

Taylor Subsumes Scott, Berry, Kahn and Plotkin*

DAVIDE BARBAROSSA, Université Paris 13, Sorbonne Paris Cité, LIPN, CNRS UMR 7030, France

GIULIO MANZONETTO, Université Paris 13, Sorbonne Paris Cité, LIPN, CNRS UMR 7030, France

The speculative ambition of replacing the old theory of program approximation based on syntactic continuity with the theory of resource consumption based on Taylor expansion and originating from the differential λ -calculus is nowadays at hand. Using this resource sensitive theory, we provide simple proofs of important results in λ -calculus that are usually demonstrated by exploiting Scott's continuity, Berry's stability or Kahn and Plotkin's sequentiality theory. A paradigmatic example is given by the Perpendicular Lines Lemma for the Böhm tree semantics, which is proved here simply by induction, but relying on the main properties of resource approximants: strong normalization, confluence and linearity.

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

Additional Key Words and Phrases: Lambda calculus, Taylor expansion, Böhm trees, Linear Logic.

ACM Reference Format:

Davide Barbarossa and Giulio Manzonetto. 2020. Taylor Subsumes Scott, Berry, Kahn and Plotkin. *Proc. ACM Program. Lang.* 4, POPL, Article 1 (January 2020), 23 pages. <https://doi.org/10.1145/3371069>

1 INTRODUCTION

The theory of analytical differentiation has been fruitfully applied to many areas of computer science, like deep learning [Baydin et al. 2017] (choosing weights in neural nets by gradient descent optimisation methods), algorithmic differentiation [Fiege et al. 2018; Griewank et al. 2018] (providing techniques for efficient differentiation via source code transformation) and, recently, programming languages with first-class differentiation have begun to appear, see e.g. [Walter and Lehmann 2013].

In the pioneering article [Ehrhard and Regnier 2003], Ehrhard and Regnier introduce a derivative operator in the higher-order functional setting, thus defining a *differential λ -calculus* whose study is shedding new light on the fundamental relationship between linearity and control of resources. The main novelty of this approach, which is also its strength, is that it brings quantitative mathematics in the programming language discipline by introducing a notion of (syntactic) derivative of a program. Just like in analysis the derivative Df furnishes information on the function f , as it measures the variation of f as its input changes, in functional programming languages it is possible to infer quantitative properties of a program P by studying its derivative DP . The differential λ -calculus has solid mathematical roots in quantitative semantics of linear logic, and extends the celebrated Curry-Howard “proofs-as-programs and formulae-as-types” correspondence as follows:

*This article is an extended version of the abstract [Barbarossa and Manzonetto 2019] presented at TLLA'2019.

Authors' addresses: Davide Barbarossa, Laboratoire d'Informatique de Paris-Nord, Université Paris 13, Sorbonne Paris Cité, LIPN, CNRS UMR 7030, 99, av. J.-B. Clément, Villetaneuse, F-93430, France, barbarossa@lipn.univ-paris13.fr; Giulio Manzonetto, Laboratoire d'Informatique de Paris-Nord, Université Paris 13, Sorbonne Paris Cité, LIPN, CNRS UMR 7030, 99, av. J.-B. Clément, Villetaneuse, F-93430, France, manzonetto@lipn.univ-paris13.fr.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2020 Copyright held by the owner/author(s).

2475-1421/2020/1-ART1

<https://doi.org/10.1145/3371069>

Function		Programming Feature		Logical System
analytical	\iff	duplication/erasure allowed	\iff	proofs with structural/logical rules
linear	\iff	resources used exactly once	\iff	proofs with logical rules only

This opens the way for the application of interesting results developed in the study of analytical differentiation to this area of computer science. For instance, Ehrhard and Regnier proposed to approximate the behaviour of an ordinary program P by performing its *Taylor expansion*:

$$P x = \sum_{n=0}^{\infty} \frac{1}{n!} (D^n P \cdot x^n) 0$$

where $\frac{1}{n!}$ is a numerical coefficient, $D^n P \cdot x^n$ stands for the n -th derivative of P applied to n non erasable nor duplicable copies of x , and 0 plays the role of a “guardian” in the sense that if $D^n P$ is unable to use exactly n times its argument x than the whole program is annihilated (it reduces to the empty program 0). By fully developing each application occurring in P into its corresponding Taylor expansion, it is possible to represent P as an infinite power series of purely differential programs all of which contain only linear applications, and regular applications to 0 . This *full* Taylor expansion defines the best linear approximation of a program and is strictly related to the length of its linear head reduction and hence to its execution time [Ehrhard and Regnier 2006a]. The target language of the (full) Taylor expansion can be presented as a *resource calculus* where terms are applied to bags (finite multisets) of linear resources, which is more familiar to computer scientists because it shares similarities with Boudol’s λ -calculus with multiplicities [Boudol 1993]. From the denotational viewpoint, this resource calculus – and therefore the whole λ -calculus via the aforementioned Taylor expansion – can be naturally interpreted in quantitative semantics of linear logic, the simplest being the relational semantics that can be also represented in logical form as a type system where types are endowed with a non-idempotent intersection [Paolini et al. 2017].

The paradigm shift initiated with the work of Ehrhard and Regnier has originated several different – but interrelated – axes of research. From a mathematical perspective, Blute, Cockett and Seely launched in the mid 2000s a research programme aiming at providing an axiomatic description of differentiation in the categorical setting. Not only these investigations were successful, leading to the notion of (*Cartesian*) *differential categories* [Blute et al. 2006, 2009], but they generated many related concepts as tangent structures [Cockett and Cruttwell 2014], categorical antiderivatives [Ehrhard 2018], integral categories [Cockett and Lemay 2019] and the like. From a more computer science oriented perspective, researchers have used the resource calculus and its quantitative models to produce resource sensitive type systems and cost models for several functional programming languages [Accattoli 2018]. For instance, De Carvalho proved that an upper bound for the number of (head) reduction steps of a program can be extracted from a type derivation in relevant systems [de Carvalho 2018], a result recently refined in [Accattoli et al. 2018] where the authors are able to calculate the exact value by exploiting a notion of “tight” typings. In [Laird et al. 2013], Laird et al. extend the relational semantics of linear logic by adding weights, that allow to provide unified account of nondeterministic, probabilistic and algebraic PCF programs. The interpretation of a program not only contains information concerning the amount of steps needed to compute a value, but also on the number of “different ways” or even its probability. This approach can be generalized further to encompass the quantum PCF of [Pagani et al. 2014] by considering categories as weights, and profunctors rather than relations, as done in [Tsukada et al. 2018] where Tsukada et al. define a ‘rigid’ Taylor expansion and show that it allows to characterize the interpretation of programs in bicategories of generalised species of structures [Fiore et al. 2007].

The approach to program approximation via Taylor expansion is formally related to the classic approximation theory, based on Böhm trees and Scott’s continuity, via a Commutation Theorem stating that it is equivalent to Taylor expand a program and then normalize it, or to compute its Böhm tree and then perform its Taylor expansion [Ehrhard and Regnier 2008]. A crucial advantage

of the former approach is that it is easily generalizable from λ -calculus to other languages thanks to its well-established and versatile logical background. Indeed, the numerical coefficients in the Taylor expansion are well suited to model nondeterministic [Bucciarelli et al. 2012], probabilistic [Lago and Zorzi 2012] and algebraic calculi [Vaux 2009] that are increasingly attracting the interest of the scientific community. For instance, in [Lago and Leventis 2019] the authors show that the Taylor expansion model is adequate for the probabilistic λ -calculus and satisfies an analogous commutation property with respect to the nondeterministic Böhm trees recently introduced in [Leventis 2018]. A similar situation has arisen in the call-by-value setting — despite the fact that the call-by-value λ -calculus was introduced by Plotkin several decades ago [Plotkin 1975], a notion of Böhm trees adapted to this paradigm has only been introduced last year [Kerinec et al. 2018] and the authors claim that they were inspired by the call-by-value Taylor expansion defined in [Ehrhard 2012].

What we find slightly disappointing in these recent articles is that researchers do make an effort to generalize such notions and results, but do not explore their consequences in the new settings (with the notable exception of [Kerinec et al. 2018], where some applications are actually given). We believe that this phenomenon has multiple reasons. On the one hand, there is a common belief that the Taylor expansion is only useful to investigate quantitative properties of programs, on the other hand, in the current literature there are not many examples of qualitative properties demonstrated via these notions. In the present paper we investigate the full potential of this approach and show that fundamental results in λ -calculus, that are usually proved using Scott’s continuity [Scott 1972], Berry’s stability [Berry 1978] or Kahn and Plotkin’s sequentiality theory [Kahn and Plotkin 1978], can be endowed with simpler proofs that exploits the Commutation Theorem and the main properties enjoyed by the resource calculus, namely confluence, strong normalization (SN) and linearity. We first exhibit a proof sketch of the Commutation Theorem to show that such a result is not so difficult to achieve, as it might seem at first sight reading [Ehrhard and Regnier 2008].

Once that this preliminary work is done, we show that if two λ -terms M, N share the same Böhm tree, this property is preserved when they are plugged in the same context $C[\]$. This is needed to ensure that the Böhm tree semantics actually induces a λ -theory and usually proved via the theory of syntactic continuity, while we proceed by structural induction on $C[\]$. We then consider more fundamental statements like the Genericity Lemma [Barendregt 1984, Prop. 14.3.24] that motivates the nowadays well established choice of identifying unsolvable λ -terms as completely undefined results and show that it is a direct consequence of the Commutation Theorem. To prove the statements expressing continuity and stability it is convenient to perform a finer analysis of resource approximants, by isolating the linearized/affine ones that represent the usual Böhm tree approximants in a minimal way, but otherwise the proof techniques employed are actually similar. We conclude by exhibiting a simple inductive proof of the Perpendicular Lines Lemma (PLL, for short) [Barendregt 1984, Thm 14.4.12] stating that if a program is constant on n “perpendicular lines”, then it must be constant everywhere. This lemma has several important consequences, among which the celebrated non-definability of “parallel or” stands out. The PLL was originally demonstrated for the Böhm tree semantics by applying Kahn and Plotkin’s theory of sequentiality, then proved for β -conversion via the Reduction under Substitution property [Endrullis and de Vrijer 2008], while a counterexample was found for its closed term model [Statman and Barendregt 1999]. As we will show, our proof technique can be used for proving PLL both for the open and for the closed term model of Böhm trees, the latter constituting — as far as we know — an original result.

As a disclaimer, we do not claim that the proofs so obtained are simpler to read and follow: on the one hand this will depend on the reader’s familiarity with linear logic and the resource calculus, on the other hand there is inevitably a price to pay when considering linearized program approximants, namely the appearance of several indices to keep track of the different occurrences of a variable. What we claim is that our proofs are based on simpler principles — induction rather than the

coinduction naturally arising when considering Böhm trees – and that the fact that approximants cannot erase or duplicate their arguments is a very strong property that can be fruitfully used to simplify the proofs (as an example, see the statement of the claim in the proof of Theorem 6.1).

Finally, we would like to stress the fact that we choose the regular untyped λ -calculus to give a glimpse of the full potential of these proof methods because it is the original framework where the Taylor expansion was introduced [Ehrhard and Regnier 2003] and the simplest – in particular, in this setting it is customary to overlook the coefficients of the Taylor expansion that represent unnecessary information for proving qualitative results. However, the proof techniques that we discuss do generalize to more elaborate non-deterministic, probabilistic, algebraic systems both typed and untyped, both in call-by-name and in call-by-value (even in call-by-push value, see [Ehrhard and Guerrieri 2016; Ehrhard and Tasson 2019]).

2 THE λ -CALCULUS IN A NUTSHELL

To keep this article as self-contained as possible, we summarize some definitions and results concerning λ -calculus that are needed subsequently. With regard to the λ -calculus, we mainly follow the notation and terminology of Barendregt's first book [Barendregt 1984].

2.1 Its syntax

The set Λ of λ -terms over an infinite set \mathbb{V} of variables is defined by the simplified grammar :

$$\Lambda : \quad L, M, N ::= x \mid \lambda x.M \mid MN \quad (\text{for } x \in \mathbb{V})$$

We assume that application associates to the left and has a higher precedence than abstraction. For instance, $\lambda xyz.xyz$ stands for $\lambda x.(\lambda y.(\lambda z.((xy)z)))$. Moreover, for $n \geq 0$, we write $M^n N$ as an abbreviation for $M(M(\cdots(MN)\cdots))$ (n times), $\lambda \vec{x}.M$ for $\lambda x_1 \dots \lambda x_n.M$ and \vec{M} for M_1, \dots, M_n .

The set $\text{FV}(M)$ of *free variables* of M and the α -conversion are defined as in Section 1.2 of [Barendregt 1984]. Hereafter, we consider λ -terms up to α -conversion and denote by $=$ syntactic equality.

Definition 2.1. A λ -term M is called *closed*, or a *combinator*, whenever $\text{FV}(M) = \emptyset$. We denote by Λ° the set of all closed λ -terms.

Let us recall the definition of k -ary contexts, that will be often considered in the rest of the paper.

Definition 2.2. Let $(\xi_n)_{n \in \mathbb{N}}$ be an enumeration of fresh variables, in particular we assume $\xi_n \notin \mathbb{V}$.

(i) For $k \in \mathbb{N}$, a k -context $C[\vec{\xi}]$ is a λ -term possibly containing the variables $\vec{\xi} = \xi_1, \dots, \xi_k$, i.e.:

$$C[\vec{\xi}], C'[\vec{\xi}] ::= \xi_i \mid x \mid \lambda x.C[\vec{\xi}] \mid (C[\vec{\xi}])(C'[\vec{\xi}]) \quad (\text{for } i \in \{1, \dots, k\})$$

The variables ξ_i occurring in $C[\vec{\xi}]$ are traditionally called *holes*.

(ii) Given a k -context $C[\vec{\xi}]$ and λ -terms M_1, \dots, M_k , we denote by $C[M_1, \dots, M_k]$ the λ -term obtained by simultaneously replacing each M_i for ξ_i in $C[\vec{\xi}]$, possibly with captures of free variables.

(iii) A 1-context is simply called a *context* and we denote its hole by ξ .

(iv) A context $C[\xi]$ is called *single-hole* whenever it contains exactly one occurrence of ξ .

Consider now an arbitrary binary relation $R \subseteq \Lambda \times \Lambda$. We say that R is *context closed* if $M R N$ entails $C[M] R C[N]$ for all single-hole contexts $C[\xi]$. The *contextual closure* of a relation R is the smallest context closed relation containing R .

Definition 2.3. The β -reduction \rightarrow_β is defined as the contextual closure of the rule

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

where $M\{N/x\}$ denotes the simultaneous substitution of N for all free occurrences of x in M , subject to the usual proviso about renaming bound variables in M to avoid capture of free variables in N .

