathsf{pv}}$, $m \in \mathcal{T}_{\mathsf{pv}}(M)$, and $k \in \mathbf{N}$. $\mathbf{split}^k(m)$ is defined if and only if M is a value.

Proof. One can check that the syntax of resource terms v that are in $\mathcal{T}_{\mathsf{pv}}(V)$ for a value V matchs exactly the resource values of Definition 2. It is easy to verify that $\mathbf{split}^k(v)$ is always defined, and that if $m \in \mathcal{T}_{\mathsf{pv}}(M)$ is not such a resource value, then $\mathbf{split}^k(m)$ is not defined. ◀

The following corollary shows that Δ_{pv} is consistent with Λ_{pv} in the following sense: an approximant of a redex in Λ_{pv} is always a redex in Δ_{pv}, and a redex in Δ_{pv} which is an approximant of a term in Λ_{pv}, is the approximation of a redex. This is mostly trivial, but for redexes of shape $\langle\lambda x m\rangle n$ (respectively $\langle\lambda x M\rangle N$), where it is a consequence of Property 1, as stated in the following corollary:

▶ **Corollary 6.** *Let $\langle\lambda x m\rangle n \in \mathcal{T}_{\mathsf{pv}}((\lambda x M)N)$. There is a term m' such that $\langle\lambda x m\rangle n \to_{\mathsf{rpv}} m'$ by reducing the most external redex if and only if N is a value. Recall moreover that $(\lambda x M)N \to_{\mathsf{pv}} M[N/x]$ if and only if N is a value.*

▶ **Lemma 7.** *If M is a value, $k \in \mathbf{N}$, $m \in \mathcal{T}_{\mathsf{pv}}(M)$ and $(m_1, \ldots, m_k) \in \mathbf{split}^k(m)$ then for all $i \in \{1, \ldots, k\}$, $m_i \in \mathcal{T}_{\mathsf{pv}}(M)$.*

Proof. By induction on M, using Property 1 :

- If $M = x$, then $m = x$ and $\mathbf{split}^k(m) = (x, \ldots, x)_k$. We conclude since $\mathcal{T}_{\mathsf{pv}}(x) = \{x\}$.
- If $M = N^!$, then $m = [n_1, \ldots, n_l]$, and for all $i \in \{1, \ldots, l\}$, $n_i \in \mathcal{T}_{\mathsf{pv}}(N)$. We have $(m_1, \ldots, m_k) = (\overline{n}_1, \ldots, \overline{n}_k)$ with $\sum_{i=1}^k \overline{n}_i = [n_1, \ldots, n_l]$. Then, each $\overline{n}_i$ is a multiset of elements in $\mathcal{T}_{\mathsf{pv}}(N)$, and $\overline{n}_i \in \mathcal{T}_{\mathsf{pv}}(N^!) = \mathcal{T}_{\mathsf{pv}}(M)$.
- If $M = (N, N')$, then $m = (n, n')$ for $n \in \mathcal{T}_{\mathsf{pv}}(N)$ and $n' \in \mathcal{T}_{\mathsf{pv}}(N')$. $(m_1, \ldots, m_k) = ((n_1, n'_1), \ldots, (n_k, n'_k))$ with $(n_1, \ldots, n_k) \in \mathbf{split}^k(N)$ and $(n'_1, \ldots, n'_k) \in \mathbf{split}^k(N')$. By induction hypothesis, for all $i \in \{1, \ldots, k\}$, $n_i \in \mathcal{T}_{\mathsf{pv}}(N)$ and $n'_i \in \mathcal{T}_{\mathsf{pv}}(N')$. Then for all i, $(n_i, n'_i) \in \mathcal{T}_{\mathsf{pv}}(N, N') = \mathcal{T}_{\mathsf{pv}}(M)$.
- If $M = \iota_j(N)$, then $m = (j, n)$ for $n \in \mathcal{T}_{\mathsf{pv}}(N)$ and $\mathbf{split}^k(m) = ((j, n_1), \ldots, (j, n_k))$ with $(n_1, \ldots, n_k) \in \mathbf{split}^k(n)$. By induction hypothesis, for all $i \in \{1, \ldots, k\}$, $n_i \in \mathcal{T}_{\mathsf{pv}}(N)$. Then for all i, $(j, n_i) \in \mathcal{T}_{\mathsf{pv}}(\iota_j(N)) = \mathcal{T}_{\mathsf{pv}}(M)$.

◀

The following substitution lemma is crucial to ensure that Taylor expansion is compatible with reduction. It will be used for proving simulation, in Proposition 9.

▶ **Lemma 8** (Substitution). *Let $m \in \mathcal{T}_{\mathsf{pv}}(M)$, $k = \mathsf{deg}_x(m)$, and $n_1, \ldots, n_k \in \mathcal{T}_{\mathsf{pv}}(N)$, for $M, N \in \Lambda_{\mathsf{pv}}$. We have $m[n_1/x_1, \ldots, n_k/x_k] \in \mathcal{T}_{\mathsf{pv}}(M[N/x])$.*

Proof. The proof is by induction on M. We only consider representative cases, the other following by similar applications of induction hypothesis.

