1 / 90 2026:7 TheoretiCS

# Extensional Taylor Expansion

**Received** Sep 17, 2024
**Revised** Oct 2, 2025
**Accepted** Dec 13, 2025
**Published** Apr 16, 2026

**Key words and phrases**
Lambda-calculus, Taylor expansion of lambda-terms, game semantics

**Lison Blondeau-Patissier** [a,b,c,d]
**Pierre Clairambault** [a,b]
**Lionel Vaux Auclair** [a,c]

**a** Aix Marseille Univ, CNRS, Marseille, France

**b** LIS, Marseille, France

**c** I2M, Marseille, France

**d** ENS de Lyon, CNRS, Université Claude Bernard Lyon 1, LIP, Lyon, France

**ABSTRACT.** We introduce a calculus of extensional resource terms. These are resource terms *à la* Ehrhard–Regnier, but in infinitely $\eta$-long form. The calculus still retains a finite syntax and dynamics: in particular, we prove strong confluence and normalization.

Then we define an extensional version of Taylor expansion, mapping ordinary $\lambda$-terms to (possibly infinite) linear combinations of extensional resource terms: like in the ordinary case, the dynamics of our resource calculus allows us to simulate the $\beta$-reduction of $\lambda$-terms; the extensional nature of this expansion shows in the fact that we are also able to simulate $\eta$-reduction.

In a sense, extensional resource terms contain a language of finite approximants of Nakajima trees, much like ordinary resource terms can be seen as a richer version of finite Böhm trees. We show that the equivalence induced on $\lambda$-terms by the normalization of extensional Taylor-expansion is nothing but $\mathbf{H}^*$, the greatest consistent sensible $\lambda$-theory – which is also the theory induced by Nakajima trees. This characterization provides a new, simple way to exhibit models of $\mathbf{H}^*$: it becomes sufficient to model the extensional resource calculus and its dynamics.

The extensional resource calculus moreover allows us to recover, in an untyped setting, a connection between Taylor expansion and game semantics that was previously limited to the typed setting. Indeed, simply typed, $\eta$-long, $\beta$-normal resource terms are known to be in bijective correspondence with plays in the sense of Hyland-Ong game semantics, up to Melliès' homotopy equivalence. Extensional resource terms are the appropriate counterpart of $\eta$-long

---

This work was partially supported by the French ANR projects DyVerSe (ANR-19-CE48-0010-01), PPS (ANR-19-CE48-0014), LambdaComb (ANR-21-CE48-0017) and Reciprog (ANR-21-CE48-0019).

**Cite as** Lison Blondeau-Patissier, Pierre Clairambault, Lionel Vaux Auclair. Extensional Taylor Expansion. TheoretiCS, Volume 5 (2026), Article 7, 1-90.

https://theoretics.episciences.org
**DOI** 10.46298/theoretics.26.7

resource terms in an untyped setting: we spell out the bijection between normal extensional resource terms and isomorphism classes of augmentations (a canonical presentation of plays up to homotopy) in the universal arena.

## 1. Introduction

The Taylor expansion of $\lambda$-terms has profoundly renewed the approximation theory of the $\lambda$-calculus by providing a quantitative alternative to order theoretic approximation techniques, the latter being famously embodied in the notion of Böhm tree [2]: a key result of Ehrhard and Regnier's seminal series of papers [19, 16] is the fact that the normal form of the Taylor expansion of a $\lambda$-term is the Taylor expansion of its Böhm tree. Taylor expansion can thus be seen as mediating between the potentially infinite dynamics of finite $\lambda$-terms, and the static but potentially infinite Böhm trees. In order to expose the context and motivations of our contributions, we find useful to first review these notions, only assuming knowledge of the ordinary $\lambda$-calculus.

### 1.1 Böhm trees and ordinary Taylor expansion

The results we present in this subsection are digested from well-established literature.[1] For the reader discovering one of the subjects, or both, it can serve as a very quick and opinionated survey. And for the reader versed in both subjects, as well as for the newcomer, we hope it will convey intuitions that they can advantageously summon up when we turn to the main matter of the paper. The expert reader might still prefer to jump directly to Section 1.2, where we discuss the literature, and some less established folklore, about extensionality in relation to Taylor expansion; or even to Section 1.3, where we outline our contributions.

**Böhm trees.** We can always write a $\lambda$-term as $M = \lambda x_1. \cdots \lambda x_k.M' \, N_1 \cdots N_l$,[2] where:

— either $M' = y$ is the **head variable** of $M$, and then $M$ is in **head normal form**;

---

[1] For a comprehensive treatment of the theory of Böhm trees, the reader may refer to classic textbooks such as Barendregt's [2] or Krivine's [28]. The first chapters of Barendregt and Manzonetto's *Satellite* [3, esp. Section 2.3] offer a modern, self-sufficient survey of the theory. Ehrhard and Regnier obtained their seminal commutation theorem, relating Taylor expansion with Böhm trees via normalization (Equation (2) below), by combining the results of two papers [19, 16]. Both papers involve considerable technical developments to unveil deep, distinctive properties of the Taylor expansion of $\lambda$-terms (a uniformity property of the support of Taylor expansion, an explicit formula for coefficients, a precise connection with execution in an abstract machine) of which the commutation theorem is but a consequence. The third author showed that the same theorem could be established in a more direct fashion, by simulating $\beta$-reduction through Taylor expansion [39]: although we do not even sketch a proof of the commutation theorem, our exposition of ordinary Taylor expansion is inspired by that latter route, as we will leverage similar techniques in our treatment of the extensional case.

[2] We use standard notational conventions to avoid repetition of parentheses: application has precedence over abstraction, and we associate applications on the left. So $\lambda x_1. \cdots \lambda x_k.M' \, N_1 \cdots N_l$ should be read as $\lambda x_1.(\cdots \lambda x_k.((M' \, N_1) \cdots N_l) \cdots)$.

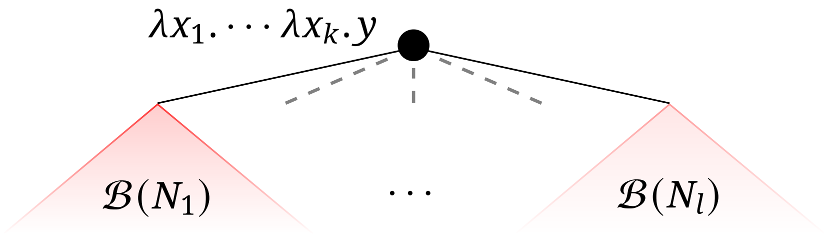

**Figure 1.** Shape of a (non-$\perp$) Böhm tree

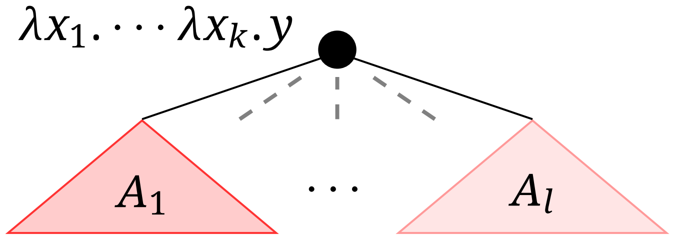

**Figure 2.** Shape of a (non-$\perp$) approximant

- or $M' = (\lambda z.P)\,N_0$ is the **head redex** of $M$, in which case the **head reduction strategy** deterministically reduces $M$ to $\mathcal{H}(M) := \lambda x_1.\cdots\lambda x_k.P\{N_0/z\}\,N_1\cdots N_l$, where $P\{N_0/z\}$ denotes the usual capture avoiding substitution.

Head reduction plays a central rôle in the theory, chiefly because head normalizable terms are exactly those terms that are **solvable**: informally, a term $M$ is solvable when it can interact with its evaluation context via normalization; a possible definition is to require the existence of a context $C[\,]$ of the shape $(\lambda z_1.\cdots\lambda z_m.[\,])\,P_1\cdots P_n$ such that $C[M]$ $\beta$-normalizes to the identity term $\lambda x.x$.

**FACT 1.1.** *The following three properties are equivalent:*

*(i) the sequence of head reductions starting from $M$ is finite ($M$ is **head normalizable**);*
*(ii) $M$ is $\beta$-equivalent to a head normal form;*
*(iii) $M$ is solvable.*

Conversely, an **unsolvable** term is one whose structure cannot be probed by the environment via normalization: applying an unsolvable term to an argument, or substituting a variable in that term with any other term, will never yield a head normal form (let alone a normal form).

The **Böhm tree** of a term $M$ is then a possibly infinite tree $\mathcal{B}(M)$, defined coinductively:

- if $M$ head normalizes to $\lambda x_1.\cdots\lambda x_k.y\,N_1\cdots N_l$ then $\mathcal{B}(M) := \lambda x_1.\cdots\lambda x_k.y\,\mathcal{B}(N_1)\cdots\mathcal{B}(N_l)$ which we consider as a tree whose root is labelled with the abstractions and head variable, and with $l$ immediate subtrees, depicted as in Figure 1 (we leave the bottom of each triangle open to indicate that the tree is possibly infinite);
- if $M$ is unsolvable, then $\mathcal{B}(M)$ is reduced to a leaf denoted $\perp$.

Like $\lambda$-terms, Böhm trees are considered up to $\alpha$-equivalence.[3]

---

[3] Although they are possibly infinite trees, the set of free variables of any subtree of $\mathcal{B}(M)$ is always finite, so this poses no particular difficulty.

**FACT 1.2.** *Writing $M =_{\mathcal{B}} M'$ when $\mathcal{B}(M) = \mathcal{B}(M')$, we obtain a $\lambda$-theory,* i.e. *a congruence on $\lambda$-terms containing $\beta$-conversion. This $\lambda$-theory is moreover* **sensible**: *it equates all unsolvable terms.*

It turns out that the only difficult part in establishing the previous fact is to show that $=_{\mathcal{B}}$ is compatible with application: if $M =_{\mathcal{B}} M'$ and $N =_{\mathcal{B}} N'$ then $M\,N =_{\mathcal{B}} M'\,N'$, which amounts to showing that $\mathcal{B}(M\,N)$ is determined by the sole information of $\mathcal{B}(M)$ and $\mathcal{B}(N)$. A standard route to establish the compatibility of $=_{\mathcal{B}}$ with syntactic constructs is to rely on **finite approximants** of Böhm trees. The latter are particular $\beta$-normal terms of $\Lambda_\perp$, the $\lambda$-calculus augmented with the "undefined" constant $\perp$, whose shape follows that of Böhm trees:

- $\perp$ is an approximant;
- if $A_1, \ldots, A_l$ are approximants, then so is $\lambda x_1.\cdots\lambda x_k.y\,A_1\cdots A_l$, which we depict as in Figure 2 (we use closed triangles for finite trees).

These approximants are thus both terms of $\Lambda_\perp$ and finite Böhm-like trees (*i.e.* trees with internal nodes as in Figure 1, and $\perp$ leaves). The **information order** $\leq_\perp$ is defined both on $\Lambda_\perp$ and on Böhm-like trees, as the partial order such that $M \leq_\perp N$ when $M$ is obtained from $N$ by replacing any number of subterms with $\perp$ (possibly infinitely many in case $M$ is an infinite Böhm-like tree). The set of finite approximants of a $\lambda$-term $M$ is then

$$\mathcal{B}_f(M) = \{A \mid \exists M' =_\beta M,\ A \leq_\perp M'\}\,.$$

The *syntactic approximation* theorem [3, Theorem 2.32] states that $\mathcal{B}(M)$ is nothing but the supremum (in the directed-complete partial order of Böhm-like trees equipped with $\leq_\perp$) of $\mathcal{B}_f(M)$. And the *syntactic continuity* theorem [3, Proposition 2.34] establishes that the notion of approximant is compatible with syntactic constructs: given a term $M$ and a context $C[\,]$, $\mathcal{B}_f(C[M])$ depends only on $\mathcal{B}_f(M)$ and $C[\,]$.[4]

**The resource calculus.** We have seen that the approximants associated with the Böhm tree interpretation can be considered as partial $\lambda$-terms in normal form. By contrast, the target of the Taylor expansion of $\lambda$-terms is supported by a language of approximants called **resource terms**, that are not necessarily normal, but whose reduction behaves linearly. These are just like ordinary $\lambda$-terms, except for the application constructor: a resource term $m$ is applied not just to one argument, but to a bag (a finite multiset) of arguments $\bar{n} = [n_1, \ldots, n_l]$, yielding a new term $m\,\bar{n}$, which we may depict as in Figure 3 (we use trapezia for forests, representing bags).

Resource terms retain a dynamics, induced by a *linear* variant of $\beta$-reduction: if $\bar{n} = [n_1, \ldots, n_l]$ and $x_1, \ldots, x_k$ enumerate the occurrences of $x$ in $m$, then $(\lambda x.m)\,\bar{n}$ reduces to $m[\bar{n}/x]$,

---

[4] Another well-known route to show that $=_{\mathcal{B}}$ is a congruence, also discussed in the *Satellite* [3], is via infinitary $\lambda$-calculi: in some sense (that can be made formal [25]), $\mathcal{B}(M\,N)$ is the normal form, for an infinitary extension of $\beta$-reduction, of the application $\mathcal{B}(M)\,\mathcal{B}(N)$.

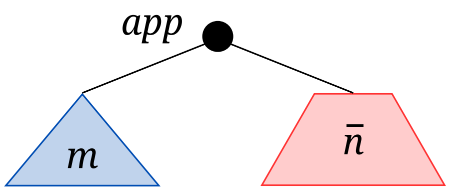

**Figure 3.** Depiction of $m\,\bar{n}$

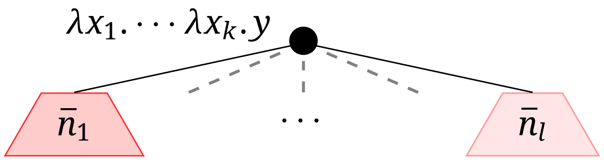

**Figure 4.** Shape of a normal resource term

the **resource substitution** of $\bar{n}$ for $x$ in $m$, defined as the finite sum of resource terms

$$\sum_{\sigma\in\mathbb{S}_k} m\{n_1,\ldots,n_k/x_{\sigma(1)},\ldots,x_{\sigma(k)}\}$$

in case $k = l$ ($\mathbb{S}_k$ denotes the set of permutations of $\{1,\ldots,k\}$), or the empty sum 0 in case $k \neq l$: each summand of $m[\bar{n}/x]$ is the result of a one-to-one substitution of the elements of $\bar{n}$ (taking multiplicities into account) for the occurrences of $x$ in $m$. The dynamics thus involves finite formal sums of expressions, and all syntactic constructs are extended to sums by linearity: e.g., application is bilinear $(\sum_{i\in I} m_i)\,(\sum_{j\in J} \bar{n}_j) \coloneqq \sum_{i\in I}\sum_{j\in J} m_i\,\bar{n}_j$. The base case of reduction is then extended to a **resource reduction** relation on finite sums of resource terms by allowing to fire a redex in any context and inside sums: for instance, if $m$ reduces to $M' = \sum_{i\in I} m'_i$, then $\lambda x.m$ reduces to $\lambda x.M' = \sum_{i\in I} \lambda x.m'_i$, and $m + N$ reduces to $M' + N$, for any finite sum $N$. Again, reducing any redex $(\lambda x.m)\,\bar{n}$ such that the cardinality of $\bar{n}$ does not match the number of occurrences of $x$ in $m$ will yield 0: by linearity, this will annihilate the summand containing this redex.

In any summand $p'$ of $m[\bar{n}/x]$, the elements of $\bar{n}$ are substituted for variable occurrences in $m$ but never duplicated: it follows that the size (*i.e.* the number of syntactic constructs) of $p'$ is strictly smaller than that of the redex $(\lambda x.m)\,\bar{n}$. It is then easy to establish that:

**FACT 1.3.** *Resource reduction is confluent and strongly normalizing.*

So any finite sum of resource terms $M$ reduces to a unique normal form $\mathcal{N}(M)$, in such a way that $\mathcal{N}(M) = \sum_{i\in I} \mathcal{N}(m_i)$ whenever $M = \sum_{i\in I} m_i$. A normal resource term is necessarily of the shape $\lambda x_1.\cdots\lambda x_k.y\,\bar{n}_1\cdots\bar{n}_l$ – see Figure 4

Observe that non-$\bot$ Böhm approximants are nothing but normal resource terms with bags of size at most one: more precisely, any non-$\bot$ approximant corresponds to a normal resource term using the empty multiset for any $\bot$ subterm, and singleton multisets for the non-$\bot$ subterms. Bags of arbitrary size are nonetheless essential for the resource calculus to also provide an approximation of $\beta$-reduction and normalization via Taylor expansion, as we outline below.

**Taylor expansion at work.** The Taylor expansion $\mathcal{T}(M)$ of a $\lambda$-term $M$ is the vector (*i.e.* the possibly infinite linear combination) of resource terms inductively defined by:

$$\mathcal{T}(x) := x \qquad \mathcal{T}(\lambda x.M) := \lambda x.\mathcal{T}(M) \qquad \mathcal{T}(M\,N) := \sum_{k\in\mathbb{N}} \frac{1}{k!}\mathcal{T}(M)\,\mathcal{T}(N)^k$$

where, again, syntactic constructs are extended to arbitrary weighted sums of terms by linearity, and $\mathcal{T}(N)^k := [\mathcal{T}(N), \dots, \mathcal{T}(N)]$ is a weighted sum of bags all of size $k$ – and $k!$ is the cardinality of $\mathbb{S}_k$.[5] Here we choose to dispense with the technicalities of dealing with infinite sums by considering coefficients in the extended half line $[0, +\infty]$, or more generally in any suitably complete semiring – a precise definition of the necessary structure will be recalled in Section 4.

Given a vector of resource terms $M$, it is often useful to consider the **promotion** of $M$, which is the vector of bags defined as $M^! := \sum_{k\in\mathbb{N}} \frac{1}{k!}M^k$. In particular, the Taylor expansion of an application can be written as $\mathcal{T}(M\,N) = \mathcal{T}(M)\,\mathcal{T}(N)^!$ (the application on the right hand side being that of the resource calculus). The crucial feature of Taylor expansion is that it allows us to decompose the usual substitution operation on $\lambda$-terms into resource substitution and promotion:

$$\mathcal{T}(M\{N/x\}) = \mathcal{T}(M)[\mathcal{T}(N)^!/x]\,. \tag{1}$$

The proof of Equation (1) is by a simple induction on $M$, relying on basic combinatoric arguments in the case of an application [39, Lemmas 4.3 and 4.7].[6] Now, we can consider the normal form of any vector as defined by linearity $\mathcal{N}(\sum_{i\in I} \alpha_i m_i) := \sum_{i\in I} \alpha_i \mathcal{N}(m_i)$ (where $I$ is not necessarily finite), and then write $M =_{\mathcal{T}} N$ when $\mathcal{N}(\mathcal{T}(M)) = \mathcal{N}(\mathcal{T}(N))$.

Showing that the equivalence relation $=_{\mathcal{T}}$ is a $\lambda$-theory is easy. Indeed, its compatibility with syntactic constructs follows from the confluence property of resource reduction: in particular, $\mathcal{N}(m\,[n_1, \dots, n_k])$ is also the normal form of $\mathcal{N}(m)\,[\mathcal{N}(n_1), \dots, \mathcal{N}(n_k)]$, which ensures that $\mathcal{N}(\mathcal{T}(M\,N))$ is also the normal form of $\mathcal{N}(\mathcal{T}(M))\,\mathcal{N}(\mathcal{T}(N))^!$, thus settling the case of application straightforwardly. Similarly, thanks to Equation (1), $\mathcal{N}(\mathcal{T}(M\{N/x\}))$ is the normal form of $\mathcal{T}(M)[\mathcal{T}(N)^!/x]$ hence of $(\lambda x.\mathcal{T}(M))\,\mathcal{T}(N)^! = \mathcal{T}((\lambda x.M)\,N)$, which ensures that $=_{\mathcal{T}}$ contains $\beta$-reduction.

Showing that $=_{\mathcal{T}}$ is also sensible provides a good example of Taylor expansion at work. More precisely, we show that:

---

[5] Note that the case of application in the definition of Taylor expansion is nothing but the usual formula defining the Taylor series of an infinitely differentiable map at 0, provided one interprets the resource application $M\,[N_1, \dots, N_k]$ (where $M, N_1, \dots, N_k$ denote $\lambda$-terms, or vectors of resource terms) as the application of the $k$-th derivative at 0 of $M$ to the tuple $\langle N_1, \dots, N_k\rangle$, this application being $k$-linear and symmetric. The resource calculus is precisely the fragment of the differential $\lambda$-calculus [18] supporting the target of this recursive Taylor expansion. We will briefly discuss this analytic interpretation again at the end of this subsection.

[6] The original proof by Ehrhard and Regnier [19, Theorem 32] follows a more contorted path, because they insist on establishing an explicit formula for the coefficients of resource terms in Taylor expansions.

**LEMMA 1.4.** *A $\lambda$-term $M$ is head normalizable iff $\mathcal{N}(\mathcal{T}(M)) \neq 0$.*

**PROOF.** First observe that if $M$ is in head normal form, then $\mathcal{T}(M)$ contains a resource term of the shape $\lambda x_1.\cdots\lambda x_k.x\,[\,]\cdots[\,]$, which is normal, so $\mathcal{N}(\mathcal{T}(M)) \neq 0$. Now, if $M$ is head normalizable, then $M$ reduces to some head normal form $M'$, and then $M =_{\mathcal{T}} M'$ because $=_{\mathcal{T}}$ contains $\beta$-reduction, so $\mathcal{N}(\mathcal{T}(M)) = \mathcal{N}(\mathcal{T}(M')) \neq 0$.

For the reverse implication, we start by observing that Taylor expansion commutes with head reduction, which we define on resource terms in the same way as on $\lambda$-terms:

$$\mathcal{H}(\lambda x_1.\cdots\lambda x_k.(\lambda y.m)\,\bar{n}_0\cdots\bar{n}_l) := \lambda x_1.\cdots\lambda x_k.m[\bar{n}_0/y]\,\bar{n}_1\cdots\bar{n}_l\,.$$

Then we define $\mathcal{H}(M)$ for any weighted sum $M$ of head reducible terms by linearity, and Equation (1) gives $\mathcal{T}(\mathcal{H}(M)) = \mathcal{H}(\mathcal{T}(M))$ for any $\lambda$-term $M$ not in head normal form. Now, assuming $\mathcal{N}(\mathcal{T}(M)) \neq 0$, we can pick an element $m$ in the support of $\mathcal{T}(M)$ such that $\mathcal{N}(m) \neq 0$, and then we show by induction on the size of $m$ that $M$ is head normalizable: either $M$ is already in head normal form; or at least one resource term $m'$ in $\mathcal{H}(m)$ is such that $\mathcal{N}(m) \neq 0$, and we observe that $m'$ is in $\mathcal{H}(\mathcal{T}(M)) = \mathcal{T}(\mathcal{H}(M))$, which ensures that $\mathcal{H}(M)$ is head normalizable by induction hypothesis. ■

Observe that the previous argument actually provides a proof of the implication from (ii) to (i) in Fact 1.1 – this implication and the one from (iii) to (i) are the only ones that are not easy consequences of the definitions. This implication is classically proved by standardization [2, Corollary 11.4.8, 36, Corollary 2.7] or reducibility techniques [28, Theorem 4.9]. This demonstrates a nice conceptual contribution of Taylor expansion: instead of reasoning on reduction paths, we can pick a well-chosen element of the Taylor expansion and use it as a decreasing measure for a proof by induction, exploiting the fact that reduction in the resource calculus reflects $\beta$-reduction.

**Taylor expansion as an alternative to Böhm trees.** The sensible $\lambda$-theories $=_{\mathcal{B}}$ and $=_{\mathcal{T}}$ actually coincide. Indeed, it is straightforward to extend the definition of Taylor expansion to Böhm-like trees, in such a way that $\mathcal{T}(\perp) = 0$, and $\mathcal{B}_f(M)$ is the set of finite approximants occurring (as resource terms) in $\mathcal{T}(\mathcal{B}(M))$: $\mathcal{B}(M)$ is entirely determined by $\mathcal{T}(\mathcal{B}(M))$. Ehrhard and Regnier's commutation theorem [16, Corollary 1] establishes the identity:

$$\mathcal{N}(\mathcal{T}(M)) = \mathcal{T}(\mathcal{B}(M)) \tag{2}$$

which ensures that $M =_{\mathcal{T}} M'$ iff $M =_{\mathcal{B}} M'$. We do not develop the proof of Equation (2): the reader may refer to Ehrhard and Regnier's papers for the original proof [19, 16], or to the arguably more direct approach by the third author [39], based on the simulation of $\beta$-reduction. One can thus view Taylor expansion as a practical alternative to Böhm trees: the normal form of Taylor expansion subsumes the approximation theory determined by Böhm trees, but Taylor expansion also provides non-normal approximants, together with an analysis of $\beta$-reduction via resource reduction.

Barbarossa and Manzonetto [1] have demonstrated at length how to leverage this approach to revisit old results and establish new ones, in a generic and principled way, systematically reasoning inductively on the size (or the length of a particular reduction path) of a well-chosen resource term.

It is worth noting that, as these authors focus on the order-based approximation theory of the $\lambda$-calculus, this latter work only relies on the qualitative version of Taylor expansion, that is obtained by replacing $\mathcal{T}(M)$ with its support set – or, equivalently, by taking scalar coefficients in the boolean semiring. This is all the more justified because, by Ehrhard and Regnier's uniformity results [19], the Taylor expansion of a pure $\lambda$-term is entirely characterized by its support. However, the quantitative information of coefficients underpins the analytic interpretation of Taylor expansion as a sum of iterated derivatives, and the full (quantitative) Taylor expansion is strictly more informative as soon as one departs from that uniform setting.

**Beyond the pure $\boldsymbol{\lambda}$-calculus and plain $\boldsymbol{\beta}$-reduction.** Indeed, one strength of Taylor expansion as a framework for programming language semantics is its modularity, which is essentially inherited from its origins in *quantitative semantics*, as initiated by Girard [22] and later revisited by Ehrhard in a typed setting [15]: the basic idea of quantitative semantics is to interpret $\lambda$-terms as generalized power series, associated with analytic maps between spaces of some suitable category. Although it will play no explicit rôle in the remainder of the paper, this analytic interpretation was crucial for the design of the differential $\lambda$-calculus and the Taylor expansion of $\lambda$-terms by Ehrhard and Regnier: these models satisfy the usual Taylor formula as an identity, of which the Taylor expansion of programs can be understood as a syntactic, computational counterpart.

It then becomes natural to account for various flavours of superposition of programs via sums. For instance in a discrete probabilistic setting, one can turn the probabilistic choice $M \oplus_p N$ (representing a choice between $M$ with probability $p$ and $N$ with probability $1 - p$) into a weighted sum: $\mathcal{T}(M \oplus_p N) := p\mathcal{T}(M) + (1 - p)\mathcal{T}(N)$. Dal Lago and Leventis have shown that, again, this corresponds to a notion of probabilistic Böhm tree [13], via normalization: it is notable that this extension is straightforward on the side of Taylor expansion, whereas the development of an adequate notion of probabilistic Böhm trees by Leventis required considerable technical work [29].

Taylor expansion moreover enjoys a tight connection with linear logic [21], which was also founded on quantitative semantics: Ehrhard's version of quantitative semantics [15] is actually a denotational model of linear logic, and it is possible to introduce a differential version of linear logic [17], together with a notion of Taylor expansion which reflects the structure of the model, and refines the Taylor expansion of $\lambda$-terms. The paradigm of Taylor expansion can then be ported to various extensions or variants of the $\lambda$-calculus, and more generally to systems that "play well" with linear logic [10, 11, 14], at the price of designing a resource calculus providing a suitable linearization of the source system.

It is reasonable to expect that such notions of Taylor expansion will yield interesting and robust approximation theories via normalization. For instance, Kerinec, Manzonetto and Pagani [27] have followed this path precisely, in the case of the call-by-value $\lambda$-calculus, and this guided their definition of an adequate notion of call-by-value Böhm tree.

## 1.2 Towards extensionality

Up to the present paper, one notable case falls outside of the scope that we have just delineated: extensionality and the $\eta$-rule. A $\lambda$-theory $\sim$ is **extensional** when $M \sim M'$ as soon as, for each term $N$, $M\,N \sim M'\,N$. Equivalently, $\sim$ is extensional if it contains the **$\eta$-rule**, reducing $\lambda x.M\,x$ to $M$ when $x$ is fresh, *i.e.* not a free variable of $M$. The least extensional $\lambda$-theory is thus the reflexive, symmetric, and transitive closure $=_{\beta\eta}$ of the union of $\beta$- and $\eta$-reductions.

**Extensionality via a global transformation.** From the viewpoint of ordinary Taylor expansion, and in contrast with $\beta$-reduction, $\eta$-reduction cannot be captured as a superposition of independent reductions on resource terms. Indeed, picking variables $x \neq y$ and any $\lambda$-term $M$, observe that $\mathcal{T}(\lambda x.y\,M) = \lambda x.y\,\mathcal{T}(M)^!$ always contains the resource term $\lambda x.y\,[\,]$ as a summand, but $\lambda x.y\,M$ $\eta$-reduces to $y$ only in case $M = x$.

Nevertheless, Manzonetto and Ruoppolo [32] introduced a *global* notion of $\eta$-reduction on sets of normal resource terms (satisfying a technical condition), which they used to characterize Morris' equivalence (the observational equivalence induced by $\beta$-normal forms). More precisely: they first consider the support set of the normal form of ordinary Taylor expansion; then they apply a further step of $\eta$-normalization guided by the global structure of this set, which yields a new set of ordinary resource terms, still in normal form; and they prove that this construction induces the same equational theory as Böhm trees up to countably many finitely nested $\eta$-expansions, which are known to capture Morris' equivalence [23].

This process is limited to a qualitative setting, and the $\eta$-rule is not reflected in resource reduction. By contrast, in the present paper, we enforce extensionality during Taylor expansion, so that $\eta$-reduction is treated just like $\beta$-reduction during normalization, all this in a quantitative setting. In the typed case, our approach can be reduced to a well-known trick: considering $\eta$-long forms.[7] We readily expose this piece of folklore as a stepping stone to the untyped case.

---

[7] For instance, the restriction of the resource calculus to $\eta$-long forms was leveraged by Tsukada *et al.* [38, 37], as well as the authors of the present paper [5], in connection with game semantics. More basically, the fact that extensionality can be enforced in a typed setting by considering $\beta$-reduction on $\eta$-long forms is very standard knowledge.

**The typed case.** Consider simply typed $\lambda$-terms *à la* Church, where the grammar of types (denoted by $\alpha$, $\beta$, $\gamma$, …) is inductively generated from a single base type $o$ by the formation of arrow types $\alpha \to \beta$. Each type $\alpha$ can be written uniquely as $\alpha = \beta_1 \to (\cdots \to (\beta_k \to o) \cdots)$: $k$ is the arity of $\alpha$, and we use the notation $\langle \beta_1, \ldots, \beta_k \rangle \to o$ in this case. A typed term $M$ is **$\eta$-long** if each occurrence of a subterm with arrow type $\alpha \to \beta$ is either an abstraction or applied to a subterm of type $\alpha$: performing a step of typed $\eta$-expansion in $M$ will always generate a $\beta$-redex. Equivalently, $M$ is $\eta$-long if each occurrence $M'$ of a variable or redex in $M$ is **fully applied**, *i.e.* it occurs in a subterm $M' \, N_1 \cdots N_k$ where $k$ is the arity of the type of $M'$. In particular, each application is part of such a full application sequence.

Given a typed $\lambda$-term $M$, an **$\eta$-long form** of $M$ is any $\eta$-expansion of $M$ that is $\eta$-long. We can always compute such an $\eta$-long form for a term of type $\alpha$, by setting:

$$x^\eta := \lambda y_1. \cdots \lambda y_k . x \, y_1^\eta \cdots y_k^\eta \qquad (\lambda z.M)^\eta := \lambda z.M^\eta \qquad (N \, P)^\eta := \lambda y_1. \cdots \lambda y_k . N^\eta \, P^\eta \, y_1^\eta \cdots y_k^\eta$$

where $k$ is the arity of $\alpha = \langle \beta_1, \ldots, \beta_k \rangle \to o$, and each $y_i$ is a fresh variable of type $\beta_i$ (this is a valid inductive definition because recursive calls are either on immediate subterms or on variables of strictly smaller type).[8] It is an easy exercise to check that: if $M$ $\beta\eta$-reduces to $N$ then $M^\eta$ $\beta$-reduces to $N^\eta$. Defining $\mathcal{T}_\eta(M) := \mathcal{T}(M^\eta)$, we can thus leverage the already established results on Taylor expansion to simulate both $\beta$- and $\eta$-reduction via resource reduction: it becomes immediate that $\mathcal{N}(\mathcal{T}_\eta(M)) = \mathcal{N}(\mathcal{T}_\eta(N))$ as soon as $M$ and $N$ are $\beta\eta$-equivalent terms of the same type.

Interestingly, the typing and extensionality constraints we consider on $\eta$-long terms admit straightforward counterparts in resource terms. Indeed, it is easy to adapt the simple type system to resource terms, in such a way that the elements of $\mathcal{T}(M)$ are all typed resource terms of the same type as $M$. Again, we say that a typed resource term is in $\eta$-long form if each occurrence of a subterm of arrow type $\alpha \to \beta$ is either an abstraction, or applied to a bag of terms all of type $\alpha$.

We can thus directly define the extensional Taylor expansion of a typed $\lambda$-term as a vector of $\eta$-long resource terms of the same type, by setting inductively:

$$\begin{aligned}
\mathcal{T}_\eta(x) &:= \lambda y_1. \cdots \lambda y_k . x \, \mathcal{T}_\eta(y_1)^! \cdots \mathcal{T}_\eta(y_k)^! \\
\mathcal{T}_\eta(\lambda z.M) &:= \lambda z.\mathcal{T}_\eta(M) \\
\mathcal{T}_\eta(N \, P) &:= \lambda y_1. \cdots \lambda y_k . \mathcal{T}_\eta(N) \, \mathcal{T}_\eta(P)^! \, \mathcal{T}_\eta(y_1)^! \cdots \mathcal{T}_\eta(y_k)^!
\end{aligned}$$

where, again, $k$ is the arity of the type, and each $y_i$ is a fresh variable of appropriate type. Moreover, $\eta$-long resource terms are stable under resource reduction, so the dynamics on which we rely is purely local, without any reference to a side condition or global rewriting constraints.

---

[8] This choice of $\eta$-long form may introduce redexes even if $M$ is $\beta$-normal. One can avoid this defect by inspecting the head structure of $M$ instead of its top-level constructor: we stick to this naïve version to ease the exposition here, but both approaches will be used in our treatment of untyped extensional Taylor expansion, in Section 5.

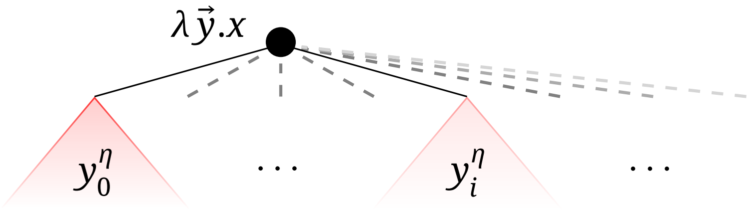

**Figure 5.** Infinite $\eta$-expansion of a variable

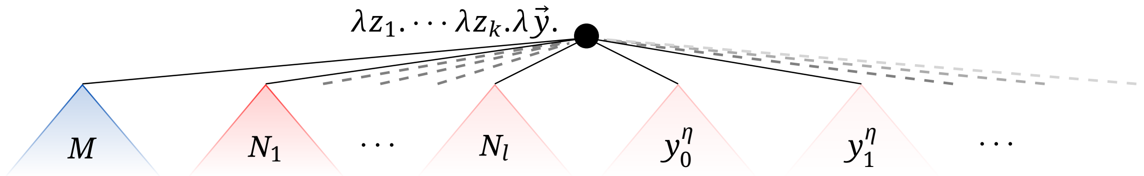

**Figure 6.** Shape of an infinitely $\eta$-long $\lambda$-term

**Enforcing $\eta$-longness in the untyped setting.** To guide the design of an extensional version of Taylor expansion, it is thus essentially sufficient to rely on $\eta$-long terms… The only issue is that there is no such thing in the untyped setting: without typing constraint, it is always possible to $\eta$-expand a term without creating any $\beta$-redex. For instance, given a variable $x$, we can consider iterated $\eta$-expansions of the head structure:

$$x \leftarrow_\eta \lambda y_1.x\, y_1 \leftarrow_\eta \lambda y_1.\lambda y_2.x\, y_1\, y_2 \leftarrow_\eta \cdots \leftarrow_\eta \lambda y_1.\cdots\lambda y_k.x\, y_1 \cdots y_k$$

or even nested $\eta$-expansions of fresh variables:

$$x \leftarrow_\eta \lambda z_1.x\, z_1 \leftarrow_\eta \lambda z_1.x\,(\lambda z_2.z_1\, z_2) \leftarrow_\eta \cdots \leftarrow_\eta \lambda z_1.x\,(\lambda z_2.z_1\,(\cdots \lambda z_k.z_{k-1}\, z_k \cdots))\,.$$

We can nonetheless consider the limit of iterating the combination of those two processes, as given by an infinite tree $x^\eta$ that we depict in Figure 5, where $\vec{y}$ is a sequence $\langle y_0, y_1, \ldots\rangle$ of fresh variables, and each $y_i^\eta$ is recursively produced in the same way. If we accept syntactic constructs with countable arity, we may thus write $x^\eta = \lambda\vec{y}.x\,\vec{y}^{\,\eta}$ where $\vec{y}^{\,\eta}$ denotes the sequence $\langle y_0^\eta, y_1^\eta, \cdots\rangle$, and we may understand $x^\eta$ as a kind of infinite term: $\lambda y_0.\lambda y_1.\cdots x\, y_0^\eta\, y_1^\eta \cdots$, where sequences of abstractions and applications account for countably iterated head expansions, and recursive calls account for nesting.

More generally, one can consider an intuitive depiction of infinitely $\eta$-long terms as given by infinite trees, as in Figure 6, where each $N_i$ denotes recursively such a tree, and $M$ is either a variable or a tree itself. The idea is to ensure that the head of the term is fully applied to countably many arguments, and this constraint is recursively applied both to subterms and fresh variables. In case $M = x$ and $k = l = 0$, we recover the particular case of $x^\eta$. Now, if we restrict to the case of $M$ being a variable, but extend the construction to allow for $\perp$-trees, the objects we have just described are nothing but Nakajima trees [34], which are canonical representatives of Böhm trees up to infinite $\eta$-expansion [2, Exercise 19.4.4].

One might attempt to equip those infinitely $\eta$-long terms with an infinitary dynamics, in the style of the infinitary $\lambda$-calculus [25] – note that the latter does not account for the application of a term to an infinite sequence of arguments. To our knowledge, however, this work has never been carried out, and it would require to tackle a number of technical issues, among which having terms with infinitely many free variables is the least problematic: e.g., one also needs to consider countably iterated head reduction, hence the simultaneous application of countably many substitutions, which is carefully avoided in the usual infinitary $\lambda$-calculus approach.

Fortunately, however, we will not need to follow that path: we only relied on infinitely $\eta$-long $\lambda$-terms as a pedagogical detour, preparing the reader for the introduction of extensional Taylor expansion, whose target is supported by a syntax of infinitely $\eta$-long, yet finite, resource terms.

## 1.3 Our contributions

In the present paper, we introduce a variant of Taylor expansion for pure, untyped $\lambda$-terms, in such a way that reduction in the associated resource calculus allows us to simulate both $\beta$- and $\eta$-reduction. We characterize the equational theory induced via normalization as the maximal consistent and sensible $\lambda$-theory, and apply this result to a particular relational model, demonstrating how this extensional Taylor expansion can be leveraged similarly to ordinary Taylor expansion. We moreover exhibit a precise correspondence between this framework and game semantics.

**The extensional resource calculus.** With the intuitions and notations of the previous subsection, one could try to naively apply ordinary Taylor expansion to an infinitely $\eta$-long term that we may denote $\lambda z_1.\dots.\lambda z_k.\lambda\vec{y}.M\,N_1\cdots N_l\,y_0^\eta\,y_1^\eta\cdots$ or even $\lambda\vec{y}.M\,N_1\cdots N_l\,y_k^\eta\,y_{k+1}^\eta\cdots$, up to $\alpha$-equivalence. In the support of this Taylor expansion, one should find resource terms such as: $\lambda\vec{y}.m\,\bar{n}_1\cdots\bar{n}_l\,\bar{p}_0\,\bar{p}_1\cdots$ where each $\bar{n}_i$ (resp. $\bar{p}_j$) is a bag of terms in the expansion of $N_i$ (resp. of $y^\eta$ for some variable $y$). Such a term still retains infinite sequences of abstracted variables and bags of arguments, but there are natural solutions to restrict this syntax to a finite setting:

- we consider $\vec{y}$ as a single abstracted variable, that we will call a **sequence variable**, and refer to the former variables $y_j$ as derived objects;
- and we impose bag arguments to be ultimately empty, considering only sequences of bags of the shape $\langle\bar{n}_1,\dots\bar{n}_l,[\,],[\,],\dots\rangle$, that we will call **streams**.

An extensional resource term will then be $\lambda\vec{y}.m\,\vec{n}$ where, inductively, $m$ is either an ordinary variable or a term itself, and $\vec{n}$ is a stream of terms. Note that we obtain for free that a stream of terms ultimately contains approximations of successive abstracted variables, just

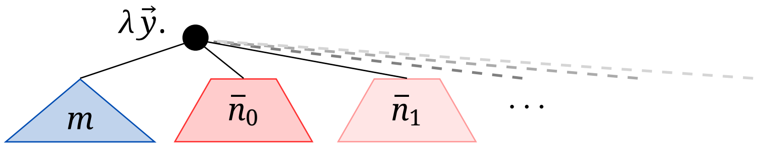

**Figure 7.** Shape of an extensional resource term

because empty bags satisfy this condition! We may again depict such a term as in Figure 7, where $\bar{n}_i$ is empty for every sufficiently large $i$.

After preliminary definitions in Section 2, we detail the syntax of this extensional resource calculus in Section 3. We equip it with a reduction derived from that of the ordinary resource calculus. In particular, one can simultaneously fire the countable sequence of redexes at the head of an expression like $(\lambda\vec{x}.m)\,\vec{n}$ in a single *full step*, to obtain a finite sum of strictly smaller terms: this process itself is essentially finite, because the induced sequence of resource substitutions is ultimately effectless – replacing non-occurring variables with the elements of empty bags. The obtained dynamics retains essential properties of resource reduction: it is confluent in a strong sense, and the size of terms is non-increasing under reduction, and even strictly decreasing for full steps. In particular, each term reduces to a unique normal form, which is a finite sum.

**Extensional Taylor expansion.** Then we turn our attention to vectors of extensional resource terms in Section 4, and show that ordinary substitution can be obtained as the composition of resource substitution and promotion, establishing an analogue of Equation (1) for arbitrary vectors (instead of Taylor expansions only). We moreover extend resource reduction to vectors, and show that it is compatible with promotion.

We leverage these results in Section 5, where we define an extensional version of Taylor expansion, mapping ordinary $\lambda$-terms to vectors of extensional resource terms, subject to the identities:

$$\mathcal{T}_\eta(x) = \lambda\vec{y}.x\,\vec{y}^{\,!} \qquad \mathcal{T}_\eta(\lambda z.M) = \lambda z.\mathcal{T}_\eta(M) \qquad \mathcal{T}_\eta(N\,P) = \lambda\vec{y}.\mathcal{T}_\eta(N)\,\mathcal{T}_\eta(P)^! :: \vec{y}^{\,!}$$

where $\vec{y}^{\,!}$ (resp. $\mathcal{T}_\eta(P)^! :: \vec{y}^{\,!}$) is the vector of streams induced by the sequence $\langle \mathcal{T}_\eta(y_0)^!, \mathcal{T}_\eta(y_1)^!, \dots\rangle$ (resp. $\langle \mathcal{T}_\eta(P)^!, \mathcal{T}_\eta(y_0)^!, \mathcal{T}_\eta(y_1)^!, \dots\rangle$). We show that this extensional Taylor expansion also enjoys a version of Equation (1), although as a reduction rather than as an identity – this is analogous to the fact that, even in a typed setting, the terms $M^\eta\{N^\eta/x\}$ and $(M\{N/x\})^\eta$ might differ, but the former $\beta$-reduces to the latter. This allows us to simulate both $\beta$- and $\eta$-reduction.

**A characterization of $\mathsf{H}^*$.** Given the constructions we have outlined, one can reasonably consider the extensional resource calculus as a language of (non-necessarily normal) finite approximants of Nakajima trees, much like ordinary resource terms for Böhm trees. We are indeed confident that an analogue of Equation (2), where one replaces ordinary Taylor expansion with extensional Taylor expansion, and Böhm trees with Nakajima trees, could be established. But our point is precisely that the technicalities of dealing with infinite $\eta$-expansion in the already infinite Böhm trees can be avoided, and that this kind of technology can profitably be replaced with Taylor expansion.

In support of this claim, we characterize the $\lambda$-theory $=_{\mathcal{T}_\eta}$ induced by the normalization of extensional Taylor expansion, in Section 6. That $=_{\mathcal{T}_\eta}$ is indeed a $\lambda$-theory follows from the inductive definition of Taylor expansion and the simulation of $\beta$-reduction, like in the ordinary case. It is moreover extensional, thanks to the simulation of $\eta$-reduction, and sensible, thanks to a variant of Lemma 1.4, that we establish essentially in the same way – although, like for substitution, extensional Taylor expansion does not commute with head reduction on the nose.

Finally, we show that $=_{\mathcal{T}_\eta}$ is nothing but $\mathbf{H}^*$, the greatest consistent sensible $\lambda$-theory. The proof is naturally based on a separability argument, showing that $=_{\mathcal{T}_\eta}$-distinct terms can be separated by a context, sending one of them to a head normalizable term, and the other one to a non-solvable one. Thanks to the properties of Taylor expansion, we are able to reason on the structure of resource terms, which allows us to adapt a well-known proof of separability for $\eta$-distinct $\beta$-normal forms [28, Chapter 5]: that we can do so, instead of having to reason on infinite objects, is a testimony of the applicability of extensional Taylor expansion.

This characterization moreover allows us to revisit previous results about $\mathbf{H}^*$ – or Nakajima trees, which are canonical representatives for $\mathbf{H}^*$ [2, Exercise 19.4.4]. E.g., to exhibit a model of $\mathbf{H}^*$, it becomes sufficient to provide a model of the extensional resource calculus. Illustrating this strategy, Section 7 gives a new proof of a result by Manzonetto [30]: $\mathbf{H}^*$ is the $\lambda$-theory induced by a well-chosen reflexive object $\mathcal{D}$ in the relational model of the simply typed $\lambda$-calculus [8].