As it is customary, we write \rightarrow_β for the multistep β -reduction and $=_\beta$ for β -conversion. Moreover, we denote by $\text{nf}_\beta(M)$ the β -normal form of M , if it exists.

Notation. We will use the following notations for specific λ -terms:

$$\begin{aligned} I &= \lambda x.x, & K &= \lambda xy.x, & F &= \lambda xy.y, & \Delta &= \lambda x.xx, & \Omega &= \Delta\Delta, \\ Y &= \lambda f.(\lambda x.f(xx))(\lambda x.f(xx)), & \Theta &= (\lambda fx.x(ffx))(\lambda fx.x(ffx)), \end{aligned}$$

where I is the identity, K and F are the first and the second projections, Y and Θ are Curry's and Turing's fixed point combinators, respectively, and Ω the paradigmatic looping combinator.

2.2 Solvability and Böhm trees

Lambda terms are classified into solvable or unsolvable depending on their capability of interaction with the environment, which is represented in this setting by a context.

Definition 2.4. A λ -term M is called solvable if there exists a context having shape $C[\xi] = (\lambda x_1 \dots \lambda x_n.\xi)N_1 \dots N_k$ such that $C[M] =_\beta I$. Otherwise, we say that M is unsolvable.

A λ -term M is in *head normal form (hnf)* if it has the shape $\lambda x_1 \dots \lambda x_n.yM_1 \dots M_k$ for some $n, k \geq 0$. It is well-known that if M has an hnf, then such a normal form can be reached by *head reductions* \rightarrow_h , i.e., by repeatedly contracting its *head redex* $\lambda x_1 \dots \lambda x_n.(\lambda y.P)QM_1 \dots M_k$.

Theorem 2.5 (Wadsworth [Wadsworth 1976]).

A λ -term M is solvable if and only if M has a head normal form.

Example 2.6. Using Wadsworth's characterization it is easy to verify that I, K, F, Δ, Y are all solvable, while Ω, YI and ΘI are unsolvable.

In order to describe abstractly the possibly infinite behaviour of a λ -term M , Barendregt introduced in [Barendregt 1977] the notion of Böhm tree of M .

Definition 2.7. The Böhm tree $\text{BT}(M)$ of a λ -term M is defined coinductively¹ as follows:

- if M is solvable then $M \rightarrow_h \lambda x_1 \dots \lambda x_n.yM_1 \dots M_k$ and we define:

$$\text{BT}(M) = \begin{array}{c} \lambda x_1 \dots \lambda x_n.y \\ \swarrow \quad \searrow \\ \text{BT}(M_1) \quad \dots \quad \text{BT}(M_k) \end{array}$$

- otherwise, M is unsolvable and we define $\text{BT}(M) = \perp$.

Example 2.8. We now provide some examples of Böhm trees of notable λ -terms:

$$\begin{array}{ccc} \text{BT}(\Delta) = \lambda x.x & \text{BT}(\Omega) = \perp & \text{BT}(Y) = \lambda f.f \\ \quad \quad \quad \downarrow & & \quad \quad \quad \downarrow \\ \quad \quad \quad x & & \quad \quad \quad f \\ & & \quad \quad \quad \downarrow \\ & & \quad \quad \quad f \\ & & \quad \quad \quad \downarrow \\ & & \quad \quad \quad f \\ & & \quad \quad \quad \vdots \end{array} \quad \text{BT}(Pz) = \begin{array}{c} \lambda x_0.x_0 \\ \quad \quad \quad \downarrow \\ \lambda x_1.x_1 \\ \quad \quad \quad \downarrow \\ \lambda x_2.x_2 \\ \quad \quad \quad \downarrow \\ \lambda x_3.x_3 \\ \quad \quad \quad \vdots \end{array}$$

where $P = Y(\lambda yzx.x(yz))$ satisfies $Pz =_\beta \lambda x.x(Pz)$. Note that $z \in \text{FV}(P')$ for all P' such that $Pz \rightarrow_\beta P'$, but it does not occur in $\text{BT}(Pz)$. In this case we say that z is "pushed into infinity".

It is well-known that Böhm trees are invariant under β -conversion [Barendregt 1984, Ch. 10].

¹The interested reader is invited to consult the article [Lassen 1999] for a more detailed discussion of the coinduction principle behind this definition.

Lemma 2.9. For $M, N \in \Lambda$, if $M =_{\beta} N$ then $\text{BT}(M) = \text{BT}(N)$.

As pointed out by Jacobs and Rutten in [Jacobs and Rutten 1997], Böhm trees constitute one of the first examples of a coinductive definition. However, the scientific community has only recently acquired enough familiarity with the coinductive principle [Kozen and Silva 2017], for decades researchers tried to harness the infinitary nature of Böhm trees with finite approximants and syntactic continuity [Barendregt 1984, §14.2]. We quickly review the classic approach based on such approximants, and recall the main notions and results.

Definition 2.10. (i) The set Λ_{\perp} of λ_{\perp} -terms is inductively defined via the simplified grammar:

$$\Lambda_{\perp} : \quad M, N ::= \perp \mid x \mid \lambda x.M \mid MN$$

(ii) Let $\leq_{\perp} \subseteq \Lambda_{\perp} \times \Lambda_{\perp}$ be the contextual preorder generated by setting $\perp \leq M$, for all $M \in \Lambda_{\perp}$.

(iii) The λ_{\perp} -terms are endowed with the reduction $\rightarrow_{\beta_{\perp}}$, namely β -reduction augmented by

$$\lambda x.\perp \rightarrow_{\perp} \perp \quad \perp M_1 \cdots M_n \rightarrow_{\perp} \perp \quad (\text{for } n > 0)$$

(iv) The subset $\mathcal{A} \subseteq \Lambda_{\perp}$ of finite approximants is defined by the following simplified grammar:

$$\mathcal{A} : \quad P, Q ::= \perp \mid \lambda x_1 \dots x_n.yP_1 \cdots P_k \quad (\text{for } n, k \geq 0)$$

(v) Two approximants $P_1, P_2 \in \mathcal{A}$ are compatible if there exists $Q \in \mathcal{A}$ such that $P_1 \leq_{\perp} Q \leq_{\perp} P_2$. In case $P_1, P_2 \neq \perp$, we must have $P_1 = \lambda x_1 \dots x_n.yP_{11} \cdots P_{1k}$ and $P_2 = \lambda x_1 \dots x_n.yP_{21} \cdots P_{2k}$ for the same $n, k \geq 0$ and for some P_{1j}, P_{2j} that are compatible for all j ($1 \leq j \leq k$).

(vi) Given two compatible $P_1, P_2 \in \mathcal{A}$, their supremum $P_1 \vee P_2$ is defined inductively as follows:

$$P_1 \vee P_2 = \begin{cases} \lambda \vec{x}.y(P_{11} \vee P_{21}) \cdots (P_{1k} \vee P_{2k}) & \text{if } P_1 = \lambda \vec{x}.yP_{11} \cdots P_{1k} \text{ and } P_2 = \lambda \vec{x}.yP_{21} \cdots P_{2k}, \\ P_1 & \text{if } P_2 = \perp, \\ P_2 & \text{if } P_1 = \perp. \end{cases}$$

(vii) Given a λ -term M , the set of finite approximants of M is defined by:

$$\mathcal{A}(M) = \{P \in \mathcal{A} \mid \exists N \in \Lambda. M \rightarrow_{\beta} N \text{ and } P \leq_{\perp} N\}$$

The finite approximants in \mathcal{A} are exactly the λ_{\perp} -terms in β_{\perp} -normal form.

Lemma 2.11. (i) $M \in \Lambda_{\perp}$ is in β_{\perp} -normal form if and only if $M \in \mathcal{A}$.

(ii) For $M \in \Lambda$, the set $\mathcal{A}(M)$ is an ideal, namely non-empty, downward closed and directed.

An issue intrinsic to these definitions is the difficulty of characterizing $\mathcal{A}(MN)$ in terms of $\mathcal{A}(M)$ and $\mathcal{A}(N)$. Indeed, not only \mathcal{A} is not closed under application, but applying PQ can give rise to a term without a β_{\perp} -normal form, like Ω . This is the main reason why, in order to prove the Approximation Theorem below, one needs to develop the whole theory of syntactic continuity.

Theorem 2.12 (Approximation Theorem). For all $M \in \Lambda$

$$\text{BT}(M) = \bigvee \mathcal{A}(M)$$

Such a supremum always exists because, by Lemma 2.11(ii), $\mathcal{A}(M)$ is an ideal with respect to \leq_{\perp} .

2.3 Lambda theories

The equational theories of λ -calculus, called “ λ -theories” become the main object of study when considering the computational equivalence more important than the process of computation itself. We now recall their definition together with some interesting properties and provide some examples.

Definition 2.13. (i) A relation $R \subseteq \Lambda \times \Lambda$ is called a congruence if it is a context closed equivalence.

(ii) A λ -theory \mathcal{T} is any congruence containing the β -conversion.

Given a λ -theory \mathcal{T} , we write $M =_{\mathcal{T}} N$ for $(M, N) \in \mathcal{T}$. The set of λ -theories, ordered by set-theoretical inclusion, constitutes a complete lattice having a quite rich structure, see for instance [Lusin and Salibra 2004]. A λ -theory \mathcal{T} is called *consistent* if $\mathcal{T} \neq \Lambda \times \Lambda$; \mathcal{T} is *sensible* if it is consistent and equates all unsolvable λ -terms. We are mainly interested in the following λ -theories.

Example 2.14. Consider the following relations on $\Lambda \times \Lambda$:

$$\begin{aligned} \lambda &= \{(M, N) \mid M =_{\beta} N\} \\ \mathcal{B} &= \{(M, N) \mid \text{BT}(M) = \text{BT}(N)\} \end{aligned}$$

Now, λ is clearly the least λ -theory, while for proving that \mathcal{B} is a λ -theory one needs to check that it is context closed (See Section 5, for an original proof of this known fact). As another example, consider also the least sensible λ -theory \mathcal{H} , generated by equating all unsolvable λ -terms. We have:

- $\lambda \subsetneq \mathcal{H}$. The inclusion is trivial because of the minimality of λ among λ -theories. It is strict because, for unsolvable λ -terms like Ω and YI , we have $\Omega \neq_{\beta} YI$ while $\Omega =_{\mathcal{H}} YI$.
- $\mathcal{H} \subsetneq \mathcal{B}$. The inclusion holds since the λ -theory \mathcal{B} is sensible, and it is strict because all fixed point combinators are equated in \mathcal{B} while they can be different in \mathcal{H} . E.g. $\Theta \neq_{\mathcal{H}} Y$, but $\Theta =_{\mathcal{B}} Y$.

We will also consider an inequational version of \mathcal{B} , where the preorder on λ -terms is inherited from Definition 2.10 as follows.

Definition 2.15. For $M, N \in \Lambda$ we define

$$M \sqsubseteq_{\mathcal{B}} N \iff \mathcal{A}(M) \subseteq \mathcal{A}(N)$$

It is easy to verify that the relation $\sqsubseteq_{\mathcal{B}}$ captures the usual approximation order on Böhm trees, namely $M \sqsubseteq_{\mathcal{B}} N$ exactly when $\text{BT}(M)$ is obtained from $\text{BT}(N)$ by replacing some subtrees by \perp . As a consequence, we have that $M =_{\mathcal{B}} N$ holds if and only if both $M \sqsubseteq_{\mathcal{B}} N$ and $N \sqsubseteq_{\mathcal{B}} M$ do.

Example 2.16. Consider two λ -terms M, N having the following infinite Böhm trees

$$\text{BT}(M) = [I, [\perp, [I, [\perp, \dots]]]], \quad \text{BT}(N) = [I, [K, [I, [K, \dots]]]],$$

where $[X, Y]$, for fresh x , denotes Church's pairing $\lambda x.xXY$. It is easy to check that $M \sqsubseteq_{\mathcal{B}} N$ holds.

We end this section by recalling that every λ -theory has an associated (open/closed) term model.

Definition 2.17. (i) Given a λ -theory \mathcal{T} , its (open) term model $\mathcal{M}_{\mathcal{T}} = (\Lambda/\mathcal{T}, \cdot)$ consists of the set of equivalence classes $[M]_{\mathcal{T}}$, together with the operation of application $[M]_{\mathcal{T}} \cdot [N]_{\mathcal{T}} = [MN]_{\mathcal{T}}$.
(ii) The closed term model $\mathcal{M}_{\mathcal{T}}^{\circ}$, is the subalgebra of $\mathcal{M}_{\mathcal{T}}$ generated by the equivalence classes of closed λ -terms.

3 THE RESOURCE CALCULUS IN A NUTSHELL

We recall the syntax and the main properties of the *resource calculus* [Ehrhard and Regnier 2003]. In this calculus, a term cannot be just applied to another term, but rather to a “bag” (finite multiset) of resources. Each resource in a bag is linear in the sense that it cannot be erased or duplicated, it must be consumed during the reduction.

Definition 3.1. The set Λ^r of resource terms and the set Λ^b of bags are defined by the simplified grammar:

$$\begin{aligned} \Lambda^r : \quad s, t, u, v & ::= x \mid \lambda x.t \mid t b \\ \Lambda^b : \quad b & ::= [t_1, \dots, t_k] \quad (\text{for } k \geq 0) \end{aligned}$$

where $[t_1, \dots, t_k]$ represents a multiset. The empty bag will be denoted by 1.

The application of a function to a bag, as in $(\lambda x.s)[t_1, \dots, t_k]$, should be understood as follows: the program s during its execution must perform n “calls” to its argument x , one for each occurrence of x in s , each time receiving as value a different t_i non-deterministically chosen from the bag (in the reduction we consider a formal sum of all possibilities). In case of a mismatch $n \neq k$, s “raises an exception” annihilating the whole term, so the outcome of the computation is the empty sum 0.

For resource terms we adopt the same conventions as for λ -terms about associativity, α -conversion, free variables. We now introduce the syntactic category of *formal sums* of resource terms.

Definition 3.2. *Let $\mathbb{2}$ be the semiring of boolean values and $2\langle\Lambda^r\rangle$ be the free 2-module generated by Λ^r .*

This amounts to saying that $+$ is idempotent with neutral element 0, so $\mathbb{T} \in 2\langle\Lambda^r\rangle$ can be seen as a finite set $\{t_1, \dots, t_k\} \in \mathcal{P}(\Lambda^r)$ that will be however represented by $t_1 + \dots + t_k$. Thus, for $t \in \Lambda^r$ and $\mathbb{S}, \mathbb{T} \in 2\langle\Lambda^r\rangle$, $t \in \mathbb{T}$, $\mathbb{T} \subseteq \mathbb{S}$ and $\mathbb{T} \cap \mathbb{S}$ have the obvious meaning. Similarly, we write $2\langle\Lambda^b\rangle$ for the set of finite sums of bags, whose elements are denoted \mathbb{B}, \mathbb{B}' , and use analogous conventions.

