



HAL
open science

Taylor expansion for Call-By-Push-Value

Jules Chouquet, Christine Tasson

► **To cite this version:**

Jules Chouquet, Christine Tasson. Taylor expansion for Call-By-Push-Value. CSL, Jan 2020, Barcelona, Spain. 10.4230/LIPIcs.CSL.2020.16 . hal-02318600

HAL Id: hal-02318600

<https://hal.science/hal-02318600>

Submitted on 17 Oct 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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 λxM 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.



XX:2 Taylor expansion for Call-By-Push-Value

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

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

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

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

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

82 The principal result of the paper is the simulation of Λ_{pv} reductions in full Taylor
83 expansion, where resource terms take coefficients in a commutative semiring. The key
84 ingredients for this simulation to run are intrinsic to the properties of Δ_{pv} : the dynamics
85 of reduction must reflect the reduction of Λ_{pv} , and the mechanisms of the calculus must
86 enjoy combinatorial properties, so that the coefficients commute with the simulation. More
87 precisely, it means that for $M, N \in \Lambda_{\text{pv}}$ such that M reduces to N , if Taylor expansion of M
88 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
89 approximating M , then we have a notion of reduction such that $\sum_{i \in I} a_i m_i \Rightarrow \sum_{j \in I} a_j n_j$,

90 and for each resource term n , its coefficient in the latter combination is the same as its
91 coefficient in the Taylor expansion of N .

92 Contents of the paper

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

103 Terminology and notations

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

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

112 If σ is a linear combination of terms $\sum_{i \in I} a_i \cdot m_i$, we use the notation $\lambda x \sigma = \sum_{i \in I} a_i \cdot \lambda x m_i$,
113 $\text{der}(\sigma) = \sum_{i \in I} a_i \cdot \text{der}(m_i)$, and $\sigma^! = \sum_{k \in \mathbf{N}} \sum_{i_1, \dots, i_k \in I} a_{i_1} \dots a_{i_k} \cdot [m_{i_1}, \dots, m_{i_k}]$. In
114 the same way, if $\tau = \sum_{j \in J} a_j \cdot n_j$, we write $(\sigma, \tau) = \sum_{i \in I} \sum_{j \in J} a_i a_j \cdot (m_i, n_j)$. $\langle \sigma \rangle \tau =$
115 $\sum_{i \in I} \sum_{j \in J} a_i a_j \cdot \langle m_i \rangle n_j$. This notation corresponds to the linearity of syntactic constructors
116 with respect to potentially infinite sums of terms that will appear in Taylor expansion.

117 2 Call-By-Push-Value

118 2.1 Syntax and operational semantics

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

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

$$121 \quad \Lambda_{\text{pv}} : M ::= x \mid \lambda x M \mid \langle M \rangle M \mid \text{case}(M, y \cdot M, z \cdot M) \mid \text{fix}_x(M) \mid (M, M) \mid \pi_1(M) \mid \pi_2(M) \mid$$

$$122 \quad M^! \mid \text{der}(M) \mid \iota_1(M) \mid \iota_2(M)$$

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

$$125 \quad V ::= x \mid M^! \mid (V, V) \mid \iota_1(M) \mid \iota_2(M)$$

126 Positive types: $A, B ::= !I \mid A \otimes B \mid A \oplus B$

127 General types : $I, J ::= A \mid A \multimap I \mid \top$

XX:4 Taylor expansion for Call-By-Push-Value

$$\begin{array}{c}
\frac{}{\Gamma, x : A \vdash x : A} \quad \frac{\Gamma \vdash M : I}{\Gamma \vdash M^! : !I} \quad \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x M : A \multimap B} \quad \frac{\Gamma \vdash M : A \multimap I \quad \Delta \vdash N : A}{\Gamma, \Delta \vdash \langle M \rangle N : I} \\
\\
\frac{\frac{\Gamma \vdash M : A \quad \Delta \vdash N : B}{\Gamma, \Delta \vdash (M, N) : A \otimes B} \quad \frac{\Gamma \vdash M : A_1 \otimes A_2}{\Gamma \vdash \pi_i(M) : A_i} \quad i \in \{1, 2\}}{\Gamma \vdash M : A_i} \quad i \in \{1, 2\} \quad \frac{\Gamma \vdash m : !A}{\Gamma \vdash \mathbf{der}(m) : A} \\
\frac{\Gamma \vdash M_1 : A \oplus B \quad \Delta \vdash M_2 : I \quad \Theta \vdash M_3 : I}{\Gamma, \Delta, \Theta \vdash \mathbf{case}(M_1, y \cdot M_2, z \cdot M_3) : I} \quad \frac{\Gamma, x : !I \vdash M : I}{\Gamma \vdash \mathbf{fix}_x(M) : I}
\end{array}$$

■ **Figure 1** Typing rules for Λ_{pv}

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

$$\begin{array}{ll}
129 \quad \langle \lambda x M \rangle V \rightarrow_{pv} M[V/x] & \mathbf{der}(M^!) \rightarrow_{pv} M \\
130 \quad \pi_i(V_1, V_2) \rightarrow_{pv} V_i & \mathbf{fix}_x(M) \rightarrow_{pv} M[(\mathbf{fix}_x(M))^! / x] \\
131 \quad \mathbf{case}(\iota_i(V), x_1 \cdot M_1, x_2 \cdot M_2) \rightarrow_{pv} M_i[V/x_i] &
\end{array}$$

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

$$\begin{array}{l}
134 \quad E ::= [] \mid \langle M \rangle E \mid \langle E \rangle M \mid \pi_i(E) \mid \iota_i(E) \mid (M, E) \mid (E, M) \mid \mathbf{case}(E, x \cdot M, y \cdot N) \mid \mathbf{der}(E) \\
135 \quad \text{and we set as an additional reduction rule } E[M] \rightarrow_{pv} E[N] \text{ for every } M, N \text{ such that} \\
136 \quad M \rightarrow_{pv} N.
\end{array}$$

137 2.2 An overview of denotational semantics and coalgebras

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

141 Let us describe briefly what is a model of Linear Logic (see [25] for a detailed presentation).
142 It is given by a category \mathcal{L} together with a **symmetric monoidal** structure $(\otimes, 1, \lambda, \rho, \alpha, \sigma)$
143 which is **closed**² and we write $X \multimap Y$ for the **object of linear morphisms**. It has a
144 **cartesian** structure with cartesian product $\&$ and terminal object \top . The category \mathcal{L}
145 is equipped with a **comonad** $! : \mathcal{L} \rightarrow \mathcal{L}$ together with a counit $\mathbf{der}_X \in \mathcal{L}(!X, X)$ and
146 a multiplication $\mathbf{dig}_X \in \mathcal{L}(!X, !!X)$. This comonad comes with a symmetric monoidal
147 structure³ from $(\mathcal{L}, \&)$ to (\mathcal{L}, \otimes) , that is two natural isomorphisms $m^0 \in \mathcal{L}(1, !\top)$ and
148 $m^2 \in \mathcal{L}(!X \otimes !Y, !(X \& Y))$.