- If $M = x$, then $m = x, k = 1$, $m[n_1/x_1] = n_1$, and $M[N/x] = N$. Then $m[n_1/x_1] \in \mathcal{T}_{\mathsf{pv}}(M[N/x])$.
- If $M = \lambda y M'$, then $\mathsf{deg}_x(M) = \mathsf{deg}_x(M'), m = \lambda y m'$ for $m' \in \mathcal{T}_{\mathsf{pv}}(M')$. By induction hypothesis, $m'[n_1/x_1, \ldots, n_k/x_k] \in \mathcal{T}_{\mathsf{pv}}(M'[N/x])$. Since $m[n_1/x_1, \ldots, n_k/x_k] = \lambda y m'[n_1/x_1, \ldots, n_k/x_k]$, we conclude.
- If $M = \langle M_1 \rangle M_2$, then $m = \langle m_1 \rangle m_2$ for $m_i \in \mathcal{T}_{\mathsf{pv}}(M_i)$, and $\mathsf{deg}_x(m) = l_1 + l_2$ for $l_1 = \mathsf{deg}_x(m_1)$ and $l_2 = \mathsf{deg}_x(m_2)$. By induction hypothesis, $m_1[n_1/x_1, \ldots, n_{l_1}/x_{l_1}] \in \mathcal{T}_{\mathsf{pv}}(M_1[N/x])$ and $m_2[n_{l_1+1}/x, \ldots, n_{l_1+l_2}/x] \in \mathcal{T}_{\mathsf{pv}}(M_2[N/x])$. Since $m[n_1/x_1, \ldots, n_k/x_k] = \langle m_1[n_1/x_1, \ldots, n_{l_1}/x_{l_1}] \rangle m_2[n_{l_1+1}/x, \ldots, n_{l_1+l_2}/x]$, and $M[N/x] = \langle M_1[N/x] \rangle M_2[N/x]$, we conclude.
- If $M = M'^!$, then $m = [m'_1, \ldots, m'_l]$ with $m'_i \in \mathcal{T}_{\mathsf{pv}}(M')$ for all i, and $\mathsf{deg}_x(m) = \sum_{i=1}^l k_i$ where $k_i = \mathsf{deg}_x(m'_i)$. By induction hypothesis, $m'_i[n_{k_{i-1}+1}/x_{k_{i-1}+1}, \ldots, n_{k_{i-1}+k_i}/x_{k_{i-1}+k_i}] \in \mathcal{T}_{\mathsf{pv}}(M'[N/x])$ for all $i \in \{1, \ldots, l\}$ (setting $k_0 = 0$). Then, $M[N/x] = (M'[N/x])^!$, and we can conclude as before.
- In $M = \mathbf{case}(M', z_1 \cdot N_1, z_2 \cdot N_2)$, then $m = (m' = (i, m'')) \cdot n_i[m''/z_i]$ for $i \in \{1, 2\}, m' \in \mathcal{T}_{\mathsf{pv}}(M'), n_i \in \mathcal{T}_{\mathsf{pv}}(N_i), m'' \in \Delta_{\mathsf{pv}}$. We conclude by induction hypothesis as above.

◀

Notice that only the case where N is a value will be used, since the other cases do not appear in the operational semantics.

We can finally prove the first simulation property:

▶ **Proposition 9** (Simulation). *If $M \to_{\mathsf{pv}} M'$, then for any $m \in \mathcal{T}_{\mathsf{pv}}(M)$, either $m \to_{\mathsf{rpv}} 0$ or there is $m' \in \mathcal{T}_{\mathsf{pv}}(M')$ such that $m \to_{\mathsf{rpv}}^{=} m'$, where $\to_{\mathsf{rpv}}^{=}$ is the reflexive closure of $\to_{\mathsf{rpv}}$.*

Proof. By induction on M :

- If $M = \pi_i((M_1, M_2))$ and $M' = M_i$, then $m = \pi_i((m_1, m_2))$ for $m_i \in \mathcal{T}_{\mathsf{pv}}(M_i)$. We conclude since $M \to_{\mathsf{pv}} M_i$ and $m \to_{\mathsf{rpv}} m_i$.
- If $M = \mathbf{der}(N^!)$ and $M' = N$, then $m = \mathbf{der}([n_1, \ldots n_k])$, with $n_i \in \mathcal{T}_{\mathsf{pv}}(N)$ for all $i \in \{1, \ldots, k\}$. We conclude since $M \to_{\mathsf{pv}} N$ and $m \to_{\mathsf{rpv}} n_1$ if $k = 1$ and $m \to_{\mathsf{rpv}} 0$ otherwise.
- If $M = \mathbf{fix}_x(N)$ and $M' = N[(\mathbf{fix}_x(N))^!/x]$, then it is easy to verify that $\mathcal{T}_{\mathsf{pv}}(M) = \mathcal{T}_{\mathsf{pv}}(M')$, using Lemma 8 and unfolding the definition of Taylor expansion of fixpoint. We need a reflexive reduction for this case.
- If $M = (\lambda y N)V$ and $M' = N[V/y]$, then $m = \langle \lambda y n \rangle v$ for $n \in \mathcal{T}_{\mathsf{pv}}(N)$ and $v \in \mathcal{T}_{\mathsf{pv}}(V)$. By Property 1, $\mathbf{split}^k(v)$ is defined for any $k \in \mathbf{N}$, then $m \to_{\mathsf{rpv}} n[v_1/y_{f(1)}, \ldots, v_k/y_{f(k)}]$ for $\mathsf{deg}_y(n) = k$ and $(v_1, \ldots, v_k) \in \mathbf{split}^k(v)$. By Lemma 7, for all $i \in \{1, \ldots, k\}$, $v_i \in \mathcal{T}_{\mathsf{pv}}(V)$, and by the substitution Lemma 8, $n[v_1/y_1, \ldots, v_k/y_k] \in \mathcal{T}_{\mathsf{pv}}(N[V/y])$.
- If $M = \mathbf{case}(\iota_i(V), x_1 \cdot M_1, x_2 \cdot M_2)$ and $M' = M_i[V/x_i]$, then, $m = ((i, v) = (j, n)) \cdot m_i[v/x_i]$ for $i, j \in \{1, 2\}, v \in \mathcal{T}_{\mathsf{pv}}(V), n \in \Delta_{\mathsf{pv}}, m_i \in \mathcal{T}_{\mathsf{pv}}(M_i)$. Either $m \to_{\mathsf{rpv}} 0$, either $(i, v) = (j, n)$ and in this case $m \to_{\mathsf{rpv}} m_i[n/x_i] = m_i[v/x_i]$. By the substitution Lemma 8 we conclude, since we have $M \to_{\mathsf{pv}} M_i[V/x_i]$ and $m_i[v/x_i] \in \mathcal{T}_{\mathsf{pv}}(M_i[V/x_i])$.
- If $M = E[N]$ and $M' = E[N']$, then we can easily show that there is a resource context e such that $m = e[n]$ and $n \in \mathcal{T}_{\mathsf{pv}}(N)$. By induction hypothesis, either $n \to_{\mathsf{rpv}} 0$, and then $e[n] = 0$, or there exists n' such that $n \to_{\mathsf{rpv}} n'$ and $n' \in \mathcal{T}_{\mathsf{pv}}(N')$. We can easily adapt the substitution Lemma to conclude $e[n'] \in \mathcal{T}_{\mathsf{pv}}(E[N'])$.