Definition 3.3. *Given $t \in \Lambda^r$, the linear substitution $t\langle b/x \rangle \in 2\langle\Lambda^r\rangle$ of a bag $b = [s_1, \dots, s_k]$ for a variable x in t , is defined as follows:*

$$t\langle b/x \rangle = \begin{cases} \sum_{\sigma \in \mathfrak{S}_k} t\{s_{\sigma(1)}/x_1, \dots, s_{\sigma(k)}/x_k\} & \text{if } \deg_x(t) = k, \\ 0 & \text{otherwise,} \end{cases}$$

where \mathfrak{S}_k is the group of permutations of $\{1, \dots, k\}$, $\deg_x(t)$ is the number of free occurrences of x in t , and x_1, \dots, x_k is an enumeration of such occurrences, so that $t\{s_{\sigma(i)}/x_i\}$ denotes the resource term obtained by substituting $s_{\sigma(i)}$ for x_i in t .

As a syntactic sugar, and not as actual syntax, we extend all constructs to sums by (bi)linearity:

$$\lambda x.(\sum_i t_i) = \lambda x. \sum_i t_i, \quad (\sum_i t_i)(\sum_j b_j) = \sum_{i,j} t_i b_j, \quad [\sum_i s_i, \mathbb{T}_1, \dots, \mathbb{T}_k] = \sum_i [s_i, \mathbb{T}_1, \dots, \mathbb{T}_k].$$

In particular $\lambda x.0 = 0s = t0 = 0$ and $[0, s_1, \dots, s_k] = 0$, therefore 0 annihilates the whole term.

Definition 3.4. *This calculus is endowed with a reduction $\rightarrow_r \subseteq \Lambda^r \times 2\langle\Lambda^r\rangle$*

$$(\lambda x.t)[s_1, \dots, s_k] \rightarrow_r t\langle [s_1, \dots, s_k]/x \rangle$$

which is the least one closed under the following rules:

$$\frac{t \rightarrow_r \mathbb{T}}{\lambda x.t \rightarrow_r \lambda x.\mathbb{T}} \quad \frac{t \rightarrow_r \mathbb{T}}{tb \rightarrow_r \mathbb{T}b} \quad \frac{b \rightarrow_r \mathbb{B}}{tb \rightarrow_r t\mathbb{B}} \quad \frac{t \rightarrow_r \mathbb{T}}{[t, s_1, \dots, s_k] \rightarrow_r [\mathbb{T}, s_1, \dots, s_k]}$$

and extends to a relation $\rightarrow_r \subseteq 2\langle\Lambda^r\rangle \times 2\langle\Lambda^r\rangle$ by setting:

$$t \rightarrow_r \mathbb{T} \text{ and } t \notin \mathbb{S} \implies t + \mathbb{S} \rightarrow_r \mathbb{T} + \mathbb{S}$$

Example 3.5. 1. $(\lambda x.x)[y] \rightarrow_r y$ while $(\lambda x.x)[y, y] \rightarrow_r 0$.

2. $(\lambda xy.x1)[z][z] \rightarrow_r (\lambda y.z1)[z] \rightarrow_r 0$, since erasing subterms is forbidden.

3. $(\lambda x.x[x][x])[y, y, z] \rightarrow_r y[y][z] + y[z][y] + z[y][y]$, because the sum is idempotent.

The following proposition guarantees that $\text{nf}(t)$ always exists and is a finite sum, possibly 0.

Proposition 3.6. *The reduction \rightarrow_r is confluent and strongly normalizable (SN).*

PROOF SKETCH. Strong normalization is trivial. Indeed, there is no duplication and contracting a redex eliminates an abstraction, hence $t \rightarrow_r \mathbb{S}$ entails that the size of t is bigger than the one of each $s \in \mathbb{S}$. One can check that the resource calculus is actually locally confluent, thus confluence follows from strong normalization and Newman’s lemma. \square

3.1 The Taylor expansion

The resource calculus can be seen as the target language of Ehrhard and Regnier's Taylor expansion of regular λ -terms. Actually, the original Taylor expansion transforms a λ -term into the power series of all its linear approximants, here, as we are considering an idempotent sum, we obtain a set.

Definition 3.7. The Taylor expansion $\mathcal{T}(M)$ of a λ -term M is a possibly infinite subset of Λ^r defined by structural induction on M as follows:

$$\begin{aligned}\mathcal{T}(x) &= \{x\}, \\ \mathcal{T}(\lambda x.M) &= \{\lambda x.t \mid t \in \mathcal{T}(M)\}, \\ \mathcal{T}(MN) &= \{tb \mid t \in \mathcal{T}(M), b \in \mathcal{M}_f(\mathcal{T}(N))\},\end{aligned}$$

where $\mathcal{M}_f(\mathcal{X})$ denotes the finite multisets of \mathcal{X} . The map $\mathcal{T}(\cdot)$ extends to $\lambda\perp$ -terms by setting $\mathcal{T}(\perp) = \emptyset$. For every $\mathcal{X} \subseteq \Lambda^r$, define $\text{FV}(\mathcal{X}) = \bigcup_{t \in \mathcal{X}} \text{FV}(t)$. It is easy to check that $\text{FV}(\mathcal{T}(M)) = \text{FV}(M)$.

Example 3.8. For $t \in \Lambda^r$, we use the notation $[n.t] = [t, \dots, t]$ (n times).

- (i) $\mathcal{T}(1) = \{\lambda x.x\}$ and $\mathcal{T}(\lambda x.x\perp) = \{\lambda x.x1\}$ (recall that 1 stands for the empty bag).
- (ii) $\mathcal{T}(\Delta) = \{\lambda x.x[n.x] \mid n \geq 0\}$, from which it follows:
- (iii) $\mathcal{T}(\Omega) = \{(\lambda x.x[n.x])[\lambda x.x[n_1.x], \dots, \lambda x.x[n_k.x]] \mid n, k, n_i \in \mathbb{N}\}$.

The Taylor expansion of a k -context is defined in the obvious way, by treating the ξ_i 's as variables.

Definition 3.9.

- (i) A resource context $c(\xi_1, \dots, \xi_k)$ is a resource term possibly containing the holes ξ_1, \dots, ξ_k .
- (ii) Given $c(\xi_1, \dots, \xi_k)$ and $t_1, \dots, t_k \in \Lambda^r$, $c(t_1, \dots, t_k)$ represents the resource term obtained by replacing t_i for all occurrences of ξ_i in $c(\xi_1, \dots, \xi_k)$, possibly with capture of free variables.
- (iii) Similarly, for $\mathbb{T}_1, \dots, \mathbb{T}_k \in 2(\Lambda^r)$, we set $c(\mathbb{T}_1, \dots, \mathbb{T}_k) = \sum_{t_1 \in \mathbb{T}_1} \dots \sum_{t_k \in \mathbb{T}_k} c(t_1, \dots, t_k)$.
- (iv) We write $\xi_i \# c(\xi_1, \dots, \xi_k)$ whenever ξ_i does not occur in $c(\xi_1, \dots, \xi_k)$.
- (v) The Taylor expansion is extended to k -contexts $C[\xi_1, \dots, \xi_k]$ by setting:

$$\mathcal{T}(\xi_i) = \{\xi_i\} \quad (\text{for } i \in \{1, \dots, k\}).$$

Example 3.10. $\mathcal{T}(\lambda x.\xi) = \{\lambda x.\xi\}$ and $\mathcal{T}(x\xi) = \{x[n.\xi] \mid n \geq 0\}$, using the notation of Example 3.8.

In the following (see Lemma 3.12), we need to represent a single element $t \in \mathcal{T}(C[M])$ by exhibiting the part of t arising from the context $C[\xi]$ separated from its parts arising from M . In other words, we would like to decompose t as $t = c(\xi|s)$ for $c(\xi|s) \in \mathcal{T}(C[\xi])$ and $s \in \mathcal{T}(M)$. Unfortunately, this is not always possible. E.g., for $C[\xi] = x\xi$, all resource contexts $c(\xi|s) \in \mathcal{T}(x\xi)$ have shape $x[\xi, \dots, \xi]$ for the same ξ so $t = x[z1, z[y]] \in \mathcal{T}(C[zy])$ cannot be decomposed in such a way because ξ can either become $z1$ or $z[y]$, but not both. For this reason, we first need to construct a resource context $c^\bullet(\xi^1, \xi^2) = x[\xi^1, \xi^2]$ where the occurrences of ξ are "linearized" (i.e. distinguished by adding different labels 1, 2) and then represent t as $c^\bullet(\xi^1, \xi^2|s) = x[z1, z[y]]$.

Definition 3.11. Consider distinguished variables $(\xi_i^\ell)_{i, \ell \in \mathbb{N}}$, where ξ_i^ℓ represents ξ_i with a label ℓ .

Definition 3.9 extends in the obvious way to resource contexts $c(\vec{\xi}_1, \dots, \vec{\xi}_k)$ with $\vec{\xi}_i = \xi_i^1, \dots, \xi_i^{n_i}$.

- (i) A resource context $c'(\xi_1^1, \dots, \xi_1^{n_1}, \dots, \xi_k^1, \dots, \xi_k^{n_k})$ is a linearization of $c(\xi_1, \dots, \xi_k)$ if it is obtained from $c(\xi_1, \dots, \xi_k)$ by replacing the j^{th} occurrence of ξ_i by the variable ξ_i^j . Since $\vec{\xi}_i$ contains a ξ_i^j for each occurrence of ξ_i in $c(\xi_1, \dots, \xi_k)$, we must have $n_i \geq \text{deg}_{\xi_i}(c(\xi_1, \dots, \xi_k))$.
- (ii) We denote by $c^\bullet(\vec{\xi}_1, \dots, \vec{\xi}_k)$ with $\vec{\xi}_i = \xi_i^1, \dots, \xi_i^{n_i}$, an arbitrary linearization of $c(\xi_1, \dots, \xi_k)$. Notice that $\xi_1, \dots, \xi_k \# c(\xi_1, \dots, \xi_k)$ entails $c^\bullet(\vec{s}_1, \dots, \vec{s}_k) = c(\xi_1, \dots, \xi_k)$ for all $s_i^j \in \Lambda^r$.

Intuitively, $c^\bullet(\vec{\xi})$ is a generic element of the sum $c([\xi^1, \dots, \xi^n]/\xi)$ where $n = \text{deg}_\xi(c(\xi|s))$ and $[\xi^1, \dots, \xi^n]/\xi$ denotes linear substitution as in Definition 3.3, but with possible capture of variables.

With these definitions in place, we can state the following properties.

Lemma 3.12. *Let $C[\xi]$ be a context and $M \in \Lambda$.*

- (i) *If $t \in \mathcal{T}(C[M])$ there are $c(\xi) \in \mathcal{T}(C[\xi])$ and $s_1, \dots, s_k \in \mathcal{T}(M)$ such that $t = c^\bullet(s_1, \dots, s_k)$.*
- (ii) *If $c(\xi) \in \mathcal{T}(C[\xi])$ but $\xi \# c(\xi)$ then $c^\bullet(s_1, \dots, s_k) \in \mathcal{T}(C[N])$ for all $s_i \in \Lambda^r$ and $N \in \Lambda$.*

The Taylor expansion enjoys the properties below, that can be promptly verified by structural induction on M .

Lemma 3.13 (Substitution). *For $M, N \in \Lambda$ and $x \in \mathbb{V}$:*

$$\mathcal{T}(M\{N/x\}) = \bigcup_{t \in \mathcal{T}(M)} \bigcup_{b \in M_f(\mathcal{T}(N))} t\langle b/x \rangle$$

Lemma 3.14. *For $M, N \in \Lambda_\perp$, if $M \leq_\perp N$ then $\mathcal{T}(M) \subseteq \mathcal{T}(N)$.*

Remark 3.15. *Since $t \in \mathcal{T}(P)$ follows the structure of the syntax tree of $P \in \mathcal{A}$, it must be an r -nf. E.g., $t \in \mathcal{T}(\lambda x.x\perp x)$ must have shape $\lambda x.x1[x, \dots, x]$, in this sense it follows its structure. To have a redex $(\lambda x.x)[y] \in \mathcal{T}(P)$ we should have $P = (\lambda x.x)y$ but this is impossible because P is β_\perp -normal.*

We now describe the classes of linear and affine resource terms that are interesting as they contain just enough information to easily reconstruct a term in Λ and Λ_\perp , respectively.

Definition 3.16.

- (i) *A resource term t is linearized (resp. affined) if every bag in t has cardinality (at most) 1.*
- (ii) *Every affined resource term $t \in \Lambda^r$ can be associated with a term $|t| \in \Lambda_\perp$ via the injection $|\cdot|$:*

$$|x| = x, \quad |\lambda x.t| = \lambda x.|t|, \quad |s[t]| = |s||t|, \quad |s1| = |s|\perp.$$

- (iii) *Conversely, every λ_\perp -term $M \neq \perp$ in \perp -normal form, can be sent into an affined $M^\circ \in \Lambda^r$ via the injection $(\cdot)^\circ$:*

$$x^\circ = x, \quad (\lambda x.M)^\circ = \lambda x.M^\circ, \quad (MN)^\circ = M^\circ[N^\circ], \quad (M\perp)^\circ = M^\circ 1.$$

For all $M \in \Lambda$ and linearized $t \in \Lambda^r$, we have $|M^\circ| = M$ and $|t|^\circ = t$, where β -nfs are sent into r -nfs and vice versa. For all $P \in \mathcal{A} - \{\perp\}$ and affined $t \in \Lambda^r$ in r -nf, we have $|P^\circ| = P$ and $|t|^\circ = t$.

Lemma 3.17. *The maps above define a bijection between:*

- (i) Λ and the set of linearized terms, where β -nfs are in correspondence with linearized r -nfs.
- (ii) $\mathcal{A} - \{\perp\}$ and the set of r -normal affined terms.

Remark 3.18. *The Taylor expansion of a λ -term M always contains a (unique) linearized term, namely M° . It also contains affined P° , for all $P \in \mathcal{A}$ satisfying $\perp \neq P \leq_\perp M$.*

4 TAYLOR'S COMMUTATION PROPERTY

Now that we have revised the definition of Taylor expansion of a λ -term, we can focus on its more dynamic aspects. In particular, we will see how to compute easily its r -normal form.

4.1 Normalizing the Taylor expansion

Since the resource calculus enjoys strong normalization the normal form of every subset of Λ^r can be defined pointwise.

Definition 4.1. *Given a possibly infinite set $X \subseteq \Lambda^r$, define:*

$$\text{NF}(X) = \bigcup \{\text{nf}(t) \mid t \in X\}$$

E.g., $\text{NF}(\Lambda^r)$ is the set of all resource terms in r -nf and $\text{NF}(\mathcal{T}(M))$ denotes the normal form of $\mathcal{T}(M)$.

