Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 347 (2019) 65–85

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Taylor Expansion, Finiteness and Strategies [1]

Jules Chouquet [2]

IRIF UMR 8243
Université Paris Diderot, Sorbonne Paris Cité, CNRS, France

Abstract

We examine some recent methods introduced to extend Ehrhard and Regnier's result on Taylor expansion: infinite linear combinations of approximants of a lambda-term can be normalized while keeping all coefficients finite. The methods considered allow to extend this result to non-uniform calculi; we show that when focusing on precise reduction strategies, such as Call-By-Value, Call-By-Need, PCF or variants of Call-By-Push-Value, the extension of Ehrhard and Regnier's finiteness result can hold or not, depending on the structure of the original calculus.
In particular, we introduce a resource calculus for Call-By-Need, and show that the finiteness result about its Taylor expansion can be derived from our Call-By-Value considerations. We also introduce a resource calculus for a presentation of PCF with an explicit fixpoint construction, and show how it interferes with the finiteness result. We examine then Ehrhard and Guerrieri's Bang Calculus which enjoys some Call-By-Push-Value features in a slightly different presentation.

Keywords: Lambda calculus, Call-By-Value, Bang Calculus, Call-By-Need, Linear logic, Taylor expansion, PCF

1 Introduction

The past decade saw the appearance a revival of Girard's quantitative semantics of λ-calculus, with proposals of new models, and extensions of the existing results to various calculi: other operational semantics (Call-by-Value, PCF, Call-By-Push-Value) or non-deterministic extensions (probabilistic [11,14], algebraic [28]). A crucial feature of quantitative interpretation is the analyticity of the functions denoting the λ-terms. Girard's original model of normal functors [15] used set-valued power series representing analytic maps between modules. Linear logic's birth is presented as a result of this study, and still plays a central role in quantitative semantics' recent works [3,27].

[1] This work is funded by the french ANR project RAPIDO (ANR-14-CE25-0007). The author thanks the anonymous reviewers for their useful comments, Lionel Vaux-Auclair and Michele Pagani for their advices, and is very grateful to Christine Tasson for her help and support.

[2] Email: `Jules.Chouquet@irif.fr` URL: `https://www.irif.fr/~chouquet`

https://doi.org/10.1016/j.entcs.2019.09.005

The models that have since been proposed in that direction have permitted the study of precise operational and quantitative properties of the calculus, such as execution time [6], or probabilities [11,14]. The relation between such aspects of the calculus and the power series-oriented semantics has been, in particular, exhibited clearly with the introduction of an interface between the syntax and the semantics: following models like Ehrhard's finiteness spaces [7], Ehrhard and Regnier constructed a variant of λ-calculus which corresponds to the multilinear approximations of the analytic maps in the models. This variant, called *resource calculus*, and coming from Boudol's calculi with multiplicities [2], is the multilinear fragment of differential λ-calculus [12], and comports multilinear terms — where "linear" has to be understood in the computational sense: the available resources are used exactly once during the computation. For example, the λ-term $(\lambda x x x)y$, which calls for the duplication of the argument, will be approximated by the resource term $\langle\lambda x\langle x\rangle[x]\rangle[y,y]$, where $\langle m\rangle\overline{n}$ stands for the multilinear application of a term to a multiset of terms: if the term in function position calls for k arguments (in our example, $\lambda x\langle x\rangle[x]$ calls for 2 arguments), then the multiset in argument position must contain k terms (in our example, $[y,y]$, for $k=2$), otherwise the reduction leads to a nullary sum of terms.

This resource calculus is said to be an interface between the original calculus and the model because it allows to mimic the identities of quantitative semantics through Taylor expansion construction, which is the subject of the present paper. Taylor expansion is a syntactic analogue to the well-known Taylor formula, and consists also in the correspondence between a non-linear object (in the model: an analytic function; in the syntax: a pure λ term) and a sum (generally infinite) of multilinear approximants (in the model: multilinear maps; in the syntax: resource terms). In ordinary λ-calculus, the key case of the definition of Taylor expansion concerns essentially the application, since it contains the non-linear part of λ-calculus, and can be presented as follows: $\mathcal{T}(MN) = \sum_{k\in\mathbf{N}} \frac{1}{k!}\langle\mathcal{T}(M)\rangle\mathcal{T}(N)^k$ where $\mathcal{T}(N)^k$ is a multiset of k copies of $\mathcal{T}(N)$. The construction is also linear in two ways: all summands are resource terms, and syntactic constructs commute with sums. A model is compatible with Taylor expansion if the interpretation of M is the same as the interpretation of $\mathcal{T}(M)$. This is a property shared by structures like the weighted relational model [21], finiteness spaces [7], probabilistic coherent spaces [5], convenient vector spaces [18] in which most of recent quantitative studies have been produced. However, this treatment of λ-calculus interpretation brings a difficulty: Taylor expansion is a potentially infinite weighted sum, and there is no guarantee that coefficients remain finite under reduction. Indeed, the notion of reduction we put on resource terms has to allow the simultaneous calculus of an unbounded number of redexes in order to simulate β-reduction.

Consider for instance $MN \to_\beta MN'$. The resource approximants of this term are of the shape $\langle m\rangle[n_1,\ldots,n_k]$ (with m being an approximant of M, and the n_i being approximants of N), and we have to reduce all the n_i in one step to reach a term $\langle m\rangle[n'_1,\ldots,n'_k]$ which approximates MN'. Consider now a family $(m_i)_{i\in\mathbf{N}}$ such that for all i, m_i is a resource term of the shape $\langle\lambda x[x]\rangle[\langle\lambda x[x]\rangle[\ldots\langle\lambda x[x]\rangle[y]]]$, with i

applications of the identity to the variable y. All these terms reduce in parallel to y, and if we define a parallel reduction $\Rightarrow$ over infinite linear combinations of terms, then we are led to let appear an infinite coefficient: $\sum_{i\in\mathbf{N}} m_i \Rightarrow \infty \cdot y$ which is not always defined in the models we consider (when the semimodule we work on is not built on a complete semiring, as is the case for example in the weighted relational model [21], all sums are not guaranteed to converge). Moreover, keeping coefficients finite is important as it implies, for instance, that the calculation of a value always terminates. A way to ensure that infinite coefficient do not appear is to show that each term has a finite number of antireducts in the combinations we consider, which is not the case in the above example. The first result concerning this issue is Ehrhard and Regnier's one: they proved that when dealing with pure non-uniform λ-calculus, coefficients stay finite under reduction and that normalization and Taylor expansion commute [13]. The problem of extending this finiteness result to other calculi has been very studied recently. Indeed, solving it is a necessary step to ensure that one can provide a quantitative interpretation of the calculus, consistent with the algebraic behaviour of the models mentioned above. Ehrhard extended that result to a non-uniform variant of System F, putting a finiteness structure on the resource terms [8]. This methods then was extended by Pagani, Tasson and Vaux-Auclair to all strongly normalizable terms in a calculus endowed with non-deterministic sums [23]. Then, Vaux-Auclair brought an even more general approach, dealing with algebraic λ-calculus, and extending the argument to weakly-normalizable terms [28,29]. We used with Vaux-Auclair the method of that latter paper to study Taylor expansion of linear logic proof nets in a comparable way [4].

The key result concerning Taylor expansion, namely, the definability of a parallel reduction on it, demands technical proofs, which differ depending on the structure of the original calculus. We propose here the study of different calculi, with different operational semantics: Call-By-Value, Bang Calculus [10], Call-By-Need [1], and PCF [25].

Our present contribution is to show that in those various calculi, the finiteness results hold, up to a convenient definition of the resource calculi and to the choice of arguments depending on those calculi's precise features. Call-By-Value Taylor expansion has already been defined by Ehrhard [9], and shown to be compatible with Böhm trees by Kerinec, Manzonetto and Pagani [17], but in a qualitative way (*i.e.* coefficients are not considered). We show in Section 2.1 that a parallel reduction is definable on Taylor expansion with coefficients, and this remains true if we provide an algebraic extension of Call-By-Value calculus, because the method used by Vaux-Auclair can be adapted to this setting.

We also define in Section 2.2 a resource calculus adapted to Call-By-Need reduction, and observe that the specificities of its operational semantics implies that its Taylor expansion consists in the same set of resource terms as Call-By-Value one.

For the two calculi of Section 2 and for the Bang calculus, the finiteness result is proven thanks to a combinatorial study of the parallel reduction and the size of resource terms in the Taylor expansion, following Vaux-Auclair's method [29]. The key result is then about cardinalities of sets, size of terms, and concerns the sets underlying to Taylor expansion. This implies in particular that uniformity is not a necessary property for the proofs to be valid, and that algebraic extensions of the calculus would not interfere with the arguments.

The situation is different in presence of an explicit fixpoint. We focus on PCF calculus to study the consequences of endowing the syntax with a fixpoint operator in a typed setting. We consider a presentation of PCF extract from Ehrhard, Pagani and Tasson's work [11], and we are motivated by the quantitative semantics they provide for it. The presence of the fixpoint is here crucial, since the argument used before cannot apply anymore, for a reason linked to the fact that there is no way to give a finiteness structure interpreting terms with fixpoints (see *e.g.* Tasson and Vaux-Auclair for a detailed examination of this point [26]). In order to show that Taylor expansion and parallel reduction are still definable in this setting, we have to provide an argument relying on a coherence relation between resource terms following Ehrhard and Regnier [13]. It is sufficient to prove the finiteness result in a uniform setting, but this reveals the fact that, when dealing with both an explicit fixpoint and non-uniformity (such as non-deterministic choice, probabilities), the finiteness result becomes false: we can build a term whose Taylor expansion contains an infinity of resource terms reducing at the same time.

