Resource approximation for the $\lambda\mu$-calculus

Davide Barbarossa
Laboratoire d'Informatique Paris-Nord, Université Sorbonne Paris-Nord
Villetaneuse, France
Dipartimento di Informatica - Scienza ed Ingegneria, Università di Bologna
Bologna, Italy
davide.barbarossa@unibo.it

Abstract

The $\lambda\mu$-calculus plays a central role in the theory of programming languages as it extends the Curry-Howard correspondence to classical logic. A major drawback is that it does not satisfy Böhm's Theorem and it lacks the corresponding notion of approximation. On the contrary, we show that Ehrhard and Regnier's Taylor expansion can be easily adapted, thus providing a resource conscious approximation theory. This produces a sensible $\lambda\mu$-theory with which we prove some advanced properties of the $\lambda\mu$-calculus, such as Stability and Perpendicular Lines Property, from which the impossibility of parallel computations follows.

CCS Concepts: • **Theory of computation** → **Lambda calculus**; *Linear logic*; Rewrite systems.

Keywords: $\lambda\mu$-calculus, Taylor expansion, resource approximation, linear logic, λ-calculus

ACM Reference Format:
Davide Barbarossa. 2022. Resource approximation for the $\lambda\mu$-calculus. In *37th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS) (LICS '22), August 2–5, 2022, Haifa, Israel.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3531130.3532469

1 Introduction

1.1 Motivation

1.1.1 Curry-Howard correspondence for classical logic.

The celebrated *Curry-Howard correspondence* states that a class of programs, written in a suitable programming language, and intuitionistic logic proofs, written in an suitable formal system, are the same mathematical objects. The typical suitable programming language is λ-calculus, and the typical suitable formal system is intuitionistic natural deduction NJ; under this correspondence, the simply typed λ-calculus is identified with NJ. A natural question is what happens for *classical* logic proofs and whether it is possible to find such a correspondence at all. In the 90's, several ways for generalising such a correspondence to this framework appeared, starting from [13] where Griffin suggests to type control operators (such as Scheme's `callcc` or Felleisen's `C` operator) with Peirce's law. One of the most notable ones is the $\lambda\mu$-calculus, introduced by Parigot in [24], which has the advantage of allowing the correspondence to take the exact same form as in the intuitionistic case: just like λ-calculus is the Turing-complete programming language in which intuitionistic logic expresses its computational content, $\lambda\mu$-calculus is the one expressing the computational content of classical logic. From the programming viewpoint, the big difference between λ-calculus and $\lambda\mu$-calculus is that the former is a *purely functional* language, while the latter is *impure*, due to the possibility of encoding control operators in it — like the already mentioned `callcc` or `C`. From the point of view of, e.g., Classical realizability [19], this corresponds to the backtracking mechanism related to classical reasoning.

1.1.2 Taylor expanding programs.

Just before the 90's another major discovery in logic and computer science appeared: Girard's linear logic [12]. This opened a whole new field of research, in which the common line is the deep role reserved to *resources* in a computation/proof. Linear logic allowed Ehrhard and Regnier to discover an astonishing correspondence between linearity in analysis and linearity in computer science, that is formalized in the *differential λ-calculus* [9] (and differential interaction nets [10]). It is possible to Taylor expand programs/proofs by – as in analysis – an infinite series of approximants weighted via the usual factorial coefficients. This is usually called the "full" or "quantitative" Taylor expansion. However, it turns out that even if we do not consider the coefficients, we still obtain a meaningful theory of program approximation: it is usually called the "qualitative" Taylor expansion, and will play a central role in the present article. Under the assumption of having an idempotent sum, the Taylor expansion is no longer a series but becomes a set, and the approximants can be written in a simple "target language", called *resource calculus* (very similar to Boudol's calculus with multiplicities [4]). In [1], it is shown that all the fundamental results in the so called *approximation theory* of λ-calculus

LICS '22, August 2–5, 2022, Haifa, Israel
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9351-5/22/08…$15.00
https://doi.org/10.1145/3531130.3532469

(Monotonicity, Genericity, Continuity, Stability, Perpendicular Lines Property), usually achieved via labelled reductions and Böhm trees [2, Chapter 14], can be actually proven – in an arguably more satisfactory way – via (qualitative) Taylor expansion; in other words, in λ-calculus, resource approximation (which is at the basis of Taylor expansion) "subsumes" Böhm trees approximation, and answers positively to a proposal expressed in [9] where Ehrhard and Regnier mention that: *"Understanding the relation between the term and its full Taylor expansion might be the starting point of a renewing of the theory of approximation."*

1.1.3 The content of this paper.

The aforementioned "approximation results" are in fact at the basis of a *mathematical study* of λ-calculus, deep from the conceptual, mathematical and computational viewpoint. A natural question is then: what about for other programming languages? There are many works which extend the notion of (qualitative as well as quantitative) Taylor expansion to other programming languages ([5, 6, 17, 21, 30]), usually concentrating on its relations with normalisation. Not always easy is, then, *applying* it to actually study the properties of the source language. This is maybe due to the fact that, unlike in the long time studied λ-calculus, for other languages one does not really know what a "mathematical theory of it" should look like. In the present work, we tackle the case of $\lambda\mu$-calculus for which, to the best of our knowledge, the problem of directly defining a Taylor expansion has never been priorly considered. In this sense, our work can be seen as a continuation of the above mentioned series of papers, and may be related to [18] where the authors study non-idempotent intersection and union types for $\lambda\mu$-calculus.

We propose here to reverse the above proposal of Ehrhard and Regnier and *start* by defining a resource sensitive version and a Taylor expansion for $\lambda\mu$-calculus, trying then to *use* this approximation machinery to prove – following [1] – mathematical properties of the language.

A notable work has to be mentioned: in [29], Vaux-Auclair defines a full differential $\lambda\mu$-calculus following the steps of [9]. Its version takes coefficients into account, and as always this raises a series of non-trivial problems to handle. However, he does not define a Taylor expansion nor does he apply those tools to find properties of the language. The present work can thus be seen also as a continuation of Vaux-Auclair's one.

There are, from our point of view, several reasons for considering the $\lambda\mu$-calculus: first of all, from the Curry-Howard point of view, it is the natural "successor" of λ-calculus. Moreover, it is a standard reference for the study of control operators in functional languages. Yet, there are just few attempts to really study its mathematical theory, and the state of the art is not comparable with the well-established one for λ-calculus. For example, Laurent in [23] makes the following observation: *"Models of the simply typed λ-calculus, of the untyped λ-calculus and of the simply typed $\lambda\mu$-calculus are well understood, but what about models of the untyped $\lambda\mu$-calculus? As far as we know, this question has been almost ignored."* With the same motivation, we look at the other major part which constitutes a mathematical theory of a programming language, namely the *theory of approximation*. In this sense, the present work can be seen as a continuation of [23].

Other points in relation with Krivine's classical realizability, proof-nets, CPS-translations and Saurin's $\Lambda\mu$-calculus will be mentioned in the conclusions.

The article is organised as follows: in Section 2 we define the resource $\lambda\mu$-calculus and prove that it is strongly normalising and confluent (Corollaries 2.13 and 2.30). In Section 3 we define the qualitative Taylor expansion and prove its main properties, which give rise to a non-trivial sensible "$\lambda\mu$-theory" (Corollary 3.15). In Section 4 we apply these approximation tools to prove two important results: Stability (Theorem 4.1) and Perpendicular Lines Property (Theorem 4.4). As a consequence, we obtain the non-representability of *parallel-or* in $\lambda\mu$-calculus (Corollaries 4.2 and 4.5).

1.2 Quick overview of $\lambda\mu$-calculus

We briefly recall the definition of the $\lambda\mu$-calculus, and introduce some basic notions and notation.

Definition 1.1. *Fix a countable set whose elements are called* variables *and a disjoint countable set whose elements are called* names. *The set $\lambda\mu$ of $\lambda\mu$-*terms *is generated by the following grammar:*

$$M ::= x \ \mid\ \lambda x.M \ \mid\ MM \ \mid\ \mu\alpha.{}_{\beta}|M|$$

(for x a variable and α, β names) in which, as usual, λ binds x in M as well as μ binds α in ${}_{\beta}|M|$. Terms are considered up to renaming of bound variables and names.

Despite not being actual subterms, words of shape ${}_{\alpha}|M|$ are called *named terms*[1]. M is said to be *named under α*.

The *k-contexts* (also called *multihole-contexts* when k is generic) $C = C\{\xi_1, \dots, \xi_k\}$ are defined as expected by:

$$C ::= x \ \mid\ \xi_i \ \mid\ \lambda x.C \ \mid\ CC \ \mid\ \mu\alpha.{}_{\beta}|C|$$

where $\{\xi_1, \dots, \xi_k\}$ is a new set whose elements are called *holes*. 1-contexts are simply called *contexts*. A context with exactly one occurrence of the hole is called *single-hole*, and as usual satisfy: $C ::= \xi_i \mid \lambda x.C \mid CM \mid MC \mid \mu\alpha.{}_{\beta}|C|$.

[1]Historically named terms are written as $[\alpha]M$, as in [24]. But this notation has to be given up since the use of square brackets is already imperatively taken by the *finite multisets*, which we will encounter constantly in the following. Another notation, used in [27], is to write $M\alpha$. However in our framework we find this notation not clear. The notation ${}_{\alpha}|M|$ should, instead, clearly show what is inside a "naming" and what is not.

Definition 1.2. *The reduction relation $\to$ of $\lambda\mu$-calculus is the contextual closure[2] of the union $\to_{\text{base}}$ of:*

$$(\lambda x.M)N \to_\lambda M\{N/x\}$$

$$(\mu\alpha._\beta|M|)N \to_\mu \mu\alpha.(_\beta|M|)_\alpha N$$

$$\mu\gamma._\alpha|\mu\beta._\eta|M|| \to_\rho \mu\gamma.(_\eta|M|\{\alpha/\beta\})$$

where $M\{N/x\}$ is the usual capture-free substitution of N for all free occurrences of x in M, $_\eta|M|\{\alpha/\beta\}$ replaces α for all the free occurrences of β in $_\eta|M|$, and $(M)_\alpha N$ is given by:

$(x)_\alpha N := x$
$(\lambda x.M)_\alpha N := \lambda x.(M)_\alpha N$
$(MP)_\alpha N := ((M)_\alpha N)((P)_\alpha N)$
$(\mu\beta._\gamma|M|)_\alpha N := \mu\beta._\gamma|(M)_\alpha N|$ *(if $\gamma \neq \alpha$)*
$(\mu\beta._\alpha|M|)_\alpha N := \mu\beta._\alpha|((M)_\alpha N)N|$.

We denote by $=_{\lambda\mu\rho}$ the equivalence induced by $\to$ on $\lambda\mu$.

The operation $(M)_\alpha N$ coincides with the substitution $M\{_\alpha|(\cdot)N|/_\alpha|\cdot|\}$: every named subterm $_\alpha|\cdot|$ of M gets substituted with the named term $_\alpha|(\cdot)N|$. Nevertheless, "morally", it is an application: each subterm of M named under α receives a copy of N to be applied to. This is why we chose the notation "$(M)_\alpha N$", which is reminiscent of the application of M to N, and the term $(M)_\alpha N$ is called the *named application of M to N through α*. Such notation is due to [29].

The reduction λ is the usual β-reduction (which we call "λ" in order to avoid confusion with names). The reduction ρ is just a renaming of names. The novelty is the μ-reduction which, in the following section, we are going to "linearise". There are many reductions that one can consider on the $\lambda\mu$-calculus; we chose to stick to those three because they are the ones considered in the original paper [24].

Theorem 1.3. *The $\lambda\mu$-calculus $(\lambda\mu, \to)$ is confluent.*

Proof. See proof of Theorem 4.1 of [25]. ◻

Lemma 1.4. *Every $\lambda\mu$-term M has the following shape:*

$$M = \lambda\vec{x}_1.\mu\alpha_1._{\beta_1}|\ldots\lambda\vec{x}_k.\mu\alpha_k._{\beta_k}|R\vec{Q}||$$

where R is either a variable, or a λ-redex or a μ-redex; furthermore, R, $\vec{Q}$, k, $\vec{x}_i$ and α_i are unique. R is called the head redex *of M if it is a $\lambda\mu$-redex, and it is called the* head variable *of M otherwise. The sequence $\lambda\vec{x}_1.\mu\alpha_1._{\beta_1}|\ldots\lambda\vec{x}_k.\mu\alpha_k._{\beta_k}|*||$ is called the* head *of M. Therefore, every $\lambda\mu\rho$-*normal *$\lambda\mu$-term M has a head variable, has no ρ-redexes in its head and (with the previous notations) $\vec{Q}$ are $\lambda\mu\rho$-normal $\lambda\mu$-terms.*

Other than what we already said in the introduction, we will not add more explanations of the logical and programming meaning of this calculus. Let us just add here the encoding of **callcc** in $\lambda\mu$-calculus: **callcc** $:= \lambda y.\mu\alpha._\alpha|y(\lambda x.\mu\delta._\alpha|x|)|$.

[2]The *contextual closure* of a binary relation $\mathcal{R}$ is the binary relation given by the set: $\{(C(\!|M|\!), C(\!|N|\!)) \mid M\mathcal{R}N \text{ and } C(\!|.|\!) \text{ single hole context}\}$.

2 Resource $\lambda\mu$-calculus

Recall that a *multiset* A on a set X is a map from X to $\mathbb{N}$. We use a multiplicative notation: the empty multiset is denoted with 1 and the union of two multisets A, B is denoted with $A * B$. The set of multisets on a set X is a monoid w.r.t. $*$, with neutral element 1. We denote with $!X$ the set of *finite multisets* on X, that is, multisets A with $X - A^{-1}(0)$ finite. Such an A will be as usual written as $A = [a_1, \ldots, a_k]$, with $A(a_i)$ repetitions for each a_i. We will sometimes write $m \pm A$ for $[m \pm a_1, \ldots, m \pm a_k]$ if $m \pm a_i$ happens to be defined.

Definition 2.1. *The set $\lambda\mu^{\mathrm{r}}$ of* resource $\lambda\mu$-terms *is given by:*

$$t ::= x \mid \lambda x.t \mid t_0[t_1, \ldots, t_n] \mid \mu\alpha._\beta|t|$$

where $[t_1, \ldots, t_n] \in !\lambda\mu^{\mathrm{r}}$ ($n \geq 0$), and it is called a bag. *Resource terms are considered up to renaming of bound variables and names.* Resource-contexts *are defined as expected. For v a variable or a name, the* degree $\deg_v(t) \in \mathbb{N}$ *of v in t, is defined as the number of free occurrences of v in t.*

The meaning of a resource sensitive application $(\lambda x.t)[\vec{u}]$ is to non-deterministically choose a way to associate each resource in the bag with exactly one occurrence of the argument x in t. It is thus natural to consider (formal) sums. If this association cannot be done without erasing or duplicating resources, then it annihilates to the empty sum 0. The operational semantics of a resource sensitive application $(\mu\alpha._\beta|t|)[\vec{u}]$ will be discussed in Definition 2.5.