Definition 4.2. For $M, N \in \Lambda$, we define $M \sqsubseteq_{\tau} N$ if and only if $\text{NF}(\mathcal{T}(M)) \subseteq \text{NF}(\mathcal{T}(N))$. Moreover, we write $M =_{\tau} N$ whenever $M \sqsubseteq_{\tau} N \sqsubseteq_{\tau} M$ holds.

By Proposition 3.6, the normal form of $\mathcal{T}(M)$ may possibly be empty but always exists. In general, it can be difficult to understand which resource terms actually belong to $\text{NF}(\mathcal{T}(M))$ for some $M \in \Lambda$. As an exercise, the reader can calculate $\text{NF}(\mathcal{T}(\Omega))$ or $\text{NF}(\mathcal{T}(Y))$.

Example 4.3.

- (i) $\mathcal{T}(1)$, $\mathcal{T}(\Delta)$ and $\mathcal{T}(\lambda x.x\perp)$ are already normal.
- (ii) $\text{NF}(\mathcal{T}(\Omega)) = \emptyset$ and $\text{NF}(\mathcal{T}(Y)) = \{\lambda f.f1, \lambda f.f[f1], \lambda f.f[f1, f[f1]], \dots\}$.

The next lemma from [Ehrhard and Regnier 2008] will be needed in Section 5.2.

Lemma 4.4. For $M \in \Lambda$ and $s, t \in \mathcal{T}(M)$, we have that $\text{nf}(s) \cap \text{nf}(t) \neq \emptyset$ entails $s = t$.

We now describe the relationship between the β -reduction of a λ -term and the r -reduction of its resource approximants.

Lemma 4.5. Let $M \in \Lambda$, $P \in \mathcal{A}$ and $t \in \Lambda^r$.

- (i) If $M \rightarrow_{\beta} N$ for some $N \in \Lambda$ and $t \in \mathcal{T}(N)$ then there exist $t' \in \mathcal{T}(M)$ and $\mathbb{T} \in 2\langle \Lambda^r \rangle$ such that $t' \rightarrow_r t + \mathbb{T}$.
- (ii) If $t \in \mathcal{T}(M)$ and $t \rightarrow_r \mathbb{T}$ then there exists $N \in \Lambda$ such that $M \rightarrow_{\beta} N$ and $\mathbb{T} \rightarrow_r \mathbb{T}' \subseteq \mathcal{T}(N)$.

As we look at terms, rather than sums, as the protagonists of resource calculus we formulate our statements for a single term as done in item (ii) of the lemma above. However, all our statements do generalize to finite sums in $2\langle \Lambda^r \rangle$.

Corollary 4.6. Let $M \in \Lambda$ and $t \in \mathcal{T}(M)$. Then there is $N \in \Lambda$ such that $M \rightarrow_{\beta} N$ and $\text{nf}(t) \subseteq \mathcal{T}(N)$.

PROOF (SKETCH). By induction on the length ℓ of a maximal reduction $t \rightarrow_r \text{nf}(t)$. If $\ell = 0$ then just take $N = M$. Otherwise, $\ell > 0$ and $t \rightarrow_r s + \mathbb{S} \rightarrow_r \text{nf}(t)$, where the latter reduction is strictly shorter. By Lemma 4.5(ii), there exist $M' \in \Lambda$ and $\mathbb{S}' \subseteq_f \mathcal{T}(M')$ such that $M \rightarrow_{\beta} M'$ and $s + \mathbb{S} \rightarrow_r \mathbb{S}' \rightarrow_r \text{nf}(t)$, so we conclude by induction hypothesis. \square

4.2 Commutation between Taylor and Böhm

Recall that the Taylor expansion $\mathcal{T}(\cdot)$ is extended to terms in Λ_{\perp} by setting $\mathcal{T}(\perp) = \emptyset$ and can be extended to sets $\mathcal{X} \subseteq \Lambda_{\perp}$, by taking $\mathcal{T}(\mathcal{X}) = \bigcup\{\mathcal{T}(P) \mid P \in \mathcal{X}\}$. This allows to define:

Definition 4.7. The Taylor expansion of $\text{BT}(M)$ is given by:

$$\mathcal{T}(\text{BT}(M)) = \mathcal{T}(\mathcal{A}(M))$$

Notice that $\text{FV}(\mathcal{T}(\text{BT}(M))) = \text{FV}(\text{BT}(M)) \subseteq \text{FV}(M)$.

Lemma 4.8. Let $M \in \Lambda$. Assume that $t \in \mathcal{T}(M)$ is in r -normal form, then there exists $P \in \mathcal{A}$ such that $t \in \mathcal{T}(P)$ and $P \leq_{\perp} M$.

PROOF. By induction on the r -normal structure of t . Assume that $t = \lambda \vec{x}. y b_1 \cdots b_k \in \mathcal{T}(M)$, then $M = \lambda \vec{x}. y M_1 \cdots M_k$ with $b_i \in M_f(\mathcal{T}(M_i))$. Now, every $s \in b_i$ is in r -normal form and belongs to $\mathcal{T}(M_i)$ whence by induction hypothesis there exists $P_s \in \mathcal{A}$ such that $P_s \leq_{\perp} M_i$ and $s \in \mathcal{T}(P_s)$. Define $P_i = \bigvee_{s \in b_i} P_s$, clearly $P_i \leq_{\perp} M_i$ whence $\lambda \vec{x}. y P_1 \cdots P_k \leq_{\perp} M$. By definition, for every $s \in b_i$, we have $P_s \leq_{\perp} P_i$, therefore Lemma 3.14 entails $s \in \mathcal{T}(P_i)$ so we conclude $t \in \mathcal{T}(\lambda \vec{x}. y P_1 \cdots P_k)$. \square

Theorem 4.9 (Commutation Theorem). For $M \in \Lambda$, we have

$$\text{NF}(\mathcal{T}(M)) = \mathcal{T}(\text{BT}(M))$$

So, if M is β -normalizable then $\text{NF}(\mathcal{T}(M)) = \mathcal{T}(\text{nf}_{\beta}(M))$.

PROOF. (\subseteq) If $t \in \text{NF}(\mathcal{T}(M))$ then there exists $t' \in \mathcal{T}(M)$ such that $t \in \text{nf}(t')$. By Corollary 4.6, there exists a λ -term N such that $M \rightarrow_{\beta} N$ and $t \in \text{nf}(t') \subseteq \mathcal{T}(N)$. So by Lemma 4.8, there exists $P \in \mathcal{A}$ such that $t \in \mathcal{T}(P)$ and $P \leq_{\perp} N$. By definition, $P \in \mathcal{A}(M)$ so we conclude $t \in \mathcal{T}(\text{BT}(M))$.

(\supseteq) Assume $t \in \mathcal{T}(\text{BT}(M))$, then there exists a reduction $M \rightarrow_{\beta} N$ such that $P \leq_{\perp} N$ and $t \in \mathcal{T}(P)$. By Remark 3.15 and Lemma 3.14 we get $t \in \text{NF}(\Lambda') \cap \mathcal{T}(N)$. We conclude because, by an iterated application of Lemma 4.5(i), we obtain a $t' \in \mathcal{T}(M)$ such that $t \in \text{nf}(t') \subseteq \text{NF}(\mathcal{T}(M))$. \square

Corollary 4.10. *M is solvable if and only if $\text{NF}(\mathcal{T}(M)) \neq \emptyset$.*

PROOF. Recall that M is solvable if and only if $\text{BT}(M) \neq \perp$, i.e. whenever there is $P \in \mathcal{A}(M) - \{\perp\}$. By definition, this holds exactly when there exists $P \in \mathcal{A}(M)$ such that $\mathcal{T}(P) \neq \emptyset$. We conclude since $\cup_{P \in \mathcal{A}(M)} \mathcal{T}(P) = \mathcal{T}(\text{BT}(M)) = \text{NF}(\mathcal{T}(M))$, where the last equality holds by Theorem 4.9. \square

Lemma 4.11.

(i) For $P, Q \in \mathcal{A}$ such that $P \neq \perp$, $P^{\circ} \in \mathcal{T}(Q)$ entails $P \leq_{\perp} Q$.

(ii) For all $M \in \Lambda$ and $P \in \mathcal{A} - \{\perp\}$, $P^{\circ} \in \text{NF}(\mathcal{T}(M))$ if and only if $P \in \mathcal{A}(M)$.

(iii) Let $M \in \Lambda$. There exists a (unique) linearized $t \in \text{NF}(\mathcal{T}(M))$ if and only if M is β -normalizable. In this case, we have $\text{nf}_{\beta}(M) = |t|$.

PROOF. (i) By structural induction on P .

(ii) (\Rightarrow) $P^{\circ} \in \text{NF}(\mathcal{T}(M))$ if there is $Q \in \mathcal{A}(M)$ such that $P^{\circ} \in \mathcal{T}(Q)$, so we conclude by (i).

(\Leftarrow) If $P \in \mathcal{A}(M)$, then $M \rightarrow_{\beta} M'$ for some M' such that $P \leq_{\perp} M'$, thus $P^{\circ} \in \mathcal{T}(M')$ by Remark 3.18. By Lemma 4.5(i), there is $t \in \mathcal{T}(M)$ such that $P^{\circ} \in \text{nf}(t)$ whence $P^{\circ} \in \text{NF}(\mathcal{T}(M))$.

(iii) If $t \in \text{NF}(\mathcal{T}(M))$ is linearized then $|t|$ is a β -normal \perp -free λ -term. By (ii), there is a reduction $M \rightarrow_{\beta} M'$ such that $|t| \leq_{\perp} M'$ but this entails $|t| = M'$. The converse implication is easy. \square

Corollary 4.12. *The relations \sqsubseteq_{τ} and $\sqsubseteq_{\mathcal{B}}$ coincide.*

PROOF. (\subseteq) Assume $M \sqsubseteq_{\tau} N$ and consider an approximant $P \in \mathcal{A}(M)$. If $P = \perp$ we are done. Otherwise, by Lemma 4.11(ii), we have $P^{\circ} \in \text{NF}(\mathcal{T}(M)) \subseteq \text{NF}(\mathcal{T}(N))$ so we conclude that $P \in \mathcal{A}(N)$.

(\supseteq) Assume that $\mathcal{A}(M) \subseteq \mathcal{A}(N)$ holds. By the Commutation Theorem, we have $\text{NF}(\mathcal{T}(M)) = \mathcal{T}(\mathcal{A}(M)) \subseteq \mathcal{T}(\mathcal{A}(N)) = \text{NF}(\mathcal{T}(N))$. \square

Thus the congruences $=_{\mathcal{B}}$ and $=_{\tau}$ coincide. In other words, considering the equalities induced on λ -terms by the Böhm tree semantics is equivalent to considering those induced by the normal form of their Taylor expansion. This is the key property that will be used in the following sections.

5 CONTINUITY AND STABILITY

We presented some well-known properties of λ -calculus and resource calculus. In particular the Commutation Theorem is due to Ehrhard and Regnier [Ehrhard and Regnier 2006a], it is well-established and has been generalized to several systems e.g. [Kerinec et al. 2018; Vaux 2019] – the most recent generalization is in the non-deterministic setting [Lago and Leventis 2019]. The proof we exhibited should convince the reader that this commutation is non-trivial, but not so difficult to demonstrate either (although considering coefficients does add a further layer of complexity).

The rest of the paper is devoted to showing that this theorem “subsumes” Scott’s (syntactic) continuity [Barendregt 1984, §14.3], Berry’s stability [Berry 1978] and Kahn-Plotkin’s sequentiality [Kahn and Plotkin 1978]. To achieve this goal, we show that many fundamental results in λ -calculus, usually demonstrated by exploiting these techniques, have simpler proofs that only rely on the Commutation Theorem and confluence, strong normalization and linearity of Λ' .

5.1 Scott's syntactic continuity

Our work starts from the consideration made in [Kerinec et al. 2018] that the monotonicity of $\text{NF}(\mathcal{T}(\cdot))$ with respect to \sqsubseteq_τ (hence, the contextuality of $=_\tau$) in the call-by-value setting has an easy inductive proof. From Corollary 4.12, it follows the contextuality of $\text{BT}(\cdot)$ which is traditionally obtained through syntactic continuity. The authors of [Kerinec et al. 2018] probably did not realize to what extent this proof-technique is generalizable and applicable. We begin by presenting how this methodology can be exploited to obtain a simple proof of the analogous statement for the regular (call-by-name) λ -calculus. We silently use Proposition 3.6, namely the confluence and SN.

Lemma 5.1 (Monotonicity of $\text{NF}(\mathcal{T}(\cdot))$). *Let $M, N \in \Lambda$ and $C[\xi]$ be a context. Then*

$$M \sqsubseteq_\tau N \implies C[M] \sqsubseteq_\tau C[N]$$

PROOF. We proceed by structural induction on $C[\xi]$.

Case $C[\xi] = x$. Trivial, as the context is constant.

Case $C[\xi] = \xi$. Trivial, as $\text{NF}(\mathcal{T}(M)) \subseteq \text{NF}(\mathcal{T}(N))$.

Case $C[\xi] = \lambda x.C'[\xi]$. For all $t \in \text{NF}(\mathcal{T}(\lambda x.C'[M]))$, there is $\lambda x.t' \in \mathcal{T}(\lambda x.C'[M])$ such that $\lambda x.t' \twoheadrightarrow_r \lambda x.\text{nf}(t') \ni t$. By induction hypothesis, we get $\text{nf}(t') \subseteq \text{NF}(\mathcal{T}(C'[M])) \subseteq \text{NF}(\mathcal{T}(C'[N]))$ whence $t \in \lambda x.\text{nf}(t') \subseteq \text{NF}(\mathcal{T}(\lambda x.C'[N]))$.

Case $C[\xi] = (C_1[\xi])(C_2[\xi])$. For every $t \in \text{NF}(\mathcal{T}((C_1[M])(C_2[M])))$ there exist $s \in \mathcal{T}(C_1[M])$ and $b \in \mathcal{M}_f(\mathcal{T}(C_2[M]))$ such that $sb \twoheadrightarrow_r \text{nf}(s)\text{nf}(b) \twoheadrightarrow_r \text{nf}(sb) \ni t$. By induction hypothesis, we have

$$\text{nf}(s) \subseteq \text{NF}(\mathcal{T}(C_1[M])) \subseteq \text{NF}(\mathcal{T}(C_1[N]))$$

and, by induction hypothesis and monotonicity of the map $\mathcal{M}_f(\cdot)$, we derive

$$\text{nf}(b) \subseteq \mathcal{M}_f(\text{NF}(\mathcal{T}(C_2[M]))) \subseteq \mathcal{M}_f(\text{NF}(\mathcal{T}(C_2[N]))),$$

so we conclude that $t \in \text{NF}(\mathcal{T}((C_1[N])(C_2[N])))$. \square

Since the relations \sqsubseteq_τ and $\sqsubseteq_{\mathcal{B}}$ coincide by Corollary 4.12, and $\text{BT}(M) = \text{BT}(N)$ holds exactly when $M \sqsubseteq_{\mathcal{B}} N \sqsubseteq_{\mathcal{B}} M$ does, we immediately obtain the contextuality of $\text{BT}(\cdot)$, namely [Barendregt 1984, Cor. 14.3.20(iii)].