◀

4.2 Embeddings of CBV and CBN

Call-By-Push-Value is known to subsume both Call-By-Name and Call-By-Value strategies. In particular, the two strategies can be embedded into Λ_{pv}. If we consider simply typed λ-calculus[5] Λ, we set two functions $()^v, ()^n : \Lambda \to \Lambda_{\mathsf{pv}}$, defined in Figure 7. We do not consider here calculi with products, or other constructors, in order to focus in a simple setting on the relation between exponentials and strategies of reduction (see Ehrhard and Tasson's work [14] for more developments). Our embeddings ensure *e.g.* the following property: $((\lambda x M)N)^v \to_{\mathsf{pv}} (M[N/x])^v$ if and only if N is a variable or an abstraction, and $((\lambda x M)N)^n \to_{\mathsf{pv}} (M[N/x])^n$ for any M, N.

From the Taylor expansion point of view, let $\mathcal{T}^n$ and $\mathcal{T}^v$ be, respectively, usual Call-By-Name expansion, and Call-By-Value expansion (first defined by Ehrhard [11]). We can check the correctness of our construction of Δ_{pv} and $\mathcal{T}_{\mathsf{pv}}$ with respect to those embeddings, using $\mathcal{T}^n$ and $\mathcal{T}^v$ defined in Figure 6. The first one is defined on Δ^n, which is the original Ehrhard and Regnier's resource calculus [9], and the second one on Δ^v, a Call-By-Value resource calculus, introduced by Ehrhard [11]. Both are described in Figure 5.

▷ Property 2. For any pure λ-term $M \in \Lambda$, $E(\mathcal{T}_{\mathsf{pv}}((M)^v)) = \mathcal{T}^v(M)$ and $E(\mathcal{T}_{\mathsf{pv}}((M)^n)) = \mathcal{T}^n(M)$, where E is the function that erases all the derelictions (that do not exist in Δ^n nor in Δ^v) in a set of terms.

Proof. The proof consists in a simple examination of the definitions. Let us start with Call-By-Value constructions: The variable case is immediate since $\mathcal{T}_{\mathsf{pv}}(x^v) = \{\mathbf{der}(x)\}^!$, and

[5] We do not make types explicit, since the translation works in the same way with pure λ-calculus (*e.g* when translated in Linear Logic proof nets). But since the target calculus is typed, this restriction is necessary

Δ^n	Δ^v
$m, n ::= x \mid \lambda x m \mid \langle m \rangle \overline{n}$	$m, n ::= [x_1, \ldots, x_k] \mid [\lambda x m_1, \ldots, \lambda x m_k] \mid \langle m \rangle n$
$\langle \lambda x m \rangle [n_1, \ldots, n_k] \to m[n_1/x_{f(1)}, \ldots, n_k/x_{f(k)}]$ if $k = \mathsf{deg}_x(m)$ and $f \in \mathfrak{S}_k$	$\langle [\lambda x m] \rangle [n_1, \ldots, n_k] \to m[n_1/x_{f(1)}, \ldots, n_k/x_{f(k)}]$ if $k = \mathsf{deg}_x(m)$ and $f \in \mathfrak{S}_k$

Figure 5 Call-By-Name and Call-By-Value resource calculi

Call-By-Name Taylor expansion	Call-By-Value Taylor expansion
$\mathcal{T}^n(x) = \{x\}$	$\mathcal{T}^v(x) = \{x\}^!$
$\mathcal{T}^n(MN) = \{\langle m \rangle \overline{n} \mid m \in \mathcal{T}^n(M), \overline{n} \in \mathcal{T}^n(N)^!\}$	$\mathcal{T}^v(MN) = \{\langle m \rangle n \mid m \in \mathcal{T}^v(M), n \in \mathcal{T}^v(N)\}$
$\mathcal{T}^n(\lambda x M) = \{\lambda x m \mid m \in \mathcal{T}^n(M)\}$	$\mathcal{T}^v(\lambda x M) = \{[\lambda x m_1, \ldots, \lambda x m_k] \mid m_i \in \mathcal{T}^v(M)\}$

Figure 6 $\mathcal{T}^v : \Lambda \to P(\Delta^v)$ and $\mathcal{T}^n : \Lambda \to P(\Delta^n)$