Definition 2.2. *Call $2\langle\lambda\mu^{\mathrm{r}}\rangle$ the free module generated by $\lambda\mu^{\mathrm{r}}$ over the boolean semiring, which simply means the set of the* formal sums *of finitely many $\lambda\mu^{\mathrm{r}}$-terms, quotiented by commutativity, idempotency and associativity of $+$. An element of $2\langle\lambda\mu^{\mathrm{r}}\rangle$ will be called a* sum *(in fact, it is just a finite subset of $\lambda\mu^{\mathrm{r}}$). We extend the constructors of $\lambda\mu^{\mathrm{r}}$ to $2\langle\lambda\mu^{\mathrm{r}}\rangle$ by linearity, setting:*

$$\left(\sum_{i_0} t_{i_0}\right)\left[\sum_{i_1} t_{i_1}, \ldots, \sum_{i_n} t_{i_n}\right] := \sum_{i_0,\ldots,i_n} t_{i_0}[t_{i_1}, \ldots, t_{i_n}]$$

and analogous for $\lambda x. \sum_i t_i$ and $\mu\alpha._\beta\left|\sum_i t_i\right|$. We denote with 0 the empty sum. It is the neutral element for $+$ and the annihilating element for the above constructors (i.e. when it appears as any subterm, the whole term becomes 0).

Let us define now a reduction in $\lambda\mu^{\mathrm{r}}$ (or, better said, in $2\langle\lambda\mu^{\mathrm{r}}\rangle$). For this, we will need to divide a multiset into a certain number of "blocks". This notion already exists in the literature of combinatorics (see for example [3]).

Definition 2.3. *A* partition *(resp.* weak partition*) of a multiset $[\vec{u}]$ is a multiset $[[\vec{v}_1], \overset{(k\geq 1)}{\ldots}, [\vec{v}_k]]$ of* non empty *(resp. possibly empty) multisets such that $[\vec{u}] = [\vec{v}_1] * \cdots * [\vec{v}_k]$. A* composition *(resp.* weak composition *- w.c. for short) of a multiset $[\vec{u}]$ is a tuple $([\vec{v}_1], \ldots, [\vec{v}_k])$ of multisets s.t. $[[\vec{v}_1], \ldots, [\vec{v}_k]]$ is a partition (resp. weak partition) of $[\vec{u}]$.*

Observe that the empty bag 1 admits no partitions but admits infinite weak partitions: they are the multisets of shape $[1, \dots, 1]$ ($h \geq 1$ times 1). Here are some other examples: the set of all the weak partitions of the bag $[x]$ is $\{[[x]], [[x], 1], [[x], 1, 1], \dots\}$. The set of all weak partitions of $[x, x]$ is $\{[[x, x]], [[x], [x]], [[x, x], 1], [[x], [x], 1], [[x, x], 1, 1], [[x], [x], 1, 1], \dots\}$.

Definition 2.4. *Let $t \in \lambda\mu^{\mathrm{r}}$ and $[\vec{u}] \in\, !\lambda\mu^{\mathrm{r}}$. The* linear substitution *$t\langle[u_1, \dots, u_k]/x\rangle \in 2\langle\lambda\mu^{\mathrm{r}}\rangle$ is defined, as usual, in Figure 1. In order to linearise the μ-reduction we introduce the* linear named application *$\langle t\rangle_\alpha[\vec{u}] \in 2\langle\lambda\mu^{\mathrm{r}}\rangle$, defined in Figure 2* [3].

Remark that, thus, if $\deg_\alpha(t) = 0$ then $\langle t\rangle_\alpha 1 := t$ and $\langle t\rangle_\alpha[v, \vec{u}] := 0$; if $\deg_\alpha(t) =: d \neq 0$ then: $\langle t\rangle_\alpha[\vec{u}]$ is the sum $\sum t\left\{{}_\alpha|(\cdot)[\vec{s}^{\,1}]|/_{\alpha|\cdot|^{(1)}}, \dots, {}_\alpha|(\cdot)[\vec{s}^{\,d}]|/_{\alpha|\cdot|^{(d)}}\right\}$, where the sum is taken over all $([\vec{s}^{\,1}], \dots, [\vec{s}^{\,d}])$ w.c. of $[\vec{u}]$ of length d and ${}_\alpha|\cdot|^{(1)}, \dots, {}_\alpha|\cdot|^{(d)}$ is any fixed enumeration of the occurrences of α in t.

Definition 2.5. *Define a reduction $\to_{\mathrm{r}} \subseteq \lambda\mu^{\mathrm{r}} \times 2\langle\lambda\mu^{\mathrm{r}}\rangle$ as the resource-context closure of the union $\to_{\mathrm{base}^{\mathrm{r}}}$ of:*

$$(\lambda x.t)[\vec{u}] \to_{\lambda^{\mathrm{r}}} t\langle[\vec{u}]/x\rangle \qquad \mu\gamma.{}_\alpha|\mu\beta.{}_\eta|t|| \to_{\rho^{\mathrm{r}}} \mu\gamma.({}_\eta|t|\{\alpha/\beta\})$$
$$(\mu\alpha.{}_\beta|t|)[\vec{u}] \to_{\mu^{\mathrm{r}}} \mu\alpha.\langle{}_\beta|t|\rangle_\alpha[\vec{u}].$$

We extend it to all $2\langle\lambda\mu^{\mathrm{r}}\rangle \times 2\langle\lambda\mu^{\mathrm{r}}\rangle$ setting:

$$\to_{\mathrm{r}} := \{(t + \mathbb{S}, \mathbb{T} + \mathbb{S}) \mid t \to_{\mathrm{r}} \mathbb{T} \text{ and } t \notin \mathbb{S}\}.$$

Observe that the analogue of Lemma 1.4 holds for $\lambda\mu^{\mathrm{r}}$-terms (in particular we will use the notion of head variable/redex).

The work [1] is an example of how a resource calculus can be useful, as it enjoys strong properties such as linearity, strong normalisation and confluence. In the resource λ-calculus the last two properties are easy; as we are going to see, in our case they are more involved.

2.1 Strong normalisation

With $\to_{\lambda^{\mathrm{r}}}$ we erase exactly one λ, with $\to_{\rho^{\mathrm{r}}}$ we erase exactly one μ. With $\to_{\mu^{\mathrm{r}}}$ however, the situation is more subtle: we are not creating nor erasing λ's or μ's (which remain thus in constant number), but we are eventually making the reduct grow by creating an arbitrarily large number of new applications. However, in order to pass from the μ-redex $(\mu\alpha.{}_\beta|t|)[\vec{u}]$ to a reduct $t' \in \mu\alpha.\langle{}_\beta|t|\rangle_\alpha[\vec{u}]$, we: first, decompose $[\vec{u}]$ in several blocks; then, erase $[\vec{u}]$; finally, put each block *inside* a certain *named* subterm of ${}_\beta|t|$. We replaced thus a bag with many new bags which are at a "deeper depth". As we will see in Remark 2.7, it will be immediate to recognize that actually this depth is necessary bounded by the number of μ-occurrences in the term, which is invariant under $\to_{\mu^{\mathrm{r}}}$, so the former subtracted to the latter should decrease. Remark that in the case $[\vec{u}] = 1$, $\deg_\mu({}_\beta|t|) = 0$ we do not create new applications but we simply erase one already existing one, so we have to make sure our measure decreases in this case as well.

Definition 2.6. *Let t be a $\lambda\mu$-term and let b be an* occurrence *of a bag or of a subterm of t. The* depth *$d_t(b) \in \mathbb{N}$ of b in t is the number of* named *subterms of t containing b.*

Remark 2.7. *By definition of the grammar of the $\lambda\mu$-calculus there are as many named subterms of t as μ-abstractions in t, i.e. $\deg_\mu(t)$. So we must have: $d_t(b) \leq \deg_\mu(t)$.*

Definition 2.8. *Define the* multiset measure *$\mathrm{m}(t) \in\, !\mathbb{N}$ of a $\lambda\mu^{\mathrm{r}}$-term t as:*

$$\mathrm{m}(t) := \deg_\mu(t) - [\, d_t(b) \mid b \textit{ occurrence of bag in } t\,].$$

Remark 2.7 assures that $\mathrm{m}(t) \in\, !\mathbb{N}$ (and not in $!\mathbb{Z}$). This is crucial because it allows us to reason by induction w.r.t. the multiset order on it. The measure $\mathrm{m}(\cdot)$ is "almost" the good one for strong normalization:

Proposition 2.9. *If $t \to_{\mu^{\mathrm{r}}} t' + \mathbb{T}$ then $\mathrm{m}(t) > \mathrm{m}(t')$.*

Proof. If $t \to_{\mu^{\mathrm{r}}} t' + \mathbb{T}$ then $t = c(\!|(\mu\alpha.{}_\beta|s|)b_0|\!)$ and $t' = c(\!|h|\!)$ with $h \in \mu\alpha.\langle{}_\beta|s|\rangle_\alpha b_0$ and c a single-hole resource context. Call $k := \deg_\mu(t) = \deg_\mu(t')$ and consider $\deg_\alpha({}_\beta|s|) \in \mathbb{N}$. The are two cases:

- Case $\deg_\alpha({}_\beta|s|) = 0$. By definition of $\to_{\mu^{\mathrm{r}}}$ this is possible only if $b_0 = 1$ (otherwise $t \to_{\mu^{\mathrm{r}}} 0$) and $h = \mu\alpha.{}_\beta|s|$. So in t there are the exact same occurrences of bags as in t' and they are at the same depth, except for b_0 which is in t but not in t'. This means that $\mathrm{m}(t) = \mathrm{m}(t') * [\, k - d_t(b_0)\,] > \mathrm{m}(t')$.

- Case $\deg_\alpha({}_\beta|s|) =: n \geq 1$. Then:

$$h = \mu\alpha.{}_\beta|s|\left\{{}_\alpha|(\cdot)b_1|/_{\alpha|\cdot|^{(1)}}, \dots, {}_\alpha|(\cdot)b_n|/_{\alpha|\cdot|^{(n)}}\right\}$$

for a w.c. $(b_1, \dots, b_n)$ of b_0. So $\mathrm{m}(t') = k - A'$ and $\mathrm{m}(t) = k - A$, with A' and A respectively the multisets:

$$\begin{array}{l} B^c_{t'} * B^s_{t'} * [\, d_{t'}(b) \mid b \textit{ in a } v \in b_i \textit{ for an } i\,] * [\, d_{t'}(b_1), \dots, d_{t'}(b_n)\,] \\ B^c_t * B^s_t * \quad [\, d_t(b) \mid b \textit{ in a } v \in b_0]\,] \quad * \quad [\, d_t(b_0)\,], \end{array}$$

where we put $B^c_t := [\, d_t(b) \mid b \textit{ in } c]$ (and analogously for s, t'). Now for $i = 1, \dots, n$ we have: $d_{t'}(b_i) = d_{t'}(h) + d_h(b_i) > d_t(b_0)$ since as one sees from the expression of h, we have $d_{t'}(h) = d_t(b_0)$ and $d_h(b_i) > 0$. Also, it is easily understood that for all b occurring in c, or occurring in s, we have: $d_{t'}(b) = d_t(b)$. Finally, observe that since $(b_1, \dots, b_n)$ is a w.c. of b_0, then: b occurs in some $v \in b_0$ iff b occurs in some $v \in b_i$ for some i. And for all such b we have: $d_{t'}(b) = d_{t'}(v) + d_v(b) > d_t(v) + d_v(b) = d_t(b)$ since $d_{t'}(v) = d_{t'}(b_i) > d_t(b_0) = d_t(v)$. All these considerations precisely mean $\mathrm{m}(t) > \mathrm{m}(t')$. □

Analogously we find:

Proposition 2.10. *If $t \to_{\lambda^{\mathrm{r}}} t' + \mathbb{T}$ then $\mathrm{m}(t) > \mathrm{m}(t')$.*

However, only $\mathrm{m}(t)$ is not enough to prove strong normalization. In fact (reasoning similarly as before):

[3] The induction takes into account also the case of named terms ${}_\eta|t|$; this is done for technical reasons.

$$x\langle[v]/x\rangle = v \qquad y\langle 1/x\rangle = y \ \ (y \neq x) \qquad (\lambda y.t)\langle[\vec{u}]/x\rangle = \lambda y.t\langle[\vec{u}]/x\rangle$$
$$x\langle 1/x\rangle = x\langle[v,w,\vec{u}]/x\rangle = 0 \quad y\langle[v,\vec{u}]/x\rangle = 0 \ \ (y \neq x) \quad (\mu\alpha._{\beta}|t|)\langle[\vec{u}]/x\rangle = \mu\alpha._{\beta}|t\langle[\vec{u}]/x\rangle|$$

$$(t[v_1,\dots,v_n])\langle[\vec{u}]/x\rangle = \sum_{([\vec{s}^{\,0}],\dots,[\vec{s}^{\,n}]) \text{ w.c. of } [\vec{u}]} t\langle[\vec{s}^{\,0}]/x\rangle \left[\, v_1\langle[\vec{s}^{\,1}]]/x\rangle, \dots, v_n\langle[\vec{s}^{\,n}]/x\rangle \,\right].$$

Figure 1. Definition of linear substitution

$$\langle x\rangle_\alpha[v,\vec{u}] = 0 \qquad \langle x\rangle_\alpha 1 = x \qquad \langle_\eta|t|\rangle_\alpha[\vec{u}] = {}_\eta|\langle t\rangle_\alpha[\vec{u}]| \quad (\textit{if } \eta \neq \alpha)$$

$$\langle\mu\gamma._\eta|t|\rangle_\alpha[\vec{u}] = \mu\gamma.\langle_\eta|t|\rangle_\alpha[\vec{u}] \quad \langle\lambda y.t\rangle_\alpha[\vec{u}] = \lambda y.\langle t\rangle_\alpha[\vec{u}] \quad \langle_\alpha|t|\rangle_\alpha[\vec{u}] = \sum_{([\vec{w}^{\,1}],[\vec{w}^{\,2}]) \text{ w.c. of } [\vec{u}]} {}_\alpha|\left(\langle t\rangle_\alpha[\vec{w}^{\,1}]\right)[\vec{w}^{\,2}]|$$

$$\langle t[v_1,\dots,v_n]\rangle_\alpha[\vec{u}] = \sum_{([\vec{w}^{\,0}],\dots,[\vec{w}^{\,n}]) \text{ w.c. of } [\vec{u}]} \left(\langle t\rangle_\alpha[\vec{w}^{\,0}]\right) \left[\, \langle v_1\rangle_\alpha[\vec{w}^{\,1}], \dots, \langle v_n\rangle_\alpha[\vec{w}^{\,n}] \,\right].$$

Figure 2. Definition of linear named application

Proposition 2.11. *If $t \to_{\rho^{\mathrm{r}}} t' + \mathbb{T}$ then $\mathrm{m}(t) \geq \mathrm{m}(t')$, and there are cases in which the equality holds, such as (for $\beta \neq \eta$): $\mathrm{m}(\mu\gamma._\alpha|\mu\beta._\eta|x||) = 1 = \mathrm{m}(\mu\gamma._\eta|x|)$ with $\mu\gamma._\alpha|\mu\beta._\eta|x|| \to_{\rho^{\mathrm{r}}} \mu\gamma._\eta|x|$.*

That is why, in order to get a strongly normalising measure, we add another component:

Definition 2.12. *We define the measure:*

$$\widetilde{\mathrm{m}}(t) := (\mathrm{m}(t), \deg_\mu(t)) \in !\mathbb{N} \times \mathbb{N}$$

ordered by the (well-founded) lexicographic order.

Corollary 2.13 (SN). *If $t \to_{\mathrm{r}} t' + \mathbb{T}$ then $\widetilde{\mathrm{m}}(t) > \widetilde{\mathrm{m}}(t')$. Therefore, the resource reduction $\to_{\mathrm{r}}$ on sums is strongly normalising.*

Proof. The only case in which $\mathrm{m}(\cdot)$ may remain constant is along a ρ^{r}-reduction, but in this case $\deg_\mu(t)$ strictly decreases. □

Before turning to the confluence, let us see some properties of the measure $\mathrm{m}(\cdot)$ that we will use in the following.