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

$$152 \quad \text{erase}_{!X} \in \mathcal{L}(!X, 1) \quad \text{split}_{!X}^2 \in \mathcal{L}(!X, !X \otimes !X)$$

153 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 \perp such that X is isomorphic to $(X \multimap \perp) \multimap \perp$

³ The two isomorphisms 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].

154 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
 155 coalgebra inherits the comonoid structure of $!P$, that is it is equipped with: $\text{erase}_P \in \mathcal{L}(P, 1)$
 156 and $\text{split}_P^2 \in \mathcal{L}(P, P \otimes P)$ defined as:

$$157 \quad \text{erase}_P : P \xrightarrow{h_P} !P \xrightarrow{w_P} 1 \quad \text{split}_P^2 : P \xrightarrow{h_P} !P \xrightarrow{c_P} !P \otimes !P \xrightarrow{\mathbf{der}_P \otimes \mathbf{der}_P} P \otimes P.$$

158 Using similar computation, we can define $\text{split}_P^k \in \mathcal{L}(P, \underbrace{P \otimes \dots \otimes P}_k)$.

159 Notice that the structure of comonad of $!$ induces a coalgebras structure on $!X$. Moreover,
 160 every construction of positive type preserves the coalgebra structure. To define the coalgebraic
 161 structure of $P \otimes Q$ where P and Q are both coalgebras, let us first define the morphisms
 162 $\mu^0 \in \mathcal{L}(1, !1)$ and $\mu^2 \in \mathcal{L}(!X \otimes !Y, !(X \otimes Y))$ as

$$163 \quad \mu^0 : 1 \xrightarrow{m^0} !\top \xrightarrow{\mathbf{dig}_\top} !!\top \xrightarrow{!(m^0)^{-1}} !1$$

$$164 \quad \mu^2 : !X \otimes !Y \xrightarrow{m^2} !(X \& Y) \xrightarrow{\mathbf{dig}_{X \& Y}} !!(X \& Y) \xrightarrow{!(m^2)^{-1}} !(X \otimes Y) \xrightarrow{!(\mathbf{der}_X \otimes \mathbf{der}_Y)} !(X \otimes Y).$$

166 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
 167 the coproduct is entirely defined by the morphisms for $i \in \{1, 2\}$: $P_i \xrightarrow{h_{P_i}} !P_i \xrightarrow{!in_i} !(P_1 \oplus P_2)$
 168 if the category has coproducts.

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

170 Example

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

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

$$180 \quad \mathbf{der}_X = \{([a], a) \mid a \in X\} \quad \mathbf{dig}_X = \{(\bar{m}, [\bar{m}_1, \dots, \bar{m}_k]) \mid \bar{m}_1 + \dots + \bar{m}_k = \bar{m}\}.$$

181 The comonoidal structure of $!X$ is

$$182 \quad \text{erase}_{!X} = \{([\], *)\} \quad \text{split}_{!X}^2 = \{(\bar{m}, (\bar{m}_1, \bar{m}_2)) \mid \bar{m}_1 + \bar{m}_2 = \bar{m}\}.$$

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

$$186 \quad h_P = \{(((i, \bar{m}_i)), (\bar{m}_Y, \bar{m}_Z)), [((i, \bar{x}_i^1), (\bar{y}_1, \bar{z}_1)), \dots, ((i, \bar{x}_i^k), (\bar{y}_k, \bar{z}_k))]\}$$

$$187 \quad \bar{m}_i = \bar{x}_i^1 + \dots + \bar{x}_i^k, \bar{m}_Y = \bar{y}_1 + \dots + \bar{y}_k, \bar{m}_Z = \bar{z}_1 + \dots + \bar{z}_k\},$$

189 and is equipped with the comonoidal structure:

$$190 \quad \text{erase}_P = \{(((i, [\]), ([\], [\]), *)\}$$

$$191 \quad \text{split}_P^2 = \{(((i, \bar{m}_i), (\bar{m}_Y, \bar{m}_Z)), ((i, (\bar{m}_i^1 + \bar{m}_i^2)), ((\bar{m}_Y^1 + \bar{m}_Y^2), (\bar{m}_Z^1 + \bar{m}_Z^2))))\}$$

$$192 \quad \bar{m}_i^1 + \bar{m}_i^2 = \bar{m}_i, \bar{m}_Y^1 + \bar{m}_Y^2 = \bar{m}_Y, \bar{m}_Z^1 + \bar{m}_Z^2 = \bar{m}_Z\}.$$

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

$$\begin{array}{c}
 \begin{array}{ccc}
 \overline{m}_i & \overline{m}_Y & \overline{m}_Z \\
 | & \underbrace{\hspace{1.5cm}} & \\
 (i, \overline{m}_i) & (\overline{m}_Y, \overline{m}_Z) & \\
 \underbrace{\hspace{2.5cm}} & & \\
 ((i, \overline{m}_i), (\overline{m}_Y, \overline{m}_Z)) & &
 \end{array} & \xrightarrow{h_P} &
 \begin{array}{ccc}
 \overline{x}_i^1 & \overline{y}^1 & \overline{z}^1 \\
 | & \underbrace{\hspace{1.5cm}} & \\
 (i, \overline{x}_i^1) & (\overline{y}^1, \overline{z}^1) & \\
 \underbrace{\hspace{2.5cm}} & & \\
 ((i, \overline{x}_i^1), (\overline{y}^1, \overline{z}^1)), & \dots & ((i, \overline{x}_i^k), (\overline{y}^k, \overline{z}^k))
 \end{array}
 \end{array}$$

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

196 3 Resource calculus for Call-By-Push-Value

197 We introduce a typed resource calculus, able to simulate the operational semantics of Λ_{pv} .
 198 The conditional construction is considered through tests of equality, and there is no explicit
 199 fixpoint. The main difference with other resource calculi, like Call-By-Name or Call-By-Value,
 200 is that redexes of shape $\langle \lambda x m \rangle \overline{n}$ are not enough to entail Λ_{pv} reduction. Indeed, the notion
 201 of value is too wide to be entirely captured in multisets of approximants: $\langle \lambda x M \rangle (V_1, V_2)$ is a
 202 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,
 203 while keeping it sensitive to resource consumption. We proceed so with the introduction of a
 204 splitting operator, which allows us to duplicate a value using the structure of its positive
 205 type.