We then consider in Section 4 Ehrhard and Guerrieri's Bang Calculus [10] (see also Guerrieri and Manzonetto [16]), which is a linear logic-inspired calculus subsuming Call-By-Name and Call-By-Value disciplines, and for which Taylor expansion has been defined in the introductory paper [10]. The Bang Calculus can be seen as an untyped variant of Levy's Call-By-Push-Value. We discuss in conclusion the distinctions and difficulties appearing when switching from the former to the latter.

Terminology

We denote as $\mathbf{N}$ the set of positive integers, whose elements will often be written $k, l, \ldots$. For $k \in \mathbf{N}$, $\mathfrak{S}_k$ is the set of permutations of $\{1, \ldots, k\}$. We use the notation $[m_1, \ldots, m_k]$ for the multisets, and $[m, \ldots, m]_k$, or just $[m]_k$ for the multiset with k occurrences of the same term m. We use the standard additive notation $[m_1, \ldots, m_k] + [m_{k+1}, \ldots, m_{k+l}]$ to represent the multiset $[m_1, \ldots, m_{k+l}]$. Multisets $[m_1, \ldots, m_k]$ might be denoted $\overline{m}$ or $\overrightarrow{m}$.

We consider $\mathsf{deg}_x(m)$ the degree of x in m as the number of free occurrences of x in m, for any term m in any of the calculi we introduce. We might represent distinct occurrences of x by $x_1, \ldots, x_k$, especially when dealing with resource reductions.

For any set X, we consider the multiset construction $X^! = \{[x_1, \ldots, x_k] \mid k \in \mathbf{N}, x_i \in X\}$. For any lambda term or resource term μ, its size $\#\mu$ is defined in the usual way : $\#x = 1, \#\lambda x\mu = \#\mu + 1, \#(\mu\mu') = \#\mu + \#\mu' + 1, \#[\mu_1, \ldots, \mu_k] = \sum_{i\in\{1,\ldots,k\}} \#\mu_i$.

In the following, the lower script $()_{\mathcal{V}}$ will often stand for Call-By-Value calculus, $()_{\mathsf{b}}$ for Bang calculus, and $()_{\mathsf{need}}$ for Call-By-Need. $()_{\mathsf{r}\mathcal{V}}$, $()_{\mathsf{rb}}$ and $()_{\mathsf{rneed}}$ refer to corresponding resource calculi.

2 Call-By-Value and Call-By-Need

2.1 Call-By-Value

Kerinec, Manzonetto and Pagani show that defining a natural coherence relation between Call-By-Value resource terms introduced by Ehrhard [9] leads to consider Taylor expansion as maximal cliques for this relation [17] (Prop. 3.16). With that result together with Ehrhard and Regnier's methods [13], we can expect that finiteness of antireducts in Taylor expansion is provable, and more precisely that for a given resource term m, and a given $\Lambda_{\mathcal{V}}$-term N, $\{n \rightrightarrows_{\mathsf{r}\mathcal{V}} m \mid n \in \mathcal{T}_{\mathcal{V}}(N)\}$ contains always at most one element, where $\Lambda_{\mathcal{V}}$ represents the Call-By-Value calculus, $\mathcal{T}_{\mathcal{V}}$ its Taylor expansion, and $\rightrightarrows_{\mathsf{r}\mathcal{V}}$ is a parallel Call-By-Value resource reduction.

But, one of our motivations is to endow $\Lambda_{\mathcal{V}}$ with non-deterministic sums, or with coefficients, so that uniformity will be lost. In such a setting, the coherence method no longer applies since it relies precisely on uniformity. Similarly to Vaux-Auclair [29], we use a technique to bound the number of antireducts of a given term in Taylor expansion, independently of uniformity hypothesis. This opens the path to the study of Call-By-Value Taylor expansion as power series with coefficients, since it ensures that coefficients remain finite under parallel reduction of combinations of terms. Moreover, Taylor normal form is always well-defined in such a setting.

Definition 2.1 [Call-By-Value calculus $\Lambda_{\mathcal{V}}$]

$$V ::= x \mid \lambda x M \qquad \Lambda_{\mathcal{V}} : M, N ::= V \mid MN$$

The reduction rule is the following : $(\lambda x M)V \to_{\beta_{\mathcal{V}}} M[V/x]$, and closed by abstraction and application contexts.

Definition 2.2 [Call-By-Value resource calculus $\Delta_{\mathcal{V}}$ [9]]

$$\Delta_{\mathcal{V}} : m, n ::= [x_1, \dots, x_k] \mid [\lambda x_1 m_1, \dots, \lambda x_k m_k] \mid mn$$

The reduction rule is the following[3], closed by application and abstraction contexts, and where $x_1, \dots, x_l$ represent l distinct occurrences of x:

$$[\lambda x m][n_1, \dots, n_l] \to_{\mathsf{r}\mathcal{V}} m_1[n_{f(1)}/x_1, \dots, n_{f(l)}/x_l] \text{ if } l = \mathsf{deg}_x(m_1),\ f \in \mathfrak{S}_l$$

The term $[\lambda y_1 m_1, \dots, \lambda y_k m_k][n_1, \dots, n_l]$ has no reduct if $k \neq 1$.

Definition 2.3 [Call-by-Value Taylor expansion [9]] Taylor expansion is defined as a function from $\Lambda_{\mathcal{V}}$ to sets of terms of $\Delta_{\mathcal{V}}$:

- $\mathcal{T}_{\mathcal{V}}(\lambda x M) = \{[\lambda x m_1, \dots, \lambda x m_n]; n \in \mathbf{N}, m_i \in \mathcal{T}_{\mathcal{V}}(M)\}$
- $\mathcal{T}_{\mathcal{V}}(x) = \{[x, \dots, x]_k \mid k \in N\}$

[3] We differ from the usual presentations, since instead of having a term reducing in a finite sum of resource terms, our reduction is non-deterministic and corresponds to a relation between a term and the support of its usual reduct.

- $\mathcal{T}_{\mathcal{V}}(MN) = \{mn \mid m \in \mathcal{T}_{\mathcal{V}}(M), n \in \mathcal{T}_{\mathcal{V}}(N)\}$

The method we use here consists in setting a measure on resource terms that is bounded for all terms belonging in Taylor expansion of some $\Lambda_{\mathcal{V}}$-term. This measure is defined below, and corresponds to the number of nested applications in a term. We show that this notion of depth will be sufficient to establish that for any integer k, resource term m, and $\Lambda_{\mathcal{V}}$-term M, $\{m' \rightrightarrows_{\mathsf{r}\mathcal{V}} m \mid \mathbf{ApD}(m') \leq k\}$ is finite. Here, $\rightrightarrows_{\mathsf{r}\mathcal{V}}$ is the parallel Call-By-Value resource reduction which has to be introduced for the following reason: it permits to simulate $\to_{\beta_{\mathcal{V}}}$ in resource calculus (in particular, it reduces all terms of a multiset in one step), it is confluent, and allows to keep a bound on applicative depth so as to apply our argument to iterated reduction, as we shall see.

Definition 2.4 [Applicative depth] For any term m of $\Delta_{\mathcal{V}}$, we define its applicative depth $\mathbf{ApD}(m)$ as follows : $\mathbf{ApD}([x_1, \dots, x_k]) = 0$, $\mathbf{ApD}([\lambda x_1 n_1, \dots \lambda x_n n_k]) = \max\{\mathbf{ApD}(n_i) \mid i \in \{1, \dots, k\}\}$, $\mathbf{ApD}(mm') = \max\{\mathbf{ApD}(m), \mathbf{ApD}(m')\} + 1$. We do the same for terms of $\Lambda_{\mathcal{V}}$: $\mathbf{ApD}(x) = 0$, $\mathbf{ApD}(\lambda x M) = \mathbf{ApD}(M)$, and $\mathbf{ApD}(NN') = \max\{\mathbf{ApD}(N), \mathbf{ApD}(N')\} + 1$.

Lemma 2.5 *Let M be a term of $\Lambda_{\mathcal{V}}$. Then for all $m \in \mathcal{T}_{\mathcal{V}}(M), \mathbf{ApD}(m) \leq \mathbf{ApD}(M)$.*

Proof. By induction on M:

- If $M = x$, then $m = [x, \dots, x]_k$, and $\mathbf{ApD}(m) = \mathbf{ApD}(M) = 0$.
- If $M = \lambda x N$, then $m = [\lambda x n_1, \dots, \lambda x n_k]$, and $\mathbf{ApD}(m) = \max\{\mathbf{ApD}(n_i) \mid i \in \{1, \dots, k\}\}$. By induction hypothesis, for all i, $\mathbf{ApD}(n_i) \leq \mathbf{ApD}(N)$. Then $\mathbf{ApD}(m) \leq \mathbf{ApD}(M)$.
- If $M = NN'$, then $m = nn'$, and $\mathbf{ApD}(m) = \max\{\mathbf{ApD}(n), \mathbf{ApD}(n')\} + 1$. By induction, $\mathbf{ApD}(n) \leq \mathbf{ApD}(N)$ and $\mathbf{ApD}(n') \leq \mathbf{ApD}(N')$, then $\mathbf{ApD}(m) \leq \mathbf{ApD}(M)$.