Lemma 2.14. *Let $c = c(\!|\xi|\!)$ be a single-hole context and t a $\lambda\mu$-term. Then: $\mathrm{m}(c(\!|t|\!)) \geq \mathrm{m}(t)$.*

Proof. We have $\mathrm{m}(c(\!|t|\!)) = A * [\, \deg_\mu(c(\!|t|\!)) - d_{c(\!|t|\!)}(b) \mid b \text{ in } t \,]$ where $A := [\, \deg_\mu(c(\!|t|\!)) - d_{c(\!|t|\!)}(b) \mid b \text{ in } c \,]$. But $\deg_\mu(c(\!|t|\!)) = \deg_\mu(c) + \deg_\mu(t)$ and, for all occurrence b in t, we have: $d_{c(\!|t|\!)}(b) = d_t(b) + d_c(\xi) \leq d_t(b) + \deg_\mu(c)$. Thus, for all occurrence b of bag in t, we have: $\deg_\mu(c(\!|t|\!)) - d_{c(\!|t|\!)}(b) \geq \deg_\mu(t) - d_t(b)$ and this last integer is exactly a generic element of $\mathrm{m}(t)$ (if it is non-empty). Hence $\mathrm{m}(c(\!|t|\!)) \geq A * \mathrm{m}(t) \geq \mathrm{m}(t)$. □

However, there are cases in which $\mathrm{m}(c(\!|t|\!)) = \mathrm{m}(t)$ even if $c \neq \xi$. For example, taking $c = \lambda x.\xi$ one has $\mathrm{m}(c(\!|t|\!)) = 1 = \mathrm{m}(t)$ for all $t \in \lambda\mu^{\mathrm{r}}$ *not* containing any bags. This is exactly why, in the following, we will consider a slightly different size, called $\boldsymbol{ms}$ (defined in Corollary 2.16).

Lemma 2.15. *Let $c = c(\!|\xi|\!)$ be a single-hole resource context and $t, s \in \lambda\mu$. Then:*

1. $\mathrm{m}(c(\!|t|\!)) = (\deg_\mu(t) + \mathrm{m}(c)) * ((\deg_\mu(c) - d_c(\xi) + \mathrm{m}(t))$.
2. *If* $\deg_\mu(s) \leq \deg_\mu(t)$ *and* $\mathrm{m}(s) < \mathrm{m}(t)$, *then* $\mathrm{m}(c(\!|s|\!)) < \mathrm{m}(c(\!|t|\!))$.

Proof sketch. Easily checked, thanks to the clear fact that if b is the occurrence of a bag in c, then $d_{c(\!|t|\!)}(b) = d_c(b)$. □

In the following, we will need a strong normalising measure which, in addition, satisfies the properties of the following Corollary 2.16. However, we have seen with some lines above that $\widetilde{\mathrm{m}}(\cdot)$ is not adapted for that. This is why we operate a last slight modification. First, let us consider the size $\mathbf{sz}(t) \in \mathbb{N}_{\geq 1}$ of resource $\lambda\mu$-terms: $\mathbf{sz}(x) := 1$, $\mathbf{sz}(\lambda x.t) := 1 + \mathbf{sz}(t) =: \mathbf{sz}(\mu\alpha._\beta|s|)$, $\mathbf{sz}(t_0[t_1,\dots,t_k]) := 1 + k + \sum_{i=0}^{k} \mathbf{sz}(t_i)$. Of course $\mathbf{sz}(t) = 1$ iff t is a variable, and for all c single-hole context, $\mathbf{sz}(c(\!|t|\!)) \geq \mathbf{sz}(t)$ where the equality holds iff $c = \xi$.

Corollary 2.16. *Define a measure $\boldsymbol{ms}(\cdot)$ of $\lambda\mu$-terms as:*

$$\boldsymbol{ms}(t) := (\widetilde{\mathrm{m}}(t), \mathbf{sz}(t)) \in !\mathbb{N} \times \mathbb{N} \times \mathbb{N}$$

ordered lexicographically (and thus well-founded). Then:

1. *t is a variable iff $\boldsymbol{ms}(t)$ takes its minimal value $(1, 0, 1)$.*
2. *For all single-hole context $c = c(\!|\xi|\!)$, we have $\boldsymbol{ms}(c(\!|t|\!)) \geq \boldsymbol{ms}(t)$, and the equality holds iff $c = \xi$.*
3. *If $t \to_{\mathrm{r}} t' + \mathbb{T}$ then $\boldsymbol{ms}(t) > \boldsymbol{ms}(t')$.*

2.2 Confluence

Due to the presence of three different reductions, the confluence or our resource $\lambda\mu$-calculus is not easy. Another difficulty is raised from the fact that we placed ourselves in a qualitative setting, that is, with idempotent sums, so that we cannot always reduce a sum component-wise. This is why we split the problem of the confluence in two steps: first, we show that the *quantitative* resource $\lambda\mu$-calculus (that is, where sum is *not* idempotent, and thus coefficients matter) is confluent (Section 2.2.1); second, we show that its confluence implies the confluence of the calculus with no coefficients (Section 2.2.2). Before all that, let us precisely explain the notion of quantitative resource calculus:

Definition 2.17. *The* quantitative resource $\lambda\mu$-calculus $\mathbb{N}\langle\lambda\mu^{\mathrm{r}}\rangle$ *is built as the qualitative one ($2\langle\lambda\mu^{\mathrm{r}}\rangle$, Definition 2.2) except for taking now "+"* non-idempotent. *We define the three base-case reductions $\to^+_{\lambda^{\mathrm{r}}}, \to^+_{\mu^{\mathrm{r}}}, \to^+_{\rho^{\mathrm{r}}}$ in $\lambda\mu^{\mathrm{r}} \times \mathbb{N}\langle\lambda\mu^{\mathrm{r}}\rangle$: the reduction $\to^+_{\rho^{\mathrm{r}}}$ is defined as usual, while $\to^+_{\lambda^{\mathrm{r}}}$ and $\to^+_{\mu^{\mathrm{r}}}$ are defined as in Definition 2.5, except for the fact that the linear substitution and linear named application are replaced with a modified version of them, denoted respectively $t\langle[\vec{u}]/x\rangle^+$ and $\langle t\rangle^+_\alpha[\vec{u}]$, and defined in the next Definition 2.19. The contextual union of the base-reductions $\to^+_{\lambda^{\mathrm{r}}}, \to^+_{\mu^{\mathrm{r}}}, \to^+_{\rho^{\mathrm{r}}}$ forms a reduction $\to^+_{\mathrm{r}}$ on $\lambda\mu^{\mathrm{r}} \times \mathbb{N}\langle\lambda\mu^{\mathrm{r}}\rangle$ which is extended to all $\mathbb{N}\langle\lambda\mu^{\mathrm{r}}\rangle \times \mathbb{N}\langle\lambda\mu^{\mathrm{r}}\rangle$ by taking $\{(t+\mathbb{S}, \mathbb{T}+\mathbb{S}) \mid t \to^+_{\mathrm{r}} \mathbb{T}\}$ (remark that we dropped the annoying condition "$t \notin \mathbb{S}$", since now coefficients matter; it is the main reason why we turn to this calculus).*

Notation 2.18. *If $[u_1, \dots, u_k]$ is a bag – with the written enumeration of (possibly multiple) elements – and W is a function $W : \{1, \dots, k\} \longrightarrow I =: \{i_0 < \dots < i_n\}$, we will sometimes denote it by $W : (u_1, \dots, u_k) \longrightarrow I$, or by $W : (\vec{u}) \longrightarrow I$. When we use such notation we mean that W generates the w.c. $([u_j \mid j \in W^{-1}(i_0)], \dots, [u_j \mid j \in W^{-1}(i_n)])$ of $[u_1, \dots, u_k]$, and denoted by $([\vec{w}^{\,i_0}], \dots, [\vec{w}^{\,i_n}])$. In the case $[\vec{u}] = 1$, we write $W : () \longrightarrow I$ and we say that there is exactly one w.c. generated by W, namely $(1, \overset{(n+1 \text{ times})}{\dots}, 1)$.*

Definition 2.19. *The quantitative version $t\langle[\vec{u}]/x\rangle^+$ of the linear substitution is defined exactly as in Figure 1 but by replacing the sum on all the $([\vec{w}^{\,0}], \dots, [\vec{w}^{\,n}])$ w.c. of $[\vec{u}]$ with the sum on all $W : (\vec{u}) \longrightarrow \{0, \dots, n\}$, and by taking the above w.c.'s as the ones generated by W. The quantitative version $\langle t\rangle^+_\alpha[\vec{u}]$ of the linear named application is defined exactly as in Figure 2 but by replacing, in the case of an application, the sum on all the $([\vec{w}^{\,0}], \dots, [\vec{w}^{\,n}])$ w.c. of $[\vec{u}]$ with the sum on all $W : (\vec{u}) \longrightarrow \{0, \dots, n\}$, and by taking the above w.c.'s as the ones generated by W. Analogously for the case of a named term, where we use $W : (\vec{u}) \longrightarrow \{1, 2\}$.*

For instance: $(\mu\alpha.{}_\alpha|\mu\eta.{}_\alpha|x||)[y,y] \to^+_{\mathrm{r}} \mu\alpha.{}_\alpha|(\mu\eta.{}_\alpha|x1|)[y,y]| + 2\,\mu\alpha.{}_\alpha|(\mu\eta.{}_\alpha|x[y]|)[y]| + \mu\alpha.{}_\alpha|(\mu\eta.{}_\alpha|x[y,y]|)1|$.

In the following, $\mathrm{supp}(\mathbb{T}) \in 2\langle\lambda\mu^{\mathrm{r}}\rangle$ is the *support* of a $\mathbb{T} \in \mathbb{N}\langle\lambda\mu^{\mathrm{r}}\rangle$, that is, the *set* of its addends (with no coefficients: $\mathrm{supp}(\mathbb{T})$ is $\mathbb{T}$ when considered with an idempotent "+").

Remark 2.20. *It is clear by the definitions that if, for $t \in \lambda\mu^{\mathrm{r}}$, one has $t \to_{\mathrm{r}} \mathbb{T}$ (in $2\langle\lambda\mu^{\mathrm{r}}\rangle$) and $t \to^+_{\mathrm{r}} \mathbb{S}$ (in $\mathbb{N}\langle\lambda\mu^{\mathrm{r}}\rangle$) by reducing the* same *redex, then $\mathrm{supp}(\mathbb{S}) = \mathbb{T}$. That is, the two reductions only differ for the coefficients. Said differently, the qualitative substitutions $t\langle[\vec{u}]/x\rangle$ and $\langle t\rangle_\alpha[\vec{u}]$ are just the quantitative substitutions $t\langle[\vec{u}]/x\rangle^+$ and $\langle t\rangle^+_\alpha[\vec{u}]$ taken with boolean coefficients.*

Remark 2.21. *Using the fact that the reduction $\to_{\mathrm{r}}$ is strongly normalising in $\lambda\mu^{\mathrm{r}}$ (Corollary 2.13), we can prove that the reduction $\to^+_{\mathrm{r}}$ is strongly normaling in $\mathbb{N}\langle\lambda\mu^{\mathrm{r}}\rangle$. It suffices to extend the strongly normalising measure $\widetilde{\mathrm{m}}(\cdot)$ of $\lambda\mu^{\mathrm{r}}$ (Corollary 2.12) to $\mathbb{N}\langle\lambda\mu^{\mathrm{r}}\rangle$ by setting $\widetilde{\mathrm{m}}(\mathbb{T}) := [\widetilde{\mathrm{m}}(t) \mid t \in \mathbb{T}] \in\ !(!\mathbb{N} \times \mathbb{N})$, and use the multiset order.*

Remark 2.22 (Embedding inside the differential $\lambda\mu$-calculus)**.** *In [28], Vaux defines a differential $\lambda\mu$-calculus, let us call it $(\lambda\mu^\partial, \to_\partial)$ in this remark, and proves its confluence. Our resource $\lambda\mu$-calcului $2\langle\lambda\mu^{\mathrm{r}}\rangle$ and $\mathbb{N}\langle\lambda\mu^{\mathrm{r}}\rangle$ are strictly related to it, as they translate into $\lambda\mu^\partial$ via[4] $(\cdot)^\partial : \lambda\mu^{\mathrm{r}} \longrightarrow \lambda\mu^\partial$ defined as: $x^\partial := x$, $(\lambda x.t)^\partial := \lambda x.t^\partial$, $(\mu\alpha.{}_\beta|t|)^\partial := \mu\alpha.{}_\beta|t^\partial|$, $(t[u_1, \dots, u_k])^\partial := \left(\mathsf{D}^k\, t^\partial \bullet (u_1^\partial, \dots, u_k^\partial)\right) 0$. We can extend it to sums, both in $2\langle\lambda\mu^{\mathrm{r}}\rangle$ and in $\mathbb{N}\langle\lambda\mu^{\mathrm{r}}\rangle$, by linearity. In the qualitative case (that is, if we consider $(\cdot)^\partial : 2\langle\lambda\mu^{\mathrm{r}}\rangle \longrightarrow \lambda\mu^\partial$), it is not a well-behaved embedding, because it does not preserve reductions. On the contrary, it does in the quantitative case (that is, if we consider $(\cdot)^\partial : \mathbb{N}\langle\lambda\mu^{\mathrm{r}}\rangle \longrightarrow \lambda\mu^\partial$), in the sense that: if $t \to^+_{\lambda\mu^{\mathrm{r}}} \mathbb{T}$ in $\mathbb{N}\langle\lambda\mu^{\mathrm{r}}\rangle$, then $t^\partial \twoheadrightarrow_\partial \mathbb{T}^\partial$ in $\lambda\mu^\partial$ ("$\twoheadrightarrow$" is the reflexive transitive closure of $\to$).*

One may wonder if it is possible to use the confluence of $(\lambda\mu^\partial, \to_\partial)$ to infer the confluence of our calculi. In fact, it is possible to show that the local confluence of $\to^+_{\lambda\mu^{\mathrm{r}}}$ follows from the confluence of $\to_\partial$. However, as the reader has probably noticed, we only talked about $\to^+_{\lambda\mu^{\mathrm{r}}}$, and not about the whole $\to^+_{\mathrm{r}} = \to^+_{\lambda\mu^{\mathrm{r}}} \cup \to^+_{\rho^{\mathrm{r}}}$. This is simply because in [28] the ρ-reduction is not considered. Remark that, even if it is possible to prove the confluence of $\to_{\rho^{\mathrm{r}}}$ by itself, we cannot use it in order to entail the confluence of $\to^+_{\mathrm{r}} = \to^+_{\lambda\mu^{\mathrm{r}}} \cup \to^+_{\rho^{\mathrm{r}}}$ by invoking the well-known Hindley-Rosen lemma. This is because $\to^+_{\rho^{\mathrm{r}}}$ and $\to^+_{\lambda\mu^{\mathrm{r}}}$ do not *commute, as the following example shows (where $\gamma \neq \eta \neq \alpha$): $\mu\alpha.{}_\alpha|(\mu\gamma.{}_\eta|x|)1|\ {}_{\mu^{\mathrm{r}}}{}^{+}\!\!\leftarrow (\mu\alpha.{}_\alpha|\mu\gamma.{}_\eta|x||)1 \to^+_{\rho^{\mathrm{r}}} (\mu\alpha.{}_\eta|x|)1 \to^+_{\rho^{\mathrm{r}}} \mu\alpha.{}_\eta|x|$, but $\mu\alpha.{}_\alpha|(\mu\gamma.{}_\eta|x|)1| \not\to^+_{\rho^{\mathrm{r}}} \mu\alpha.{}_\eta|x|$. In the previous "non-reduction", the blocked ρ-redex can be unblocked by performing a μ-reduction (and the diagram closes). O. Laurent suggests (private communication) that we could still use the confluence of $(\lambda\mu^\partial, \to_\partial)$ in order to obtain the confluence of $\to^+_{\mathrm{r}} = \to^+_{\lambda\mu^{\mathrm{r}}} \cup \to^+_{\rho^{\mathrm{r}}}$ passing through a factorization lemma: if $t \twoheadrightarrow^+_{\mathrm{r}} \mathbb{T}$ then $t \twoheadrightarrow^+_{\lambda\mu^{\mathrm{r}}} \mathbb{T}' \twoheadrightarrow^+_{\rho^{\mathrm{r}}} \mathbb{T}$, for some $\mathbb{T}'$.*

2.2.1 Confluence of $(\mathbb{N}\langle\lambda\mu^{\mathrm{r}}\rangle, \to^+_{\mathrm{r}})$.

We present here a proof which essentially consists in closing the diagrams of all the possible critical pairs.