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

$$\begin{array}{l}
 208 \quad \Delta_{\text{pv}} : m ::= x \mid 1 \mid 2 \mid \lambda x m \mid \langle m \rangle m \mid (m = m) \cdot m \mid (m, m) \mid \pi_1(m) \mid \pi_2(m) \\
 209 \quad \quad \quad \mid [m, \dots, m] \mid \mathbf{der}(m) \\
 210
 \end{array}$$

211 *We distinguish the values of the calculus:*

$$212 \quad v ::= x \mid 1 \mid 2 \mid [m, \dots, m] \mid (v, v)$$

$$\begin{array}{c}
 \frac{}{\Gamma, x : A \vdash x : A} \quad \frac{\Gamma \vdash m_i : I, i \in \{1, \dots, k\}}{\Gamma \vdash [m_1, \dots, m_k] : !I} \quad \frac{\Gamma, x : A \vdash m : B}{\Gamma \vdash \lambda x m : A \multimap B} \\
 \frac{\Gamma \vdash m : A \multimap I}{\Gamma, \Delta \vdash \langle m \rangle n : I} \quad \frac{\Delta \vdash n : A}{\Gamma \vdash \mathbf{der}(m) : A} \\
 \frac{\Gamma \vdash m : A \quad \Delta \vdash n : B}{\Gamma, \Delta \vdash (m, n) : A \otimes B} \quad \frac{\Gamma \vdash m : A_1 \otimes A_2}{\Gamma \vdash \pi_i(m) : A_i} \quad i \in \{1, 2\} \quad \frac{\Gamma \vdash m : A_i}{\Gamma \vdash (i, m) : A_1 \oplus A_2} \quad i \in \{1, 2\} \\
 \frac{\Gamma \vdash m_1 : A_1 \oplus A_2 \quad \Delta \vdash m_2 : A_i \quad \Theta \vdash m_3 : I}{\Gamma, \Delta, \Theta \vdash (m_1 = (i, m_2)) \cdot m_3 : I}
 \end{array}$$

■ **Figure 3** Typing rules for Δ_{pv}

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

XX:8 Taylor expansion for Call-By-Push-Value

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

■ If $m = e[m']$ and $n = e[n']$ for $n \rightarrow_{\text{rpv}} n'$, we conclude by induction hypothesis.

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

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

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} :*

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

▷ **Property 1.** Let $M \in \Lambda_{\text{pv}}$, $m \in \mathcal{T}_{\text{pv}}(M)$, 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 $\mathcal{T}_{\text{pv}}(V)$ for a value V matches exactly the resource values of Definition 2. It is easy to verify that $\mathbf{split}^k(v)$ is always defined, and that if $m \in \mathcal{T}_{\text{pv}}(M)$ 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 $\langle \lambda x m \rangle n \in \mathcal{T}_{\text{pv}}(\langle \lambda x M \rangle N)$. There is a term m' such that $\langle \lambda x m \rangle n \rightarrow_{\text{rpv}} m'$ by reducing the most external redex if and only if N is a value. Recall moreover that $\langle \lambda x M \rangle N \rightarrow_{\text{pv}} M[N/x]$ if and only if N is a value.*

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

287 **Proof.** By induction on M , using Property 1 :

- 288 ■ If $M = x$, then $m = x$ and $\mathbf{split}^k(m) = (x, \dots, x)_k$. We conclude since $\mathcal{T}_{\text{pv}}(x) = \{x\}$.
- 289 ■ If $M = N^l$, then $m = [n_1, \dots, n_l]$, and for all $i \in \{1, \dots, l\}$, $n_i \in \mathcal{T}_{\text{pv}}(N)$. We have
 290 $(m_1, \dots, m_k) = (\bar{n}_1, \dots, \bar{n}_k)$ with $\sum_{i=1}^k \bar{n}_i = [n_1, \dots, n_l]$. Then, each \bar{n}_i is a multiset of
 291 elements in $\mathcal{T}_{\text{pv}}(N)$, and $\bar{n}_i \in \mathcal{T}_{\text{pv}}(N^l) = \mathcal{T}_{\text{pv}}(M)$.
- 292 ■ If $M = (N, N')$, then $m = (n, n')$ for $n \in \mathcal{T}_{\text{pv}}(N)$ and $n' \in \mathcal{T}_{\text{pv}}(N')$. $(m_1, \dots, m_k) =$
 293 $((n_1, n'_1), \dots, (n_k, n'_k))$ with $(n_1, \dots, n_k) \in \mathbf{split}^k(N)$ and $(n'_1, \dots, n'_k) \in \mathbf{split}^k(N')$. By
 294 induction hypothesis, for all $i \in \{1, \dots, k\}$, $n_i \in \mathcal{T}_{\text{pv}}(N)$ and $n'_i \in \mathcal{T}_{\text{pv}}(N')$. Then for all i ,
 295 $(n_i, n'_i) \in \mathcal{T}_{\text{pv}}(N, N') = \mathcal{T}_{\text{pv}}(M)$.
- 296 ■ If $M = \iota_j(N)$, then $m = (j, n)$ for $n \in \mathcal{T}_{\text{pv}}(N)$ and $\mathbf{split}^k(m) = ((j, n_1), \dots, (j, n_k))$ with
 297 $(n_1, \dots, n_k) \in \mathbf{split}^k(n)$. By induction hypothesis, for all $i \in \{1, \dots, k\}$, $n_i \in \mathcal{T}_{\text{pv}}(N)$.
 298 Then for all i , $(j, n_i) \in \mathcal{T}_{\text{pv}}(\iota_j(N)) = \mathcal{T}_{\text{pv}}(M)$.