□

As explained in introduction, the notion of reduction we focus on must be a parallel one. Indeed, it is straightforward that for resource terms in Taylor expansion of some Call-By-Value term, the reduction $\to_{\mathsf{r}\mathcal{V}}$ is not sufficient to simulate $\to_{\beta_{\mathcal{V}}}$. Indeed, if we consider $M \to_{\beta_{\mathcal{V}}} M'$, and $n = [\lambda x_1 m_1, \dots, \lambda x_k m_k] \in \mathcal{T}_{\mathcal{V}}(\lambda x M)$, if $k > 1$, there is generally[4] no $n' \in \Delta_{\mathcal{V}}$ such that $n \to_{\mathsf{r}\mathcal{V}} n'$ and $n' \in \mathcal{T}_{\mathcal{V}}(\lambda x M')$. In order to examine resource dynamics that correspond to $\to_{\beta_{\mathcal{V}}}$ reduction, and thus to semantics identities, we must introduce the following parallel resource reduction:

Definition 2.6 [Parallel reduction of resource terms $\rightrightarrows_{\mathsf{r}\mathcal{V}}$]

- $m \rightrightarrows_{\mathsf{r}\mathcal{V}} m$ for all $m \in m$
- If $m_i \rightrightarrows_{\mathsf{r}\mathcal{V}} m'_i$ for all $i \in \{1, \dots, k\}$, then $[m_1, \dots, m_k] \rightrightarrows_{\mathsf{r}\mathcal{V}} [m'_1, \dots m'_k]$

[4] It can happen that the m_i are already in $\mathcal{T}_{\mathcal{V}}(M')$, for example if M, M' are values and the m_i are empty bags.

- If $m \rightrightarrows_{\mathsf{r}\mathcal{V}} m'$ and $n \rightrightarrows_{\mathsf{r}\mathcal{V}} n'$, $mn \rightrightarrows_{\mathsf{r}\mathcal{V}} m'n'$.
- If $m_i \rightrightarrows_{\mathsf{r}\mathcal{V}} m'_i$ for all $i \in \{1, \dots, k\}$, $n \rightrightarrows_{\mathsf{r}\mathcal{V}} n'$, and $k = \mathsf{deg}_x(n)$ then we set $[\lambda xn][m_1, \dots, m_k] \rightrightarrows_{\mathsf{r}\mathcal{V}} n'[m'_{f(1)}/x_1, \dots, m'_{f(k)}/x_k]$ for any permutation $f \in \mathfrak{S}_k$.

Note that $\rightrightarrows_{\mathsf{r}\mathcal{V}}$ is strictly included in the reflexive transitive closure of $\to_{\mathsf{r}\mathcal{V}}$. The redexes created during the reduction are not reduced by $\rightrightarrows_{\mathsf{r}\mathcal{V}}$. For example, one can check that for $m = [\lambda x[x][z]][\lambda y[y]]$, $m \to^*_{\mathsf{r}\mathcal{V}} [z]$, but $m \rightrightarrows_{\mathsf{r}\mathcal{V}} [\lambda y[y]][z]$ and cannot be reduced further in the same parallel step.

The following lemma ensures that the applicative depth of resource terms allows to bound the loss of size during parallel reduction, and that this measure is still bounded after one reduction step: we will then be able to extend the argument to iterated reduction.

Lemma 2.7 *There exist non decreasing functions $\varphi : (\mathbf{N} \times \mathbf{N}) \to \mathbf{N}$ and $\psi : \mathbf{N} \to \mathbf{N}$ such that for any $m, n \in \Delta_{\mathcal{V}}$, if $m \rightrightarrows_{\mathsf{r}\mathcal{V}} n$, then $\#m \leq \varphi(\#n, \mathbf{ApD}(m))$. Moreover, $\mathbf{ApD}(n) \leq \psi(\mathbf{ApD}(m))$.*

The proof can be obtained as an adaptation of Vaux-Auclair's result [29] (Lemma 26, $\varphi(k, l) = 4^l k$ and $\psi(k) = 2^k k$), to a Call-By-Value setting, as we sketch below:

The idea here is to prevent an unbounded collapse of the size under parallel reduction. A critical example of what we call a collapse (because the size reduces drastically, since an unbounded number of variables disappear) is the set: $X = \{[\lambda x_1 x_1][[\lambda x_2 x_2][\dots[\lambda x_k x_k][z]]\dots]]] \mid k \in \mathbf{N}\}$. We observe that for all $m \in X$, $m \rightrightarrows_{\mathsf{r}\mathcal{V}} [z]$. There is no way to give a bound to the size of antireducts of $[z]$ in X, and we can immediately see that the applicative depth is not bounded in X either.

Vaux-Auclair observed that the collapse depends on the depth of antireducts, and that the cardinality of multisets does not interfere with it. Then, since Taylor expansion consists of sets of terms of unbounded multisets, but of bounded depth, it excludes subsets of terms like X in the above example.

The adaptation of this result to Call-By-Value is then a corollary of Lemma 2.5 and relies on the structure of the sets $\mathcal{T}_{\mathcal{V}}(M)$, having the necessary properties for Lemma 2.7 to be valid. We conclude by observing that $\rightrightarrows_{\mathsf{r}\mathcal{V}}$ has the same combinatorial properties that the unrestricted parallel resource reduction of Vaux-Auclair's works, since it is strictly included in it and behaves in the same way: it is a particular case of Ehrhard and Regnier's resource reduction.

Corollary 2.8 *Let $n \in \Delta_{\mathcal{V}}$, $k \in \mathbf{N}$ and $M \in \Lambda_{\mathcal{V}}$. $\{m \in \mathcal{T}_{\mathcal{V}}(M) \mid m \rightrightarrows^k_{\mathsf{r}\mathcal{V}} n\}$ is finite.*

Proof. It is sufficient to observe that for any given $k \in \mathbf{N}$, $\{m \in \Delta_{\mathcal{V}} \mid \#m \leq k\}$ is finite. Then, by Lemma 2.5, terms in $\{m \in \mathcal{T}_{\mathcal{V}}(M) \mid m \rightrightarrows_{\mathsf{r}\mathcal{V}} n\}$ have a bounded applicative depth. We then show by induction on k that $\{m \rightrightarrows^k_{\mathsf{r}\mathcal{V}} n \mid \mathbf{ApD}(m) \leq l\}$ is finite for any l. By Lemma 2.7, we conclude for $k = 1$. Then, the result can be iterated since if $m \rightrightarrows^{k+1}_{\mathsf{r}\mathcal{V}} n$, $m \rightrightarrows_{\mathsf{r}\mathcal{V}} m' \rightrightarrows^k_{\mathsf{r}\mathcal{V}} n$. By Lemma 2.7, $\mathbf{ApD}(m') \leq \psi(\mathbf{ApD}(m))$, we apply induction hypothesis and conclude for all $k \in \mathbf{N}$. □

Let us consider now a quantitative version of Taylor expansion as a power series with coefficients $\mathcal{T}^q_{\mathcal{V}}(M) = \sum_{m\in\mathcal{T}_{\mathcal{V}}(M)} \left(\mathcal{T}^q_{\mathcal{V}}(M)\right)_m \cdot m$, where $\left(\mathcal{T}^q_{\mathcal{V}}(M)\right)_m$ represents the coefficient of m in $\mathcal{T}^q_{\mathcal{V}}(M)$ taken in some semiring $\mathbf{S}$. Motivated by the quantitative semantics identities which identify the denotation of M with its Taylor normal form, we want to define this object, but we have first to tackle a difficulty: the combination $\mathfrak{tnf}(M) = \sum_{m\in\mathcal{T}_{\mathcal{V}}(M)} \left(\mathcal{T}^q_{\mathcal{V}}(M)\right)_m \cdot \mathsf{nf}(m)$ (where nf is the normal form operator, always defined for resource terms, since resource reduction induces a strict decrease of the size of terms) might contain terms with infinite coefficients. It is the case if $\{m \in \mathcal{T}_{\mathcal{V}}(M) \mid (\mathcal{T}^q_{\mathcal{V}}(M)i)_m \neq 0, m \rightrightarrows^*_{\mathsf{r}\mathcal{V}} \mathsf{nf}(m)\}$ is infinite.

Let us define a reduction relation between infinite linear combination of terms denoted as: $\sum_{i\in I} a_i m_i \Rightarrow \sum_{i\in I} a_i n_i$ if $m_i \rightrightarrows_{\mathsf{r}\mathcal{V}} n_i$ for all $i \in I$, and if for all n_i, $\{j \in I \mid m_j \rightrightarrows_{\mathsf{r}\mathcal{V}} n_i, a_j \neq 0\}$ is finite. Corollary 2.8 ensures that the reduction $\mathcal{T}^q_{\mathcal{V}}(M) \Rightarrow \xi$ is well-defined for any combination ξ. In particular, the simulation of a reduction $M \to_{\beta_{\mathcal{V}}} N$ through a reduction $\mathcal{T}^q_{\mathcal{V}}(M) \Rightarrow \mathcal{T}^q_{\mathcal{V}}(N)$ is well-defined. If M is normalizable, and $M \to^k_{\beta_{\mathcal{V}}} \mathsf{nf}(M)$, we can also deduce immediately $\mathcal{T}^q_{\mathcal{V}}(M) \Rightarrow^k \mathfrak{tnf}(M)$ and that for all $m \in |\mathfrak{tnf}(M)|$ ($|\mathfrak{tnf}(M)|$being the support of $\mathfrak{tnf}(M)$), we have $(\mathfrak{tnf}(M))_m$ finite.

As a further study in that direction, we shall mention Kerinec, Manzonetto and Pagani's works [17], that defines Call-By-Value Böhm trees and establish $BT(\mathcal{T}_{\mathcal{V}}(M)) = \mathfrak{tnf}(M)$ in a qualitative setting. We leave as a future work the quantitative extension of this equation, but we have set the possibility of dealing with finite coefficients in the reduction of Call-By-Value Taylor expansion.