Corollary 5.2. *Let M, N be λ -terms and $C[\xi]$ be a context. Then*

$$\text{BT}(M) = \text{BT}(N) \implies \text{BT}(C[M]) = \text{BT}(C[N])$$

We conclude that the relation \mathcal{B} , as defined in Example 2.14, is actually a λ -theory.

As a more substantial example, we now present a short proof of the Genericity Lemma [Barendregt 1984, Prop. 14.3.24], which constitutes a fundamental result in λ -calculus – its informal meaning is that if a λ -term N reducing to a completely defined output (a β -normal form) contains a subterm M which is meaningless (undefined), then M does not have any influence on the computation of this value, and therefore it may be replaced by any λ -term. In other words, the Genericity Lemma motivates the equivalence between “meaningless” and unsolvable λ -terms, which in its turn justifies the definition of the λ -theory \mathcal{H} (Example 2.14).

Theorem 5.3 (Genericity Lemma). *Let M be an unsolvable λ -term and $C[\xi]$ be a context. If $C[M]$ has a β -normal form then $C[M] =_\beta C[N]$ for all $N \in \Lambda$.*

PROOF. By Lemma 4.11(iii), there is a linearized $t \in \text{NF}(\mathcal{T}(C[M]))$ such that $|t| = \text{nf}_\beta(C[M])$. By Lemma 3.12(i), there exist $c(\xi) \in \mathcal{T}(C[\xi])$ and $s_1, \dots, s_k \in \mathcal{T}(M)$ such that $t \in \text{nf}(c^\bullet(s_1, \dots, s_k))$. By Corollary 4.10 we derive $\text{nf}(s_i) = 0$ and, by confluence, any reduction from $c^\bullet(s_1, \dots, s_k)$ to its normal form $\text{nf}(c^\bullet(s_1, \dots, s_k)) = t + \mathbb{T}$ factorizes as

$$c^\bullet(s_1, \dots, s_k) \twoheadrightarrow_r c^\bullet(0, \dots, 0) \twoheadrightarrow_r t + \mathbb{T}$$

Now, if a ξ actually occurs in $c(\xi)$ then we get $c^\bullet(0, \dots, 0) = 0$, but since the reduction $0 \rightarrow_r t + \mathbb{T}$ is impossible, we must have $\xi \# c(\xi)$. By Lemma 3.12(ii), we get $c^\bullet(s_1, \dots, s_k) \in \mathcal{T}(C[N])$ which implies $t \in \text{NF}(\mathcal{T}(C[N]))$ and since t is linearized we obtain $\text{nf}_\beta(C[N]) = |t|$ by Lemma 4.11(iii). \square

Our proof is certainly simpler than the original one in [Barendregt 1984], based on a topological argument, and we believe it is even simpler than those presented in [Kuper 1995; Takahashi 1994] (based on purely syntactic considerations), but we leave it to the reader to judge. Notice that we are exploiting a crucial property of the resource calculus, namely that a resource term cannot erase any subterm during the reduction. Indeed, in order to satisfy the Genericity Lemma, the context $C[\xi]$ needs to discard all occurrences of its hole ξ during its reduction to a β -normal form. This is impossible for its resource approximants $c(\xi) \in \mathcal{T}(C[\xi])$, whence $\text{nf}(c(\xi)) \neq 0$ is only possible if its hole ξ did not occur in $c(\xi)$ in the first place, namely whenever $\xi \# c(\xi)$.

We now give a “soft” proof of Corollary 3 in [Takahashi 1994]. Keeping Theorem 2.5 in mind, such a statement can be seen as a generalization of the Genericity Lemma from β -normalizable to head-normalizable λ -terms.

Proposition 5.4. *Let $M \in \Lambda$ be unsolvable and $C[\xi]$ a context. If $C[M]$ is solvable then $C[N]$ is solvable for all $N \in \Lambda$.*

PROOF. By Corollary 4.10 and Lemma 5.1, there exists $t \in \text{NF}(\mathcal{T}(C[M])) \subseteq \text{NF}(\mathcal{T}(C[N]))$, whence $\text{NF}(\mathcal{T}(C[N])) \neq \emptyset$ from which it follows that $C[N]$ is solvable. \square

We conclude this subsection by demonstrating Scott’s syntactic continuity itself, as formulated in [Barendregt 1984, Prop. 14.3.19]. The intuitive meaning of this statement is that, when looking at a context as a map $C[\xi] : \mathcal{M}_B \rightarrow \mathcal{M}_B$, a “finite portion” of its output can only be generated by a “finite portion” of its input. In this setting, a finite portion of the input (resp. output) of $C[M]$ corresponds to an approximant $Q \in \mathcal{A}(M)$ (resp. $P \in \mathcal{A}(C[M])$). In order to apply our technique based on Taylor expansions, we are going to associate a resource term $t \in \mathcal{T}(\text{BT}(M))$ with an approximant $P_t \in \mathcal{A}(M)$ by mapping the empty bag 1 to \perp and taking the supremum w.r.t. \leq_\perp . Recall from Definition 2.10(vi) that \vee denotes the supremum between approximants in \mathcal{A} .

Definition 5.5. *For $M =_\beta \lambda\vec{x}.y.M_1 \cdots M_k$ and $t \in \mathcal{T}(\text{BT}(M))$, we define by structural induction on t an approximant $P_t \in \mathcal{A}(M)$. Since $t \in \mathcal{T}(\text{BT}(M))$, it must have shape $t = \lambda\vec{x}.y.b_1 \cdots b_k$ for some $b_i \in \mathcal{M}_f(\mathcal{T}(\text{BT}(M_i)))$ with $1 \leq i \leq k$, whence we define:*

$$P_t = \lambda\vec{x}.y (\vee_{s_1 \in b_1} P_{s_1}) \cdots (\vee_{s_k \in b_k} P_{s_k})$$

where $P_s \in \mathcal{A}(M_i)$ for $s \in b_i$ and we assume that the “empty” supremum is equal to \perp , i.e. $\vee_{s \in \emptyset} s = \perp$.

The underlying fact that we are exploiting in a crucial way in this definition is that a resource term $t \in \mathcal{T}(\text{BT}(M))$ is “internally” coherent (i.e. coherent with itself) in the sense that all terms belonging to the same bag in t have a similar structure. For instance, the resource term $x[y, z]$ is not internally coherent because its bag contains terms y, z that are not coherent with each other. However terms that are not internally coherent cannot belong to the Taylor expansion of the same $\text{BT}(M)$. This can be formally demonstrated, as done in [Ehrhard and Regnier 2008], by introducing a coherence relation $s \circ t$ capturing the fact that s, t are coherent both internally and with each other, and then proving that $\mathcal{T}(\text{BT}(M))$ is a clique w.r.t. \circ .

This coherence assumption guarantees in Definition 5.5 that the resource terms belonging to the same bag b_i are associated with approximants that are compatible w.r.t. \leq_\perp , therefore their supremum is always well defined. In our formulation of this definition, this is not formally needed because the induction hypothesis already guarantees that, for every $s \in b_i$, $P_s \in \mathcal{A}(M_i)$ for the same λ -term M_i , and this set is an ideal (by Lemma 2.11(ii)).

Lemma 5.6. *For all M solvable and $t \in \mathcal{T}(\text{BT}(M))$ we have $t \in \mathcal{T}(P_t)$.*

PROOF. We proceed by structural induction on t .

By Theorem 2.5, $M =_{\beta} \lambda \vec{x}. y M_1 \cdots M_k$ whence $t = \lambda \vec{x}. y b_1 \cdots b_k$ for some $b_i \in \mathcal{M}_f(\mathcal{T}(\text{BT}(M_i)))$ with $1 \leq i \leq k$. Now, if b_i is the empty bag then there is nothing to check. Otherwise, every $s \in b_i$ is structurally smaller than t , so by induction hypothesis it belongs to $\mathcal{T}(P_s)$. By Lemma 3.14, for every $s \in b_i$ we have $\mathcal{T}(P_s) \subseteq \mathcal{T}(\bigvee_{s_i \in b_i} P_{s_i})$ so we conclude $t \in \mathcal{T}(\lambda \vec{x}. y (\bigvee_{s_1 \in b_1} P_{s_1}) \cdots (\bigvee_{s_k \in b_k} P_{s_k}))$. \square

Example 5.7.

(i) *For all affined $t \in \mathcal{T}(\text{BT}(M))$ we have $P_t = |t|$.*

(ii) *$P_{(\lambda x. x[x, x])} = \Delta$ and $P_{(\lambda x. x \perp)} = \lambda x. x \perp$.*

(iii) *$P_{(\lambda f. f[f \perp, f[f \perp]])} = \lambda f. f(f(f \perp))$.*

Everything is now in place to prove Scott's continuity. The scrupulous reader will notice that we apply to terms in Λ_{\perp} some definitions and results originally stated for Λ , e.g., $\mathcal{A}(-)$, $\sqsubseteq_{\mathcal{B}}$, Lemma 4.11. This should not be particularly troubling because the constant \perp is observationally indistinguishable from the λ -term Ω , i.e., $\mathcal{T}(\perp) = \text{NF}(\mathcal{T}(\Omega)) = \emptyset$.

Lemma 5.8 (Scott's continuity, [Barendregt 1984, Prop. 14.3.19]). *Let $M \in \Lambda$ and $C[\xi]$ be a context. For all approximants $P \in \mathcal{A}(C[M])$, there exists $Q \in \mathcal{A}(M)$ such that $P \sqsubseteq_{\mathcal{B}} C[Q]$.*

PROOF. Let $P \in \mathcal{A}(C[M])$. Since P is $\beta \perp$ -normal, we have $\mathcal{A}(P) = \{P' \in \mathcal{A} \mid P' \leq_{\perp} P\}$, whence it is a finite set $\mathcal{A}(P) = \{P_1, \dots, P_k\} \cup \{\perp\}$ where $k \geq 0$ and each $P_i \neq \perp$. As $P_i \in \mathcal{A}(P) \subseteq \mathcal{A}(C[M])$, by Lemma 4.11(ii), we have $P_i^{\circ} \in \text{NF}(\mathcal{T}(C[M]))$ so there exists $t_i \in \mathcal{T}(C[M])$ such that $P_i^{\circ} \in \text{nf}(t_i)$. By Lemma 3.12(i), there are $c_i(\xi) \in \mathcal{T}(C[\xi])$ and $s_i^1, \dots, s_i^{n_i} \in \mathcal{T}(M)$ such that $t_i = c_i^{\circ}(\vec{s}_i)$, so any reduction $c_i^{\circ}(\vec{s}_i) \rightarrow_r \text{nf}(c_i^{\circ}(\vec{s}_i))$ factorizes as follows

$$c_i^{\circ}(\vec{s}_i) \rightarrow_r c_i^{\circ}(\text{nf}(s_i^1), \dots, \text{nf}(s_i^{n_i})) \rightarrow_r \text{nf}(c_i^{\circ}(\vec{s}_i)) \ni P_i^{\circ}$$

where $0 \neq \text{nf}(s_i^m) \subseteq \text{NF}(\mathcal{T}(M)) = \mathcal{T}(\text{BT}(M))$ for $1 \leq m \leq n_i$. As there are finitely many such s_i^m and each $\text{nf}(s_i^m)$ is a finite sum, the following set

$$Q = \bigcup_{i=1}^k \bigcup_{m=1}^{n_i} \text{nf}(s_i^m)$$

is finite, so we can take $Q = \bigvee \{P_u \mid u \in Q\} \in \mathcal{A}(M)$. Now, we know that $P_i^{\circ} \in \text{nf}(c(\!|u_i^1, \dots, u_i^{n_i} \!|))$ for some $u_i^m \in Q$ ($1 \leq m \leq n_i$) and, since by Lemmas 5.6 and 3.14 every $u_i^m \in \mathcal{T}(P_{u_i^m}) \subseteq \mathcal{T}(Q)$, we derive that $P_i^{\circ} \in \text{NF}(\mathcal{T}(C[Q]))$. By Lemma 4.11(ii), we conclude that $\mathcal{A}(P) \subseteq \mathcal{A}(C[Q])$. \square

5.2 Berry's stability

The theory of stability was developed by Berry, while studying sequential computations and full abstraction for PCF [Berry 1978]. We exhibit a new proof of the Stability Theorem as formulated in [Barendregt 1984, Thm. 14.4.10]. The original demonstration exploits a *causality relation* capturing the fact that suitable subtrees of $\text{BT}(C[M_1, \dots, M_n])$ "are caused by" some argument M_i .

Definition 5.9. *Given a non-empty subset $\mathcal{X} \subseteq \Lambda$, define its \mathcal{B} -infimum as*

$$\prod \mathcal{X} = \bigcap_{M \in \mathcal{X}} \mathcal{A}(M)$$

We say that \mathcal{X} is bounded, in symbols $\mathcal{X} \uparrow$, if there exists an $N \in \Lambda$ satisfying, for all $M \in \mathcal{X}$, $M \sqsubseteq_{\mathcal{B}} N$.

Example 5.10. (i) *For all $M \in \Lambda$, we have $\prod \{M\} = \mathcal{A}(M)$.*

(ii) *$\prod \{1, \Omega\} = \{\perp\}$ and $\prod \{\lambda x. x \Omega, \Delta\} = \{\perp, \lambda x. x \perp\}$.*

(iii) *$\prod \{\lambda f. f^n(\Omega) \mid n \geq 2\} = \{\perp, \lambda f. f \perp, \lambda f. f(f \perp)\}$.*

A subset $\mathcal{A} \subseteq \mathcal{A}$ is equal to $\mathcal{A}(M)$ for some $M \in \Lambda$ if and only if it is directed w.r.t. \leq_{\perp} , it contains a finite number of free variables and is r.e. (after coding), by [Barendregt 1984, Thm. 10.1.23]. The last condition follows since λ -calculus is not only a formal system, but a Turing-complete programming language as well. Now, for infinite bounded subsets $\mathcal{X} \subseteq \Lambda$, it is possible that $\prod \mathcal{X}$ is not r.e., so it does not correspond to the set of approximants of any M . The \mathcal{B} -infimum $\prod \mathcal{X}$ is however always directed, whence its supremum gives a so-called ‘‘Böhm-like’’ tree.