299 ◀

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

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

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

- 306 ■ If $M = x$, then $m = x, k = 1, m[n_1/x_1] = n_1$, and $M[N/x] = N$. Then $m[n_1/x_1] \in$
 307 $\mathcal{T}_{\text{pv}}(M[N/x])$.
- 308 ■ If $M = \lambda y M'$, then $\text{deg}_x(M) = \text{deg}_x(M'), m = \lambda y m'$ for $m' \in \mathcal{T}_{\text{pv}}(M')$. By induc-
 309 tion hypothesis, $m'[n_1/x_1, \dots, n_k/x_k] \in \mathcal{T}_{\text{pv}}(M'[N/x])$. Since $m[n_1/x_1, \dots, n_k/x_k] =$
 310 $\lambda y m'[n_1/x_1, \dots, n_k/x_k]$, we conclude.
- 311 ■ If $M = \langle M_1 \rangle M_2$, then $m = \langle m_1 \rangle m_2$ for $m_i \in \mathcal{T}_{\text{pv}}(M_i)$, and $\text{deg}_x(m) = l_1 + l_2$ for
 312 $l_1 = \text{deg}_x(m_1)$ and $l_2 = \text{deg}_x(m_2)$. By induction hypothesis, $m_1[n_1/x_1, \dots, n_{l_1}/x_{l_1}] \in$
 313 $\mathcal{T}_{\text{pv}}(M_1[N/x])$ and $m_2[n_{l_1+1}/x, \dots, n_{l_1+l_2}/x] \in \mathcal{T}_{\text{pv}}(M_2[N/x])$. Since $m[n_1/x_1, \dots, n_k/x_k] =$
 314 $\langle m_1[n_1/x_1, \dots, n_{l_1}/x_{l_1}] \rangle m_2[n_{l_1+1}/x, \dots, n_{l_1+l_2}/x]$, and $M[N/x] = \langle M_1[N/x] \rangle M_2[N/x]$,
 315 we conclude.
- 316 ■ If $M = M^l$, then $m = [m'_1, \dots, m'_l]$ with $m'_i \in \mathcal{T}_{\text{pv}}(M')$ for all i , and $\text{deg}_x(m) = \sum_{i=1}^l k_i$
 317 where $k_i = \text{deg}_x(m'_i)$. By induction hypothesis, $m'_i[n_{k_{i-1}+1}/x_{k_{i-1}+1}, \dots, n_{k_{i-1}+k_i}/x_{k_{i-1}+k_i}] \in$
 318 $\mathcal{T}_{\text{pv}}(M'[N/x])$ for all $i \in \{1, \dots, l\}$ (setting $k_0 = 0$). Then, $M[N/x] = (M'[N/x])^l$, and
 319 we can conclude as before.
- 320 ■ In $M = \text{case}(M', z_1 \cdot N_1, z_2 \cdot N_2)$, then $m = (m' = (i, m'')) \cdot n_i[m''/z_i]$ for $i \in \{1, 2\}, m' \in$
 321 $\mathcal{T}_{\text{pv}}(M'), n_i \in \mathcal{T}_{\text{pv}}(N_i), m'' \in \Delta_{\text{pv}}$. We conclude by induction hypothesis as above.

322 ◀

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

325 We can finally prove the first simulation property:

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

328 **Proof.** By induction on M :

XX:10 Taylor expansion for Call-By-Push-Value

- 329 ■ If $M = \pi_i((M_1, M_2))$ and $M' = M_i$, then $m = \pi_i((m_1, m_2))$ for $m_i \in \mathcal{T}_{\text{pv}}(M_i)$. We
330 conclude since $M \rightarrow_{\text{pv}} M_i$ and $m \rightarrow_{\text{rpv}} m_i$.
- 331 ■ If $M = \mathbf{der}(N^!)$ and $M' = N$, then $m = \mathbf{der}([n_1, \dots, n_k])$, with $n_i \in \mathcal{T}_{\text{pv}}(N)$ for all
332 $i \in \{1, \dots, k\}$. We conclude since $M \rightarrow_{\text{pv}} N$ and $m \rightarrow_{\text{rpv}} n_1$ if $k = 1$ and $m \rightarrow_{\text{rpv}} 0$
333 otherwise.
- 334 ■ If $M = \mathbf{fix}_x(N)$ and $M' = N[(\mathbf{fix}_x(N))^! / x]$, then it is easy to verify that $\mathcal{T}_{\text{pv}}(M) =$
335 $\mathcal{T}_{\text{pv}}(M')$, using Lemma 8 and unfolding the definition of Taylor expansion of fixpoint. We
336 need a reflexive reduction for this case.
- 337 ■ If $M = (\lambda y N)V$ and $M' = N[V/y]$, then $m = \langle \lambda y n \rangle v$ for $n \in \mathcal{T}_{\text{pv}}(N)$ and $v \in \mathcal{T}_{\text{pv}}(V)$. By
338 Property 1, $\mathbf{split}^k(v)$ is defined for any $k \in \mathbf{N}$, then $m \rightarrow_{\text{rpv}} n[v_1/y_{f(1)}, \dots, v_k/y_{f(k)}]$ for
339 $\mathbf{deg}_y(n) = k$ and $(v_1, \dots, v_k) \in \mathbf{split}^k(v)$. By Lemma 7, for all $i \in \{1, \dots, k\}$, $v_i \in \mathcal{T}_{\text{pv}}(V)$,
340 and by the substitution Lemma 8, $n[v_1/y_1, \dots, v_k/y_k] \in \mathcal{T}_{\text{pv}}(N[V/y])$.
- 341 ■ 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, $m = ((i, v) = (j, n)) \cdot$
342 $m_i[v/x_i]$ for $i, j \in \{1, 2\}$, $v \in \mathcal{T}_{\text{pv}}(V)$, $n \in \Delta_{\text{pv}}$, $m_i \in \mathcal{T}_{\text{pv}}(M_i)$. Either $m \rightarrow_{\text{rpv}} 0$, either
343 $(i, v) = (j, n)$ and in this case $m \rightarrow_{\text{rpv}} m_i[n/x_i] = m_i[v/x_i]$. By the substitution Lemma 8
344 we conclude, since we have $M \rightarrow_{\text{pv}} M_i[V/x_i]$ and $m_i[v/x_i] \in \mathcal{T}_{\text{pv}}(M_i[V/x_i])$.
- 345 ■ If $M = E[N]$ and $M' = E[N']$, then we can easily show that there is a resource context e
346 such that $m = e[n]$ and $n \in \mathcal{T}_{\text{pv}}(N)$. By induction hypothesis, either $n \rightarrow_{\text{rpv}} 0$, and then
347 $e[n] = 0$, or there exists n' such that $n \rightarrow_{\text{rpv}} n'$ and $n' \in \mathcal{T}_{\text{pv}}(N')$. We can easily adapt
348 the substitution Lemma to conclude $e[n'] \in \mathcal{T}_{\text{pv}}(E[N'])$.
349 ◀

350 4.2 Embeddings of CBV and CBN

351 Call-By-Push-Value is known to subsume both Call-By-Name and Call-By-Value strategies.
352 In particular, the two strategies can be embedded into Λ_{pv} . If we consider simply typed
353 λ -calculus⁵ Λ , we set two functions $()^v, ()^n : \Lambda \rightarrow \Lambda_{\text{pv}}$, defined in Figure 7. We do not
354 consider here calculi with products, or other constructors, in order to focus in a simple
355 setting on the relation between exponentials and strategies of reduction (see Ehrhard and
356 Tasson's work [14] for more developments). Our embeddings ensure *e.g.* the following
357 property: $((\lambda x M)N)^v \rightarrow_{\text{pv}} (M[N/x])^v$ if and only if N is a variable or an abstraction, and
358 $((\lambda x M)N)^n \rightarrow_{\text{pv}} (M[N/x])^n$ for any M, N .