**Where this all comes from: Taylor expansion and game semantics.** The present work was actually motivated by an ongoing effort to expose the close connections between Taylor expansion and game semantics. In a typed setting, $\eta$-long, $\beta$-normal resource terms were known to be in bijective correspondence with plays in the sense of Hyland-Ong game semantics [24], up to Opponent's scheduling of the independent explorations of separate branches of the term, as formalized by Melliès' homotopy equivalence on plays [33]: this correspondence was first unveiled by Tsukada and Ong [38] via two bijections with particular elements of the relational model of the simply typed $\lambda$-calculus; and we later exhibited a direct correspondence, underlying a quantitative denotational interpretation of (non-necessarily normal) resource terms as strategies [5].

To recast this correspondence in an untyped setting, we needed an untyped analogue of $\eta$-long, $\beta$-normal resource terms: these are the normal forms of our extensional resource calculus. The first sections of the paper make no reference to game semantics, as we focus on developing the theory of extensional Taylor expansion, and its applications. Nonetheless, we dedicate Section 8 to spelling out the bijection between normal extensional resource terms and (isomorphism classes of) augmentations in the universal arena: augmentations were introduced by the first two authors [4], as an alternative presentation of plays up to homotopy; and the universal arena is the standard interpretation of the “type” of pure, untyped $\lambda$-terms in game semantics [26].[9]

We do not go beyond this static correspondence: as discussed in our concluding Section 9, we leave for future work the definition of a denotational semantics of extensional resource terms as strategies, as well as its relationship with Taylor expansion. In passing, we also establish a correspondence between *positions* of the universal arena, which represent states of computation (a position essentially records the part of the arena that is explored by a play), and *relational types*, *i.e.* the elements of the reflexive object $\mathcal{D}$. We moreover show that, through both correspondences, the position reached by an augmentation is nothing but the (uniquely defined) relational type of the associated normal term.

**Comparison with the resource calculus with tests.** It is notable that, in the introduction of their seminal paper [38], Tsukada and Ong claimed that their results could be adapted to the untyped setting, relying on the *resource calculus with tests* of Bucciarelli, Carraro, Ehrhard and Manzonetto [7]. The latter is an extension of the ordinary resource calculus designed to associate a syntactic counterpart to *every* point of $\mathcal{D}$: from this, the authors derive a full abstraction result for the resource calculus with tests, that they are able to lift to a version with promotion (itself an extension of the differential $\lambda$-calculus [18]). Tsukada and Ong’s claim was prompted by the fact that this calculus provides constructions both for applying a term to a denumerable sequence of empty bags (the *cork* constructor $\tau m$, yielding a *test*), and for abstracting over a denumerable sequence of fresh variables (the dual *uncork* constructor $\bar{\tau} a$, where $a$ is a test, yielding a term).

It turns out, however, that this calculus is not readily fit for the task: the original version of its language is too rich (it contains normal forms that do not correspond to plays) so it must be restricted; and at the same time its constructions for infinite sequences of abstractions, and for applications to infinite sequences of bags are not canonical. As a consequence, even though one can devise an appropriate notion of $\eta$-longness in that setting,[10] the syntax still distinguishes between normal forms that represent the same play up to homotopy. For the same reasons, the resource calculus with tests is not an appropriate target language for extensional Taylor expansion. For the sake of comparison, we provide a brief account of the resource calculus with tests in Section 3.4, in light of the key features of the extensional resource calculus. In particular, we describe an $\eta$-long fragment of the former and outline how one could recover the latter as a quotient.

---

[9] Notably, Ker *et al.* [26] have shown that this arena provides an extensional reflexive object in the category of innocent strategies, and that the induced interpretation of untyped $\lambda$-terms enjoys a strong connection with Nakajima trees. In particular, its $\lambda$-theory is also $\mathbf{H}^*$.

[10] To our knowledge, such a notion remained to be introduced before our own work: Bucciarelli *et al.* did not discuss extensionality nor $\eta$-longness in the context of their calculus, as their interest was elsewhere.

**How to read this paper.** The paper is long, but it is structured in such a way that the reader can browse through a selection of its content depending on their interests.

A programming language semanticist who is already convinced of the merits of the Taylor expansion might be content as early as after reaching the end of Section 5, where we show that extensional Taylor expansion is compatible with both $\beta$- and $\eta$-reductions. A $\lambda$-calculist in search of a practical alternative to Nakajima trees will want to continue with Section 6, and might also read Section 7 for an example of application. Both profiles can dispense with Section 8 altogether. In any case, we discourage the reader without any background in game semantics to discover the subject with the present paper: an introductory account of game semantics in relation to Taylor expansion can rather be found in the above-cited works [38, 5]. On the other hand, a game semanticist might want to jump to Section 8 just after getting acquainted with the extensional resource calculus in Section 3 (the correspondence between positions and relational types additionally refers to the relational model of the extensional resource calculus given Section 7.2, which can be read independently from Sections 4 to 6, as well as from the rest of Section 7). We summarize those possible routes through the content of the paper in Figure 8.

All along Sections 3 to 6, we maintain our bias in favour of the quantitative version of Taylor expansion. We have already explained why we consider quantitative Taylor expansion as the primitive notion, of which the qualitative version is a mere by-product – *qui peut le plus, peut le moins*. The coefficients of Taylor expansion, as well as those generated by resource reduction, are moreover relevant in connection with game semantics: the correspondence between terms and strategies is quantitative [5]! Nonetheless, for the results we develop in Sections 6 and 7, the qualitative version is sufficient; and Section 8 involves normal terms only, and does not mention resource reduction nor Taylor expansion. A reader interested only in those applications might thus read the earlier sections without worrying too much about coefficients, and even skip some computations.

## 2. Preliminaries on sequences and bags

**Tuples and bags.** If $X$ is a set, we write $X^* = \bigcup_{n\in\mathbb{N}} X^n$ for the set of finite lists, or tuples, of elements of $X$, ranged over by $\vec{a}, \vec{b}$, *etc.* We write $\langle a_1, \dots, a_n\rangle = \langle a_i\rangle_{1\leq i\leq n}$ to list the elements of a tuple, $\varepsilon$ for the empty tuple, $|\vec{a}|$ for the length of $\vec{a}$, and denote concatenation simply by juxtaposition, e.g., $\vec{a}\,\vec{b}$. If $a \in X$ and $\vec{b}$ is a tuple, we write $a :: \vec{b}$ for the tuple obtained by pushing $a$ at the head of $\vec{b}$: this **cons** operation generates $X^*$ inductively from $\varepsilon$.

We write $\mathfrak{M}_\mathrm{f}(X)$ for the set of finite multisets of elements of $X$, which we call **bags**, ranged over by $\bar{a}, \bar{b}$, *etc.* We write $[a_1, \dots, a_n]$ for the bag $\bar{a}$ defined by a list $\vec{a} = \langle a_1, \dots, a_n\rangle$ of elements: we say that $\vec{a}$ is an **enumeration** of $\bar{a}$ in this case. We write $[\,]$ for the empty bag, and use $*$ for bag concatenation. We moreover write $|\bar{a}|$ for the length of $\bar{a}$: $|\bar{a}|$ is the length of any

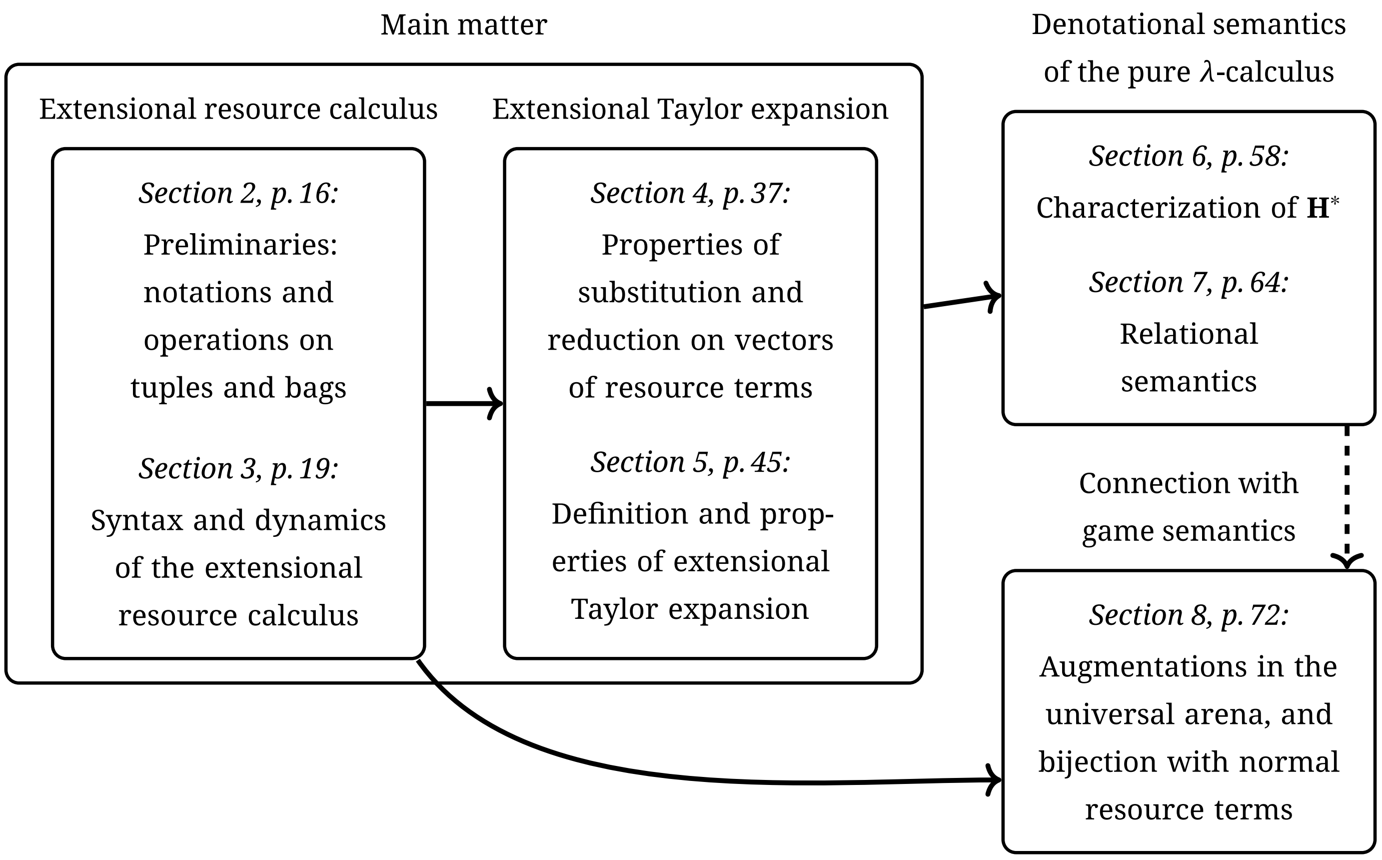

**Figure 8.** A map of the contributions of the paper. Arrows represent dependencies; the dashed arrow stands for the dependency of some results of Section 8 on the definition of relational typing in Section 7.2.

enumeration of $\bar{a}$. We may abuse notation and use a tuple $\vec{a}$ or a bag $\bar{a}$ for the set of its elements: e.g., we write $a_i \in [a_1, \dots, a_n]$.

We will often need to *partition* bags, which requires some care. For $k \in \mathbb{N}$, a **$k$-partitioning** of $\bar{a}$ is a function $p : \{1, \dots, |\bar{a}|\} \longrightarrow \{1, \dots, k\}$: we write $p : \bar{a} \lhd k$. Given an enumeration $\langle a_1, \dots, a_n \rangle$ of $\bar{a}$ and $J = \{j_1, \dots, j_l\} \subseteq \{1, \dots, n\}$ with $\#J = l$, we write $\bar{a} \upharpoonright J := [a_{j_i}, \dots, a_{j_l}]$ for the **restriction** of $\bar{a}$ to $J$. The **$k$-partition** of $\bar{a}$ associated with $p : \bar{a} \lhd k$ is then the tuple $\langle \bar{a} \upharpoonright p^{-1}(1), \dots, \bar{a} \upharpoonright p^{-1}(k) \rangle$, where $p^{-1}(i) := \{j \mid p(j) = i\}$ for $1 \leq i \leq k$, so that

$$\bar{a} = \bar{a} \upharpoonright p^{-1}(1) * \cdots * \bar{a} \upharpoonright p^{-1}(k) \ .$$

There is a (temporary) abuse of notation here, as the definitions of restrictions and $k$-partitions depend on the chosen enumeration of $\bar{a}$. But having fixed $\bar{a}$ and $k$, neither the set of $k$-partitions of $\bar{a}$, nor the *number* of partitionings $p$ of $\bar{a}$ yielding a given $\langle \bar{a}_1, \dots, \bar{a}_n \rangle$, depend on the enumeration. So for any function $f : \mathfrak{M}_{\mathrm{f}}(X)^k \to \mathcal{M}$ (for $\mathcal{M}$ a commutative monoid, noted additively), the sum

$$\sum_{\bar{a} \lhd \bar{a}_1 * \cdots * \bar{a}_k} f(\bar{a}_1, \dots, \bar{a}_k) \quad := \quad \sum_{p : \bar{a} \lhd k} f(\bar{a} \upharpoonright p^{-1}(1), \dots, \bar{a} \upharpoonright p^{-1}(k))$$

is independent of the enumeration. When indexing a sum with $\bar{a} \lhd \bar{a}_1 * \cdots * \bar{a}_k$ we thus mean to sum over all partitionings $p : \bar{a} \lhd k$, $\bar{a}_i$ being shorthand for $\bar{a} \restriction p^{-1}(i)$ in the summand, and the result being independent of the choice of an enumeration. This construction is easily proved to be associative, in the sense that, e.g.:

$$\sum_{\bar{a} \lhd \bar{a}_1 * \bar{a}'} \; \sum_{\bar{a}' \lhd \bar{a}_2 * \bar{a}_3} f(\bar{a}_1, \bar{a}_2, \bar{a}_3) \qquad = \qquad \sum_{\bar{a} \lhd \bar{a}_1 * \bar{a}_2 * \bar{a}_3} f(\bar{a}_1, \bar{a}_2, \bar{a}_3)\ .$$

The **isotropy degree** $\mathsf{d}(\bar{a})$ of a bag $\bar{a}$ of length $k$ is the cardinality of the stabilizer of any enumeration $\langle a_1, \dots, a_k \rangle$ of $\bar{a}$ under the action of the group $\mathbb{S}_k$ of permutations of $\{1, \dots, k\}$: namely, $\mathsf{d}(\bar{a}) := \#\{\sigma \in \mathbb{S}_k \mid \langle a_1, \dots, a_k \rangle = \langle a_{\sigma(1)}, \dots, a_{\sigma(k)} \rangle\}$. The following result is a routine exercise in combinatorics:

**FACT 2.1.** *If* $\bar{a} = \bar{a}_1 * \cdots * \bar{a}_n$ *then* $\mathsf{d}(\bar{a}) = \#\{p : \bar{a} \lhd n \mid \bar{a} \restriction p^{-1}(i) = \bar{a}_i \text{ for } 1 \leq i \leq n\} \times \prod_{i=1}^{n} \mathsf{d}(\bar{a}_i)$.

**Sequences of bags and streams.** We will also use possibly infinite sequences of bags, with a finiteness constraint: only finitely many bags may be non-empty. We write $\mathcal{S}_{\mathrm{f}}(X)$ for the set $\mathfrak{M}_{\mathrm{f}}(X)^*$ of tuples of bags, and we write $\mathcal{S}(X)$ for the subset of $\mathfrak{M}_{\mathrm{f}}(X)^{\mathbb{N}}$ such that $\langle \bar{a}_i \rangle_{i \in \mathbb{N}} \in \mathcal{S}(X)$ iff $\{i \in \mathbb{N} \mid |\bar{a}_i| > 0\}$ is finite. We denote elements of $\mathcal{S}_{\mathrm{f}}(X)$ or $\mathcal{S}(X)$ as $\vec{a}, \vec{b}$, *etc.* just like for plain tuples, and we reserve the name **stream** for the elements of $\mathcal{S}(X)$.

We write $\iota := \langle [\,] \rangle_{i \in \mathbb{N}}$ for the **empty stream**. Note that streams are inductively generated from $\iota$, by the **cons** operation defined by

$$(\bar{a} :: \vec{b})_i := \begin{cases} \bar{a} & \text{if } i = 0 \\ \bar{b}_j & \text{if } i = j + 1 \end{cases} \qquad (\text{writing } \vec{b} = \langle \bar{b}_j \rangle_{j \in \mathbb{N}})$$

subject to the identity $[\,] :: \iota = \iota$. We can thus reason inductively on streams, treating $\iota$ as the base case, and considering $\vec{b}$ as a “strict sub-stream” of $\bar{a} :: \vec{b}$ when $\bar{a} :: \vec{b} \neq \iota$.

We also define the **range** of a stream $\vec{a} = \langle \bar{a}_i \rangle_{i \in \mathbb{N}} \in \mathcal{S}_{\mathrm{f}}(A)$ as the minimal length of a prefix containing all non-empty bags: $\mathsf{r}(\vec{a}) := \max\{i + 1 \in \mathbb{N} \mid \bar{a}_i \neq 0\}$. Equivalently, we can define $\mathsf{r}$ inductively by setting: $\mathsf{r}(\iota) := 0$ and $\mathsf{r}(\bar{a} :: \vec{b}) := \mathsf{r}(\vec{b}) + 1$ if $\bar{a} :: \vec{b} \neq \iota$.

A $k$-**partitioning** $p : \vec{a} \lhd k$ of $\vec{a} = \langle \bar{a}_1, \dots, \bar{a}_n \rangle \in \mathcal{S}_{\mathrm{f}}(X)$ is a tuple $p = \langle p_1, \dots, p_n \rangle$ of $k$-partitionings $p_i : \bar{a}_i \lhd k$. This defines a **partition** $\langle \vec{a} \restriction p^{-1}(1), \dots, \vec{a} \restriction p^{-1}(k) \rangle$, component-wise: each $\vec{a} \restriction p^{-1}(i)$ is the sequence $\langle \bar{a}_1 \restriction p_1^{-1}(i), \dots, \bar{a}_n \restriction p_n^{-1}(i) \rangle$. We obtain $\vec{a} = \vec{a} \restriction p^{-1}(1) * \cdots * \vec{a} \restriction p^{-1}(k)$, where we apply the concatenation of bags component-wise, to sequences all of the same length. Just as before, we write

$$\sum_{\vec{a} \lhd \vec{a}_1 * \cdots * \vec{a}_k} f(\vec{a}_1, \dots, \vec{a}_k) := \sum_{p : \vec{a} \lhd k} f(\vec{a} \restriction p^{-1}(1), \dots, \vec{a} \restriction p^{-1}(k))\ ,$$

the result of the sum being independent from the enumerations of the bags of $\vec{a}$.

Similarly, a $k$-**partitioning** $p : \vec{a} \lhd k$ of a stream $\vec{a} = \langle \bar{a}_i \rangle_{i \in \mathbb{N}}$ is a sequence $p = \langle p_i \rangle_{i \in \mathbb{N}}$ of $k$-partitionings $p_i : \bar{a}_i \lhd k$: note that a stream $\vec{a}$ has only finitely many $k$-partitionings, because $\bar{a}_i$ is empty for sufficiently large values of $i$. A $k$-partitioning of a stream $\vec{a}$ defines a **partition** $\langle \vec{a} \upharpoonright p^{-1}(1), \dots, \vec{a} \upharpoonright p^{-1}(k) \rangle$, component-wise: each $\vec{a} \upharpoonright p^{-1}(j)$ is the sequence $\langle \bar{a}_i \upharpoonright p_i^{-1}(j) \rangle_{i \in \mathbb{N}}$. We obtain $\vec{a} = \vec{a} \upharpoonright p^{-1}(1) * \cdots * \vec{a} \upharpoonright p^{-1}(k)$, where we apply the concatenation of bags component-wise. And we write

$$\sum_{\vec{a} \lhd \vec{a}_1 * \cdots * \vec{a}_k} f(\vec{a}_1, \dots, \vec{a}_k) := \sum_{p : \vec{a} \lhd k} f(\vec{a} \upharpoonright p^{-1}(1), \dots, \vec{a} \upharpoonright p^{-1}(k)) ,$$

which is always a finite sum, whose result is independent from the enumerations of the bags of $\vec{a}$.

## 3. The extensional resource calculus

In this section, we introduce our extensional version of the resource calculus, whose terms are the infinitely $\eta$-long resource terms described in the introduction: $\lambda \vec{x}.e\, \vec{n}$ where $e$ is a term or variable and $\vec{n}$ is a stream of terms. It will be practical to more generally introduce various syntactic categories, such as *base terms* of the shape $e\, \vec{n}$, as in the body of the previous term. We will collectively refer to these categories as *resource terms*, calling *value terms* those of the first form.

We then discuss suitable notions of resource reduction, taking into account the presence of infinite sequences of abstractions, applied to streams of arguments.

### 3.1 Syntax of the calculus

We fix an infinite countable set $\mathcal{V}$ of **value variables** (or, simply, **variables**), which we denote by letters $x, y, z$. We also fix an infinite countable set $\mathcal{V}_\mathrm{s}$ of **sequence variables**, which we denote by letters $\vec{x}, \vec{y}, \vec{z}$, and with each sequence variable $\vec{x}$, we associate a sequence $\langle \vec{x}(i) \rangle_{i \in \mathbb{N}}$ of value variables, in such a way that for each $x \in \mathcal{V}$, there exists a unique pair $\langle \vec{x}, i \rangle$ such that $x = \vec{x}(i)$: sequence variables partition value variables. We will in general identify $\vec{x}$ with the corresponding sequence of value variables. We may also abuse notation and use $\vec{x}$ for its image set: for instance we may write $x \in \vec{x}$ instead of $x \in \{\vec{x}(i) \mid i \in \mathbb{N}\}$. The use of sequence variables will allow us to manage infinite sequences of $\lambda$-abstractions, without needing to resort to de Bruijn indices or other techniques for dealing with $\alpha$-equivalence.

**Terms.** We define **value terms** ($m, n, p \in \Delta_\mathrm{v}$), **base terms** ($a, b, c \in \Delta_\mathrm{b}$), **bag terms** ($\bar{m}, \bar{n}, \bar{p} \in \Delta_!$) and **stream terms** ($\vec{m}, \vec{n}, \vec{p} \in \Delta_\mathrm{s}$), inductively by the rules of Figure 9.[11] A **head expression** ($e, f, g \in \Delta_\mathrm{h}$) is a value term or variable, so that a base term is necessarily of the form $e\, \vec{m}$ where $e$ is a head expression and $\vec{m}$ is a stream term. A **resource term** (denoted by $u, v, w$) is any of a

---

[11] For now, we overload notations and use $[-]$, $\iota$ and $::$ as term formers: they will soon recover their meaning as constructions of bags and streams.

$$\frac{\vec{x} \in \mathcal{V}_{\mathrm{s}} \qquad a \in \Delta_{\mathrm{b}}}{\lambda \vec{x}.a \in \Delta_{\mathrm{v}}}\,(\lambda) \qquad \frac{m \in \Delta_{\mathrm{v}} \qquad \vec{n} \in \Delta_{\mathrm{s}}}{m\,\vec{n} \in \Delta_{\mathrm{b}}}\,(app) \qquad \frac{x \in \mathcal{V} \qquad \vec{n} \in \Delta_{\mathrm{s}}}{x\,\vec{n} \in \Delta_{\mathrm{b}}}\,(\mathcal{V})$$

$$\frac{m_1 \in \Delta_{\mathrm{v}} \qquad \cdots \qquad m_k \in \Delta_{\mathrm{v}}}{[m_1, \dots, m_k] \in \Delta_!}\,(!) \qquad \frac{}{\iota \in \Delta_{\mathrm{s}}}\,(\iota) \qquad \frac{\bar{m} \in \Delta_! \qquad \vec{n} \in \Delta_{\mathrm{s}}}{\bar{m} :: \vec{n} \in \Delta_{\mathrm{s}}}\,(::)$$

**Figure 9.** Rules for constructing extensional resource terms

value term, base term, stream term or bag term; and a **resource expression** (denoted by $q, r, s$) is any of a value variable or a resource term. Note that, despite the seemingly infinitary flavour of abstractions and applications, the inductive definition of the syntax yields a countable set of resource expressions.

The actual objects of the calculus are value terms, as these will form the target of Taylor expansion. Nonetheless, base terms, bag terms and stream terms also constitute meaningful computational entities, as will be clear in the next sections. On the other hand, plain variables *should not* be considered as entities of the calculus by themselves: in the context of extensional Taylor expansion, a variable must always come with the stream it is applied to, and the Taylor expansion of a single variable will always be an infinite sum of normal value terms (see Section 5).[12] We consider head expressions only because it allows us to treat uniformly both forms of base terms; and we consider resource expressions because, when we reason inductively on terms, it is sometimes simpler to have variables as a base case. We may write $\Delta_{\mathrm{t}}$ (resp. $\Delta_{\mathrm{e}}$) for any of the four sets $\Delta_{\mathrm{v}}$, $\Delta_{\mathrm{b}}$, $\Delta_!$, or $\Delta_{\mathrm{s}}$ (resp. $\Delta_{\mathrm{h}}$, $\Delta_{\mathrm{b}}$, $\Delta_!$, or $\Delta_{\mathrm{s}}$).[13]

The set of **free variables** of a resource expression is defined as follows:

$$\mathcal{V}(x) := \{x\}\,, \qquad \mathcal{V}(\lambda\vec{x}.a) := \mathcal{V}(a) \setminus \vec{x}\,, \qquad \mathcal{V}(e\,\vec{m}) := \mathcal{V}(e) \cup \mathcal{V}(\vec{m})\,,$$
$$\mathcal{V}([m_1, \dots, m_k]) := \bigcup_{1 \le i \le k} \mathcal{V}(m_i)\,, \qquad \mathcal{V}(\iota) := \emptyset\,, \qquad \mathcal{V}(\bar{m} :: \vec{n}) := \mathcal{V}(\bar{m}) \cup \mathcal{V}(\vec{n})\,.$$

This is always a finite set. Then we define $\mathcal{V}_{\mathrm{s}}(u) := \bigcup_{x \in \mathcal{V}(u)} \mathcal{V}_{\mathrm{s}}(x)$ with $\mathcal{V}_{\mathrm{s}}(\vec{x}(i)) := \{\vec{x}\}$, which is also finite. We can thus define $\alpha$-equivalence as usual, despite the fact that binding a single sequence variable simultaneously binds an infinite family of value variables. In particular, given a value term $m$ and a finite set $V$ of sequence variables that are not free in $m$, we can always assume, up to $\alpha$-equivalence, that $m$ is of the form $\lambda\vec{y}.a$ where $\vec{y}$ contains no variable $\vec{x}(i)$ with $\vec{x} \in V$.

---

[12] This distinction also shows in the semantics. The various categories of resource terms will have a finite interpretation in the relational semantics (Section 7). And the normal forms among each category will represent isomorphism classes of augmentations on corresponding arenas in game semantics (Section 8). On the other hand, the relational interpretation of a variable is an infinite set; and, in game semantics, it corresponds to a whole *copycat* strategy.

[13] We do mean these as placeholders, not as unions of sets: this will be especially relevant when we consider, e.g., sums of resource terms, which we always implicitly restrict to a given syntactic category.

In addition to $\alpha$-equivalence, we consider resource expressions up to permutations of elements in a bag $[m_1, \ldots, m_k]$, and up to the identity $[\,] :: \iota = \iota$, so that $\Delta_!$ is identified with $\mathfrak{M}_{\mathrm{f}}(\Delta_{\mathrm{v}})$ and $\Delta_{\mathrm{s}}$ is identified with $\mathcal{S}(\Delta_{\mathrm{v}})$.

We write $q\{f/x\}$ for the ordinarily defined, capture avoiding **substitution** of a head expression $f$ for a value variable $x$ in any resource expression $q$: note that this preserves the syntactic category of $q$ in the sense that if $q \in \Delta_{\mathrm{v}}$ (resp. $\Delta_{\mathrm{h}}$, $\Delta_!$, $\Delta_{\mathrm{b}}$, $\Delta_{\mathrm{s}}$) then $q\{f/x\} \in \Delta_{\mathrm{v}}$ (resp. $\Delta_{\mathrm{h}}$, $\Delta_!$, $\Delta_{\mathrm{b}}$, $\Delta_{\mathrm{s}}$).

We define the **size** $\#q \in \mathbb{N}$ of a resource expression $q$ inductively as follows:

$$\#x := 1 \qquad \#(\lambda\vec{x}.a) := 1 + \#a \qquad \#(e\,\vec{m}) := 1 + \#e + \#\vec{m}$$

$$\#[m_1, \ldots, m_k] := \sum_{i=1}^{k} \#m_i \qquad \#(\bar{m} :: \vec{n}) := \begin{cases} 0 & \text{if } \bar{m} :: \vec{n} = \iota \\ \#\bar{m} + \#\vec{n} & \text{otherwise} \end{cases} .$$

In particular, $\#\vec{m} = \sum_{i \in \mathbb{N}} \#\bar{m}_i$ for any stream term $\vec{m} = \langle \bar{m}_i \rangle_{i \in \mathbb{N}}$. In short, $\#q$ is nothing but the number of abstractions, applications and variable occurrences in $q$: this number is always finite, even though each abstraction on a sequence variable binds an infinite sequence of value variables, and each application is to a stream of terms, which is an infinite sequence of bags. In particular, $\#e \geq 1$ for any head expression $e$, $\#a \geq 2$ for any base term $a$, $\#m \geq 3$ for any value term $m$, and $\#\bar{m} \geq 3|\bar{m}|$ for any bag term $\bar{m}$.

We similarly define the **number of occurrences** $|q|_x$ of $x \in \mathcal{V}$ in $q$ inductively as follows:

$$|x|_x := 1 \qquad |y|_x := 0 \text{ (for each } y \neq x) \qquad |\lambda\vec{x}.a|_x := |a|_x \qquad |e\,\vec{m}|_x := |e|_x + |\vec{m}|_x$$

$$|[m_1, \ldots, m_k]|_x := \sum_{i=1}^{k} |m_i|_x \qquad |\bar{m} :: \vec{n}|_x := \begin{cases} 0 & \text{if } \bar{m} :: \vec{n} = \iota \\ |\bar{m}|_x + |\vec{n}|_x & \text{otherwise} \end{cases}$$

(choosing $\vec{x} \not\ni x$ in the abstraction case) and obtain that $|q|_x \leq \#q$, and that $|q|_x = 0$ iff $x \notin \mathcal{V}(q)$, by straightforward inductions.

**EXAMPLE 3.1.** Let us review the smallest (in terms of the size just defined, as well as in the sense of inductive constructions) terms we can consider in each syntactic category. The smallest bag term (resp. stream term) is of course the empty bag $[\,]$ (resp. the empty stream $\iota$). Given a variable $x$, we can then form the base term $x\,\iota$. Abstracting over a sequence variable $\vec{y}$, we obtain the closed value term $\mathsf{p}_i := \lambda\vec{y}.\vec{y}(i)\,\iota$ if $x = \vec{y}(i)$, or the value term $\mathsf{p}_x := \lambda\vec{y}.x\,\iota$ where $x$ occurs free if $\vec{y} \not\ni x$.

Then one can form more elaborate terms such as

$$\mathsf{c}_x = \lambda\vec{y}.x\,[\mathsf{p}_{\vec{y}(0)}] :: \iota \qquad \text{and} \qquad \mathsf{c}'_x = \lambda\vec{y}.x\,[\mathsf{c}_{\vec{y}(0)}, \mathsf{c}_{\vec{y}(0)}] :: \iota$$

for each $x \in \mathcal{V}$, that we will use as running examples below. ◆

**Sums of Terms.** As in the ordinary resource calculus, the reduction of resource terms produces sums: a value (resp. base, bag, stream) term will reduce to a finite sum of value (resp. base, bag, stream) terms.

If $X$ is a set, we write $\Sigma X$ for the set of finite formal sums of elements $X$ – those may be equivalently presented as finite multisets, but we adopt a distinct notation if only to impose a distinction with bag terms. Given a sum $A = \sum_{i \in I} a_i$, we write $\mathrm{supp}(A) := \{a_i \mid i \in I\}$ for its **support** set. We may abuse notation and write $a \in A$ instead of $a \in \mathrm{supp}(A)$.

We call **value sums** (resp. **base sums**, **bag sums**, **stream sums**) the elements of $\Sigma\Delta_{\mathrm{v}}$ (resp. $\Sigma\Delta_{\mathrm{b}}$, $\Sigma\Delta_{!}$, $\Sigma\Delta_{\mathrm{s}}$), which we denote with capital letters $M, N, P$ (resp. $A, B, C$; $\bar{M}, \bar{N}, \bar{P}$; $\vec{M}, \vec{N}, \vec{P}$). As announced we may write $\Sigma\Delta_{\mathrm{t}}$ for any of $\Sigma\Delta_{\mathrm{v}}$, $\Sigma\Delta_{\mathrm{b}}$, $\Sigma\Delta_{!}$, or $\Sigma\Delta_{\mathrm{s}}$; and we may call **term sum** any value sum, base sum, bag sum, or stream sum, which we then denote by $U, V, W$.

We also call **head sum** (resp. **resource sum**) any of a value sum (resp. term sum) or of a value variable, which we denote by $E, F, G$ (resp. by $Q, R, S$). We abuse notation and write $\Sigma\Delta_{\mathrm{h}}$ for $\mathcal{V} \cup \Sigma\Delta_{\mathrm{v}}$, and then write $\Sigma\Delta_{\mathrm{e}}$ for any of $\Sigma\Delta_{\mathrm{h}}$, $\Sigma\Delta_{\mathrm{b}}$, $\Sigma\Delta_{!}$, or $\Sigma\Delta_{\mathrm{s}}$. Again, we introduce head sums and expression sums only as technical devices allowing us to simplify some definitions or proofs. Note that we do not need to consider sums of head expressions mixing value terms and variables.

We then extend term formers and operations to all syntactic categories by linearity so that, for any finite sets $I$ and $J$, we have:

$$\lambda\vec{x}.\Big(\sum_{i \in I} a_i\Big) = \sum_{i \in I} \lambda\vec{x}.a_i \qquad \Big(\sum_{i \in I} m_i\Big)\,\vec{N} = \sum_{i \in I} m_i\,\vec{N} \qquad e\,\Big(\sum_{j \in J} \vec{n}_j\Big) = \sum_{j \in J} e\,\vec{n}_j$$

$$\Big[\sum_{i \in I} m_i\Big] = \sum_{i \in I} [m_i] \qquad \Big(\sum_{i \in I} \bar{m}_i\Big) * \Big(\sum_{j \in J} \bar{n}_j\Big) = \sum_{\langle i,j \rangle \in I \times J} \bar{m}_i * \bar{n}_j$$

$$\Big(\sum_{i \in I} \bar{m}_i\Big) :: \Big(\sum_{j \in J} \vec{n}_j\Big) = \sum_{\langle i,j \rangle \in I \times J} \bar{m}_i :: \vec{n}_j \,.$$

We can thus extend ordinary substitution to sums of resource expressions: $Q\{U/x\}$ is obtained by substituting $U$ for $x$ in each summand of $Q$, up to the previous identities. Note this operation is linear in $Q$ but not in $U$. In particular, $q\{0/x\} = 0$ if $x \in \mathcal{V}(q)$, and $q\{0/x\} = q$ otherwise.

**Two flavours of extensional resource reduction.** A redex in the extensional resource calculus is naturally a base term of the shape $(\lambda\vec{x}.a)\,\vec{n}$. Intuitively, this involves an infinite sequence of abstractions, over the family of value variables $\langle \vec{x}(i) \rangle_{i \in \mathbb{N}}$, applied to the sequence of bags in the stream $\vec{n} = \langle \bar{n}_i \rangle_{i \in \mathbb{N}}$. One way to reduce this redex is to apply what we will call a *full step*: this yields the base sum $a[\vec{n}/\vec{x}]$ obtained from the body $a$ by the resource substitution of each bag $\bar{n}_i$ for the corresponding variable $\vec{x}(i)$. As we have argued in the introduction, although this seems to involve infinitely many simultaneous substitutions, only a finite number of them have an effect: for a sufficiently large $i$, $\vec{x}(i)$ does not occur in $a$, and $\bar{n}_i$ is empty.

Moreover, full-step reduction does not allow us to single out the application of the first abstraction in the sequence to the first bag in the stream, which would be the analogue of a redex in the $\lambda$-calculus, via Taylor expansion.

Writing $\lambda\vec{x}.a = \lambda x.m$ and $\vec{n} = \bar{n} :: \vec{p}$, we will therefore also consider the *fine step* reducing $(\lambda\vec{x}.a)\,\vec{n}$ to $m[\bar{n}/x]\,\vec{p}$. Beware that the abstraction over $x$ is just a notation rather than a constructor in our syntax: by this we mean that we consider the value term $m = \lambda\vec{y}.b$, where $b$ is like $a$, but with $\vec{x}(0)$ renamed as $x$, and each $\vec{x}(i+1)$ renamed as $\vec{y}(i)$ – we will make this definition formal below. Also note that each term in the base sum $m[\bar{n}/x]\,\vec{p}$ is again a redex: to actually consume a redex, some form of full step is still needed. Since each stream only contains finitely many non empty bags, it is ultimately sufficient to consider a redex of the shape $(\lambda\vec{x}.a)\,\iota$, whose full-step reduction amounts to erase the whole term (replacing it with 0) in case any variable $\vec{x}(i)$ occurs in $a$. We take this as an additional case for fine-step reduction.

Both forms of resource reduction will be useful in our study of extensional Taylor expansion. We start with fine-step reduction, for which we prove confluence. Then we consider full-step reduction, for which we prove strong normalization (which fails in the fine-step case). Combining those results, we deduce that each resource term has a unique normal form, common to both reductions.

## 3.2 Fine-step dynamics

To define fine-step reduction, we need three ingredients: the *resource substitution* of a bag for a variable; a way to formalize the notation $\lambda\vec{x}.a = \lambda x.m$, which involves *shifting* the indices of the value variables $\vec{x}(i)$; and a description of the *erasure* of $\vec{x}$ performed in $a$ when reducing $(\lambda\vec{x}.a)\,\iota$.

**Resource substitution.** We consider the resource substitution $q[\bar{n}/x]$ of a bag $\bar{n} = [n_1, \ldots, n_k]$ for a value variable $x$ in a resource expression $q$. As for the ordinary resource calculus, the definition amounts to enumerate the occurrences $x_1, \ldots, x_l$ of $x$ in $q$, and then set:

$$q[\bar{n}/x] := \begin{cases} \sum_{\sigma\in\mathbb{S}_k} q\{n_1, \ldots, n_k/x_{\sigma(1)}, \ldots, x_{\sigma(k)}\} & \text{if } k = l \\ 0 & \text{otherwise} \end{cases}$$

where $\mathbb{S}_k$ is the set of all permutations of $\{1, \ldots, k\}$, and $q\{n_1, \ldots, n_k/x_1, \ldots, x_k\}$ denotes the simultaneous capture avoiding substitution of each term $m_i$ for the corresponding occurrence $x_i$.

Note that, although it is intuitively clear, this definition relies on a notion of occurrence that is not well defined because the order of elements in a bag is not fixed. To be carried out formally, one should introduce a rigid variant of the calculus, and then show that the result does not depend on the choice of a rigid representative. A more annoying issue is that this global definition does not follow the inductive structure of expressions. We will prefer the following presentation:

**DEFINITION 3.2.** We define the **resource substitution** $q[\bar{n}/x]$ of a bag term $\bar{n}$ for a value variable $x$ in a resource expression $q$ by induction on $q$ as follows:

$$\begin{aligned}
y[\bar{n}/x] &:= \begin{cases} n & \text{if } x = y \text{ and } \bar{n} = [n] \\ y & \text{if } x \neq y \text{ and } \bar{n} = [\,] \\ 0 & \text{otherwise} \end{cases} \\
(\lambda\vec{y}.a)[\bar{n}/x] &:= \lambda\vec{y}.a[\bar{n}/x] \\
(e\,\vec{m})[\bar{n}/x] &:= \sum_{\bar{n}\lhd\bar{n}_1*\bar{n}_2} e[\bar{n}_1/x]\,\vec{m}[\bar{n}_2/x] \\
[m_1,\dots,m_k][\bar{n}/x] &:= \sum_{\bar{n}\lhd\bar{n}_1*\dots*\bar{n}_k} [m_1[\bar{n}_1/x],\dots,m_k[\bar{n}_k/x]] \\
\iota[\bar{n}/x] &:= \begin{cases} \iota & \text{if } \bar{n} = [\,] \\ 0 & \text{otherwise} \end{cases} \\
(\bar{m} :: \vec{p})[\bar{n}/x] &:= \sum_{\bar{n}\lhd\bar{n}_1*\bar{n}_2} \bar{m}[\bar{n}_1/x] :: \vec{p}[\bar{n}_2/x] \qquad \text{if } \bar{m} :: \vec{p} \neq \iota
\end{aligned}$$

where, in the abstraction case, $\vec{y}$ is chosen so that $x \notin \vec{y}$ and $\vec{y} \cap \mathcal{V}(\bar{n}) = \emptyset$.

Observe that if $q$ is a variable (resp. value term, base term, bag term, stream term) then $q[\bar{n}/x]$ is a head sum (resp. value sum, base sum, bag sum, stream sum). So we may write $q[\bar{n}/x] \in \Sigma\Delta_{\mathrm{e}}$ (resp. $u[\bar{n}/x] \in \Sigma\Delta_{\mathrm{t}}$) if $q \in \Delta_{\mathrm{e}}$ (resp. $u \in \Delta_{\mathrm{t}}$) keeping implicit the fact that the underlying syntactic category is the same.

Moreover note that the distinction between empty and non-empty streams is only made so that the definition is inductive and non-ambiguous. It is nonetheless obvious that:

$$\sum_{\bar{n}\lhd\bar{n}_1*\bar{n}_2} [\,][\bar{n}_1/x] :: \iota[\bar{n}_2/x] = \begin{cases} [\,] :: \iota = \iota & \text{if } \bar{n} = [\,] \\ 0 & \text{otherwise} \end{cases}$$

so that the condition $\bar{m} :: \vec{p} \neq \iota$ can be ignored in the last case of the definition (a similar remark straightforwardly applies to the above definitions of the size of a term, and of the number of occurrences of a variable in a term).

More generally, if $\vec{m} = \bar{m}_1 :: \cdots :: \bar{m}_k :: \iota$, then

$$\vec{m}[\bar{n}/x] = \sum_{\bar{n}\lhd\bar{n}_1*\dots*\bar{n}_k} \bar{m}_1[\bar{n}_1/x] :: \cdots :: \bar{m}_k[\bar{n}_k/x] :: \iota$$

and, equivalently,

$$\langle\bar{m}_i\rangle_{i\in\mathbb{N}}[\bar{n}/x] = \sum_{p:\bar{n}\lhd\mathbb{N}} \langle\bar{m}_i[\bar{n}\upharpoonright p^{-1}(i)/x]\rangle_{i\in\mathbb{N}}$$

where we generalize $k$-partitionings to $\mathbb{N}$-partitionings in the obvious way. Also, one can check that:

$$q[[\,]/x] \quad = \quad \begin{cases} 0 & \text{if } x \in \mathcal{V}(q) \\ q & \text{otherwise} \end{cases} \quad = \quad q\{0/x\} \;.$$

And, in general, $q[\bar{n}/x] \neq 0$ iff $|q|_x = \#\bar{n}$.

**EXAMPLE 3.3.** Using the notations of Example 3.1, we obtain that $\mathsf{p}_i[[\,]/x] = \mathsf{p}_i$, and $\mathsf{p}_i[\bar{m}/x] = 0$ as soon as $\bar{m} \neq [\,]$. And we obtain $\mathsf{p}_x[[m]/x] = \lambda\vec{y}.m\,\iota$ for any value term $m$ (choosing $\vec{y} \notin \mathcal{V}_{\mathrm{s}}(m)$), and $\mathsf{p}_x[\bar{m}/x] = 0$ as soon as $|\bar{m}| \neq 1$.

Minimalistic examples yielding sums are

$$[\mathsf{p}_x, \mathsf{p}_x][[p_1, p_2]/x] = [\lambda\vec{y}.p_1\,\iota, \lambda\vec{y}.p_2\,\iota] + [\lambda\vec{y}.p_2\,\iota, \lambda\vec{y}.p_1\,\iota] = 2[\lambda\vec{y}.p_1\,\iota, \lambda\vec{y}.p_2\,\iota]$$

and

$$([\mathsf{p}_x] :: [\mathsf{p}_x] :: \iota)[[p_1, p_2]/x] = [\lambda\vec{y}.p_1\,\iota] :: [\lambda\vec{y}.p_2\,\iota] :: \iota + [\lambda\vec{y}.p_2\,\iota] :: [\lambda\vec{y}.p_1\,\iota] :: \iota$$

for any value terms $p_1$ and $p_2$ (choosing $\vec{y} \notin \mathcal{V}_{\mathrm{s}}(p_1) \cup \mathcal{V}_{\mathrm{s}}(p_2)$). ◆

Contrarily to what happens with ordinary substitution, the combinatorics of resource substitution is very regular. Indeed, except for the occurrences of the variable that is substituted, nothing is erased or discarded from the substituted bag nor from the expression in which the substitution takes place. In particular, the size of the terms produced by a substitution is determined by the length and size of the substituted bag and the size of the term in which the substitution is performed; and free variables are preserved. Formally, a straightforward induction on resource expressions yields:

**LEMMA 3.4.** *If $q' \in q[\bar{n}/x]$ then $|q|_x = |\bar{n}|$ and $\#q' = \#q + \#\bar{n} - |\bar{n}|$. If moreover $y \neq x$ then $|q'|_y = |q|_y + |\bar{n}|_y$ (in particular, $\mathcal{V}(q') = \mathcal{V}(q) \cup \mathcal{V}(\bar{n})$).*

Resource substitution is extended to sums by linearity:

$$\left(\sum_{i=1}^{k} q_i\right)\left[\sum_{j=1}^{l} \bar{m}_j/x\right] := \sum_{i=1}^{k}\sum_{j=1}^{l} q_i[\bar{m}_j/x] \;.$$

One can easily check that, with that extension, all the identities defining resource substitution in Definition 3.2 also hold if we replace terms with sums.

The usual result on the commutation of substitutions holds:

**LEMMA 3.5.** *We have $q[\bar{m}/x][\bar{n}/y] = \sum_{\bar{n} \lhd \bar{n}_1 * \bar{n}_2} q[\bar{n}_1/y][\bar{m}[\bar{n}_2/y]/x]$ whenever $x \notin \mathcal{V}(\bar{n}) \cup \{y\}$.*

**PROOF.** The proof is straightforward by induction on $q$, using the associativity of sums over partitionings. ■

It is also easy to describe the effect of performing an ordinary substitution after a resource substitution:

**LEMMA 3.6.** *We have* $q[\bar{m}/x]\{N/y\} = q\{N/y\}[\bar{m}\{N/y\}/x]$ *whenever* $x \notin \mathcal{V}(N) \cup \{y\}$.