[4]Here we are considering that the reader knowns the syntax of $\lambda\mu^\partial$.

Remark 2.23. *We can extend the definition of linear substitution and linear named application to sums by linearity. Analogously, the renaming of a sum $\mathbb{T}\{\alpha/\beta\}$ is defined componentwise. With these definitions in place one checks that base-step-reduction lifts to sums, i.e.* $(\mu\alpha.{}_{\beta}|\mathbb{T}|)\,[\vec{\mathbb{U}}] \twoheadrightarrow^{+}_{\mu^{\mathrm{r}}} \mu\alpha.\,\langle\mathbb{T}\rangle^{+}_{\alpha}\,[\vec{\mathbb{U}}]$ *and analogously for* $(\lambda x.\mathbb{T})\,[\vec{\mathbb{U}}]$ *and* $\mu\alpha.{}_{\beta}|\mu\gamma.{}_{\eta}|\mathbb{T}||$. *One can also check that $\rightarrow^{+}_{\mathrm{r}}$ on $\mathbb{N}\langle\lambda\mu^{\mathrm{r}}\rangle$ is contextual.*

Notation 2.24. *In this section we will sometimes use the following notation: for α, β, η names, we set $\delta^{\alpha}_{\eta}(\beta)$ to be α if $\beta = \eta$, or η otherwise.*

The following is the crucial technical lemma.

Lemma 2.25. *Let $t, s \in \lambda\mu^{\mathrm{r}}$, x a variable, α, β names and $[\vec{u}]$ a bag. If $s \rightarrow^{+}_{\mathrm{r}} \mathbb{S}$ then:*

1. $s\{\alpha/\beta\} \twoheadrightarrow^{+}_{\mathrm{r}} \mathbb{S}\{\alpha/\beta\}$
2. $t\langle[s,\vec{u}]/x\rangle^{+} \twoheadrightarrow^{+}_{\mathrm{r}} t\langle[\mathbb{S},\vec{u}]/x\rangle^{+}$
3. $s\langle[\vec{u}]/x\rangle^{+} \twoheadrightarrow^{+}_{\mathrm{r}} \mathbb{S}\langle[\vec{u}]/x\rangle^{+}$
4. $\langle t\rangle^{+}_{\alpha}[s,\vec{u}] \twoheadrightarrow^{+}_{\mathrm{r}} \langle t\rangle^{+}_{\alpha}[\mathbb{S},\vec{u}]$
5. $\mu\alpha.\langle{}_{\beta}|t|\rangle^{+}_{\alpha}[s,\vec{u}] \twoheadrightarrow^{+}_{\mathrm{r}} \mu\alpha.\langle{}_{\beta}|t|\rangle^{+}_{\alpha}[\mathbb{S},\vec{u}]$
6. $\langle s\rangle^{+}_{\alpha}[\vec{u}] \twoheadrightarrow^{+}_{\mathrm{r}} \langle\mathbb{S}\rangle^{+}_{\alpha}[\vec{u}]$.
7. $\mu\alpha.\langle{}_{\beta}|s|\rangle^{+}_{\alpha}[\vec{u}] \twoheadrightarrow^{+}_{\mathrm{r}} \mu\alpha.\langle{}_{\beta}|\mathbb{S}|\rangle^{+}_{\alpha}[\vec{u}]$.

Before proving it, let us remark that, in the qualitative setting, it is false. For instance, if $s \rightarrow_{\mathrm{r}} s'$, then $(x[x])\langle[s,s]/x\rangle = s[s] \not\twoheadrightarrow_{\mathrm{r}} s[s'] + s'[s] = (x[x])\langle[s,s']/x\rangle$. In the quantitative case, instead, $(x[x])\langle[s,s]/x\rangle^{+} = 2\,s[s] \twoheadrightarrow^{+}_{\mathrm{r}} s[s'] + s'[s] = (x[x])\langle[s,s']/x\rangle^{+}$.

Proof sketch of Lemma 2.25. 1). Induction on s. The only interesting cases are:

- Case $s = \mu\gamma.{}_{\eta}|s'|$: we have two subcases: Subcase $s \rightarrow^{+}_{\mathrm{r}} \mathbb{S}$ is performed by reducing s': easy by inductive hypothesis. Subcase $s' = \mu\gamma'.{}_{\eta'}|s''|$ and $s \rightarrow^{+}_{\mathrm{r}} \mathbb{S}$ is performed by reducing its leftmost ρ-redex: then $s = \mu\gamma.{}_{\eta}|\mu\gamma'.{}_{\eta'}|s''||$, $\mathbb{S} = \mu\gamma.{}_{\eta'}|s''|\{\eta/\gamma'\}$ and we have the four sub-subcases $\eta = \beta$ and $\eta' = \beta$, or $\eta = \beta$ and $\eta' \neq \beta$, or $\eta \neq \beta$ and $\eta' = \beta$, or $\eta \neq \beta$ and $\eta' \neq \beta$. They are all similar, let us only show the second one, for which we have:

$$\begin{aligned}\mathbb{S}\{\alpha/\beta\} = {}& \mu\gamma.{}_{\eta'}|s''|\{\alpha/\beta,\alpha/\gamma'\} = \mu\gamma.{}_{\delta^{\alpha}_{\eta'}(\gamma')}|s''\{\alpha/\beta,\alpha/\gamma'\}|\\ s\{\alpha/\beta\} = {}& \mu\gamma.{}_{\alpha}|\mu\gamma'.{}_{\eta'}|s''\{\alpha/\beta\}|| \rightarrow^{+}_{\rho} \mu\gamma.{}_{\eta'}|s''\{\alpha/\beta\}|\{\alpha/\gamma'\}\\ = {}& \mu\gamma.{}_{\delta^{\alpha}_{\eta'}(\gamma')}|s''\{\alpha/\beta,\alpha/\gamma'\}| = \mathbb{S}\{\alpha/\beta\}.\end{aligned}$$

- Case $s = s'[\vec{v}]$: we have four subcases depending on how the reduction $s \rightarrow^{+}_{\mathrm{r}} \mathbb{S}$ is performed. The only interesting one is the subcase $s' = \mu\gamma.{}_{\eta}|s''|$ and $s \rightarrow^{+}_{\mathrm{r}} \mathbb{S}$ is performed by reducing the μ-redex s, for which we have: $\mathbb{S} = \mu\gamma.\langle{}_{\eta}|s''|\rangle^{+}_{\gamma}[\vec{v}]$ and $s\{\alpha/\beta\} = (\mu\gamma.{}_{\delta^{\alpha}_{\eta}(\beta)}|s''\{\alpha/\beta\}|)[\vec{v}\{\alpha/\beta\}] \rightarrow_{\mu^{\mathrm{r}}}$-reduces to $\mu\gamma.\langle{}_{\delta^{\alpha}_{\eta}(\beta)}|s''\{\alpha/\beta\}|\rangle^{+}_{\gamma}[\vec{v}\{\alpha/\beta\}]$ which in turn coincides with the sum $\mu\gamma.\langle{}_{\eta}|s''|\{\alpha/\beta\}\rangle^{+}_{\gamma}[\vec{v}\{\alpha/\beta\}] = \mathbb{S}\{\alpha/\beta\}$.

(2). Induction on t. The only non-trivial case is when t is $v_0[v_1,\dots,v_n]$. In this case we can write $t\langle[s,\vec{u}]/x\rangle^{+}$ as:

$$\sum_{W}\sum_{j=0}^{n}(\,v_0\langle[\vec{w}^{\,0}] * [s]^{j}_{0}/x\rangle\,)\,[\dots,v_i\langle[\vec{w}^{\,i}] * [s]^{j}_{i}/x\rangle,\dots]$$

where $W : (\vec{u}) \longrightarrow \{1,\dots,n\}$ and we put $[s]^{j}_{i}$ to be the singleton multiset $[s]$ if $i = j$, and the empty mulitset 1 if $i \neq j$.. Fix now a $W : (\vec{u}) \longrightarrow \{1,\dots,n\}$ (together with its generated w.c.) and consider each of the $n+1$ elements of the sum on j. We write the case for $j = 0$, but the other cases are exactly the same. Since $j = 0$, the element is $(\,v_0\langle[\vec{w}^{\,0}] * [s]/x\rangle\,)\,[\dots,v_i\langle[\vec{w}^{\,i}]/x\rangle,\dots]$ and by inductive hypothesis it $\twoheadrightarrow^{+}_{\mathrm{r}}$-reduces to $(\,v_0\langle[\vec{w}^{\,0}] * [\mathbb{S}]/x\rangle\,)\,[\dots,v_i\langle[\vec{w}^{\,i}]/x\rangle,\dots]$. Now summing up all the elements for $j = 0,\dots,n$ and $W : (\vec{u}) \longrightarrow \{1,\dots,n\}$ we obtain the following sum:

$$\sum_{W}\sum_{j=0}^{n}(\,v_0\langle[\vec{w}^{\,0}] * [\mathbb{S}]^{j}_{0}/x\rangle\,)[\dots,v_i\langle[\vec{w}^{\,i}] * [\mathbb{S}]^{j}_{i}/x\rangle,\dots].$$

which can be shown to be the desired $(\,v_0[v_1,\dots,v_n]\,)\,\langle[\mathbb{S},\vec{u}]/x\rangle^{+}$.

(3). Induction on s. We only show the case $s = \mu\alpha.{}_{\beta}|s'|$, which splits in two subcases: the subcase where $s \rightarrow^{+}_{\mathrm{r}} \mathbb{S}$ is performed by reducing s' is immediate. The subcase where $s' = \mu\gamma.{}_{\eta}|s''|$ and $s \rightarrow^{+}_{\mathrm{r}} \mathbb{S}$ is performed by reducing its leftmost ρ-redex goes as follows: we have $\mathbb{S} = \mu\alpha.{}_{\eta}|s''|\{\beta/\gamma\}$ and
$s\langle[\vec{u}]/x\rangle^{+} = \mu\alpha.{}_{\beta}|\mu\gamma.{}_{\eta}|s''\langle[\vec{u}]/x\rangle^{+}||$
$\rightarrow_{\rho^{\mathrm{r}}} \mu\alpha.{}_{\eta}|s''\langle[\vec{u}]/x\rangle^{+}|\{\beta/\gamma\} = \mu\alpha.{}_{\eta}|s''|\langle[\vec{u}]/x\rangle^{+}\{\beta/\gamma\}$
$= \mu\alpha.{}_{\eta}|s''|\{\beta/\gamma\}\langle[\vec{u}]/x\rangle^{+} = \mathbb{S}\langle[\vec{u}]/x\rangle^{+}$.

(4). Induction on t. Similar to point (2).

(5). It is easy discriminating the cases $\alpha = \beta$ and $\alpha \neq \beta$ and concluding by point (4).

(6). Induction on $s \in \lambda\mu$. The only interesting cases are:

- Case $s = \mu\beta.{}_{\gamma}|s'|$. We have two subcases: the subcase where $s \rightarrow^{+}_{\mathrm{r}} \mathbb{S}$ is performed by reducing s', so $\mathbb{S} = \mu\beta.{}_{\gamma}|\mathbb{S}'|$ with $s' \rightarrow_{\mathrm{r}} \mathbb{S}'$, is easy by inductive hypothesis (however remark that we *cannot* immediately apply the inductive hypothesis on ${}_{\gamma}|s'|$, simply because the named term ${}_{\gamma}|s'| \notin \lambda\mu^{\mathrm{r}}$). The subcase where $s' = \mu\gamma'.{}_{\eta}|s''|$ (with $\gamma \neq \gamma'$) and $s \rightarrow^{+}_{\mathrm{r}} \mathbb{S}$ is performed by reducing its leftmost ρ-redex goes as follows: we have $\mathbb{S} = \mu\beta.{}_{\eta}|s''|\{\gamma/\gamma'\}$ and we split in two sub-subcases depending whether $\alpha \neq \gamma$ or $\alpha = \gamma$. Let us only show this last sub-subcase: We have (putting $W : (\vec{u}) \longrightarrow \{1,2\}$):

$$\begin{array}{lll}\langle s\rangle^{+}_{\alpha}[\vec{u}] & = & \sum_{W}\mu\beta.{}_{\alpha}|\,(\,\mu\gamma'.\langle{}_{\eta}|s''|\rangle^{+}_{\alpha}[\vec{w}^{\,1}]\,)\,[\vec{w}^{\,2}]|\\ & \twoheadrightarrow^{+}_{\mu^{\mathrm{r}}} & \sum_{W}\mu\beta.{}_{\alpha}|\mu\gamma'.\langle\langle{}_{\eta}|s''|\rangle^{+}_{\alpha}[\vec{w}^{\,1}]\rangle^{+}_{\gamma'}[\vec{w}^{\,2}]|\\ & \twoheadrightarrow^{+}_{\rho^{\mathrm{r}}} & \sum_{W}\mu\beta.\langle\langle{}_{\eta}|s''|\rangle^{+}_{\alpha}[\vec{w}^{\,1}]\rangle^{+}_{\gamma'}[\vec{w}^{\,2}]\,\{\alpha/\gamma'\}\\ & = & \sum_{W}\langle\langle\mu\beta.{}_{\eta}|s''|\rangle^{+}_{\alpha}[\vec{w}^{\,1}]\rangle^{+}_{\gamma'}[\vec{w}^{\,2}]\,\{\alpha/\gamma'\}\\ & = & \langle\mu\beta.{}_{\eta}|s''|\,\{\alpha/\gamma'\}\rangle^{+}_{\alpha}[\vec{u}]\\ & = & \langle\mathbb{S}\rangle^{+}_{\alpha}[\vec{u}].\end{array}$$

- Case $s = s'[v_1,\dots,v_n]$: we have four subcases depending on how the reduction $s \rightarrow^{+}_{\mathrm{r}} \mathbb{S}$ is performed. We only show the one in which $s' = \mu\gamma.{}_{\eta}|s''|$ (with $\gamma \neq \alpha$) and $s \rightarrow^{+}_{\mathrm{r}} \mathbb{S}$ is performed by reducing the μ-redex s. In this subcase we have

$\mathbb{S} = \mu\gamma.\langle_\eta|s''|\rangle^+_\gamma[\vec{v}]$ and (putting $W : (\vec{u}) \longrightarrow \{0, \ldots, n\}$):

$$\begin{array}{rcl} \langle s\rangle^+_\alpha[\vec{u}] & = & \sum\limits_W (\langle \mu\gamma.{}_\eta|s''|\rangle^+_\alpha[\vec{w}^{\,0}])\,[\ldots, \langle v_i\rangle^+_\alpha[\vec{w}^{\,i}], \ldots] \\ & \twoheadrightarrow^+_{\mu^{\mathrm{r}}} & \mu\gamma.\sum\limits_W \langle\langle_\eta|s''|\rangle^+_\alpha[\vec{w}^{\,0}]\rangle^+_\gamma[\ldots, \langle v_i\rangle^+_\alpha[\vec{w}^{\,i}], \ldots] \\ & = & \mu\gamma.\langle\langle_\eta|s''|\rangle^+_\gamma[\vec{v}]\rangle^+_\alpha[\vec{u}] \\ & = & \langle\mathbb{S}\rangle^+_\alpha[\vec{u}]. \end{array}$$

(7). It is immediate by discriminating the cases $\alpha = \beta$ and $\alpha \neq \beta$ and then concluding by point (6). ☐

Proposition 2.26. *The reduction $\to^+_{\mathrm{r}}$ is locally confluent in $\mathbb{N}\langle\lambda\mu^{\mathrm{r}}\rangle$.*

Proof sketch. We show, by induction on a single-hole resource context c, that if $t \to^+_{\mathrm{base}^{\mathrm{r}}} \mathbb{T}$ and $c(\!|t|\!) \to^+_{\mathrm{r}} \mathbb{T}_2$, then there is $\mathbb{T}' \in \mathbb{N}\langle\lambda\mu^{\mathrm{r}}\rangle$ s.t. $c(\!|\mathbb{T}|\!) \twoheadrightarrow^+_{\mathrm{r}} \mathbb{T}' \;{}^{+}_{\mathrm{r}}\!\twoheadleftarrow \mathbb{T}_2$. The proof crucially uses Lemma 2.25 and Remark 2.23. All the cases of the induction are either easy by induction, or they reduce to the case $c = \xi$, so this is the only one we sketch below.