$\mathcal{T}^v(x) = \{x\}^!$. $\mathcal{T}_{\mathsf{pv}}((\lambda x M)^v) = \{[\lambda x m_1, \ldots, \lambda x m_k] \mid k \in \mathbf{N}, m_i \in \mathcal{T}_{\mathsf{pv}}(M^v)\}$, we conclude since by induction hypothesis, $E(\mathcal{T}_{\mathsf{pv}}(M^v)) = \mathcal{T}^v(M)$ and $\mathcal{T}^v(\lambda x M) = \{[\lambda x m'_1, \ldots, \lambda x m'_l] \mid l \in \mathbf{N}, m'_i \in \mathcal{T}^v(M)\}$. The application case is managed with a similar argument with induction hypothesis, and with the fact that $E(\langle \mathbf{der}(M) \rangle N) = \langle E(M) \rangle E(N)$.

For Call-By-Name, we only consider the application case (the other being straightforward): $\mathcal{T}_{\mathsf{pv}}((MN)^n) = \{\langle m \rangle \overline{n} \mid m \in \mathcal{T}_{\mathsf{pv}}(M^n), \overline{n} \in \mathcal{T}_{\mathsf{pv}}(N^n)^!\}$. By induction hypothesis, $E(\mathcal{T}_{\mathsf{pv}}(M^n)) = \mathcal{T}^n(M)$ and $E(\mathcal{T}_{\mathsf{pv}}(N^n)) = \mathcal{T}^n(N)$, and we can conclude. ◀

Together with the simulation property of $\mathcal{T}_{\mathsf{pv}}$ (Property 9), Property 2 proves that Call-By-Push-Value subsumes both Call-By-Name and Call-By-Value strategies, and that remains valid at a resource level.

4.3 Finiteness

The following lemma ensures that one can consider a quantitative version of Taylor expansion $\mathcal{T}_{\mathsf{pv}}$, and extend the resource reduction to an infinite and weighted setting. The conditions of validity of this result have been widely studied in non uniform settings, Linear-Logic proof nets, or various strategies of reduction [2, 3, 26, 27]. This is necessary for proving Lemma 15 that state that coefficients remain finite under reduction.

▶ **Lemma 10** (Finiteness of antireduction). *Let $n \in \Delta_{\mathsf{pv}}$ and M in Λ_{pv}. $\{m \in \mathcal{T}_{\mathsf{pv}}(M) \mid m \to_{\mathsf{rpv}}^{=} n\}$ is finite.*

(sketch). We do not detail the proof, since we can adapt the first author's work [2] for PCF. The idea is to extend Ehrhard and Regnier's original proof [10], defining a coherence relation on resource terms in a way $\mathcal{T}_{\mathsf{pv}}(M)$ is always a maximal clique for this relation. In particular, $\bigcup_{k \in \mathbf{N}} \mathbf{fix}_x^k(m)$ must be a clique.

Then, it remains to show that the reduction preserves coherence, and that if m, m' are coherent, and both reduce to n, then $m = m'$. We conclude that there cannot be several distinct resource terms in $\mathcal{T}_{\mathsf{pv}}(M)$ reducing to a common term. ◀

4.4 Taylor expansion with coefficients

In the remainder of this section, we will consider infinite linear combinations of resource terms. Those terms will take coefficients in an arbitrary commutative semiring $\mathbf{S}$ with fractions: a

Call-By-Name translation	Call-By-Value translation
$(x)^n = \mathbf{der}(x)$ $(MN)^n = \langle M^n \rangle (N^n)^!$ $(\lambda x M)^n = \lambda x M^n$	$(x)^v = \mathbf{der}(x)^!$ $(MN)^v = \langle \mathbf{der}(M) \rangle N$ $(\lambda x M)^v = (\lambda x M^v)^!$

Figure 7 Both translations are functions from Λ to Λ_{pv}.

semiring in which every natural number $k \neq 0 \in \mathbf{N}$ admits a multiplicative inverse, written $\frac{1}{k}$. For a combination $\varphi = \sum_{i \in I} a_i \cdot m_i \in \mathbf{S}^{\Delta_{\mathsf{pv}}}$, and for a resource term $m \in \Delta_{\mathsf{pv}}$, we denote by $(\varphi)_m$ the coefficient of m in φ, that correspond to $\prod_{m_i = m} a_i$.

All the constructors of Δ_{pv} are linear, in the sense that we can write *e.g.* $\lambda x \left(\sum_{i \in I} a_i \cdot m_i\right) = \sum_{i \in I} a_i \cdot \lambda x m_i$, (see Introduction for those notations). This allows us to give the definition of full Taylor expansion with coefficients as follows:

▶ **Definition 11** (Full Taylor expansion). *Let $\mathbf{S}$ be any commutative semiring with fractions. We define quantitative Taylor expansion, which is a function $()^* : \Lambda_{\mathsf{pv}} \to \mathbf{S}^{\Delta_{\mathsf{pv}}}$, and consists in linear combinations of elements in $\mathcal{T}_{\mathsf{pv}}$.*

- $x^* = x$.
- $(\lambda x M)^* = \lambda x M^*$
- $(\langle M \rangle N)^* = \langle M^* \rangle N^*$
- $((M, N))^* = (M^*, N^*)$
- $(\iota_i(M))^* = (i, M^*)$
- $(\pi_i(M))^* = \pi_i((^*M))$
- $\mathbf{case}(M, x_1 \cdot N_1, x_2 \cdot N_2)^* = \sum_{i \in \{1,2\}} \sum_{r \in \Delta_{\mathsf{pv}}} ((M^*) = (i, r)) \cdot (N_i[M/x_i])^*$
- $(M^!)^* = \sum_{k \in \mathbf{N}} \frac{1}{k!} [M^*, \ldots, M^*]_k$
- $(\mathbf{der}(M))^* = \mathbf{der}(M^*)$

Taylor expansion of fixpoints is defined inductively. We set a combination $\mathbf{fix}_x(M)^{*k}$ for all $k \in \mathbf{N}$, which corresponds to k unfoldings of M in x, as a quantitative version of the sets $\mathbf{fix}_x^k(m)$ of Definition 5.