2.2 Call-By-Need

Our resource-oriented study of strategies of reductions calls to observe the particular setting of Call-By-Need, which is a strategy that optimises the number of reduction steps: in Call-By-Need, the reduction does not proceed to useless or inefficient reduction as in Call-By-Value *and* Call-By-Name. Assume $M \to M'$: useless reduction would be the Call-By-Value evaluation of M in $N = (\lambda yx)M$, because $N \to_\beta y$ and $M \to_{\beta_{\mathcal{V}}} (\lambda yx)M' \to_{\beta_{\mathcal{V}}} y$: Call-By-Name strategy proceeds to the erasure of the argument, which is optimal regarding to the number of reduction steps. An inefficient reduction would be the Call-By-Name reduction of the most external redex in $N' = (\lambda xxx)M$ before the evaluation of M, because $N' \to_\beta MM \to_\beta M'M \to_\beta M'M'$ while $N \to_{\beta_{\mathcal{V}}} (\lambda xxx)M' \to_{\beta_{\mathcal{V}}} M'M'$: Call-By-Value is optimal in this case. Call-By-Need is optimal in both cases, since it prohibits the reduction of terms like $(\lambda yM)N$ if y has no free occurrences in M, or in general if N is doomed to be erased in a further reduction. And if not, Call-By-Need demands N to be reduced to a value before substituted in M. We give some intuitions sufficient to make some observations :

- We can define a resource calculus corresponding to Call-By-Need, in which the reduction rules enjoy the property of optimization explained in the above paragraph (neither useless nor inefficient reductions).
- Taylor expansion in this setting leads to the identical construction as the Call-By-

Value one. The distinction between the two calculi is not visible at this level, but when observing the evaluation steps and the distinct normal forms [5].

We introduce a resource calculus able to simulate Call-By-Need reduction: our terms are approximants of Call-By-Need syntax introduced by Pedrot and Saurin [24], which is shown to be equivalent to Ariola and Felleisen's original calculus [1]. We formalize this simulation in Appendix A, but the intuition is sufficient here to understand the remarks below.

Pedrot and Saurin's calculus presents Call-By-Need reduction in a quite concise syntax, since the constraints on the reduction are contained in the contexts and the marking of lambdas: a term λxt will be rewritten lxt when it is in the function position of a redex, or shall be reduced to one. Then, the contexts e are able to characterize reducible redexes in the following sense: a term of the shape $e[lxe[[x,\ldots,x]]]c[\overrightarrow{v}]$ is a term where the bag of variables x is at a place where it won't be erased. Then, the argument, if it is a value, can be substituted. We detail neither this dynamics nor the sharing properties, and refer to Pedrot and Saurin's paper for explanations of the calculus, because those properties at the resource level are similar: the reader can nonetheless ascertain that the definition of the contexts ensures that the reduction enjoys the optimization properties described in the above paragraph.

Definition 2.9 [Call-By-Need resource calculus Δ_{need}]

$$\begin{array}{ll} \text{terms}: \ t,u ::= [v_1,\ldots,v_k] \mid tu \mid lxt & \text{values}: \ v ::= x \mid \lambda xt \\ \text{contexts}: \ c ::= [] \mid c_1[lxc_2]t & e ::= [] \mid et \mid lxe \mid c[lxe_1[[x,\ldots,x]]e_2 \end{array}$$

Reduction rules:

$$c[[\lambda xt]] \rightarrow_{\mathsf{rneed}} c[lxt]$$
$$c[[\lambda x_1 t_1,\ldots,\lambda x_k t_k]]u \rightarrow_{\mathsf{rneed}} 0 \text{ if } k\neq 0$$
$$c_1[lxe[[x_1,\ldots,x_k]]c_2[[v_1,\ldots,v_m]] \rightarrow_{\mathsf{rneed}} 0 \text{ if } m<k$$
$$e[t] \rightarrow_{\mathsf{rneed}} e[t'] \text{ if } t \rightarrow_{\mathsf{rneed}} t'$$
$$c_1[lxe[[x_1,\ldots,x_k]]c_2[\overrightarrow{v}] \rightarrow_{\mathsf{rneed}} c_2[c_1[lxe[[v_1,\ldots,v_k]]]\overrightarrow{v}'] \text{ for all } [v_1,\ldots,v_k]+\overrightarrow{v}'=\overrightarrow{v}$$

From a resource point of view, which is ours, the first observation is that the terms above correspond to Call-By-Value ones. Indeed, if we omit the terms with marked lambdas lxt, the syntax is the same, and the distinction is all contained in operational semantics. In particular, if we define Taylor expansion of Call-By-Need following that presentation, we shall proceed exactly as we did for Call-By-Value. Corollary 2.8 would apply immediately when the convenient notion of parallel reduction is defined. It is not hard to examine the dynamics and to conclude that bounds about size and depth can be established for this calculus.

[5] We shall precise that Call-By-Need and Call-By-Value normal forms are distinct, since Call-By-Need is observationally equivalent to weak Call-By-Name (where no reductions are allowed under lambda's). This was established for instance, and in an elegant way by Kesner [19].

One can consider an approximation relation $\lhd$ between Call-By-Need terms and Δ_{need}, defined informally as follows : $m \lhd M$ if m has recursively *the same shape* than M (following notation used by Tsukada, Asada and Ong to define Taylor expansion [27]). See the appendix for a formal presentation. One can then consider some λ-term M, and convince oneself that $\{m \in \Delta_{\mathsf{need}} \mid m \lhd M\}$ and $\mathcal{T}_{\mathcal{V}}(M)$ are exactly the same sets. In particular, if we define a parallel, confluent extension of $\to_{\mathsf{rneed}}$, say $\rightrightarrows_{\mathsf{need}}$, the following proposition is easily derived from considerations of Section 2.1:

Proposition 2.10 *Let M be a term of Call-By-Need, and $n \in \Delta_{\mathsf{need}}$. $\{m \lhd M \mid m \rightrightarrows_{\mathsf{need}} n\}$ is finite.*

It is sufficient to observe that the reduction can be seen as a particular case of resource reduction, in the sense that we can bound the growing of applicative depth under parallel reduction, and that there is no arbitrary collapse during parallel reduction : the arguments differ from Call-By-Value in the management of the contexts and of the sharing properties of the calculus, but that do not intervene in the key properties we need.

We give below an example of Call-By-Need Taylor expansion to illustrate the interaction between Call-By-Need reduction and our resource constructions.

Example 2.11 Consider the following λ term : $M = (\lambda z(\lambda xy)(II)(zz))(II)$, where $I = \lambda xx$. We can already see that the most external abstraction calls for a duplication of the evaluation (II) if we stand in a Call-By-name discipline, while the subterm $(\lambda xy)(II)$ calls for a useless evaluation of (II) if we are in a Call-By-Value discipline. The Call-By-Need evaluation starts by reducing the rightmost (II) to I, and then, other reductions are forbidden, except a possible garbage collection rule that leads immediately to the term yI, which is the common normal form of M.

Let us consider Taylor expansion of M, and see how Call-By-Value and Call-By-Need differ in the reductions (we omit terms of the shape $[\lambda x_1, \ldots, \lambda x_n]t$, since they reduce to 0 in both calculi):

$$\mathcal{T}_{\mathcal{V}}(M) = \bigcup_{\vec{k} \in \mathbf{N}} \left\{ \left[\lambda z \left[\lambda x[y]_{k_y}\right] \left([\lambda x[x]_{k_x}][\lambda x'[x']_{k_{x'}}]_{k_{\lambda x'}}\right) \left([z]_{k_z}[z]_{k'_z}\right)\right] ([\lambda w[w]_{k_w}][\lambda v[v]_{k_v}]_{k_{\lambda v}}) \right\}$$

If we follow the reduction, we are led to observe that there is only one term of the above sum that reduces to the normal resource term $[y][\lambda v[v]]$ [6] : It is the point of $\mathcal{T}_{\mathcal{V}}(M)$ where $k_x, k_{\lambda'_x} = 0$, $k_z, k'_z, k_y = 1$, $k_w, k_{\lambda v}, k_v = 2$. Call-By-Need reduction is more permissive in this sense, since all cardinalities are acceptable for $k_{\lambda x'}$ and k_x : those terms are the arguments of the function subterm $\lambda x[y]$, and shall never be evaluated, but erased by the garbage collection, if considered. In Call-By-Value reduction, if $k_{\lambda x'}$ and k_x are not null, there will be an evaluation (a useless one, corresponding to the reduction of the intern II to I in M) before ending with the reduction to 0.

---

[6] Notice that this a single example taken among the infinite set $\mathcal{T}_{\mathcal{V}}(yI)$.

3 PCF

We announced in the introduction that the results we exhibited for Call-By-Value and Call-By-Need are valid even in the presence of an explicit fixpoint, but we are forced to change our proof method. Indeed, Ehrhard already observed that an explicit fixpoint prevents the calculus to be endowed with a finiteness structure [7]. Since resource calculus' purpose is to mimick the identities of the semantics, it is not surprising that the finiteness property of the reduction of Taylor expansion becomes false when an explicit fixpoint is added to the syntax. The intuition behind this remark is that it contains potentially an infinite number of applications, and then its interpretation explodes the cell of finiteness structures, and the resource calculus necessary to simulate this dynamic contains approximants of the fixpoint construction, but those approximants are of unbounded applicative depth.

We propose the study of a calculus endowed with such an explicit fixpoint constructor. We focus on a variant of Plotkin's **PCF** similar to Ehrhard Pagani and Tasson calculus [11].

A quantitative model of that language has already been proposed by Ehrhard and Tasson in the category of probabilistic coherence spaces [5,11]. In this model, the derivation operation is not always defined, but the Taylor formula, which takes the derivatives only at zero, is valid and is then subject to our considerations.