Reversing the order of substitutions, there is no straightforward way to describe the resource sum $q\{M/x\}[\bar{n}/y]$, because the occurrences of $y$ in $M$ may be duplicated. Still, it is easy to treat the special case where $M$ is reduced to a variable, as the ordinary substitution amounts to a renaming of variables:

**LEMMA 3.7.** *We have* $q\{y/x\}[\bar{n}/y] = q[\bar{n}/x]$ *if* $x = y$ *or* $y \notin \mathcal{V}(q)$.

**Shifting and erasing sequence variables.** If $q$ is a resource expression and $\vec{x} \in \mathcal{V}_s$, we write $q[\vec{x}\uparrow]$ for the term obtained by replacing each $\vec{x}(i)$ occurring free in $q$ with $\vec{x}(i+1)$. Similarly, if $\vec{x}(0)$ does not occur in $q$, we write $q[\vec{x}\downarrow]$ for the expression obtained by replacing each $\vec{x}(i+1)$ in $q$ with $\vec{x}(i)$, so that $q = q[\vec{x}\downarrow][\vec{x}\uparrow]$ in this case — and $r = r[\vec{x}\uparrow][\vec{x}\downarrow]$ for any expression $r$. Given a variable $x$ and a value term $m = \lambda\vec{x}.a$, with $\vec{x}$ chosen so that $x \notin \vec{x}$, we define $\lambda x.m := \lambda\vec{x}.a[\vec{x}\uparrow]\{\vec{x}(0)/x\}$.

**EXAMPLE 3.8.** Using the terms of Example 3.1 again, we have (assuming $x \notin \{y, z\} \cup \vec{z}$ and $y \notin \vec{z}$)

$$\mathsf{p}_x = \lambda z.\mathsf{p}_x \qquad \mathsf{p}_0 = \lambda z.\mathsf{p}_z \qquad \mathsf{p}_{i+1} = \lambda z.\mathsf{p}_i \qquad \mathsf{c}_x = \lambda y.\lambda\vec{z}.x\,[\mathsf{p}_y] :: \iota \qquad \mathsf{c}'_x = \lambda y.\lambda\vec{z}.x\,[\mathsf{c}_y, \mathsf{c}_y] :: \iota\,.$$

◆

We define the **erasure** of the sequence variable $\vec{x}$ in an expression $q$ by:

$$q \lightning \vec{x} := \begin{cases} q & \text{if } \vec{x} \cap \mathcal{V}(q) = \emptyset \\ 0 & \text{otherwise} \end{cases}.$$

In other words, $q \lightning \vec{x} = 0$ if some $\vec{x}(i)$ occurs free in $q$ and $q \lightning \vec{x} = q$ otherwise.

**LEMMA 3.9.** *We have* $q[\vec{x}\uparrow] \lightning \vec{x} = q \lightning \vec{x}$. *Moreover, if* $\vec{x}(0) \notin \mathcal{V}(q)$ *then* $q \lightning \vec{x} = q[\vec{x}\downarrow] \lightning \vec{x}$.

**PROOF.** Direct from the definitions, since $\vec{x}(i+1) \in \mathcal{V}(q[\vec{x}\uparrow])$ iff $\vec{x}(i) \in \mathcal{V}(q)$, and $\vec{x}(i) \in \mathcal{V}(q[\vec{x}\downarrow])$ iff $\vec{x}(i+1) \in \mathcal{V}(q)$. ■

Both shifts and erasure are linearly extended to resource sums:

$$(\sum_{i=1}^{k} q_i)[\vec{x}\uparrow] := \sum_{i=1}^{k} q_i[\vec{x}\uparrow] \qquad (\sum_{i=1}^{k} q_i)[\vec{x}\downarrow] := \sum_{i=1}^{k} q_i[\vec{x}\downarrow] \qquad (\sum_{i=1}^{k} q_i) \lightning \vec{x} := \sum_{i=1}^{k} q_i \lightning \vec{x}$$

and these operations commute with substitution in the following sense:

$$\frac{}{(\lambda x.m)\,\bar{n} :: \vec{p} \mapsto_{\mathrm{r}} m[\bar{n}/x]\,\vec{p}}\;(\mathrm{r}_\beta) \qquad \frac{}{(\lambda \vec{x}.a)\,\iota \mapsto_{\mathrm{r}} a \lightning \vec{x}}\;(\mathrm{r}_\iota)$$

$$\frac{a \mapsto_{\mathrm{r}} A'}{\lambda \vec{x}.a \mapsto_{\mathrm{r}} \lambda \vec{x}.A'}\;(\mathrm{r}_\lambda) \qquad \frac{m \mapsto_{\mathrm{r}} M'}{m\,\vec{n} \mapsto_{\mathrm{r}} M'\,\vec{n}}\;(\mathrm{r}_{appL}) \qquad \frac{\vec{n} \mapsto_{\mathrm{r}} \vec{N}'}{e\,\vec{n} \mapsto_{\mathrm{r}} e\,\vec{N}'}\;(\mathrm{r}_{appR})$$

$$\frac{m \mapsto_{\mathrm{r}} M'}{[m] * \bar{n} \mapsto_{\mathrm{r}} [M'] * \bar{n}}\;(\mathrm{r}_{!}) \qquad \frac{\bar{m} \mapsto_{\mathrm{r}} \bar{M}'}{\bar{m} :: \vec{n} \mapsto_{\mathrm{r}} \bar{M}' :: \vec{n}}\;(\mathrm{r}_{::L}) \qquad \frac{\vec{n} \mapsto_{\mathrm{r}} \vec{N}'}{\bar{m} :: \vec{n} \mapsto_{\mathrm{r}} \bar{m} :: \vec{N}'}\;(\mathrm{r}_{::R})$$

**Figure 10.** Rules of fine-step extensional resource reduction

**LEMMA 3.10.** *If $\vec{x} \notin \mathcal{V}_{\mathrm{s}}(\bar{M})$, then $Q[\bar{M}/\vec{x}(i)][\vec{x}\uparrow] = Q[\vec{x}\uparrow][\bar{M}/\vec{x}(i{+}1)]$ and $Q[\bar{M}/\vec{x}(i{+}1)][\vec{x}\downarrow] = Q[\vec{x}\downarrow][\bar{M}/\vec{x}(i)]$ (assuming $\vec{x}(0) \notin \mathcal{V}(Q)$ in that case). And if $x \notin \vec{x}$, then we have $Q[\bar{M}/x][\vec{x}\uparrow] = Q[\vec{x}\uparrow][\bar{M}[\vec{x}\uparrow]/x]$, $Q[\bar{M}/x][\vec{x}\downarrow] = Q[\vec{x}\downarrow][\bar{M}[\vec{x}\downarrow]/x]$ (assuming $\vec{x}(0) \notin \mathcal{V}(Q) \cup \mathcal{V}(\bar{M})$ in that case) and $Q[\bar{M}/x] \lightning \vec{x} = (Q \lightning \vec{x})[\bar{M} \lightning \vec{x}/x]$.*

**PROOF.** In case $Q = q \in \Delta_{\mathrm{e}}$, each result follows by a straightforward induction on $q$. We deduce the general result by linearity. ■

**Resource reduction.** Now, we can define fine-step resource reduction by:

$$(\lambda x.m)\,\bar{n} :: \vec{p} \mapsto_{\mathrm{r}_0} m[\bar{n}/x]\,\vec{p} \qquad \text{and} \qquad (\lambda \vec{x}.a)\,\iota \mapsto_{\mathrm{r}_0} a \lightning \vec{x}\,.$$

Note that the first reduction step, which uses the just introduced notation for the abstraction of a single variable, amounts to:

$$(\lambda \vec{x}.a)\,\bar{n} :: \vec{p} \mapsto_{\mathrm{r}_0} (\lambda \vec{x}.a[\bar{n}/\vec{x}(0)][\vec{x}\downarrow])\,\vec{p}$$

(choosing $\vec{x} \notin \mathcal{V}_{\mathrm{s}}(\bar{n})$) thanks to Lemma 3.10.

We extend these base reduction steps by applying them under any context, and then applying them in parallel within sums of expressions. Formally:

**DEFINITION 3.11.** **Fine-step resource reduction** is the relation from resource expressions to sums of resource expressions defined by the rules of Figure 10.

Resource reduction is then extended to a relation on term sums by setting $U \to_{\mathrm{r}} U'$ iff $U = \sum_{i=1}^{k} u_i$ and $U' = \sum_{i=1}^{k} U'_i$ with $u_i \mapsto_{\mathrm{r}}^{?} U'_i$ for $1 \leq i \leq k$, where $\mapsto_{\mathrm{r}}^{?}$ is the reflexive closure of $\mapsto_{\mathrm{r}}$ ($u \mapsto_{\mathrm{r}}^{?} U'$ if $u \mapsto_{\mathrm{r}} U'$ or $U' = u$).

We also write $\to_{\mathrm{r}}^{*}$ for the reflexive and transitive closure of $\to_{\mathrm{r}}$, and $\to_{\mathrm{r}}^{k}$ for its $k$-th iteration.

Note that, as particular case, whenever $\vec{y} \notin \mathcal{V}_{\mathrm{s}}(a)$, we have

$$(\lambda x.\lambda \vec{y}.a)\,\bar{n} :: \iota \to_{\mathrm{r}} (\lambda \vec{y}.a[\bar{n}/x])\,\iota \to_{\mathrm{r}} a[\bar{n}/x] \lightning \vec{y} = a[\bar{n}/x]\,.$$

**EXAMPLE 3.12.** The smallest reducible base terms one can form are $\mathsf{p}_x\,\iota$ for $x \in \mathcal{V}$, and $\mathsf{p}_i\,\iota$ for $i \in \mathbb{N}$. Recalling the computations of Example 3.8, we obtain $\mathsf{p}_x\,\iota \mapsto_{\mathrm{r}} \mathsf{p}_x[[\,]/z]\,\iota = \mathsf{p}_x\,\iota$ by $(\mathrm{r}_\beta)$, and $\mathsf{p}_x\,\iota \mapsto_{\mathrm{r}} (x\,\iota) \lightning \vec{y} = x\,\iota$ by $(\mathrm{r}_\iota)$. Similarly, we obtain $\mathsf{p}_0\,\iota \mapsto_{\mathrm{r}} \mathsf{p}_x[[\,]/x]\,\iota = 0\,\iota = 0$ and $\mathsf{p}_{i+1}\,\iota \mapsto_{\mathrm{r}} \mathsf{p}_i[[\,]/x]\,\iota = \mathsf{p}_i\,\iota$ by $(\mathrm{r}_\beta)$, and $\mathsf{p}_i\,\iota \mapsto_{\mathrm{r}} (\vec{y}(i)\,\iota) \lightning \vec{y} = 0\,\iota = 0$ by $(\mathrm{r}_\iota)$.

We moreover have: $\mathsf{p}_y[[\mathsf{p}_x]/y] = \lambda\vec{z}.\mathsf{p}_x\,\iota \mapsto_{\mathrm{r}} \lambda\vec{z}.x\,\iota = \mathsf{p}_x$ for any variables $x$ and $y$, whence the following sequence of reductions:

$$\mathsf{p}_0\,[\mathsf{p}_x] :: \iota \to_{\mathrm{r}} \mathsf{p}_x[[\mathsf{p}_x]/x]\,\iota \to_{\mathrm{r}} \mathsf{p}_x\,\iota \to_{\mathrm{r}} x\,\iota$$

ending on an irreducible base term. Note that the middle step can be derived either by reducing $\mathsf{p}_y[[\mathsf{p}_x]/y]$ to $\mathsf{p}_x$ as above, or by reducing $\mathsf{p}_y[[\mathsf{p}_x]/y]\,\iota = (\lambda\vec{z}.\mathsf{p}_x\,\iota)\,\iota \mapsto_{\mathrm{r}} (\mathsf{p}_x\,\iota) \lightning \vec{z} = \mathsf{p}_x\,\iota$ by $(\mathrm{r}_\iota)$.

The previous computations entail $\lambda\vec{z}.\mathsf{p}_0\,[\mathsf{p}_x] :: \iota \to_{\mathrm{r}}^3 \mathsf{p}_x$ and $\lambda\vec{z}.\mathsf{p}_0\,[\mathsf{p}_{\vec{z}(i)}] :: \iota \to_{\mathrm{r}}^3 \mathsf{p}_i$. This suggests defining a family of value terms $\mathsf{p}_{x,k}$ for $k \in \mathbb{N}$, by setting $\mathsf{p}_{x,0} := \mathsf{p}_x$ and $\mathsf{p}_{x,k+1} := \lambda\vec{z}.\mathsf{p}_0\,[\mathsf{p}_{x,k}] :: \iota$. The previous computation shows $\mathsf{p}_{x,1} \to_{\mathrm{r}}^3 \mathsf{p}_{x,0}$, and an easy generalization entails $\mathsf{p}_{x,k} \to_{\mathrm{r}}^{3k} \mathsf{p}_x$. Similarly setting $\mathsf{p}_{i,0} := \mathsf{p}_i$ and $\mathsf{p}_{i,k+1} := \lambda\vec{z}.\mathsf{p}_0\,[\mathsf{p}_{\vec{z}(i),k}] :: \iota$, we obtain $\mathsf{p}_{i,k} \to_{\mathrm{r}}^{3k} \mathsf{p}_i$. A more intricate example is:

$$\begin{aligned}
\mathsf{c}'_x\,[\mathsf{p}_0, \mathsf{p}_0] :: \iota &\to_{\mathrm{r}}^2 (x\,[\mathsf{c}_y, \mathsf{c}_y] :: \iota)[[\mathsf{p}_0, \mathsf{p}_0]/y] \\
&= 2x\,[\mathsf{c}_y[[\mathsf{p}_0]/y], \mathsf{c}_y[[\mathsf{p}_0]/y]] :: \iota \\
&= 2x\,[\lambda\vec{z}.\mathsf{p}_0\,[\mathsf{p}_{\vec{z}(0)}] :: \iota, \lambda\vec{z}.\mathsf{p}_0\,[\mathsf{p}_{\vec{z}(0)}] :: \iota] :: \iota \\
&= 2x\,[\mathsf{p}_{0,1}, \mathsf{p}_{0,1}] :: \iota \\
&\to_{\mathrm{r}}^6 2x\,[\mathsf{p}_0, \mathsf{p}_0] :: \iota\,.
\end{aligned}$$

◆

Observe that $\to_{\mathrm{r}}$ is automatically reflexive. The reader acquainted with the ordinary resource calculus might be surprised by this choice of definition, as it immediately forbids strong normalizability. However, $\mapsto_{\mathrm{r}}$ is already reflexive on terms containing a redex of the form $a = (\lambda\vec{x}.b)\,\iota$ with $\vec{x} \cap \mathcal{V}(b) = \emptyset$: applying rule $(\mathrm{r}_\beta)$ yields a reduction $a \mapsto_{\mathrm{r}} a$. Nonetheless, resource reduction is weakly normalizing, in the sense that any resource sum reduces to a sum of irreducible expressions: this will follow from strong normalizability of full-step reduction – the latter essentially amounts to iterate $(\mathrm{r}_\beta)$ until $(\mathrm{r}_\iota)$ can be applied. For now, we establish the confluence of fine-step reduction.

First, the extension of $\mapsto_{\mathrm{r}}$ to $\to_{\mathrm{r}}$ ensures that $\to_{\mathrm{r}}$ is linear and compatible (with syntactic constructs), in the following sense:

**LEMMA 3.13.** *Resource reduction $\to_{\mathrm{r}}$ is reflexive and:*

1. *if $U \to_{\mathrm{r}} U'$ and $V \to_{\mathrm{r}} V'$ then $U + V \to_{\mathrm{r}} U' + V'$;*
2. *if $A \to_{\mathrm{r}} A'$ then $\lambda\vec{x}.A \to_{\mathrm{r}} \lambda\vec{x}.A'$;*
3. *if $\vec{N} \to_{\mathrm{r}} \vec{N}'$ then $E\,\vec{N} \to_{\mathrm{r}} E\,\vec{N}'$ and $\bar{M} :: \vec{N} \to_{\mathrm{r}} \bar{M} :: \vec{N}'$;*
4. *if $M \to_{\mathrm{r}} M'$ then $\lambda x.M \to_{\mathrm{r}} \lambda x.M'$, $M\,\bar{N} \to_{\mathrm{r}} M'\,\bar{N}$ and $[M] \to_{\mathrm{r}} [M']$;*

5. *if* $\bar{M} \to_{\mathrm{r}} \bar{M}'$ *then* $\bar{M} * \bar{N} \to_{\mathrm{r}} \bar{M}' * \bar{N}$ *and* $\bar{M} :: \vec{N} \to_{\mathrm{r}} \bar{M}' :: \vec{N}$.

*Moreover,* $(\lambda\vec{x}.A)\,(\bar{N} :: \vec{P}) \to_{\mathrm{r}} \big(\lambda\vec{x}.A[\bar{N}/\vec{x}(0)][\vec{x}\downarrow]\big)\,\vec{P}$ *and* $(\lambda\vec{x}.A)\,\iota \to_{\mathrm{r}} A \lightning \vec{x}$.

**PROOF.** Each statement is a direct consequence of the definitions, by linearity. ■

We then obtain that $\to_{\mathrm{r}}$ is compatible with substitution in the following sense:

**LEMMA 3.14.** *If* $U \to_{\mathrm{r}} U'$ *then for any bag sum* $\bar{N}$ *we have* $U[\bar{N}/x] \to_{\mathrm{r}} U'[\bar{N}/x]$. *And if* $\bar{M} \to_{\mathrm{r}} \bar{M}'$ *then for any resource sum* $Q$ *we have* $Q[\bar{M}/x] \to_{\mathrm{r}} Q[\bar{M}'/x]$.

**PROOF.** We establish the first statement in the case $U = u \mapsto_{\mathrm{r}} U'$, by induction on that reduction. For the two base cases, we use Lemmas 3.5 and 3.10. The other cases follow directly from the induction hypothesis, and the extension to $\to_{\mathrm{r}}$ is straightforward.

We establish the second statement in the case $Q = q \in \Delta_{\mathrm{e}}$ and $\bar{M} = \bar{m} \mapsto_{\mathrm{r}} \bar{M}'$, by induction on $q$. Note that we must have $\bar{m} = [m] * \bar{n}$ and $\bar{M}' = [M'] * \bar{n}$ with $m \mapsto_{\mathrm{r}} M'$: then all cases are straightforward, noting that any sum $\sum_{\bar{m} \lhd \bar{m}_1 * \bar{m}_2} f(\bar{m}_1, \bar{m}_2)$ can be written as $\sum_{\bar{n} \lhd \bar{n}_1 * \bar{n}_2} f([m] * \bar{n}_1, \bar{n}_2) + f(\bar{n}_1, [m] * \bar{n}_2)$. The general result follows by linearity. ■

Moreover, resource reduction preserves free variables and is compatible with shifts and erasure:

**LEMMA 3.15.** *If* $u \mapsto_{\mathrm{r}} U'$ *and* $u' \in U'$ *then* $|u|_x = |u'|_x$. *In particular,* $\mathcal{V}(u) = \mathcal{V}(u')$.

**PROOF.** Straightforward by induction on the reduction $u \mapsto_{\mathrm{r}} U'$, using Lemma 3.4 in the case of rule $(\mathrm{r}_\beta)$. ■

**LEMMA 3.16.** *If* $U \to_{\mathrm{r}} U'$ *then* $U \lightning \vec{x} \to_{\mathrm{r}} U' \lightning \vec{x}$ *and* $U[\vec{x}\uparrow] \to_{\mathrm{r}} U'[\vec{x}\uparrow]$. *If moreover* $\vec{x}(0) \notin \mathcal{V}(U)$ *then* $U[\vec{x}\downarrow] \to_{\mathrm{r}} U'[\vec{x}\downarrow]$.

**PROOF.** Again, the result is proved first for a reduction $u \mapsto_{\mathrm{r}} U'$, then generalized by linearity. Each step is straightforward. ■

**THEOREM 3.17 (Confluence of $\to_{\mathrm{r}}$).** *Resource reduction* $\to_{\mathrm{r}}$ *has the diamond property: if* $U \to_{\mathrm{r}} U_1$ *and* $U \to_{\mathrm{r}} U_2$ *then there exists* $U'$ *such that* $U_1 \to_{\mathrm{r}} U'$ *and* $U_2 \to_{\mathrm{r}} U'$.

**PROOF.** We first establish, by induction on resource terms, that: if $u \mapsto_{\mathrm{r}} U_1$ and $u \mapsto_{\mathrm{r}} U_2$ then there exists $U'$ such that $U_1 \to_{\mathrm{r}} U'$ and $U_2 \to_{\mathrm{r}} U'$.

The crucial case is that of head reducible base terms. Assume, e.g., $U_1$ is obtained by $(\mathrm{r}_\beta)$: then $u = (\lambda\vec{x}.a)\,(\bar{n} :: \vec{p})$, and $U_1 = \big(\lambda\vec{x}.a[\bar{n}/\vec{x}(0)][\vec{x}\downarrow]\big)\,\vec{p}$.

If $U_2$ is obtained by $(\mathrm{r}_\beta)$ too, then $U_1 = U_2$ and we conclude by the reflexivity of $\to_{\mathrm{r}}$, setting $U' := U_1$.

If $U_2$ is obtained by $(\mathrm{r}_\iota)$ then $\bar{n} :: \vec{p} = \iota$ and: either $\vec{x}(0) \in \mathcal{V}(a)$ and $U_1 = U_2 = 0$, and we conclude again by reflexivity; or $\vec{x}(0) \notin \mathcal{V}(a)$ and $U_1 = (\lambda\vec{x}.a[\vec{x}\downarrow])\,\iota \mapsto_{\mathrm{r}} a[\vec{x}\downarrow] \lightning \vec{x} = U_2$ by Lemma 3.9.

If $U_2$ is obtained by $(\mathrm{r}_{appL})$ then $U_2 = (\lambda\vec{x}.A')\,(\bar{n} :: \vec{p})$ with $a \mapsto_{\mathrm{r}} A'$, and we can set $U' := \big(\lambda\vec{x}.A'[\bar{n}/\vec{x}(0)][\vec{x}\downarrow]\big)\,\vec{p}$, to obtain $U_1 \to_{\mathrm{r}} U'$ by Lemmas 3.13, 3.14 and 3.16, and $U_2 \to_{\mathrm{r}} U'$ by Lemma 3.13.

If $U_2$ is obtained by $(\mathrm{r}_{appR})$ then either $U_2 = (\lambda\vec{x}.a)\,(\bar{N}' :: \vec{p})$ with $\bar{n} \mapsto_{\mathrm{r}} \bar{N}'$, and we can set $U' := \big(\lambda\vec{x}.a[\bar{N}'/\vec{x}(0)][\vec{x}\downarrow]\big)\,\vec{p}$, to obtain $U_1 \to_{\mathrm{r}} U'$ by Lemmas 3.13, 3.14 and 3.16, and $U_2 \to_{\mathrm{r}} U'$ by Lemma 3.13; or $U_2 = (\lambda\vec{x}.a)\,(\bar{n} :: \vec{P}')$ with $\vec{p} \mapsto_{\mathrm{r}} \vec{P}'$, and we can set $U' := \big(\lambda\vec{x}.a[\bar{n}/\vec{x}(0)][\vec{x}\downarrow]\big)\,\vec{P}'$, to obtain both $U_1 \to_{\mathrm{r}} U'$ and $U_2 \to_{\mathrm{r}} U'$ by Lemma 3.13.

By symmetry, we have treated all the cases where at least one of $U_1$ or $U_2$ is obtained by $(\mathrm{r}_\beta)$. Now assume, e.g., $U_1$ is obtained by $(\mathrm{r}_\iota)$: then $u = (\lambda\vec{x}.a)\,\iota$, and $U_1 = a \lightning \vec{x}$. Note that $(\mathrm{r}_{appR})$ cannot be applied in this case. If $U_2$ is also obtained by $(\mathrm{r}_\iota)$, we have $U_1 = U_2$ and we conclude by reflexivity. This leaves only the case of $U_2$ being obtained by $(\mathrm{r}_{appL})$: $U_2 = (\lambda\vec{x}.A')\,\iota$, with $a \mapsto_{\mathrm{r}} A'$. Then we set $U' := A' \lightning \vec{x}$ and obtain $U_1 \to_{\mathrm{r}} U'$ by Lemma 3.16 and $U_2 \to_{\mathrm{r}} U'$ by Lemma 3.13.

We are only left with compatibility rules, all falling in two cases: if both rules reduce the same subterm (*i.e.* the rules are the same and, in the case of $(\mathrm{r}_!)$, the reduced subterm is the same), we conclude by the induction hypothesis, together with Lemma 3.13; otherwise, if the rules are distinct (*i.e.* $(\mathrm{r}_{appL})$ *vs* $(\mathrm{r}_{appR})$ for base terms, or $(\mathrm{r}_{::L})$ *vs* $(\mathrm{r}_{::R})$ for stream terms) or are two instances of $(\mathrm{r}_!)$ on distinct subterms, (*i.e.* $u = [m_1] * [m_2] * \bar{n}$, $U_1 = [M_1'] * [m_2] * \bar{n}$, and $U_2 = [m_1] * [M_2'] * \bar{n}$, with each $m_i \mapsto_{\mathrm{r}} M_i'$) then we apply Lemma 3.13 directly.

Now, if $u \mapsto_{\mathrm{r}}^? U_1$ and $u \mapsto_{\mathrm{r}}^? U_2$, then we obtain $U'$ such that $U_1 \to_{\mathrm{r}} U'$ and $U_2 \to_{\mathrm{r}} U'$: if $U_1 = u$ or $U_2 = u$, the result is trivial; and otherwise, the previous result applies.

Now we extend the result to the reduction of term sums: assume $U \to_{\mathrm{r}} U_1$ and $U \to_{\mathrm{r}} U_2$. We can write

$$U = \sum_{i=1}^{k} u_i, \quad U_1 = \sum_{i=1}^{k} U_i^1 \quad \text{and} \quad U_2 = \sum_{i=1}^{k} U_i^2$$

so that $u_i \mapsto_{\mathrm{r}}^? U_i^j$ for $1 \leq i \leq k$ and $1 \leq j \leq 2$. We obtain $U_i'$ such that $U_i^j \to_{\mathrm{r}} U_i'$ for $1 \leq i \leq k$ and $1 \leq j \leq 2$, and then set $U' = \sum_{i=1}^{k} U_i'$, so that $U_1 \to_{\mathrm{r}} U'$ and $U_2 \to_{\mathrm{r}} U'$ by Lemma 3.13. ■

## 3.3 Full-step dynamics

We now present the full-step resource reduction whose main quality is that it enjoys strong normalization. We first need to introduce the resource substitution of a stream for a sequence variable.

**DEFINITION 3.18.** We define the **resource substitution** $q[\vec{n}/\vec{x}]$ of a stream $\vec{n} = \langle \bar{n}_i \rangle_{i\in\mathbb{N}}$ for a sequence variable $\vec{x}$ in a resource expression $q$ by induction on $q$ as follows:

$$
\begin{aligned}
y[\vec{n}/\vec{x}] &:= \begin{cases} n & \text{if } y = \vec{x}(i),\ \bar{n}_i = [n] \text{ and } \bar{n}_j = [\,] \text{ for } j \in \mathbb{N} \setminus \{i\} \\ y & \text{if } y \notin \vec{x} \text{ and } \vec{n} = \iota \\ 0 & \text{otherwise} \end{cases} \\
(\lambda\vec{y}.a)[\vec{n}/\vec{x}] &:= \lambda\vec{y}.a[\vec{n}/\vec{x}] \\
(e\,\vec{m})[\vec{n}/\vec{x}] &:= \sum_{\vec{n} \lhd \vec{n}_1 * \vec{n}_2} e[\vec{n}_1/\vec{x}]\,\vec{m}[\vec{n}_2/\vec{x}] \\
[m_1, \dots, m_k][\vec{n}/\vec{x}] &:= \sum_{\vec{n} \lhd \vec{n}_1 * \cdots * \vec{n}_k} [m_1[\vec{n}_1/\vec{x}], \dots, m_k[\vec{n}_k/\vec{x}]] \\
\iota[\vec{n}/\vec{x}] &:= \begin{cases} \iota & \text{if } \vec{n} = \iota \\ 0 & \text{otherwise} \end{cases} \\
(\bar{m} :: \vec{p})[\vec{n}/\vec{x}] &:= \sum_{\vec{n} \lhd \vec{n}_1 * \vec{n}_2} \bar{m}[\vec{n}_1/\vec{x}] :: \vec{p}[\vec{n}_2/\vec{x}] \qquad \text{if } \bar{m} :: \vec{p} \neq \iota
\end{aligned}
$$

where, in the abstraction case, $\vec{y}$ is chosen so that $\vec{x} \neq \vec{y}$ and $\vec{y} \cap \mathcal{V}(\vec{n}) = \emptyset$.

As for fine-step resource substitution, the condition in the last case of the definition can be dropped. Moreover, if $\vec{m} = \bar{m}_1 :: \cdots :: \bar{m}_k :: \iota$, then

$$\vec{m}[\vec{n}/\vec{x}] = \sum_{\vec{n} \lhd \vec{n}_1 * \cdots * \vec{n}_k} \bar{m}_1[\vec{n}_1/\vec{x}] :: \cdots :: \bar{m}_k[\vec{n}_k/\vec{x}] :: \iota$$

and, equivalently,

$$\langle \bar{m}_i \rangle_{i\in\mathbb{N}}[\vec{n}/\vec{x}] = \sum_{p:\vec{n} \lhd \mathbb{N}} \langle \bar{m}_i[\vec{n} \upharpoonright p^{-1}(i)/\vec{x}] \rangle_{i\in\mathbb{N}} \ .$$

Also, one can check that:

**LEMMA 3.19.** *We have:*

$$q[\iota/\vec{x}] = q \lightning \vec{x}$$

*and, assuming* $\vec{x} \notin \mathcal{V}_{\mathrm{s}}(\bar{m})$*, we have:*

$$q[\bar{m} :: \vec{n}/\vec{x}] = q[\bar{m}/\vec{x}(0)][\vec{x}\downarrow][\vec{n}/\vec{x}] \ .$$

*More generally, if* $\vec{n} = \bar{n}_0 :: \cdots :: \bar{n}_{k-1} :: \iota$*, and* $\vec{x} \notin \mathcal{V}_{\mathrm{s}}(\vec{n})$*, we have:*

$$
\begin{aligned}
q[\vec{n}/\vec{x}] &= q[\bar{n}_0/\vec{x}(0)][\vec{x}\downarrow] \cdots [\bar{n}_{k-1}/\vec{x}(0)][\vec{x}\downarrow] \lightning \vec{x} \\
&= q[\bar{n}_0/\vec{x}(0)] \cdots [\bar{n}_{k-1}/\vec{x}(k-1)] \lightning \vec{x} \ .
\end{aligned}
$$

**PROOF.** The first two statements are established directly from the definitions, by induction on $q$. If $\vec{n} = \bar{n}_0 :: \cdots :: \bar{n}_{k-1} :: \iota$, and $\vec{y} \cap \mathcal{V}(\vec{n}) = \emptyset$, we can iterate $k$ times the second statement then

$$\frac{}{(\lambda\vec{x}.a)\,\vec{n} \mapsto_{\mathrm{R}} a[\vec{n}/\vec{x}]}\,(\mathrm{R}_\beta) \qquad \frac{a \mapsto_{\mathrm{R}} A'}{\lambda\vec{x}.a \mapsto_{\mathrm{R}} \lambda\vec{x}.A'}\,(\mathrm{R}_\lambda) \qquad \frac{m \mapsto_{\mathrm{R}} M'}{m\,\vec{n} \mapsto_{\mathrm{R}} M'\,\vec{n}}\,(\mathrm{R}_{appL}) \qquad \frac{\vec{n} \mapsto_{\mathrm{R}} \vec{N}'}{e\,\vec{n} \mapsto_{\mathrm{R}} e\,\vec{N}'}\,(\mathrm{R}_{appR})$$

$$\frac{m \mapsto_{\mathrm{R}} M'}{[m] * \bar{n} \mapsto_{\mathrm{R}} [M'] * \bar{n}}\,(\mathrm{R}_!) \qquad \frac{\bar{m} \mapsto_{\mathrm{R}} \bar{M}'}{\bar{m} :: \vec{n} \mapsto_{\mathrm{R}} \bar{M}' :: \vec{n}}\,(\mathrm{R}_{::L}) \qquad \frac{\vec{n} \mapsto_{\mathrm{R}} \vec{N}'}{\bar{m} :: \vec{n} \mapsto_{\mathrm{R}} \bar{m} :: \vec{N}'}\,(\mathrm{R}_{::R})$$

**Figure 11.** Rules of full-step extensional resource reduction

---

use the first one to obtain

$$q[\vec{n}/\vec{x}] = q[\bar{n}_0/\vec{x}(0)][\vec{x}\downarrow]\cdots[\bar{n}_{k-1}/\vec{x}(0)][\vec{x}\downarrow] \lightning \vec{x}$$

and then we obtain the final identity by iterating Lemmas 3.9 and 3.10. ■

Substitution of streams enjoys the same regularity as substitution of bags, w.r.t. the size of expressions. Namely, setting $|\vec{n}| := \sum_{i\in\mathbb{N}} |\bar{n}_i|$ when $\vec{n} = \langle \bar{n}_i \rangle_{i\in\mathbb{N}}$, we obtain:

**LEMMA 3.20.** *If $q' \in q[\vec{n}/\vec{x}]$ with $\vec{n} = \langle \bar{n}_i \rangle_{i\in\mathbb{N}}$, then $|q|_{\vec{x}(i)} = |\bar{n}_i|$ for $i \in \mathbb{N}$, and $\#q' = \#q + \#\vec{n} - |\vec{n}|$. If moreover $y \notin \vec{x}$ then $|q'|_y = |q|_y + |\vec{n}|_y$ (in particular, $y \in \mathcal{V}(q')$ iff $y \in \mathcal{V}(q) \cup \mathcal{V}(\vec{n})$).*

**PROOF.** Using Lemma 3.19, it is sufficient to iterate Lemma 3.4. ■

**DEFINITION 3.21.** **Full-step resource reduction** is the relation from resource terms to resource sums defined by the rules of Figure 11, and then extended to a relation on term sums by setting $U \to_{\mathrm{R}} U'$ iff $U = \sum_{i=0}^{k} u_i$ and $U' = \sum_{i=0}^{k} U'_i$, with $u_0 \mapsto_{\mathrm{R}} U'_0$ and $u_i \mapsto^?_{\mathrm{R}} U'_i$ for $1 \le i \le k$.

Note that, here, we do impose at least one summand to be reduced. Using the notation for abstraction over single variables, we can reformulate the base case of reduction $(\mathrm{R}_\beta)$ as follows:

**LEMMA 3.22.** *We have $(\lambda x_1.\cdots\lambda x_k.\lambda\vec{y}.a)\,\bar{n}_1 :: \cdots :: \bar{n}_k :: \vec{p} \mapsto_{\mathrm{R}} a[\bar{n}_1/x_1]\cdots[\bar{n}_1/x_1][\vec{p}/\vec{y}]$. In particular, if $\vec{y} \notin \mathcal{V}_{\mathrm{s}}(a)$ then $(\lambda x_1.\cdots\lambda x_k.\lambda\vec{y}.a)\,\bar{n}_1 :: \cdots :: \bar{n}_k :: \iota \mapsto_{\mathrm{R}} a[\bar{n}_1/x_1]\cdots[\bar{n}_1/x_1]$.*

**PROOF.** The first statement implies the second one by the first identity of Lemma 3.19. We have:

$$\begin{aligned}(\lambda x_1.\cdots\lambda x_k.\lambda\vec{y}.a)\,\bar{n}_1 :: \cdots :: \bar{n}_k :: \vec{p} &= \big(\lambda\vec{y}.a[\vec{y}\uparrow]\{\vec{y}(0)/x_k\}\cdots[\vec{y}\uparrow]\{\vec{y}(0)/x_1\}\big)\,\bar{n}_1 :: \cdots :: \bar{n}_k :: \vec{p}\\ &\mapsto_{\mathrm{R}} a[\vec{y}\uparrow]\{\vec{y}(0)/x_k\}\cdots[\vec{y}\uparrow]\{\vec{y}(0)/x_1\}[\bar{n}_1 :: \cdots :: \bar{n}_k :: \vec{p}/\vec{y}]\end{aligned}$$

where we chose the $x_i$'s and $\vec{y}$ not free in the $\bar{n}_i$'s nor in $\vec{p}$. Then we conclude by iterating $k$ times the following observation: for any term $u$, bag term $\bar{n}$, sequence term $\vec{p}$, variable $x \notin \mathcal{V}(\bar{n} :: \vec{p})$

and sequence variable $\vec{y} \notin \mathcal{V}_{\mathrm{s}}(\bar{n} :: \vec{p}) \cup \mathcal{V}_{\mathrm{s}}(x)$, Lemmas 3.7, 3.10 and 3.19 entail

$$\begin{aligned} u[\vec{y}\uparrow]\{\vec{y}(0)/x\}[\bar{n} :: \vec{p}/\vec{y}] &= u[\vec{y}\uparrow]\{\vec{y}(0)/x\}[\bar{n}/\vec{y}(0)][\vec{y}\downarrow][\vec{p}/\vec{y}] \\ &= u[\vec{y}\uparrow][\bar{n}/x][\vec{y}\downarrow][\vec{p}/\vec{y}] \\ &= u[\vec{y}\uparrow][\vec{y}\downarrow][\bar{n}[\vec{y}\downarrow]/x][\vec{p}/\vec{y}] \\ &= u[\bar{n}/x][\vec{p}/\vec{y}] . \end{aligned}$$

■

**EXAMPLE 3.23.** We consider the same reducible terms as in Example 3.12. We have reductions $\mathsf{p}_x\, \iota \mapsto_{\mathrm{R}} (x\, \iota)[\iota/\vec{y}] = x\, \iota$ and $\mathsf{p}_i\, \iota \mapsto_{\mathrm{R}} (\vec{y}(i)\, \iota)[\iota/\vec{y}] = 0$ by $(\mathrm{R}_\beta)$, ending on normal forms: notice that these are the only possible full-step reductions starting from those base terms.

We moreover have $\mathsf{p}_0\, [\mathsf{p}_x] :: \iota \to_{\mathrm{R}} (\vec{y}(0)\, \iota)[[\mathsf{p}_x] :: \iota/\vec{y}] = \mathsf{p}_x\, \iota \to_{\mathrm{R}} (x\, \iota)[\iota/\vec{z}] = x\, \iota$: notice how the first two fine steps ($(\mathrm{r}_\beta)$ followed by $(\mathrm{r}_\iota)$) of the corresponding reduction sequence of Example 3.12 are performed in a single full step, but the last one reduces a newly created redex. As a consequence, we naturally obtain $\mathsf{p}_{x,k} \to_{\mathrm{R}}^{2k} \mathsf{p}_x$ and $\mathsf{p}_{i,k} \to_{\mathrm{R}}^{2k} \mathsf{p}_i$.

This allows us to revisit the final reduction sequence of Example 3.12 as follows:

$$\mathsf{c}'_x\, [\mathsf{p}_0, \mathsf{p}_0] :: \iota \to_{\mathrm{R}} (x\, [\mathsf{c}_y, \mathsf{c}_y] :: \iota)[[\mathsf{p}_0, \mathsf{p}_0]/y] = 2x\, [\mathsf{p}_{0,1}, \mathsf{p}_{0,1}] :: \iota \to_{\mathrm{R}}^{4} 2x\, [\mathsf{p}_0, \mathsf{p}_0] :: \iota .$$

◆

Recall that, here, we require at least one element in a sum to be reduced. This, together with the fact that the reduction of a redex yields a sum of smaller terms, ensures that full-step resource reduction is strongly normalizing:

**LEMMA 3.24.** *If $u \mapsto_{\mathrm{R}} U'$ and $u' \in U'$ then $\# u > \# u'$.*

**PROOF.** The proof is direct by induction on the reduction, using Lemma 3.20 in the redex case. ■

By a standard argument, we obtain:

**COROLLARY 3.25 (Strong normalization for $\to_{\mathrm{R}}$).** *There is no infinite sequence $\langle U_i \rangle_{i \in \mathbb{N}}$ with $U_i \to_{\mathrm{R}} U_{i+1}$ for $i \in \mathbb{N}$.*

**PROOF.** To each term sum, we associate the multiset of the sizes of its elements: under full-step reduction, this measure is strictly decreasing for the multiset order. ■

Moreover, full-step reduction is a particular case of iterated fine-step reduction:

**LEMMA 3.26.** *If $Q \to_{\mathrm{R}} Q'$ then $Q \to_{\mathrm{r}}^{*} Q'$.*

**PROOF.** It is sufficient to consider the case of $Q = q \mapsto_{\mathrm{R}} Q'$. The proof is then by induction on this reduction: the case of $(\mathrm{R}_\beta)$ follows from Lemma 3.19. All the other cases follow straightforwardly from the induction hypothesis, using Lemma 3.13. ■

**THEOREM 3.27 (Weak normalization for $\to_{\mathrm{r}}$).** *For every resource sum $Q$ there exists a sum $Q'$ of $\mapsto_{\mathrm{r}}$-irreducible expressions such that $Q \to_{\mathrm{r}}^{*} Q'$, and this sum is uniquely defined.*

**PROOF.** We obtain $Q'$ by the previous lemma, observing that an expression is $\mapsto_{\mathrm{r}}$-reducible iff it is $\mapsto_{\mathrm{R}}$-reducible. Unicity follows from the confluence of $\to_{\mathrm{r}}$, together with the fact that if $Q'$ is a sum of $\mapsto_{\mathrm{r}}$-irreducible expressions and $Q' \to_{\mathrm{r}} Q''$ then $Q' = Q''$. ■

Given any resource sum $Q$, we denote by $\mathcal{N}(Q)$ the unique sum of irreducible expressions such that $Q \to_{\mathrm{r}}^{*} \mathcal{N}(Q)$ and call $\mathcal{N}(Q)$ the **normal form** of $Q$.

**THEOREM 3.28 (Confluence of $\to_{\mathrm{R}}^{*}$).** *The reduction $\to_{\mathrm{R}}$ is confluent and $Q \to_{\mathrm{R}}^{*} \mathcal{N}(Q)$.*

**PROOF.** By the previous two results, $\mathcal{N}(Q)$ is the unique $\mapsto_{\mathrm{R}}$-irreducible form of $Q$. ■

Recall that a base term is either a redex or of the shape $x\,\vec{m}$. In particular, there is no closed normal base term: it follows that $\mathcal{N}(a) = 0$ for any closed base term $a$.

## 3.4 Comparison with the resource calculus with tests

Although it was designed following the intuitions exposed in our introduction (and by analogy with game semantics, along the lines of Section 8), the extensional resource calculus shares striking features with the *resource calculus with tests*. The latter was designed by Bucciarelli *et al.* [7] to ensure the definability of each element of a particular reflexive object in the cartesian closed category of sets and multirelations (the relational model of the simply typed $\lambda$-calculus), which induces an extensional model of the pure $\lambda$-calculus [8].[14]

The syntax and dynamics of both calculi involve a finitary notion of resource reduction in presence of infinite sequences of abstractions, and applications to infinite sequences of bags: this concomitance is not fortuitous, and it deserves a more detailed comparison. In the remainder of this section, we thus review those similarities more in detail, and also outline key differences: in particular, we explain how one can view value terms (resp. base terms) in the extensional resource calculus as *particular* terms (resp. tests) of the resource calculus with tests, up to some natural quotient on the syntax.[15]

**The resource calculus with tests.** The syntax of the resource calculus with tests involves three categories of expressions: *terms*, which include the terms of the ordinary resource calculus; *bags* of terms; and *tests*, which are obtained from bags by the so-called *cork* construct $\tau$, and are injected back into terms by the dual *uncork* construct $\bar{\tau}$. Intuitively, a test $a = \tau[m_1, \dots, m_k]$ is interpreted as the result of feeding each term $m_i$ with an infinite sequence of empty bags

---

[14] Knowledge of the latter model helps, but is not necessary for the discussion below. Optionally, the reader may refer to the first paragraphs of Section 7, where we recall its definition.

[15] The reader not interested in the details of the comparison can safely jump to Section 4, on the way to extensional Taylor expansion; or to Section 8 for the correspondence with plays up to homotopy in game semantics.

(the empty stream $\iota$ in our terminology); and a term $\overline{\tau}a$ is viewed as the abstraction of $a$ over a denumerable sequence of dummy variables ($\lambda\vec{z}.a$ with $\vec{z}$ chosen fresh, in our notation). The reduction reflects these intuitions.

Formally, one defines a reduction relation $\mapsto_\tau$ from expressions (terms, bags or tests) to sums of expressions, as the compatible closure of the four base steps:

$$(\lambda x.m)\,\bar{n} \mapsto_\tau m[\bar{n}/x] \qquad\qquad (\overline{\tau}a)\,\bar{n} \mapsto_\tau \begin{cases} \overline{\tau}a & \text{if } \bar{n} = [\,] \\ 0 & \text{otherwise} \end{cases}$$

$$b \mid \tau[\lambda x.m] \mapsto_\tau b \mid \tau[m\{0/x\}] \qquad\qquad b \mid \tau[\overline{\tau}a] \mapsto_\tau b \mid a$$

where $b \mid a$ denotes the parallel composition of tests, obtained by concatenating the underlying bags: $\tau\bar{m} \mid \tau\bar{n} = \tau(\bar{m} * \bar{n})$. This relation from expressions to sums of expressions is extended to a binary reduction relation on sums of expressions $\to_\tau$ by linearity, in the same fashion as in Definition 3.21: one requires that at least one term of a sum is reduced. The obtained reduction $\to_\tau$ is then easily shown to be confluent and strongly normalizing, as for ordinary resource reduction [7, Theorem 3.22].

**Parallel composition of tests.** A first discrepancy with the extensional resource calculus is the presence of an operation of parallel composition on tests: this feature was crucially used by Bucciarelli *et al.* to obtain their full abstraction results.

Indeed, by inspecting the shape of normal forms, one observes that the only closed test in normal form is the empty test $\tau[\,]$ [7, Lemma 3.23]. If one closes an arbitrary test by substituting all its free variables with closed bags, and then considers (the support of) the resulting normal form, the outcome is thus boolean: it is either $\emptyset$ (failure) or $\{\tau[\,]\}$ (success). Semantically, the effect of parallel composition is then to sum over the resources (denoted by closed bags) required by successful tests: this is the key to defining separating test contexts, representing all points of the model, and constitutes the main ingredient of the full abstraction proof [7, Section 5].