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

365 ▷ **Property 2.** For any pure λ -term $M \in \Lambda$, $E(\mathcal{T}_{\text{pv}}((M)^v)) = \mathcal{T}^v(M)$ and $E(\mathcal{T}_{\text{pv}}((M)^n)) =$
366 $\mathcal{T}^n(M)$, where E is the function that erases all the derelictions (that do not exist in Δ^n nor
367 in Δ^v) in a set of terms.

368 **Proof.** The proof consists in a simple examination of the definitions. Let us start with
369 Call-By-Value constructions: The variable case is immediate since $\mathcal{T}_{\text{pv}}(x^v) = \{\mathbf{der}(x)\}^!$, 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
$m, n ::= x \mid \lambda x m \mid \langle m \rangle \bar{n}$	$m, n ::= [x_1, \dots, x_k] \mid [\lambda x m_1, \dots, \lambda x m_k] \mid \langle m \rangle n$
$\langle \lambda x m \rangle [n_1, \dots, n_k] \rightarrow m[n_1/x_{f(1)}, \dots, n_k/x_{f(k)}]$ if $k = \text{deg}_x(m)$ and $f \in \mathfrak{S}_k$	$\langle [\lambda x m] \rangle [n_1, \dots, n_k] \rightarrow m[n_1/x_{f(1)}, \dots, n_k/x_{f(k)}]$ if $k = \text{deg}_x(m)$ 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
$\mathcal{T}^n(x) = \{x\}^!$	$\mathcal{T}^v(x) = \{x\}^!$
$\mathcal{T}^n(MN) = \{\langle m \rangle \bar{n} \mid m \in \mathcal{T}^n(M), \bar{n} \in \mathcal{T}^n(N)^!\}$	$\mathcal{T}^v(MN) = \{\langle m \rangle n \mid m \in \mathcal{T}^v(M), n \in \mathcal{T}^v(N)\}$
$\mathcal{T}^n(\lambda x M) = \{\lambda x m \mid m \in \mathcal{T}^n(M)\}$	$\mathcal{T}^v(\lambda x M) = \{[\lambda x m_1, \dots, \lambda x m_k] \mid m_i \in \mathcal{T}^v(M)\}$

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

370 $\mathcal{T}^v(x) = \{x\}^!$. $\mathcal{T}_{\text{pv}}((\lambda x M)^v) = \{[\lambda x m_1, \dots, \lambda x m_k] \mid k \in \mathbf{N}, m_i \in \mathcal{T}_{\text{pv}}(M^v)\}$, we conclude
 371 since by induction hypothesis, $E(\mathcal{T}_{\text{pv}}(M^v)) = \mathcal{T}^v(M)$ and $\mathcal{T}^v(\lambda x M) = \{[\lambda x m'_1, \dots, \lambda x m'_l] \mid$
 372 $l \in \mathbf{N}, m'_i \in \mathcal{T}^v(M)\}$. The application case is managed with a similar argument with
 373 induction hypothesis, and with the fact that $E(\langle \text{der}(M) \rangle N) = \langle E(M) \rangle E(N)$.

374 For Call-By-Name, we only consider the application case (the other being straightforward):
 375 $\mathcal{T}_{\text{pv}}((MN)^n) = \{\langle m \rangle \bar{n} \mid m \in \mathcal{T}_{\text{pv}}(M^n), \bar{n} \in \mathcal{T}_{\text{pv}}(N^n)^!\}$. By induction hypothesis,
 376 $E(\mathcal{T}_{\text{pv}}(M^n)) = \mathcal{T}^n(M)$ and $E(\mathcal{T}_{\text{pv}}(N^n)) = \mathcal{T}^n(N)$, and we can conclude. ◀

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

380 4.3 Finiteness

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

386 ▶ **Lemma 10** (Finiteness of antireduction). *Let $n \in \Delta_{\text{pv}}$ and M in Λ_{pv} . $\{m \in \mathcal{T}_{\text{pv}}(M) \mid$
 387 $m \rightarrow_{\text{pv}}^{\infty} n\}$ is finite.*

388 **(sketch)**. We do not detail the proof, since we can adapt the first author's work [2] for PCF.
 389 The idea is to extend Ehrhard and Regnier's original proof [10], defining a coherence relation
 390 on resource terms in a way $\mathcal{T}_{\text{pv}}(M)$ is always a maximal clique for this relation. In particular,
 391 $\bigcup_{k \in \mathbf{N}} \text{fix}_x^k(m)$ must be a clique.

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

395 4.4 Taylor expansion with coefficients

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

XX:12 Taylor expansion for Call-By-Push-Value

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

■ **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 $\varphi = \sum_{i \in I} a_i \cdot m_i \in \mathbf{S}^{\Delta_{\text{pv}}}$, and for a resource term $m \in \Delta_{\text{pv}}$, 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. $\lambda x (\sum_{i \in I} a_i \cdot m_i) = \sum_{i \in I} a_i \cdot \lambda x m_i$, (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 $(\cdot)^* : \Lambda_{\text{pv}} \rightarrow \mathbf{S}^{\Delta_{\text{pv}}}$, and consists in linear combinations of elements in \mathcal{T}_{pv} .*

■ $x^* = x$.

■ $(\lambda x M)^* = \lambda x M^*$

■ $(\langle M \rangle N)^* = \langle M^* \rangle N^*$

■ $((M, N))^* = (M^*, N^*)$

■ $(\iota_i(M))^* = (i, M^*)$

■ $(\pi_i(M))^* = \pi_i((^*M))$

■ $\mathbf{case}(M, x_1 \cdot N_1, x_2 \cdot N_2)^* = \sum_{i \in \{1,2\}} \sum_{r \in \Delta_{\text{pv}}} ((M^*) = (i, r)) \cdot (N_i[M/x_i])^*$

■ $(M^!)^* = \sum_{k \in \mathbf{N}} \frac{1}{k!} [M^*, \dots, M^*]_k$

■ $(\mathbf{der}(M))^* = \mathbf{der}(M^*)$

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.