There is a ground type ι corresponding to integers, and the syntax of types is given by :

$$\sigma, \tau, \ldots ::= \iota \mid \sigma \to \tau$$

Definition 3.1 [**PCF**] Let $\underline{k}$ range over $\mathbf{N}$ and x over a countably infinite set of variables.

$$M, N, \ldots ::= \underline{k} \mid x \mid \mathbf{suc}(M) \mid \lambda x M \mid MN \mid \mathbf{If}(M, N, z \cdot N') \mid \mathbf{fix}(M)$$

Reduction rules :

$$\begin{array}{ll} (\lambda x M)N \to_{\mathrm{pcf}} M[N/x] & \mathbf{fix}(M) \to_{\mathrm{pcf}} M(\mathbf{fix}(M)) \\ \mathbf{If}(\underline{k+1}, M, x \cdot N) \to_{\mathrm{pcf}} N[\overline{k}/x] & \mathbf{If}(\underline{0}, M, x \cdot N) \to_{\mathrm{pcf}} M \\ \mathbf{suc}(\underline{k}) \to_{\mathrm{pcf}} \underline{k+1} & \end{array}$$

We define evaluation contexts E, for all terms T, U.

$$E ::= [] \mid EM \mid ME \mid \mathbf{If}(E, M, x \cdot N) \mid \mathbf{suc}(E) \mid \lambda x E$$

and we set as an additional reduction rule $E[M] \to_{\mathrm{pcf}} E[M']$ for each M, M' such that $M \to_{\mathrm{pcf}} M'$.

Definition 3.2 [**PCF** resource calculus $\Delta_{\mathbf{PCF}}$]

$$\Delta_{\mathbf{PCF}} : m, n ::= x \mid \underline{k} \mid \lambda x m \mid \langle m \rangle [n_1, \ldots, n_k] \mid (m = m') \cdot n$$

Reduction rules:

- $\langle \lambda x m\rangle[n_1, \ldots, n_k] \to_{\mathsf{rpcf}} m[n_{f(1)}/x_1, \ldots, n_{f(k)}/x_k]$ for all $f \in \mathfrak{S}_k$ if $\mathsf{deg}_x(m) = k$.
- $(m = \underline{k}) \cdot n \to_{\mathsf{rpcf}} n$ if $m = \underline{k}$.
- We define evaluation contexts e, for all terms m, n of $\Delta_{\mathbf{PCF}}$:

$$e ::= [] \mid \langle e\rangle \overline{m} \mid \langle m\rangle e \mid [e, m_1, \ldots, m_k] \mid (e = \underline{k}) \cdot m \mid \lambda x e$$

  and set the additional rule $e[m] \to_{\mathsf{rpcf}} e[m']$ if $m \to_{\mathsf{rpcf}} m'$ by one of the above rules.

We define for all $n \in \mathbf{N}$ a set of terms $\mathbf{fix}_n$ as follows, with $\mathbf{fix}_0 = \emptyset$:

$$\mathbf{fix}_{n+1} = \left\{ \lambda x \langle x\rangle \left[\langle f_1\rangle[x]_{l_1}, \ldots, \langle f_k\rangle[x]_{l_k}\right] \mid k, l_1, \ldots, l_k \in \mathbf{N}, \forall i \leq k : f_i \in \mathbf{fix}_n \right\}.$$

We can now define the sets of resource terms corresponding to Taylor expansion of **PCF**:

- $\mathcal{T}_{\mathsf{pcf}}(x) = \{x\}$
- $\mathcal{T}_{\mathsf{pcf}}(\underline{k}) = \{\underline{k}\}$
- $\mathcal{T}_{\mathsf{pcf}}(\mathbf{suc}(M)) = \{(m = \underline{k}) \cdot \underline{k+1} \mid m \in \mathcal{T}_{\mathsf{pcf}}(M), k \in \mathbf{N}\}$
- $\mathcal{T}_{\mathsf{pcf}}(\lambda x M) = \{\lambda x m \mid m \in \mathcal{T}_{\mathsf{pcf}}(M)\}$
- $\mathcal{T}_{\mathsf{pcf}}(MN) = \{\langle m\rangle \overline{n} \mid m \in \mathcal{T}_{\mathsf{pcf}}(M), \overline{n} \in \mathcal{T}_{\mathsf{pcf}}(N)^!\}$
- $\mathcal{T}_{\mathsf{pcf}}(\mathbf{fix}(M)) = \{\langle f\rangle \overline{m} \mid f \in \bigcup_{k \in \mathbf{N}} \mathbf{fix}_k, \overline{m} \in \mathcal{T}_{\mathsf{pcf}}(M)^!\}$
- $\mathcal{T}_{\mathsf{pcf}}(\mathbf{If}(M, N, x \cdot N')) = \{(m = \underline{0}) \cdot n \mid m \in \mathcal{T}_{\mathsf{pcf}}(M), n \in \mathcal{T}_{\mathsf{pcf}}(N)\} \cup \{(m = \underline{k+1}) \cdot n'[\underline{k}/x] \mid m \in \mathcal{T}_{\mathsf{pcf}}(M), n' \in \mathcal{T}_{\mathsf{pcf}}(N'), k \in \mathbf{N}\}$

Remark 3.3 A first notable observation we do with respect to the considerations of Section 2.1 is that the first property of Call-By-Value Taylor expansion (Lemma 2.5) is no more valid in **PCF**. Indeed, provided we extend the definition of applicative depth to $\Delta_{\mathbf{PCF}}$, in the natural way, for all $n \in \mathbf{N}$, the set $\mathbf{fix}_n$ is made of terms whose applicative depth belongs to $\{1, \ldots, n\}$. We conclude immediately that for any term $M \in \mathbf{PCF}$, if M has a subterm of the shape $\mathbf{fix}x(M)$, then $\mathcal{T}_{\mathsf{pcf}}(M)$ is a set of terms of unbounded applicative depth.

We are not able anymore to adapt Vaux-Auclair's method to **PCF**. Notice that this point would hold in every calculus with explicit fixpoint, and a similar study could be done with PCF, for example. We have to come back to Ehrhard and Regnier's works [13] if we hope to achieve the wanted finiteness results. This implies that our framework will be uniform, and that the argument would not apply to an algebraic, or even non-deterministic setting, since uniformity would be lost. The extension of the result to such a system remains an open question. In the following, we introduce a binary coherence relation and use Ehrhard and Regnier's method in order to establish that it fits to specific constructions as conditional and explicit fixpoint in our resource construction $\Delta_{\mathbf{PCF}}$.

Example 3.4 [Probabilistic **PCF**] Consider Ehrhard and Tasson's probabilistic **PCF** [11], which is an extension of **PCF** with, in particular a coin constructor, which reduction rule is the following: $\mathsf{coin}(p) \to \underline{0}$ with probability p, and $\mathsf{coin}(p) \to \underline{1}$ with probability $1-p$, for all $p \in [0,1]$. The natural extension of Taylor expansion to this new setting is $\mathcal{T}'_{\mathsf{pcf}}(\mathsf{coin}(p)) = \{0,1\}$ [7], (which is a non-uniform set, according to Definition 3.5). Let us consider now the following term: $M = \mathbf{fix}(\mathbf{If}(\mathsf{coin}(p), \lambda xx, z \cdot \lambda xy))$ for some fixed p. The conditional reduces to λxx with probability p, and to λxy with probability $(1-p)$. If we develop the definition of $\mathcal{T}'_{\mathsf{pcf}}$, we can observe that $\mathcal{T}'_{\mathsf{pcf}}(M)$ contains as a subset $X = \{\langle f\rangle[(0=0)\cdot\lambda xx]_k + [(1=1)\cdot\lambda xy] \mid k \in \mathbf{N}, f \in \mathbf{fix}_{k+1}\}$ (the argument of f is the sum of the two multisets). The normal form of $\mathcal{T}_{\mathsf{pcf}}(M)$ is not empty since each element of X eventually reduces to y. But precisely, y has then an infinite number of antireducts in X, hence in $\mathcal{T}'_{\mathsf{pcf}}(M)$, which contradicts our finiteness result. Notice that it does not mean that the coefficient of y would be necessarily infinite, in the weighted definition of $\mathcal{T}'_{\mathsf{pcf}}$, but that the finiteness result is not general anymore and shall be replaced by a close study of the coefficients involved in Taylor expansion establishing the convergence, or divergence of the weighted infinite sums of terms. For further details about Taylor expansion of probabilistic lambda calculus, we refer to the recent study of Dal Lago and Leventis [20].

3.1 Uniformity and coherence

We define a binary coherence relation between elements of $\Delta_{\mathbf{PCF}}$, which is shown to be stable under parallel reduction $\rightrightarrows_{\text{pcf}}$, and is such that Taylor expansion of a term in **PCF** defines always a clique for this relation. This will lead us to infer the finiteness result : for all $n \in \Delta_{\mathbf{PCF}}, M \in \mathbf{PCF}, \{m \in \mathcal{T}_{\mathsf{pcf}}(M) \mid m \rightrightarrows_{\text{pcf}} n\}$ has at most one element.

Definition 3.5 [Coherence on resource terms of $\Delta_{\mathbf{PCF}}$]

- $x \frown x$ for all x.
- $\underline{k} \frown \underline{k}'$ if $k = k'$.
- $\lambda xm \frown \lambda xm'$ if $m \frown m'$.
- $\langle m\rangle\overline{n} \frown \langle m'\rangle\overline{n}'$ if $m \frown m'$ and $\overline{n} \frown \overline{n}'$.
- $(m = \underline{k})\cdot n \frown (m' = \underline{k}')\cdot n'$ if $m \frown m'$ and $\underline{k} = \underline{k}' \to n \frown n'$
- $[m_1, \dots m_k] \frown [m_{k+1}, \dots, m_{k+l}]$ if $\forall i,j \in \{1,\dots,k+l\}, m_i \frown m_j$.