The absence of this feature in the extensional resource calculus shows in both syntax and semantics: a normal base term is necessarily the application of a free variable to a normal stream, so the normal form of a closed base term can only be 0 (all tests fail!); and in Section 7, we will give examples of points of the relational model that are not in the interpretation of any normal extensional resource term (see Proposition 7.4 in particular).

In the remaining stages of this comparison, we will thus only consider tests of the shape $\tau[m]$ that we simply write $\tau m$. The two reduction rules on tests become $\tau(\lambda x.m) \mapsto_\tau \tau(m\{0/x\})$ and $\tau(\overline{\tau}a) \mapsto_\tau a$, which are direct analogues of $(\lambda x.m)\,\iota = (\lambda x.m)\,[\,] :: \iota \mapsto_{\mathrm{r}} m[[\,]/x]\,\iota$ by $(\mathrm{r}_\beta)$ and $(\lambda\vec{x}.a)\,\iota \mapsto_{\mathrm{r}} a \lightning \vec{x} = a$ by $(\mathrm{r}_\iota)$ when $\vec{x} \notin \mathcal{V}_{\mathrm{s}}(a)$.

**Extensional resource terms as $\eta$-long terms with tests.** We are now ready to make the correspondence between both calculi more precise. Intuitively, given an expression of the extensional resource calculus, one can obtain an expression of the resource calculus with tests by:

- replacing each base term $e\,\vec{n}$ with a finite sequence of applications to the bags of some prefix of $\vec{n} = \langle \bar{n}_i \rangle_{i \in \mathbb{N}}$, containing all non empty bags, terminated by a cork: this yields a test $\tau(e\,\bar{n}_0 \cdots \bar{n}_{k-1})$ – in particular, $e\,\iota$ can be mapped to $\tau e$, but also to $\tau(e\,[\,])$;
- replacing each value term $\lambda\vec{x}.a$ with a sequence of abstractions on a finite prefix of $\vec{x}$, binding all value variables $\vec{x}(i)$ occurring free in $a$, on top of an uncorked version of $a$: this yields a term $\lambda\vec{x}(0).\cdots\lambda\vec{x}(k-1).\bar{\tau}a$ – in particular, we may obtain $\bar{\tau}a$ when no $\vec{x}(i)$ is free in $a$, but also $\lambda z.\bar{\tau}a$ for any fresh variable $z$.

More formally, we define the **representation relation** $u \triangleleft u'$ where $u$ is a value term $m$ (resp. base term $a$, bag term $\bar{m}$, or stream term $\vec{n}$) and $u'$ is a term $m'$ (resp. test $a'$, bag $\bar{m}'$, or finite tuple of bags $\vec{n}'$), by the following inductive rules:

$$\frac{a \triangleleft a' \qquad \vec{x}(i) \notin \mathcal{V}(a') \text{ for } i \geq k}{\lambda\vec{x}.a \triangleleft \lambda\vec{x}(0).\cdots\lambda\vec{x}(k-1).\bar{\tau}a'} \qquad \frac{\vec{n} \triangleleft \vec{n}'}{x\,\vec{n} \triangleleft \tau(x\,\vec{n}')} \qquad \frac{m \triangleleft m' \qquad \vec{n} \triangleleft \vec{n}'}{m\,\vec{n} \triangleleft \tau(m'\,\vec{n}')}$$

$$\frac{m_1 \triangleleft m'_1 \qquad \cdots \qquad m_k \triangleleft m'_k}{[m_1, \ldots, m_k] \triangleleft [m'_1, \ldots, m'_k]} \qquad \frac{}{\iota \triangleleft \varepsilon} \qquad \frac{\bar{m} \triangleleft \bar{m}' \qquad \vec{n} \triangleleft \vec{n}'}{\bar{m} :: \vec{n} \triangleleft \bar{m}' :: \vec{n}'}$$

where $m'\,\vec{n}'$ in the application rule denotes the iterated application of $m'$ to the bags of $\vec{n}'$. This relation yields a notion of $\eta$-longness for resource terms with tests: a term (resp. a test, or a bag) is in **$\eta$-long form** when it represents a value term (resp. a base term, or a bag term). Of course, one could devise a direct definition, without reference to the extensional resource calculus, that would essentially amount to forgetting about the left-hand side of each judgement in the above rules.

Note in particular that such an $\eta$-long term starts with a denumerable sequence of abstractions (a finite sequence of regular abstractions, followed by $\bar{\tau}$), and that each variable and each redex is applied to a denumerable sequence of bags (a finite sequence of bags, followed by $\tau$). Clearly, each value term admits infinitely many representations: it is sufficient to write each stream as a sequence of bags followed by $\iota$, and to pick a rank $k$ for each abstraction $\lambda\vec{x}.a$ so that no $\vec{x}(i)$ occurs in $a$ for $i \geq k$. Moreover, considered from right to left, the representation relation is functional, so that it defines a map sending each $\eta$-long term $m$ to the unique value term $\widetilde{m}$ such that $\widetilde{m} \triangleleft m$. This map induces an equivalence relation $\sim$ on $\eta$-long terms: one can check that it is the congruence generated by the identities $\tau(m\,[\,]) \sim \tau m$ and $\lambda x.\bar{\tau}a \sim \bar{\tau}a$ when $x \notin \mathcal{V}(a)$. One can thus see value terms as $\sim$-classes of $\eta$-long terms with tests.

A reader comfortable with relational semantics will easily check that the representation relation is semantically valid: $[\![q]\!] = [\![q']\!]$ as soon as $q \triangleleft q'$, where $[\![q]\!]$ is defined in Section 7 and $[\![q']\!]$ is the semantics given by Bucciarelli *et al.* [7, Section 4.3] (the reverse implication cannot hold, as the semantics is not even injective on extensional resource terms, *cf.* Example 7.3).

**Fine-step reduction as resource reduction on terms with tests.** A simple inspection of the definitions shows that any representation of a redex is a reducible test, and that $\widetilde{m}$ is normal iff $m$ is. Note, however, that $\mapsto_\tau$ does not commute with $\sim$ on the nose. For instance, consider $\mathsf{p}_x\,\iota \triangleleft \tau\bar{\tau}\tau x \sim \tau((\bar{\tau}\tau x)\,[\,]) \sim \tau(\lambda y.\bar{\tau}\tau x) \sim \tau((\lambda y.\bar{\tau}\tau x)\,[\,])$ with $y \neq x$. Mimicking the reduction $\mathsf{p}_x\,\iota \mapsto_{\mathrm{r}} \mathsf{p}_x\,\iota$ by $(\mathrm{r}_\beta)$, the last three representations reduce to the first one:

$$\tau((\bar{\tau}\tau x)\,[\,]) \mapsto_\tau \tau(\bar{\tau}\tau x) \qquad \tau(\lambda y.\bar{\tau}\tau x) \mapsto_\tau \tau(\bar{\tau}\tau x)\{0/y\} \qquad \tau(\lambda y.(\bar{\tau}\tau x)\,[\,]) \mapsto_\tau \tau(\bar{\tau}\tau x)[[\,]/y]$$

each time using a different reduction rule. And reducing the first representation yields $\tau\bar{\tau}\tau x \mapsto_\tau \tau x$, reflecting the reduction $\mathsf{p}_x\,\iota \mapsto_{\mathrm{r}} x\,\iota$ by $(\mathrm{r}_\iota)$. One can nonetheless observe that (sums of) $\eta$-long terms are stable under reduction, and that extensional resource reduction amounts to the reduction of $\eta$-long terms *up to* $\sim$ (extended to sums in the obvious way).

We could thus reconstruct the extensional resource calculus, *a posteriori*, as the quotient by $\sim$ of the $\eta$-long fragment of the resource calculus with tests. It is nonetheless more convenient to work with a direct definition of the language and (especially) of the dynamics, rather than having to reason up to $\sim$ – and a workable presentation of full-step reduction $\to_{\mathrm{R}}$ through that lens seems out of reach.

Finally, note that the resource calculus with tests does not feature a primitive construction for abstractions over infinitely many variables, all being potentially free in the immediate subterm. This is of little consequence when dealing with single terms or finite sums of terms, which have finitely many free variables, but the extensional Taylor expansion, described in Section 5, naturally involves such infinite abstractions (see, e.g., the expansion of variables in Section 5.1): having them reflected in the syntax of resource terms simplifies the exposition dramatically.

## 4. Resource vectors

Extensional Taylor expansion will map $\lambda$-terms to infinite linear combinations of value terms. Since resource reduction generates term sums, we will have to consider infinite weighted sums of term sums which, *a priori*, involve infinite sums of coefficients (see Example 4.8 below). It is reasonable to expect that, like in the ordinary case, the uniformity of Taylor expansion would allow us to consider finite scalar sums only, but we leave this for future work, as discussed in the conclusion of the paper (Section 9).

Instead, we impose that (countably) infinite sums are always defined by taking coefficients in a **complete commutative semiring** [20, Section VI.2], *i.e.* a set $\mathbb{K}$ equipped with: a sum operator $\sum : \mathbb{K}^I \to \mathbb{K}$ on countable families that we denote by $\sum_{i\in I}\alpha_i := \sum\langle\alpha_i\rangle_{i\in A}$, satisfying $\sum_{i\in\{j\}}\alpha_i = \alpha_j$, and $\sum_{i\in I}\alpha_i = \sum_{j\in J}\sum_{i\in I_j}\alpha_i$ for any partitioning of $I$ into $\{I_j \mid j \in J\}$; and a commutative monoid structure, denoted multiplicatively, which distributes over $\sum$. A direct consequence of the axioms is that finite sums are associative and commutative. We write $0 \in \mathbb{K}$

for the empty sum and denote binary and finite sums as usual. Equipped with finite sums and products, $\mathbb{K}$ is then a commutative semiring in the usual sense. Moreover, $\mathbb{K}$ is automatically **positive**: if $\alpha_1 + \alpha_2 = 0$ then $\alpha_1 = \alpha_2 = 0$. We also write $1 \in \mathbb{K}$ for the multiplicative unit. Then there is a unique semiring morphism from $\mathbb{N}$ to $\mathbb{K}$: this sends $n \in \mathbb{N}$ to $\sum_{i=1}^{n} 1 \in \mathbb{K}$, and is not necessarily injective.

In order to account for the coefficients of Taylor expansion, we moreover assume that $\mathbb{K}$ **has fractions**, meaning that each $n \in \mathbb{N} \setminus \{0\}$, seen as an element of $\mathbb{K}$, has a multiplicative inverse in $\mathbb{K}$. The semiring of booleans $\mathbb{B}$ and the extended real half line $\overline{\mathbb{R}^+}$, both equipped with the usual operations, are such complete semirings with fractions.

We write $\mathbb{K}^X$ for the semimodule of possibly infinite linear combinations of elements of a given countable set $X$, with coefficients in $\mathbb{K}$: equivalently, these are the $X$-indexed families of elements of $\mathbb{K}$. We will call any such $A \in \mathbb{K}^X$ a **vector**, and we write $A.a \in \mathbb{K}$ for the value of $A$ at $a$, *i.e.* the **coefficient** of $a$ in $A$. We write $\mathrm{supp}(A) = \{a \in X \mid A.a \neq 0\}$ for its **support**. We will often abuse notation and write $a \in A$ for $A.a \neq 0$. Given countable families $\langle A_i \rangle_{i \in I} \in (\mathbb{K}^X)^I$ of vectors and $\langle \alpha_i \rangle_{i \in I} \in \mathbb{K}^I$ of coefficients, we write $\sum_{i \in I} \alpha_i A_i \in \mathbb{K}^X$ for the vector $A$ defined by $A.a \coloneqq \sum_{i \in I} \alpha_i A_i.a \in \mathbb{K}$.

Using the additive monoid structure of $\mathbb{K}$, each finite sum $A \in \Sigma X$ (and in particular each element of $X$) induces a vector with finite support $\widetilde{A} \in \mathbb{K}^X$. Then, for any vector $A$, we have $A = \sum_{a \in A} (A.a)\widetilde{a}$. Note that, again, this embedding of $\Sigma X$ in $\mathbb{K}^X$ need not be injective: for instance if $\mathbb{K} = \mathbb{B}$, $\widetilde{A}$ is nothing but the support of $A$. We will however abuse notation and generally write $A$ instead of $\widetilde{A}$: the implicit application of the embedding should be clear from the context. E.g., if we write a vector $\sum_{i \in I} \alpha_i A_i \in \mathbb{K}^X$ where $\alpha_i \in \mathbb{K}$ and $A_i \in \Sigma X$ for every $i \in I$, this should be read as $\sum_{i \in I} \alpha_i \widetilde{A_i}$.

## 4.1 Vectors of resource terms

As we have already announced above, the extensional Taylor expansion of a pure $\lambda$-term will yield a vector of value terms; moreover, as for ordinary Taylor expansion, the case of an application term will rely on the *promotion* of a vector of value terms to a vector of bag terms. And the latter operation of promotion allows relating ordinary substitution with resource substitution: an analogue of Equation (1) (Page 6) holds for vectors of terms, without reference to Taylor expansion itself (Lemma 4.4). In order to deal with the possible renaming of bound variables during substitutions, we will restrict our attention to vectors of terms whose global set of free sequence variables is finite. In the present subsection, we present this notion formally, and introduce both kinds of substitution.

We call **value vector** any vector $M \in \mathbb{K}^{\Delta_{\mathrm{v}}}$ such that $\mathcal{V}_{\mathrm{s}}(M) \coloneqq \bigcup_{m \in M} \mathcal{V}_{\mathrm{s}}(m)$ is finite — note that $\mathcal{V}(M) \coloneqq \bigcup_{m \in M} \mathcal{V}(m)$ might very well be infinite, but the hypothesis on $\mathcal{V}_{\mathrm{s}}(M)$ is sufficient to ensure that we can always find variables that are not free in $M$. We use the same typographic conventions for value vectors as for value sums and write $\mathbb{K}\langle \Delta_{\mathrm{v}} \rangle$ for the set of value vectors (thus denoted by $M, N, P$). We similarly define **base vectors** (denoted by $A, B, C \in \mathbb{K}\langle\Delta_b\rangle$), **bag vectors** (denoted by $\bar{M}, \bar{N}, \bar{P} \in \mathbb{K}\langle\Delta_!\rangle$), and **stream vectors** (denoted by $\vec{M}, \vec{N}, \vec{P} \in \mathbb{K}\langle\Delta_s\rangle$). Note that we do not impose any other bound on the shape of terms in the definition of vectors: e.g., the length of bags in $\bar{M} \in \mathbb{K}\langle\Delta_!\rangle$ is not bounded in general.

As for sums, we may call **head vector** (denoted $E, F, G \in \mathbb{K}\langle\Delta_h\rangle$) any of a value vector or of a value variable. And we call **term vector** (resp. **resource vector**) any value vector (resp. head vector), base vector, bag vector, or stream vector, which we then denote by a letter among $U, V, W$ (resp. $Q, R, S$). And we write $\mathbb{K}\langle\Delta_t\rangle$ (resp. $\mathbb{K}\langle\Delta_e\rangle$) any of $\mathbb{K}\langle\Delta_v\rangle$ (resp. $\mathbb{K}\langle\Delta_h\rangle$), $\mathbb{K}\langle\Delta_b\rangle$, $\mathbb{K}\langle\Delta_!\rangle$, or $\mathbb{K}\langle\Delta_s\rangle$.

We extend term constructors to term vectors by linearity as we did for sums, and we extend resource substitution to vectors by bilinearity, by setting:

$$Q[\bar{N}/x] := \sum_{q\in\Delta_e} \sum_{\bar{n}\in\Delta_!} (Q.q)(\bar{N}.\bar{n})\, q[\bar{n}/x] \ .$$

Again, one can easily check that, with that extension, all the identities defining resource substitution in Definition 3.2 also hold if we replace terms with vectors. Similarly, for full-step resource substitution, we set:

$$Q[\vec{N}/\vec{x}] := \sum_{q\in\Delta_e} \sum_{\vec{n}\in\Delta_!} (Q.q)(\vec{N}.\vec{n})\, q[\vec{n}/\vec{x}] \ .$$

Having extended term constructors to resource vectors, it is also straightforward to define the ordinary (capture avoiding) **substitution** $q\{F/x\} \in \mathbb{K}\langle\Delta_e\rangle$ of a head vector $F \in \mathbb{K}\langle\Delta_h\rangle$ for a value variable $x$ in any resource expression $q \in \Delta_e$:

**DEFINITION 4.1.** Let $q \in \Delta_e$ and $F \in \mathbb{K}\langle\Delta_h\rangle$. We define $q\{F/x\} \in \mathbb{K}\langle\Delta_e\rangle$ by induction on $q$:

$$\begin{aligned}
y\{F/x\} &:= \begin{cases} F & \text{if } x = y \\ y & \text{otherwise} \end{cases} \\
(\lambda\vec{y}.a)\{F/x\} &:= \lambda\vec{y}.a\{F/x\} \\
(e\,\vec{m})\{F/x\} &:= e\{F/x\}\,\vec{m}\{F/x\} \\
[m_1,\ldots,m_k]\{F/x\} &:= [m_1\{F/x\},\ldots,m_k\{F/x\}] \\
\iota\{F/x\} &:= \iota \\
(\bar{m} :: \vec{p})\{F/x\} &:= \bar{m}\{F/x\} :: \vec{p}\{F/x\} \qquad \text{if } \bar{m} :: \vec{p} \neq \iota
\end{aligned}$$

where, in the abstraction case, $\vec{y}$ is chosen so that $x \notin \vec{y}$ and $\mathcal{V}(F) \cap \vec{y} = \emptyset$.[16]

---

[16] The assumption that $\mathcal{V}_s(F)$ is finite ensures that this requirement can be fulfilled.

Note that, as for the substitution of finite sums for variables,

$$q\{0/x\} = q[[\,]/x] = \begin{cases} 0 & \text{if } x \in \mathcal{V}(q) \\ q & \text{otherwise} \end{cases} .$$

It is also easy to check that, if $|q|_x = 1$, then $q\{F/x\} = q[[F]/x]$, which is linear in $F$. In general, however, $q\{F/x\}$ is not linear in $F$: e.g., when $x \notin \mathcal{V}(q)$, $q\{0/x\} = q \neq 0$. On the other hand, we can extend this definition to substitution inside a resource vector, by linearity: we set $Q\{F/x\} := \sum_{q\in\Delta_e}(Q.q)q\{F/x\} \in \mathbb{K}\langle\Delta_e\rangle$ for any $Q \in \mathbb{K}\langle\Delta_e\rangle$. Then one can check that, with that extension, all the identities in the previous definition also hold if we replace terms with vectors.

Following a similar pattern, one defines the simultaneous substitution $Q\{\vec{F}/\vec{x}\}$ of the tuple $\vec{F} = \langle F_0, \ldots, F_{k-1}\rangle$ (resp. the sequence $\vec{F} = \langle F_i\rangle_{i\in\mathbb{N}}$, assuming $\mathcal{V}_s(\vec{F})$ is finite) of head vectors (note that, despite the similar notations, these are *not* stream vectors) for the tuple $\vec{x} = \langle x_0, \ldots, x_{k-1}\rangle$ of variables (resp. the sequence $\vec{x} = \langle x_i\rangle_{i\in\mathbb{N}}$) of variables in $Q$. In the finite case, moreover assuming $\vec{x} \cap \mathcal{V}(\vec{F}) = \emptyset$, we have

$$Q\{\vec{F}/\vec{x}\} = Q\{F_0/x_0\}\cdots\{F_{k-1}/x_{k-1}\} .$$

And in the infinite case, again assuming $\vec{x} \cap \mathcal{V}(\vec{F}) = \emptyset$, we intuitively have

$$Q\{\vec{F}/\vec{x}\} = Q\{F_0/x_0\}\{F_1/x_1\}\cdots .$$

Formally, if we also assume $\mathcal{V}(Q) \cap \vec{x}$ is finite (which is automatic when $Q \in \Sigma\Delta_e$), then we have

$$Q\{\vec{F}/\vec{x}\} = Q\{F_0/x_0\}\cdots\{F_{k-1}/x_{k-1}\}$$

for any $k$ such that $x_i \in \mathcal{V}(Q)$ implies $i < k$. We will most often consider the case where $\vec{x}$ is in fact a sequence variable and $\vec{x} \notin \mathcal{V}_s(\vec{F})$ — identifying $\vec{x}$ with $\langle\vec{x}(i)\rangle_{i\in\mathbb{N}}$ as we announced. The latter condition is not restrictive: $Q$ being a resource vector, the additional condition on $\mathcal{V}_s(\vec{F})$ ensures that we can always find $\vec{y} \notin \mathcal{V}_s(Q) \cup \mathcal{V}_s(\vec{F})$ and write $Q\{\vec{F}/\vec{x}\} = Q\{\vec{y}/\vec{x}\}\{\vec{F}/\vec{y}\}$.

Note that, writing $\vec{0}$ for the sequence of empty value sums, we have $Q\{\vec{0}/\vec{x}\} = Q\not{\iota}_{\vec{x}} = Q[\iota/\vec{x}]$: the erasure of $\vec{x}$ amounts to substituting 0 for each $\vec{x}(i)$.

## 4.2 Promotion and the Taylor expansion formula for substitution

We are now ready to define the promotion of a value vector, and show that the ordinary substitution of a vector amounts to the resource substitution of its promotion: this is the Taylor expansion formula for substitution. Note that the operation of promotion here is exactly the same as for the ordinary Taylor expansion, and both substitution mechanisms are the usual ones: as a consequence, this result (Lemma 4.4) is stated and proved similarly, although the underlying term language is different. We thus only provide references and sketches of proofs.

On the other hand, it will also be useful to define the promotion of a *sequence* of value vectors, and to establish a similar formula for the substitution of a sequence of value vectors for a sequence variable (Lemma 4.6). The latter requires a bit of care to deal with infinite sequences of terms, but the actually novel phenomenon regarding substitution will only appear when we consider extensional Taylor expansion of $\lambda$-terms, in Section 5: there, the analogue of Equation (1) does not hold as an identity.

Given a value vector $M$, and $\bar{m} = [m_1, \dots, m_k] \in \Delta_!$, we write $M^{\bar{m}} := \prod_{i=1}^{k} M.m_i$. Then we define the bag vector $M^k := [M, \dots, M]$ (with $k$ copies of $M$), and obtain:

$$M^k \quad = \quad \sum_{\langle m_1,\dots,m_k\rangle \in \Delta_\mathrm{v}^k} M^{[m_1,\dots,m_k]}[m_1, \dots, m_k] \ .$$

Then we set

$$M^! = \sum_{k\in\mathbb{N}} \frac{1}{k!} M^k \in \mathbb{K}\langle\Delta_!\rangle$$

which we call the **promotion** of $M$.

A straightforward computation shows that promotion commutes with substitution:

**LEMMA 4.2.** *For any $M$ and $N \in \mathbb{K}\langle\Delta_\mathrm{v}\rangle$, $N^!\{M/x\} = N\{M/x\}^!$. And for any $\vec{M} \in \mathbb{K}\langle\Delta_\mathrm{v}\rangle^{\mathbb{N}}$ such that $\mathcal{V}_\mathrm{s}(\vec{M})$ is finite, we have $N^!\{\vec{M}/\vec{x}\} = N\{\vec{M}/\vec{x}\}^!$.*

It will also be useful to compute the coefficient of a bag in the promotion of a value vector (recalling that $\mathrm{d}(\bar{m})$ is the isotropy degree of $\bar{m}$, as defined in Section 2):

**LEMMA 4.3.** *If $\bar{m} \in M^!$ and $|\bar{m}| = k$ then $M^!.\bar{m} = \frac{M^{\bar{m}}}{\mathrm{d}(\bar{m})}$.*

**PROOF.** This is an easy result on the combinatorics of multisets (see, e.g., [35, Lemma 4.4]). ■

**LEMMA 4.4 (Taylor expansion of substitution).** *For any $Q \in \mathbb{K}\langle\Delta_\mathrm{e}\rangle$ and $M \in \mathbb{K}\langle\Delta_\mathrm{v}\rangle$, $Q\{M/x\} = Q[M^!/x]$.*

**PROOF.** The proof is essentially the same as in the ordinary resource calculus [39, Lemma 4.8]. By linearity, it is sufficient to consider the case of $Q = q \in \Delta_\mathrm{e}$. We first show that the identities defining $q \mapsto q\{M/x\}$ (as in Definition 4.1) are also valid for $q \mapsto q[M^!/x]$: here the definition of sums over partitionings of bags, as used in Definition 3.2, is crucial, in conjunction with Fact 2.1 and Lemma 4.3. The result follows by induction on $q$. ■

Similarly, we associate a **promotion stream vector** $\vec{M}^! \in \mathbb{K}\langle\Delta_\mathrm{s}\rangle$ with each sequence $\vec{M} = \langle M_i\rangle_{i\in\mathbb{N}} \in \mathbb{K}\langle\Delta_\mathrm{v}\rangle^{\mathbb{N}}$ of value vectors such that $\mathcal{V}_\mathrm{s}(\vec{M})$ is finite (again note that, despite the similar notation, the latter sequence is *not* a stream vector). First observe that, by construction, $(M_i^!).[\,] = 1$ for each $i \in \mathbb{N}$. Then we define $\vec{M}^!$ by its coefficients: for every $\vec{m} = \langle\bar{m}_i\rangle_{i\in\mathbb{N}} \in \Delta_\mathrm{s}$, we can set $(\vec{M}^!).\vec{m} := \prod_{i\in\mathbb{N}}(M_i^!).\bar{m}_i$, where only finitely many factors $(M_i^!).\bar{m}_i$ are not 1. In particular, we have $(\vec{M}^!).\iota = 1$ and $(M :: \vec{N}^!).(\bar{m} :: \vec{n}) = (M^!).\bar{m} \times (\vec{N}^!).\vec{n}$, so that $(M :: \vec{N})^! = M^! :: \vec{N}^!$.

Then we obtain the analogue of Lemma 4.2 for the promotion of sequences of value vectors:

**LEMMA 4.5.** *For any $M \in \mathbb{K}\langle\Delta_{\mathrm{v}}\rangle$ and $\vec{N} \in \mathbb{K}\langle\Delta_{\mathrm{v}}\rangle^{\mathbb{N}}$ such that $\mathcal{V}_{\mathrm{s}}(\vec{N})$ is finite, we have $\vec{N}^!\{M/x\} = \vec{N}\{M/x\}^!$. And for any $\vec{M} \in \mathbb{K}\langle\Delta_{\mathrm{v}}\rangle^{\mathbb{N}}$ with $\mathcal{V}_{\mathrm{s}}(\vec{M})$ finite, we have $\vec{N}^!\{\vec{M}/\vec{x}\} = \vec{N}\{\vec{M}/\vec{x}\}^!$.*

**PROOF.** Fix $\vec{p} \in \Delta_{\mathrm{s}}$: we can write $\vec{p} = \bar{p}_1 :: \cdots :: \bar{p}_k :: \iota$. Then write $\vec{N} = N_1 :: \cdots :: N_k :: \vec{P}$. We obtain

$$\begin{aligned}
(\vec{N}^!\{M/x\}).\vec{p} &= (N_1^!\{M/x\} :: \cdots :: N_k^!\{M/x\} :: \vec{P}^!\{M/x\}).\vec{p} \\
&= (N_1^!\{M/x\}).\bar{p}_1 \times \cdots \times (N_k^!\{M/x\}).\bar{p}_k \times (\vec{P}^!\{M/x\}).\iota \\
&= (N_1\{M/x\}^!).\bar{p}_1 \times \cdots \times (N_k\{M/x\}^!).\bar{p}_k \times (\vec{P}\{M/x\}^!).\iota \\
&= (N_1\{M/x\}^! :: \cdots :: N_k\{M/x\}^! :: \vec{P}\{M/x\}^!).\vec{p} \\
&= (\vec{N}\{M/x\}^!).\vec{p}
\end{aligned}$$

where each identity $(N_i^!\{M/x\}).\bar{p}_i = (N_i\{M/x\}^!).\bar{p}_i$ follows from Lemma 4.2, and $(\vec{P}^!\{M/x\}).\iota = (\vec{P}^!).\iota = 1 = (\vec{P}\{M/x\}^!).\iota$ by definition. The proof of the second statement is similar. ■

We obtain the analogue of Lemma 4.4 as well:

**LEMMA 4.6.** *For any $Q \in \mathbb{K}\langle\Delta_{\mathrm{e}}\rangle$, and any $\vec{M} \in \mathbb{K}\langle\Delta_{\mathrm{v}}\rangle^{\mathbb{N}}$ such that $\mathcal{V}_{\mathrm{s}}(\vec{N})$ is finite, we have $Q\{\vec{M}/\vec{x}\} = Q[\vec{M}^!/\vec{x}]$.*

**PROOF.** Write $\vec{M} = \langle M_i\rangle_{i\in\mathbb{N}}$. By linearity, it is sufficient to consider the case of $Q = q \in \Delta_{\mathrm{e}}$. It is possible to follow the same pattern as in the proof of Lemma 4.4, but we can also deduce the present result from Lemma 4.4 itself. Indeed, $\mathcal{V}(q)$ is finite, hence we can choose $k$ such that $i \geq k$ implies $\vec{x}(i) \notin \mathcal{V}(q)$, to obtain

$$q\{\vec{M}/\vec{x}\} = q\{M_0/\vec{x}(0)\} \cdots \{M_{k-1}/\vec{x}(k-1)\}$$

as discussed above, and also

$$q[\vec{M}^!/\vec{x}] = q[M_0^!/\vec{x}(0)] \cdots [M_{k-1}^!/\vec{x}(k-1)]$$

by Lemma 3.19 — assuming, w.l.o.g., that $\vec{x} \cap \bigcup_{i<k} \mathcal{V}(M_i) = \emptyset$. It is then sufficient to iterate Lemma 4.4. ■

Alternatively to the above definition, we could introduce $\vec{M}^!$ similarly to the promotion of value vectors, as follows. First, we call **degree stream** any sequence of natural numbers $\vec{k} = \langle k_i\rangle_{i\in\mathbb{N}} \in \mathbb{N}^{\mathbb{N}}$ with finite support: $\{i \in \mathbb{N} \mid k_i \neq 0\}$ is finite. We will write $\mathbb{N}_{\mathrm{s}}$ for the set of degree streams. Degree streams could thus be identified with finite multisets of natural numbers, but we use different notations to fit the way we use them. A more relevant intuition is to identify $\mathbb{N}_{\mathrm{s}}$ with the set of streams over a singleton set $\mathcal{S}(\{*\})$: we denote $\iota := \langle 0\rangle_{i\in\mathbb{N}} \in \mathbb{N}_{\mathrm{s}}$, and if $k \in \mathbb{N}$ and $\vec{l} \in \mathbb{N}_{\mathrm{s}}$, we write $k :: \vec{l} \in \mathbb{N}_{\mathrm{s}}$ for the stream obtained by pushing $k$ at the head of $\vec{l}$. Given $\vec{M} \in \mathbb{K}\langle\Delta_{\mathrm{v}}\rangle^{\mathbb{N}}$ and $\vec{k} \in \mathbb{N}_{\mathrm{s}}$, we define $\vec{M}^{\vec{k}}$ inductively as follows: $\vec{M}^{\iota} := \iota$ and $(M :: \vec{N})^{k :: \vec{l}} := M^k :: N^{\vec{l}}$ when $k :: \vec{l} \neq \iota$. We moreover define $\vec{k}! \in \mathbb{N}$ by setting $\vec{k}! = \prod_{i \in \mathbb{N}} k_i!$, which satisfies: $\iota! = 1$ and $(k :: \vec{l})! = k! \times \vec{l}!$.

We obtain:

**LEMMA 4.7.** *For any sequence $\vec{M} \in \mathbb{K}\langle\Delta_{\mathrm{v}}\rangle^{\mathbb{N}}$ of value vectors such that $\mathcal{V}_{\mathrm{s}}(\vec{M})$ is finite, we have $\vec{M}^! = \sum_{\vec{k} \in \mathbb{N}_{\mathrm{s}}} \frac{1}{\vec{k}!} \vec{M}^{\vec{k}}$.*

**PROOF.** It is sufficient to check that: $\left(\sum_{\vec{k} \in \mathbb{N}_{\mathrm{s}}} \frac{1}{\vec{k}!} \vec{M}^{\vec{k}}\right).\iota = 1$ and that

$$\left(\sum_{\vec{k} \in \mathbb{N}_{\mathrm{s}}} \frac{1}{\vec{k}!} (M :: \vec{N})^{\vec{k}}\right).(\bar{m} :: \vec{n}) = (M^!).\bar{m} \times \left(\sum_{\vec{k} \in \mathbb{N}_{\mathrm{s}}} \frac{1}{\vec{k}!} \vec{N}^{\vec{k}}\right).\vec{n}$$

which follows directly from the definitions. ■

This alternative presentation of the promotion of a sequence of value vectors will be useful to establish its compatibility with reduction (Lemma 4.12 below).

## 4.3 Reduction of resource vectors

Now we present the notion of reduction on term vectors that we will use to simulate both $\beta$- and $\eta$-reductions via extensional Taylor expansion: we define the **resource reduction on term vectors** by setting $U \rightsquigarrow U'$ if $U = \sum_{i \in I} \alpha_i u_i$ and $U' = \sum_{i \in I} \alpha_i U'_i$ with $u_i \in \Delta_{\mathrm{t}}$, $U'_i \in \Sigma\Delta_{\mathrm{t}}$ and $u_i \to_{\mathrm{r}}^* U'_i$ for $i \in I$. We also define the **normal form of a term vector**, point-wise:

$$\mathcal{N}(U) := \sum_{u \in U} (U.u)\mathcal{N}(u)\ .$$

Note in particular that, in the definition of resource reduction, we do not impose the terms $u_i$ to be pairwise distinct, and that the number of $\to_{\mathrm{r}}$ reduction steps from each $u_i$ is not bounded. We will see that the latter observation is crucial for our purposes: contrasting with the ordinary case where parallel reduction is sufficient to capture $\beta$-reduction [39], the ability to iterate reductions on each element of a term vector is essential to obtain our simulation results in Section 5 – see the proof of Lemma 5.4 and the subsequent discussion.

**EXAMPLE 4.8.** Recalling the reductions of Example 3.12, observe that we have $\sum_{i \in \mathbb{N}} \mathsf{p}_i\, \iota \rightsquigarrow 0$ by picking a reduction $\mathsf{p}_i\, \iota \to_{\mathrm{r}} 0$ for each $i \in \mathbb{N}$. On the other hand, we have, e.g., $\sum_{k \in \mathbb{N}} \mathsf{p}_{0,k} \rightsquigarrow \sum_{k \in \mathbb{N}} \mathsf{p}_0$, which demonstrates that, without any particular constraint on the support of term vectors, infinite sums of coefficients are generated by reduction and normalization. ◆

**LEMMA 4.9.** *For any term vector $U$, we have $U \rightsquigarrow \mathcal{N}(U)$. If moreover $U = \sum_{i \in I} \alpha_i U_i$ with $U_i \in \mathbb{K}\langle\Delta_{\mathrm{t}}\rangle$ for $i \in I$, then $\mathcal{N}(U) = \sum_{i \in I} \alpha_i \mathcal{N}(U_i)$. Finally, if $U \rightsquigarrow U'$ then $\mathcal{N}(U) = \mathcal{N}(U')$.*

**PROOF.** The first statement follows from the definitions, observing that $u \rightarrow_{\mathrm{r}}^{*} \mathcal{N}(u)$. The second one follows from the linearity of $\mathcal{N}(-)$. And if $U \rightsquigarrow U'$ then we can write $U = \sum_{i \in I} \alpha_i u_i$ and $U' = \sum_{i \in I} \alpha_i U'_i$ with $u_i \rightarrow_{\mathrm{r}}^{*} U'_i$ for $i \in I$: then, by confluence of $\rightarrow_{\mathrm{r}}^{*}$, $\mathcal{N}(u_i) = \mathcal{N}(U'_i)$ for each $i \in I$, and we conclude by the previous point. ■

The confluence of $\rightsquigarrow$ follows directly. Moreover, $\rightsquigarrow$ is linear and compatible, in the following sense:

**LEMMA 4.10.** *The relation $\rightsquigarrow$ is reflexive and:*

1. *if $U_i \rightsquigarrow U'_i$ with $U_i, U'_i \in \mathbb{K}\langle\Delta_{\mathrm{t}}\rangle$ for $i \in I$, and $\bigcup_{i \in I} \mathcal{V}_{\mathrm{s}}(U_i)$ is finite, then $\sum_{i \in I} \alpha_i U_i \rightsquigarrow \sum_{i \in I} \alpha_i U'_i$;*
2. *if $A \rightsquigarrow A'$ then $\lambda\vec{x}.A \rightsquigarrow \lambda\vec{x}.A'$; and if $M \rightsquigarrow M'$ then $\lambda x.M \rightsquigarrow \lambda x.M'$;*
3. *if $\vec{N} \rightsquigarrow \vec{N}'$ then $x\,\vec{N} \rightsquigarrow x\,\vec{N}'$; if moreover $M \rightsquigarrow M'$, then $M\,\vec{N} \rightsquigarrow M'\,\vec{N}'$;*
4. *if $M \rightsquigarrow M'$ then $[M] \rightsquigarrow [M']$; and if $\bar{M} \rightsquigarrow \bar{M}'$ and $\bar{N} \rightsquigarrow \bar{N}'$ then $\bar{M} * \bar{N} \rightsquigarrow \bar{M}' * \bar{N}'$;*
5. *if $\bar{M} \rightsquigarrow \bar{M}'$ and $\vec{N} \rightsquigarrow \vec{N}'$ then $\bar{M} :: \vec{N} \rightsquigarrow \bar{M}' :: \vec{N}'$;*
6. *if $U \rightsquigarrow U'$ and $\bar{N} \rightsquigarrow \bar{N}'$ then $U[\bar{N}/x] \rightsquigarrow U'[\bar{N}'/x]$.*

*Moreover, $(\lambda\vec{x}.A)\,(\bar{N} :: \vec{P}) \rightsquigarrow \big(\lambda\vec{x}.A[\bar{N}/\vec{x}(0)][\vec{x}\downarrow]\big)\,\vec{P}$ and $(\lambda\vec{x}.A)\,\vec{M} \rightsquigarrow A[\vec{M}/\vec{x}]$.*

**PROOF.** Each result follows from the definitions, also using Lemma 3.13 for Items 2 to 5, Lemma 3.14 for item 6, and Lemma 3.26 for the big-step redex case. ■

The reduction of value vectors is moreover compatible with promotion. We first establish:

**LEMMA 4.11.** *If $M_i \rightsquigarrow M'_i$ for $i \in \mathbb{N}$, then for all $\vec{k} \in \mathbb{N}_{\mathrm{s}}$, $\langle M_i\rangle_{i \in \mathbb{N}}^{\vec{k}} \rightsquigarrow \langle M'_i\rangle_{i \in \mathbb{N}}^{\vec{k}}$.*

**PROOF.** We reason by induction on $\vec{k}$. If $\vec{k} = \iota$, the result holds by reflexivity. Otherwise, we write $\vec{k} = k :: \vec{l}$, and $N_i = M_{i+1}$ and $N'_i = M'_{i+1}$ for $i \in \mathbb{N}$, so that $\langle M_i\rangle_{i \in \mathbb{N}}^{\vec{k}} = M_0^k :: \langle N_i\rangle_{i \in \mathbb{N}}^{\vec{l}}$ and similarly for $\langle M'_i\rangle_{i \in \mathbb{N}}^{\vec{k}}$. We have $M_0^k \rightsquigarrow M_0'^{k}$ by Item 4 of Lemma 4.10, and $\langle N_i\rangle_{i \in \mathbb{N}}^{\vec{l}} \rightsquigarrow \langle N'_i\rangle_{i \in \mathbb{N}}^{\vec{l}}$ by induction hypothesis. We conclude by Item 5 of Lemma 4.10. ■

**LEMMA 4.12.** *If $M \rightsquigarrow M'$ then $M^{!} \rightsquigarrow M'^{!}$. And if $M_i \rightsquigarrow M'_i$ with $\bigcup_{i \in \mathbb{N}} \mathcal{V}_{\mathrm{s}}(M_i)$ finite, then $(\langle M_i\rangle_{i \in \mathbb{N}})^{!} \rightsquigarrow (\langle M'_i\rangle_{i \in \mathbb{N}})^{!}$.*

**PROOF.** We have $M^{!} = \sum_{k \in \mathbb{N}} \frac{1}{k!} M^k$ and similarly for $M'^{!}$, with $M^k \rightsquigarrow M'^k$ for each $k$ by Item 4 of Lemma 4.10, and we conclude by Item 1 of Lemma 4.10. The second statement is established similarly, thanks to Lemmas 4.7 and 4.11. ■

Together with Lemma 4.10 (Item 6) and Lemma 4.4, the previous lemma entails:

**COROLLARY 4.13.** *If $U \rightsquigarrow U'$ and $N \rightsquigarrow N'$ then $U\{N/x\} \rightsquigarrow U'\{N'/x\}$.*

We refer to Lemmas 4.10 and 4.12 and Corollary 4.13 collectively as the **compatibility properties** of $\rightsquigarrow$, which straightforwardly extend to its (reflexive and) transitive closure $\rightsquigarrow^*$: we use these properties extensively in the following sections, and keep them implicit most of the time.

Note that we do not establish the transitivity of $\rightsquigarrow$. Consider two reductions $U = \sum_{i\in I} \alpha_i u_i \rightsquigarrow \sum_{i\in I} \alpha_i U'_i = U'$ and $U' = \sum_{j\in J} \beta_j u'_j \rightsquigarrow \sum_{j\in J} \beta_j U''_j = U''$. Intuitively, to deduce a reduction $U \rightsquigarrow U''$ using the transitivity of $\to^*_{\mathrm{r}}$, we would need to "synchronize" the two writings of $U'$ in a way that is compatible with the families of reductions $u_i \to^*_{\mathrm{r}} U'_i\ and\ u'_j \to^*_{\mathrm{r}} U''_j$: there is no obvious way to perform this synchronization, and the transitivity of $\rightsquigarrow$ is an open question at the time of writing. Fortunately, we do not need to rely on it: we rather use $\rightsquigarrow^*$, or resort to reason component-wise and use the transitivity of $\to^*_{\mathrm{r}}$ instead.

## 5. Extensional Taylor expansion

A first possible definition of extensional Taylor expansion amounts to performing infinite $\eta$-expansion on the fly, in order to produce value vectors, setting:

$$\mathcal{T}(x) := \lambda\vec{y}.x\,\vec{\mathcal{T}}(\vec{y})^! \qquad \mathcal{T}(\lambda x.M) := \lambda x.\mathcal{T}(M) \qquad \mathcal{T}(M\,N) := \lambda\vec{y}.\mathcal{T}(M)\,\mathcal{T}(N)^! :: \vec{\mathcal{T}}(\vec{y})^!$$

where $\vec{\mathcal{T}}(\vec{y})$ denotes the sequence $\langle \mathcal{T}(\vec{y}(i))\rangle_{i\in\mathbb{N}}$ of value vectors.

Note that, although this definition seems to be circular in the variable case, it can be done by defining the coefficient of a resource term in the Taylor expansion of a $\lambda$-term, by induction on the resource term, as we will do below. Moreover observe that this definition does not preserve normal forms: the application case always introduces redexes. This can be fixed by defining Taylor expansion based on the head structure of terms, where $\eta$-expansion does not create redexes. We thus define extensional Taylor expansion in several steps:

- first the *expansion of variables*, where all the apparent circularity resides: $x^\eta$;
- then the *structural expansion* of $\lambda$-terms, straightforwardly as above: $\mathcal{T}_\eta(M)$;
- and the *head expansion*, avoiding the creation of redexes: $\mathcal{T}_h(M)$.

Then we show that $\mathcal{T}_\eta(M)$ reduces to $\mathcal{T}_h(M)$, and that it allows us to simulate $\beta$-reduction and $\eta$-reduction steps by resource reduction. The crux of the construction is to show that the expansion of a variable is compatible with substitution: $x^\eta\{m/x\}$ and $m\{x^\eta/x\}$ reduce to $m$.

We thus obtain not just one, but two notions of extensional Taylor expansion of a $\lambda$-term: its structural expansion and its head expansion. And since one reduces to the other, they have the same normal form, so both notions induce the same $\lambda$-theory via normalization, and we may pick one or the other depending on our objectives. For instance, the simulation of $\beta$- and $\eta$-reductions is easier to establish for the structural version, but the head version will be used crucially in the next section, to show that the induced $\lambda$-theory is sensible (it equates all unsolvable terms).

## 5.1 Infinitely $\eta$-expanded variables

We define simultaneously the **value expansion** $x^\eta \in \mathbb{K}\langle\Delta_{\mathrm{v}}\rangle$ of a variable $x$ and the **stream expansion** $\vec{x}^{\,!} \in \mathbb{K}\langle\Delta_{\mathrm{s}}\rangle$ of a sequence variable $\vec{x}$ so that:

$$\begin{array}{rl} & x^\eta = \lambda\vec{y}.x\,\vec{y}^{\,!} \quad (\text{choosing } \vec{y} \not\ni x) \\ \text{and} & \vec{x}^{\,!} = (\vec{x}^{\,\eta})^! \quad \text{where } \vec{x}^{\,\eta} := \langle \vec{x}(i)^\eta\rangle_{i\in\mathbb{N}}\,. \end{array}$$

To be formal, we define the coefficients of these vectors by mutual induction on value terms and on stream terms:

$$\begin{aligned} x^\eta.u &:= \begin{cases} \vec{y}^{\,!}.\vec{m} & \text{if } u = \lambda\vec{y}.x\,\vec{m} \text{ with } \vec{y} \not\ni x, \\ 0 & \text{otherwise} \end{cases} \\ \vec{x}^{\,!}.\vec{m} &:= \prod_{i\in\mathbb{N}} (\vec{x}(i)^\eta)^!.\bar{m}_i \qquad \text{if } \vec{m} = \langle \bar{m}_i\rangle_{i\in\mathbb{N}} \end{aligned}$$

which ensures that the previous two identities hold. We moreover write $x^! := (x^\eta)^!$.