Theorem 5.11 (Stability). *Let $C[\xi_1, \dots, \xi_n]$ be a n -context and $\mathcal{I} = \{1, \dots, n\}$. For all $i \in \mathcal{I}$, take non-empty subsets $\mathcal{X}_i \subseteq \Lambda$ and $M_i \in \Lambda$. Assume, for all $i \in \mathcal{I}$, that $\mathcal{X}_i \uparrow$ and $\mathcal{A}(M_i) = \prod \mathcal{X}_i$ then*

$$\mathcal{A}(C[M_1, \dots, M_n]) = \prod \{C[N_1, \dots, N_n] \mid \forall i \in \mathcal{I}. N_i \in \mathcal{X}_i\}$$

PROOF. By Corollary 4.12, $\mathcal{X}_i \uparrow$ means that there exists $L_i \in \Lambda$ such that $\bigcup_{N_i \in \mathcal{X}_i} \text{NF}(\mathcal{T}(N_i)) \subseteq \text{NF}(\mathcal{T}(L_i))$, we can assume $\text{NF}(\mathcal{T}(M_i)) = \bigcap_{N_i \in \mathcal{X}_i} \text{NF}(\mathcal{T}(N_i))$ and it is enough to show:

$$\text{NF}(\mathcal{T}(C[M_1, \dots, M_n])) = \bigcap \{\text{NF}(\mathcal{T}(C[N_1, \dots, N_n])) \mid \forall i \in \mathcal{I}. N_i \in \mathcal{X}_i\}$$

(\subseteq) Clearly, for all $i \in \mathcal{I}$ and $N_i \in \mathcal{X}_i$, we have $\text{NF}(\mathcal{T}(M_i)) \subseteq \text{NF}(\mathcal{T}(N_i))$, therefore we conclude $\text{NF}(\mathcal{T}(C[M_1, \dots, M_n])) \subseteq \text{NF}(\mathcal{T}(C[N_1, \dots, N_n]))$ by Lemma 5.1.

(\supseteq) Let $t \in \bigcap_{\vec{N} \in \vec{\mathcal{X}}} \text{NF}(\mathcal{T}(C[N_1, \dots, N_n]))$ where $\vec{N} \in \vec{\mathcal{X}}$ stands for $N_1 \in \mathcal{X}_1, \dots, N_n \in \mathcal{X}_n$. For every $\vec{N} \in \vec{\mathcal{X}}$, by Lemma 3.12(i) there exist $c_{\vec{N}}(\xi_1, \dots, \xi_n) \in \mathcal{T}(C[\xi_1, \dots, \xi_n])$ and, for every i , $\vec{s}_{\vec{N}}^i = s_{\vec{N}}^{i1}, \dots, s_{\vec{N}}^{ik_i} \in \mathcal{T}(N_i)$ such that (using confluence to factorize the reduction):

$$\begin{aligned} c_{\vec{N}}^{\bullet}(\vec{s}_{\vec{N}}^1, \dots, \vec{s}_{\vec{N}}^n) &\rightarrow_r c_{\vec{N}}^{\bullet}(\text{nf}(s_{\vec{N}}^{11}), \dots, \text{nf}(s_{\vec{N}}^{1k_1}), \dots, \text{nf}(s_{\vec{N}}^{n1}), \dots, \text{nf}(s_{\vec{N}}^{nk_n})) \\ &\rightarrow_r \text{nf}(c_{\vec{N}}^{\bullet}(\vec{s}_{\vec{N}}^1, \dots, \vec{s}_{\vec{N}}^n)) \ni t \end{aligned}$$

So, for all $i \in \mathcal{I}$ and j ($1 \leq j \leq k_i$), there exist $v_{\vec{N}}^{ij} \in \text{nf}(s_{\vec{N}}^{ij}) \subseteq \text{NF}(\mathcal{T}(N_i)) \subseteq \text{NF}(\mathcal{T}(L_i))$ such that $\text{nf}(c_{\vec{N}}^{\bullet}(\vec{v}_{\vec{N}}^1, \dots, \vec{v}_{\vec{N}}^n)) = t + \mathbb{T}_{\vec{N}}$ for some $\mathbb{T}_{\vec{N}}$. From the last inclusion, we obtain a resource term $u_{\vec{N}}^{ij} \in \mathcal{T}(L_i)$ such that $\text{nf}(u_{\vec{N}}^{ij}) = v_{\vec{N}}^{ij} + \mathbb{T}_{\vec{N}}^{ij}$ for some $\mathbb{T}_{\vec{N}}^{ij}$. By composing the reductions:

$$c_{\vec{N}}^{\bullet}(\vec{v}_{\vec{N}}^1, \dots, \vec{v}_{\vec{N}}^n) \rightarrow_r t + \mathbb{T}_{\vec{N}} \quad \text{and} \quad u_{\vec{N}}^{ij} \rightarrow_r v_{\vec{N}}^{ij} + \mathbb{T}_{\vec{N}}^{ij} \quad (\forall i \in \mathcal{I}, 1 \leq j \leq k_i)$$

we obtain $t \in \text{nf}(c_{\vec{N}}^{\bullet}(\vec{u}_{\vec{N}}^1, \dots, \vec{u}_{\vec{N}}^n))$. This holds for all $\vec{N} \in \vec{\mathcal{X}}$, thus $t \in \bigcap_{\vec{N} \in \vec{\mathcal{X}}} \text{nf}(c_{\vec{N}}^{\bullet}(\vec{u}_{\vec{N}}^1, \dots, \vec{u}_{\vec{N}}^n))$. Since $c_{\vec{N}}^{\bullet}(\vec{u}_{\vec{N}}^1, \dots, \vec{u}_{\vec{N}}^n) \in \mathcal{T}(C[L_1, \dots, L_n])$ and the λ -terms L_i 's are independent from N_1, \dots, N_n we can apply Lemma 4.4, so we obtain that the set $\{c_{\vec{N}}^{\bullet}(\vec{u}_{\vec{N}}^1, \dots, \vec{u}_{\vec{N}}^n) \mid \vec{N} \in \vec{\mathcal{X}}\}$ is actually a singleton. Moreover, its only element is of the form $c^{\bullet}(\vec{u}^1, \dots, \vec{u}^n)$ for some $c(\xi_1, \dots, \xi_n) \in \mathcal{T}(C[\xi_1, \dots, \xi_n])$ and $u^{ij} \in \mathcal{T}(L_i)$. Recalling that $v_{\vec{N}}^{ij} \in \text{nf}(u^{ij}) \subseteq \text{NF}(\mathcal{T}(L_i))$, we obtain by Corollary 4.6 that there exists L'_i such that $L_i \rightarrow_{\beta} L'_i$ and $v_{\vec{N}}^{ij} \in \mathcal{T}(L'_i)$ for all $1 \leq j \leq k_i$. That is, for every $\vec{N} \in \vec{\mathcal{X}}$, we derive:

$$c^{\bullet}(\vec{v}_{\vec{N}}^1, \dots, \vec{v}_{\vec{N}}^n) \in \mathcal{T}(C[L'_1, \dots, L'_n]). \quad (1)$$

Now, we already know that $t \in \text{nf}(c_{\vec{N}}^{\bullet}(\vec{v}_{\vec{N}}^1, \dots, \vec{v}_{\vec{N}}^n))$, from which it follows:

$$t \in \bigcap_{\vec{N} \in \vec{\mathcal{X}}} \text{nf}(c^{\bullet}(\vec{v}_{\vec{N}}^1, \dots, \vec{v}_{\vec{N}}^n)). \quad (2)$$

By (1), (2) and the fact that the L'_i 's are independent from N_1, \dots, N_n we can apply Lemma 4.4, thus we obtain that $\{c^{\bullet}(\vec{v}_{\vec{N}}^1, \dots, \vec{v}_{\vec{N}}^n) \mid \vec{N} \in \vec{\mathcal{X}}\} = \{c^{\bullet}(\vec{v}^1, \dots, \vec{v}^n)\}$ with $t \in \text{nf}(c^{\bullet}(\vec{v}^1, \dots, \vec{v}^n))$. Since $v^{ij} \in \text{NF}(\mathcal{T}(N_i))$ for all $i \in \mathcal{I}$, this means $v^{i1}, \dots, v^{ik_i} \in \bigcap_{i \in \mathcal{I}} \text{NF}(\mathcal{T}(N_i)) = \text{NF}(\mathcal{T}(M_i))$.

We conclude that $t \in \text{nf}(c^{\bullet}(\vec{v}_1, \dots, \vec{v}_n)) \subseteq \text{NF}(\mathcal{T}(C[M_1, \dots, M_n]))$. \square

6 THE PERPENDICULAR LINES LEMMA

We present the Perpendicular Lines Lemma (PLL, for short), namely Theorem 14.4.12 in [Barendregt 1984], and show that it can be proved in a particularly elegant way, by applying the techniques previously introduced. In our personal opinion this constitutes the most outstanding example in the present paper, both for the simplicity of the proof and for the importance of the result. For example, as a consequence, one immediately derives the fact that the “parallel or” is not definable in λ -calculus. Before going further, we discuss the intuitive meaning of the PLL, its validity in different term models (Definition 2.17), and the techniques employed in the proofs from the literature.

Let \mathbb{R} be the set of reals. In the Euclidean plane \mathbb{R}^2 , the lines $(1, y)_{y \in \mathbb{R}} := \{(1, y) \mid y \in \mathbb{R}\}$ and $(x, 2)_{x \in \mathbb{R}}$ are perpendicular and intersect in the point $(1, 2)$. Analogously, in the space \mathbb{R}^3 , the lines $(x, 1, 2)_{x \in \mathbb{R}}$, $(3, y, 4)_{y \in \mathbb{R}}$ and $(5, 6, z)_{z \in \mathbb{R}}$ are perpendicular even if they do not intersect. Translated in terms of λ -calculus, one could say that $\{(X, K) \mid X \in \Lambda\}$ and $\{(F, Y) \mid Y \in \Lambda\}$ are “perpendicular”, and similarly in higher dimensions. Intuitively, PLL states that if an n -context $C[\xi_1, \dots, \xi_n]$, seen as a function from Λ^n to Λ , is constant on n perpendicular lines, then it must be constant everywhere.

We now exhibit and discuss the statement, the proof will follow.

Theorem 6.1 (Perpendicular Lines Lemma). *Let $n \geq 0$, $I = \{1, \dots, n\}$, $C[\xi_1, \dots, \xi_n]$ be a n -context, $(M_{ij})_{(i,j) \in I \times I}$ and $(N_i)_{i \in I}$ be sequences of λ -terms. Assume that*

$$\forall Z \in \Lambda \left\{ \begin{array}{ll} C[Z, M_{12}, \dots, M_{1n}] & =_{\mathcal{B}} N_1 \\ C[M_{21}, Z, \dots, M_{2n}] & =_{\mathcal{B}} N_2 \\ & \vdots \\ C[M_{n1}, \dots, M_{n(n-1)}, Z] & =_{\mathcal{B}} N_n \end{array} \right.$$

then $\forall Z_1, \dots, Z_n \in \Lambda, C[Z_1, \dots, Z_n] =_{\mathcal{B}} N_1 =_{\mathcal{B}} \dots =_{\mathcal{B}} N_n$.

As in the original lemma we consider Böhm-tree equality, but the statement can be rewritten using $=_{\mathcal{T}}$ for any λ -theory \mathcal{T} and we can meaningfully wonder whether PLL is valid in the term model $\mathcal{M}_{\mathcal{T}}$, written $\mathcal{M}_{\mathcal{T}} \models \text{PLL}$. A similar question can be raised for the closed term model $\mathcal{M}_{\mathcal{T}}^o$ (intuitively, replace “ $\forall Z \in \Lambda$ ” by “ $\forall Z \in \Lambda^o$ ” in the statement). The proof given in [Barendregt 1984] exploits Kahn and Plotkin’s sequentiality theory (developed in [Kahn and Plotkin 1978] in the context of concrete domains), which is stronger than Berry’s stability, and shows that $\mathcal{M}_{\mathcal{B}} \models \text{PLL}$. As we are considering $=_{\mathcal{B}}$, a context $C[\xi]$ can display a constant behaviour for several reasons:

- (i) $C[\xi]$ does not contain the hole ξ at all (the trivial case);
- (ii) ξ is “erased” during its reduction as in $C[\xi] = (\lambda xy.y)\xi$;
- (iii) ξ is “hidden” behind an unsolvable as in $C[\xi] = \Omega \xi$;
- (iv) ξ is pushed into infinity like the variable z in $\text{BT}(Pz)$ from Example 2.8, namely $C[\xi] = P\xi$.

The fact that $\mathcal{M}_{\lambda} \models \text{PLL}$ was only suggested in [Barendregt 1984], and proved later on by Endrullis and de Vrijer [Endrullis and de Vrijer 2008] that applied van Daalen’s *Reduction under Substitution* property [van Daalen 1980], which is a strengthening of the famous “Barendregt Lemma”. As we are now considering $=_{\beta}$, a constant context $C[\xi]$ can only display two possible behaviours, namely (i) or (ii) in the list above. In particular, $C[\xi]$ cannot push its hole ξ into infinity. Moreover, since we are working in the open term model, the hypotheses of PLL must hold when taking $Z = x$ and $Z = y$ for $x \neq y$. This is a strong assumption because x and y are completely defined different values thus enforcing a “maximal” distinction among the terms. These are the main ingredients used in [Endrullis and de Vrijer 2008] to derive $\mathcal{M}_{\lambda} \models \text{PLL}$ and they are crucial – indeed, as shown in [Statman and Barendregt 1999], the property fails in the closed term model $\mathcal{M}_{\lambda}^o \not\models \text{PLL}$. The last result relies on the existence of so-called Plotkin terms [Plotkin 1974], namely β -distinct terms $M, N \in \Lambda^o$ satisfying $ML =_{\beta} NL$ for all $L \in \Lambda^o$.

We now show our proof of Theorem 6.1, where the use of Taylor expansion guarantees properties stronger than those described for λ , despite the fact that we consider equality in \mathcal{B} . Indeed, while a constant context $C[\xi]$ may exhibit all the aforementioned behaviours (i)-(iv), a resource context $c(\xi)$ belonging to its Taylor expansion $\mathcal{T}(C[\xi])$ must be constant for trivial reasons:

- (i) either because $c(\xi) \rightarrow_r 0$,
- (ii) or because its hole ξ does not occur in $c(\xi)$, i.e. $\xi \# c(\xi)$.

In fact, since the resource calculus is linear and SN, $c(\xi)$ cannot erase any occurrence of ξ nor push ξ into infinity. These are the intuitions behind the claim formulated below.

PROOF OF THEOREM 6.1. Assume without loss of generality that $n = 3$ (the same assumption is considered in [Barendregt 1984, Proof of Thm. 14.4.12]). By Corollary 4.12, the system of equations becomes:

$$\forall Z \in \Lambda \begin{cases} C[Z, M_{12}, M_{13}] & =_{\tau} & N_1 \\ C[M_{21}, Z, M_{23}] & =_{\tau} & N_2 \\ C[M_{31}, M_{32}, Z] & =_{\tau} & N_3 \end{cases} \quad (3)$$

We show that under these hypotheses the following holds.