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

$$(\mathbf{fix}_x(M))^{*k+1} = \sum_{m \in \mathcal{T}_{\text{pv}}(M)} \sum_{\bar{m} \in (\mathbf{fix}_x^k(M))!} (M^*)_m \prod_{i=1}^{\text{deg}_x(m)} ((\mathbf{fix}_x(M))^{*k})_{\bar{m}_i}^!$$

$$m[\bar{m}_1/x_1, \dots, \bar{m}_{\text{deg}_x(m)}/x_{\text{deg}_x(m)}]$$

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 $\mathbf{split}_+^k(v) = (v, \dots, v)_k$.*

If $v = \bar{m}$, then $\mathbf{split}_+^k(v) = \sum_{\bar{m}_1 + \dots + \bar{m}_k = \bar{m}} \frac{|\bar{m}|!}{|\bar{m}_1|! \dots |\bar{m}_k|!} \cdot (\bar{m}_1, \dots, \bar{m}_k)$. If $v = (v_1, v_2)$, then

$\mathbf{split}_+^k(v)$ is defined as following, setting $\vec{v}_i = (v_{i,1}, \dots, v_{i,k})$:

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

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

433 ► **Definition 13** (Quantitative resource reduction $\rightarrow_{\text{rpv}+}$). Let $m \in \Delta_{\text{pv}}$ and $k = \text{deg}_x(m)$.

$$434 \quad \langle \lambda x m \rangle v \rightarrow_{\text{rpv}+} \sum_{(v_1, \dots, v_k) \in \Delta_{\text{pv}}^k} \left(\text{split}_+^k(v) \right)_{(v_1, \dots, v_k)} m[v_1/x_1, \dots, v_k/x_k]$$

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

436 Notice that if $m \rightarrow_{\text{rpv}+} \sum_{i=1}^k a_i \cdot n_i$, then for all $i \in \{1, \dots, k\}$ such that $a_i \neq 0$, we have
437 $m \rightarrow_{\text{rpv}} n_i$.

438 ► **Definition 14** (Reduction between combinations). We define a reduction $\Rightarrow_{\subseteq} \mathbf{S}^{\Delta_{\text{pv}}} \times \mathbf{S}^{\Delta_{\text{pv}}}$.

439 Given a family of resource terms $(m_i)_{i \in I}$ and a family of finite sums of resources terms
440 $(\nu_i)_{i \in I}$ such that for all $i \in I$, and for all $n \in |\nu_i|$ the set $\{j \in I \mid m_j \rightarrow_{\text{rpv}+}^{\equiv} n\}$ is finite.

441 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 $m_i \rightarrow_{\text{rpv}}^{\equiv} n_i$ for all $i \in I$.

442 ► **Lemma 15.** Let $M \in \Lambda_{\text{pv}}$ with $M^* = \sum_{i \in I} a_i \cdot m_i$ and $\varphi = \sum_{i \in I} a_i \cdot \nu_i$ such that
443 $m_i \rightarrow_{\text{rpv}+}^{\equiv} \nu_i$ for all $i \in I$. Then, for all $i \in I$ and for all $n \in |\nu_i|$, n has a finite coefficient in
444 φ .

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

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

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

$$448 \quad \sum_{v \in \mathcal{T}_{\text{pv}}(V)} \sum_{\substack{(v_1, \dots, v_k) \\ \in \text{split}^k(v)}} (V^*)_v \left(\text{split}_+^k(v) \right)_{(v_1, \dots, v_k)} \cdot m[v_1/x_1, \dots, v_k/x_k]$$

$$449 \quad = \sum_{\substack{(v_1, \dots, v_k) \\ \in \mathcal{T}_{\text{pv}}(V)^k}} \prod_{i=1}^k (V^*)_{v_i} \cdot m[v_1/x_1, \dots, v_k/x_k]$$

451 **Proof.** The proof is by induction on V .

452 ■ If V is a variable, then all the coefficients $(V^*)_{v_i}$ are equal to 1, and the result is trivial.

453 ■ If $V = N^!$, then we want to establish the following, for any $k \in \mathbf{N}$:

$$454 \quad \sum_{\substack{\bar{n} \\ \in \mathcal{T}_{\text{pv}}(N)^!}} \sum_{\substack{(\bar{n}_1, \dots, \bar{n}_k) \\ \in \text{split}^k(\bar{n})}} \left(\text{split}_+^k(\bar{n}) \right)_{(\bar{n}_1, \dots, \bar{n}_k)} \prod_{i=1}^{|\bar{n}|} (N^*)_{n_i} \frac{1}{|\bar{n}|!} \cdot m[\bar{n}_1/x_1, \dots, \bar{n}_k/x_k]$$

$$455 \quad = \sum_{\substack{(\bar{n}_1, \dots, \bar{n}_k) \\ \in \mathcal{T}_{\text{pv}}(N^!)^k}} \frac{1}{|\bar{n}_1|! \dots |\bar{n}_k|!} \prod_{i=1}^k \prod_{j=1}^{|\bar{n}_i|} (N^*)_{n_{i,j}} \cdot m[\bar{n}_1/x_1, \dots, \bar{n}_k/x_k]$$

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

458 This equation is verified by looking at the definition of split_+^k . $\left(\text{split}_+^k(\bar{n}) \right)_{(\bar{n}_1, \dots, \bar{n}_k)}$ is

459 equal to $\frac{|\bar{n}|!}{|\bar{n}_1|! \dots |\bar{n}_k|!}$, which is enough to simplify the above equation and conclude this
460 case.