We have $c(\!|t|\!) = t \to^+_{\mathrm{base}^{\mathrm{r}}} \mathbb{T}$ and we only have the three base-cases of Definition 2.17.

Case $t = (\lambda x.s)[\vec{u}]$ and $\mathbb{T} = s\langle[\vec{u}]/x\rangle^+$. Then $c(\!|t|\!) = t \to^+_{\mathrm{r}} \mathbb{T}_2$ (on a different redex than t) can only be performed either by reducing s, or by reducing an element w of $[\vec{u}]$. We have thus the two easy respective diagrams.

Case $t = (\mu\alpha.{}_\beta|s|)[\vec{u}]$ and $\mathbb{T} = \mu\alpha.\langle_\beta|s|\rangle^+_\alpha[\vec{u}]$. Then $c(\!|t|\!) = t \to^+_{\mathrm{r}} \mathbb{T}_2$ (on a different redex than t) can only be performed either by reducing s, giving rise to an easy diagram, or by reducing an element w of $[\vec{u}]$, giving rise to an easy diagram, or if $s = \mu\gamma.{}_\eta|s'|$ and we reduce the ρ-redex $\ldots_\beta|\mu\gamma.\ldots|$. In the latter case we split into the case $\alpha \neq \beta$, the case $\alpha = \beta, \gamma \neq \eta, \eta = \alpha$, the case $\alpha = \beta, \gamma \neq \eta, \eta \neq \alpha$, and the case $\alpha = \beta, \eta = \gamma$ (with necessarily $\gamma \neq \alpha$). These four cases respectively correspond to four non-trivial (but similar) diagrams, of which we only show the one corresponding to the case $\alpha = \beta, \gamma \neq \eta, \eta = \alpha$: $(\mu\alpha.{}_\alpha|\mu\gamma.{}_\alpha|s'||)[\vec{u}]$ reduces both to $\mathbb{U} := \mu\alpha.\langle_\alpha|\mu\gamma.{}_\alpha|s'||\rangle^+_\alpha[\vec{u}]$ and to $v := (\mu\alpha.{}_\alpha|s'\{\alpha/\gamma\}|)[\vec{u}]$. Now, $v \to^+_{\mathrm{r}} \mu\alpha.\langle_\alpha|s'\{\alpha/\gamma\}|\rangle^+_\alpha[\vec{u}] = \sum\limits_W \mu\alpha.{}_\alpha|(\langle s'\{\alpha/\gamma\}\rangle^+_\alpha[\vec{w}^{\,0}])[\vec{w}^{\,1}]|$ $=: \mathbb{V}$ (with $W : (\vec{u}) \to \{1,2\}$), while it is easy to see that (with $W : (\vec{u}) \to \{1,2\}, D : (\vec{w}^{\,0}) \to \{1,2\}$) $\mathbb{U} \twoheadrightarrow^+_{\mathrm{r}}$-reduces to $\sum\limits_W \sum\limits_D \mu\alpha.{}_\alpha|\mu\gamma.{}_\alpha|\langle(\langle s'\rangle^+_\alpha[\vec{d}^{\,0}])[\vec{d}^{\,1}]\rangle^+_\gamma[\vec{w}^{\,1}]||$ which in turn $\twoheadrightarrow^+_{\mathrm{r}}$-reduces to $\sum\limits_W \sum\limits_D \mu\alpha.{}_\alpha|\langle(\langle s'\rangle^+_\alpha[\vec{d}^{\,0}])[\vec{d}^{\,1}]\rangle^+_\gamma[\vec{w}^{\,1}]|\{\alpha/\gamma\} =: \mathbb{U}'$. We can show that $\mathbb{V} = \mathbb{U}'$, so the diagram is closed.

Case $t = \mu\gamma.{}_\alpha|\mu\beta.{}_\eta|s||$ and $\mathbb{T} = \mu\gamma.{}_\eta|s|\{\alpha/\beta\}$. Then $c(\!|t|\!) = t \to^+_{\mathrm{r}} \mathbb{T}_2$ (on a different redex than t) can be only performed either by reducing s, which gives an easy diagram, or if $s = \mu\gamma'.{}_{\eta'}|s'|$ and we reduce the ρ-redex $\ldots_\eta|\mu\gamma'.\ldots|$. Putting $\delta_0 := \delta^\alpha_\eta(\beta)$, $\delta'_1 := \delta^\alpha_{\eta'}(\beta)$, $\delta_1 := \delta^{\delta_0}_{\delta'_1}(\gamma')$, $\delta'_2 := \delta^\eta_{\eta'}(\gamma')$ and $\delta_2 := \delta^\alpha_{\delta'_2}(\beta)$, the latter case gives the following diagram: $\mu\gamma.{}_\alpha|\mu\beta.{}_\eta|\mu\gamma'.{}_{\eta'}|s'|||$ reduces both to $\mathbb{U} := \mu\gamma.{}_{\delta_0}|\mu\gamma'.{}_{\delta'_1}|s'\{\alpha/\beta\}||$ and to $\mathbb{V} := \mu\gamma.{}_\alpha|\mu\beta.{}_{\delta'_2}|s'\{\eta/\gamma'\}||$, while $\mathbb{U} \twoheadrightarrow^+_{\mathrm{r}}$-reduces to $\mu\gamma.{}_{\delta_1}|s'\{\alpha/\beta\}\{\delta_0/\gamma'\}|$ and $\mathbb{V} \twoheadrightarrow^+_{\mathrm{r}}$-reduces to $\mu\gamma.{}_{\delta_2}|s'\{\eta/\gamma'\}\{\alpha/\beta\}|$. We can show those sums equal, so the diagram is closed. ☐

Corollary 2.27. *The reduction $\to^+_{\mathrm{r}}$ is confluent in $\mathbb{N}\langle\lambda\mu^{\mathrm{r}}\rangle$.*

Proof. By the well-known Newman Lemma, thanks to Remark 2.21 and Proposition 2.26. ☐

2.2.2 From the confluence of $(\mathbb{N}\langle\lambda\mu^{\mathrm{r}}\rangle, \to^+_{\mathrm{r}})$ to the confluence of $(2\langle\lambda\mu^{\mathrm{r}}\rangle, \to_{\mathrm{r}})$.

Definition 2.28. *The reduction $\Rightarrow \subseteq \mathbb{N}\langle\lambda\mu^{\mathrm{r}}\rangle \times \mathbb{N}\langle\lambda\mu^{\mathrm{r}}\rangle$ is defined as the contextual closure of the relation:*

$$\{(m\,t + \mathbb{S}, m\,\mathbb{T} + \mathbb{S}) \mid m \in \mathbb{N},\ t \to^+_{\mathrm{r}} \mathbb{T},\ t \notin \mathrm{supp}(\mathbb{S})\}.$$

We have $\Rightarrow \subseteq \twoheadrightarrow^+_{\mathrm{r}}$. It is also easily seen that if $\mathbb{T} \to_{\mathrm{r}} \mathbb{S}$ in $2\langle\lambda\mu^{\mathrm{r}}\rangle$, then for all $m_t \in \mathbb{N}$ (with $t \in \mathbb{T}$), we have $\sum\limits_{t\in\mathbb{T}} m_t t \Rightarrow \mathbb{S}'$ (in $\mathbb{N}\langle\lambda\mu^{\mathrm{r}}\rangle$), with $\mathrm{supp}(\mathbb{S}') = \mathbb{S}$.

Corollary 2.29. *The reduction $\to_{\mathrm{r}}$ in $2\langle\lambda\mu^{\mathrm{r}}\rangle$ is locally confluent.*

Proof. Let $\mathbb{T}_1 \;{}_{\mathrm{r}}\!\leftarrow t \to_{\mathrm{r}} \mathbb{T}_2$ in $2\langle\lambda\mu^{\mathrm{r}}\rangle$. Since we know that $\to_{\mathrm{r}}$ is strongly normalising (Corollary 2.13), there are (in $2\langle\lambda\mu^{\mathrm{r}}\rangle$) reductions $\mathbb{T}_1 \twoheadrightarrow_{\mathrm{r}} \mathbb{S}_1$ and $\mathbb{T}_2 \twoheadrightarrow_{\mathrm{r}} \mathbb{S}_2$, with $\mathbb{S}_1, \mathbb{S}_2$ r-normal. Therefore we have (in $\mathbb{N}\langle\lambda\mu^{\mathrm{r}}\rangle$) reductions $t \Rightarrow \cdots \Rightarrow \mathbb{S}'_1$ and $t \Rightarrow \cdots \Rightarrow \mathbb{S}'_2$, for some $\mathbb{S}'_1, \mathbb{S}'_2 \in \mathbb{N}\langle\lambda\mu^{\mathrm{r}}\rangle$ s.t. $\mathrm{supp}(\mathbb{S}'_i) = \mathbb{S}_i$. But then, since $\mathbb{S}_i$ is r-normal, $\mathbb{S}'_i$ must be $\to^+_{\mathrm{r}}$-normal. Now because of Corollary 2.27, it must be $\mathbb{S}'_1 = \mathbb{S}'_2$, and therefore $\mathbb{S}_1 = \mathrm{supp}(\mathbb{S}'_1) = \mathrm{supp}(\mathbb{S}'_2) = \mathbb{S}_2$. Hence, we found a common reduct of $\mathbb{T}_1, \mathbb{T}_2$. ☐

Corollary 2.30 (Confluence)**.** *The reduction $\to_{\mathrm{r}}$ is confluent on $2\langle\lambda\mu^{\mathrm{r}}\rangle$.*

Proof. By Newman Lemma, thanks to Corollary 2.13 and Corollary 2.29. ☐

3 Qualitative Taylor expansion

3.1 Crucial properties

The calculus and its resource sensitive version are almost the same; the Taylor expansion map makes us pass from one to the other.

Definition 3.1. *The (qualitative) Taylor expansion is the map $\mathcal{T} : \lambda\mu \to \mathcal{P}(\lambda\mu^{\mathrm{r}})$ defined as:*

$$\begin{array}{ll} \mathcal{T}(x) := \{x\} & \mathcal{T}(\lambda x.M) := \{\lambda x.t \mid t \in \mathcal{T}(M)\} \\ \multicolumn{2}{l}{\mathcal{T}(\mu\alpha.{}_\beta|M|) := \{\mu\alpha.{}_\beta|t| \mid t \in \mathcal{T}(M)\}} \\ \multicolumn{2}{l}{\mathcal{T}(MN) := \{t[\vec{u}] \mid t \in \mathcal{T}(M),\ [\vec{u}] \in !\mathcal{T}(N)\}.} \end{array}$$

Since $\to_{\mathrm{r}}$ is confluent and strongly normalising, all resource terms t have a unique r-normal form $\mathrm{nf}_{\mathrm{r}}(t) \in 2\langle\lambda\mu^{\mathrm{r}}\rangle$ (it can be 0). Therefore, for all $M \in \lambda\mu$ there always exists $\mathrm{NF}\mathcal{T}(M) := \bigcup\limits_{t\in\mathcal{T}(M)} \mathrm{nf}_{\mathrm{r}}(t) \subseteq \lambda\mu^{\mathrm{r}}$ (in general infinite, thus not a sum). This allows to endow $\lambda\mu$ with a preorder:

$$M \leq N \textit{ iff } \mathrm{NF}\mathcal{T}(M) \subseteq \mathrm{NF}\mathcal{T}(N).$$

Theorem 3.2 (Monotonicity)**.** *For C a context, the map $C(\!|\cdot|\!) : \lambda\mu \to \lambda\mu$ is monotone w.r.t. $\leq$.*

Proof. Induction on C, as in [1]. ☐

The following technical lemma says that Taylor expansion behaves well w.r.t. substitutions.

Lemma 3.3. *One has:*

1. $\mathcal{T}(M\{\alpha/\beta\}) = \mathcal{T}(M)\{\alpha/\beta\}$
2. $\mathcal{T}(M\{N/x\}) = \bigcup_{t\in\mathcal{T}(M)} \bigcup_{\vec{u}\in\,!\mathcal{T}(N)} t\langle[\vec{u}]/x\rangle$
3. $\mathcal{T}((M)_\alpha N) = \bigcup_{t\in\mathcal{T}(M)} \bigcup_{\vec{u}\in\,!\mathcal{T}(N)} \langle t\rangle_\alpha[\vec{u}]$.

Proof sketch. (1). Straightforward induction on M.
(2). Induction on M as one does for λ-calculus. The only new case is $M = \mu\beta.{}_\alpha|P|$ but it is done straightforwardly exactly as the case $M = \lambda x.P$.
(3). Induction on M. Not more difficult than (2). □

The following important "simulation property" says in which sense the elements of $\mathcal{T}(M)$ approximate M.

Proposition 3.4. *If $M \to_{\text{base}} N$, then:*

1. *for all $s \in \mathcal{T}(M)$ there exist $\mathbb{T} \subseteq \mathcal{T}(N)$ s.t. $s \twoheadrightarrow_{\text{r}} \mathbb{T}$*
2. *for all $s' \in \mathcal{T}(N)$ there is $s \in \mathcal{T}(M)$ s.t. $s \twoheadrightarrow_{\text{r}} s' + \mathbb{T}$ for some sum $\mathbb{T} \subseteq \mathcal{T}(N)$.*

Furthermore, the same property lifts to all $\to$, that is, if $M \to N$ then point (1) and (2) hold.