Claim. For all $c(\xi_1, \xi_2, \xi_3) \in \mathcal{T}(C[\xi_1, \xi_2, \xi_3])$, if $\text{nf}(c(\xi_1, \xi_2, \xi_3)) \neq 0$ then $\xi_1, \xi_2, \xi_3 \# c(\xi_1, \xi_2, \xi_3)$.

We proceed by induction on the size of $c(\xi_1, \xi_2, \xi_3)$. If $t \in \text{nf}(c(\xi_1, \xi_2, \xi_3))$ then t has shape $t = \lambda \vec{x}. \chi b_1 \cdots b_k$ for some $\chi \in \mathbb{V} \cup \{\xi_1, \xi_2, \xi_3\}$. By Corollary 4.6, there exists a reduction $C[\xi_1, \xi_2, \xi_3] \rightarrow_{\beta} C'[\xi_1, \xi_2, \xi_3]$ such that $\text{nf}(c(\xi_1, \xi_2, \xi_3)) \subseteq \mathcal{T}(C'[\xi_1, \xi_2, \xi_3])$, whence

$$C'[\xi_1, \xi_2, \xi_3] = \lambda \vec{x}. \chi C_1[\xi_1, \xi_2, \xi_3] \cdots C_k[\xi_1, \xi_2, \xi_3]$$

with every $b_j \in \mathcal{M}_f(\mathcal{T}(C_j[\xi_1, \xi_2, \xi_3]))$. We split into subcases, depending on the shape of χ .

- $\chi = \xi_i$ for some $i \leq 3$, say, $i = 2$. In this case we have $\lambda \vec{x}. \xi_2 1 \cdots 1 \in \mathcal{T}(C'[\xi_1, \xi_2, \xi_3])$ therefore, by Lemma 4.5(i) there exists $c_0(\xi_1, \xi_2, \xi_3) \in \mathcal{T}(C[\xi_1, \xi_2, \xi_3])$ such that $\lambda \vec{x}. \xi_2 1 \cdots 1 \in \text{nf}(c_0(\xi_1, \xi_2, \xi_3))$, whence $\xi_1, \xi_3 \# c_0(\xi_1, \xi_2, \xi_3)$ since no hole can be erased along reductions. By Lemma 3.12(ii) and instantiating the 2nd equation of the system with $Z = z \notin \text{FV}(N_2)$ we have, for all $\vec{s}_1, \vec{s}_3 \in \Lambda'$, $\lambda \vec{x}. z 1 \cdots 1 \in \text{nf}(c_0^*(\vec{s}_1, \vec{z}, \vec{s}_3)) \subseteq \text{NF}(\mathcal{T}(C[M_{21}, z, M_{23}])) \subseteq \text{NF}(\mathcal{T}(N_2))$. Thus we obtain $z \in \text{FV}(\text{NF}(\mathcal{T}(N_2))) = \text{FV}(\mathcal{T}(\text{BT}(N_2))) = \text{FV}(\text{BT}(N_2)) \subseteq \text{FV}(N_2)$, from which we derive a contradiction.
- $\chi = y$. For all $i \leq 3$, by applying the i -th equation, we derive that $N_i \rightarrow_h \lambda \vec{x}. y N_{i1} \cdots N_{ik}$ with $N_{ij} =_{\tau} C_j[M_{i1}, \dots, M_{i(i-1)}, Z, M_{i(i+1)}, \dots, M_{i3}]$ for all $Z \in \Lambda$ and $j = 1, \dots, k$. In other words, for every such j we found a context $C_j[\xi_1, \xi_2, \xi_3]$ and λ -terms N_{1j}, N_{2j}, N_{3j} satisfying:

$$\forall Z \in \Lambda \begin{cases} C_j[Z, M_{12}, M_{13}] & =_{\tau} & N_{1j} \\ C_j[M_{21}, Z, M_{23}] & =_{\tau} & N_{2j} \\ C_j[M_{31}, M_{32}, Z] & =_{\tau} & N_{3j} \end{cases}$$

Since every $c_j(\xi_1, \xi_2, \xi_3) \in b_j$ is normal $\text{nf}(c_j(\xi_1, \xi_2, \xi_3)) \neq 0$. As it belongs to $\mathcal{T}(C_j[\xi_1, \xi_2, \xi_3])$ and has size strictly smaller than $c(\xi_1, \xi_2, \xi_3)$, by induction hypothesis $\xi_1, \xi_2, \xi_3 \# c_j(\xi_1, \xi_2, \xi_3)$, from which it follows $\xi_1, \xi_2, \xi_3 \# \lambda \vec{x}. y b_1 \cdots b_k$ and, ultimately, $\xi_1, \xi_2, \xi_3 \# c(\xi_1, \xi_2, \xi_3)$.

This completes the proof of the claim, we now prove the main statement.

Consider fixed $Z_1, Z_2, Z_3 \in \Lambda$, we prove $\text{NF}(\mathcal{T}(C[Z_1, Z_2, Z_3])) = \text{NF}(\mathcal{T}(N_i))$ for all $i (1 \leq i \leq 3)$ by showing the two inclusions.

- (\subseteq) Let $t \in \text{NF}(\mathcal{T}(C[Z_1, Z_2, Z_3]))$. By definition, there is $t' \in \mathcal{T}(C[Z_1, Z_2, Z_3])$ such that $t \in \text{nf}(t')$. By Lemma 3.12(i), $t' = c^*(\vec{s}_1, \vec{s}_2, \vec{s}_3)$ for some $c(\xi_1, \xi_2, \xi_3) \in \mathcal{T}(C[\xi_1, \xi_2, \xi_3])$ and $\vec{s}_i \in \mathcal{T}(Z_i)$.

Since $c^\bullet(\vec{s}_1, \vec{s}_2, \vec{s}_3) \neq 0$ entails $c(\xi_1, \xi_2, \xi_3) \neq 0$, the claim ensures that none of the holes ξ_i may actually occur in $c(\xi_1, \xi_2, \xi_3)$. Thus, by applying Lemma 3.12(ii), we derive

$$t' = c^\bullet(\vec{s}_1, \vec{s}_2, \vec{s}_3) \in \mathcal{T}(C[M_{i_1}, \dots, M_{i_{i-1}}, Z, M_{i_{i+1}}, \dots, M_{i_3}])$$

From the i -th equation, we get $\text{NF}(\mathcal{T}(C[M_{i_1}, \dots, M_{i_{i-1}}, Z, M_{i_{i+1}}, \dots, M_{i_3}])) \subseteq \text{NF}(\mathcal{T}(N_i))$, so we conclude $t \in \text{NF}(\mathcal{T}(N_i))$.

- (\supseteq) For the converse we consider the case $i = 1$, the others being analogous. If $t \in \text{NF}(\mathcal{T}(N_1)) \subseteq \text{NF}(\mathcal{T}(C[Z_1, M_{12}, M_{13}]))$ then there are $c(\xi_1, \xi_2, \xi_3) \in \mathcal{T}(C[\xi_1, \xi_2, \xi_3])$, $\vec{s} \in \mathcal{T}(Z_1)$, $\vec{u} \in \mathcal{T}(M_{12})$ and $\vec{v} \in \mathcal{T}(M_{13})$ such that $t \in \text{nf}(c^\bullet(\vec{s}, \vec{u}, \vec{v}))$ which entails $\text{nf}(c(\xi_1, \xi_2, \xi_3)) \neq 0$. Now, the claim ensures that $\xi_1, \xi_2, \xi_3 \# c(\xi_1, \xi_2, \xi_3)$ so, by Lemma 3.12(ii), we get $c^\bullet(\vec{s}, \vec{u}, \vec{v}) \in \mathcal{T}(C[Z_1, Z_2, Z_3])$ from which it follows $t \in \text{NF}(\mathcal{T}(C[Z_1, Z_2, Z_3]))$.

As $=_\tau$ and $=_{\mathcal{B}}$ coincide by Corollary 4.12, we conclude that $C[Z_1, Z_2, Z_3] =_{\mathcal{B}} N_i$ for all i . \square

As promised, the undefinability of the “parallel or”, whose proof is taken from [Barendregt 1984, pag. 380], follows immediately.

Corollary 6.2. *The “parallel or” is not λ -definable: there is no λ -term F such that, for all $M, N \in \Lambda$:*

$$FMN =_{\beta} \begin{cases} 1 & \text{if } M \text{ or } N \text{ are solvable,} \\ U & \text{for some unsolvable } U \in \Lambda, \text{ otherwise.} \end{cases}$$

PROOF. Otherwise we would have, for all $Z \in \Lambda$, $F \mid Z =_{\mathcal{B}} FZ \mid =_{\mathcal{B}} 1$, while $F \Omega \Omega =_{\mathcal{B}} \Omega$, thus contradicting PLL. \square

The Proof of Theorem 6.1 can be easily modified to encompass the closed statement as well.

Theorem 6.3. $M_{\mathcal{B}}^0 \models PLL$

PROOF. The reader can check that the only point where we exploited open terms, is the case $\chi = \xi_2$ in the proof of the claim, where we take $Z = z \notin \text{FV}(N_2)$. It is therefore enough to rewrite that part as follows: “By instantiating Z in the 2nd equation of (3) respectively with $Z_1 = \lambda z_1 \dots z_k. K$ and $Z_2 = \lambda z_1 \dots z_k. F$ we would get $C[M_{21}, Z_1, M_{23}] \rightarrow_{\beta} \lambda \vec{x}. Z_1(C_1[M_{21}, Z_1, M_{23}]) \dots (C_k[M_{21}, Z_1, M_{23}]) \rightarrow_{\beta} \lambda \vec{x}. K =_{\mathcal{B}} N_2$ and $C[M_{21}, Z_2, M_{23}] \rightarrow_{\beta} \lambda \vec{x}. F =_{\mathcal{B}} N_2$, which is impossible since $\lambda \vec{x}. K \neq_{\mathcal{B}} \lambda \vec{x}. F$.”

All other cases remain unchanged. \square

To the best of our knowledge, the result above is original.

7 COMPARISON WITH LABELLED λ -CALCULUS

We have seen that approximating the behaviour of a λ -term by calculating its Taylor expansion allows to abandon proof-techniques based on coinduction and reducibility candidates in favour of the inductive principle. We do not claim that this is the first successful attempt: already in [Wadsworth 1976], Wadsworth defined a *labelled λ -calculus* to calculate an approximant $P \in \mathcal{A}(M)$ in an effective way. The idea is to annotate every subterm of M with a certain *amount of energy*, represented by a natural number n , which is then decremented along the reduction:

$$\begin{aligned} (\lambda x. M)^{n+1} N &\rightarrow_{\beta_\ell} (M\{N^n/x\})^n \\ (\lambda x. M)^0 N &\rightarrow_{\beta_\perp} (M\{\perp/x\})^0 \end{aligned}$$

When two labels n, m are in each other proximity, they are unified by taking the minimum as in $(M^n)^m \rightarrow_\ell M^{\min\{m, n\}}$. The labelled λ -calculus Λ^ℓ so obtained clearly enjoys strongly normalization, and can be proved to be confluent as well. One retrieves an approximant $P \in \mathcal{A}(M)$ starting from a fully annotated version M^ℓ of M by first computing $\text{nf}_{\beta_\ell \beta_\perp \perp \ell}(M^\ell)$ and then removing the labels. By exploiting the properties above, it is possible to give nice proofs of Church-Rosser

and standardization for λ -calculus [Barendregt 1984, Ch. 14]. From a semantic perspective, the labelled λ -calculus can be interpreted in well-stratified reflexive objects \mathcal{D} living in cpo-enriched categories [Manzonetto 2009]. It was first used by Hyland [Hyland 1975] to prove that Scott's \mathcal{D}_∞ and Plotkin's \mathcal{P}_ω induce a λ -theory including \mathcal{B} .

Comparing the labelled λ -calculus and the resource calculus we can find some common principles: in both cases the idea is to harness infinite reductions by bounding the availability of subprograms. However, this idea is applied in different (somewhat dual) ways: in Λ^ℓ the bound is on the contraction of redexes, so it is the λ -abstraction that exhibits a restricted behaviour, while in Λ^r the restriction is on the amount of resources a program has available. Another difference is that the labels of Λ^ℓ give an *upper bound* on the energy that can be consumed by a λ -term M , while a resource term $t \in \mathcal{T}(M)$ that does not reduce to 0 must contain *the exact number* of resources needed by M in order to compute its outcome. Moreover, while both Λ^ℓ and Λ^r are confluent and strongly normalizing, only the latter enjoys *linearity* that prevents a resource term from erasing or duplicating its subterms during its execution – a property that is crucial in our proof of PLL.

In conclusion, we believe that the labelled λ -calculus is a valid instrument, but rather *ad hoc*, since it is driven by purely syntactic considerations. The resource calculus is more natural because it arises from semantic considerations, namely Ehrhard's observation that in Köthe spaces a derivative operator is actually at hand [Ehrhard 2002]. Similarly, the Taylor expansion arises from the analysis of Girard's translation $(\cdot)^\bullet$ of λ -calculus into Linear Logic proof-nets sending $(MN)^\bullet$ to $M^\bullet(N^\bullet)^\bullet$. In fact, the advantage of the resource calculus is that it has a nowadays well-understood logical background having its roots in Differential Linear Logic [Ehrhard and Regnier 2006b] and this is the reason why it has been generalized so easily to non-deterministic, probabilistic, algebraic call-by-name and call-by-value calculi. Also from the semantic perspective, Λ^r can be naturally interpreted in every (linear) reflexive object living in a (Cartesian closed) differential category [Blute et al. 2009], and the Taylor expansion can be used to obtain combinatorial proofs of (denotational) Approximation Theorems [Manzonetto and Ruoppolo 2014].

8 CONCLUSIONS

We have shown that fundamental theorems in λ -calculus whose classic demonstrations require advanced topological or coinductive techniques can be endowed with simpler proofs based on Ehrhard and Regnier's Commutation Theorem between Böhm trees and Taylor Expansions normalization. We believe that such a technique is rather versatile and analogous results could be obtained in other paradigms (e.g. call-by-value and call-by-need) as well as in the settings of algebraic, non-deterministic, probabilistic and quantum λ -calculi. Besides exploring the consequences of the Commutation Theorem for these languages, the real breakthrough would be the design of a unifying framework allowing to transfer both syntactic and semantic properties from a source language to many different target languages. We believe that an excellent starting point to achieve this goal is represented by Ehrhard and Guerrieri's *bang calculus* [Ehrhard and Guerrieri 2016], a λ -calculus with explicit promotion that has arisen from a deep analysis of differential linear logic. From an operational point of view, both the call-by-name and the call-by-value λ -calculi can be faithfully interpreted in the bang calculus via translation – indeed, it turns out that the latter can be seen as an untyped version of Levy's call-by-push value calculus [Levy 2006]. As shown in [Guerrieri and Manzonetto 2018], these translations are 'faithful' in a technical sense – one reduction step in the source language corresponds to exactly one step in the target and *viceversa*. (As noticed by Faggian, this entails that Standardization Theorems for both paradigms can be transferred from the analogous result enjoyed by the bang calculus.) Of course, in order to become a candidate general enough to embed all languages mentioned above, the bang calculus should be

endowed with coefficients from a semiring as done in [Laird et al. 2013] and non-determinism in the spirit of [Goubault-Larrecq 2019].