- $(\mathbf{fix}_x(M))^{*0} = (M[[]/x])^*$

$$(\mathbf{fix}_x(M))^{*k+1} = \sum_{m \in \mathcal{T}_{\mathsf{pv}}(M)} \sum_{\overrightarrow{m} \in (\mathbf{fix}_x^k(M))^!} (M^*)_m \prod_{i=1}^{\mathsf{deg}_x(m)} ((\mathbf{fix}_x(M))^{*k})^!_{\overline{m}_i} \cdot$$
$$m[\overline{m}_1/x_1, \ldots, \overline{m}_{\mathsf{deg}_x(m)}/x_{\mathsf{deg}_x(m)}]$$

and we set $(\mathbf{fix}_x(M))^* = \sum_{k \in \mathbf{N}} (\mathbf{fix}_x(M))^{*k}$.

We also need to give a quantitative version of the splitting operator, in order to make one step-reduction commute with quantitative Taylor expansion defined above.

▶ **Definition 12** (Quantitative split). *We define for all $k \in \mathbf{N}$ and all resource value v the weighted finite sum $\mathbf{split}_+^k(v)$ as follows : if $v \in \{1, 2\}$ or $v = x$, then $\mathbf{split}_+^k(v) = (v, \ldots, v)_k$. If $v = \overline{m}$, then $\mathbf{split}_+^k(v) = \sum_{\overline{m}_1 + \ldots + \overline{m}_k = \overline{m}} \frac{|\overline{m}|!}{|\overline{m}_1|! \ldots |\overline{m}_k|!} \cdot (\overline{m}_1, \ldots, \overline{m}_k)$. If $v = (v_1, v_2)$, then $\mathbf{split}_+^k(v)$ is defined as following, setting $\overrightarrow{v}_i = (v_{i,1}, \ldots, v_{i,k})$:*

$$\sum_{\substack{(v_{1,1}, \ldots, v_{1,k}) \\ \in |\mathbf{split}_+^k(v_1)|}} \sum_{\substack{(v_{2,1}, \ldots, v_{2,k}) \\ \in |\mathbf{split}_+^k(v_2)|}} \left(\mathbf{split}_+^k(v_1)\right)_{\overrightarrow{v}_1} \left(\mathbf{split}_+^k(v_2)\right)_{\overrightarrow{v}_2} \cdot ((v_{1,1}, v_{2,1}), \ldots, (v_{1,k}, v_{2,k}))$$

We now introduce a reduction rule that takes into account the coefficients of definition 12.

▶ **Definition 13** (Quantitative resource reduction $\to_{\mathsf{rpv}+}$). *Let $m \in \Delta_{\mathsf{pv}}$ and $k = degx(m)$.*

$$\langle \lambda x m \rangle v \to_{\mathsf{rpv}+} \sum_{(v_1,\ldots,v_k) \in \Delta_{\mathsf{pv}}^k} \left(\mathbf{split}_+^k(v)\right)_{(v_1,\ldots,v_k)} m[v_1/x_1,\ldots,v_k/x_k]$$

If $m \to_{\mathsf{rpv}} n$ by reducing a redex of another shape than $\langle \lambda x m \rangle n$, then we also set $m \to_{\mathsf{rpv}+} n$.

Notice that if $m \to_{\mathsf{rpv}+} \sum_{i=1}^k a_i \cdot n_i$, then for all $i \in \{1,\ldots,k\}$ such that $a_i \neq 0$, we have $m \to_{\mathsf{rpv}} n_i$.

▶ **Definition 14** (Reduction between combinations). *We define a reduction $\Rightarrow \subseteq \mathbf{S}^{\Delta_{\mathsf{pv}}} \times \mathbf{S}^{\Delta_{\mathsf{pv}}}$. Given a family of resource terms $(m_i)_{i \in I}$ and a family of finite sums of resources terms $(\nu_i)_{i \in I}$ such that for all $i \in I$, and for all $n \in |\nu_i|$ the set $\{j \in I \mid m_j \to_{\mathsf{rpv}+}^{=} n\}$ is finite.*

In that case, we set $\sum_{i \in I} a_i \cdot m_i \Rightarrow \sum_{i \in I} a_i \cdot n_i$ as soon as $m_i \to_{\mathsf{rpv}}^{=} n_i$ for all $i \in I$.

▶ **Lemma 15.** *Let $M \in \Lambda_{\mathsf{pv}}$ with $M^* = \sum_{i \in I} a_i \cdot m_i$ and $\varphi = \sum_{i \in I} a_i \cdot \nu_i$ such that $m_i \to_{\mathsf{rpv}+}^{=} \nu_i$ for all $i \in I$. Then, for all $i \in I$ and for all $n \in |\nu_i|$, n has a finite coefficient in φ.*

In other words, the reduction $\Rightarrow$ is always defined on Taylor expansion.