Lemma 3.6 *For all term M of* **PCF***, $\mathcal{T}_{\mathsf{pcf}}(M)$ is a clique for the relation $\frown$.*

Proof. The proof is by induction on M. We only detail the fixpoint case, other constructions following from straightforward induction steps. Let $M = \mathbf{fix}(N)$. $\mathcal{T}_{\mathsf{pcf}}(M) = \{\langle f\rangle\overline{n}; f \in \cup_{k\in\mathbf{N}}\mathbf{fix}_k, \overline{n} \in \mathcal{T}(N)^!\}$. Induction hypothesis implies that it is sufficient to prove that $\cup_{k\in\mathbf{N}}\mathbf{fix}_k$ is a clique. Let $f \in \mathbf{fix}_k$ and $f' \in \mathbf{fix}_l$ for $k,l \in \mathbf{N}$.

---

[7] Because we do not need to consider the coefficients for the argument. Otherwise, the definition is $\mathcal{T}'_{\mathsf{pcf}}(\mathsf{coin}(p)) = p\cdot 0 + (1-p)\cdot 1$.

- If $k = l$, then by induction (starting from 1 because $\mathbf{fix}_0 = \emptyset$):
  - If $k, l = 1$, then $f = f' = \lambda x\langle x\rangle[]$ ($\mathbf{fix}_1$ is a singleton).
  - If $k = l = k' + 1$, then $f = \lambda x\langle x\rangle[\langle f_1\rangle\overline{x}, \ldots, \langle f_{l'}\rangle\overline{x}]$, with for all $i \in \{1, \ldots, l'\}$, $f_i \in \mathbf{fix}_{k'}$ and $f' = \lambda x\langle x\rangle[\langle f'_1\rangle\overline{x}, \ldots, \langle f'_{l''}\rangle\overline{x}]$ with for all $j \in \{1, \ldots, l''\}$, $f'_j \in \mathbf{fix}_{k'}$. By induction hypothesis, $f_i \frown f'_j$ for all $i \in \{1, \ldots, l'\}$ and all $j \in \{1, \ldots, l''\}$. Moreover, all bags $\overline{x}$ are pairwise coherent. So, $f \frown f'$.
- If $k > l$. Then, by induction on k :
  - If $k = 1$, then $f' = \lambda x\langle x\rangle[]$ and $f = \lambda x\langle x\rangle\overline{n}$ for $\overline{n} \in \mathbf{fix}^!_{m-1}$. By definition, $[] \frown \overline{n}$ for all $\overline{n}$, so $f \frown f'$.
  - If $k > 1$ then $f = \lambda x\langle x\rangle[\langle f_1\rangle\overline{x}, \ldots, \langle f_{l'}\rangle\overline{x}]$, where for all $i \in \{1, \ldots, l'\}$, $f_i \in \mathbf{fix}_{l-1}$. Moreover, $f' = \lambda x\langle x\rangle[\langle f'_1\rangle\overline{x}, \ldots, \langle f'_{l''}\rangle\overline{x}]$ where for all $j \in \{1, \ldots, l''\}$, $f'_j \in \mathbf{fix}_{k-1}$. By induction hypothesis, $f_i \frown f'_j$ for all $i \in \{1, \ldots, l'\}$ and all $j \in \{1, \ldots, l''\}$, because $f_i \in \mathbf{fix}_{l-1}$, $f'_j \in \mathbf{fix}_{k-1}$ and $l - 1 > k - 1$. Bags $\overline{x}$ being pairwise coherent, we have $f \frown f'$.

□

The coherence relation we introduced allows to compare terms having *the same shape*. In particular, if two terms are coherent and are redexes, there is a way to have a pair of reductions leading to reducts also pairwise coherent, even if coherence is not preserved by reduction in general. This point is made explicit in the following lemma.

The constraint we set on the two reductions is to avoid pairs like $\langle\lambda xm\rangle\overline{n} \frown \langle\lambda xm'\rangle\overline{n}'$, reducing respectively to $m[n_{f(1)}/x_1, \ldots, n_{f(k)/x_k}]$ and to $\langle\lambda xm\rangle\overline{n}''$ if $\overline{n}' \to_{\mathsf{rpcf}} \overline{n}''$. In this case, the two reducts are obviously not pairwise coherent.

Lemma 3.7 *Let m, m', n, n' such that $m \frown m'$, and either:*

- $m = \langle\lambda xr\rangle\overline{u}$, $m' = \langle\lambda xr'\rangle\overline{u}'$, $n = r[u_1/x_{f(1)}, \ldots, u_k/x_{f(k)}]$ *and* $n' = r'[u'_1/x_{f'(1)}, \ldots, u'_{k'}/x_{f'(k')}]$
- $m = (u = (j, v)) \cdot r$, $m' = (u' = (j', v')) \cdot r'$, $n = r$ *and* $n' = r'$.
- $m = e[u]$, $m' = e[u']$, $n = e[r]$ *and* $n' = e[r']$ *with the pair of reductions* $n \to_{\mathsf{rpcf}} r$ $u' \to_{\mathsf{rpcf}} r'$ *follows any of the above schemes.*

Then $n \frown n'$. If moreover $n = n'$, then $m = m'$.

Proof. We refer to Ehrhard and Regnier's proof of Theorem 10 [13] for the first redex case. The other reductions follow by a straightforward induction. □

Our goal now is to show, thanks to the previous results, that if for some M, two terms $m, m' \in \mathcal{T}_{\mathsf{pcf}}(M)$ reduce to a same term n, then $m = m'$. We establish this for parallel resource reduction $\rightrightarrows_{\mathrm{pcf}}$ defined below.

Definition 3.8 [Parallel reduction of **PCF** resource terms]

- $m \rightrightarrows_{\mathrm{pcf}} m$ for all $m \in \Delta_{\mathbf{PCF}}$.
- If $m_i \rightrightarrows_{\mathrm{pcf}} m'_i$ for all $i \in \{1, \ldots, n\}$ then $[m_1 \ldots, m_n] \rightrightarrows_{\mathrm{pcf}} [m'_1, \ldots m'_n]$.

- If $m \rightrightarrows_{\text{pcf}} m'$, $[n_1 \dots, n_k] \rightrightarrows_{\text{pcf}} [n'_1, \dots n'_k]$ $\mathsf{deg}_x(m) = k$, and $f \in \mathfrak{S}_k$, then $\langle \lambda x m \rangle [n_1, \dots, n_k] \rightrightarrows_{\text{pcf}} m'[n'_1/x_{f(1)}, \dots, n'_k/x_{f(k)}]$.
- If $m \rightrightarrows_{\text{pcf}} m'$ then $\lambda x m \rightrightarrows_{\text{pcf}} \lambda x m'$
- If $m \rightrightarrows_{\text{pcf}} m'$ and $n \rightrightarrows_{\text{pcf}} n'$, then $(m = \underline{k}) \cdot n \rightrightarrows_{\text{pcf}} (m' = \underline{k}) \cdot n'$
- If $m \rightrightarrows_{\text{pcf}} m'$ and $n \rightrightarrows_{\text{pcf}} n'$ then $(m = \underline{k}) \cdot n \rightrightarrows_{\text{pcf}} n'$ if $m = \underline{k}$.
- If $m \rightrightarrows_{\text{pcf}} m'$ and $\overline{n} \rightrightarrows_{\text{pcf}} \overline{n}'$ then $\langle m \rangle \overline{n} \rightrightarrows_{\text{pcf}} \langle m' \rangle \overline{n}'$.

Lemma 3.9 *Let m, m', n, n' such that $m \subset n$, $m \rightrightarrows_{pcf} m'$ and $n \rightrightarrows_{pcf} n'$. If $m' = n'$ then $m = n$.*

Proof. The proof is by induction on m. The contextual reductions follow from induction hypothesis, and redex cases follow from Lemma 3.7. We only give the initialisation and examples of such two reductions, the other calling for identical arguments.

- If $m = x$ then $m' = n = n' = x$.
- If $m = \lambda x s$, then $n = \lambda x r$ for $s \subset r$, $m' = \lambda x s'$ for $s \rightrightarrows_{\text{pcf}} s'$ and $n' = \lambda x r'$ for $r \rightrightarrows_{\text{pcf}} r'$. By induction hypothesis, and since $s' = r'$, we have $s = r$ and then $m = n$.
- If $m = \langle \lambda x r \rangle [s_1, \dots, s_n]$ and $m' = r'[s'_1/x_{f(1)}, \dots, s'_k/x_{f(k)}]$, then we observe that there exist $v \subset r, w_i \subset s_i$ such that $n = \langle \lambda x v \rangle [w_1/x_{g(1)}, \dots, w_l/x_{g(l)}]$. Since $m \rightrightarrows_{\text{pcf}} m'$, by a classical standardization argument, $m \to_{\mathsf{rpcf}} m'' = r[s_1/x_{f(1)}, \dots, s_k/x_{f(k)}] \rightrightarrows_{\text{pcf}} m'$ and $n \to_{\mathsf{rpcf}} n'' = v[w_1/x_{g(1)}, \dots, w_l/x_{g(l)}] \rightrightarrows_{\text{pcf}} n'$. We can now apply Lemma 3.7 and deduce that $m'' \subset n''$, and conclude by induction hypotheses as above.