The stream expansion of a sequence variable is subject to the following recursive characterization, where $Q[\vec{x}\uparrow]^k$ denotes $Q[\vec{x}\uparrow]\cdots[\vec{x}\uparrow]$ (with $k$ applications of $-[\vec{x}\uparrow]$):

**LEMMA 5.1.** *We have*

$$\vec{x}^{\,!} = \vec{x}(0)^! :: \vec{x}^{\,!}[\vec{x}\uparrow]$$

*and, more generally, for every $k \in \mathbb{N}$:*

$$\vec{x}^{\,!} = \vec{x}(0)^! :: \cdots :: \vec{x}(k-1)^! :: \vec{x}^{\,!}[\vec{x}\uparrow]^k\,.$$

**PROOF.** The second statement just iterates the first one, which is established component-wise: it is easy to check from the definition that $\vec{x}^{\,!}.\vec{m} = (\vec{x}(0)^! :: \vec{x}^{\,!}[\vec{x}\uparrow]).\vec{m}$ for every $\vec{m} \in \Delta_{\mathrm{s}}$.. ■

We can compute the coefficient of a value term in $x^\eta$ directly. We first define the **multiplicity coefficient** $\mathsf{m}(q)$ of any resource expression $q$ inductively as follows:

$$\begin{array}{rlrl} \mathsf{m}(x) &:= 1 & \mathsf{m}(q\,\vec{n}) &:= \mathsf{m}(q) \times \mathsf{m}(\vec{n}) \\ \mathsf{m}(\lambda\vec{x}.a) &:= \mathsf{m}(a) & \mathsf{m}([m_1,\dots,m_k]) &:= \mathsf{d}([m_1,\dots,m_k]) \times \displaystyle\prod_{i=1}^{k} \mathsf{m}(m_i) \\ \mathsf{m}(\iota) &:= 1 & \mathsf{m}(\bar{m} :: \vec{n}) &:= \mathsf{m}(\bar{m}) \times \mathsf{m}(\vec{n}) \end{array}$$

so that $\mathsf{m}(u)$ is the product of isotropy degrees (see Section 2, Page 18) of the bags of $u$. Then:

**LEMMA 5.2.** *If $m \in x^\eta$ (resp. $\bar{m} \in x^!$; $\vec{m} \in \vec{x}^{\,!}$) then $x^\eta.m = \frac{1}{\mathsf{m}(m)}$ (resp. $x^{\,!}.\bar{m} = \frac{1}{\mathsf{m}(\bar{m})}$; $\vec{x}^{\,!}.\vec{m} = \frac{1}{\mathsf{m}(\vec{m})}$).*

**PROOF.** The proof is straightforward by induction on terms, using Lemma 4.3 in the case of a bag. ■

**EXAMPLE 5.3.** Let us review some of the smallest terms in $x^\eta$ and $\vec{y}^{\,!}$ (or, rather, in their respective support sets), together with their multiplicity coefficients. First observe that, by Lemma 5.1, any element of $\vec{y}^{\,!}$ is of the shape $\bar{n}_1 :: \cdots :: \bar{n}_k :: \iota$ with $\bar{n}_i \in \vec{y}(i)^!$ for $1 \leq i \leq k$.

For any variable $x$, the smallest value term in $x^\eta$ is the term $\mathsf{p}_x = \lambda\vec{y}.x\,\iota$ of Example 3.1. We thus have $[\mathsf{p}_{\vec{y}(0)}] :: \iota \in \vec{y}^{\,!}$, hence $\mathsf{c}_x = \lambda\vec{y}.x\,[\mathsf{p}_{\vec{y}(0)}] :: \iota \in x^\eta$, and thus also $\lambda\vec{y}.x\,[\mathsf{c}_{\vec{y}(0)}] :: \iota \in x^\eta$. This suggests defining $\mathsf{c}_{x,k}$ for $k \in \mathbb{N}$ inductively by $\mathsf{c}_{x,0} \coloneqq \mathsf{p}_x$ and $\mathsf{c}_{x,k+1} \coloneqq \lambda\vec{y}.x\,[\mathsf{c}_{\vec{y}(0),k}] :: \iota$ so that $\mathsf{c}_x = \mathsf{c}_{x,1}$ and $\mathsf{c}_{x,k} \in x^\eta$ for each $k \in \mathbb{N}$.

The multiplicity coefficient of each of the previous examples is 1, but we also have, e.g., $\mathsf{c}'_x = \lambda\vec{y}.x\,[\mathsf{c}_{\vec{y}(0)}, \mathsf{c}_{\vec{y}(0)}] :: \iota \in x^\eta$, with multiplicity coefficient 2, observing that $\mathsf{m}([\mathsf{c}_{\vec{y}(0)}, \mathsf{c}_{\vec{y}(0)}]) = \mathsf{d}([\mathsf{c}_{\vec{y}(0)}, \mathsf{c}_{\vec{y}(0)}]) = 2$.

◆

The value expansion of a variable is meant to behave like an identity morphism on value terms: as we have announced, we have $x^\eta\{m/x\} \rightsquigarrow m$ and $m\{x^\eta/x\} \rightsquigarrow m$ for any $m \in \Delta_{\mathrm{v}}$. To establish these properties, we will actually show that, for every $m \in \Delta_{\mathrm{v}}$, there is exactly one element of $x^\eta$ contributing to each of those reductions. More precisely:

**LEMMA 5.4.** *For any resource term $u$, any value variable $x$, and any sequence variable $\vec{x}$, the following holds:*

1. *there exists $\mathsf{c}^-\langle x, u\rangle \in x^!$ such that $u[\mathsf{c}^-\langle x, u\rangle/x] \to_{\mathrm{r}}^* \mathsf{m}(\mathsf{c}^-\langle x, u\rangle)u$, and $u[\bar{p}/x] \to_{\mathrm{r}}^* 0$ for any other $\bar{p} \in x^!$;*
2. *there exists $\mathsf{c}^-\langle \vec{x}, u\rangle \in \vec{x}^{\,!}$ such that $u[\mathsf{c}^-\langle \vec{x}, u\rangle/\vec{x}] \to_{\mathrm{r}}^* \mathsf{m}(\mathsf{c}^-\langle \vec{x}, u\rangle)u$, and $u[\vec{p}/\vec{x}] \to_{\mathrm{r}}^* 0$ for any other $\vec{p} \in \vec{x}^{\,!}$;*
3. *if $u = m \in \Delta_{\mathrm{v}}$ then there exists $\mathsf{c}^+\langle x, m\rangle \in x^\eta$ such that $\mathsf{c}^+\langle x, m\rangle\{m/x\} \to_{\mathrm{r}}^* \mathsf{m}(\mathsf{c}^+\langle x, m\rangle)m$, and $p\{m/x\} \to_{\mathrm{r}}^* 0$ for any other $p \in x^\eta$;*
4. *if $u = \bar{m} \in \Delta_!$ then there exists $\mathsf{c}^+\langle x, \bar{m}\rangle \in x^!$ such that $\mathsf{c}^+\langle x, \bar{m}\rangle[\bar{m}/x] \to_{\mathrm{r}}^* \mathsf{m}(\mathsf{c}^+\langle x, \bar{m}\rangle)\bar{m}$, and $\bar{p}[\bar{m}/x] \to_{\mathrm{r}}^* 0$ for any other $\bar{p} \in x^!$;*
5. *if $u = \vec{m} \in \Delta_{\mathrm{s}}$ then:*
   - *there exists $\mathsf{c}^+\langle \vec{x}, \vec{m}\rangle \in \vec{x}^{\,!}$ such that $\mathsf{c}^+\langle \vec{x}, \vec{m}\rangle[\vec{m}/\vec{x}] \to_{\mathrm{r}}^* \mathsf{m}(\mathsf{c}^+\langle \vec{x}, \vec{m}\rangle)\vec{m}$, and $\vec{p}\,[\vec{m}/\vec{x}] \to_{\mathrm{r}}^* 0$ for any other $\vec{p} \in \vec{x}^{\,!}$;*
   - *and there exists $\mathsf{c}\langle x, \vec{m}\rangle \in x^\eta$ such that $\mathsf{c}\langle x, \vec{m}\rangle\,\vec{m} \to_{\mathrm{r}}^* \mathsf{m}(\mathsf{c}\langle x, \vec{m}\rangle)(x\,\vec{m})$, and $p\,\vec{m} \to_{\mathrm{r}}^* 0$ for any other $p \in x^\eta$.*

We will deduce the announced identity behaviour from Items 1 and 3. The other items will be leveraged similarly (see the corresponding items of Lemma 5.7), but they are already useful here as intermediate steps in the proof of Lemma 5.4: we prove all five items simultaneously by induction on $u$. Along the proof, we define $\mathsf{c}^-\langle x, u\rangle \in x^!$, $\mathsf{c}^-\langle \vec{x}, u\rangle \in \vec{x}^{\,!}$, $\mathsf{c}^+\langle x, m\rangle \in x^\eta$,

$$\mathsf{c}^-\langle x, \lambda\vec{y}.b\rangle := \mathsf{c}^-\langle x, b\rangle \qquad \mathsf{c}^-\langle x, [m_1, \dots, m_k]\rangle := \mathsf{c}^-\langle x, m_1\rangle * \cdots * \mathsf{c}^-\langle x, m_k\rangle$$
$$\mathsf{c}^-\langle x, y\,\vec{n}\rangle := \mathsf{c}^-\langle x, \vec{n}\rangle \qquad \mathsf{c}^-\langle x, x\,\vec{n}\rangle := [\mathsf{c}\langle x, \vec{n}\rangle] * \mathsf{c}^-\langle x, \vec{n}\rangle \qquad \mathsf{c}^-\langle x, m\,\vec{n}\rangle := \mathsf{c}^-\langle x, m\rangle * \mathsf{c}^-\langle x, \vec{n}\rangle$$
$$\mathsf{c}^-\langle x, \iota\rangle := [\,] \qquad \mathsf{c}^-\langle x, \bar{m} :: \vec{n}\rangle := \mathsf{c}^-\langle x, \bar{m}\rangle * \mathsf{c}^-\langle x, \vec{n}\rangle$$
$$\mathsf{c}^-\langle \vec{x}, u\rangle := \mathsf{c}^-\langle \vec{x}(0), u\rangle :: \cdots :: \mathsf{c}^-\langle \vec{x}(l-1), u\rangle :: \iota$$
$$\mathsf{c}^+\langle x, \lambda\vec{y}.b\rangle := \lambda\vec{y}.x\,\mathsf{c}^-\langle \vec{y}, b\rangle \qquad \mathsf{c}^+\langle x, [m_1, \dots, m_k]\rangle := [\mathsf{c}^+\langle x, m_1\rangle, \dots, \mathsf{c}^+\langle x, m_k\rangle]$$
$$\mathsf{c}^+\langle \vec{x}, \iota\rangle := \iota \qquad \mathsf{c}^+\langle \vec{x}, \bar{m} :: \vec{n}\rangle := \mathsf{c}^+\langle \vec{x}(0), \bar{m}\rangle :: \mathsf{c}^+\langle \vec{x}, \vec{n}\rangle[\vec{x}\uparrow]$$
$$\mathsf{c}\langle x, \bar{m}\rangle := \lambda\vec{y}.x\,\mathsf{c}^+\langle \vec{y}, \bar{m}\rangle$$

**Figure 12.** Definition of copycat terms (assuming $y \neq x$, and choosing $\vec{y} \not\ni x$ whenever relevant; and choosing $l$ such that no $\vec{x}(i)$ with $i \geq l$ is free in $u$ in the definition of $\mathsf{c}^-\langle \vec{x}, u\rangle$)

$\mathsf{c}^+\langle x, \bar{m}\rangle \in x^!$, $\mathsf{c}^+\langle \vec{x}, \vec{m}\rangle \in \vec{x}^{\,!}$ and $\mathsf{c}\langle x, \bar{m}\rangle \in x^\eta$, that we call **copycat terms:**[17] we summarize their mutually inductive definitions in Figure 12, so that it is safe to skip the proof.

**PROOF OF LEMMA 5.4.** Note that Item 2 is obtained by iterating Item 1. Indeed, if $\vec{p} = \langle \bar{p}_i\rangle_{i\in\mathbb{N}} \in \vec{x}^{\,!}$ and $k \in \mathbb{N}$ is such that $\vec{x}(i) \notin \mathcal{V}(u)$ when $i \geq k$, then $u[\vec{p}/\vec{x}] \to_{\mathrm{r}}^* 0$ as soon as $\bar{p}_i \neq [\,]$ for some $i \geq k$. And, otherwise, $u[\vec{p}/\vec{x}] = u[\bar{p}_0/\vec{x}(0)] \cdots [\bar{p}_{k-1}/\vec{x}(k-1)]$ — observing that no $\vec{x}(i)$ is free in $\bar{p}_j$ when $i \neq j$. We then apply Item 1 to the same term $u$ and each $\vec{x}(i)$ for $i < k$, and write

$$u' := u[\mathsf{c}^-\langle \vec{x}(0), u\rangle/\vec{x}(0)] \cdots [\mathsf{c}^-\langle \vec{x}(k-1), u\rangle/\vec{x}(k-1)]$$

and

$$\mathsf{c}^-\langle \vec{x}, u\rangle := \mathsf{c}^-\langle \vec{x}(0), u\rangle :: \cdots :: \mathsf{c}^-\langle \vec{x}(k-1), u\rangle :: \iota$$

so that $u' = u[\mathsf{c}^-\langle \vec{x}, u\rangle/\vec{x}]$. By the properties of each $\mathsf{c}^-\langle \vec{x}(i), u\rangle$, together with Lemma 3.14, we obtain $u' \to_{\mathrm{r}}^* (\prod_{i=0}^{k-1} \mathsf{m}(\mathsf{c}^-\langle \vec{x}(i), u\rangle))u = \mathsf{m}(\mathsf{c}^-\langle \vec{x}, u\rangle)u$. Moreover, we have $u[\bar{p}/\vec{x}(i)] \to_{\mathrm{r}}^* 0$ for any $\bar{p} \in \vec{x}(i)^!$ other than $\mathsf{c}^-\langle \vec{x}(i), u\rangle$, for $0 \leq i < k$: recalling that we have $u[\bar{p}/\vec{x}(i)] \to_{\mathrm{r}}^* 0$ for any $\bar{p} \neq [\,]$ for $i \geq k$, $\mathsf{c}^-\langle \vec{x}, u\rangle$ is the only $\vec{p} \in \vec{x}^{\,!}$ with $\mathcal{N}(u[\vec{p}/\vec{x}]) \neq 0$.

So it is sufficient to establish Item 1, and possibly one of Items 3 to 5, depending on the shape of $u$, simultaneously by induction on $u$.

**Value terms:** If $u = m \in \Delta_{\mathrm{v}}$, we prove Items 1 and 3. We can write $m = \lambda\vec{y}.b$, assuming w.l.o.g. that $x \notin \vec{y}$. Applying the induction hypothesis (Item 1) to $b$ entails $b[\mathsf{c}^-\langle x, b\rangle/x] \to_{\mathrm{r}}^* \mathsf{m}(\mathsf{c}^-\langle x, b\rangle)b$ and $b[\bar{p}/x] \to_{\mathrm{r}}^* 0$ for any other $\bar{p} \in x^!$: we set $\mathsf{c}^-\langle x, m\rangle := \mathsf{c}^-\langle x, b\rangle$, and deduce Item 1 for $m$ by Lemma 3.13, observing that $m[\bar{p}/x] = \lambda\vec{y}.b[\bar{p}/x]$ for any $\bar{p} \in \Delta_!$. Moreover,

---

[17] Although we do not develop the analogy further, a game semanticist reader may infer a precise correspondence between copycat terms and the isomorphism classes of augmentations in a copycat strategy over the universal arena, in light of Section 8.

we can write $x^{\eta} = \lambda\vec{y}.x\,\vec{y}^{\,!}$: for each $p \in x^{\eta}$, we have $p = \lambda\vec{y}.x\,\vec{p}$ with $\vec{p} \in \vec{y}^{\,!}$. The induction hypothesis (Item 2) yields $b[\mathsf{c}^{-}\langle\vec{y}, b\rangle/\vec{y}] \to_{\mathrm{r}}^{*} \mathsf{m}(\mathsf{c}^{-}\langle\vec{y}, b\rangle)b$, and $b[\vec{p}/\vec{y}] \to_{\mathrm{r}}^{*} 0$ for any other $\vec{p} \in \vec{y}^{\,!}$. We set $\mathsf{c}^{+}\langle x, m\rangle := \lambda\vec{y}.x\,\mathsf{c}^{-}\langle\vec{y}, b\rangle \in x^{\eta}$, and obtain Item 3 by Lemma 3.13, observing that $(\lambda\vec{y}.x\,\vec{p})\{m/x\} = \lambda\vec{y}.m\,\vec{p} \to_{\mathrm{r}} \lambda\vec{y}.b[\vec{p}/\vec{y}]$ for any $\vec{p} \in \Delta_{\mathrm{s}}$.

**Base terms:** If $u = a \in \Delta_{\mathrm{b}}$, we only have to prove Item 1. There are three possible cases: either $a = y\,\vec{n}$ with $y \neq x$, or $a = x\,\vec{n}$, or $a = m\,\vec{n}$. If $a = y\,\vec{n}$ with $y \neq x$, we apply the induction hypothesis (Item 1) to $\vec{n}$, set $\mathsf{c}^{-}\langle x, a\rangle := \mathsf{c}^{-}\langle x, \vec{n}\rangle$, and conclude as in the abstraction case.

If $a = x\,\vec{n}$, then for each $\bar{p} = [p_1, \ldots, p_k] \in x^{!}$, we have $a[\bar{p}/x] = 0$ if $k = 0$ and, otherwise, $a[\bar{p}/x] = \sum_{i=1}^{k} p_i\,\vec{n}[\bar{p}'_i/x]$ where each $\bar{p}'_i$ is such that $\bar{p} = [p_i] * \bar{p}'_i$. The induction hypothesis (Items 1 and 5) applied to $\vec{n}$ yields $\mathsf{c}^{-}\langle x, \vec{n}\rangle \in x^{!}$ and $\mathsf{c}\langle x, \vec{n}\rangle \in x^{\eta}$, and we set $\mathsf{c}^{-}\langle x, a\rangle := [\mathsf{c}\langle x, \vec{n}\rangle] * \mathsf{c}^{-}\langle x, \vec{n}\rangle \in x^{!}$. If $\bar{p} \neq \mathsf{c}^{-}\langle x, a\rangle$, then each element in the previous sum normalizes to 0. And, if $\bar{p} = \mathsf{c}^{-}\langle x, a\rangle$, we obtain $a[\bar{p}/x] \to_{\mathrm{r}}^{*} (l \times \mathsf{m}(\mathsf{c}\langle x, \vec{n}\rangle) \times \mathsf{m}(\mathsf{c}^{-}\langle x, \vec{n}\rangle))a$ where $l = \#\{i \mid p_i = \mathsf{c}\langle x, \vec{n}\rangle\}$. Fact 2.1 entails $\mathsf{m}(\mathsf{c}^{-}\langle x, a\rangle) = l \times \mathsf{m}(\mathsf{c}\langle x, \vec{n}\rangle) \times \mathsf{m}(\mathsf{c}^{-}\langle x, \vec{n}\rangle)$, and we obtain Item 1 for $a$.

If $a = m\,\vec{n}$, then for each $\bar{p} \in x^{!}$, we have $a[\bar{p}/x] = \sum_{\bar{p} \triangleleft \bar{p}_1 * \bar{p}_2} m[\bar{p}_1/x]\,\vec{n}[\bar{p}_2/x]$. The induction hypothesis (Item 1) applied to $m$ and $\vec{n}$ yields $\mathsf{c}^{-}\langle x, m\rangle$ and $\mathsf{c}^{-}\langle x, \vec{n}\rangle \in x^{!}$, and we set $\mathsf{c}^{-}\langle x, a\rangle := \mathsf{c}^{-}\langle x, m\rangle * \mathsf{c}^{-}\langle x, \vec{n}\rangle \in x^{!}$. If $\bar{p} \neq \mathsf{c}^{-}\langle x, a\rangle$, then each element in the previous sum normalizes to 0. And, if $\bar{p} = \mathsf{c}^{-}\langle x, a\rangle$, we obtain $a[\bar{p}/x] \to_{\mathrm{r}}^{*} \big(l \times \mathsf{m}(\mathsf{c}^{-}\langle x, m\rangle) \times \mathsf{m}(\mathsf{c}^{-}\langle x, \vec{n}\rangle)\big)a$ where $l = \#\{p : \bar{p} \triangleleft 2 \mid \bar{p} \upharpoonright p^{-1}(1) = \mathsf{c}^{-}\langle x, m\rangle\}$. Again, we apply Fact 2.1 to check that $\mathsf{m}(\mathsf{c}^{-}\langle x, a\rangle) = l \times \mathsf{m}(\mathsf{c}^{-}\langle x, m\rangle) \times \mathsf{m}(\mathsf{c}^{-}\langle x, \vec{n}\rangle)$, which yields Item 1 for $a$.

**Bag terms:** If $u = \bar{m} \in \Delta_{!}$, we prove Items 1 and 4. We can write $\bar{m} = [m_1, \ldots, m_k]$, and $\bar{m}[\bar{p}/x] = \sum_{\bar{p} \triangleleft \bar{p}_1 * \ldots * \bar{p}_k} [m_1[\bar{p}_1/x], \ldots, m_k[\bar{p}_k/x]]$. The induction hypothesis (Item 1) applied to each $m_i$ yields $\mathsf{c}^{-}\langle x, m_i\rangle \in x^{!}$ and we set $\mathsf{c}^{-}\langle x, \bar{m}\rangle := \mathsf{c}^{-}\langle x, m_1\rangle * \cdots * \mathsf{c}^{-}\langle x, m_k\rangle$. If $\bar{p} \neq \mathsf{c}^{-}\langle x, \bar{m}\rangle$, then each element in the previous sum normalizes to 0. And, if $\bar{p} = \mathsf{c}^{-}\langle x, \bar{m}\rangle$, we obtain $a[\bar{p}/x] \to_{\mathrm{r}}^{*} (l \times \prod_{i=1}^{k} \mathsf{m}(\mathsf{c}^{-}\langle x, m_i\rangle))\bar{m}$ where $l = \#\{p : \bar{p} \triangleleft k \mid \bar{p} \upharpoonright p^{-1}(i) = \mathsf{c}^{-}\langle x, m_i\rangle \text{ for } 1 \leq i \leq k\}$. Item 1 for $\bar{m}$ follows, again applying Fact 2.1. The induction hypothesis (Item 3) applied to each $m_i$ also yields $\mathsf{c}^{+}\langle x, m_i\rangle \in x^{\eta}$ and we set $\mathsf{c}^{+}\langle x, \bar{m}\rangle := [\mathsf{c}^{+}\langle x, m_1\rangle, \ldots, \mathsf{c}^{+}\langle x, m_k\rangle] \in x^{!}$. Assume $\bar{p} = [p_1, \ldots, p_l] \in x^{!}$. Since $x$ occurs exactly once in each $p_i$, we have $\bar{p}[\bar{m}/x] \to_{\mathrm{r}}^{*} 0$ when $k \neq l$. And if $k = l$, we have

$$\bar{p}[\bar{m}/x] = \sum_{\sigma \in \mathbb{S}_k} [p_1\{m_{\sigma(1)}/x\}, \ldots, p_1\{m_{\sigma(k)}/x\}] \,.$$

Hence, if $\bar{p} \neq \mathsf{c}^{+}\langle x, \bar{m}\rangle$, then each element in the previous sum normalizes to 0. And, if $\bar{p} = \mathsf{c}^{+}\langle x, \bar{m}\rangle$, we obtain $a[\bar{p}/x] \to_{\mathrm{r}}^{*} (l \times \prod_{i=1}^{k} \mathsf{m}(\mathsf{c}^{+}\langle x, m_i\rangle))\bar{m}$ where $l = \#\{\sigma \in \mathbb{S}_k \mid p_i = \mathsf{c}^{+}\langle x, m_{\sigma(i)}\rangle \text{ for } 1 \leq i \leq k\} = \mathsf{d}(\mathsf{c}^{+}\langle x, \bar{m}\rangle)$. Item 4 for $\bar{m}$ follows since $l \times \prod_{i=1}^{k} \mathsf{m}(\mathsf{c}^{+}\langle x, m_i\rangle) = \mathsf{m}(\mathsf{c}^{+}\langle x, \bar{m}\rangle)$ by definition.

**Stream terms:** Finally, if $u = \vec{m} \in \Delta_{\mathrm{s}}$ then we prove Items 1 and 5. Note that the first statement of Item 5 entails the second one. Indeed, if $p \in x^{\eta}$ then we can write $p = \lambda\vec{y}.x\,\vec{p}$ with $\vec{p} \in \vec{y}^{\,!}$, and then $p\,\vec{m} \to_{\mathrm{r}} x\,\vec{p}[\vec{m}/\vec{y}]$: we set $\mathsf{c}\langle x, \vec{m}\rangle := \lambda\vec{y}.x\,\mathsf{c}^{+}\langle\vec{y}, \vec{m}\rangle$.

In case $u = \iota$, Items 1 and 5 are straightforward, with $\mathsf{c}^{-}\langle x, \iota\rangle := [\,]$, and $\mathsf{c}^{+}\langle\vec{x}, \iota\rangle := \iota$.

It remains only to establish Item 1 and the first statement of Item 5 for $u = \bar{m} :: \vec{n} \neq \iota$, assuming the induction hypothesis holds for $\bar{m}$ and $\vec{n}$. We obtain Item 1 from the induction hypothesis (Item 1) applied to $\bar{m}$ and $\vec{n}$, similarly to the case of $u = m\,\vec{n}$. By Lemma 5.1, for any $\vec{p} \in \vec{x}^{\,!}$, we can write $\vec{p} = \bar{p}_0 :: \vec{p}\,'[\vec{x}\uparrow]$ with $\bar{p}_0 \in \vec{x}(0)^{!}$ and $\vec{p}\,' \in \vec{x}^{\,!}$, to obtain $\vec{p}[u/\vec{x}] = \bar{p}_0[\bar{m}/\vec{x}(0)] :: \vec{p}\,'[\vec{n}/\vec{x}]$. We apply the induction hypothesis to $\bar{m}$ (Item 4) and $\vec{n}$ (Item 5) to obtain $\mathsf{c}^{+}\langle\vec{x}(0), \bar{m}\rangle$ and $\mathsf{c}^{+}\langle\vec{x}, \vec{n}\rangle$. We set $\mathsf{c}^{+}\langle\vec{x}, u\rangle := \mathsf{c}^{+}\langle\vec{x}(0), \bar{m}\rangle :: \mathsf{c}^{+}\langle\vec{x}, \vec{n}\rangle[\vec{x}\uparrow] \in \vec{x}^{\,!}$ (Lemma 5.1 again). We have $\mathsf{m}(\mathsf{c}^{+}\langle\vec{x}, u\rangle)u = \mathsf{m}(\mathsf{c}^{+}\langle\vec{x}(0), \bar{m}\rangle)\times\mathsf{m}(\mathsf{c}^{+}\langle\vec{x}, \vec{n}\rangle[\vec{x}\uparrow]) = \mathsf{m}(\mathsf{c}^{+}\langle\vec{x}(0), \bar{m}\rangle)\times\mathsf{m}(\mathsf{c}^{+}\langle\vec{x}, \vec{n}\rangle)$ (shifts clearly preserve the isotropy degree). Finally, we apply Lemma 3.19 to obtain $\mathsf{c}^{+}\langle\vec{x}, u\rangle[u/\vec{x}] = \mathsf{c}^{+}\langle\vec{x}(0), \bar{m}\rangle[\bar{m}/\vec{x}(0)] :: \mathsf{c}^{+}\langle\vec{x}, \vec{n}\rangle[\vec{n}/\vec{x}] \to_{\mathrm{r}}^{*} (\mathsf{m}(\mathsf{c}^{+}\langle\vec{x}(0), \bar{m}\rangle) \times \mathsf{m}(\mathsf{c}^{+}\langle\vec{x}, \vec{n}\rangle))(\bar{m} :: \vec{n}) = \mathsf{m}(\mathsf{c}^{+}\langle\vec{x}, u\rangle)u$, and $\vec{p}[u/\vec{x}] \to_{\mathrm{r}}^{*} 0$ for any other $\vec{p} \in \vec{x}^{\,!}$. ■

Observe that the use of iterated reduction $\to_{\mathrm{r}}^{*}$ is crucial in the previous proof: in particular, to obtain Item 3 in the value case, we define $\mathsf{c}^{+}\langle x, \lambda\vec{y}.b\rangle := \lambda\vec{y}.x\,\mathsf{c}^{-}\langle\vec{y}, b\rangle$, and use one step of resource reduction to apply Item 1 to the underlying base term; and to obtain the second part of Item 5, we define $\mathsf{c}\langle x, \vec{m}\rangle := \lambda\vec{y}.x\,\mathsf{c}^{+}\langle\vec{y}, \vec{m}\rangle$, and use one step of resource reduction to apply the first part; since each item inductively depends on the others, we do need to iterate those reductions.

A straightforward induction allows checking that $|\mathsf{c}^{-}\langle x, u\rangle| = |u|_x$: in particular, if $x \notin \mathcal{V}(u)$, then $\mathsf{c}^{-}\langle x, u\rangle = [\,]$. Moreover, if $\vec{x} \notin \mathcal{V}_{\mathrm{s}}(u)$ then $\mathsf{c}^{-}\langle\vec{x}, u\rangle = \iota$ directly by definition.

**EXAMPLE 5.5.** The reader may check that $\mathsf{c}\langle x, \iota\rangle = \mathsf{p}_x$, hence $\mathsf{c}^{-}\langle x, \mathsf{p}_x\rangle = \mathsf{c}^{-}\langle x, x\,\iota\rangle = [\mathsf{p}_x]$. Indeed, $\mathsf{p}_x\iota \to_{\mathrm{r}} x\,\iota$ and $\mathsf{p}_x\{\mathsf{p}_x/x\} = \mathsf{p}_x[[\mathsf{p}_x]/x] \to_{\mathrm{r}} \mathsf{p}_x$ as in Example 3.12. Moreover, $\mathsf{c}^{-}\langle x, y\,\iota\rangle = [\,]$ when $x \neq y$, hence $\mathsf{c}^{-}\langle\vec{y}, \vec{y}(0)\,\iota\rangle = [\mathsf{p}_{\vec{y}(0)}] :: \iota$ and $\mathsf{c}^{-}\langle\vec{y}, x\,\iota\rangle = \iota$ when $x \notin \vec{y}$. We obtain $\mathsf{c}^{+}\langle x, \mathsf{p}_0\rangle = \lambda\vec{y}.x\,[\mathsf{p}_{\vec{y}(0)}] :: \iota = \mathsf{c}_x$ (as defined in Example 3.1) and $\mathsf{c}^{+}\langle y, \mathsf{p}_x\rangle = \mathsf{p}_y$ for any (non-necessarily distinct) variables $x$ and $y$. And, indeed, $\mathsf{c}_x\{\mathsf{p}_0/x\} = \lambda\vec{y}.\mathsf{p}_0\,[\mathsf{p}_{\vec{y}(0)}] :: \iota \to_{\mathrm{r}}^{*} \mathsf{p}_0$ as in Example 3.12, and we have already recalled that $\mathsf{p}_y\{\mathsf{p}_x/y\} \to_{\mathrm{r}} \mathsf{p}_x$.

We also obtain $\mathsf{c}\langle x, [\mathsf{p}_y] :: \iota\rangle = \lambda\vec{z}.x\,\mathsf{c}^{+}\langle\vec{z}, [\mathsf{p}_y] :: \iota\rangle = \lambda\vec{z}.x\,[\mathsf{c}^{+}\langle\vec{z}(0), \mathsf{p}_y\rangle] :: \iota = \lambda\vec{z}.x\,[\mathsf{p}_{\vec{z}(0)}] :: \iota = \lambda y.\lambda\vec{z}.x\,[\mathsf{p}_y] :: \iota = \mathsf{c}_x$ for any variables $x$ and $y$. And, indeed, $\mathsf{c}_x\,[\mathsf{p}_y] :: \iota \to_{\mathrm{r}}^{*} (x\,[\mathsf{p}_y] :: \iota)[[\mathsf{p}_y]/y] = x\,[\mathsf{p}_y[\mathsf{p}_y/y]] :: \iota \to_{\mathrm{r}}^{*} x\,[\mathsf{p}_y] :: \iota$.

Now consider the resource sequence $\vec{m} = [\mathsf{p}_0, \mathsf{p}_0] :: \iota = \mathsf{p}_0^2 :: \iota$ with $\mathsf{m}(\vec{m}) = 2$. We have $\mathsf{c}^{+}\langle\vec{y}, \vec{m}\rangle = \mathsf{c}^{+}\langle\vec{y}(0), \mathsf{p}_0^2\rangle :: \iota = \mathsf{c}^{+}\langle\vec{y}(0), \mathsf{p}_0\rangle^2 :: \iota = \mathsf{c}_{\vec{y}(0)}^2 :: \iota$. So $\mathsf{c}\langle x, \vec{m}\rangle = \lambda\vec{y}.x\,\mathsf{c}_{\vec{y}(0)}^2 :: \iota = \mathsf{c}'_x$. And, indeed, $\mathsf{c}\langle x, \vec{m}\rangle\,\vec{m} = \mathsf{c}'_x\,\mathsf{p}_0^2 :: \iota \to_{\mathrm{r}}^{*} 2x\,\mathsf{p}_0^2 :: \iota = \mathsf{m}(\vec{m})x\,\vec{m}$ as in Example 3.12. ◆

**REMARK 5.6.** It is no accident that $\mathsf{c}^{-}\langle x, \mathsf{p}_x\rangle = [\mathsf{p}_x]$ and $\mathsf{c}^{+}\langle y, \mathsf{p}_x\rangle = \mathsf{p}_y = \mathsf{p}_x\{y/x\}$ in Example 5.5: as a general fact, elements of the expansion of variables are their own copycat terms.

More precisely, one can prove that, for any value variables $x$ and $y$ and any sequence variables $\vec{x}$ and $\vec{y}$, we have:

1. if $m \in x^\eta$ then $\mathsf{c}^-\langle x, m\rangle = [m]$ and $\mathsf{c}^+\langle y, m\rangle = m\{y/x\}$;
2. if $\bar{m} \in x^!$ then $\mathsf{c}^-\langle x, \bar{m}\rangle = \bar{m}$ and $\mathsf{c}^+\langle y, \bar{m}\rangle = \bar{m}\{y/x\}$;
3. if $\vec{m} \in \vec{x}^{\,!}$ then $\mathsf{c}^-\langle \vec{x}, \vec{m}\rangle = \vec{m}$, $\mathsf{c}^+\langle \vec{y}, \vec{m}\rangle = \vec{m}\{\vec{y}/\vec{x}\}$ and $\mathsf{c}\langle y, \vec{m}\rangle = \lambda\vec{x}.y\,\vec{m}$.

We do not develop this result further as it plays no role in the sequel. A game semanticist reader detailing the proof might be reminded of the composition of copycat plays.

**LEMMA 5.7.** *For any resource term $u$, any value variable $x$, and any sequence variable $\vec{x}$, the following holds:*

1. $u\{x^\eta/x\} \rightsquigarrow u$;
2. $u\{\vec{x}^{\,\eta}/\vec{x}\} \rightsquigarrow u$;
3. *if* $u = m \in \Delta_{\mathrm{v}}$ *then* $x^\eta\{m/x\} \rightsquigarrow m$;
4. *if* $u = \bar{m} \in \Delta_!$ *then* $x^![\bar{m}/x] \rightsquigarrow \bar{m}$;
5. *if* $u = \vec{m} \in \Delta_{\mathrm{s}}$ *then* $\vec{x}^{\,!}[\vec{m}/\vec{x}] \rightsquigarrow \vec{m}$ *and* $x^\eta\,\vec{m} \rightsquigarrow x\,\vec{m}$;
6. *if* $u = a \in \Delta_{\mathrm{b}}$ *then* $(\lambda\vec{x}.a)\,\vec{x}^{\,!} \rightsquigarrow a$.

**PROOF.** Items 1 to 5 follow from the corresponding items of Lemma 5.4, combined with Lemma 5.2. For Item 6, write $(\lambda\vec{x}.a)\,\vec{x}^{\,!} = \sum_{\vec{p}\in\vec{x}^{\,!}} \frac{1}{\mathsf{m}(\vec{p})}(\lambda\vec{x}.a)\,\vec{p}$ thanks to Lemma 5.2, observe that $(\lambda\vec{x}.a)\,\vec{p} \mapsto_{\mathrm{r}} a[\vec{p}/x]$ and conclude by Item 2 of Lemma 5.4. ■

**LEMMA 5.8.** *For any term vector $U$, any value variable $x$, and any sequence variable $\vec{x}$, the following holds:*

1. $U\{x^\eta/x\} \rightsquigarrow U$ *and, for any variable $y$,* $U\{x^\eta/y\} \rightsquigarrow U\{x/y\}$;
2. $U\{\vec{x}^{\,\eta}/\vec{x}\} \rightsquigarrow U$ *and, for any sequence variable $\vec{y}$,* $U\{\vec{x}^{\,\eta}/\vec{y}\} \rightsquigarrow U\{\vec{x}/\vec{y}\}$;
3. *if* $U = M \in \mathbb{K}\langle\Delta_{\mathrm{v}}\rangle$ *then* $x^\eta\{M/x\} \rightsquigarrow M$*; moreover, if* $\vec{x} \notin \mathcal{V}_{\mathrm{s}}(M)$*, then* $\lambda\vec{x}.M\,\vec{x}^{\,!} \rightsquigarrow M$;
4. *if* $U = \bar{M} \in \mathbb{K}\langle\Delta_!\rangle$ *then* $x^![\bar{M}/x] \rightsquigarrow \bar{M}$;
5. *if* $U = \vec{M} \in \mathbb{K}\langle\Delta_{\mathrm{s}}\rangle$ *then* $\vec{x}^{\,!}[\vec{M}/\vec{x}] \rightsquigarrow \vec{M}$ *and* $x^\eta\,\vec{M} \rightsquigarrow x\,\vec{M}$.
6. *if* $U = A \in \mathbb{K}\langle\Delta_{\mathrm{b}}\rangle$ *then* $(\lambda\vec{x}.A)\,\vec{x}^{\,!} \rightsquigarrow A$ *and, for any sequence variable $\vec{y}$,* $(\lambda\vec{y}.A)\,\vec{x}^{\,!} \rightsquigarrow A\{\vec{x}/\vec{y}\}$.

*Finally, for any sequence $\vec{M}$ of value vectors such that $\mathcal{V}_{\mathrm{s}}(\vec{M})$ is finite, we have* $\vec{x}^{\,!}\{\vec{M}/\vec{x}\} \rightsquigarrow \vec{M}^!$.

**PROOF.** Except for the second part of Item 3, Items 1 to 6 follow directly from the corresponding items of Lemma 5.7 by linearity and compatibility. If $M \in \mathbb{K}\langle\Delta_{\mathrm{v}}\rangle$ and $\vec{x} \notin \mathcal{V}_{\mathrm{s}}(M)$, we can write $M = \lambda\vec{x}.A$, and we obtain $\lambda\vec{x}.M\,\vec{x}^{\,!} \rightsquigarrow M$ by Item 6. Finally, for any sequence $\vec{M}$ of value vectors such that $\mathcal{V}_{\mathrm{s}}(\vec{M})$ is finite, Lemma 4.5 gives $\vec{x}^{\,!}\{\vec{M}/\vec{x}\} = \vec{x}^{\,!}[\vec{M}^!/\vec{x}]$ and we conclude by Item 5. ■

### 5.2 Two flavours of extensional Taylor expansion

Write $\Lambda$ for the set of pure, ordinary $\lambda$-terms, that we denote by letters $M, N, P$, and write $\to_\beta$ (resp. $\to_\eta$) for the $\beta$-reduction (resp. $\eta$-reduction) relation, induced by the application of the base case $(\lambda x.M)\, N \to_\beta M\{N/x\}$ (resp. $\lambda x.M\, x \to_\eta M$ when $x$ fresh) in any context. We write $\to_{\beta\eta}$ for their union. We may also write $M \leftarrow_\eta N$ when $N \to_\eta M$.

**EXAMPLE 5.9.** We will use the following $\lambda$-terms as running examples:

- the identity $\mathbf{I} := \lambda x.x$;
- the Church numeral $\mathbf{1} := \lambda x.\lambda y.x\, y$;
- the term $\mathbf{J} = \mathbf{R_J}\, \mathbf{R_J}$ with $\mathbf{R_J} := \lambda z.\lambda x.\lambda y.x\, (z\, z\, y)$ (a version of Wadsworth's $\mathbf{J}$ combinator).

Observe that $\mathbf{I} \leftarrow_\eta \mathbf{1}$. Moreover, $\mathbf{J} \to_\beta \lambda x.\lambda y.x\, (\mathbf{J}\, y)$, hence $\mathbf{J}\, y \to_\beta^2 \lambda y'.y\, (\mathbf{J}\, y')$. It follows that

$$\mathbf{J} \to_\beta \lambda x.\lambda y_0.x\, (\mathbf{J}\, y_0) \to_\beta^2 \lambda x.\lambda y_0.x\, (\lambda y_1.y_0\, (\mathbf{J}\, y_1)) \to_\beta^2 \lambda x.\lambda y_0.x\, (\lambda y_1.y_0\, (\lambda y_2.y_1\, (\mathbf{J}\, y_2))) \to_\beta^2 \cdots$$

to be compared with nested $\eta$-expansions of $\mathbf{I}$:

$$\mathbf{I} \leftarrow_\eta \lambda x.\lambda y_0.x\, y_0 \leftarrow_\eta \lambda x.\lambda y_0.x\, (\lambda y_1.y_0\, y_1) \leftarrow_\eta \lambda x.\lambda y_0.x\, (\lambda y_1.y_0\, (\lambda y_2.y_1\, y_2)) \leftarrow_\eta \cdots$$

Intuitively (and this can be made formal), both sequences converge to the same infinite tree, which is the Böhm tree of $\mathbf{J}$: in that sense, $\mathbf{J}$ computes the result of infinitely many nested $\eta$-expansions of the identity. ◆

We define the **structural Taylor expansion** $\mathcal{T}_\eta(M)$ of an ordinary $\lambda$-term $M$ by induction on $M$ as follows:

$$\mathcal{T}_\eta(x) := x^\eta \qquad \mathcal{T}_\eta(\lambda x.M) := \lambda x.\mathcal{T}_\eta(M) \qquad \mathcal{T}_\eta(M\, N) := \lambda \vec{y}.\mathcal{T}_\eta(M)\, \mathcal{T}_\eta^!(N) :: \vec{y}^{\,!}$$

together with $\mathcal{T}_\eta^!(M) := \mathcal{T}_\eta(M)^!$, where $\vec{y}$ is chosen fresh in the application case.

The **head Taylor expansion** $\mathcal{T}_h(M)$ is defined inductively on the head structure of $M \in \Lambda$, for which we first need to introduce some notations. Given a sequence $\vec{N} = \langle N_1, \dots, N_k \rangle$ of $\lambda$-terms, we write the **iterated application** $M\, \vec{N} := M\, N_1 \cdots N_k$. Similarly, if $\vec{N} = \langle \bar{N}_1, \dots, \bar{N}_k \rangle \in \mathbb{K}\langle \Delta_! \rangle^k$ is a sequence of bag vectors and $\vec{M} \in \mathbb{K}\langle \Delta_s \rangle$ is a stream vector, we write the **concatenation** $\vec{N}\, \vec{M} := \bar{N}_1 :: \cdots :: \bar{N}_k :: \vec{M}$.

Recall that if $M$ is a $\lambda$-term, then:

- either $M$ is an abstraction;
- or we can write $M = x\, \vec{N}$;
- or we can write $M = P\, \vec{N}$ where $P$ is an abstraction and $\vec{N} \neq \varepsilon$.

Then we define:

$$\mathcal{T}_h(\lambda x.M) := \lambda x.\mathcal{T}_h(M) \qquad \mathcal{T}_h(x\, \vec{N}) := \lambda \vec{y}.x\, \mathcal{T}_h^!(\vec{N})\, \vec{y}^{\,!} \qquad \mathcal{T}_h(P\, \vec{N}) := \lambda \vec{y}.\mathcal{T}_h(P)\, \mathcal{T}_h^!(\vec{N})\, \vec{y}^{\,!}$$

together with $\mathcal{T}^!_h(M) := \mathcal{T}_h(M)^!$ and $\mathcal{T}^!_h(\langle M_1, \dots, M_k\rangle) := \langle \mathcal{T}^!_h(M_1), \dots, \mathcal{T}^!_h(M_k)\rangle$, choosing $\vec{y}$ fresh in the last two cases, and assuming $P$ is an abstraction and $\vec{N} \neq \varepsilon$ in the application case. We may also write $\mathcal{T}_h(\langle M_1, \dots, M_k\rangle) := \langle \mathcal{T}_h(M_1), \dots, \mathcal{T}_h(M_k)\rangle$, a tuple of value vectors. Observe that $\mathcal{T}_h(x) = \mathcal{T}_h(x\,\varepsilon) = \lambda\vec{y}.x\,\varepsilon\,\vec{y}^{\,!} = x^\eta = \mathcal{T}_\eta(x)$.

Finally, if $\vec{x} = \langle x_1, \dots, x_k\rangle$ we write $\lambda\vec{x}.M := \lambda x_1.\cdots\lambda x_k.M$ for $M$ a $\lambda$-term or a value vector, so that: $\mathcal{T}_\eta(\lambda\vec{x}.M) = \lambda\vec{x}.\mathcal{T}_\eta(M)$ and $\mathcal{T}_h(\lambda\vec{x}.M) = \lambda\vec{x}.\mathcal{T}_h(M)$.

Although this plays no particular role in the present section, as we do not rely on uniformity in our study of extensional Taylor expansion, we obtain the same characterization of coefficients as for ordinary Taylor expansion (the multiplicity degree $\mathsf{m}(m)$ was defined on Page 46):[18]

**LEMMA 5.10.** *If $m \in \mathcal{T}_\eta(M)$ (resp. $m \in \mathcal{T}_h(M)$) then $\mathcal{T}_\eta(M).m = \frac{1}{\mathsf{m}(m)}$ (resp. $\mathcal{T}_h(M).m = \frac{1}{\mathsf{m}(m)}$).*

**PROOF.** As for Lemma 5.2, the proof is straightforward by induction on terms, using Lemma 4.3 in the case of a bag. ■

**EXAMPLE 5.11.** We have $\mathcal{T}_\eta(\mathbf{I}) = \mathcal{T}_h(\mathbf{I})$: indeed $\mathcal{T}_h(\mathbf{I}) = \lambda x.\lambda\vec{y}.x\,\vec{y}^{\,!} = \lambda x.x^\eta = \mathcal{T}_\eta(\mathbf{I})$. The elements of $\mathcal{T}_\eta(\mathbf{I}) = \mathcal{T}_h(\mathbf{I})$ are the value terms of the shape $\lambda x.m$ with $m \in x^\eta$: the smallest such term is $\lambda x.\mathsf{p}_x = \mathsf{p}_0$, but we can consider arbitrarily complex ones such as $\lambda x.\mathsf{c}_x = \lambda\vec{y}.\vec{y}(0)\,[\mathsf{p}_{\vec{y}(1)}] :: \iota$, or $\lambda x.\mathsf{c}'_x = \lambda\vec{y}.\vec{y}(0)\,\mathsf{c}^2_{\vec{y}(1)} :: \iota$, all in normal form. This contrasts with ordinary Taylor expansion, where $\mathcal{T}(x) = x$ and $\mathcal{T}(\mathbf{I}) = \mathbf{I}$.