Proof. This is an immediate consequence of Lemma 10 and Definition 13. ◀

▶ **Lemma 16.** *Let $m \in \Delta_{\mathsf{pv}}$, with $\mathsf{deg}_x(m) = k$, and V a value of Λ_{pv}.*

$$\sum_{v \in \mathcal{T}_{\mathsf{pv}}(V)} \sum_{\substack{(v_1,\ldots,v_k) \\ \in \mathbf{split}^k(v)}} (V^*)_v \left(\mathbf{split}_+^k(v)\right)_{(v_1,\ldots,v_k)} \cdot m[v_1/x_1,\ldots,v_k/x_k]$$

$$= \sum_{\substack{(v_1,\ldots,v_k) \\ \in \mathcal{T}_{\mathsf{pv}}(V)^k}} \prod_{i=1}^k (V^*)_{v_i} \cdot m[v_1/x_1,\ldots,v_k/x_k]$$

Proof. The proof is by induction on V.

- If V is a variable, then all the coefficients $(V^*)_{v_i}$ are equal to 1, and the result is trivial.
- If $V = N^!$, then we want to establish the following, for any $k \in \mathbf{N}$:

$$\sum_{\substack{\overline{n} \\ \in \mathcal{T}_{\mathsf{pv}}(N)^!}} \sum_{\substack{(\overline{n}_1,\ldots,\overline{n}_k) \\ \in \mathbf{split}^k(\overline{n})}} \left(\mathbf{split}_+^k(\overline{n})\right)_{(\overline{n}_1,\ldots,\overline{n}_k)} \prod_{i=1}^{|\overline{n}|} (N^*)_{n_i} \frac{1}{|\overline{n}|!} \cdot m[\overline{n}_1/x_1,\ldots,\overline{n}_k/x_k]$$

$$= \sum_{\substack{(\overline{n}_1,\ldots,\overline{n}_k) \\ \in \mathcal{T}_{\mathsf{pv}}(N^!)^k}} \frac{1}{|\overline{n}_1|!\ldots|\overline{n}_k|!} \prod_{i=1}^k \prod_{j=1}^{|\overline{n}_i|} (N^*)_{n_{i,j}} \cdot m[\overline{n}_1/x_1,\ldots,\overline{n}_k/x_k]$$

Where for all $i \leq k$, $\overline{n}_i = [n_{i,1},\ldots,n_{i,|\overline{n}_i|}]$.

This equation is verified by looking at the definition of $\mathbf{split}_+^k$. $\left(\mathbf{split}_+^k(\overline{n})\right)_{(\overline{n}_1,\ldots,\overline{n}_k)}$ is equal to $\frac{|\overline{n}|!}{|\overline{n}_1|!\ldots|\overline{n}_k|!}$, which is enough to simplify the above equation and conclude this case.

- If $V = (V_1, V_2)$. Then we want to establish:

$$\sum_{\substack{(v_1,v_2)\\ \in \mathcal{T}_{\mathsf{pv}}((V_1,V_2))}} \sum_{\substack{(u_1,\ldots,u_k)\\ \in \mathbf{split}^k((v_1,v_2))}} (V_1,V_2)^*{}_{(v_1,v_2)} \left(\mathbf{split}^k_+((v_1,v_2))\right)_{(u_1,\ldots,u_k)} \cdot m[u_1/x_1,\ldots,u_k/x_k]$$

$$= \sum_{\substack{(u_1,\ldots,u_k)\\ \in \mathcal{T}_{\mathsf{pv}}((V_1,V_2))^k}} \prod_{i=1}^{k} (V_1^*)_{v_{1,i}} \prod_{j=1}^{k} (V_2^*)_{v_{2,j}} \cdot m[u_1/x_1,\ldots,u_k/x_k]$$

Where $(u_1,\ldots,u_k) = ((v_{1,1},v_{2,1}),\ldots,(v_{1,k},v_{2,k}))$, for $(v_{i,1},\ldots,v_{i,k}) \in \mathbf{split}^k(v_i)$.
By induction hypothesis, we have for $i \in \{1,2\}$:

$$\sum_{v_i \in \mathcal{T}_{\mathsf{pv}}(V_i)} \sum_{\substack{(v_{i,1},\ldots,v_{i,k})\\ \in \mathbf{split}^k(v_i)}} (V_i^*)_{v_i} \left(\mathbf{split}^k_+(v_i)\right)_{(v_{i,1},\ldots,v_{i,k})} \cdot m[v_{i,1}/x_1,\ldots,v_{i,k}/x_k]$$

$$= \sum_{\substack{(v_{i,1},\ldots,v_{i,k})\\ \in \mathcal{T}_{\mathsf{pv}}(V_i)^k}} \prod_{j=1}^{k} (V_i^*)_{v_{i,j}} \cdot m[v_{i,1}/x_1,\ldots,v_{i,k}/x_k]$$

Which allows us to conclude this case since $((V_1,V_2)^*)_{(v_{1,i},v_{2,j})} = (V_1^*)_{v_{1,i}} \times (V_2^*)_{v_{2,j}}$ and $\left(\mathbf{split}^k_+((v_1,v_2))\right)_{(u_1,\ldots,u_k)} = \prod_{i=1}^{2} \left(\mathbf{split}^k_+(v_i)\right)_{(v_{i,1},\ldots,v_{i,k})}$
- The case $V = \iota_i(V')$ is proved in the same way by induction hypothesis. ◀

▷ Property 3. $(M[N/x])^* =$

$$\sum_{m \in \mathcal{T}_{\mathsf{pv}}(M)} \sum_{(n_1,\ldots,n_k) \in \mathcal{T}_{\mathsf{pv}}(N)^k} (M^*)_m \prod_{i=1}^{k} (N^*)_{n_i} \cdot m[n_1/x_1,\ldots,n_k/x_k]$$

where $k = \mathsf{deg}_x(m)$.