□

By combining Lemmas 3.7 and 3.9, we obtain the following result which was our goal, and which is a particular case of the finiteness property of Taylor expansion :

Corollary 3.10 *Let $n \in \Delta_{\mathbf{PCF}}$, and M a term of* **PCF**. *$\#\{m \in \Delta_{\mathbf{PCF}}; m \in \mathcal{T}_{\mathsf{pcf}}(M), m \rightrightarrows n\} \leq 1$.*

We can then reproduce the arguments we set for Call-By-Value to state that a quantitative version of Taylor expansion makes sense, with respect to the finiteness of coefficients keeping true under reduction, and then in Taylor normal form.

4 The Bang calculus

Pursuing the investigation of the relations between Taylor expansion and reduction strategies, Ehrhard and Guerrieri introduced a fine grain calculus that permits the embedding of Call-By-Name and Call-By-Value in it [10]. Guerrieri and Manzonetto studied more recently the correspondence between these embeddings and the respective operational semantics of Call-By-Name and Call-By-Value [16]. We briefly explain how our finiteness result applies in this particular setting.

A motivation behind this study is to approach a calculus close to Levy's Call-By-Push-Value[22]. In order to do so, a first step is to study then Bang Calculus that has similar properties to Call-By-Push-Value, with respect to the embeddings of distinct strategies of evaluation. We are interested in Call-By-Push-Value because Ehrhard and Tasson provided an interpretation of it in probabilistic coherent spaces, and extending our results to a Call-By-Push-Value resource calculus would generalize the construction for all the evaluation strategies that can be embedded here. This extension in discussed in the conclusion.

Bang calculus also differs from **PCF** because of **PCF** being typed, in order to stay consistent with Ehrhard, Pagani and Tasson's model, while Ehrhard and Guerrieri use reflexive objects of shape $X \cong\, !X \,\&\, (!X \multimap X)$ in the semantics, so as to work is an untyped setting.

This *Bang calculus* denoted as Λ_{b} and the corresponding resource calculus denoted as Δ_{b} are defined below. In addition to the usual β-reduction rule, the calculus is endowed with a dereliction/promotion rule $()^!/\mathbf{der}$, which "opens" a multiset, giving access to its content. It is equivalent to the opening of a box in Linear Logic proof nets,

Definition 4.1 [Bang Calculus Λ_{b} [10]]

$$V ::= x \mid M^! \qquad \Lambda^! : M, N ::= \lambda x M \mid \mathbf{der}(M) \mid MN$$

The reduction rules are the following : $\mathbf{der}(M^!) \to_{\mathsf{b}} M \qquad (\lambda x M)V \to_{\mathsf{b}} M[V/x]$

Definition 4.2 [Bang resource calculus Δ_{b} [10]]

$$\Delta_{\mathsf{b}} : m, n ::= x \mid \lambda x m \mid \langle m \rangle n \mid [m_1, \ldots, m_k] \mid \mathbf{der}(m)$$

The reduction rules are the following :

$$\mathbf{der}([m]) \to_{\mathsf{rb}} m \qquad \langle \lambda x m \rangle y \to_{\mathsf{rb}} m[y/x]$$
$$\langle \lambda x m \rangle [n_1, \ldots, n_k] \to_{\mathsf{rb}} m[n_1/x_{f(1)}, \ldots, n_k/x_{f(k)}] \text{ if } k = \mathsf{deg}_x(m) \text{ and } f \in \mathfrak{S}_k$$

The Call-By-Name and Call-By-Value embeddings $()^{\mathsf{name}}$ and $()^{\mathsf{val}}$ of usual pure λ-calculus into the Bang calculus run as follows:

$$\begin{aligned} x^{\mathsf{name}} &= \mathbf{der}(x) & x^{\mathsf{val}} &= x \\ (\lambda x M)^{\mathsf{name}} &= \lambda x M^{\mathsf{name}} & (\lambda x M)^{\mathsf{val}} &= (\lambda x M^{\mathsf{val}})^! \\ (MN)^{\mathsf{name}} &= (M^{\mathsf{name}})(N^{\mathsf{name}})^! & (MN)^{\mathsf{val}} &= (\mathbf{der}(M^{\mathsf{val}}))N^{\mathsf{val}} \end{aligned}$$

These two embeddings follow the well-known translations of intuitionistic implication $A \to B$ to linear logic formulas $!A \multimap B$ (Call-By-Name) and $!A \multimap !B$ (Call-By-Value). This is a striking feature of linear logic to permit the distinction between the two evaluation strategies through the management of exponentials formulas. The multiset construction of Taylor expansion defined below is then directed by the promotion construct $()^!$, while, roughly speaking, in Call-By-Name or Call-By-Value, it was directed by the intuitive distinction function/argument.

Definition 4.3 [Bang Calculus Taylor expansion [10]]

$$\mathcal{T}_{\mathsf{b}}(x) = x \qquad \mathcal{T}_{\mathsf{b}}(\lambda x M) = \{\lambda x m \mid m \in \mathcal{T}_{\mathsf{b}}(M)\}$$
$$\mathcal{T}_{\mathsf{b}}(MN) = \{\langle m\rangle n \mid m \in \mathcal{T}_{\mathsf{b}}(M), n \in \mathcal{T}_{\mathsf{b}}(N)\} \qquad \mathcal{T}_{\mathsf{b}}(\mathbf{der}(M)) = \{\mathbf{der}(m) \mid \mathbf{m} \in \mathcal{T}_{\mathsf{b}}(M)\}$$
$$\mathcal{T}_{\mathsf{b}}(M^!) = \{[m_1, \dots, m_k] \mid k \in \mathbf{N}, \forall i : m_i \in \mathcal{T}_{\mathsf{b}}(M)\}$$

We can observe that for any $\langle \lambda x m\rangle n \in \mathcal{T}_{\mathsf{b}}(((\lambda x M)N)^{\mathsf{val}})$, reducing the external redex is possible if and only if N is a value. Indeed, in this case, n is a multiset. Otherwise, n must be evaluated before the external redex. On the other hand, if $\langle m\rangle n \in \mathcal{T}_{\mathsf{b}}(((\lambda x M)\, N)^{\mathsf{name}})$, then n is always a multiset and the external redex can always be reduced. We even have, for any term M of pure λ-calculus, a close correspondence between $\mathcal{T}_{\mathsf{b}}(M^{\mathsf{val}})$ and $\mathcal{T}_{\mathcal{V}}(M)$, up to some technical differences related to the dereliction and the variables, but the multiset structure is the same. This remark also holds for usual Call-By-Name Taylor expansion $\mathcal{T}(M)$ defined by Ehrhard and Regnier, and $\mathcal{T}_{\mathsf{b}}(M^{\mathsf{name}})$. We can now announce the finiteness result for the Bang calculus, because we can naturally define a parallel reduction $\rightrightarrows_{\mathsf{rb}}$ included in the reflexive transitive closure of $\to_{\mathsf{rb}}$, necessary for simulate β-reduction.

Proposition 4.4 *Let $M \in \Lambda^!$, $n \in \Delta^!$, $k \in \mathbf{N}$. $\{m \in \mathcal{T}_{\mathsf{b}}(M) \mid m \rightrightarrows_{\mathsf{rb}}^k n\}$ is finite.*

We do not detail the proof, because it would rely on the same ingredients and arguments we set for Call-By-Value: it is not difficult to observe that for all $M \in \Lambda^!$, $\mathcal{T}_{\mathsf{b}}(M)$ enjoys the necessary properties for proposition 4.4 to be established. Notice that we did not consider here any σ-rules, as Guerrieri and Ehrhard, but it will be necessary in further works in order to study issues about normal forms, and clashes during the reduction. An observation of Definition 4.3 is sufficient to see that the applicative depth of terms in $\mathcal{T}_{\mathsf{b}}(M)$ cannot exceed the depth in M, and the argument about the size, leading to Lemma 2.8 in the Call-By-Value calculus appears valid for the Bang calculus too. In other words, the applicability of the method does not depend essentially on the choice of a reduction strategy, because in particular the depth of resource terms appearing in Taylor expansion does not depend on how we deal with exponentials.

5 Conclusion and perspectives

We introduced the necessary definitions to study Taylor expansion in various settings which demand respectively distinct proof methods. A possible extension of this work is to generalize these results in a common setting thanks to our work with Vaux-Auclair on linear logic-proof nets [4]. Indeed, there exist already well-known embeddings of Call-By-Name and Call-By-Value into proof nets, and the results about Taylor expansion become then a syntactic work, of presenting these translations and proving that they commute with Taylor expansion (*e.g.* that Taylor expansion of the proof net coming from a Call-By-Value translation corresponds to the translation of Call-By-Value Taylor expansion of Section 2.1). But a construction of proof nets corresponding to Call-By-Need, Bang Calculus and to **PCF** would be of great interest in that perspective.

The other direction of work that is suggested by our study is to define Taylor expansion for Levy's Call-By-Push-Value [22]. Indeed, we saw with Bang Calculus that a calculus endowing Call-By-Name and Call-By-Value can be simulated by a resource calculus, and that the finiteness property of its Taylor expansion can be adapted from Call-By-Name and Call-By-Value constructions, and that the extension to an explicit exponential and dereliction causes no damage to the good behaviour of our proof methods. On the other hand, we saw with **PCF** that we can deal with a typed setting with explicit fixpoint, if we come back to arguments relying on coherence between resource terms. Then we have good hope to extend our results to a linear logic-based variant of Call-By-Push-Value, interpreted in Ehrhard and Tasson's work [14]. But we cannot proceed easily in that way, because of the expressivity of the calculus, and the management of types both making it difficult to introduce a resource calculus with an adequate notion of reduction. In particular, an application $(\lambda x M)V$ is a redex if V has a positive type (that is $!, \otimes$, or $\oplus$). This implies that the quantitative interpretation of V will not always correspond to the exponential, and hence will not always be considered as a multiset, from a resource point of view, but can be a pair (v_1, v_2) for example, if V is of a product type. In that case, Ehrhard and Tasson use semantical arguments (relied to the presence of morphisms of coalgebras in the model) to state that positive types are freely duplicable, but giving an account of this property in the syntax is not a trivial task and calls for a detailed construction of a particular resource calculus, that cannot be easily imported from the existent ones, where only the reducible applications have always an argument interpreted by an exponential, and thus approximated by a multiset.