We have $\mathcal{T}_h(\mathbf{1}) = \lambda x.\lambda y.\lambda\vec{z}.x\,y^! :: \vec{z}^{\,!} = \lambda x.\lambda\vec{z}.x\,\vec{z}^{\,!} = \mathcal{T}_h(\mathbf{I})$. On the other hand, $\mathcal{T}_\eta(\mathbf{1}) = \lambda x.\lambda y.\lambda\vec{z}.x^\eta\,y^! :: \vec{z}^{\,!} = \lambda x.\lambda\vec{z}.x^\eta\,\vec{z}^{\,!}$. Its elements are of the shape $\lambda x.\lambda\vec{z}.m\,\vec{p}$ with $m \in x^\eta$ and $\vec{p} \in \vec{z}^{\,!}$: the smallest one is thus $\lambda x.\lambda\vec{z}.\mathsf{p}_x\,\iota$, whose normal form is $\mathsf{p}_0 \in \mathcal{T}_h(\mathbf{1})$. More generally, by Lemma 5.4, the normal form of such an element is $0$, unless $m = \mathsf{c}\langle x, \vec{p}\rangle$, in which case its normal form is $\mathsf{m}(\vec{p})\lambda x.\lambda\vec{z}.x\,\vec{p}$. This ensures that $\mathcal{T}_\eta(\mathbf{1}) \rightsquigarrow \mathcal{T}_h(\mathbf{1})$, a particular instance of Theorem 5.16 that we establish below.

For $\mathbf{J}$ and $\mathbf{R}_\mathbf{J}$, we will only consider head Taylor expansion to keep the notations manageable. We obtain $\mathcal{T}_h(\mathbf{J}) = \lambda\vec{z}.\mathcal{T}_h(\mathbf{R}_\mathbf{J})\,\mathcal{T}^!_h(\mathbf{R}_\mathbf{J}) :: \vec{z}^{\,!}$, where $\mathcal{T}_h(\mathbf{R}_\mathbf{J}) = \lambda z.\lambda x.\lambda y.\lambda\vec{z}_1.x\,\left(\lambda\vec{z}_2.z\,z^! :: y^! :: \vec{z}^{\,!}_2\right)^! :: \vec{z}^{\,!}_1$. Elements of $\mathcal{T}_h(\mathbf{R}_\mathbf{J})$ are all in normal form, and the smallest one is

$$\mathsf{r}_0 := \lambda z.\lambda x.\lambda y.\lambda\vec{z}_1.x\,\iota = \lambda z.\lambda x.\lambda\vec{z}.x\,\iota = \lambda z.\lambda x.\mathsf{p}_x = \lambda z.\mathsf{p}_0$$

so that the smallest element of $\mathcal{T}_h(\mathbf{J})$ is $\lambda\vec{z}.\mathsf{r}_0\,\iota$, for which we observe $\lambda\vec{z}.\mathsf{r}_0\,\iota \to_\mathrm{r} \lambda\vec{z}.\mathsf{p}_0\,\iota \to_\mathrm{r} 0$. To get a non-zero normal form, we can observe that $\mathsf{r}_0\,[\,] :: [\mathsf{p}_x] :: \iota \to_\mathrm{r} \mathsf{p}_0\,[\mathsf{p}_x] :: \iota \to^*_\mathrm{r} x\,\iota$ so that

$$\mathcal{T}_h(\mathbf{J}) \ni \lambda\vec{z}.\mathsf{r}_0\,[\,] :: [\mathsf{p}_{\vec{z}(0)}] :: \iota = \lambda x.\lambda\vec{z}.\mathsf{r}_0\,[\,] :: [\mathsf{p}_x] :: \iota \to^*_\mathrm{r} \lambda x.\lambda\vec{z}.x\,\iota = \mathsf{p}_0 \in \mathcal{T}_\eta(\mathbf{I})\ .$$

This suggests defining

$$\mathbf{r}_1 := \lambda z.\lambda x.\lambda y.\lambda\vec{z}_1.x\,\left[\lambda\vec{z}_2.z\,[\,] :: [\mathbf{p}_y] :: \iota\right] :: \iota \in \mathcal{T}_h(\mathbf{R}_\mathbf{J})$$

---

[18] This result, together with Lemma 6.8, will ensure that the $\lambda$-theory induced by the normal form of extensional Taylor expansion, to be considered in Section 6, does not depend on the choice of the semiring of coefficients: it is entirely determined by the support of extensional Taylor expansion.

to obtain (recalling from Example 5.5 that $\mathsf{c}\langle x, [\mathsf{p}_y] :: \iota\rangle = \mathsf{c}_x$):

$$\begin{aligned}
\mathcal{T}_h(\mathbf{J}) \ni \lambda\vec{z}.\mathsf{r}_1\,[\mathsf{r}_0] :: [\mathsf{c}_{\vec{z}(0)}] :: [\mathsf{p}_{\vec{z}(1)}] :: \iota \quad &= \quad \lambda x.\lambda y.\lambda\vec{z}.\mathsf{r}_1\,[\mathsf{r}_0] :: [\mathsf{c}_x] :: [\mathsf{p}_y] :: \iota \\
&\to_{\mathrm{r}}^{*} \quad \lambda x.\lambda y.\lambda\vec{z}_1.\mathsf{c}_x \left[\lambda\vec{z}_2.\mathsf{r}_0\,[\,] :: \left[\mathsf{p}_y[[\mathsf{p}_y]/y]\right] :: \iota\right] :: \iota \\
&\to_{\mathrm{r}}^{*} \quad \lambda x.\lambda y.\lambda\vec{z}_1.\mathsf{c}_x \left[\lambda\vec{z}_2.\mathsf{r}_0\,[\,] :: [\mathsf{p}_y] :: \iota\right] :: \iota \\
&\to_{\mathrm{r}}^{*} \quad \lambda x.\lambda y.\lambda\vec{z}_1.\mathsf{c}_x \left[\lambda\vec{z}_2.y\,\iota\right] :: \iota \\
&\to_{\mathrm{r}}^{*} \quad \lambda x.\lambda y.\lambda\vec{z}_1.x \left[\mathbf{p}_y\right] :: \iota \in \mathcal{T}_\eta(\mathbf{I}) \ .
\end{aligned}$$

These examples are hints to the fact that $\mathcal{T}_h(\mathbf{J}) \rightsquigarrow \mathcal{T}_\eta(\mathbf{I})$, as we will establish below (Example 5.18). ◆

We show that the reduction of resource vectors simulates both $\beta$-reduction and $\eta$-reduction, through structural Taylor expansion. The case of $\eta$-reduction is easy:

**THEOREM 5.12.** *If $M \to_\eta M'$ then $\mathcal{T}_\eta(M) \rightsquigarrow \mathcal{T}_\eta(M')$.*

**PROOF.** By compatibility, it is sufficient to consider the base case: $M = \lambda x.M'\,x$ with $x$ fresh. We have $\mathcal{T}_\eta(M) = \lambda x.\lambda\vec{y}.\mathcal{T}_\eta(M')\,(x^\eta)^! :: \vec{y}^{\,!} = \lambda\vec{y}.\mathcal{T}_\eta(M')\,(\vec{y}(0)^\eta)^! :: (\vec{y}^{\,!})[\vec{y}\uparrow] = \lambda\vec{y}.\mathcal{T}_\eta(M')\,\vec{y}^{\,!}$ by Lemma 5.1, and we conclude by Lemma 5.8. ■

To deal with $\beta$-reduction, we first need to consider the interplay between structural Taylor expansion and substitution. As we have announced, they do not commute on the nose, and the analogue of Equation (1) (Page 6) is a reduction rather than an identity:

**LEMMA 5.13.** *For all $\lambda$-terms $M$ and $N$, $\mathcal{T}_\eta(M)\{\mathcal{T}_\eta(N)/x\} \rightsquigarrow \mathcal{T}_\eta(M\{N/x\})$.*

**PROOF.** The proof is by induction on $M$.

If $M = x$ then $\mathcal{T}_\eta(M)\{\mathcal{T}_\eta(N)/x\} = x^\eta\{\mathcal{T}_\eta(N)/x\} \rightsquigarrow \mathcal{T}_\eta(N)$ by Lemma 5.8 and we conclude since $M\{N/x\} = N$.

If $M = y \neq x$ then $\mathcal{T}_\eta(M)\{\mathcal{T}_\eta(N)/x\} = y^\eta\{\mathcal{T}_\eta(N)/x\} = y^\eta = \mathcal{T}_\eta(y)$ and we conclude since $M\{N/x\} = y$.

If $M = \lambda z.M'$ (choosing $z \notin \mathcal{V}(N) \cup \{x\}$) then

$$\begin{aligned}
\mathcal{T}_\eta(M)\{\mathcal{T}_\eta(N)/x\} &= (\lambda z.\mathcal{T}_\eta(M'))\{\mathcal{T}_\eta(N)/x\} \\
&= \lambda z.\mathcal{T}_\eta(M')\{\mathcal{T}_\eta(N)/x\} \\
&\rightsquigarrow \lambda z.\mathcal{T}_\eta(M'\{N/x\}) && \text{by induction hypothesis} \\
&= \mathcal{T}_\eta(M\{N/x\}) \ .
\end{aligned}$$

If $M = M'\, M''$ then

$$\begin{aligned}
&\mathcal{T}_\eta(M)\{\mathcal{T}_\eta(N)/x\} \\
&= (\lambda \vec{y}.\mathcal{T}_\eta(M')\ (\mathcal{T}_\eta(M''))^! :: \vec{y}^{\,!})\{\mathcal{T}_\eta(N)/x\} \\
&= \lambda \vec{y}.(\mathcal{T}_\eta(M')\{\mathcal{T}_\eta(N)/x\})\ (\mathcal{T}_\eta(M'')\{\mathcal{T}_\eta(N)/x\})^! :: \vec{y}^{\,!} && \text{by Lemma 4.2} \\
&\rightsquigarrow \lambda \vec{y}.(\mathcal{T}_\eta(M'\{N/x\}))\ \mathcal{T}_\eta(M''\{N/x\})^! :: \vec{y}^{\,!} && \text{by induction hypothesis} \\
&= \mathcal{T}_\eta(M'\{N/x\}\ M''\{N/x\})\ . \\
&= \mathcal{T}_\eta(M\{N/x\})\ .
\end{aligned}$$

■

**THEOREM 5.14.** *If $M \to_\beta M'$ then $\mathcal{T}_\eta(M) \rightsquigarrow^* \mathcal{T}_\eta(M')$.*

**PROOF.** By compatibility, it is sufficient to consider the case of a redex, $M = (\lambda x.N)\, P$:

$$\begin{aligned}
\mathcal{T}_\eta(M) = \lambda \vec{y}.(\lambda x.\mathcal{T}_\eta(N))\ \mathcal{T}_\eta(P)^! :: \vec{y}^{\,!} &\rightsquigarrow \lambda \vec{y}.\mathcal{T}_\eta(N)[\mathcal{T}_\eta(P)^!/x]\ \vec{y}^{\,!} \\
&= \lambda \vec{y}.\mathcal{T}_\eta(N)\{\mathcal{T}_\eta(P)/x\}\ \vec{y}^{\,!} && \text{by Lemma 4.4} \\
&\rightsquigarrow \lambda \vec{y}.\mathcal{T}_\eta(N\{P/x\})\ \vec{y}^{\,!} && \text{by Lemma 5.13} \\
&\rightsquigarrow \mathcal{T}_\eta(N\{P/x\}) && \text{by Lemma 5.8 .}
\end{aligned}$$

■

Versions of Theorems 5.12 and 5.14 also hold for $\mathcal{T}_h$, although the proof is a bit more contorted, due to the necessary inspection of the head structure of terms: for that reason, we postpone this result to the end of the present section (Theorem 5.20). Instead, we proceed to showing that both flavours of extensional Taylor expansion have the same normal forms. We will rely on the following technical lemma, which will also be used repeatedly later on:

**LEMMA 5.15.** *For every $M \in \Lambda$ and $\vec{N} \in \Lambda^k$ such that $\vec{y} \notin \mathcal{V}_s(M) \cup \mathcal{V}_s(\vec{N})$, $\lambda \vec{y}.\mathcal{T}_h(M)\ \mathcal{T}_h^!(\vec{N})\ \vec{y}^{\,!} \rightsquigarrow^* \mathcal{T}_h(M\,\vec{N})$.*

**PROOF.** If $k = 0$, then we conclude directly by Lemma 5.8. If $M$ is an abstraction and $k > 0$, then we apply the reflexivity of $\rightsquigarrow^*$.

If $M = z\,\vec{P}$ then $\mathcal{T}_h(M) = \lambda \vec{y}.z\ \mathcal{T}_h^!(\vec{P})\ \vec{y}^{\,!}$ hence

$$\begin{aligned}
\lambda \vec{y}.\mathcal{T}_h(M)\ \mathcal{T}_h^!(\vec{N})\ \vec{y}^{\,!} &\rightsquigarrow \lambda \vec{y}.(z\ \mathcal{T}_h^!(\vec{P})\ \vec{y}^{\,!})[\mathcal{T}_h^!(\vec{N})\ \vec{y}^{\,!}/\vec{y}] \\
&= \lambda \vec{y}.(z\ \mathcal{T}_h^!(\vec{P})\ \vec{y}^{\,!})\{\mathcal{T}_h^!(\vec{N})\ \vec{y}^{\,\eta}/\vec{y}\} && \text{by Lemma 4.6} \\
&= \lambda \vec{y}.z\ \mathcal{T}_h^!(\vec{P})\ \vec{y}^{\,!}\{\mathcal{T}_h^!(\vec{N})\ \vec{y}^{\,\eta}/\vec{y}\} \\
&\rightsquigarrow \lambda \vec{y}.z\ \mathcal{T}_h^!(\vec{P})\ \mathcal{T}_h^!(\vec{N})\ \vec{y}^{\,\eta} && \text{by Lemma 5.8} \\
&= \mathcal{T}_h(z\,\vec{P}\,\vec{N})\ .
\end{aligned}$$

If $M = M'\,\vec{P}$ where $M'$ is an abstraction and $|\vec{P}| > 0$, then $\mathcal{T}_h(M) = \lambda \vec{y}.\mathcal{T}_h(\vec{M'})\ \mathcal{T}_h^!(\vec{P})\ \vec{y}^{\,!}$ and we reason as in the previous case. ■

**THEOREM 5.16.** *For every $\lambda$-term $M$, $\mathcal{T}_\eta(M) \rightsquigarrow^* \mathcal{T}_h(M)$.*

**PROOF.** We reason by induction on $M$. If $M = x \in \mathcal{V}$, we have already observed that $\mathcal{T}_h(M) = x^\eta = \mathcal{T}_\eta(M)$. If $M$ is an abstraction, we apply the induction hypothesis. Otherwise, $M = N\,P$ so that:

$$\begin{aligned} \mathcal{T}_\eta(M) = \lambda\vec{y}.\mathcal{T}_\eta(N)\,\mathcal{T}_\eta(P)^! :: \vec{y}^{\,!} &\rightsquigarrow^* \lambda\vec{y}.\mathcal{T}_h(N)\,\mathcal{T}_h(P)^! :: \vec{y}^{\,!} && \text{by induction hypothesis} \\ &\rightsquigarrow^* \mathcal{T}_h(N\,P) && \text{by Lemma 5.15.} \end{aligned}$$

■

**COROLLARY 5.17.** *For any $\lambda$-term $M$, $\mathcal{N}(\mathcal{T}_\eta(M)) = \mathcal{N}(\mathcal{T}_h(M))$. If moreover $M =_{\beta\eta} M'$ then $\mathcal{N}(\mathcal{T}_h(M)) = \mathcal{N}(\mathcal{T}_h(M'))$. In particular if $M$ is normalizable then $\mathcal{N}(\mathcal{T}_h(M)) = \mathcal{T}_h(\mathcal{N}(M))$.*

For each $M \in \Lambda$, we write $\mathcal{N}\mathcal{T}_\eta(M)$ for $\mathcal{N}(\mathcal{T}_\eta(M))$: by Corollary 5.17, we also have $\mathcal{N}\mathcal{T}_\eta(M) = \mathcal{N}(\mathcal{T}_h(M))$. In our introduction, we announced that extensional Taylor expansion would provide a practical alternative to Nakajima trees [34], *i.e.* Böhm trees of $\lambda$-terms quotiented by infinitely nested infinite $\eta$-expansions. We will show in the next section that, indeed, the normalization of Taylor expansion induces the same $\lambda$-theory $\mathbf{H}^*$ as Nakajima trees, but here we first demonstrate extensional Taylor expansion at work, by showing that $\mathcal{N}\mathcal{T}_\eta(\mathbf{J}) = \mathcal{T}_\eta(\mathbf{I})$. It is intuitively obvious that $\mathbf{J}$ converges to the result of applying infinitely nested $\eta$-expansions to $\mathbf{I}$, but the latter notion is not so easy to describe formally nor to reason about; on the other hand, we show that reasoning inductively on extensional resource terms is sufficient to obtain the announced identity.

**EXAMPLE 5.18.** We prove $\mathcal{N}\mathcal{T}_\eta(\mathbf{J}) = \mathcal{T}_\eta(\mathbf{I})$. Recall that $\mathcal{T}_\eta(\mathbf{I}) = \lambda x.x^\eta$ and, by Corollary 5.17, $\mathcal{N}\mathcal{T}_\eta(\mathbf{J}) = \mathcal{N}\mathcal{T}_\eta(\lambda x.\mathbf{J}\,x) = \lambda x.\mathcal{N}\mathcal{T}_\eta(\mathbf{J}\,x)$: it will thus be sufficient to prove that $\mathcal{N}\mathcal{T}_\eta(\mathbf{J}\,x) = x^\eta$ for any variable $x$. Moreover recall from Example 5.9 that $\mathbf{J}\,x =_\beta \lambda y.x\,(\mathbf{J}\,y)$: writing $\mathsf{J}_x := \mathcal{N}\mathcal{T}_\eta(\mathbf{J}\,x)$ and $\mathsf{J}^!_x := (\mathsf{J}_x)^!$, Corollary 5.17 again entails $\mathsf{J}_x = \mathcal{N}\mathcal{T}_\eta(\lambda y.x\,(\mathbf{J}\,y)) = \lambda y.\lambda\vec{z}.x\,\mathsf{J}^!_y :: \vec{z}^{\,!} = \lambda\vec{z}.x\,\mathsf{J}^!_{\vec{z}(0)} :: \vec{z}^{\,!}[\vec{z}\uparrow]$. We establish by induction on resource terms that:

- for any value term $m$, $x^\eta.m = \mathsf{J}_x.m$;
- for any bag term $\bar{m}$, $x^{\,!}.\bar{m} = \mathsf{J}^!_x.\bar{m}$;
- for any stream term $\vec{m}$, $\vec{x}^{\,!}.\vec{m} = (\mathsf{J}^!_{\vec{x}(0)} :: \vec{x}^{\,!}[\vec{x}\uparrow]).\vec{m}$.

Consider a value term $m = \lambda\vec{z}.y\,\vec{n}$. If $y \neq x$, then $\mathsf{J}_x.m = 0 = x^\eta.m$. Otherwise, by induction hypothesis on $\vec{n}$, $\mathsf{J}_x.m = (\mathsf{J}^!_{\vec{z}(0)} :: \vec{z}^{\,!}[\vec{z}\uparrow]).\vec{n} = \vec{z}^{\,!}.\vec{n} = x^\eta.m$. Now consider a bag term $\bar{m} = [m_1, \ldots, m_k]$: applying the induction to each $m_i$, we obtain $\mathsf{J}^!_x.\bar{m} = \frac{1}{k!}\prod_{i=1}^k \mathsf{J}_x.m_i = \frac{1}{k!}\prod_{i=1}^k x^\eta.m_i = x^{\,!}.\bar{m}$. Finally consider a stream term $\vec{m}$. If $\vec{m} = \iota$ then $(\mathsf{J}^!_{\vec{x}(0)} :: \vec{x}^{\,!}[\vec{x}\uparrow]).\vec{m} = \mathsf{J}^!_{\vec{x}(0)}.[\,] \times \vec{x}^{\,!}[\vec{x}\uparrow].\iota = 1 = \vec{x}^{\,!}.\vec{m}$. Otherwise, we can write $\vec{m} = \bar{m} :: \vec{n}$ and apply the induction hypothesis to $\bar{m}$, to obtain: $(\mathsf{J}^!_{\vec{x}(0)} :: \vec{x}^{\,!}[\vec{x}\uparrow]).\vec{m} = \mathsf{J}^!_{\vec{x}(0)}.\bar{m} \times \vec{x}^{\,!}[\vec{x}\uparrow].\vec{n} = \vec{x}(0)^{\,!}.\bar{m} \times \vec{x}^{\,!}[\vec{x}\uparrow].\vec{n} = \vec{x}^{\,!}.\vec{m}$ by Lemma 5.1. ◆

We conclude this section by showing that head Taylor expansion, like structural Taylor expansion, sends $\beta\eta$-reduction steps to resource reductions on value vectors. This result can be taken as an indication of the robustness of the notion, but is not actually needed: Corollary 5.17 is sufficient to ensure that $\mathcal{N}\mathcal{T}_\eta(M)$ depends only on the $\beta\eta$-class of $M$; and the important property of $\mathcal{T}_h$ is rather its commutation with head reduction, that we establish and exploit in the next section. The reader may thus safely skip Theorem 5.20 below. On the other hand, the following variant of Lemma 5.13 for $\mathcal{T}_h$ will be useful:

**LEMMA 5.19.** *For all $\lambda$-terms $M$ and $N$, $\mathcal{T}_h(M)\{\mathcal{T}_h(N)/x\} \rightsquigarrow^* \mathcal{T}_h(M\{N/x\})$.*

**PROOF.** The proof is by induction on $M$. The case of $\lambda x.N$ is settled by applying the induction hypothesis to $N$ as in the proof of Lemma 5.13.

If $M = M'\,\vec{P}$ where $M'$ is an abstraction and $|\vec{P}| > 0$ then

$$\mathcal{T}_h(M)\{\mathcal{T}_h(N)/x\} = \lambda\vec{y}.\big(\mathcal{T}_h(M')\{\mathcal{T}_h(N)/x\}\big)\,\big(\mathcal{T}_h^!(\vec{P})\{\mathcal{T}_h(N)/x\}\,\vec{y}^{\,!}\big)$$

where, by Lemma 4.5,

$$\mathcal{T}_h^!(\vec{P})\{\mathcal{T}_h(N)/x\}\,\vec{y}^{\,!} = (\mathcal{T}_h(P_0)\{\mathcal{T}_h(N)/x\})^! :: \cdots :: (\mathcal{T}_h(P_k)\{\mathcal{T}_h(N)/x\})^! :: \vec{y}^{\,!}$$

if $\vec{P} = \langle P_0, \dots, P_k\rangle$. Applying the induction hypothesis to $M'$ and to each $P_i$, we obtain

$$\mathcal{T}_h(M)\{\mathcal{T}_h(N)/x\} \rightsquigarrow^* \lambda\vec{y}.\big(\mathcal{T}_h(M'\{N/x\})\big)\,\big(\mathcal{T}_h^!(\vec{P}\{N/x\})\,\vec{y}^{\,!}\big)$$

and conclude, observing that $M'\{N/x\}$ is an abstraction and $|\vec{P}\{N/x\}| > 0$.

If $M = z\,\vec{P}$ with $z \neq x$ then

$$\mathcal{T}_h(M)\{\mathcal{T}_h(N)/x\} = \lambda\vec{y}.z\,\big(\mathcal{T}_h^!(\vec{P})\{\mathcal{T}_h(N)/x\}\,\vec{y}^{\,!}\big)$$

and we obtain $\mathcal{T}_h(M)\{\mathcal{T}_h(N)/x\} \rightsquigarrow^* \lambda\vec{y}.z\,\big(\mathcal{T}_h^!(\vec{P}\{N/x\})\,\vec{y}^{\,!}\big)$ as in the previous case.

If $M = x\,\vec{P}$ then

$$\mathcal{T}_h(M)\{\mathcal{T}_h(N)/x\} = \lambda\vec{y}.\mathcal{T}_h(N)\,\big(\mathcal{T}_h^!(\vec{P})\{\mathcal{T}_h(N)/x\}\,\vec{y}^{\,!}\big)$$

and again $\mathcal{T}_h(M)\{\mathcal{T}_h(N)/x\} \rightsquigarrow^* \lambda\vec{y}.\mathcal{T}_h(N)\,\big(\mathcal{T}_h^!(\vec{P}\{N/x\})\,\vec{y}^{\,!}\big)$. We conclude by Lemma 5.15. ■

**THEOREM 5.20.** *If $M \to_{\beta\eta} M'$ then $\mathcal{T}_h(M) \rightsquigarrow \mathcal{T}_h(M')$.*

**PROOF.** The proof is by induction on $M$.

We first treat the case of an abstraction $M = \lambda x.N$, necessarily with $x \notin \mathcal{V}(M')$. If moreover $M' = \lambda x.N'$ with $N \to_{\beta\eta} N'$, then we can apply the induction hypothesis directly, using the compatibility of $\rightsquigarrow$. Otherwise, we must have $N = M'\,x$ so that $M \to_\eta M'$, and $M'$ is not an abstraction (if it is, we can write $M' = \lambda x.N'$ and have $N \to_\beta N'$). In that case, we write $M' = M_0\,\vec{P}$ where $M_0$ is a variable or an abstraction (with $|\vec{P}| > 0$ in the latter case). Writing $E = M_0$ if $M_0$ is a variable, and $E = \mathcal{T}_h(M_0)$ otherwise, we obtain $\mathcal{T}_h(M) = \lambda x.\lambda\vec{y}.E\,\mathcal{T}_h^!(\vec{P})\,x^! :: \vec{y}^{\,!} = \lambda\vec{y}.E\,\mathcal{T}_h^!(\vec{P}) :: \vec{y}^{\,!} = \mathcal{T}_h(M')$.

If $M = x\,\vec{P}$, then the reduction $M \to_{\beta\eta} M'$ occurs necessarily in an element of $\vec{P}$ and we apply the induction hypothesis to that element and the compatibility of $\rightsquigarrow$.

We finally treat the case of $M = (\lambda x.M_0)\,(N :: \vec{P})$. If the reduction occurs in $M_0$, in $N$, or in an element of $\vec{P}$, then again we apply the induction hypothesis and the compatibility of $\rightsquigarrow$, the head structure of $M'$ being the same. If $M' = (M_0\{N/x\})\,\vec{P}$ so that $M \to_\beta M'$, then

$$\begin{aligned}
\mathcal{T}_h(M) &= \lambda\vec{y}.(\lambda x.\mathcal{T}_h(M_0))\,\mathcal{T}_h^!(N) :: \mathcal{T}_h^!(\vec{P})\,\vec{y}^{\,!} \\
&\rightsquigarrow \lambda\vec{y}.(\mathcal{T}_h(M_0)\{\mathcal{T}_h(N)/x\})\,\mathcal{T}_h^!(\vec{P})\,\vec{y}^{\,!} && \text{by Lemma 4.4} \\
&\rightsquigarrow \lambda\vec{y}.\mathcal{T}_h(M_0\{N/x\})\,\mathcal{T}_h^!(\vec{P})\,\vec{y}^{\,!} && \text{by Lemma 5.19} \\
&\rightsquigarrow \mathcal{T}_h(M') && \text{by Lemma 5.15.}
\end{aligned}$$

Finally, if $M_0 = M_0'\,x$ with $x \notin \mathcal{V}(M_0')$ and $M' = M_0'\,(N :: \vec{P})$ so that $M \to_\eta M'$, then observe that we also have $M' = (M_0\{N/x\})\,\vec{P}$, and we are back to the previous case. ■

# 6. Characterization of $\mathbf{H}^*$

In this section, we study the equational theory induced by the normalization of extensional Taylor expansion. Writing $M =_{\mathcal{T}_\eta} M'$ if $\mathcal{N}\mathcal{T}_\eta(M) = \mathcal{N}\mathcal{T}_\eta(M')$, we will show that $=_{\mathcal{T}_\eta}$ is the maximum consistent and sensible $\lambda$-theory $\mathbf{H}^*$ (Theorem 6.13). The latter is known to be characterized by Nakajima trees [34, 2, Exercise 19.4.4], but our results demonstrate how extensional Taylor expansion allows us to dispense with such intrinsically infinite objects, in the same fashion as ordinary Taylor expansion provides an alternative to Böhm trees, based on finite syntactic approximants.[19]

**LEMMA 6.1.** *The equivalence relation $=_{\mathcal{T}_\eta}$ on $\Lambda$ is an extensional $\lambda$-theory.*

**PROOF.** That $=_{\mathcal{T}_\eta}$ is a congruence follows directly from the compatibility of $\rightsquigarrow$. Moreover, by Corollary 5.17, $=_{\mathcal{T}_\eta}$ contains $\to_\beta$ and $\to_\eta$. ■

## 6.1 Sensibility

We now show that, like ordinary Taylor expansion, extensional Taylor expansion allows us to characterize the head normalizability of $\lambda$-terms.

**THEOREM 6.2.** *A $\lambda$-term is head normalizable iff $\mathcal{N}\mathcal{T}_\eta(M) \neq 0$.*

As in the proof of Lemma 1.4, the forward implication is easy:

---

[19] A game semanticist reader could rightfully argue that classical game semantics already provide an alternative to Nakajima trees, based on finite approximants. A key difference is that (ordinary as well as extensional) resource terms are not necessarily normal: they provide a language of finite, syntactic approximants of $\lambda$-terms *and* of their (potentially infinite) normal forms.

**LEMMA 6.3.** *If $M$ is $\beta\eta$-equivalent to a head normal form then $\mathcal{N}\mathcal{T}_\eta(M) \neq 0$.*

**PROOF.** If $M =_{\beta\eta} M' = \lambda\vec{x}.z\,\vec{P}$, then $m := \lambda\vec{x}.\lambda\vec{y}.z\,\iota \in \mathcal{T}_h(M')$, hence $m \in \mathcal{N}\mathcal{T}_\eta(M)$. ∎

For the other implication, as for Lemma 1.4, we apply the head reduction strategy for resource reduction until we reach a head normal form of some term in the image of Taylor expansion; and we show that this reduction path can be reflected as a head normalization sequence for the source $\lambda$-term. To ensure that head reduction terminates in the resource calculus, we have to consider full-step reduction: this will make the proof significantly more complex than that of Lemma 1.4, because such a full-step is not necessarily the translation of a single head $\beta$-reduction step.

Moreover, it will be more practical to rely on $\mathcal{T}_h$ instead of $\mathcal{T}_\eta$, because the latter does not preserve the head structure of terms: when $M$ is in head normal form, the elements of $\mathcal{T}_\eta(M)$ need not be (consider $M = y\,N$); and if $M$ has a head redex, the head redex in an element of $\mathcal{T}_\eta(M)$ is not necessarily the image of that of $M$ (consider $M = (\lambda x.N)\,P_1\,P_2$). This will require additional work to establish the properties of $\mathcal{T}_h$ w.r.t. reduction, instead of relying on those of $\mathcal{T}_\eta$ via Corollary 5.17, as was sufficient before.

Given a redex $a \in \Delta_{\mathrm{b}}$, we write $\mathcal{H}_{\mathrm{R}}(a)$ for the base sum obtained by fully reducing it: namely, if $a = (\lambda\vec{y}.b)\,\vec{n}$ then $\mathcal{H}_{\mathrm{R}}(a) := b[\vec{n}/\vec{y}]$. We say that a value term $m = \lambda\vec{x}.a$ is **head reducible** if $a$ is a redex, and then set $\mathcal{H}_{\mathrm{R}}(m) := \lambda\vec{x}.\mathcal{H}_{\mathrm{R}}(a)$ so that $m \mapsto_{\mathrm{R}} \mathcal{H}_{\mathrm{R}}(m)$; otherwise we say that $m$ is in **head normal form**. **Head reduction** is defined on sums and vectors by linearity: $\mathcal{H}_{\mathrm{R}}(M) := \sum_{m\in M}(M.m)\mathcal{H}_{\mathrm{R}}(m)$ for any $M \in \mathbb{K}\langle\Delta_{\mathrm{v}}\rangle$ whose support contains head reducible terms only, so that $M \rightsquigarrow \mathcal{H}_{\mathrm{R}}(M)$ in this case. By Lemma 3.20, if $m$ is head reducible and $m' \in \mathcal{H}_{\mathrm{R}}(m)$ then $\#m' < \#m$: the head reduction sequence of a value sum always terminates.

With these definitions in place, a $\lambda$-term $M$ is head reducible iff one value term in $\mathcal{T}_h(M)$ is (and then every term in $\mathcal{T}_h(M)$ is head reducible). However, as we have already stated, head reduction does not commute with head Taylor expansion on the nose: in general $\mathcal{H}_{\mathrm{R}}(\mathcal{T}_h(M)) \neq \mathcal{T}_h(\mathcal{H}(M))$ (recall that $\mathcal{H}(M)$ denotes the head reduct of $M$). Consider, for instance, $M = (\lambda x_1.\lambda x_2.x_1)\,y_1\,y_2$: it is easy to check that $\mathcal{H}_{\mathrm{R}}(\mathcal{T}_h(M)) = \lambda\vec{z}.y_1^\eta\,\vec{z}^{\,!}$, while $y_2$ still occurs in $\mathcal{H}(M)$, hence in $\mathcal{T}_h(\mathcal{H}(M))$; on the other hand, $\mathcal{H}_{\mathrm{R}}(\mathcal{T}_h(M)) \rightsquigarrow y_1^\eta$ and $\mathcal{H}(\mathcal{H}(M)) = y_1$. And we will indeed show that a head reduction step from the expansion of a head redex can always be extended to the image of a head reduction sequence (Lemma 6.5).

**LEMMA 6.4.** *For any base vector of the form $A = (\lambda x_1.\cdots\lambda x_k.\lambda\vec{y}.B)\,\bar{N}_1 :: \cdots :: \bar{N}_k :: \vec{P} \in \mathbb{K}\langle\Delta_{\mathrm{b}}\rangle$ we have $\mathcal{H}_{\mathrm{R}}(A) = B[\bar{N}_1/x_1]\cdots[\bar{N}_k/x_k][\vec{P}/\vec{y}]$.*

**PROOF.** By linearity, it is sufficient to consider the case of a base term:

$$A = a = (\lambda x_1.\cdots\lambda x_k.\lambda\vec{y}.b)\,\bar{n}_1 :: \cdots :: \bar{n}_k :: \vec{p} \in \Delta_{\mathrm{b}}\ .$$

By definition,

$$a = \big(\lambda \vec{y}.b[\vec{y}\uparrow]\{\vec{y}(0)/x_k\} \cdots [\vec{y}\uparrow]\{\vec{y}(0)/x_1\}\big)\, \bar{n}_1 :: \cdots :: \bar{n}_k :: \vec{p}$$

hence

$$\mathcal{H}_{\mathrm{R}}(a) = b[\vec{y}\uparrow]\{\vec{y}(0)/x_k\} \cdots [\vec{y}\uparrow]\{\vec{y}(0)/x_1\}[\bar{n}_1 :: \cdots :: \bar{n}_k :: \vec{p}/\vec{y}]$$

and we conclude by iterating Lemmas 3.10 and 3.19. ■

**LEMMA 6.5.** *For every head reducible $M \in \Lambda$, there exists $k \in \mathbb{N}$ such that $\mathcal{H}^k(M)$ is defined and $\mathcal{H}_{\mathrm{R}}(\mathcal{T}_h(M)) \rightsquigarrow^* \mathcal{T}_h(\mathcal{H}^k(M))$.*

**PROOF.** We reason by induction on $M$. If $M = \lambda y.N$ then we apply the induction hypothesis to $N$, and conclude by compatibility. Otherwise, $M = P\,\vec{N}$ where $P$ is an abstraction and $\vec{N} = \langle N_0, \dots, N_l \rangle$ is a non empty sequence of $\lambda$-terms. We write $P = \lambda \vec{x}.P'$ where $\vec{x} = \langle x_0, \dots x_k \rangle$ is a non empty tuple of pairwise distinct variables and $P'$ is not an abstraction: $P' = P''\,\vec{N}'$ and, either $P'' = z$ is a variable, or $P''$ is an abstraction and $|\vec{N}'| > 0$. Then we can write $\mathcal{T}_h(P') = \lambda \vec{z}.E\,\mathcal{T}_h^{!}(\vec{N}')\vec{z}^{\,!}$ where $E = z$ or $E = \mathcal{T}_h(P'')$, and $\vec{z}$ is fresh. Choosing $\vec{y}$ fresh, we obtain:

$$\mathcal{T}_h(M) = \lambda \vec{y}.\mathcal{T}_h(P)\,\mathcal{T}_h^{!}(\vec{N})\,\vec{y}^{\,!} = \lambda \vec{y}.(\lambda \vec{x}.\lambda \vec{z}.E\,\mathcal{T}_h^{!}(\vec{N}')\vec{z}^{\,!})\,\mathcal{T}_h^{!}(\vec{N})\,\vec{y}^{\,!}\ .$$

If $k \leq l$ then we write $\vec{N}'' := \langle N_0, \dots, N_k \rangle$ and $\vec{N}''' := \langle N_{k+1}, \dots, N_l \rangle$, and we obtain $\mathcal{H}^k(M) = (P'\{\vec{N}''/\vec{x}\})\,\vec{N}'''$. In this case, we obtain

$$\begin{aligned}
&\mathcal{H}_{\mathrm{R}}(\mathcal{T}_h(M)) && \\
&= \lambda \vec{y}.(E\,\mathcal{T}_h^{!}(\vec{N}')\,\vec{z}^{\,!})[\mathcal{T}_h^{!}(\vec{N}'')/\vec{x}][\mathcal{T}_h^{!}(\vec{N}''')\,\vec{y}^{\,!}/\vec{z}] && \text{by Lemma 6.4} \\
&= \lambda \vec{y}.(E\,\mathcal{T}_h^{!}(\vec{N}')\,\vec{z}^{\,!})\{\mathcal{T}_h(\vec{N}'')/\vec{x}\}[\mathcal{T}_h^{!}(\vec{N}''')\,\vec{y}^{\,!}/\vec{z}] && \text{by Lemma 4.4} \\
&= \lambda \vec{y}.E\{\mathcal{T}_h(\vec{N}'')/\vec{x}\}\,\mathcal{T}_h^{!}(\vec{N}')\{\mathcal{T}_h(\vec{N}'')/\vec{x}\}\,\vec{z}^{\,!}[\mathcal{T}_h^{!}(\vec{N}''')\,\vec{y}^{\,!}/\vec{z}] && \text{since } \vec{z} \text{ is fresh} \\
&\rightsquigarrow \lambda \vec{y}.E\{\mathcal{T}_h(\vec{N}'')/\vec{x}\}\,\mathcal{T}_h^{!}(\vec{N}')\{\mathcal{T}_h(\vec{N}'')/\vec{x}\}\,\mathcal{T}_h^{!}(\vec{N}''')\,\vec{y}^{\,!} && \text{by Lemma 5.8} \\
&\rightsquigarrow^* \lambda \vec{y}.E\{\mathcal{T}_h(\vec{N}'')/\vec{x}\}\,\mathcal{T}_h^{!}(\vec{N}'\{\vec{N}''/\vec{x}\})\,\mathcal{T}_h^{!}(\vec{N}''')\,\vec{y}^{\,!} && \text{by Lemmas 4.2 and 5.19} \\
&= \lambda \vec{y}.E\{\mathcal{T}_h(\vec{N}'')/\vec{x}\}\,\mathcal{T}_h^{!}(\vec{N}'\{\vec{N}''/\vec{x}\}\,\vec{N}''')\,\vec{y}^{\,!}\ . &&
\end{aligned}$$

If $P' = z\,\vec{N}'$ with $z \notin \vec{x}$ then $E\{\mathcal{T}_h(\vec{N}'')/\vec{x}\} = z$ and we conclude since we have $\mathcal{H}^k(M) = z\,(\vec{N}'\{\vec{N}''/\vec{x}\})\,\vec{N}'''$. If $P' = x_i\,\vec{N}'$ then $E\{\mathcal{T}_h(\vec{N}'')/\vec{x}\} = \mathcal{T}_h(N_i)$ and we conclude by Lemma 5.15 since $\mathcal{H}^k(M) = N_i\,(\vec{N}'\{\vec{N}''/\vec{x}\})\,\vec{N}'''$. And if $P' = P''\,\vec{N}'$ with $P''$ an abstraction and $|\vec{N}'| > 0$, then $E = \mathcal{T}_h(P'')$ hence $E\{\mathcal{T}_h(\vec{N}'')/\vec{x}\} \rightsquigarrow^* \mathcal{T}_h(P''\{\vec{N}''/\vec{x}\})$ by Lemma 5.19, hence

$$\mathcal{H}_{\mathrm{R}}(\mathcal{T}_h(M)) \rightsquigarrow^* \lambda \vec{y}.\mathcal{T}_h(P''\{\vec{N}''/\vec{x}\})\,\mathcal{T}_h^{!}(\vec{N}'\{\vec{N}''/\vec{x}\}\,\vec{N}''')\,\vec{y}^{\,!}$$

and we conclude again by Lemma 5.15 since $\mathcal{H}^k(M) = \big(\vec{P}''\{\vec{N}''/\vec{x}\}\big)\,\big(\vec{N}'\{\vec{N}''/\vec{x}\}\big)\,\vec{N}'''$.

Now if $k > l$, then we write $\vec{x}' := \langle x_0, \dots, x_l \rangle$ and $\vec{x}'' := \langle x_{l+1}, \dots, x_k \rangle$ so that $\mathcal{H}^{l}(M) = \lambda \vec{x}''.P'\{\vec{N}/\vec{x}'\}$. We also write $k' := k - l - 1$ and $\langle x'_0, \dots, x'_{k'} \rangle := \vec{x}''$. Then we compute:

$$
\begin{aligned}
&\mathcal{H}_{\mathrm{R}}(\mathcal{T}_h(M)) \\
&= \lambda \vec{y}.(E\, \mathcal{T}^{!}_{h}(\vec{N}')\, \vec{z}^{\,!})[\mathcal{T}^{!}_{h}(\vec{N})/\vec{x}'][\vec{y}(0)^{!}/x'_0] \cdots [\vec{y}(k')^{!}/x'_{k'}][\vec{y}^{\,!}[\vec{y}\uparrow]^{k'+1}/\vec{z}] && \text{by Lemmas 5.1 and 6.4} \\
&= \lambda \vec{y}.(E\, \mathcal{T}^{!}_{h}(\vec{N}')\, \vec{z}^{\,!})[\mathcal{T}^{!}_{h}(\vec{N})/\vec{x}'][\vec{y}^{\,!}[\vec{y}\uparrow]^{k'+1}/\vec{z}][\vec{y}(k')^{!}/x'_{k'}] \cdots [\vec{y}(0)^{!}/x'_0] && \text{since } \vec{y} \text{ and } \vec{z} \text{ were fresh} \\
&= \lambda \vec{y}.(E\, \mathcal{T}^{!}_{h}(\vec{N}')\, \vec{z}^{\,!})[\mathcal{T}^{!}_{h}(\vec{N})/\vec{x}'][\vec{y}^{\,!}/\vec{z}][\vec{y}\uparrow]^{k'+1}[\vec{y}(k')^{!}/x'_{k'}] \cdots [\vec{y}(0)^{!}/x'_0] && \text{by Lemma 3.10} \\
&= \lambda \vec{y}.(E\, \mathcal{T}^{!}_{h}(\vec{N}')\, \vec{z}^{\,!}[\vec{y}^{\,!}/\vec{z}])[\mathcal{T}^{!}_{h}(\vec{N})/\vec{x}'][\vec{y}\uparrow]^{k'+1}[\vec{y}(k')^{!}/x'_{k'}] \cdots [\vec{y}(0)^{!}/x'_0] && \text{since } \vec{y} \text{ and } \vec{z} \text{ were fresh} \\
&\leadsto \lambda \vec{y}.(E\, \mathcal{T}^{!}_{h}(\vec{N}')\, \vec{y}^{\,!})[\mathcal{T}^{!}_{h}(\vec{N})/\vec{x}'][\vec{y}\uparrow]^{k'+1}[\vec{y}(k')^{!}/x'_{k'}] \cdots [\vec{y}(0)^{!}/x'_0] && \text{by Lemma 5.8} \\
&= \lambda \vec{y}.(E\, \mathcal{T}^{!}_{h}(\vec{N}')\, \vec{y}^{\,!})\{\mathcal{T}_h(\vec{N})/\vec{x}'\}[\vec{y}\uparrow]^{k'+1}\{\vec{y}(k')^{\eta}/x'_{k'}\} \cdots \{\vec{y}(0)^{\eta}/x'_0\} && \text{by Lemma 4.4} \\
&\leadsto^{*} \lambda \vec{y}.(E\, \mathcal{T}^{!}_{h}(\vec{N}')\, \vec{y}^{\,!})\{\mathcal{T}_h(\vec{N})/\vec{x}'\}[\vec{y}\uparrow]^{k'+1}\{\vec{y}(k')/x'_{k'}\} \cdots \{\vec{y}(0)/x'_0\} && \text{by Lemma 5.8} \\
&= \lambda \vec{y}.(E\, \mathcal{T}^{!}_{h}(\vec{N}')\, \vec{y}^{\,!})\{\mathcal{T}_h(\vec{N})/\vec{x}'\}[\vec{y}\uparrow]^{k'}\{\vec{y}(k'-1)/x'_{k'}\} \cdots \{\vec{y}(0)/x'_1\}[\vec{y}\uparrow]\{\vec{y}(0)/x'_0\} \\
& && \text{by Lemma 3.10} \\
&= \lambda x'_0.\lambda \vec{y}.(E\, \mathcal{T}^{!}_{h}(\vec{N}')\, \vec{y}^{\,!})\{\mathcal{T}_h(\vec{N})/\vec{x}'\}[\vec{y}\uparrow]^{k'}\{\vec{y}(k'-1)/x'_{k'}\} \cdots \{\vec{y}(0)/x'_1\} && \text{by definition} \\
&\leadsto^{*} \lambda \vec{x}''.\lambda \vec{y}.(E\, \mathcal{T}^{!}_{h}(\vec{N}')\, \vec{y}^{\,!})\{\mathcal{T}_h(\vec{N})/\vec{x}'\} && \text{iterating previous steps}
\end{aligned}
$$

It will thus be sufficient to establish $\lambda \vec{x}''.\lambda \vec{y}.(E\, \mathcal{T}^{!}_{h}(\vec{N}')\, \vec{y}^{\,!})\{\mathcal{T}_h(\vec{N})/\vec{x}'\} \leadsto^{*} \mathcal{T}_h(P'\{\vec{N}/\vec{x}'\})$ which is done like in the case $k \leq l$, by applying Lemmas 4.2 and 5.19, and inspecting the shape of $P'$. ■

We are now ready to complete the proof of Theorem 6.2:

**LEMMA 6.6.** *If $\mathcal{N}\mathcal{T}_\eta(M) \neq 0$ then $M$ is head normalizable.*

**PROOF.** If $\mathcal{N}\mathcal{T}_\eta(M) \neq 0$ then there exists $m \in \mathcal{T}_h(M)$ such that $\mathcal{N}(m) \neq 0$. The proof is by induction on $\#m$. If $M$ is in head normal form then we conclude directly. Otherwise, $\mathcal{N}(\mathcal{H}_{\mathrm{R}}(m)) \neq 0$, and we can pick $m' \in \mathcal{H}_{\mathrm{R}}(m)$ such that $\mathcal{N}(m') \neq 0$. In particular, $\#m' < \#m$. By Lemma 6.5, we obtain $m' \to_{\mathrm{r}}^{*} M''$ with $\mathrm{supp}(M'') \subseteq \mathrm{supp}(\mathcal{T}_\eta(\mathcal{H}^{k}(M)))$. Again, $\mathcal{N}(M'') = \mathcal{N}(m') \neq 0$ and we can pick $m'' \in M''$ such that $\mathcal{N}(m'') \neq 0$. We also have $m'' \in \mathcal{T}_\eta(\mathcal{H}^{k}(M))$ and, moreover, $\#m'' \leq \#m' < \#m$ so the induction hypothesis applies: $\mathcal{H}^{k}(M)$ is head normalizable, hence $M$ is head normalizable too. ■

**COROLLARY 6.7.** *The $\lambda$-theory $=_{\mathcal{T}_\eta}$ is sensible.*

**PROOF.** By Lemma 6.6, $\mathcal{N}\mathcal{T}_\eta(M) = 0$ as soon as $M$ is not head normalizable. ■

We conclude the present subsection by showing that the characterization of coefficients in the extensional Taylor expansion of a term (Lemma 5.10) remains valid after normalization. Our proof relies on Lemma 6.6.