XX:14 Taylor expansion for Call-By-Push-Value

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

$$\begin{aligned}
 462 \quad & \sum_{\substack{(v_1, v_2) \\ \in \mathcal{T}_{\text{pv}}((V_1, V_2)}} \sum_{\substack{(u_1, \dots, u_k) \\ \in \mathbf{split}^k((v_1, v_2))}} (V_1, V_2)^*_{(v_1, v_2)} \left(\mathbf{split}_+^k((v_1, v_2)) \right)_{(u_1, \dots, u_k)} \cdot m[u_1/x_1, \dots, u_k/x_k] \\
 463 \quad & = \sum_{\substack{(u_1, \dots, u_k) \\ \in \mathcal{T}_{\text{pv}}((V_1, V_2))^k}} \prod_{i=1}^k (V_1^*)_{v_{1,i}} \prod_{j=1}^k (V_2^*)_{v_{2,j}} \cdot m[u_1/x_1, \dots, u_k/x_k] \\
 464 \quad &
 \end{aligned}$$

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

$$\begin{aligned}
 467 \quad & \sum_{v_i \in \mathcal{T}_{\text{pv}}(V_i)} \sum_{\substack{(v_{i,1}, \dots, v_{i,k}) \\ \in \mathbf{split}^k(v_i)}} (V_i^*)_{v_i} \left(\mathbf{split}_+^k(v_i) \right)_{(v_{i,1}, \dots, v_{i,k})} \cdot m[v_{i,1}/x_1, \dots, v_{i,k}/x_k] \\
 468 \quad & = \sum_{\substack{(v_{i,1}, \dots, v_{i,k}) \\ \in \mathcal{T}_{\text{pv}}(V_i)^k}} \prod_{j=1}^k (V_i^*)_{v_{i,j}} \cdot m[v_{i,1}/x_1, \dots, v_{i,k}/x_k] \\
 469 \quad &
 \end{aligned}$$

470 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
 471 $\left(\mathbf{split}_+^k((v_1, v_2)) \right)_{(u_1, \dots, u_k)} = \prod_{i=1}^2 \left(\mathbf{split}_+^k(v_i) \right)_{(v_{i,1}, \dots, v_{i,k})}$

472 ■ The case $V = \iota_i(V')$ is proved in the same way by induction hypothesis.

473

474 ▷ Property 3. $(M[N/x])^* =$

$$475 \quad \sum_{m \in \mathcal{T}_{\text{pv}}(M)} \sum_{(n_1, \dots, n_k) \in \mathcal{T}_{\text{pv}}(N)^k} (M^*)_m \prod_{i=1}^k (N^*)_{n_i} \cdot m[n_1/x_1, \dots, n_k/x_k]$$

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

477 **Proof.** Easy induction on M . ◀

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

480 ► **Theorem 17.** *Let $M, M' \in \Lambda_{\text{pv}}$, if $M \rightarrow_{\text{pv}} M'$, then $M^* \Rightarrow M'^*$.*

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

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

$$\begin{aligned}
 484 \quad & \sum_{n \in \mathcal{T}_{\text{pv}}(N)} \sum_{v \in \mathcal{T}_{\text{pv}}(V)} (N^*)_n (V^*)_v \cdot \langle \lambda x n \rangle v \\
 485 \quad & \Rightarrow \sum_{n \in \mathcal{T}_{\text{pv}}(N)} \sum_{v \in \mathcal{T}_{\text{pv}}(V)} \sum_{\substack{(v_1, \dots, v_k) \\ \in \mathbf{split}^k(v)}} (N^*)_n (V^*)_v \left(\mathbf{split}_+^k(v) \right)_{(v_1, \dots, v_k)} \cdot n[v_1/x_1, \dots, v_k/x_k] \\
 486 \quad & = \sum_{n \in \mathcal{T}_{\text{pv}}(N)} \sum_{(v_1, \dots, v_k) \in \mathcal{T}_{\text{pv}}(V)^k} (N^*)_n \prod_{i=1}^k (V^*)_{v_i} \cdot n[v_1/x_1, \dots, v_k/x_k] \\
 487 \quad &
 \end{aligned}$$

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

489 ■ 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 $M^* =$

$$490 \quad \sum_{j \in \{1,2\}} \sum_{r \in \Delta_{pv}} ((i, V)^* = (j, r)) \cdot N_j^*[V^*/x_{j,1}, \dots, V^*/x_{j,k}]$$

$$491 \quad \Rightarrow N_i^*[V^*/x_{i,1}, \dots, V^*/x_{i,k}]$$

493 Which is equal to $(N[V/x])^*$ by Property 3.

494 ■ If $M = \mathbf{der}(N^!)$ and $M' = N$, then we verify immediately $(\mathbf{der}(N^!))^* = \mathbf{der}((N^!)^*) =$
 495 $\mathbf{der}((N^*)^!) = N^*$, since $\mathbf{der}([n_1, \dots, n_k]) \rightarrow_{\text{rpv}} 0$ if $k \neq 1$.

496 ■ If $M = \mathbf{fix}_x(N)$, then, $M^* = (M[(\mathbf{fix}_x M)^! / x])^*$. Property 3 and an examination of the
 497 definition of Taylor expansion of fixpoint is sufficient to verify this point.

498 ■ The projections rules are obtained by a straightforward application of the definitions. ◀

499

500 5 Conclusions

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

507 ■ The calculus can be extended in order to define inductive and coinductive datatypes.
 508 Integers, for instance, could be defined by adding to our syntax $()$: $\underline{0} = \iota_1()$, $\underline{k+1} = \iota_2(\underline{k})$,
 509 and all integers defined in this way have the type $\iota = (1 \oplus \iota)$. The successor \mathbf{suc} can then
 510 be defined as the second injection. Then, if x has no free occurrence in N_1 , the term
 511 $\mathbf{case}(M, x \cdot N_1, y \cdot N_2)$ is an adequate encoding of an “if zero” conditional $\mathbf{If}(M, N_1, y \cdot N_2)$
 512 (where the value to which M evaluates is passed to the following computation).

513 The coinductive datatype of streams can also be defined: let A be a positive type,
 514 $S_A = !(A \otimes S_A)$ is the type of lazy streams of type A (the tail of the stream being always
 515 encapsulated in an exponential, the evaluation is postponed). We can construct a term
 516 of type $S_A \multimap \iota \multimap A$ which computes the k -th element of a stream:

$$517 \quad \mathbf{fix}_f (\lambda x \lambda y (\mathbf{If}(y, \pi_1(\mathbf{der}(x)), z \cdot \langle \mathbf{der}(f) \rangle \pi_2 \langle \mathbf{der}(x) \rangle z)))$$

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

$$519 \quad \mathbf{fix}_f (\lambda g (\mathbf{der}(g)\underline{0}, \langle \mathbf{der}(f) \rangle (\lambda x \langle \mathbf{der}(g) \rangle \mathbf{suc}(x))^!))$$

520 which builds a stream by applying inductively a function to an integer. There are other
 521 classical constructions, such as lists, that can be constructed with these ingredients. For
 522 a more detailed presentation, see Ehrhard and Tasson’s work [14]. We have good hope
 523 that this kind of extensions can be incorporated in our resource driven-constructions.