Proof sketch. Points 1) and 2) are easy using Lemma 3.3. The "furthermore" part is by induction on the single-hole context C s.t. $M = C(\!|M'|\!)$, $N = C(\!|N'|\!)$ and $M' \to_{\text{base}} N'$. □

The following technical lemma is an adaptation of [11, Theorem 20].

Lemma 3.5. *Let P, Q be $\lambda\mu$-terms, $p, p' \in \mathcal{T}(P)$ and $[\vec{d}], [\vec{d}\,'] \in\, !\mathcal{T}(Q)$. Then $p = p'$ and $[\vec{d}] = [\vec{d}\,']$ are entailed by any of the following three[5] conditions:*

1. *if $p\langle[\vec{d}]/x\rangle \cap p'\langle[\vec{d}\,']/x\rangle \neq \emptyset$*
2. *if $\langle p\rangle_\gamma[\vec{d}] \cap \langle p'\rangle_\gamma[\vec{d}\,'] \neq \emptyset$*
3. *if $\langle_\eta|p|\rangle_\gamma[\vec{d}] \cap \langle_\eta|p'|\rangle_\gamma[\vec{d}\,'] \neq \emptyset$.*

The following "non-interference property" (Theorem 3.6) was first proved by Ehrhard and Regnier in [11, Theorem 22] for the λ-calculus. It is known that it fails in MELL. A natural question, to which we do not have an answer yet, is what is the threshold, between λ-calculus and MELL, where this property starts failing. It is important also because somehow it is linked to the possibility of defining a *coherence* on resource terms for which Taylor expansion is a maximal clique. We show below that the result holds in $\lambda\mu$-calculus.

Theorem 3.6. *If $t, s \in \mathcal{T}(M)$, $t \neq s$, then $\text{nf}_{\text{r}}(t) \cap \text{nf}_{\text{r}}(s) = \emptyset$.*

[5]Remark that point 3. (used in the proof of Theorem 3.6) is *not* an inductive step of point 2., simply because ${}_\eta|p| \notin \lambda\mu^{\text{r}}$. Therefore we treat is separately. This is due to the fact that we are in $\lambda\mu$-calculus and not in Saurin's $\Lambda\mu$-calculus.

Proof. By induction on $\mathbf{ms}(t)$ we prove that for all $s \in \lambda\mu^{\text{r}}$, if $t, s \in \mathcal{T}(M)$ for some $M \in \lambda\mu$, and if there is $h \in \text{nf}_{\text{r}}(t) \cap \text{nf}_{\text{r}}(s)$, then $t = s$.

Case $\mathbf{ms}(t) = (1, 0, 1)$. Then (Corollary 2.16) t is a variable, thus M is the same variable and therefore $s = t$.

Case $\mathbf{ms}(t) > (1, 0, 1)$. By Lemma 1.4, M has shape:

$$M = \lambda\vec{x}_1\mu\alpha_1.{}_{\beta_1}|\ldots\lambda\vec{x}_k\mu\alpha_k.{}_{\beta_k}|RQ_1\ldots Q_n||$$

for R either a variable, or a λ-redex or a μ-redex. Since the series of λ and μ abstractions (with their respective namings) will play no role in the following, in this proof we shorten $\lambda\vec{x}_1\mu\alpha_1.{}_{\beta_1}|\ldots\lambda\vec{x}_k\mu\alpha_k.{}_{\beta_k}|\ldots||$ to just $\vec{\lambda\mu}|\ldots|$. So: $t = \vec{\lambda\mu}|t'[\vec{u}^1]\ldots[\vec{u}^n]|$ and $s = \vec{\lambda\mu}|s'[\vec{v}^1]\ldots[\vec{v}^n]|$ for $t', s' \in \mathcal{T}(R)$ and $[\vec{u}^i], [\vec{v}^i] \in\, !\mathcal{T}(Q_i)$. We have now three subcases depending on the shape of R.

Subcase R variable, say $R = x$. Then $t' = s' = x$. *W.l.o.g.* $n \geq 1$, otherwise it is trivial that $t = s$. Now say $[\vec{u}^i] =: [u^i_1, \ldots, u^i_{m_i}]$ and $[\vec{v}^i] =: [v^i_1, \ldots, v^i_{m'_i}]$ for $i = 1, \ldots, n$. By confluence we have $h \in \text{nf}_{\text{r}}(\vec{\lambda\mu}|x\,\text{nf}_{\text{r}}([\vec{u}^1])\ldots\text{nf}_{\text{r}}([\vec{u}^n])|)$, so $h \in \text{nf}_{\text{r}}(\vec{\lambda\mu}|x[\vec{d}^1]\ldots[\vec{d}^n]|)$ for some $d^i_j \in \text{nf}_{\text{r}}(u^i_j)$. Similarly, we get: $h \in \text{nf}_{\text{r}}(\vec{\lambda\mu}|x[\vec{d}'^1]\ldots[\vec{d}'^n]|)$ for some $d'^i_j \in \text{nf}_{\text{r}}(v^i_j)$. So it must be $m_i = m'_i$ ($i = 1, \ldots, n$) and:

$$h = \vec{\lambda\mu}'|x[d^1_1, \ldots, d^1_{m_1}]\cdots[d^n_1, \ldots, d^n_{m_n}]|$$

for some head $\vec{\lambda\mu}'$, some $d^i_j \in \text{nf}_{\text{r}}(u^i_j) \cap \text{nf}_{\text{r}}(v^i_{\sigma_i(j)})$ and permutations σ_i on m_i elements. But $u^i_j, v^i_j \in \mathcal{T}(Q_i)$ and $\mathbf{ms}(u^i_j) < \mathbf{ms}(t)$ since u^i_j is a strict subterm of t. So we can apply the inductive hypothesis to each u^i_j and obtain $u^i_j = v^i_{\sigma_i(j)}$. Hence, $t = s$.

Subcase $R = (\lambda y.P)N$. It is the same argument as the following subcase, so we skip it.

Subcase $R = (\mu\gamma.{}_\eta|P|)N$. Then $t' = (\mu\gamma.{}_\eta|p|)[\vec{d}]$ and $s' = (\mu\gamma.{}_\eta|p'|)[\vec{d}\,']$ for $p, p' \in \mathcal{T}(P)$ and $[\vec{d}], [\vec{d}\,'] \in\, !\mathcal{T}(N)$. By confluence on $\lambda\mu^{\text{r}}$ we have:

$$\text{nf}_{\text{r}}(t) = \text{nf}_{\text{r}}(\vec{\lambda\mu}|(\mu\gamma.\langle_\eta|p|\rangle_\gamma[\vec{d}])[\vec{u}^1]\ldots[\vec{u}^n]|).$$

So there is $h_1 \in \mu\gamma.\langle_\eta|p|\rangle_\gamma[\vec{d}]$ s.t. $h \in \text{nf}_{\text{r}}(\widetilde{h_1})$ where: $\widetilde{h_1} := \vec{\lambda\mu}|h_1[\vec{u}^1]\ldots[\vec{u}^n]|$. Analogously we find a $h_2 \in \mu\gamma.\langle_\eta|p'|\rangle_\gamma[\vec{d}\,']$ s.t. $h \in \text{nf}_{\text{r}}(\widetilde{h_2})$, where: $\widetilde{h_2} := \vec{\lambda\mu}|h_2[\vec{v}^1]\ldots[\vec{v}^n]|$. By Lemma 3.3 we have $h_1, h_2 \in \mathcal{T}(\mu\gamma.(_\eta|P|)_\gamma N)$ and so $\widetilde{h_1}, \widetilde{h_2}$ belong to $\mathcal{T}(\vec{\lambda\mu}|(\mu\gamma.(_\eta|P|)_\gamma N)Q_1\cdots Q_n|)$. This and the fact that $h \in \text{nf}_{\text{r}}(\widetilde{h_1}) \cap \text{nf}_{\text{r}}(\widetilde{h_2})$ mean that $\widetilde{h_1}$ satisfies both the hypotheses of the inductive hypothesis. Moreover, since $t' \to_{\mu^{\text{r}}} h_1 + \mathbb{T}$ for some sum $\mathbb{T}$, then $\text{m}(h_1) < \text{m}(t')$. And since the number of μ's is constant under μ-reduction, $\deg_\mu(t') = \deg_\mu(h_1)$. Therefore we can apply Lemma 2.15(2) and obtain: $\text{m}(\widetilde{h_1}) = \text{m}(\vec{\lambda\mu}|h_1[\vec{u}^1]\ldots[\vec{u}^n]|) < \text{m}(\vec{\lambda\mu}|t'[\vec{u}^1]\ldots[\vec{u}^n]|) = \text{m}(t)$. So $\mathbf{ms}(\widetilde{h_1}) < \mathbf{ms}(t)$ and we can safely apply the inductive hypothesis obtaining $\widetilde{h_1} = \widetilde{h_2}$. Looking at the definition of

$\widetilde{h}_1, \widetilde{h}_2$, we get $h_1 = h_2$ as well as $[\vec{u}^{\,i}] = [\vec{v}^{\,i}]$ $(i = 1, \ldots, n)$. But now we have:

$$\mu\gamma.\langle_\eta|p|\rangle_\gamma[\vec{d}] \ni h_1 = h_2 \in \mu\gamma.\langle_\eta|p|\rangle_\gamma[\vec{d}\,']$$

so Lemma 3.5 gives $p = p'$ and $[\vec{d}] = [\vec{d}\,']$, i.e. $t' = s'$. If we look at the shape of t, s, this last information together with $[\vec{u}^{\,1}] = [\vec{v}^{\,1}], \ldots, [\vec{u}^{\,n}] = [\vec{v}^{\,n}]$, precisely means $t = s$. □

We conclude with a useful property (Corollary 3.8). It follows from the following proposition, which in turn easily follows by Lemma 3.3.

Proposition 3.7. *If $\mathcal{T}(M) \ni t \to_{\text{base}} \mathbb{T}'$ then there is $N \in \lambda\mu$ s.t. $\mathbb{T}' \subseteq \mathcal{T}(N)$ and $M \to N$.*

Corollary 3.8. *For all $\mathbb{T} \subseteq \mathcal{T}(M)$, there exist $N \in \lambda\mu$ s.t. $M \twoheadrightarrow N$ and $\mathrm{nf}_{\mathrm{r}}(\mathbb{T}) \subseteq \mathcal{T}(N)$.*

Proof sketch. One first generalises Proposition 3.7 to sums (instead of a term t in the statement); then, we prove the desired result by induction on the length of a maximal reduction $\mathbb{T} \twoheadrightarrow_{\mathrm{r}} \mathrm{nf}_{\mathrm{r}}(\mathbb{T})$. □

3.2 The $\lambda\mu$-theory $=_\tau$

Mimicking the definitions for λ-calculus we say that:

Definition 3.9.
1. *An equivalence $\mathcal{R}$ on $\lambda\mu$ is a* congruence *iff $\mathcal{R}$ is contextual.*
2. *A congruence $\mathcal{R}$ is a $\lambda\mu$-*theory *iff $\mathcal{R} \supseteq =_{\lambda\mu\rho}$.*
3. *The* term algebra *of a $\lambda\mu$-theory $\mathcal{R}$ is the quotient $\lambda\mu/_{\mathcal{R}}$. A $\lambda\mu$-theory $\mathcal{R}$ is* non-trivial *iff $\lambda\mu/_{\mathcal{R}} \neq \{*\}$.*

It is clear that $=_{\lambda\mu\rho}$ is a $\lambda\mu$-theory. Now fix the equivalence $M =_\tau N$ iff $\mathrm{NF}\mathcal{T}(M) = \mathrm{NF}\mathcal{T}(N)$. Actually, $=_\tau$ is a non-trivial $\lambda\mu$-theory. In fact, the contextuality follows immediately from the Theorem 3.2; the fact that it contains $=_{\lambda\mu\rho}$ easily follows from confluence and Proposition 3.4; and it is clearly non-trivial: $\lambda x.x \neq_\tau \emptyset =_\tau (\lambda x.xx)(\lambda x.xx) =: \Omega$.

In λ-calculus, $=_\tau$ is the λ-theory equating Böhm trees. In particular, it is sensible (i.e. it equates all unsolvables). We will see (Corollary 3.15) that in our case it is still sensible.

Definition 3.10. *A $\lambda\mu$-term M is a* head normal form *(*hnf *for short) iff there are no ρ-redexes in its head (remember Lemma 1.4) and it has a head variable. We define the exact same notion for $\lambda\mu^{\mathrm{r}}$.*

Definition 3.11. *The* head reduction *is the partial function $\mathrm{H} : \lambda\mu \to \lambda\mu$ obtained defining $\mathrm{H}(M)$ via the following algorithm:*
1. *ρ-reduce the leftmost ρ-redex in the head of M, if any*
2. *otherwise, $\lambda\mu$-reduce the head redex of M, if any*
3. *otherwise, $\mathrm{H}(M)$ is not defined.*

$\mathrm{H}(M)$ is not *defined iff M is a hnf. We say that* head reduction starting on M terminates *iff there is a (necessarily unique) $n \geq 0$ s.t. $\mathrm{H}^n(M)$ is a hnf. Here we mean as usual that $\mathrm{H}^0(M) := M$.*

We extend the same definitions to resource terms, and set $\mathrm{H}(t) := \emptyset$ whenever t is a hnf. Moreover, we set $\mathrm{H}^0(t) := t \in 2\langle\Lambda^{\mathrm{r}}\rangle$ and, for $n \geq 0$:

$$\mathrm{H}^{n+1}(t) := \sum_{t_1 \in \mathrm{H}(t)} \sum_{t_2 \in \mathrm{H}(t_1)} \cdots \sum_{t_{n+1} \in \mathrm{H}(t_n)} t_{n+1} \quad \in 2\langle\Lambda^{\mathrm{r}}\rangle.$$

We have $\mathrm{H}^1(t) = \mathrm{H}(t)$ and $\mathrm{H}^{n+1}(t) = \sum_{t' \in \mathrm{H}(t)} \mathrm{H}^n(t')$.

Lemma 3.12. *If s only contains empty bags (if any) and $s \in \mathrm{nf}_{\mathrm{r}}(t)$, then $s \in H^n(t)$ for some $n \geq 0$.*

Proof sketch. If t is a hnf, $s \in \mathrm{nf}_{\mathrm{r}}(t)$ entails that t already contains only empty bags, as any eventual bag of s is empty and reductions cannot erase non-empty bags; but in a hnf the reduction can only take place *inside* some bag, so it must be $s = t$ and we are done. If t is *not* hnf, by confluence $t \to_{\mathrm{r}} \mathrm{H}(t) \twoheadrightarrow_{\mathrm{r}} \mathrm{nf}_{\mathrm{r}}(t) \ni s$. So there is a $t_1 \in \mathrm{H}(t)$ s.t. $s \in \mathrm{nf}_{\mathrm{r}}(t_1)$. Now we reason as in the beginning: if t_1 is hnf we are done; if t_1 is not, we repeat the argument finding some t_2. By the well-foundedness of $\widetilde{\mathrm{m}}(.)$, we cannot repeat the argument forever and we must end on a hnf, so we conclude. □

Set $\mathrm{H}(\mathcal{T}(M)) := \bigcup_{t \in \mathcal{T}(M)} \mathrm{H}(t) \subseteq \lambda\mu^{\mathrm{r}}$. The following lemma is easy using Definition 3.10 and Lemma 3.3.

Lemma 3.13. *If $M \in \lambda\mu$ with $\mathrm{H}(M)$ defined, we have:*

$$\mathcal{T}(\mathrm{H}(M)) = \mathrm{H}(\mathcal{T}(M)).$$

The following proposition shows that $\lambda\mu$-calculus enjoys a notion of *solvability* analogue to the one of λ-calculus.