From the point of view of denotational semantics, the most advanced attempt to provide a unifying framework encompassing algebraic, non-deterministic, probabilistic and quantum calculi is given in [Tsukada et al. 2018], where the authors develop a profunctorial semantics generalizing the relational model of Linear Logic in the call-by-name setting. As noticed in [Ehrhard and Guerrieri 2016], every categorical model of Linear Logic well-suited to interpret the bang calculus is also suitable to interpret both paradigms – it would be therefore interesting to develop a profunctorial semantics of the bang calculus and provide a parametric Adequacy Theorem that might be instantiated to all aforementioned languages and paradigms.

ACKNOWLEDGMENTS

We thank Henk Barendregt for introducing us to the perpendicular lines lemma, and raising the question whether it holds for the closed term model of \mathcal{B} . We are grateful to the anonymous reviewers for their suggestions that allowed us to improve our presentation considerably.

REFERENCES

- Beniamino Accattoli. 2018. (In)Efficiency and Reasonable Cost Models. *Electr. Notes Theor. Comput. Sci.* 338 (2018), 23–43.
- Beniamino Accattoli, Stéphane Graham-Lengrand, and Delia Kesner. 2018. Tight typings and split bounds. *PACMPL* 2, ICFP (2018), 94:1–94:30. <https://doi.org/10.1145/3236789>
- Davide Barbarossa and Giulio Manzonetto. 2019. About the power of Taylor expansion. 3rd International Workshop on Trends in Linear Logic and Applications.
- Henk P. Barendregt. 1977. The type free lambda calculus. In *Handbook of Mathematical Logic*, J. Barwise (Ed.). Studies in Logic and the Foundations of Mathematics, Vol. 90. North-Holland, Amsterdam, 1091–1132.
- Henk P. Barendregt. 1984. *The lambda-calculus, its syntax and semantics* (revised ed.). Number 103 in Studies in Logic and the Foundations of Mathematics. North-Holland.
- Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. 2017. Automatic Differentiation in Machine Learning: a Survey. *Journal of Machine Learning Research* 18 (2017), 153:1–153:43. <http://jmlr.org/papers/v18/468.html>
- G rard Berry. 1978. Stable Models of Typed lambda-Calculi. In *ICALP (Lecture Notes in Computer Science)*, Vol. 62. Springer, 72–89.
- Richard Blute, J. Robin B. Cockett, and Robert A. G. Seely. 2006. Differential categories. *Mathematical Structures in Computer Science* 16, 6 (2006), 1049–1083.
- Richard Blute, J. Robin B. Cockett, and Robert A. G. Seely. 2009. Cartesian differential categories. *Theory and Applications of Categories* 22, 23 (2009), 622–672.
- G rard 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)*, Eike Best (Ed.), Vol. 715. Springer, 1–6. https://doi.org/10.1007/3-540-57208-2_1
- Antonio Bucciarelli, Thomas Ehrhard, and Giulio Manzonetto. 2012. A relational semantics for parallelism and non-determinism in a functional setting. *Ann. Pure Appl. Logic* 163, 7 (2012), 918–934. <https://doi.org/10.1016/j.apal.2011.09.008>
- J. Robin B. Cockett and Geoff S. H. Cruttwell. 2014. Differential Structure, Tangent Structure, and SDG. *Applied Categorical Structures* 22, 2 (2014), 331–417. <https://doi.org/10.1007/s10485-013-9312-0>
- J. Robin B. Cockett and Jean-Simon Lemay. 2019. Integral categories and calculus categories. *Mathematical Structures in Computer Science* 29, 2 (2019), 243–308. <https://doi.org/10.1017/S0960129518000014>
- Daniel de Carvalho. 2018. Execution time of λ -terms via denotational semantics and intersection types. *Mathematical Structures in Computer Science* 28, 7 (2018), 1169–1203. <https://doi.org/10.1017/S0960129516000396>
- Thomas Ehrhard. 2002. On K the Sequence Spaces and Linear Logic. *Mathematical Structures in Computer Science* 12, 5 (2002), 579–623. <https://doi.org/10.1017/S0960129502003729>
- Thomas Ehrhard. 2012. Collapsing non-idempotent intersection types. In *Computer Science Logic (CSL'12) - 26th International Workshop/21st Annual Conference of the EACSL, CSL 2012, September 3-6, 2012, Fontainebleau, France (LIPIcs)*, Patrick C geleski and Arnaud Durand (Eds.), Vol. 16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 259–273. <https://doi.org/10.4230/LIPIcs.CSL.2012.259>
- Thomas Ehrhard. 2018. An introduction to differential linear logic: proof-nets, models and antiderivatives. *Mathematical Structures in Computer Science* 28, 7 (2018), 995–1060.

- Thomas Ehrhard and Giulio Guerrieri. 2016. The Bang Calculus: an untyped lambda-calculus generalizing call-by-name and call-by-value. In *Proceedings of the 18th International Symposium on Principles and Practice of Declarative Programming, Edinburgh, United Kingdom, September 5-7, 2016*, James Cheney and Germán Vidal (Eds.). ACM, 174–187. <https://doi.org/10.1145/2967973.2968608>
- Thomas Ehrhard and Laurent Regnier. 2003. The differential lambda-calculus. *Theor. Comput. Sci.* 309, 1-3 (2003), 1–41.
- Thomas Ehrhard and Laurent Regnier. 2006a. Böhm Trees, Krivine’s Machine and the Taylor Expansion of Lambda-Terms. In *CiE (Lecture Notes in Computer Science)*, Vol. 3988. Springer, 186–197.
- Thomas Ehrhard and Laurent Regnier. 2006b. Differential interaction nets. *Theor. Comput. Sci.* 364, 2 (2006), 166–195.
- Thomas Ehrhard and Laurent Regnier. 2008. Uniformity and the Taylor expansion of ordinary lambda-terms. *Theor. Comput. Sci.* 403, 2-3 (2008), 347–372.
- Thomas Ehrhard and Christine Tasson. 2019. Probabilistic call by push value. *Logical Methods in Computer Science* 15, 1 (2019).
- Jörg Endrullis and Roel C. de Vrijer. 2008. Reduction Under Substitution. In *Rewriting Techniques and Applications, 19th International Conference, RTA 2008 (Lecture Notes in Computer Science)*, Andrei Voronkov (Ed.), Vol. 5117. Springer, 425–440.
- Sabrina Fiege, Andrea Walther, Kshitij Kulshreshtha, and Andreas Griewank. 2018. Algorithmic differentiation for piecewise smooth functions: a case study for robust optimization. *Optimization Methods and Software* 33, 4-6 (2018), 1073–1088. <https://doi.org/10.1080/10556788.2017.1333613>
- Marcelo P. Fiore, Nicola Gambino, J. Martin E. Hyland, and Glynn Winskel. 2007. The cartesian closed bicategory of generalised species of structures. *J. London Maths. Soc* 77 (2007), 203–220.
- Jean Goubault-Larrecq. 2019. A Probabilistic and Non-Deterministic Call-by-Push-Value Language. In *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019*. IEEE, 1–13. <https://doi.org/10.1109/LICS.2019.8785809>
- Andreas Griewank, Richard Hasenfelder, Manuel Radons, Lutz Lehmann, and Tom Streubel. 2018. Integrating Lipschitzian dynamical systems using piecewise algorithmic differentiation. *Optimization Methods and Software* 33, 4-6 (2018), 1089–1107. <https://doi.org/10.1080/10556788.2017.1378653>
- Giulio Guerrieri and Giulio Manzonetto. 2018. The Bang Calculus and the Two Girard’s Translations. In *Proceedings Joint International Workshop on Linearity & Trends in Linear Logic and Applications, Linearity-TLLA@FLoC 2018, Oxford, UK, 7-8 July 2018. (EPTCS)*, Thomas Ehrhard, Maribel Fernández, Valeria de Paiva, and Lorenzo Tortora de Falco (Eds.), Vol. 292. 15–30. <https://doi.org/10.4204/EPTCS.292.2>
- Martin Hyland. 1975. A syntactic characterization of the equality in some models for the λ -calculus. *Journal London Mathematical Society* (2) 12(3) (1975), 361–370.
- Bart Jacobs and Jan Rutten. 1997. A Tutorial on (Co)Algebras and (Co)Induction. *EATCS Bulletin* 62 (1997), 62–222.
- Gilles Kahn and Gordon D. Plotkin. 1978. *Domaines Concrets*. Technical Report 333. Rapport INRIA-LABORIA.
- Emma Kerinec, Giulio Manzonetto, and Michele Pagani. 2018. Revisiting Call-by-value Böhm trees in light of their Taylor expansion. *CoRR* abs/1809.02659 (2018). Accepted in Logical Methods in Computer Science.
- Dexter Kozen and Alexandra Silva. 2017. Practical coinduction. *Mathematical Structures in Computer Science* 27, 7 (2017), 1132–1152.
- Jan Kuper. 1995. Proving the Genericity Lemma by Leftmost Reduction is Simple. In *RTA (Lecture Notes in Computer Science)*, Vol. 914. Springer, 271–278.
- 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)*, Herman Geuvers (Ed.), Vol. 131. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 13:1–13:16. <https://doi.org/10.4230/LIPIcs.FSCD.2019.13>
- Ugo Dal Lago and Margherita Zorzi. 2012. Probabilistic operational semantics for the lambda calculus. *RAIRO - Theor. Inf. and Applic.* 46, 3 (2012), 413–450.
- Jim Laird, Giulio Manzonetto, Guy McCusker, and Michele Pagani. 2013. Weighted Relational Models of Typed Lambda-Calculi. In *28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013*. IEEE Computer Society, 301–310. <https://doi.org/10.1109/LICS.2013.36>
- Soren B. Lassen. 1999. Bisimulation in Untyped Lambda Calculus: Böhm Trees and Bisimulation up to Context. *Electr. Notes Theor. Comput. Sci.* 20 (1999), 346–374.
- Thomas Leventis. 2018. Probabilistic Böhm Trees and Probabilistic Separation. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018*, Anuj Dawar and Erich Grädel (Eds.). ACM, 649–658. <https://doi.org/10.1145/3209108.3209126>
- Paul Blain Levy. 2006. Call-by-push-value: Decomposing call-by-value and call-by-name. *Higher-Order and Symbolic Computation* 19, 4 (2006), 377–414. <https://doi.org/10.1007/s10990-006-0480-6>
- Stefania Lusina and Antonino Salibra. 2004. The Lattice of Lambda Theories. *J. Log. Comput.* 14, 3 (2004), 373–394.

- Giulio Manzonetto. 2009. A General Class of Models of \mathcal{H}^* . In *Mathematical Foundations of Computer Science 2009, 34th International Symposium, MFCS 2009, Novy Smokovec, High Tatras, Slovakia, August 24-28, 2009. Proceedings (Lecture Notes in Computer Science)*, Rastislav Kráľovic and Damian Niwinski (Eds.), Vol. 5734. Springer, 574–586. https://doi.org/10.1007/978-3-642-03816-7_49
- Giulio Manzonetto and Domenico Ruoppolo. 2014. Relational Graph Models, Taylor Expansion and Extensionality. *Electr. Notes Theor. Comput. Sci.* 308 (2014), 245–272.
- Rob P. Nederpelt, J. Herman. Geuvers, and Roel C. de Vrijer (Eds.). 1994. *Selected Papers on Automath*. Studies in Logic and the Foundations of Mathematics, Vol. 133. North-Holland, Amsterdam.
- Michele Pagani, Peter Selinger, and Benoît Valiron. 2014. Applying quantitative semantics to higher-order quantum computing. In *The 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '14, San Diego, CA, USA, January 20-21, 2014*, Suresh Jagannathan and Peter Sewell (Eds.). ACM, 647–658. <https://doi.org/10.1145/2535838.2535879>
- Luca Paolini, Mauro Piccolo, and Simona Ronchi Della Rocca. 2017. Essential and relational models. *Mathematical Structures in Computer Science* 27, 5 (2017), 626–650. <https://doi.org/10.1017/S0960129515000316>
- Gordon D. Plotkin. 1974. The lambda-Calculus is ω -Incomplete. *Journal of Symbolic Logic* 39, 2 (1974), 313–317.
- Gordon D. Plotkin. 1975. Call-by-Name, Call-by-Value and the lambda-Calculus. *Theor. Comput. Sci.* 1, 2 (1975), 125–159. [https://doi.org/10.1016/0304-3975\(75\)90017-1](https://doi.org/10.1016/0304-3975(75)90017-1)
- Dana S. Scott. 1972. Continuous lattices. In *Toposes, Algebraic Geometry and Logic (Lecture Notes in Mathematics)*, Lawvere (Ed.), Vol. 274. Springer, 97–136.
- Richard Statman and Henk Barendregt. 1999. Applications of Plotkin-Terms: Partitions and Morphisms for Closed Terms. *J. Funct. Program.* 9, 5 (1999), 565–575.
- Masako Takahashi. 1994. A Simple Proof of the Genericity Lemma. In *Logic, Language and Computation (Lecture Notes in Computer Science)*, Vol. 792. Springer, 117–118.
- Takeshi Tsukada, Kazuyuki Asada, and C.-H. Luke Ong. 2018. Species, Profunctors and Taylor Expansion Weighted by SMCC: A Unified Framework for Modelling Nondeterministic, Probabilistic and Quantum Programs. In *LICS*. ACM, 889–898.
- Dirk van Daalen. 1980. *The Language Theory of Automath*. Ph.D. thesis. Technical University Eindhoven. Large parts of the thesis, including the treatment of reduction under substitution, have been reproduced in [Nederpelt et al. 1994].
- Lionel Vaux. 2009. The algebraic lambda calculus. *Mathematical Structures in Computer Science* 19, 5 (2009), 1029–1059.
- Lionel Vaux. 2019. Normalizing the Taylor expansion of non-deterministic λ -terms, via parallel reduction of resource vectors. *Logical Methods in Computer Science* Volume 15, Issue 3 (2019).
- Christopher P. Wadsworth. 1976. The Relation Between Computational and Denotational Properties for Scott's \mathcal{D}_∞ -Models of the Lambda-Calculus. *SIAM J. Comput.* 5, 3 (1976), 488–521.
- Sebastian F. Walter and Lutz Lehmann. 2013. Algorithmic differentiation in Python with AlgoPy. *J. Comput. Science* 4, 5 (2013), 334–344. <https://doi.org/10.1016/j.jocs.2011.10.007>