References

[1] Z. M. Ariola, M. Felleisen, J. Maraist, M. Odersky, and P. Wadler. The call-by-need lambda calculus. In *Conference Record of POPL'95*, 1995.

[2] G. Boudol. The lambda-calculus with multiplicities (abstract). In *CONCUR'93*, 1993.

[3] S. Castellan, P. Clairambault, H. Paquet, and G. Winskel. The concurrent game semantics of probabilistic PCF. In *LICS 2018, Oxford, UK, July 09-12, 2018*, pages 215–224, 2018.

[4] J. Chouquet and L. Vaux Auclair. An application of parallel cut elimination in unit-free multiplicative linear logic to the Taylor expansion of proof nets. In *CSL 2018*, 2018.

[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] Daniel de Carvalho. Execution time of lambda-terms via denotational semantics and intersection types. *CoRR*, abs/0905.4251, 2009. arXiv:0905.4251.

[7] T. Ehrhard. Finiteness spaces. *Mathematical Structures in Computer Science*, 15(4):615–646, 2005.

[8] T. Ehrhard. A finiteness structure on resource terms. In *LICS 2010, 11-14 July 2010, Edinburgh, United Kingdom*, 2010.

[9] T. Ehrhard. Collapsing non-idempotent intersection types. In *Computer Science Logic CSL 2012*, pages 259–273, 2012.

[10] T. Ehrhard and G. Guerrieri. The bang calculus: an untyped lambda-calculus generalizing call-by-name and call-by-value. In *Principles and Practice of Declarative Programming*, 2016.

[11] T. Ehrhard, M. Pagani, and C. Tasson. Full abstraction for probabilistic PCF. *CoRR*, abs/1511.01272, 2015. arXiv:1511.01272.

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

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

[14] T. Ehrhard and C. Tasson. Probabilistic call by push value. *CoRR*, abs/1607.04690, 2016. arXiv:1607.04690.

[15] J-Y. Girard. Normal functors, power series and lambda-calculus. *Annals of Pure and Applied Logic*, 37(2):129, 1988.

[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] D. Kesner. Reasoning about call-by-need by means of types. In *Foundations of Software Science and Computation Structures - 19th International Conference, FOSSACS 2016*, 2016.

[20] 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.

[21] J. Laird, G. Manzonetto, G. McCusker, and M. Pagani. Weighted relational models of typed lambda-calculi. In *LICS 2013*, pages 301–310, 2013.

[22] 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.

[23] 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.

[24] P-M. Pédrot and A. Saurin. Classical by-need. In *Programming Languages and Systems - 25th European Symposium on Programming, ESOP 2016*, pages 616–643, 2016.

[25] Gordon D. Plotkin. LCF considered as a programming language. *Theor. Comput. Sci.*, 5(3):223–255, 1977.

[26] C. Tasson and L. Vaux. Transport of finiteness structures and applications. *Mathematical Structures in Computer Science*, 28(7):1061–1096, 2018.

[27] T. Tsukada, K. Asada, and C.-H. L. Ong. Species, profunctors and Taylor expansion weighted by SMCC. In *LICS 2018, Oxford, UK, July 09-12, 2018*, pages 889–898, 2018.

[28] L. Vaux. The algebraic lambda calculus. *Mathematical Structures in Computer Science*, 19(5):1029–1059, 2009.

[29] 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.

A Call-By-Need approximation

Definition A.1 [Call-By-Need calculus Λ_{need} [24]]

$$\begin{aligned}
\text{Terms : } & T, U ::= x \mid \lambda x T \mid lx T \mid TU \\
\text{Values : } & V ::= \lambda x T \\
\text{Contexts : } & C ::= [] \mid C_1[lx C_2]T \\
& E ::= [] \mid ET \mid lxE \mid C[lxE_1[x]]E_2 \\
\text{Reduction rules : } & C[\lambda x T]U \to_{\mathsf{need}} C[lxT]U \\
& C_1[lxE[x]]C_2[V] \to_{\mathsf{need}} C_2[C_1[lxE[V]]V]
\end{aligned}$$

Definition A.2 [Resource approximation of Call-By-Need calculus]

- Approximation of Λ_{need} terms:
  - $\cdot$ $[x, \dots, x] \lhd x$
  - $\cdot$ $[\lambda x t_1, \dots, \lambda x t_k] \lhd \lambda x T$ if $\forall i \in \{1, \dots, k\}, t_i \lhd T$
  - $\cdot$ $tu \lhd TU$ if $t \lhd T$ et $u \lhd U$
  - $\cdot$ $lxt \lhd lxT$ if $t \lhd T$.
- Approximation of contexts C. $c \lhd C$ if :
  - $\cdot$ $c = C = []$
  - $\cdot$ $c = c_1[lxc_2]t$, $C = C_1[lxC_2]T$, $c_1 \lhd C_1, c_2 \lhd C_2$ et $t \lhd T$.
- Approximation of contexts E. $e \lhd E$ if :
  - $\cdot$ $e = E = []$
  - $\cdot$ $e = lxe_2, E = lxE_2$, et $e_2 \lhd E_2$
  - $\cdot$ $e = e_2t, E = E_2T, e2 \lhd E_2$ et $t \lhd T$
  - $\cdot$ $e = c[lxe_2[[x, \dots, x]]]e_3, E = C[lxE_2[x]]E_3, e_2 \lhd E_2, e_3 \lhd E_3$ et $c \lhd C$.

Lemma A.3 *If $t \lhd T$ and $c \lhd C$, then $c[t] \lhd C[T]$*

Proof. Induction on contexts :

- $c = []$, straightforward by the hypothesis $t \lhd T$
- $c = c_1[lxc_2]u$. By definition, $C = C_1[lxC_2]U$ with $c_i \lhd C_i$ and $u \lhd U$. By induction hypothesis, we have $c_2[t] \lhd C_2[T]$. We extend (by definition of $\lhd$) to $lxc_2[t] \lhd lxC_2[T]$. By induction hypothesis on c_1, and because $u \lhd U$, we deduce $c_1[lxc2[t]]u \lhd C_1[lxC_2[T]]U$.

□

Lemma A.4 *If $e \lhd E$ and $t \lhd T$, then $e[t] \lhd E[T]$*

Proof. Similar induction on contexts e. □

Lemma A.5

(i) *If $t \lhd T$ and $T = C[U]$, then there exists $u \lhd U$ and $c \lhd C$ such that $t = c[u]$.*

(ii) *If $t \lhd T$ and $T = E[U]$, then there exists $u \lhd U$ and $e \lhd E$ such that $t = e[u]$.*

Proof. (i) induction on C.

- If $C = []$, then $T = U$, and t fits.
- If $C = C_1[lxC_2]T'$, then by definition of $\lhd$, if $t \lhd T$, $t = st'$ with $s \lhd C_1[lxC2[U]]$ and $t' \lhd T'$. By induction hypothesis on C_1, $s = c_1[lxw]$ with $c_1 \lhd C_1$ and $w \lhd C_2[U]$.
  Similarly, by induction hypothesis on C_2, $w = c_2[u]$ with $c_2 \lhd C_2$ and $u \lhd U$. We then have $t = c_1[lxc_2[u]]t'$, and $c_i \lhd C_i, u \lhd U, t' \lhd T'$.

(ii) : similar induction on E; □

Lemma A.6 *Let $T \to_{\mathsf{need}} U$. For all $t \lhd T$, either $t \to_{\mathsf{rneed}} 0$, or there is some $u \lhd U$ such that $t \to_{\mathsf{rneed}} u$.*

Proof. Induction on the reduction:

- $T = C[\lambda x T']T''$, and $U = C[lxT_1]T_2$. By Lemma A.5, if $t \lhd T$, there exist $t_1, \ldots t_k \lhd t', t'' \lhd T''$, $c \lhd C$ such that $t = c[[\lambda x t_1, \ldots \lambda x t_k]]t''$. If $k \neq 1$, $t \to_{\mathsf{rneed}} 0$. Otherwise, $t \to_{\mathsf{rneed}} c[lxt_1]t'' \lhd U$.
- $T = C_1[lxE[x]]C[V]$, $U = C[C_1[lxE[V]]V]$. By Lemma A.5, if $t \lhd T$, then $t = c'[lxe[[x_1, \ldots, x_k]]c[[v_1, \ldots, v_k] + \overrightarrow{v}]$ with $c' \lhd C', c \lhd C, e \lhd E, \forall i : v_i \lhd V$. We have $t \to_{\mathsf{rneed}} c[c_1[lxe[[v_1, \ldots, v_k]]]\overrightarrow{v}] \lhd U$

$\square$

Lemma A.7 *Let $T \to_{\mathsf{need}} U$. For all $u \lhd U$, there exists $t \lhd T$ such that $t \to u$.*

Proof. Induction on the reduction.

- $U = C[lxU']U''$ et $T = C[\lambda x U']U''$. By Lemma A.5, if $u \lhd U$, then $u = c[lxu']u''$ for some $c \lhd C, u' \lhd U', u'' \lhd u''$. It is then sufficient to consider $t = c[[\lambda x u']]u''$, because in that case, $t \lhd T$.
- $U = C[C_1[lxE[V]]V]$, $T = C_1[lxE[x]]C[V]$: similar argument.

$\square$