Proposition 3.14. *For $M \in \lambda\mu$, the following are equivalent:*
1. *$M =_{\lambda\mu\rho} H$ with H hnf*
2. *Head reduction starting on M terminates*
3. *$\mathrm{NF}\mathcal{T}(M) \neq \emptyset$.*

We call $M \in \lambda\mu$ solvable *iff it satisfies any of the previous equivalent conditions. Otherwise, M is called* unsolvable.

Proof. (1⇒2). By confluence M and H have a common $\lambda\mu\rho$-redex M_0. Since H is a hnf, M_0 is too. Let s_0 be the unique resource $\lambda\mu$-term in $\mathcal{T}(M_0)$ s.t. all its bags (if any) are empty. This term clearly exists. Note that, by construction, s_0 is r-normal. By repeatedly applying Proposition 3.4 one can check that we obtain an $s \in \mathcal{T}(M)$ s.t. $s_0 \in \mathrm{nf}_{\mathrm{r}}(s)$. Now, by Lemma 3.12, $s_0 \in \mathrm{H}^n(s)$ for some $n \geq 0$. By repeatedly applying Lemma 3.13, we find that $s_0 \in \mathrm{H}^n(\mathcal{T}(M)) = \mathcal{T}(\mathrm{H}^n(M))$. Finally, since s_0 is a hnf, so it must be $\mathrm{H}^n(M)$.

(2⇒3). Easy.

(3⇒1). If $\mathrm{NF}\mathcal{T}(M) \neq 0$ there is $t \in \mathcal{T}(M)$ s.t. $\mathrm{nf}_{\mathrm{r}}(t) \neq 0$. By Corollary 3.8, $M \twoheadrightarrow N$ for some $N \in \lambda\mu$ s.t. $\mathrm{nf}_{\mathrm{r}}(t) \subseteq \mathcal{T}(N)$. So $\mathcal{T}(N)$ contains at least a hnf, and thus N must be a hnf too. □

Corollary 3.15. *The $\lambda\mu$-theory $=_\tau$ is* sensible *(that is, it equates all unsolvable terms).*

4 Applying the approximation theory

4.1 Stability

The Stability Property gives sufficient conditions for a context to commute with intersections in $\lambda\mu/_{=_\tau}$, i.e. (the intersections are defined below):

$$C(\!|\bigcap_{i_1} N_{i_1},\ldots,\bigcap_{i_n} N_{i_n}|\!) =_\tau \bigcap_{i_1\ldots,i_n} C(\!|N_{i_1},\ldots,N_{i_n}|\!).$$

Given a non-empty subset $\mathcal{X} \subseteq \lambda\mu$, call its *$\mathcal{T}$-infimum* the set $\bigcap \mathcal{X} := \bigcap_{M\in\mathcal{X}} \mathrm{NF}\mathcal{T}(M) \subseteq \lambda\mu^{\mathrm{r}}$. We say that $\mathcal{X}$ is *bounded* iff there exists an $L \in \lambda\mu$ such that $M \le L$ for all $M \in \mathcal{X}$. Write $M =_\tau \bigcap \mathcal{X}$ instead of $\mathrm{NF}\mathcal{T}(M) = \bigcap \mathcal{X}$. Observe that (in case it exists) an M s.t. $M =_\tau \bigcap \mathcal{X}$ need *not* to be unique, so $\bigcap \mathcal{X}$ does not identify a unique λ-term.

Theorem 4.1 (Stability). *Let C be an n-ary $\lambda\mu$-context and fix non empty bounded $\mathcal{X}_1,\ldots,\mathcal{X}_n \subseteq \lambda\mu^{\mathrm{r}}$. For all $M_1,\ldots,M_n \in \lambda\mu$ s.t. $M_i =_\tau \bigcap \mathcal{X}_i$ ($i = 1,\ldots,n$) we have:*

$$C(\!|M_1,\ldots,M_n|\!) =_\tau \bigcap_{\substack{N_1\in\mathcal{X}_1\\ \cdots\\ N_n\in\mathcal{X}_n}} C(\!|N_1,\ldots,N_n|\!).$$

Proof. Non-trivial, but exactly as done in [1] for λ-calculus (using Theorem 3.6). □

Using the usual encoding of booleans and pairs ($\texttt{True} := \lambda xy.x$, $\texttt{False} := \lambda xy.y$, $\langle M,N\rangle := \lambda z.zMN$) we have the non-implementability of the following parallel-or.

Corollary 4.2. *There is* no $\texttt{Por} \in \lambda\mu$ *s.t. for all $M,N \in \lambda\mu$,*

$$\begin{cases} \texttt{Por}\langle M,N\rangle & =_\tau & \texttt{True} & \text{if } M \neq_\tau \Omega \text{ or } N \neq_\tau \Omega \\ \texttt{Por}\langle M,N\rangle & =_\tau & \Omega & \text{if } M =_\tau N =_\tau \Omega. \end{cases}$$

Proof. Otherwise, for the context $C := \texttt{Por}\,\xi$, by Theorem 4.1 we would have the contradiction: $\texttt{True} =_\tau C(\!|\langle\texttt{True},\Omega\rangle|\!) \cap C(\!|\langle\Omega,\texttt{True}\rangle|\!) =_\tau C(\!|\langle\texttt{True},\Omega\rangle \cap \langle\Omega,\texttt{True}\rangle|\!) =_\tau C(\!|\langle\Omega,\Omega\rangle|\!) =_\tau \Omega$. □

4.2 The perpendicular Lines Property

The perpendicular lines Property (PLP for short) states that, fixed a term $\lambda z_1\ldots z_n.F \in \lambda\mu$, if the function $\vec{M} \in \lambda\mu^n/_{=_\tau} \to (\lambda\vec{z}.F)\vec{M} \in \lambda\mu/_{=_\tau}$ is constant on n "perpendicular lines" (in the sense of the statement, Theorem 4.4), then it is constant everywhere. Lemma 4.3 below is the crucial ingredient for the proof of PLP, and we use in it all the strong properties of resource approximation (linearity, SN and confluence).

Lemma 4.3. *Fix $\vec{z} := z_1,\ldots z_n$ distinct variables and let $t \in \lambda\mu^{\mathrm{r}}$. Suppose that:*

i. $\mathrm{nf_r}(t) \neq 0$
ii. *there is $F \in \lambda\mu$ s.t. $t \in \mathcal{T}(F)$*
iii. *there are $\{M_{ij}\}_{1\le i\neq j\le n} \subseteq \lambda\mu$ s.t. the function mapping $\vec{M} \in \lambda\mu^n/_{=_\tau}$ to $(\lambda\vec{z}.F)\vec{M} \in \lambda\mu/_{=_\tau}$ is constant on the following "perpendicular lines" of $\lambda\mu^n/_{=_\tau}$:*

$$\begin{array}{l} l_1 = \{(Z,\ M_{12},\ \ldots\ldots,\ M_{1n}) \mid Z \in \lambda\mu\} \\ l_2 = \{(M_{21},\ Z,\ \ldots\ldots,\ M_{2n}) \mid Z \in \lambda\mu\} \\ \qquad\qquad\ddots \\ l_n = \{(M_{n1},\ \ldots,\ M_{n(n-1)},\ Z) \mid Z \in \lambda\mu\}. \end{array} \tag{1}$$

Then $\deg_{z_1}(t) = \cdots = \deg_{z_n}(t) = 0$.

Proof. Induction on the size $\mathbf{ms}(t)$ of $t \in \lambda\mu^{\mathrm{r}}$.

- Case $\mathbf{ms}(t) = (1,0,1)$. Then t is a variable (Corollary 2.16). If $t = z_i$ for some i then the i-th line of (1) gives the contradiction:

$$N_i =_\tau (\lambda\vec{z}.z_i)M_{i1}\cdots M_{i(i-1)}ZM_{i(i+1)}\cdots M_{in} =_\tau Z$$

for all $Z \in \lambda\mu$. Hence, it must be $\deg_{z_1}(t) = \cdots = \deg_{z_n}(t) = 0$.

- Case $\mathbf{ms}(t) > (1,0,1)$. By (i) there is $u \in \mathrm{nf_r}(t)$. As u is normal, it has shape: $u = \vec{\lambda\mu}|y[\vec{u}^{\,1}]\ldots[\vec{u}^{\,m}]|$ for some $m \ge 0$, some variable y, some normal bags $[\vec{u}^{\,j}]$, and where we have shorten, as before, a series $\lambda\vec{x}_1\mu\alpha_1.\beta_1|\ldots\lambda\vec{x}_k\mu\alpha_k.\beta_k|\ldots||$ of λ and μ abstraction by just $\vec{\lambda\mu}|\ldots|$. By (ii) $t \in \mathcal{T}(F)$, so that by Corollary 3.8 there is $Q \in \lambda\mu$ s.t. $F \twoheadrightarrow Q$ and $u \in \mathcal{T}(Q)$. So Q must have shape: $Q = \vec{\lambda\mu}|yQ_1\cdots Q_m|$ for some Q_j's in $\lambda\mu$ s.t. $[\vec{u}_j] \in\, !\,\mathcal{T}(Q_j)$ for all $j = 1,\ldots,m$. Now there are two possibilities: either $y = z_i$ for some $i = 1,\ldots,n$, either $y \neq z_i$ for all i.

Suppose $y = z_i$. Then, for $\vec{q} := q_1,\ldots,q_m$ fresh variables, we can chose $Z := \lambda\vec{q}.\texttt{True} \in \lambda\mu$ (or $Z := \texttt{True}$ if $m = 0$) in the i-th line l_i of (1), and since by (iii) $\lambda\vec{z}.F$ is constant (mod $=_\tau$) on l_i, we can compute its value as:

$$\begin{array}{ll} & (\lambda\vec{z}.F)M_{i1}\cdots M_{i(i-1)}\,(\lambda\vec{q}.\texttt{True})\,M_{i(i+1)}\cdots M_{in} \\ =_\tau & Q\{M_{i1}/z_1,\ldots,(\lambda\vec{q}.\texttt{True})/z_i,\ldots,M_{in}/z_n\} \\ =_\tau & \vec{\lambda\mu}|(\lambda\vec{q}.\texttt{True})\widetilde{Q_{i1}}\cdots\widetilde{Q_{im}}| \\ =_\tau & \vec{\lambda\mu}|\texttt{True}| \end{array}$$

where we set $\widetilde{Q_{ij}} := Q_j\{M_{i1}/z_1,\ldots,(\lambda\vec{q}.\texttt{True})/z_i,\ldots,M_{in}/z_n\}$. The first equality holds because $F \twoheadrightarrow Q$ and $=_\tau$ is finer than $=_{\lambda\mu\rho}$, and the second equality holds because $y = z_i$. In the same way, choosing $Z := \lambda\vec{q}.\texttt{False} \in \lambda\mu$ in l_i we find that the value (mod $=_\tau$) of $\lambda\vec{z}.F$ on l_i is $\vec{\lambda\mu}|\texttt{False}|$. But this is impossible because $\texttt{True} \neq_\tau \texttt{False}$.

Therefore, it must be $y \neq z_i$ for all i. Note that *w.l.o.g.* $m \ge 1$ (indeed if $m = 0$, from the fact that $y \neq z_i$ for all i we already get $\deg_{z_i}(u) = 0$ and, as $u \in \mathrm{nf_r}(t)$ and in $\lambda\mu^{\mathrm{r}}$ one cannot erase non-empty bags, we are done). Now fix $i \in \{1,\ldots,n\}$ and $Z', Z'' \in \lambda\mu$. Similarly as before, choosing $Z := Z'$ in l_i and using what we found so far, putting $Q'_{ij} := Q_j\{M_{i1}/z_1,\ldots,Z'/z_i,\ldots,M_{in}/z_n\}$, since $\lambda\vec{z}.F$ is constant (mod $=_\tau$) on l_i, we can compute its value as:

$$\begin{array}{ll} & (\lambda\vec{z}.F)M_{i1}\ldots M_{i(i-1)}Z'M_{i(i+1)}\ldots M_{in} \\ =_\tau & Q\{M_{i1}/z_1,\ldots,Z'/z_i,\ldots,M_{in}/z_n\} \\ =_\tau & \vec{\lambda\mu}|yQ'_{i1}\ldots Q'_{im}| \end{array}$$

where the last equality holds since y is *not* one of the z_i's. Choosing Z'' instead of Z' and putting Q''_{ij} the same as Q'_{ij}

but with Z'' instead of Z', one has that the value (mod $=_\tau$) of $\lambda\vec{z}.F$ on l_i is $\vec{\lambda\mu}|yQ''_{i1}\dots Q''_{im}|$. So we have $\vec{\lambda\mu}|yQ'_{i1}\dots Q'_{im}| = \vec{\lambda\mu}|yQ''_{i1}\dots Q''_{im}|$, which easily entails: $Q'_{i1} =_\tau Q''_{i1}, \dots, Q'_{im} =_\tau Q''_{im}$. But by construction we have:

$$Q'_{ij} =_\tau (\lambda\vec{z}.Q_j)M_{i1}\dots M_{i(i-1)}Z'M_{i(i+1)}\dots M_{in}$$
$$Q''_{ij} =_\tau (\lambda\vec{z}.Q_j)M_{i1}\dots M_{i(i-1)}Z''M_{i(i+1)}\dots M_{in}.$$

So if we remember that Z', Z'' were generic in $\lambda\mu$, the previous equalities $Q'_{ij} = Q''_{ij}$ precisely say that $\lambda\vec{z}.Q_j$ is constant on the line l_i. And since this holds for all $i = 1, \dots, n$, we have just found that $\lambda\vec{z}.Q_j$ satisfies (iii). And since we have equalities $Q'_{ij} = Q''_{ij}$ for all $j = 1, \dots, m$, we have that each $\lambda\vec{z}.Q_1, \dots, \lambda\vec{z}.Q_k$ satisfies (iii). We can now comfortably apply the induction hypothesis on any $s \in [\vec{u}^j]$. In fact, as $[\vec{u}^j]$ is normal, $\mathrm{nf_r}(s) \neq 0$, i.e. (i); as $[\vec{u}^j] \in !\mathcal{T}(Q_j)$, we have $s \in \mathcal{T}(Q_j)$, i.e. (ii); and we just found that $\lambda\vec{z}.Q_j$ satisfies (iii); finally, s is a strict subterm of $u \in \mathrm{nf_r}(t)$, thus (Corollary 2.16) $\mathbf{ms}(s) < \mathbf{ms}(u) \leq \mathbf{ms}(t)$. Therefore, the inductive hypothesis gives $\deg_{z_1}(s) = \dots = \deg_{z_n}(s) = 0$. Since this is true for all s in all $[\vec{u}^j]$, $j = 1, \dots, m$, we get $\deg_{z_1}(u) = \dots = \deg_{z_n}(u) = 0$. And now $u \in \mathrm{nf_r}(t)$ entails $\deg_{z_1}(t) = \dots = \deg_{z_n}(t) = 0$. ◻

Theorem 4.4 (Perpendicular Lines Property). *Suppose that for some fixed $\{M_{ij}\}_{1\leq i\neq j\leq n}$, $\{N_i\}_{1\leq i\leq n} \subseteq \lambda\mu$, the system of equations:*

$$\begin{cases} (\lambda z_1\dots z_n.F)\ Z\, M_{12}\ \dots\dots\ M_{1n} & =_\tau\ N_1 \\ (\lambda z_1\dots z_n.F)\ M_{21}\, Z\ \dots\dots\ M_{2n} & =_\tau\ N_2 \\ \qquad\qquad\qquad\qquad\ddots & \ \ \vdots \\ (\lambda z_1\dots z_n.F)\ M_{n1}\ \dots\ M_{n(n-1)}\, Z & =_\tau\ N_n \end{cases}$$

holds for all $Z \in \lambda\mu$. Then:

$$(\lambda z_1\dots z_n.F)Z_1\dots Z_n =_\tau N_1$$

for all $Z_1, \dots, Z_n \in \lambda\mu$.