**LEMMA 6.8.** *If $m \in \mathcal{N}\mathcal{T}_\eta(M)$ then $\mathcal{N}\mathcal{T}_\eta(M).m = \frac{1}{\mathrm{m}(m)}$.*

**PROOF.** The proof is by induction on $\#m$. Since $m \in \mathcal{N}\mathcal{T}_\eta(M)$, Lemma 6.6 ensures that $M$ is head normalizable: $M =_\beta M'$ with $M' = \lambda x_1. \cdots \lambda x_k.y\,\vec{N}$, and $\vec{N} = \langle N_0, \ldots, N_{l-1}\rangle$. By Corollary 5.17, $\mathcal{N}\mathcal{T}_\eta(M) = \mathcal{N}\mathcal{T}_\eta(M')$. Moreover, $\mathcal{T}_h(M') = \lambda x_1. \cdots \lambda x_k.\lambda\vec{z}.y\,\mathcal{T}_h^!(\vec{N})\,\vec{z}^{\,!}$, where $\vec{z}$ is chosen fresh. Then we can write $m = \lambda x_1. \cdots \lambda x_k.\lambda\vec{z}.y\,\bar{n}_0 :: \cdots :: \bar{n}_{l+l'-1} :: \iota$, where $\bar{n}_i \in \mathcal{N}\mathcal{T}_\eta(N_i)^!$ for $0 \leq i < l$ and $\bar{n}_{l+j} \in \vec{z}(j)^!$ for $0 \leq j < l'$. For $0 \leq i < l$, applying the induction hypothesis to the elements of $\bar{n}_i$, and then Lemma 4.3, we obtain $\mathcal{N}\mathcal{T}_\eta(N_i)^!.\bar{n}_i = \frac{1}{\mathsf{m}(\bar{n}_i)}$; and for $0 \leq j < l'$, we have $\vec{z}(j)^!.\bar{n}_{l+j} = \frac{1}{\mathsf{m}(\bar{n}_{l+j})}$ by Lemma 5.2. The result follows by the definition of $\mathsf{m}(m)$. ■

This ensures that the definition of $=_{\mathcal{T}_\eta}$ does not depend on the choice of the semiring of coefficients.

**COROLLARY 6.9.** *We have $M =_{\mathcal{T}_\eta} M'$ iff $\operatorname{supp}(\mathcal{N}\mathcal{T}_\eta(M)) = \operatorname{supp}(\mathcal{N}\mathcal{T}_\eta(M'))$.*

## 6.2 Böhm-out via Taylor expansion

We have established that $=_{\mathcal{T}_\eta}$ is a sensible extensional $\lambda$-theory. It is obviously consistent since, e.g., $x \neq_{\mathcal{T}_\eta} y$ when $x \neq y$.

To establish that $=_{\mathcal{T}_\eta}$ is indeed maximum among sensible consistent $\lambda$-theories, we show that it contains the observational equivalence induced by head normal forms: writing $\Lambda_{\mathrm{hn}}$ for the set of head normalizable $\lambda$-terms, $M$ and $N$ are **observationally equivalent**, and we write $M =_{\mathrm{hn}} N$ if, for each $\lambda$-term context $C[\,]$, $C[M] \in \Lambda_{\mathrm{hn}}$ iff $C[N] \in \Lambda_{\mathrm{hn}}$. It is easy to check that a sensible $\lambda$-theory identifying two $=_{\mathrm{hn}}$-distinct terms is inconsistent [2, Lemma 16.2.4]. It follows that all sensible consistent $\lambda$-theories are included in $=_{\mathrm{hn}}$, so it remains only to prove that $=_{\mathcal{T}_\eta}$ contains $=_{\mathrm{hn}}$.

Our objective is thus to show that if $M \neq_{\mathcal{T}_\eta} N$ then there is a context $C[\,]$ such that one of $C[M]$ and $C[N]$ is head normalizable, and the other one is not — in this case we say that $C[\,]$ **separates** $M$ from $N$. By Corollary 6.9, the assumption $M \neq_{\mathcal{T}_\eta} N$ amounts to the existence of a normal value term $m$ such that $m \in \mathcal{N}\mathcal{T}_\eta(M) \setminus \mathcal{N}\mathcal{T}_\eta(N)$ — or *vice versa*. We show that the standard Böhm-out technique to separate $\beta\eta$-distinct normal $\lambda$-terms can be adapted to this setting, by reasoning on normal value terms instead: most of what follows is standard material about the $\lambda$-calculus, and only the final result relies on the properties of extensional Taylor expansion.

Following the Böhm-out technique, we will only use separating contexts corresponding to **Böhm transformations**, which are generated by composing the following basic transformations: $\tau_N : M \mapsto M\,N$ for $N \in \Lambda$ (corresponding to the context $[\,]\,N$); and $\sigma^x_N : M \mapsto M\{N/x\}$ for $N \in \Lambda$ and $x \in \mathcal{V}$ (corresponding to the context $(\lambda x.[\,])\,N$). We use a postfix notation for the application of Böhm transformations and use the sequential order for their composition so that $M\tau\sigma = (M\tau)\sigma$ for any Böhm transformations $\tau$ and $\sigma$. We may apply a Böhm transformation $\sigma$ to a tuple of terms: $\langle M_1, \ldots, M_k\rangle\sigma := \langle M_1\sigma, \ldots, M_k\sigma\rangle$.

We say that $M$ and $N$ are **strongly separable**, and write $M \bowtie N$, if there exists a Böhm transformation $\tau$ such that $M\tau =_\beta \lambda x.\lambda y.x$ and $N\tau =_\beta \lambda x.\lambda y.y$. And we say that $M$ is **separable** from $N$, and write $M \ltimes N$, if there exists a Böhm transformation $\tau$ such that $M\tau \in \Lambda_{\mathrm{hn}}$ and $N\tau \notin \Lambda_{\mathrm{hn}}$. Note that strong separability implies separability and is symmetric.[20]

The following are direct consequences of the definitions or basic exercises in $\lambda$-calculus.

**FACT 6.10.** *We have $M \bowtie N$ as soon as one of the following holds:*

- $M = x\,\vec{M}$ *and* $N = y\,\vec{N}$ *with* $x \neq y$ *or* $|\vec{M}| \neq |\vec{N}|$;
- $M = x\,\vec{M}$ *and* $N = \lambda\vec{y}.y\,\vec{N}$ *with* $y \in \vec{y}$;
- $M = \lambda\vec{y}.\lambda x.x\,\vec{M}$ *and* $N = \lambda\vec{z}.\lambda x.x\,\vec{N}$ *with* $|\vec{y}| \neq |\vec{z}|$ *or* $|\vec{M}| \neq |\vec{N}|$;

*and we have $M \ltimes N$ as soon as one of the following holds:*

- $M \in \Lambda_{\mathrm{hn}}$ *and* $N \notin \Lambda_{\mathrm{hn}}$;
- $M\,P \ltimes N\,P$ *for some* $P \in \Lambda$;
- $M\{P/x\} \ltimes N\{P/x\}$ *for some* $x \in \mathcal{V}$ *and* $P \in \Lambda$;
- $M =_{\beta\eta} M' \ltimes N' =_{\beta\eta} N$;
- $M = \lambda x.M'$ *and* $N = \lambda x.N'$ *with* $M' \ltimes N'$.

For each $k \in \mathbb{N}$, we write $\rho_k := \lambda x_1.\ldots\lambda x_k.\lambda y.y\,x_1\cdots x_k \in \Lambda$. For $l \in \mathbb{N}$, $\vec{x} = \langle x_1,\ldots,x_l\rangle \in \mathcal{V}^l$ and $\vec{k} = \langle k_1,\ldots,k_l\rangle \in \mathbb{N}^l$, we define the Böhm transformation $\sigma^{\vec{x}}_{\vec{k}} := \sigma^{x_1}_{\rho_{k_1}}\cdots\sigma^{x_l}_{\rho_{k_l}}$, which is a sequence of substitutions. We say that $M$ is **Böhm-separable** from $N$, and write $M \ltimes_{\mathcal{B}} N$, if, for each $l \in \mathbb{N}$, each tuple $\vec{x} \in \mathcal{V}^l$ of pairwise distinct variables and each tuple $\vec{k} \in \mathbb{N}^l$ of pairwise distinct and sufficiently large integers, we have $M\sigma^{\vec{x}}_{\vec{k}} \ltimes N\sigma^{\vec{x}}_{\vec{k}}$. Taking $l = 0$, Böhm-separability implies separability.

The results of the following lemma are, again, standard material. They adapt the setup used by Krivine for the strong separation of $\eta$-distinct $\beta$-normal forms [28, Chapter 5].

**LEMMA 6.11.** *We have $M \ltimes_{\mathcal{B}} N$ as soon as one of the following holds:*

1. $M \in \Lambda_{\mathrm{hn}}$ *and* $N \notin \Lambda_{\mathrm{hn}}$;
2. $M =_{\beta\eta} M' \ltimes_{\mathcal{B}} N' =_{\beta\eta} N$;
3. $M = \lambda x.M'$ *and* $N = \lambda x.N'$ *with* $M' \ltimes_{\mathcal{B}} N'$;
4. $M = x\,\vec{M}$ *and* $N = y\,\vec{N}$ *with* $x \neq y$ *or* $|\vec{M}| \neq |\vec{N}|$;
5. $M = x\,M_0\cdots M_k$ *and* $N = x\,N_0\cdots N_k$ *with* $M_i \ltimes_{\mathcal{B}} N_i$ *for some* $i \leq k$.

We are now ready to establish the separation theorem. The canonical structure of normal extensional resource terms allows us to proceed by a simple induction, very similar to the proof of strong separation on $\beta$-normal $\lambda$-terms (compare, e.g., with [28, Lemma 5.10]).

---

[20] Our terminology deviates from the one used by, e.g., Coppo *et al.* [12]: in their terminology, one would say that the set $\{M, N\}$ is separable (resp. semi-separable) if $M \bowtie N$ (resp. $M \ltimes N$ or $N \ltimes M$).

**THEOREM 6.12 (Separation).** *If $M \neq_{\mathcal{T}_\eta} N$ then $M \ltimes N$ or $N \ltimes M$.*

**PROOF.** We prove by induction on $m$ that, if $m \in \mathcal{NT}_\eta(M) \setminus \mathcal{NT}_\eta(N)$, then $M \ltimes_{\mathcal{B}} N$. In this case $\mathcal{NT}_\eta(M) \neq 0$, hence $M \in \Lambda_{\mathrm{hn}}$ by Lemma 6.6. If $N \notin \Lambda_{\mathrm{hn}}$, we conclude directly by Item 1 of Lemma 6.11.

Otherwise, since $=_{\mathcal{T}_\eta}$ is a $\lambda$-theory, Item 2 allows us to $\beta$-reduce both $M$ and $N$ to bring them into head normal form: $M = \lambda\vec{x}.x\,\vec{M}$ and $N = \lambda\vec{x}'.x'\,\vec{N}$, with $\vec{M} = \langle M_1, \ldots, M_k\rangle$ and $\vec{N} = \langle N_1, \ldots, N_{k'}\rangle$. Then we can write $m = \lambda\vec{x}.\lambda\vec{y}.x\,\vec{m}$. And since $=_{\mathcal{T}_\eta}$ is extensional, Item 2 again allows to $\eta$-expand inside $M$ and $N$. We can thus ensure that $|\vec{x}| = |\vec{x}'|$, and that $k$ is large enough to write $\vec{m} = \bar{m}_1 :: \cdots :: \bar{m}_k :: \iota$. By $\alpha$-equivalence, we can further assume $\vec{x} = \vec{x}'$. By iterating Item 3, and observing that $\lambda z.p \in \mathcal{NT}_\eta(\lambda z.P)$ iff $p \in \mathcal{NT}_\eta(P)$, we can assume that $M = x\,\vec{M}$, $N = x'\,\vec{N}$ and $m = \lambda\vec{y}.x\,\vec{m}$.

If $x \neq x'$ or $|\vec{M}| \neq |\vec{N}|$, we conclude by Item 4. Otherwise, observe that

$$\mathcal{NT}_\eta(M) = \lambda\vec{y}.x\,\mathcal{NT}_\eta(M_1)^! :: \cdots :: \mathcal{NT}_\eta(M_k)^! :: \vec{y}^{\,!} \quad \text{and}$$
$$\mathcal{NT}_\eta(N) = \lambda\vec{y}.x\,\mathcal{NT}_\eta(N_1)^! :: \cdots :: \mathcal{NT}_\eta(N_k)^! :: \vec{y}^{\,!}\ .$$

Since $\iota \in \vec{y}^{\,!}$ and $m \notin \mathcal{NT}_\eta(N)$, there must be $i$ such that $\bar{m}_i \notin \mathcal{NT}_\eta(N_i)^!$. Hence there must be $m_i \in \bar{m}_i$ such that $m_i \notin \mathcal{NT}_\eta(N_i)$. Since moreover $\bar{m}_i \in \mathcal{NT}_\eta(M_i)^!$, we have $m_i \in \mathcal{NT}_\eta(M_i)$. By applying the induction hypothesis to $m_i$, we obtain $M_i \ltimes_{\mathcal{B}} N_i$, and conclude using Item 5. ■

We have thus established that, if $M \neq_{\mathcal{T}_\eta} N$ then $M \neq_{\mathrm{hn}} N$, which is sufficient to ensure that $=_{\mathcal{T}_\eta}$ contains every other consistent and sensible $\lambda$-theory.

**THEOREM 6.13.** *Given two $\lambda$-terms $M$ and $N$, we have $M =_{\mathcal{T}_\eta} N$ iff $M =_{\mathrm{hn}} N$, and $=_{\mathcal{T}_\eta}$ is the maximum consistent and sensible $\lambda$-theory* $\mathbf{H}^*$.

In particular, Example 5.18 provides a new proof that $\mathbf{J} =_{\mathrm{hn}} \mathbf{I}$.

## 7. A relational model

By Theorem 6.13, to construct a model of $\mathbf{H}^*$, it is sufficient to give a model of the reduction of resource vectors. We exploit this approach to give a new proof of the fact that $\mathbf{H}^*$ is the $\lambda$-theory induced by a particular extensional reflexive object in the relational model of the simply typed $\lambda$-calculus.

We define the set $\mathcal{D}$ of **relational types** as $\mathcal{D} := \bigcup_{k\in\mathbb{N}} \mathcal{D}_k$ with $\mathcal{D}_0 := \emptyset$ and $\mathcal{D}_{k+1} := \mathcal{S}(\mathcal{D}_k)$. In other words, the elements of $\mathcal{D}$ are generated from $\iota$ by iterating the construction $\langle\bar{\alpha}, \beta\rangle \in \mathfrak{M}_{\mathrm{f}}(\mathcal{D}) \times \mathcal{D} \mapsto \bar{\alpha} :: \beta \in \mathcal{D}$, subject to the identity $[\,] :: \iota = \iota$. Note that $\mathcal{D}$ is nothing but the extensional reflexive object of the cartesian closed category **MRel** put forward by Bucciarelli, Ehrhard and Manzonetto [8] — as an example of the construction of an extensional $\lambda$-theory based on reflexive object in a cartesian closed category having “not enough points”. The fact that this $\lambda$-theory characterizes $\mathbf{H}^*$ was later proved by Manzonetto [30].

## 7.1 Relational semantics of λ-terms as a type system

Let us recall that the interpretation of $\lambda$-terms in this model can be described by a kind of non-idempotent intersection type system: the “::” constructor acts as an arrow connective, while the monoid structure of multisets induces the non-idempotent intersection operator.

More explicitly, we first define **relational typing contexts** (denoted by greek capitals $\Gamma$, $\Delta$, $\Phi$) as functions $\mathcal{V} \to \mathfrak{M}_{\mathrm{f}}(\mathcal{D})$ with finite support, *i.e.* $\Gamma : \mathcal{V} \to \mathfrak{M}_{\mathrm{f}}(\mathcal{D})$ is a relational typing context if $\{x \in \mathcal{V} \mid \Gamma(x) \neq [\,]\}$ is finite. We write $\star$ for the empty context: $\star(x) = [\,]$ for each $x \in \mathcal{V}$. We write $x : \bar{m}$ for the context $\Gamma$ such that $\Gamma(x) = \bar{m}$ and $\Gamma(y) = [\,]$ when $x \neq y$. And we define the concatenation of contexts point-wise: $(\Gamma * \Delta)(x) := \Gamma(x) * \Delta(x)$. We also write $\Gamma - x$ for the context such that $(\Gamma - x)(x) = [\,]$ and $(\Gamma - x)(y) = \Gamma(y)$ when $x \neq y$.

Similarly, if $\vec{\alpha} = \langle \bar{\alpha}_i \rangle_{i \in \mathbb{N}} \in \mathcal{S}(\mathcal{D})$, we write $\vec{x} : \vec{\alpha}$ for the context $\Gamma$ such that $\Gamma(\vec{x}(i)) = \bar{\alpha}_i$ for $i \in \mathbb{N}$, and $\Gamma(y) = [\,]$ if $y \notin \vec{x}$. We also write $\Gamma - \vec{x}$ for the context such that $(\Gamma - \vec{x})(\vec{x}(i)) = [\,]$ and $(\Gamma - \vec{x})(y) = \Gamma(y)$ when $y \notin \vec{x}$. Finally, $\Gamma(\vec{x})$ denotes the sequence $\langle \Gamma(\vec{x}(i)) \rangle_{i \in \mathbb{N}}$, which is a stream because $\Gamma$ has finite support.

The relational semantics $[\![M]\!]$ can then be computed as the set

$$\{\langle \Gamma, \alpha \rangle \mid \Gamma \vdash M : \alpha\}$$

where the type system is described by the following rules:

$$\frac{}{x : [\alpha] \vdash x : \alpha} \qquad \frac{\Gamma \vdash M : \beta}{\Gamma - x \vdash \lambda x.M : \Gamma(x) :: \beta}$$

$$\frac{\Gamma \vdash M : \bar{\alpha} :: \beta \qquad \Delta \vdash_! N : \bar{\alpha}}{\Gamma * \Delta \vdash M\,N : \beta} \qquad \frac{\Gamma_1 \vdash M : \alpha_1 \qquad \cdots \qquad \Gamma_k \vdash M : \alpha_k}{\Gamma_1 * \cdots * \Gamma_k \vdash_! M : [\alpha_1, \ldots, \alpha_k]}$$

where the additional kind of judgement ($\Gamma \vdash_! M : \bar{\alpha}$) used in the application rule reflects the promotion rule of linear logic.

It is well known that this interpretation is stable under $\beta$-reduction and $\eta$-expansion, and that it factors through (ordinary) Taylor expansion (see, e.g., [31, Proposition 5.9]), the semantics of resource terms being given by a natural variant of the above rules:

$$[\![M]\!] = [\![\mathcal{T}(M)]\!] = [\![\mathcal{N}(\mathcal{T}(M))]\!] = \bigcup_{m \in \mathcal{T}(M)} [\![m]\!] \,.$$

We will now show a similar result for our extensional version of Taylor expansion. Since this semantics is set based, we only consider the qualitative version of extensional Taylor expansion, *i.e.* we fix the set of coefficients to be the boolean semiring $\mathbb{B}$.

A novel feature of our factorization is that, for extensional normal resource terms, the relational semantics is always a singleton: each normal resource term is mapped to a single point in the relational semantics. Even more strikingly, the interpretation of a non-normal resource term is not only a finite set, but has at most one element. By contrast, in the ordinary resource calculus, normal terms always have an infinite semantics: for instance, the semantics of a single variable $x$ is $\{\langle x : [\alpha], \alpha\rangle \mid \alpha \in \mathcal{D}\}$ (the same as if it was considered as a $\lambda$-term). Also, the compatibility of the semantics with $\eta$-expansion is reflected by the properties of Taylor expansion, instead of being an additional property of the model, to be checked by hand.

## 7.2 Relational semantics of extensional resource terms

Each syntactic category of expressions will correspond to a specific category of types:

- value expressions will be typed with elements of $\mathcal{D}$;
- bag terms with elements of $\mathfrak{M}_{\mathrm{f}}(\mathcal{D})$;
- stream terms with elements of $\mathcal{S}(\mathcal{D})$;
- base terms with a single, newly introduced base type $o$.

The underlying idea is that a value expression of type $\alpha = \langle \bar{\alpha}_i \rangle_{i\in\mathbb{N}}$ expects a sequence of bags $\vec{m} = \langle \bar{m}_i \rangle_{i\in\mathbb{N}}$ with $\bar{m}_i$ of type $\bar{\alpha}_i$ to produce a successful interaction (of type $o$). Although the types are incidentally taken from the same set $\mathcal{D} = \mathcal{S}(\mathcal{D})$, the intuitive meaning of typing for value expressions and for stream terms is thus quite different. To fit this intuition, we redefine $\mathcal{D}$ simply to be able to distinguish between $\mathcal{D}$ and $\mathcal{S}(\mathcal{D})$: we believe this will clarify the presentation, although the remainder of this section might be carried out identically without this notational trick.

We define the sets $\mathcal{D}_{\mathrm{v}}$ of **value types** and $\mathcal{D}_{\mathrm{s}}$ of **stream types**, simultaneously by mutual induction as follows:

- $\alpha \in \mathcal{D}_{\mathrm{v}}$ if $\alpha = \vec{\alpha} \multimap o$ with $\vec{\alpha} \in \mathcal{D}_{\mathrm{s}}$;
- $\vec{\alpha} \in \mathcal{D}_{\mathrm{s}}$ if $\vec{\alpha} \in \mathcal{S}(\mathcal{D}_{\mathrm{v}})$;

so that $\vec{\alpha} \mapsto \vec{\alpha} \multimap o$ defines a bijection from $\mathcal{D}_{\mathrm{s}} = \mathcal{S}(\mathcal{D}_{\mathrm{v}})$ to $\mathcal{D}_{\mathrm{v}}$. We will also write $\mathcal{D}_! := \mathfrak{M}_{\mathrm{f}}(\mathcal{D}_{\mathrm{v}})$ for the set of **bag types** and $\mathcal{D}_{\mathrm{b}} := \{o\}$ for the singleton containing the **base type**. We call **type term** (denoted by $\rho, \sigma, \tau$) any of a value type, bag type, stream type or the base type.

A type context is now a function $\Gamma : \mathcal{V} \to \mathcal{D}_!$, whose value is almost always the empty bag. The type system involves four kinds of jugdements:

$$\Gamma \vdash_{\mathrm{v}} m : \alpha \qquad \Gamma \vdash_! \bar{m} : \bar{\alpha} \qquad \Gamma \vdash_{\mathrm{s}} \vec{m} : \vec{\alpha} \qquad \Gamma \vdash_{\mathrm{b}} a : o$$

and we denote by $\Gamma \vdash u : \rho$ any judgement as above. The rules are in Figure 13. Note that, In particular, the last rule allows us to derive $\star \vdash_! [\,] : [\,]$.

As we have announced, contrasting with the ordinary resource calculus, the relational interpretation of an extensional resource term will have at most one element:

$$\frac{\Gamma \vdash_{\mathrm{v}} m : \vec{\alpha} \multimap o \qquad \Delta \vdash_{\mathrm{s}} \vec{n} : \vec{\alpha}}{\Gamma * \Delta \vdash_{\mathrm{b}} m\,\vec{n} : o} \qquad \frac{\Gamma \vdash_{\mathrm{s}} \vec{m} : \vec{\alpha}}{\Gamma * x : [\vec{\alpha} \multimap o] \vdash_{\mathrm{b}} x\,\vec{m} : o}$$

$$\frac{}{\star \vdash_{\mathrm{s}} \iota : \iota} \qquad \frac{\Gamma \vdash_{!} \bar{m} : \bar{\alpha} \qquad \Delta \vdash_{\mathrm{s}} \vec{n} : \vec{\beta}}{\Gamma * \Delta \vdash_{\mathrm{s}} (\bar{m} :: \vec{n}) : (\bar{\alpha} :: \vec{\beta})}$$

$$\frac{\Gamma \vdash_{\mathrm{b}} a : o}{\Gamma - \vec{x} \vdash_{\mathrm{v}} \lambda \vec{x}.a : \Gamma(\vec{x}) \multimap o} \qquad \frac{\Gamma_1 \vdash_{\mathrm{v}} m_1 : \alpha_1 \qquad \cdots \qquad \Gamma_k \vdash_{\mathrm{v}} m_k : \alpha_k}{\Gamma_1 * \cdots * \Gamma_k \vdash_{!} [m_1, \ldots, m_k] : [\alpha_1, \ldots, \alpha_k]}$$

**Figure 13.** Relational typing system for the extensional resource calculus

---

**LEMMA 7.1.** *Each resource term $u$ admits at most one derivable typing judgement. If moreover $u$ is normal, then it is typable.*

**PROOF.** The proof is straightforward, by induction on $u$. Note that $u$ being normal forbids the first rule for base terms, which is the only one in which a constraint on the premises is imposed. ■

We write $\vdash u$ when $u$ is typable, and in this case we write $\mathrm{ctx}(u)$ and $\mathrm{type}(u)$ respectively for the unique context and type term such that $\mathrm{ctx}(u) \vdash u : \mathrm{type}(u)$ is derivable.

**LEMMA 7.2.** *If $\vdash u$ then $|\mathrm{ctx}(u)(x)|$ is the number of occurrences of $x$ in $u$.*

**PROOF.** By a straightforward induction on $u$. ■

Note that the functions $\mathrm{type}(-)$ and $\mathrm{ctx}(-)$ are not injective, even jointly:

**EXAMPLE 7.3.** Consider

$$\vec{m}_1 := [\lambda \vec{y}.x\ \iota] :: [\lambda \vec{y}.x\ [m] :: \iota] :: \iota$$

and

$$\vec{m}_2 := [\lambda \vec{y}.x\ [m] :: \iota] :: [\lambda \vec{y}.x\ \iota] :: \iota$$

where $m$ is a typeable closed value and $x \notin \vec{y}$, so that $\vec{m}_1$ and $\vec{m}_2$ differ only by the order of their first two bags. Then

$$\mathrm{type}(\vec{m}_1) = \mathrm{type}(\vec{m}_2) = [\iota \multimap o] :: [\iota \multimap o] :: \iota$$

because the variables of $\vec{y}$ have no occurrences in subterms, while

$$\mathrm{ctx}(\vec{m}_1) = \mathrm{ctx}(\vec{m}_2) = x : [\iota \multimap o, ([\mathrm{type}(m)] :: \iota) \multimap o]$$

because $x$ occurs in twice in each of $\vec{m}_1$ and $\vec{m}_2$, once applied to $\iota$, and then applied to $[m] :: \iota$. The relational semantics cannot *"see"* that these occurrences are swapped.[21] ◆

Moreover, not all types are inhabited by a closed term. This is easily observed on normal terms:

**PROPOSITION 7.4.** *There is no normal term $u$ such that*

- *$u = a$ with $\star \vdash_{\mathrm{b}} a : o$, or*
- *$u = m$ with $\star \vdash_{\mathrm{v}} m : \iota \multimap o$, or*
- *$u = \bar{m}$ with $\star \vdash_{!} \bar{m} : [\iota \multimap o]$, or*
- *$u = \vec{m}$ with $\star \vdash_{\mathrm{v}} \vec{m} : [\iota \multimap o] :: \iota$.*

**PROOF.** Given the shape of the unique applicable rule for a normal base term, we have $\mathsf{ctx}(a) \neq \star$ when $\vdash a$ and $a$ is normal. Each of the other three statements follows directly from the previous one. ■

The same result holds for all terms, as will follow from Lemma 7.8, which ensures the invariance of typing under reduction. We first characterize typing in substitutions, in the next two lemmas.

**LEMMA 7.5.** *If $u' \in u[\bar{n}/x]$ and $\vdash u'$ then $\vdash u$, $\vdash \bar{n}$, $\mathsf{type}(\bar{n}) = \mathsf{ctx}(u)(x)$, $\mathsf{type}(u') = \mathsf{type}(u)$, and $\mathsf{ctx}(u') = (\mathsf{ctx}(u) - x) * \mathsf{ctx}(\bar{n})$.*

**PROOF.** The proof is by induction on $u$.

If $u = [m_1, \dots, m_k]$ then we must have $u' = [m'_1, \dots, m'_k]$ and $\bar{n} = \bar{n}_1 * \cdots * \bar{n}_k$ with $m'_i \in \bar{m}[\bar{n}_i/x]$ for $1 \leq i \leq k$. Then $\vdash m'_i$ and we apply the induction hypothesis to each $m_i$ for $1 \leq i \leq k$. We obtain that $\vdash m_i$ and $\vdash \bar{n}_i$ for $1 \leq i \leq k$: it follows that $\vdash u$ and $\vdash \bar{n}$. Moreover, $\mathsf{type}(\bar{n}_i) = \mathsf{ctx}(m_i)(x)$ for $1 \leq i \leq k$, so $\mathsf{type}(\bar{n}) = \mathsf{type}(\bar{n}_1) * \cdots * \mathsf{type}(\bar{n}_k) = (\mathsf{ctx}(m_1) * \cdots * \mathsf{ctx}(m_k))(x) = \mathsf{ctx}(u)(x)$. We also have $\mathsf{type}(m'_i) = \mathsf{type}(m_i)$ for $1 \leq i \leq k$, hence $\mathsf{type}(u') = \mathsf{type}(u)$. Finally, we have $\mathsf{ctx}(m'_i) = (\mathsf{ctx}(m_i) - x) * \mathsf{ctx}(\bar{n}_i)$ for $1 \leq i \leq k$, hence $\mathsf{ctx}(u') = \mathsf{ctx}(m'_1) * \cdots * \mathsf{ctx}(m'_k) = (\mathsf{ctx}(u) - x) * \mathsf{ctx}(\bar{n})$,

If $u = \iota$ then $u' = \iota$ and $\bar{n} = [\,]$, which entails all the desired properties. The cases of $u = \bar{m} :: \vec{p} \neq \iota$ and $u = m\,\vec{p}$ are similar to that of $u = [m_1, m_2]$. The case of $u = \lambda \vec{y}.a$ (choosing $\vec{y} \not\ni x$ and $\vec{y} \cap \mathcal{V}(\bar{n}) = \emptyset$) is similar to that of $u = [m_1]$.

The only remaining case is that of $u = z\,\vec{m}$. If $z \neq x$, the treatment is, again, similar to that of $u = [m_1]$. Now assume $z = x$. Then we can write $u' = n\,\vec{m}'$ and $\bar{n} = [n] * \bar{n}_1$ with $\vec{m}' \in \vec{m}[\bar{n}_1/x]$. We have $\vdash n$ and $\vdash \vec{m}'$, and moreover $\mathsf{type}(n) = \mathsf{type}(\vec{m}') \multimap o$. We apply the induction hypothesis to $\vec{m}$. We obtain $\vdash \vec{m}$ and $\vdash \bar{n}_1$: it follows that $\vdash u$ and $\vdash \bar{n}$. Moreover, $\mathsf{type}(\vec{m}') = \mathsf{type}(\vec{m})$ and $\mathsf{type}(\bar{n}_1) = \mathsf{ctx}(\vec{m})(x)$. So $\mathsf{type}(\bar{n}) = [\mathsf{type}(n)] * \mathsf{type}(\bar{n}_1) = [\mathsf{type}(\vec{m}') \multimap$

---

[21] This contrasts with the one-to-one correspondence between normal extensional resource terms and (isomorphism classes of) augmentations in game semantics, to be developed in Section 8.

$o] * \mathrm{ctx}(\vec{m})(x) = [\mathrm{type}(\vec{m}) \multimap o] * \mathrm{ctx}(\vec{m})(x) = \mathrm{ctx}(u)(x)$ and $\mathrm{ctx}(u') = \mathrm{ctx}(n) * \mathrm{ctx}(\vec{m}') = \mathrm{ctx}(n) * (\mathrm{ctx}(\vec{m}) - x) * \mathrm{ctx}(\bar{n}_1) = (\mathrm{ctx}(u) - x) * \mathrm{ctx}(\bar{n})$ since $\mathrm{ctx}(u) - x = \mathrm{ctx}(\vec{m}) - x$. ■

**LEMMA 7.6.** *If* $\vdash u$, $\vdash \bar{n}$, *and* $\mathrm{type}(\bar{n}) = \mathrm{ctx}(u)(x)$, *then there exists* $u' \in u[\bar{n}/x]$ *such that* $(\mathrm{ctx}(u) - x) * \mathrm{ctx}(\bar{n}) \vdash u' : \mathrm{type}(u)$.

**PROOF.** Write $U' := u[\bar{n}/x]$. The proof is by induction on $u$.

If $u = [m_1, \dots, m_k]$ then $\vdash m_i$ for $1 \leq i \leq k$. Since $\mathrm{type}(\bar{n}) = \mathrm{ctx}(u)(x) = \mathrm{ctx}(m_1)(x) * \cdots * \mathrm{ctx}(m_k)(x)$, we can write $\bar{n} = \bar{n}_1 * \cdots * \bar{n}_k$ with $\mathrm{type}(\bar{n}_i) = \mathrm{ctx}(m_i)(x)$ for $1 \leq i \leq k$. We apply the induction hypothesis to $m_i$ for $1 \leq i \leq k$: we obtain $m'_i \in m_i[\bar{n}_i/x]$ such that $(\mathrm{ctx}(m_i) - x) * \mathrm{ctx}(\bar{n}_i) \vdash_{\mathrm{v}} m'_i : \mathrm{type}(m_i)$. We conclude by setting $u' := [m'_1, \dots, m'_k]$.

As in the previous lemma, the cases of streams, values, and base terms $p\,\vec{m}$ or $z\,\vec{m}$ with $z \neq x$ follow the same pattern as for bags.

Finally, if $u = x\,\vec{m}$ then $\vdash \vec{m}$ and $\mathrm{type}(\bar{n}) = [\mathrm{type}(\vec{m}) \multimap o] * \mathrm{ctx}(\vec{m})(x)$. Then we can write $\bar{n} = [n] * \bar{n}_1$ with $\mathrm{type}(n) = \mathrm{type}(\vec{m}) \multimap o$ and $\mathrm{type}(\bar{n}_1) = \mathrm{ctx}(\vec{m})(x)$. We apply the induction hypothesis to $\vec{m}$, and obtain $\vec{m}' \in \vec{m}[\bar{n}_1/x]$ such that $(\mathrm{ctx}(\vec{m}) - x) * \mathrm{ctx}(\bar{n}_1) \vdash_{\mathrm{s}} \vec{m}' : \mathrm{type}(\vec{m})$. We conclude by setting $u' := n\,\vec{m}'$. ■

**LEMMA 7.7.** *We have* $\vdash m$ *iff* $\vdash \lambda x.m$ *and then:*

- $\mathrm{type}(\lambda x.m) = (\mathrm{ctx}(m)(x) :: \vec{\beta}) \multimap o$ *iff* $\mathrm{type}(m) = \vec{\beta} \multimap o$;
- $\mathrm{ctx}(\lambda x.m) = \mathrm{ctx}(m) - x$.

**PROOF.** Direct application of the definitions. ■

Now we extend the typing system to sums: if $U \in \Sigma\Delta_{\mathrm{t}}$, we set $\Gamma \vdash U : \rho$ when there exists $u \in \mathrm{supp}(U)$ such that $\Gamma \vdash u : \rho$. We obtain:

**LEMMA 7.8.** *Assume* $U \to_{\mathrm{r}} U'$. *We have* $\Gamma \vdash U : \rho$ *iff* $\Gamma \vdash U' : \rho$.

**PROOF.** We first treat the case of redexes. First assume $U = (\lambda x.m)\,\bar{n} :: \vec{p}$ and $U' = (m[\bar{n}/x])\,\vec{p}$.

If $\Gamma \vdash_{\mathrm{b}} U : o$ then $\vdash m$, $\vdash \bar{n}$ and $\vdash \vec{p}$, and moreover: $\mathrm{type}(\bar{n}) = \mathrm{ctx}(m)(x)$, $\mathrm{type}(m) = \mathrm{type}(\vec{p}) \multimap o$ and $\Gamma = (\mathrm{ctx}(m) - x) * \mathrm{ctx}(\bar{n}) * \mathrm{ctx}(\vec{p})$. Lemma 7.6 yields $m' \in m[\bar{n}/x]$ such that $(\mathrm{ctx}(m) - x) * \mathrm{ctx}(\bar{n}) \vdash_{\mathrm{v}} m' : \mathrm{type}(m)$, and we set $u' := m'\,\vec{p}$ to obtain $u' \in U'$ and $\Gamma \vdash_{\mathrm{b}} u' : o$.

Conversely, if $\Gamma \vdash_{\mathrm{b}} U' : o$, there exists $m' \in m[\bar{n}/x]$ such that $\Gamma \vdash_{\mathrm{b}} m'\,\vec{p} : o$. It follows that $\vdash m'$ and $\vdash \vec{p}$, and moreover $\Gamma = \mathrm{ctx}(m') * \mathrm{ctx}(\vec{p})$ and $\mathrm{type}(m') = \mathrm{type}(\vec{p}) \multimap o$. Lemma 7.5 entails that $\vdash m$, $\vdash \bar{n}$, and moreover $\mathrm{type}(\bar{n}) = \mathrm{ctx}(m)(x)$, $\mathrm{type}(m) = \mathrm{type}(m')$, and $\mathrm{ctx}(m') = (\mathrm{ctx}(m) - x) * \mathrm{ctx}(\bar{n})$. It follows that $\mathrm{type}(\lambda x.m) = (\mathrm{type}(\bar{n}) :: \mathrm{type}(\vec{p})) \multimap o$ and $\Gamma = \mathrm{ctx}(\lambda x.m) * \mathrm{ctx}(\bar{n}) * \mathrm{ctx}(\vec{p})$, hence $\Gamma \vdash_{\mathrm{b}} U : o$.

Now assume $U = (\lambda \vec{x}.a)\,\iota$ and $U' = a \lightning \vec{x}$.

If $\Gamma \vdash_{\mathrm{b}} U : o$ then $\vdash a$ and $\mathrm{ctx}(a)(\vec{x}) = \iota$. By Lemma 7.2, $\vec{x} \cap \mathcal{V}(a) = \emptyset$ so $U' = a$. Moreover, $\Gamma = \mathrm{ctx}(a) - \vec{x} = \mathrm{ctx}(a)$.

Conversely, if $\Gamma \vdash_{\mathrm{b}} U' : o$ then $U' \neq 0$ hence $U' = a$ with $\vec{x} \cap \mathcal{V}(a) = \emptyset$. Hence, by Lemma 7.2, $\mathrm{ctx}(a)(\vec{x}) = \iota$ so $\mathrm{type}(\lambda\vec{x}.a) = \iota \multimap o$ and $\mathrm{ctx}(\lambda\vec{x}.a) = \Gamma$. We obtain $\Gamma \vdash_{\mathrm{b}} U : o$.

Next, we treat the case of $U = u \in \Delta_{\mathrm{t}}$, and $U \mapsto_{\mathrm{r}} U'$. We obtain the result by a straightforward induction on the definition of $\mapsto_{\mathrm{r}}$.

Finally, if $U = u + V$ and $U' = U' + V$, with $u \mapsto_{\mathrm{r}} U'$, we conclude directly from the previous case, observing that $\Gamma \vdash u+V : \rho$ (resp. $\Gamma \vdash U'+V : \rho$) iff $\Gamma \vdash u : \rho$ (resp. $\Gamma \vdash U' : \rho$) or $\Gamma \vdash V : \rho$. ■

**LEMMA 7.9.** *For any $u \in \Delta_{\mathrm{t}}$:*

- *either $\nvdash u$ and $\mathcal{N}(u) = 0$;*
- *or $\vdash u$, and then $\Gamma \vdash \mathcal{N}(u) : \rho$ iff $\Gamma = \mathrm{ctx}(u)$ and $\rho = \mathrm{type}(u)$ — in particular $\mathcal{N}(u) \neq 0$.*

**PROOF.** Since $u \to_{\mathrm{r}}^{*} \mathcal{N}(u)$, we can apply the previous lemma:

- if $\nvdash u$ then $\nvdash \mathcal{N}(u)$, hence $\mathcal{N}(u) = 0$ by Lemma 7.1;
- if $\vdash u$ then $\Gamma \vdash \mathcal{N}(u) : \rho$ iff $\Gamma \vdash u : \rho$ iff $\Gamma = \mathrm{ctx}(u)$ and $\rho = \mathrm{type}(u)$, again by Lemma 7.1. ■

## 7.3 Taylor expansion of relational semantics

We now show that the equational theory induced by the relational interpretation of $\lambda$-terms in $\mathcal{D}$ is nothing but $=_{\mathcal{T}_\eta}$, or equivalently $\mathbf{H}^*$. We first reformulate the relational type system for $\lambda$-terms, to account for the distinction between $\mathcal{D}_{\mathrm{v}}$ and $\mathcal{D}_{\mathrm{s}}$:

$$\frac{}{x : [\alpha] \vdash x : \alpha} \qquad \frac{\Gamma \vdash M : \vec{\beta} \multimap o}{\Gamma - x \vdash \lambda x.M : (\Gamma(x) :: \vec{\beta}) \multimap o}$$

$$\frac{\Gamma \vdash M : (\bar{\alpha} :: \vec{\beta}) \multimap o \qquad \Delta \vdash_{!} N : \bar{\alpha}}{\Gamma * \Delta \vdash M\,N : \vec{\beta} \multimap o} \qquad \frac{\Gamma_1 \vdash M : \alpha_1 \qquad \cdots \qquad \Gamma_k \vdash M : \alpha_k}{\Gamma_1 * \cdots * \Gamma_k \vdash_{!} M : [\alpha_1, \dots, \alpha_k]}$$

For any $U \in \mathbb{B}\langle\Delta_{\mathrm{t}}\rangle$, we set $\Gamma \vdash U : \rho$ iff there is $u \in U$ with $\Gamma \vdash u : \rho$. Lemma 7.8 entails:

**LEMMA 7.10.** *If $U, U' \in \mathbb{B}\langle\Delta_{\mathrm{t}}\rangle$ and $U \rightsquigarrow U'$ then $\Gamma \vdash U : \rho$ iff $\Gamma \vdash U' : \rho$.*

Then we show that the relational interpretation factors through Taylor expansion:

**THEOREM 7.11.** *For any $\lambda$-term $M$, the following are equivalent:*

$$\Gamma \vdash M : \alpha; \qquad \Gamma \vdash \mathcal{T}_\eta(M) : \alpha; \qquad \Gamma \vdash \mathcal{T}_h(M) : \alpha\,.$$

The equivalence between the last two statements follow from Lemma 7.10 and Theorem 5.16. A first step to establish the equivalence with the first one is to consider the expansion of variables.

**LEMMA 7.12.** *Let $x \in \mathcal{V}$ and $\vec{x} \in \mathcal{V}_{\mathrm{s}}$. Then:*

- *for each* $\alpha \in \mathcal{D}_{\mathrm{v}}$, *there is a unique* $\mathrm{w}_x(\alpha) \in x^\eta$ *with* $\mathrm{type}(\mathrm{w}_x(\alpha)) = \alpha$;
- *for each* $\bar\alpha \in \mathcal{D}_!$, *there is a unique* $\mathrm{w}^!_x(\alpha) \in (x^\eta)^!$ *with* $\mathrm{type}(\mathrm{w}^!_x(\bar\alpha)) = \bar\alpha$;
- *for each* $\vec\alpha \in \mathcal{D}_{\mathrm{s}}$, *there is a unique* $\mathrm{w}_{\vec x}(\vec\alpha) \in \vec x^{\,!}$ *with* $\mathrm{type}(\mathrm{w}_{\vec x}(\vec\alpha)) = \vec\alpha$.

*Moreover:*

$$x : [\alpha] \vdash_{\mathrm{v}} \mathrm{w}_x(\alpha) : \alpha, \quad x : \bar\alpha \vdash_! \mathrm{w}^!_x(\bar\alpha) : \bar\alpha, \quad \vec x : \vec\alpha \vdash_{\mathrm{s}} \mathrm{w}_{\vec x}(\vec\alpha) : \vec\alpha,$$

*and:*

- *if* $m \in x^\eta$ *then* $\vdash m$ *and* $m = \mathrm{w}_x(\mathrm{type}(m))$;
- *if* $\bar m \in (x^\eta)^!$ *then* $\vdash \bar m$ *and* $\bar m = \mathrm{w}^!_x(\mathrm{type}(\bar m))$;
- *if* $\vec m \in \vec x^{\,!}$ *then* $\vdash \vec m$ *and* $\vec m = \mathrm{w}_{\vec x}(\mathrm{type}(\vec m))$.

**PROOF.** We define $\mathrm{w}_x(\alpha)$, $\mathrm{w}^!_x(\bar\alpha)$ and $\mathrm{w}_{\vec x}(\vec\alpha)$ by mutual induction on $\alpha$, $\bar\alpha$ and $\vec\alpha$. Given $\alpha = \vec\alpha \multimap o$, we chose $\vec y \not\ni x$ and set $\mathrm{w}_x(\alpha) := \lambda\vec y.x\,\mathrm{w}_{\vec y}(\vec\alpha)$. If $\bar\alpha = [\alpha_1, \dots, \alpha_k]$, we set $\mathrm{w}^!_x(\bar\alpha) := [\mathrm{w}_x(\alpha_1), \dots, \mathrm{w}_x(\alpha_k)]$. Finally, if $\vec\alpha = \langle\bar\alpha_i\rangle_{i\in\mathbb{N}}$, we set $\mathrm{w}_{\vec x}(\vec\alpha) := \langle\mathrm{w}^!_{\vec x(i)}(\bar\alpha_i)\rangle_{i\in\mathbb{N}}$.