Proof. Easy induction on M. ◀

We can finally state the main result of this section and of the paper: Theorem 17 establishes the simulation of Λ_{pv} operational semantics in Taylor expansion with coefficients.

▶ **Theorem 17.** *Let $M, M' \in \Lambda_{\mathsf{pv}}$, if $M \to_{\mathsf{pv}} M'$, then $M^* \Rightarrow M'^*$.*

Proof. We use Proposition 9, and verify that it extends to full Taylor expansion, keeping all coefficients in the right place.

- If $M = \langle \lambda x N \rangle V$ and $M' = N[V/x]$, then $M^* =$

$$\sum_{n \in \mathcal{T}_{\mathsf{pv}}(N)} \sum_{v \in \mathcal{T}_{\mathsf{pv}}(V)} (N^*)_n (V^*)_v \cdot \langle \lambda x n \rangle v$$

$$\Rightarrow \sum_{n \in \mathcal{T}_{\mathsf{pv}}(N)} \sum_{v \in \mathcal{T}_{\mathsf{pv}}(V)} \sum_{\substack{(v_1,\ldots,v_k)\\ \in \mathbf{split}^k(v)}} (N^*)_n (V^*)_v \left(\mathbf{split}^k_+(v)\right)_{(v_1,\ldots,v_k)} \cdot n[v_1/x_1,\ldots,v_k/x_k]$$

$$= \sum_{n \in \mathcal{T}_{\mathsf{pv}}(N)} \sum_{(v_1,\ldots,v_k) \in \mathcal{T}_{\mathsf{pv}}(V)^k} (N^*)_n \prod_{i=1}^{k} (V^*)_{v_i} \cdot n[v_1/x_1,\ldots,v_k/x_k]$$

The last equality is obtained by Lemma 16, and is equal to $N[V/x]^*$ by Property 3.

- If $M = \mathbf{case}((\iota_i(V), x_1 \cdot M_1, x_2 \cdot M_2))$ and $M' = M_i[V/x_i]$, then $M^* =$

$$\sum_{j\in\{1,2\}} \sum_{r\in\Delta_{\mathsf{pv}}} ((i,V)^* = (j,r)) \cdot N_j^*[V^*/x_{j,1}, \dots, V^*/x_{j,k}]$$

$$\Rightarrow N_i^*[V^*/x_{i,1}, \dots, V^*/x_{i,k}]$$

  Which is equal to $(N[V/x])^*$ by Property 3.
- If $M = \mathbf{der}(N^!)$ and $M' = N$, then we verify immediately $(\mathbf{der}(N^!))^* = \mathbf{der}((N^!)^*) = \mathbf{der}((N^*)^!) = N^*$, since $\mathbf{der}([n_1, \dots, n_k]) \to_{\mathsf{rpv}} 0$ if $k \neq 1$.
- If $M = \mathbf{fix}_x(N)$, then, $M^* = (M[(\mathbf{fix}_x M)^!/x])^*$. Property 3 and an examination of the definition of Taylor expansion of fixpoint is sufficient to verify this point.
- The projections rules are obtained by a straightforward application of the definitions.

◀

5 Conclusions

We have introduced a new resource calculus reflecting Call-By-Push-Value resource handling and based on Linear Logic semantics. We have then defined Taylor expansion for Call-By-Push-Value as an approximation theory of Call-By-Push-Value encounting for resources. Then, we have shown that it behaves well with respect to the original operational semantics: Taylor expansion with coefficients commutes with reduction in Λ_{pv}. For future work, three directions shall be explored:

- The calculus can be extended in order to define inductive and coinductive datatypes. Integers, for instance, could be defined by adding to our syntax (): $\underline{0} = \iota_1(), \underline{k+1} = \iota_2(\underline{k})$, and all integers defined in this way have the type $\iota = (1 \oplus \iota)$. The successor **suc** can then be defined as the second injection. Then, if x has no free occurrence in N_1, the term $\mathbf{case}(M, x \cdot N_1, y \cdot N_2)$ is an adequate encoding of an "if zero" conditional $\mathbf{If}(M, N_1, y \cdot N_2)$ (where the value to which M evaluates is passed to the following computation).
  The coinductive datatype of streams can also be defined: let A be a positive type, $S_A = !(A \otimes S_A)$ is the type of lazy streams of type A (the tail of the stream being always encapsulated in an exponential, the evaluation is postponed). We can construct a term of type $S_A \multimap \iota \multimap A$ which computes the k-th element of a stream:

$$\mathbf{fix}_f\ (\lambda x \lambda y(\mathbf{If}(y, \pi_1(\mathbf{der}(x)), z \cdot \langle \mathbf{der}(f)\rangle \pi_2 \langle \mathbf{der}(x)\rangle z)))$$

  and a term of type $!(\iota \multimap A) \multimap S_A$:

$$\mathbf{fix}_f\ \left(\lambda g \left(\mathbf{der}(g)\underline{0}, \langle \mathbf{der}(f)\rangle(\lambda x \langle \mathbf{der}(g)\rangle \mathbf{suc}(x))^!\right)\right)$$

  which builds a stream by applying inductively a function to an integer. There are other classical constructions, such as lists, that can be constructed with these ingredients. For a more detailed presentation, see Ehrhard and Tasson's work [14]. We have good hope that this kind of extensions can be incorporated in our resource driven-constructions.
- Extend our constructions in a probabilistic setting, to fit with existing quantitative models like probabilistic coherence spaces. Indeed Lemma 10, which is crucial to define reduction on quantitative Taylor expansion, strongly relies on the uniformity of the calculus, *i.e* we use the fact that all resource terms appearing in the Taylor expansion of a Call-By-Push-Value term have the same *shape* (there is a correspondance between their syntactic trees). The extension seems highly non trivial. But, Dal Lago and Leventis' recent work [19] might be a starting point.