524 ■ Extend our constructions in a probabilistic setting, to fit with existing quantitative
 525 models like probabilistic coherence spaces. Indeed Lemma 10, which is crucial to define
 526 reduction on quantitative Taylor expansion, strongly relies on the uniformity of the
 527 calculus, *i.e.* we use the fact that all resource terms appearing in the Taylor expansion of
 528 a Call-By-Push-Value term have the same *shape* (there is a correspondance between their
 529 syntactic trees). The extension seems highly non trivial. But, Dal Lago and Leventis’
 530 recent work [19] might be a starting point.

531 — References

- 532 1 Samson Abramsky and Guy McCusker. Call-by-value games. In *CSL*, volume 1414 of *Lecture*
533 *Notes in Computer Science*, pages 1–17. Springer, 1997.
- 534 2 J. Chouquet. Taylor expansion, finiteness and strategies. In *MFPS 2019*, 2019.
- 535 3 Jules Chouquet and Lionel Vaux Auclair. An application of parallel cut elimination in unit-free
536 multiplicative linear logic to the taylor expansion of proof nets. In *CSL*, volume 119 of *LIPICs*,
537 pages 15:1–15:17. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.
- 538 4 Pierre-Louis Curien, Marcelo P. Fiore, and Guillaume Munch-Maccagnoni. A theory of effects
539 and resources: adjunction models and polarised calculi. In *POPL 2016, St. Petersburg, FL,*
540 *USA, January 20 - 22, 2016*, 2016.
- 541 5 V. Danos and T. Ehrhard. Probabilistic coherence spaces as a model of higher-order probabil-
542 istic computation. *Inf. Comput.*, 209(6):966–991, 2011.
- 543 6 Jeff Egger, Rasmus Ejlers Møgelberg, and Alex K. Simpson. The enriched effect calculus:
544 syntax and semantics. *J. Log. Comput.*, 24:615–654, 2014.
- 545 7 T. Ehrhard. On köthe sequence spaces and linear logic. *Mathematical Structures in Computer*
546 *Science*, 12(5):579–623, 2002.
- 547 8 T. Ehrhard. Finiteness spaces. *Mathematical Structures in Computer Science*, 2005.
- 548 9 T. Ehrhard and L. Regnier. The differential lambda-calculus. *Theor. Comput. Sci.*, 2003.
- 549 10 T. Ehrhard and L. Regnier. Uniformity and the Taylor expansion of ordinary lambda-terms.
550 *Theor. Comput. Sci.*, 403(2-3):347–372, 2008.
- 551 11 Thomas Ehrhard. Collapsing non-idempotent intersection types. In *CSL*, volume 16 of *LIPICs*,
552 pages 259–273. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012.
- 553 12 Thomas Ehrhard. Call-by-push-value from a linear logic point of view. In *ESOP*, volume 9632
554 of *Lecture Notes in Computer Science*, pages 202–228. Springer, 2016.
- 555 13 Thomas Ehrhard and Giulio Guerrieri. The bang calculus: an untyped lambda-calculus
556 generalizing call-by-name and call-by-value. In *PPDP*, pages 174–187. ACM, 2016.
- 557 14 Thomas Ehrhard and Christine Tasson. Probabilistic call by push value. *Logical Methods in*
558 *Computer Science*, 15(1), 2019.
- 559 15 J.-Y. Girard. Linear logic. *Theor. Comput. Sci.*, 50:1–102, 1987.
- 560 16 Giulio Guerrieri and Giulio Manzonetto. The bang calculus and the two girard’s translations.
561 *CoRR*, abs/1904.06845, 2019. URL: <http://arxiv.org/abs/1904.06845>, arXiv:1904.06845.
- 562 17 E. Kerinec, G. Manzonetto, and M. Pagani. Revisiting Call-by-value Böhm trees in light of
563 their Taylor expansion. *CoRR*, abs/1809.02659, 2018. arXiv:1809.02659.
- 564 18 M. Kerjean and C. Tasson. Mackey-complete spaces and power series - a topological model of
565 differential linear logic. *Mathematical Structures in Computer Science*, 28(4):472–507, 2018.
- 566 19 Ugo Dal Lago and Thomas Leventis. On the Taylor expansion of probabilistic λ -terms. *CoRR*,
567 abs/1904.09650, 2019. URL: <http://arxiv.org/abs/1904.09650>, arXiv:1904.09650.
- 568 20 Olivier Laurent and Laurent Regnier. About Translations of Classical Logic into Polarized
569 Linear Logic. In *18th IEEE Symposium on Logic in Computer Science (LICS 2003), 22-25*
570 *June 2003, Ottawa, Canada, Proceedings*, pages 11–20. IEEE Computer Society, 2003. URL:
571 <http://dx.doi.org/10.1109/LICS.2003.1210040>, doi:10.1109/LICS.2003.1210040.
- 572 21 P. B. Levy. Call-by-push-value: Decomposing call-by-value and call-by-name. *Higher-Order*
573 *and Symbolic Computation*, 19(4):377–414, 2006.
- 574 22 Michael Marz, Alexander Rohr, and Thomas Streicher. Full Abstraction and Universality via
575 Realisability. In *14th Annual IEEE Symposium on Logic in Computer Science, Trento, Italy,*
576 *July 2-5, 1999*, pages 174–182. IEEE Computer Society, 1999. URL: [http://dx.doi.org/10.](http://dx.doi.org/10.1109/LICS.1999.782612)
577 [1109/LICS.1999.782612](http://dx.doi.org/10.1109/LICS.1999.782612), doi:10.1109/LICS.1999.782612.
- 578 23 Damiano Mazza. An infinitary affine lambda-calculus isomorphic to the full lambda-calculus.
579 In *LICS*, pages 471–480. IEEE Computer Society, 2012.
- 580 24 Damiano Mazza, Luc Pellissier, and Pierre Vial. Polyadic approximations, fibrations and
581 intersection types. *PACMPL*, 2(POPL):6:1–6:28, 2018.

- 582 25 P-A. Melliès. Categorical semantics of linear logic. In *In: Interactive Models of Computation*
583 *and Program Behaviour, Panoramas et Synthèses 27, Société Mathématique de France 1–196,*
584 *2009.*
- 585 26 M. Pagani, C. Tasson, and L. Vaux. Strong normalizability as a finiteness structure via the
586 Taylor expansion of lambda-terms. In *FOSSACS 2016*, pages 408–423, 2016.
- 587 27 L. Vaux. Taylor expansion, β -reduction and normalization. In *26th EACSL Annual Conference*
588 *on Computer Science Logic, CSL 2017, Stockholm, Sweden*, pages 39:1–39:16, 2017.