Proof. It follows from Lemma 4.3 as done in [1]. ◻

Corollary 4.5. *There is* no $\texttt{Por}' \in \lambda\mu$ *s.t. for all* $Z \in \lambda\mu$ *one has at the same time* $\texttt{Por}'\,\texttt{True}\,Z =_\tau \texttt{True}$, $\texttt{Por}'\,Z\,\texttt{True} =_\tau \texttt{True}$, $\texttt{Por}'\,\texttt{False}\,\texttt{False} =_\tau \texttt{False}$.

The non existence of parallelism in $\lambda\mu$-calculus is known as folklore via arguments involving stable models: here we proved it solely via Taylor expansion.

5 Conclusions and Future Works

In [23] Laurent studies the mathematics of (untyped) $\lambda\mu$-calculus via its denotational semantics; this paper does it by developing a theory of program approximation based on Linear Logic resources. In particular, we proved that the approximation theory satisfies strong normalisation and confluence (non-trivial results in this setting), the Monotonicity Property, the Non-Interference Property, that it induces a sensible $\lambda\mu$-theory, and that it can be used as a tool in order to obtain the Stability Property and the Perpendicular Lines Property, and thus the impossibility of parallel computations in the language. A first natural question immediately arises:

1- Does Taylor expansion allow to find interesting properties *not* satisfied by λ-calculus, but that are enjoyed by the $\lambda\mu$-calculus due to the presence of **`callcc`**?

For future investigations, we believe that it would be interesting to integrate this approach with the *differential* extension of $\lambda\mu$-calculus defined in [29], via the mentioned translation $(\cdot)^\partial$, in order to explore quantitative properties.

The following two questions are maybe the most significant ones:

2- Does it makes sense to introduce Böhm trees for the $\lambda\mu$-calculus? For instance, for the call-by-value λ-calculus, the Taylor expansion has provided in [17] invaluable guidance for finding a meaningful notion of trees satisfying Ehrhard and Regnier's *commutation formula*; the same methodology could maybe be applied here. However, in [7] it is shown that $\lambda\mu$-calculus does *not* enjoy Böhm's separation property. David and Py's counterexample could hence be an indication that, instead, Böhm trees are not a "good" notion for $\lambda\mu$-calculus. The best way of proceeding would be, in that case, to consider the natural extension of $\lambda\mu$-calculus given by Saurin's $\Lambda\mu$-calculus [27]. It was introduced precisely to satisfy Böhm's property and, as a matter of fact, in [27] Saurin proposes a definition of Böhm trees for his $\Lambda\mu$-calculus.

3- Does $\Lambda\mu$-calculus enjoys the same approximation theory developed in this paper? On one hand, many constructions we did in this chapter seem possible also in Saurin's calculus, on the other hand we used the fact that the number of μ's in a term is the same as the named subterms, e.g. in Remark 2.7. In general, one could wonder which one, between $\lambda\mu$ and $\Lambda\mu$, should be the "canonical lambda-mu-calculus": from a proof-theoretical perspective $\lambda\mu$-calculus precisely corresponds to Parigot's CD-derivations, but $\Lambda\mu$-calculus satisfies more desirable properties (Böhm separation). Moreover, in [26], Saurin adapts usual techniques of λ-calculus to $\Lambda\mu$-calculus: he studies the notion of solvability, proves a standardization theorem and studies more in detail the notion of Böhm trees. A very interesting future direction of research would be, hence, to develop our theory of resource approximation for Saurin's calculus, and study its relation with his theory of Böhm approximation. In any case, we look at the fact that the Taylor expansion works so nicely in $\lambda\mu$-calculus – and this regardless of a notion of Böhm trees – as a *a posteriori* confirmation of the high power of this form of approximation.

There are at least three other interesting points in relation with strictly related areas:

3- The $\lambda\mu$-calculus can be translated in the λ-calculus via the CPS-translations (see e.g. de Groote's one in [8]). What does our theory of approximation correspond to under this translation?

4- In order to perform a deeper logical analysis, one should consider translations into Linear Logic. It is known from [22] that $\lambda\mu$-calculus translates into polarized proof nets. Taylor expansion for proof-nets is possible, but the construction can be complex: in fact one of the interests in *directly* defining a Taylor expansion for a certain "λ-calculus style" programming language (as we did for $\lambda\mu$-calculus, and as one does for λ-calculus) is precisely to avoid that complexity. In our case we have just shown that, at the end of the day, the theory of resource approximation for $\lambda\mu$-calculus can be developed with essentially the same methodology as in λ-calculus. This leads to asking what makes a Taylor expansion "easy", and should be considered in relation to the already mentioned possibility of the existence of a coherence relation for which $\mathcal{T}(M)$ is a clique. This motivates an investigation of the complexity of the definition of a Taylor expansion of a programming language/proof system, which may be related to the notion of connectedness of proof-nets, whose study starts in [14]. Such question should be considered in relation with the so-called problem of the "inversion of Taylor expansion" [15, 16] and the problem of "injectivity" of denotational models (in particular, the relational one) for Linear Logic.

5- The $\lambda\mu$-calculus is not the only way of extending the Curry-Howard correspondence to classical logic. Another notable one is the already mentioned *Krivine's classical realizability*, which is a "machine to extract computational content from proofs + axioms" (for almost all mathematics, such as the one formalizable in ZF+AC, see [20]). There are translations between $\lambda\mu$-calculus and Krivine's calculus, and *vice-versa*. What does our work say about Krivine's realisability?

Acknowledgments

Great help was given by Giulio Manzonetto and Lorenzo Tortora de Falco; crucial remarks were given by the anonymous reviewers; we had useful discussions with Olivier Laurent, Lionel Vaux. Thanks to all of them, as well as to Federico Olimpieri, who first questioned us about Taylor expansion for $\lambda\mu$-calculus. This work has been partially supported by the ERC CoG 818616 "DIAPASoN".

References

[1] Davide Barbarossa and Giulio Manzonetto. 2020. Taylor subsumes Scott, Berry, Kahn and Plotkin. *Proc. ACM Program. Lang.* 4, POPL (2020), 1:1–1:23. https://doi.org/10.1145/3371069

[2] Hendrik Pieter Barendregt. 1984. *The lambda calculus - its syntax and semantics*. Studies in logic and the foundations of mathematics, Vol. 103. North-Holland.

[3] Edward A Bender. 1974. Partitions of multisets. *Discrete Mathematics* 9, 4 (1974), 301–311.

[4] Gérard Boudol. 1993. The Lambda-Calculus with Multiplicities (Abstract). In *CONCUR '93, 4th International Conference on Concurrency Theory, Hildesheim, Germany, August 23-26, 1993, Proceedings (Lecture Notes in Computer Science, Vol. 715)*, Eike Best (Ed.). Springer, 1–6. https://doi.org/10.1007/3-540-57208-2_1

[5] Jules Chouquet. 2019. Taylor Expansion, Finiteness and Strategies. In *Proceedings of the Thirty-Fifth Conference on the Mathematical Foundations of Programming Semantics, MFPS 2019, London, UK, June 4-7, 2019 (Electronic Notes in Theoretical Computer Science, Vol. 347)*, Barbara König (Ed.). Elsevier, 65–85. https://doi.org/10.1016/j.entcs.2019.09.005

[6] Jules Chouquet and Christine Tasson. 2020. Taylor expansion for Call-By-Push-Value. In *28th EACSL Annual Conference on Computer Science Logic, CSL 2020, January 13-16, 2020, Barcelona, Spain (LIPIcs, Vol. 152)*, Maribel Fernández and Anca Muscholl (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 16:1–16:16. https://doi.org/10.4230/LIPIcs.CSL.2020.16

[7] René David and Walter Py. 2001. Lambda-mu-Calculus and Böhm's Theorem. *J. Symb. Log.* 66, 1 (2001), 407–413. https://doi.org/10.2307/2694930

[8] Philippe de Groote. 1994. A CPS-Translation of the Lambda-μ-Calculus. In *Trees in Algebra and Programming - CAAP'94, 19th International Colloquium, Edinburgh, UK, April 11-13, 1994, Proceedings (Lecture Notes in Computer Science, Vol. 787)*, Sophie Tison (Ed.). Springer, 85–99. https://doi.org/10.1007/BFb0017475

[9] Thomas Ehrhard and Laurent Regnier. 2003. The differential lambda-calculus. *Theor. Comput. Sci.* 309, 1-3 (2003), 1–41. https://doi.org/10.1016/S0304-3975(03)00392-X

[10] Thomas Ehrhard and Laurent Regnier. 2006. Differential interaction nets. *Theor. Comput. Sci.* 364, 2 (2006), 166–195. https://doi.org/10.1016/j.tcs.2006.08.003

[11] Thomas Ehrhard and Laurent Regnier. 2008. Uniformity and the Taylor expansion of ordinary lambda-terms. *Theor. Comput. Sci.* 403, 2-3 (2008), 347–372. https://doi.org/10.1016/j.tcs.2008.06.001

[12] Jean-Yves Girard. 1987. Linear logic. *Theoretical Computer Science* 50, 1 (1987), 1 – 101. https://doi.org/10.1016/0304-3975(87)90045-4

[13] Timothy Griffin. 1990. A Formulae-as-Types Notion of Control. In *Conference Record of the Seventeenth Annual ACM Symposium on Principles of Programming Languages, San Francisco, California, USA, January 1990*, Frances E. Allen (Ed.). ACM Press, 47–58. https://doi.org/10.1145/96709.96714

[14] Giulio Guerrieri, Luc Pellissier, and Lorenzo Tortora de Falco. 2016. Computing Connected Proof(-Structure)s From Their Taylor Expansion. In *1st International Conference on Formal Structures for Computation and Deduction, FSCD 2016, June 22-26, 2016, Porto, Portugal (LIPIcs, Vol. 52)*, Delia Kesner and Brigitte Pientka (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 20:1–20:18. https://doi.org/10.4230/LIPIcs.FSCD.2016.20

[15] Giulio Guerrieri, Luc Pellissier, and Lorenzo Tortora de Falco. 2019. Proof-Net as Graph, Taylor Expansion as Pullback. In *Logic, Language, Information, and Computation - 26th International Workshop, WoLLIC 2019, Utrecht, The Netherlands, July 2-5, 2019, Proceedings (Lecture Notes in Computer Science, Vol. 11541)*, Rosalie Iemhoff, Michael Moortgat, and Ruy J. G. B. de Queiroz (Eds.). Springer, 282–300. https://doi.org/10.1007/978-3-662-59533-6_18

[16] Giulio Guerrieri, Luc Pellissier, and Lorenzo Tortora de Falco. 2020. Glueability of Resource Proof-Structures: Inverting the Taylor Expansion. In *28th EACSL Annual Conference on Computer Science Logic, CSL 2020, January 13-16, 2020, Barcelona, Spain (LIPIcs, Vol. 152)*, Maribel Fernández and Anca Muscholl (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 24:1–24:18. https://doi.org/10.4230/LIPIcs.CSL.2020.24

[17] Emma Kerinec, Giulio Manzonetto, and Michele Pagani. 2020. Revisiting Call-by-value Böhm trees in light of their Taylor expansion. *Log. Methods Comput. Sci.* 16, 3 (2020). https://lmcs.episciences.org/6638

[18] Delia Kesner and Pierre Vial. 2017. Types as Resources for Classical Natural Deduction. In *2nd International Conference on Formal Structures for Computation and Deduction, FSCD 2017, September 3-9, 2017, Oxford, UK (LIPIcs, Vol. 84)*, Dale Miller (Ed.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 24:1–24:17. https://doi.org/10.4230/LIPIcs.FSCD.2017.24

[19] Jean-Louis Krivine. 2009. Realizability in classical logic. *Panoramas et synthèses* 27 (2009), 197–229. https://hal.archives-ouvertes.fr/hal-00154500

[20] Jean-Louis Krivine. 2020. A program for the full axiom of choice. arXiv:2006.05433 [cs.LO] https://arxiv.org/abs/2006.05433

[21] Ugo Dal Lago and Thomas Leventis. 2019. On the Taylor Expansion of Probabilistic lambda-terms. In *4th International Conference on Formal Structures for Computation and Deduction, FSCD 2019, June 24-30, 2019, Dortmund, Germany (LIPIcs, Vol. 131)*, Herman Geuvers (Ed.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 13:1–13:16. https://doi.org/10.4230/LIPIcs.FSCD.2019.13

[22] Olivier Laurent. 2003. Polarized proof-nets and $\lambda\mu$-calculus. *Theor. Comput. Sci.* 290, 1 (2003), 161–188. https://doi.org/10.1016/S0304-3975(01)00297-3

[23] Olivier Laurent. 2004. On the denotational semantics of the untyped lambda-mu calculus. (Jan. 2004). http://perso.ens-lyon.fr/olivier.laurent/lmmodels.pdf Unpublished note.

[24] Michel Parigot. 1992. Lambda-Mu-Calculus: An Algorithmic Interpretation of Classical Natural Deduction. In *Logic Programming and Automated Reasoning,International Conference LPAR'92, St. Petersburg, Russia, July 15-20, 1992, Proceedings (Lecture Notes in Computer Science, Vol. 624)*, Andrei Voronkov (Ed.). Springer, 190–201. https://doi.org/10.1007/BFb0013061

[25] Walter Py. 1998. Confluence en lambda-mu-calcul. *Ph.D.Thesis, Université de Savoie* (1998). http://www.lama.univ-savoie.fr/pagesmembres/david/ftp/py.pdf

[26] Alexis Saurin. 2010. Standardization and Böhm Trees for $\Lambda\mu$-Calculus. In *Functional and Logic Programming, 10th International Symposium, FLOPS 2010, Sendai, Japan, April 19-21, 2010. Proceedings (Lecture Notes in Computer Science, Vol. 6009)*, Matthias Blume, Naoki Kobayashi, and Germán Vidal (Eds.). Springer, 134–149. https://doi.org/10.1007/978-3-642-12251-4_11

[27] Alexis Saurin. 2012. Böhm theorem and Böhm trees for the $\Lambda\mu$-calculus. *Theoretical Computer Science* 435 (2012), 106 – 138. https://doi.org/10.1016/j.tcs.2012.02.027 Functional and Logic Programming.

[28] Lionel Vaux. 2007. *λ-calcul différentiel et logique classique : interactions calculatoires. (Differential λ-calculus and classical logic : their computational interactions).* Ph. D. Dissertation. University of the Mediterranean, Marseille, France. https://tel.archives-ouvertes.fr/tel-00194149

[29] Lionel Vaux. 2007. The differential $\lambda\mu$-calculus. *Theor. Comput. Sci.* 379, 1-2 (2007), 166–209. https://doi.org/10.1016/j.tcs.2007.02.028

[30] Lionel Vaux. 2019. Normalizing the Taylor expansion of non-deterministic λ-terms, via parallel reduction of resource vectors. *Log. Methods Comput. Sci.* 15, 3 (2019). https://doi.org/10.23638/LMCS-15(3:9)2019