References

1. Samson Abramsky and Guy McCusker. Call-by-value games. In *CSL*, volume 1414 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 1997.
2. J. Chouquet. Taylor expansion, finiteness and strategies. In *MFPS 2019*, 2019.
3. Jules Chouquet and Lionel Vaux Auclair. An application of parallel cut elimination in unit-free multiplicative linear logic to the taylor expansion of proof nets. In *CSL*, volume 119 of *LIPIcs*, pages 15:1–15:17. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.
4. Pierre-Louis Curien, Marcelo P. Fiore, and Guillaume Munch-Maccagnoni. A theory of effects and resources: adjunction models and polarised calculi. In *POPL 2016, St. Petersburg, FL, USA, January 20 - 22, 2016*, 2016.
5. V. Danos and T. Ehrhard. Probabilistic coherence spaces as a model of higher-order probabilistic computation. *Inf. Comput.*, 209(6):966–991, 2011.
6. Jeff Egger, Rasmus Ejlers Møgelberg, and Alex K. Simpson. The enriched effect calculus: syntax and semantics. *J. Log. Comput.*, 24:615–654, 2014.
7. T. Ehrhard. On köthe sequence spaces and linear logic. *Mathematical Structures in Computer Science*, 12(5):579–623, 2002.
8. T. Ehrhard. Finiteness spaces. *Mathematical Structures in Computer Science*, 2005.
9. T. Ehrhard and L. Regnier. The differential lambda-calculus. *Theor. Comput. Sci.*, 2003.
10. T. Ehrhard and L. Regnier. Uniformity and the Taylor expansion of ordinary lambda-terms. *Theor. Comput. Sci.*, 403(2-3):347–372, 2008.
11. Thomas Ehrhard. Collapsing non-idempotent intersection types. In *CSL*, volume 16 of *LIPIcs*, pages 259–273. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012.
12. Thomas Ehrhard. Call-by-push-value from a linear logic point of view. In *ESOP*, volume 9632 of *Lecture Notes in Computer Science*, pages 202–228. Springer, 2016.
13. Thomas Ehrhard and Giulio Guerrieri. The bang calculus: an untyped lambda-calculus generalizing call-by-name and call-by-value. In *PPDP*, pages 174–187. ACM, 2016.
14. Thomas Ehrhard and Christine Tasson. Probabilistic call by push value. *Logical Methods in Computer Science*, 15(1), 2019.
15. J-Y. Girard. Linear logic. *Theor. Comput. Sci.*, 50:1–102, 1987.
16. Giulio Guerrieri and Giulio Manzonetto. The bang calculus and the two girard's translations. *CoRR*, abs/1904.06845, 2019. URL: `http://arxiv.org/abs/1904.06845`, `arXiv:1904.06845`.
17. E. Kerinec, G. Manzonetto, and M. Pagani. Revisiting Call-by-value Böhm trees in light of their Taylor expansion. *CoRR*, abs/1809.02659, 2018. `arXiv:1809.02659`.
18. M. Kerjean and C. Tasson. Mackey-complete spaces and power series - a topological model of differential linear logic. *Mathematical Structures in Computer Science*, 28(4):472–507, 2018.
19. Ugo Dal Lago and Thomas Leventis. On the Taylor expansion of probabilistic λ-terms. *CoRR*, abs/1904.09650, 2019. URL: `http://arxiv.org/abs/1904.09650`, `arXiv:1904.09650`.
20. Olivier Laurent and Laurent Regnier. About Translations of Classical Logic into Polarized Linear Logic. In *18th IEEE Symposium on Logic in Computer Science (LICS 2003), 22-25 June 2003, Ottawa, Canada, Proceedings*, pages 11–20. IEEE Computer Society, 2003. URL: `http://dx.doi.org/10.1109/LICS.2003.1210040`, `doi:10.1109/LICS.2003.1210040`.
21. P. B. Levy. Call-by-push-value: Decomposing call-by-value and call-by-name. *Higher-Order and Symbolic Computation*, 19(4):377–414, 2006.
22. Michael Marz, Alexander Rohr, and Thomas Streicher. Full Abstraction and Universality via Realisability. In *14th Annual IEEE Symposium on Logic in Computer Science, Trento, Italy, July 2-5, 1999*, pages 174–182. IEEE Computer Society, 1999. URL: `http://dx.doi.org/10.1109/LICS.1999.782612`, `doi:10.1109/LICS.1999.782612`.
23. Damiano Mazza. An infinitary affine lambda-calculus isomorphic to the full lambda-calculus. In *LICS*, pages 471–480. IEEE Computer Society, 2012.
24. Damiano Mazza, Luc Pellissier, and Pierre Vial. Polyadic approximations, fibrations and intersection types. *PACMPL*, 2(POPL):6:1–6:28, 2018.

25 P-A. Melliès. Categorical semantics of linear logic. In *In: Interactive Models of Computation and Program Behaviour, Panoramas et Synthèses 27, Société Mathématique de France 1–196*, 2009.

26 M. Pagani, C. Tasson, and L. Vaux. Strong normalizability as a finiteness structure via the Taylor expansion of lambda-terms. In *FOSSACS 2016*, pages 408–423, 2016.

27 L. Vaux. Taylor expansion, β-reduction and normalization. In *26th EACSL Annual Conference on Computer Science Logic, CSL 2017, Stockholm, Sweden*, pages 39:1–39:16, 2017.