The fact that these expressions are the only ones satisfying the requirements, together with the induced typing judgements, are easily established by induction on type terms. Finally, the last three items are obtained by mutual induction on resource terms. ■

**PROOF OF THEOREM 7.11.** We establish that:

- $\Gamma \vdash M : \alpha$ iff there exists $m \in \mathcal{T}_\eta(M)$ with $\Gamma \vdash_{\mathrm{v}} m : \alpha$, and
- $\Gamma \vdash_! M : \bar\alpha$ iff there exists $\bar m \in \mathcal{T}^!_\eta(M)$ with $\Gamma \vdash_! \bar m : \bar\alpha$;

by induction on $M$. The second statement follows directly from the first: if $\bar\alpha = [\alpha_1, \dots, \alpha_k]$ then

$$\begin{aligned} \Gamma \vdash_! M : \bar\alpha \quad & \text{iff} \quad \Gamma = \Gamma_1 * \cdots * \Gamma_k \text{ with } \Gamma_i \vdash M : \alpha_i \text{ for } 1 \le i \le k \\ & \text{iff} \quad \Gamma = \Gamma_1 * \cdots * \Gamma_k \text{ with } \Gamma_i \vdash_{\mathrm{v}} \mathcal{T}_\eta(M) : \alpha_i \text{ for } 1 \le i \le k \\ & \text{iff} \quad \Gamma \vdash_! \mathcal{T}_\eta(M) : \bar\alpha. \end{aligned}$$

First assume $M = x$. If $\Gamma \vdash M : \alpha$ then $\Gamma = x : [\alpha]$ and we have $x : [\alpha] \vdash_{\mathrm{v}} \mathrm{w}_x(\alpha) : \alpha$ with $\mathrm{w}_x(\alpha) \in x^\eta$. Conversely, if $m \in x^\eta$ with $\Gamma \vdash_{\mathrm{v}} m : \alpha$, we must have $m = \mathrm{w}_x(\alpha)$, hence $\Gamma = x : [\alpha]$.

Now assume $M = \lambda x.N$. If $\Gamma \vdash M : \alpha$ then $\Gamma = \Delta - x$ and $\alpha = (\Delta(x) :: \vec\gamma) \multimap o$, with $\Delta \vdash N : \vec\gamma \multimap o$. By induction hypothesis, this holds iff there exists $n \in \mathcal{T}_\eta(N)$ with $\Delta \vdash_{\mathrm{v}} n : \vec\gamma \multimap o$, which is equivalent to $\Delta - x \vdash_{\mathrm{v}} \lambda x.n : (\Delta(x) :: \vec\gamma) \multimap o$, *i.e.* $\Gamma \vdash_{\mathrm{v}} \lambda x.n : \alpha$.

Finally, assume $M = N\,P$. If $\Gamma \vdash M : \vec\alpha \multimap o$ then we can write $\Gamma = \Delta * \Phi$ and there exists $\bar\gamma \in \mathcal{D}_!$ such that $\Delta \vdash N : \bar\gamma :: \vec\alpha \multimap o$ and $\Phi \vdash_! P : \bar\gamma$. By induction hypothesis, we obtain $n \in \mathcal{T}_\eta(N)$ and $\bar p \in \mathcal{T}_\eta(P)^!$ such that $\Delta \vdash_{\mathrm{v}} n : \bar\gamma :: \vec\alpha \multimap o$ and $\Phi \vdash_! \bar p : \bar\gamma$. Let $\vec x$ be a fresh sequence variable. We have $\Phi * \vec x : \vec\alpha \vdash_{\mathrm{s}} \bar p :: \mathrm{w}_{\vec x}(\vec\alpha) : \bar\gamma :: \vec\alpha$ and then $\Delta * \Phi * \vec x : \vec\alpha \vdash_{\mathrm{b}} n\,\bar p :: \mathrm{w}_{\vec x}(\vec\alpha) : o$, and finally $\Gamma \vdash_{\mathrm{v}} \lambda\vec x.n\,\bar p :: \mathrm{w}_{\vec x}(\vec\alpha) : \vec\alpha \multimap o$, We conclude since $\lambda\vec x.n\,\bar p :: \mathrm{w}_{\vec x}(\vec\alpha) \in \mathcal{T}_\eta(M)$.

Conversely, if $m \in \mathcal{T}_\eta(M)$ with $\Gamma \vdash_{\mathrm{v}} m : \vec\alpha \multimap o$, then we must have $m = \lambda\vec x.n\,\bar p :: \vec m$ with $\vec x$ fresh, $n \in \mathcal{T}_\eta(N)$, $\bar p \in \mathcal{T}_\eta(P)^!$ and $\vec m \in \vec x^{\,!}$, so that $\Gamma = \Gamma' - \vec x$ with $\Gamma'(\vec x) = \vec\alpha$ and $\Gamma' \vdash_{\mathrm{b}} n\,\bar p :: \vec m : o$. Since $\vec x$ is fresh, $\Gamma'(\vec x) = \mathrm{ctx}(\vec m)(\vec x)$, hence $\vec m = \mathrm{w}_{\vec x}(\vec\alpha)$ and $\vec x : \vec\alpha \vdash_{\mathrm{s}} \vec m : \vec\alpha$. Then there must exist

$\bar{\gamma} \in \mathcal{D}_!$ and contexts $\Delta$ and $\Phi$ such that $\Gamma = \Delta * \Phi$, $\Delta \vdash_{\mathrm{v}} n : (\bar{\gamma} :: \vec{\alpha}) \multimap o$ and $\Phi \vdash_! \bar{p} : \bar{\gamma}$. By induction hypothesis, we obtain $\Delta \vdash N : (\bar{\gamma} :: \vec{\alpha}) \multimap o$ and $\Phi \vdash_! P : \bar{\gamma}$, hence $\Gamma \vdash M : \vec{\alpha} \multimap o$. ■

Corollary 5.17, Lemma 7.10, and Theorems 6.2 and 7.11, together with the inductive definition of the relational semantics entail that:

**COROLLARY 7.13.** *Setting $M =_{\mathcal{D}} N$ when $[\![M]\!] = [\![N]\!]$ defines a sensible extensional $\lambda$-theory.*

As we have stated above, this property is already well known [8]: the originality of our approach is to relate this semantics with extensional Taylor expansion. In particular, this allows us to give a new proof that $=_{\mathcal{D}}$ is $\mathbf{H}^*$: this was first established by Manzonetto [30], who gave sufficient axiomatic conditions on a cartesian closed category to host a reflexive object modelling $\mathbf{H}^*$. Here the result comes directly from the properties of extensional Taylor expansion:

**COROLLARY 7.14.** *The relations $=_{\mathcal{T}_\eta}$ and $=_{\mathcal{D}}$ coincide – hence $=_{\mathcal{D}}$ is $\mathbf{H}^*$.*

**PROOF.** We have just established that $=_{\mathcal{D}}$ is a consistent sensible $\lambda$-theory, hence $=_{\mathcal{D}} \subseteq =_{\mathrm{hn}}$ and we obtain $=_{\mathcal{D}} \subseteq =_{\mathcal{T}_\eta}$ by Theorem 6.12. For the reverse inclusion, assume $M =_{\mathcal{T}_\eta} N$: by Corollary 5.17 and Lemma 7.10 we obtain $[\![\mathcal{T}_\eta(M)]\!] = [\![\mathcal{T}_\eta(N)]\!]$, and Theorem 7.11 yields $[\![M]\!] = [\![N]\!]$. ■

## 8. Normal resource terms as augmentations

In this section, we show the correspondence with game semantics that initially motivated the introduction of extensional resource terms.

We briefly recall the basics of *pointer concurrent games* in the style of Blondeau-Patissier and Clairambault [4]. In this brief presentation we shall introduce *augmentations* as representations of normal extensional resource terms,[22] but only consider them as static objects: we leave for further work an account of the dynamics of extensional resource terms in game semantics, as previously developed in the typed case [5].

### 8.1 Arenas and their constructions

As usual in game semantics, the game is played on an *arena* – the arena presents all the available observable computational events on a given type, along with their causal dependencies, given by a forest order: we call **forest** any poset $A$ such that the principal ideal $[a]_A := \{a' \in |A| \mid a' \leq_A a\}$ of any element $a$ is finite and linearly ordered by $\leq_A$.

**DEFINITION 8.1.** An **arena** is $A = \langle |A|, \leq_A, \mathrm{pol}_A \rangle$ where the pair $\langle |A|, \leq_A \rangle$ (the set of **events** and their **causal** order) defines a countable forest, and the **polarity function** $\mathrm{pol}_A : |A| \to \{-, +\}$

---

[22] In fact, it is the other way around: extensional resource terms were initially designed by the authors so that normal extensional resource terms are representations of augmentations on the universal arena.

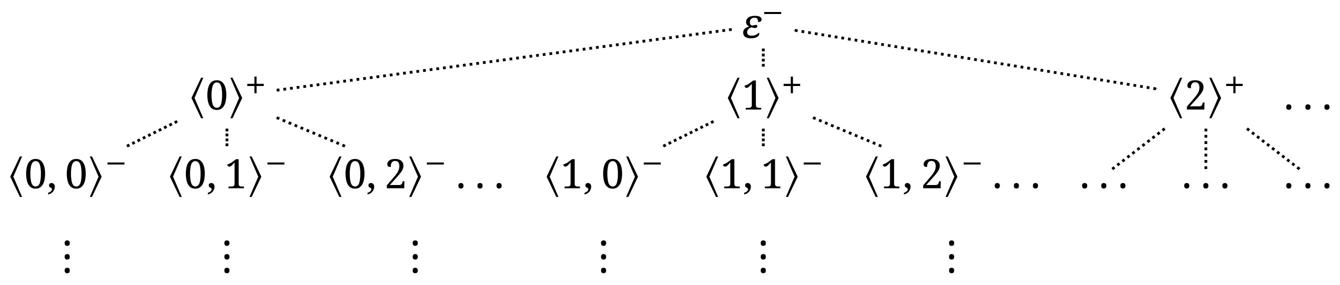

**Figure 14.** The universal arena **U**

is **alternating**: whenever $a_1 \rightarrowtriangle_A a_2$, we have $\mathrm{pol}_A(a_1) \neq \mathrm{pol}_A(a_2)$ (where $a_1 \rightarrowtriangle_A a_2$ means $a_1 <_A a_2$ with no event strictly between). A **negative arena** additionally satisfies: $\mathrm{pol}_A(a) = -$ for all $a \in \min(A) := \{a \in |A| \mid a \text{ minimal}\}$.

We say that a negative arena is **pointed** if it has a unique minimal event, called the **initial move**. An **isomorphism** of arenas, written $\varphi : A \cong B$, is a bijection on events preserving and reflecting all structure.

**Universal arena.** Usually, the arena represents the *type*. In this paper, as we are interested in an untyped language, we shall mainly work with one ambient arena **U**, that we call the *universal arena* (this is the maximal single-tree arena in the sense of Ker *et al.* [26]):

**DEFINITION 8.2.** The **universal arena**, denoted by **U**, has the following data:

*events*: the set $|\mathbf{U}| = \mathbb{N}^*$ of lists of natural numbers,
*causality*: the prefix ordering,
*polarity*: $\mathrm{pol}_{\mathbf{U}}(l) = -$ if $l$ has even length, + otherwise.

The universal arena is represented in Figure 14, read from top to bottom and where dotted lines represent immediate causal dependency. It is clear that **U** is a negative arena, with unique minimal event $\varepsilon$. We can think of events in this arena as moves exploring the structure of a closed Nakajima tree. Opponent starts computation with $\varepsilon^-$, exploring the root of the tree, prefixed by a countable sequence of abstractions $\lambda x_0. \cdots \lambda x_i. \cdots$. This Opponent move enables countably many Player moves $\langle i\rangle^+$, one for each $i \in \mathbb{N}$. Playing $\langle i\rangle^+$ corresponds to having an occurrence of $x_i$ as head variable, with countably many arguments. Opponent can access the $j$-th argument of this occurrence by playing $\langle i, j\rangle^-$. This process continues in this way, indefinitely. Note that the arena only describes the types of moves that can be played along this exploration process: a state of the process will be captured by the notion of *position* (an isomorphism class of *configurations*) and a run of the process itself will be abstracted as an *isogmentation* (an isomorphism class of *augmentations* [5]) – these concepts will be recalled below.

**Constructions.** We shall use some other constructions on arenas: the **atomic arena** $o$ has just one (negative) move $q$. The **tensor** $A_1 \otimes A_2$ of arenas $A_1$ and $A_2$ has as events the tagged disjoint union $|A_1 \otimes A_2| = |A_1| + |A_2| = \{1\} \times |A_1| \uplus \{2\} \times |A_2|$, and other components inherited. The **hom-arena** $A_1 \vdash A_2$ is $A_1^\perp \otimes A_2$, where the **dual** $A^\perp$ of $A$ has the same components but the polarity reversed. If $A_1$ is a negative arena and $A_2$ is pointed, the **arrow** $A_1 \Rightarrow A_2$, a pointed arena, has the same components as $A_1 \vdash A_2$ except that the events of $A_1$ are set to depend on the initial event of $A_2$. In the sequel, we shall also use the obvious generalization of the tensor with arbitrary (at most countable) arity, and write $A^{\mathbb{N}} = \bigotimes_{n \in \mathbb{N}} A$.

**Isomorphisms.** The following easy lemma states a few useful isomorphisms.

**LEMMA 8.3.** *For all arenas $\Gamma, A, B, C$ with $\Gamma$ and $A$ negative and $B$ pointed, we have isomorphisms:*

$$\begin{array}{rcrcll}
\mathsf{curry}_{\Gamma,A,B} & : & \Gamma \otimes A \Rightarrow B & \cong & \Gamma \Rightarrow (A \Rightarrow B) & : \ \mathsf{uncurry}_{\Gamma,A,B} \\
\mathsf{pack}_C & : & C \otimes C^{\mathbb{N}} & \cong & C^{\mathbb{N}} & : \ \mathsf{unpack}_C \\
\mathsf{unfold} & : & \mathbf{U} & \cong & \mathbf{U}^{\mathbb{N}} \Rightarrow o & : \ \mathsf{fold}
\end{array}$$

*supported by the bijections:*

$$\left.\begin{array}{r} \langle 1, \langle 1, g\rangle\rangle \\ \langle 1, \langle 2, a\rangle\rangle \\ \langle 2, b\rangle \end{array}\right\} \in |\Gamma \otimes A \Rightarrow B| \leftrightarrow |\Gamma \Rightarrow (A \Rightarrow B)| \ni \left\{\begin{array}{r} \langle 1, g\rangle \\ \langle 2, \langle 1, a\rangle\rangle \\ \langle 2, \langle 2, b\rangle\rangle \end{array}\right.$$

$$\left.\begin{array}{r} \langle 1, c\rangle \\ \langle 2, \langle i, c\rangle\rangle \end{array}\right\} \in |C \otimes C^{\mathbb{N}}| \leftrightarrow |C^{\mathbb{N}}| \ni \left\{\begin{array}{r} \langle 0, c\rangle \\ \langle i+1, c\rangle \end{array}\right. .$$

$$\left.\begin{array}{r} \varepsilon \\ i :: \vec{n} \end{array}\right\} \in |\mathbf{U}| \leftrightarrow |\mathbf{U}^{\mathbb{N}} \Rightarrow o| \ni \left\{\begin{array}{r} \langle 2, q\rangle \\ \langle 1, \langle i, \vec{n}\rangle\rangle \end{array}\right.$$

There is a category **ArIso** of arenas and isomorphisms between them, with full subcategories $\mathbf{ArIso}_{-}$, restricted to negative arenas, and $\mathbf{ArIso}_{\bullet}$, restricted to pointed arenas. The tensor and arrow constructions act functorially on isomorphisms of arenas, yielding functors

$$\begin{array}{rcrcl}
\otimes & : & \mathbf{ArIso} \times \mathbf{ArIso} & \to & \mathbf{ArIso} \\
\Rightarrow & : & \mathbf{ArIso}_{-} \times \mathbf{ArIso}_{\bullet} & \to & \mathbf{ArIso}_{\bullet}
\end{array}$$

obtained again in the obvious way, since the events of both $A \otimes B$ and $A \Rightarrow B$ are obtained as a disjoint union. Note $\Rightarrow$ is covariant in both components.

With this observation, Lemma 8.3 induces isomorphisms $\mathbf{U} \cong \mathbf{U}^{\mathbb{N}} \Rightarrow o \cong (\mathbf{U} \otimes \mathbf{U}^{\mathbb{N}}) \Rightarrow o \cong \mathbf{U} \Rightarrow (\mathbf{U}^{\mathbb{N}} \Rightarrow o) \cong \mathbf{U} \Rightarrow \mathbf{U}$, yielding the following:

**COROLLARY 8.4.** *We have an isomorphism of arenas:*

$$\mathsf{fun} \ : \ \mathbf{U} \ \cong \ \mathbf{U} \Rightarrow \mathbf{U} \ : \ \mathsf{unfun}$$

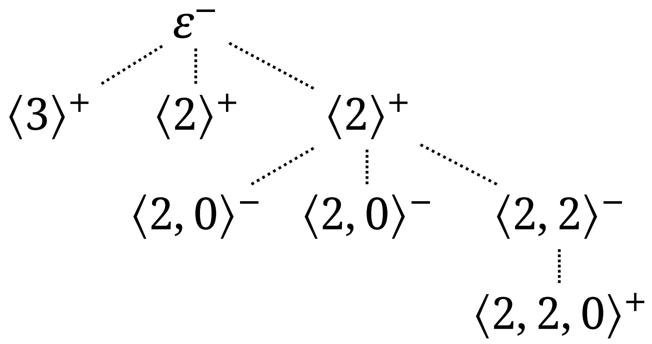

**Figure 15.** A configuration on **U**

*supported by the bijection*

$$\left.\begin{array}{r} \varepsilon \\ 0 :: \vec{n} \\ (i+1) :: \vec{n} \end{array}\right\} \in |\mathbf{U}| \leftrightarrow |\mathbf{U} \Rightarrow \mathbf{U}| \ni \left\{\begin{array}{r} \langle 2, \varepsilon\rangle \\ \langle 1, \vec{n}\rangle \\ \langle 2, i :: \vec{n}\rangle \end{array}\right. .$$

We shall establish a correspondence between closed normal extensional resource terms and *isogmentations* on **U**. But before defining those, we must first define an adequate notion of *state* on an arena.

## 8.2 Configurations and positions

Usually, in Hyland-Ong games, after defining arenas, one would go on to define *plays*, which are certain strings of moves of the arena (allowing replicated moves) additionally equipped with *pointers*, and satisfying a few further conditions. In pointer concurrent games, instead, our first step is to define *configurations*, which correspond to plays with pointers without the chronological ordering.

**Configurations.** We start with the definition:

**DEFINITION 8.5.** A **configuration** of the arena $A$, written $x \in \mathscr{C}(A)$, is $x = \langle |x|, \leq_x, \partial_x\rangle$ s.t. $\langle |x|, \leq_x\rangle$ is a finite forest, and the **display map** $\partial_x : |x| \to |A|$ is a function such that

*minimality-respecting:* for $a \in |x|$, $a$ is $\leq_x$-minimal iff $\partial_x(a)$ is $\leq_A$-minimal,
*causality-preserving:* for $a_1, a_2 \in |x|$, if $a_1 \rightarrowtriangle_x a_2$ then $\partial_x(a_1) \rightarrowtriangle_A \partial_x(a_2)$,

and $x$ is **pointed** (noted $x \in \mathscr{C}_\bullet(A)$) if it has exactly one minimal event $\mathsf{init}(x)$. We write $\varnothing$ for the empty configuration.

Configurations are a first step in capturing a notion of "thick subtree" [6] of an arena $A$: an exploration of the arena, keeping the same causal structure, but with the additional capacity to duplicate branches. As an example, we display in Figure 15 a configuration of the arena **U** introduced above. The figure shows the tree structure along with the display map, conveyed via the labelling of the nodes; and showcases the ability to replay the same move.

Though the definition of a configuration does not include a polarity function on $x$, one may be readily deduced by setting $\mathsf{pol}_x(a) = \mathsf{pol}_A(\partial_x(a))$ for $a \in x$. We write $a^-$ (respectively $a^+$) for $a$ such that $\mathsf{pol}_x(a) = -$ (respectively $\mathsf{pol}(a) = +$).

Let us give a few constructions on configurations. We start with the **tensor product**: for any arenas $A$ and $B$, and configurations $x \in \mathscr{C}(A)$ and $y \in \mathscr{C}(B)$, we define $x \otimes y \in \mathscr{C}(A \otimes B)$ by $|x \otimes y| = |x| + |y|$, the rest of the structure being inherited. This construction is easily generalized to the tensor $\otimes\vec{x} = \bigotimes_{i\in I} x_i$ of a finite family of configurations, with $|\bigotimes_{i\in I} x_i| = \sum_{i\in I} |x_i|$. The other constructions are variations on this theme. For $x \in \mathscr{C}(A)$ and $y \in \mathscr{C}(B)$, $x \vdash y \in \mathscr{C}(A \vdash B)$ is defined like $x \otimes y$. If $B$ and $y$ are pointed, then $x \Rightarrow y \in \mathscr{C}(A \Rightarrow B)$ is defined as $x \vdash y$ where, additionally, all events of $x$ are set to depend on the unique initial move of $y$.

If $x \in \mathscr{C}(A)$ and $y \in \mathscr{C}(A^{\mathbb{N}})$, we define $x :: y \in \mathscr{C}(A^{\mathbb{N}})$ from $x \otimes y \in \mathscr{C}(A \otimes A^{\mathbb{N}})$ by applying the isomorphism of arenas $\mathsf{pack}_A : A \otimes A^{\mathbb{N}} \simeq A^{\mathbb{N}}$: in other words, events and causality are the same, and $\partial_{x :: y} = \mathsf{pack}_A \circ \partial_{x\otimes y}$. Note that $x :: y = \emptyset$ iff $x$ and $y$ are both empty. By analogy with the range of streams, we also define the **range** of $x \in \mathscr{C}(A^{\mathbb{N}})$ as $\mathsf{r}(x) := \max\{i + 1 \in \mathbb{N} \mid \exists a \in |x|,\ \partial_x(a) \in \{i\} \times |A|\}$. By construction, $\mathsf{r}(\emptyset) = 0$ and $\mathsf{r}(x :: y) = \mathsf{r}(y) + 1$ if at least one of $x$ and $y$ is non-empty.

We will also consider **sums** of configurations. Given a finite family $\vec{x} = \langle x_i \rangle_{i\in I} \in \mathscr{C}(A)^I$, we define $\sum \vec{x} = \sum_{i\in I} x_i \in \mathscr{C}(A)$, as for the tensor $\otimes\vec{x}$, except for the display map: $\partial_{\sum \vec{x}}(j, a) = \partial_{x_j}(a)$. If moreover $\vec{x} \in \mathscr{C}_\bullet(A)^I$, we also write $[\![\vec{x}]\!]$, $[\![x_i \mid i \in I]\!]$ or $[\![x_i]\!]_{i\in I}$ for $\sum \vec{x} \in \mathscr{C}(A)$. Note that $\sum \varepsilon = \emptyset$ is the empty configuration. Moreover, if $\vec{x} = \langle x_i \rangle_{i\in I} \in \mathscr{C}_\bullet(A)^I$, then $[\![x_i]\!]_{i\in I} = \emptyset$ iff $\vec{x} = \varepsilon$.

**Symmetry and positions.** Configurations do not adequately capture thick subtrees by themselves, because they carry arbitrary "names" for events, which are irrelevant. Configurations should rather be considered up to *symmetry*: a **symmetry** $\varphi : x \cong_A y$ is an order-isomorphism such that $\partial_y \circ \varphi = \partial_x$. Symmetry classes of configurations are called **positions**: the set of positions on $A$ is written $\mathsf{Pos}(A)$, and they are ranged over by $\mathsf{x}, \mathsf{y}$, *etc.* A position $\mathsf{x}$ is **pointed**, written $\mathsf{x} \in \mathsf{Pos}_\bullet(A)$, if any of its representatives is. We also define the **range** of $\mathsf{x} \in \mathsf{Pos}(\mathbf{U}^{\mathbb{N}})$ as $\mathsf{r}(\mathsf{x}) := \mathsf{r}(x)$ for any representative $x$ of $\mathsf{x}$. It is easy to check that our constructions on configurations preserve symmetry so that, e.g., the sum $\sum \vec{\mathsf{x}} \in \mathsf{Pos}(A)$ of a finite family $\vec{\mathsf{x}} \in \mathsf{Pos}(A)^I$ is well defined.

If $x \in \mathscr{C}(A)$, we write $\overline{x} \in \mathsf{Pos}(A)$ for the corresponding position. Reciprocally, if $\mathsf{x} \in \mathsf{Pos}(A)$, we fix $\underline{\mathsf{x}} \in \mathscr{C}(A)$ a representative.

**LEMMA 8.6.** *For any arenas $A$, $A'$ and $B$ with $B$ pointed, the earlier constructions on configurations induce bijections:*

$$\begin{array}{rcrcl}
(- \otimes -) & : & \mathsf{Pos}(A) \times \mathsf{Pos}(A') & \simeq & \mathsf{Pos}(A \otimes A') \\
(- \vdash -) & : & \mathsf{Pos}(A) \times \mathsf{Pos}(A') & \simeq & \mathsf{Pos}(A \vdash A') \\
(- \Rightarrow -) & : & \mathsf{Pos}(A) \times \mathsf{Pos}_\bullet(B) & \simeq & \mathsf{Pos}_\bullet(A \Rightarrow B) \\
(- \parr -) & : & \mathsf{Pos}(A) \times \mathsf{Pos}(A^{\mathbb{N}}) & \simeq & \mathsf{Pos}(A^{\mathbb{N}}) \\
\llbracket - \rrbracket & : & \mathfrak{M}_{\mathrm{f}}(\mathsf{Pos}_\bullet(B)) & \simeq & \mathsf{Pos}(B)
\end{array}$$

**PROOF.** For the first four bijections, this is a straightforward verification that the corresponding constructions on configurations preserve and reflect symmetry, and are essentially surjective.

For $\llbracket - \rrbracket$, consider $\mu \in \mathfrak{M}_{\mathrm{f}}(\mathsf{Pos}_\bullet(B))$. Fix $\langle \mathsf{x}_i \rangle_{i \in I}$ an enumeration of $\mu$, and for each $i \in I$, fix $x_i \in \mathsf{x}_i$ a representative. Then, set

$$\llbracket \mu \rrbracket := \overline{\llbracket x_i \mid i \in I \rrbracket} \in \mathsf{Pos}(B)\,.$$

First, we show that this does not depend on the choice of the enumeration $\langle \mathsf{x}_i \rangle_{i \in I}$ nor of the representatives $\langle x_i \rangle_{i \in I}$: any other enumeration of $\mu$ is $\langle \mathsf{x}_{\pi(j)} \rangle_{j \in J}$ with $\pi : J \simeq I$, and then for any choice of representatives $\langle y_j \rangle_{j \in J}$ we have symmetries $\theta_j : x_{\pi(j)} \cong_B y_j$ for $j \in J$; then

$$\begin{array}{rccc}
\theta & : & \llbracket x_i \rrbracket_{i \in I} & \cong_B & \llbracket y_j \rrbracket_{j \in J} \\
& & (\pi(j), a) & \mapsto & \langle j, \theta_j(a) \rangle
\end{array}$$

is a valid symmetry. It follows that $\llbracket - \rrbracket : \mathfrak{M}_{\mathrm{f}}(\mathsf{Pos}_\bullet(B)) \to \mathsf{Pos}(B)$ is well-defined.

For injectivity, consider representatives $\langle x_i \rangle_{i \in I}$, $\langle y_j \rangle_{j \in J}$, and some symmetry $\theta : \llbracket x_i \mid i \in I \rrbracket \cong_B \llbracket y_j \mid j \in J \rrbracket$. As $\theta$ is an order-isomorphism between forests, it is determined by a bijection between minimal events, and a subsequent order-isomorphism for each minimal event. But since the $x_i$'s and $y_j$'s are pointed, minimal events of $\llbracket x_i \mid i \in I \rrbracket$ are in canonical bijection with $I$ – and likewise for $\llbracket y_j \mid j \in J \rrbracket$, so that $\theta$ is determined by some $\pi : J \simeq I$ and a family $\langle \theta_j : x_{\pi(j)} \cong_B y_j \rangle_{j \in J}$. But this is exactly what is needed to ensure that $\langle x_i \rangle_{i \in I}$ and $\langle y_j \rangle_{j \in J}$ represent the same element of $\mathfrak{M}_{\mathrm{f}}(\mathsf{Pos}_\bullet(B))$.

For surjectivity, consider $\mathsf{x} \in \mathsf{Pos}(B)$ represented by some $x \in \mathscr{C}(B)$. As a configuration, $x$ is a forest. If $I$ is the set of its minimal events and, for $i \in I$, $x_i$ is the corresponding tree with root $i$, then clearly $x \cong_B \llbracket x_i \mid i \in I \rrbracket$. ■

Note that, as a particular case, we have

$$(- \Rightarrow o) : \mathsf{Pos}(A) \times \mathsf{Pos}_\bullet(o) \simeq \mathsf{Pos}_\bullet(A \Rightarrow o)$$

and $\mathsf{Pos}_\bullet(o)$ is a singleton so that $\mathsf{Pos}(A) \times \mathsf{Pos}_\bullet(o) \simeq \mathsf{Pos}(A)$. Up to the isomorphism of arenas $\mathsf{fold} : \mathbf{U}^{\mathbb{N}} \Rightarrow o \cong \mathbf{U}$, and up to a minor abuse of notation, we obtain

$$(- \Rightarrow o) : \mathsf{Pos}(\mathbf{U}^{\mathbb{N}}) \simeq \mathsf{Pos}_\bullet(\mathbf{U})\,.$$

## 8.3 Positions of the universal arena

We use the above to show that the pointed positions of the universal arena **U** are in one-to-one correspondence with the elements of the particular relational model of the pure $\lambda$-calculus [8] that we studied in Section 7. We recall from our presentation in Section 7.2 that these elements are the value types $\vec{\alpha} \multimap o \in \mathcal{D}_{\mathrm{v}}$ where, inductively, $\vec{\alpha}$ ranges over the set of stream types $\mathcal{D}_{\mathrm{s}} = \mathcal{S}(\mathcal{D}_{\mathrm{v}})$; and a bag type is just a bag of values types $\bar{m} \in \mathcal{D}_{!} = \mathfrak{M}_{\mathrm{f}}(\mathcal{D}_{\mathrm{v}})$. Formally, we exhibit bijections

$$\kappa_{\mathrm{v}} : \mathcal{D}_{\mathrm{v}} \simeq \mathsf{Pos}_{\bullet}(\mathbf{U})\,, \quad \kappa_{!} : \mathcal{D}_{!} \simeq \mathsf{Pos}(\mathbf{U}) \quad \text{and} \quad \kappa_{\mathrm{s}} : \mathcal{D}_{\mathrm{s}} \simeq \mathsf{Pos}(\mathbf{U}^{\mathbb{N}})\,.$$

**DEFINITION 8.7.** We define the functions $\kappa_{\mathrm{v}}$, $\kappa_{!}$ and $\kappa_{\mathrm{s}}$ simultaneously by induction on type terms:

$$\kappa_{\mathrm{v}}(\vec{\alpha} \multimap o) := \kappa_{\mathrm{s}}(\vec{\alpha}) \Rightarrow o \qquad \kappa_{!}([\alpha_1, \dots, \alpha_k]) := \llbracket \kappa_{\mathrm{v}}(\alpha_1), \dots, \kappa_{\mathrm{v}}(\alpha_k) \rrbracket$$

$$\kappa_{\mathrm{s}}(\iota) := \overline{\varnothing} \qquad \kappa_{\mathrm{s}}(\bar{\alpha} :: \vec{\beta}) := \kappa_{!}(\bar{\alpha}) \mathbin{\overset{\circ\circ}{\circ\circ}} \kappa_{\mathrm{s}}(\vec{\beta}) \quad (\text{if } \bar{\alpha} :: \vec{\beta} \neq \iota).$$

By construction, $\kappa_{\mathrm{s}}(\bar{\alpha} :: \vec{\beta}) = \overline{\varnothing}$ iff $\bar{\alpha} :: \vec{\beta} = \iota$, so that $\kappa_{\mathrm{s}}(\bar{\alpha} :: \vec{\beta}) = \kappa_{!}(\bar{\alpha}) \mathbin{\overset{\circ\circ}{\circ\circ}} \kappa_{\mathrm{s}}(\vec{\beta})$ also in this case.

To show that these define bijections as stated above, we will reason on the *size* of positions: if $A$ is an arena and $\mathrm{x} \in \mathsf{Pos}(A)$, then the **size** of x, written #x, is the number of events in any of its representatives. In particular, $\#\mathrm{x} = 0$ iff x is the empty position. From the definitions of bijections in Lemma 8.6, it is simple to obtain the following identities:

$$\#(\mathrm{x} \Rightarrow o) = \#\mathrm{x} + 1 \qquad \#\llbracket \mathrm{x}_i \mid i \in I \rrbracket = \sum_{i \in I} \#\mathrm{x}_i \qquad \#(\mathrm{x} \mathbin{\overset{\circ\circ}{\circ\circ}} \mathrm{y}) = \#\mathrm{x} + \#\mathrm{y}$$

– indeed, it suffices to observe these identities on representatives.

**THEOREM 8.8.** *The functions*

$$\kappa_{\mathrm{v}} : \mathcal{D}_{\mathrm{v}} \to \mathsf{Pos}_{\bullet}(\mathbf{U}), \quad \kappa_{!} : \mathcal{D}_{!} \to \mathsf{Pos}(\mathbf{U}) \quad \textit{and} \quad \kappa_{\mathrm{s}} : \mathcal{D}_{\mathrm{s}} \to \mathsf{Pos}(\mathbf{U}^{\mathbb{N}})$$

*are bijections.*

**PROOF.** For injectivity, we reason by induction on type terms. If $\kappa_{\mathrm{v}}(\vec{\alpha} \multimap o) = \kappa_{\mathrm{v}}(\vec{\beta} \multimap o)$ then $\kappa_{\mathrm{s}}(\vec{\alpha}) \Rightarrow o = \kappa_{\mathrm{s}}(\vec{\beta}) \Rightarrow o$, hence $\kappa_{\mathrm{s}}(\vec{\alpha}) = \kappa_{\mathrm{s}}(\vec{\beta})$ by Lemma 8.6: by induction hypothesis we have $\vec{\alpha} = \vec{\beta}$ hence $\vec{\alpha} \multimap o = \vec{\beta} \multimap o$.

If $\kappa_{!}(\bar{\alpha}) = \kappa_{!}(\bar{\beta})$ with $\bar{\alpha} = [\alpha_1, \dots, \alpha_k]$ and $\bar{\beta} = [\beta_1, \dots, \beta_l]$, we obtain $\llbracket \kappa_{\mathrm{v}}(\alpha_1), \dots, \kappa_{\mathrm{v}}(\alpha_k) \rrbracket = \llbracket \kappa_{\mathrm{v}}(\beta_1), \dots, \kappa_{\mathrm{v}}(\beta_l) \rrbracket$. Lemma 8.6 ensures that $k = l$ and, up to reordering, $\kappa_{\mathrm{v}}(\alpha_i) = \kappa_{\mathrm{v}}(\beta_i)$ for $1 \leq i \leq k$: by induction hypothesis, we have $\alpha_i = \beta_i$ for $1 \leq i \leq k$, hence $\bar{\alpha} = \bar{\beta}$.

We have already observed that $\kappa_{\mathrm{s}}(\vec{\alpha}) = \overline{\varnothing}$ iff $\vec{\alpha} = \iota$. If $\mathrm{x} = \kappa_{\mathrm{s}}(\bar{\alpha} :: \vec{\beta}) = \kappa_{\mathrm{s}}(\bar{\alpha}' :: \vec{\beta}')$ with $\bar{\alpha} :: \vec{\beta} \neq \iota$, then $\mathrm{x} \neq \overline{\varnothing}$, hence $\bar{\alpha}' :: \vec{\beta}' \neq \iota$. We then write $\mathrm{x} = \kappa_{!}(\bar{\alpha}) \mathbin{\overset{\circ\circ}{\circ\circ}} \kappa_{\mathrm{v}}(\vec{\beta}) = \kappa_{!}(\bar{\alpha}') \mathbin{\overset{\circ\circ}{\circ\circ}} \kappa_{\mathrm{v}}(\vec{\beta}')$, and it is again sufficient to apply Lemma 8.6 and the induction hypothesis.

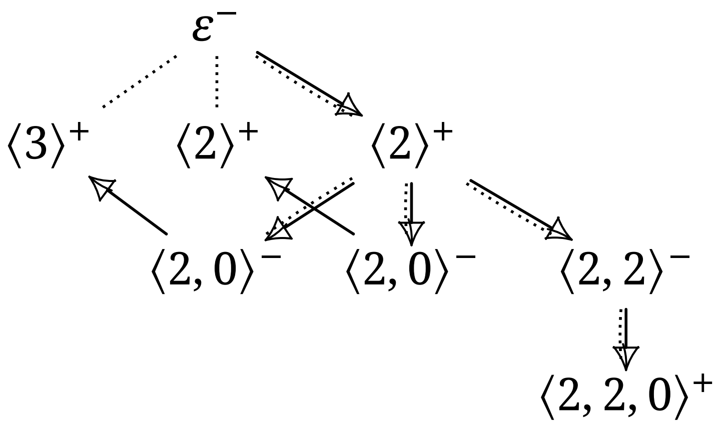

**Figure 16.** An augmentation on **U**

For surjectivity, we prove by induction on $d \in \mathbb{N}$ that, for each $\mathrm{x} \in \mathrm{Pos}_\bullet(\mathbf{U})$ (resp. $\mathrm{x} \in \mathrm{Pos}(\mathbf{U})$, $\mathrm{x} \in \mathrm{Pos}(\mathbf{U}^{\mathbb{N}})$) such that $\#\mathrm{x} \leq d$, there exists $\alpha \in \mathcal{D}_{\mathrm{v}}$ (resp. $\bar{\alpha} \in \mathcal{D}_!$, $\vec{\alpha} \in \mathcal{D}_{\mathrm{s}}$) such that $\kappa_{\mathrm{v}}(\alpha) = \mathrm{x}$ (resp. $\kappa_!(\bar{\alpha}) = \mathrm{x}$, $\kappa_{\mathrm{s}}(\vec{\alpha}) = \mathrm{x}$).

If $\mathrm{x} \in \mathrm{Pos}_\bullet(\mathbf{U})$ then, by Lemma 8.6, there exists $\mathrm{y} \in \mathrm{Pos}(\mathbf{U}^{\mathbb{N}})$ such that $\mathrm{x} = \mathrm{y} \Rightarrow o$. Then $\#\mathrm{y} < \#\mathrm{x}$ and by induction hypothesis there is $\vec{\alpha}$ such that $\mathrm{y} = \kappa_{\mathrm{s}}(\vec{\alpha})$: we obtain $\kappa_{\mathrm{v}}(\vec{\alpha} \multimap o) = \mathrm{y} \Rightarrow o = \mathrm{x}$.

If $\mathrm{x} \in \mathrm{Pos}(U)$, by Lemma 8.6, there exists $[\mathrm{x}_1, \dots, \mathrm{x}_k] \in \mathfrak{M}_{\mathrm{f}}(\mathrm{Pos}_\bullet(\mathbf{U}))$ such that $\mathrm{x} = [\![\mathrm{x}_1, \dots, \mathrm{x}_k]\!]$. Then $\#\mathrm{x}_i \leq \#\mathrm{x}$ and the previous paragraph gives $\alpha_i$ such that $\mathrm{x}_i = \kappa_{\mathrm{v}}(\alpha_i)$ for $1 \leq i \leq k$: we obtain $\mathrm{x} = \kappa_!([\alpha_1, \dots, \alpha_k])$.

If $\mathrm{x} \in \mathrm{Pos}(\mathbf{U}^{\mathbb{N}})$, we prove by a further induction on $\mathrm{r}(\mathrm{x})$ that there exists $\vec{\alpha} \in \mathcal{D}_{\mathrm{s}}$ such that $\kappa_{\mathrm{s}}(\vec{\alpha}) = \mathrm{x}$. We have $\vec{\varnothing} = \kappa_{\mathrm{s}}(\iota)$ by definition, so we can assume $\mathrm{x} \neq \vec{\varnothing}$. Lemma 8.6 yields $\mathrm{y} \in \mathrm{Pos}(\mathbf{U})$ and $\mathrm{z} \in \mathrm{Pos}(\mathbf{U}^{\mathbb{N}})$ such that $\mathrm{x} = \mathrm{y} \parr \mathrm{z}$. Then $\#\mathrm{y} \leq \#\mathrm{x}$, $\#\mathrm{z} \leq \#\mathrm{x}$, and, since one of y and z must be non-empty, $\mathrm{r}(\mathrm{z}) < \mathrm{r}(\mathrm{x})$: the induction hypothesis yields $\vec{\beta}$ such that $\mathrm{z} = \kappa_{\mathrm{s}}(\vec{\beta})$. Moreover, the previous paragraph gives $\bar{\alpha}$ such that $\mathrm{y} = \kappa_!(\bar{\alpha})$. We get $\mathrm{x} = \kappa_{\mathrm{s}}(\bar{\alpha} :: \vec{\beta})$. ■

## 8.4 Augmentations and isogmentations

### 8.4.1 Definitions

**Augmentations.** From the above, we have seen that *positions* are exactly points of the relational semantics (or intersection types), and that configurations are concrete representations for positions. An *augmentation* is then a configuration enriched with causal links: we will see that this extra structure amounts exactly to the specification a normal term of the corresponding type.

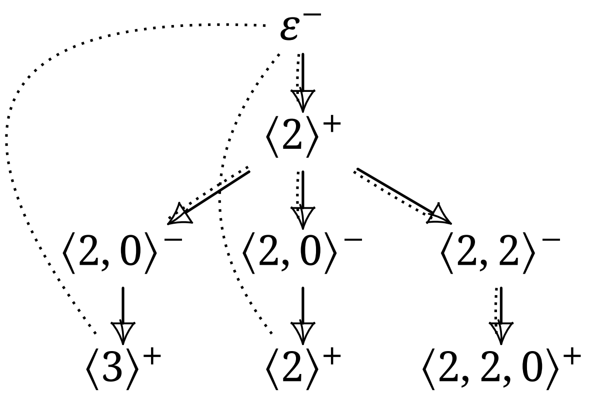

**Figure 17.** The same augmentation following the dynamic causality

**DEFINITION 8.9.** An **augmentation** on the arena $A$ is a tuple $\mathsf{q} = \langle |\mathsf{q}|, \leq_{\llparenthesis \mathsf{q} \rrparenthesis}, \leq_{\mathsf{q}}, \partial_{\mathsf{q}}\rangle$, where $\llparenthesis \mathsf{q} \rrparenthesis = \langle |\mathsf{q}|, \leq_{\llparenthesis \mathsf{q} \rrparenthesis}, \partial_{\mathsf{q}}\rangle \in \mathscr{C}(A)$, and $\langle |\mathsf{q}|, \leq_{\mathsf{q}}\rangle$ is a forest satisfying:

*rule-abiding:* if $a_1 \leq_{\llparenthesis \mathsf{q} \rrparenthesis} a_2$, then $a_1 \leq_{\mathsf{q}} a_2$,
*courteous:* if $a_1 \rightarrowtriangle_{\mathsf{q}} a_2$ with $\mathrm{pol}(a_1) = +$ or $\mathrm{pol}(a_2) = -$, then $a_1 \rightarrowtriangle_{\llparenthesis \mathsf{q} \rrparenthesis} a_2$,
*deterministic:* if $a^- \rightarrowtriangle_{\mathsf{q}} a_1^+$ and $a^- \rightarrowtriangle_{\mathsf{q}} a_2^+$, then $a_1 = a_2$,
*+-covered:* if $a \in |\mathsf{q}|$ is maximal in $\mathsf{q}$, then $\mathrm{pol}(a) = +$,
*negative:* if $a \in \min(\mathsf{q})$, then $\mathrm{pol}(a) = -$.

We then write $\mathsf{q} \in \mathrm{Aug}(A)$, and call $\llparenthesis \mathsf{q} \rrparenthesis \in \mathscr{C}(A)$ the **desequentialization** of $\mathsf{q}$.

Finally, $\mathsf{q}$ is **pointed** if it has a unique minimal event. In that case, we write $\mathsf{q} \in \mathrm{Aug}_\bullet(A)$.

We sometimes refer to $\leq_{\llparenthesis \mathsf{q} \rrparenthesis}$ as the **static** partial order, given that it corresponds to the causal constraints imported from the "type". Likewise, we often refer to $\leq_{\mathsf{q}}$ as the **dynamic** partial order, which will reflect the structure of a term. The **size** of an augmentation $\mathsf{q}$, written $\#\mathsf{q}$, is simply the cardinality of its set of events.

Moreover, for any negative arena $\Gamma$ and $\mathsf{q} \in \mathrm{Aug}(\Gamma \vdash \mathbf{U}^{\mathbb{N}})$, we define the **range** of $\mathsf{q}$ as $\mathrm{r}(\mathsf{q}) := \max\{i + 1 \in \mathbb{N} \mid \exists a \in |\mathsf{q}|,\ q \in |\mathbf{U}|,\ \partial_{\mathsf{q}}(a) = \langle 2, \langle i, q\rangle\rangle\}$. By *negative*, we have $\mathrm{r}(\mathbf{q}) = 0$ iff $\mathbf{q} = \overline{\varnothing}$.

Figure 16 shows a pointed augmentation, obtained by enriching the configuration in Figure 15 with an adequate dynamic causal order, presented here with triangle arrows $\rightarrowtriangle$ – following our drawing convention, we display the (Hasse diagram of the) static causality with dotted lines, and the (Hasse diagram of the) dynamic causality as $\rightarrowtriangle$. Figure 17 shows the same augmentation, but rearranged so as to emphasize the tree structure of $\rightarrowtriangle$ – it is this tree structure that we will show exactly corresponds to the tree structure of an extensional resource term; here

$$\lambda \vec{x}.\vec{x}(2)\ [\lambda \vec{y}.\vec{x}(3)\ \iota, \lambda \vec{y}.\vec{x}(2)\ \iota] :: [] :: [\lambda \vec{y}.\vec{y}(0)\ \iota] :: \iota\,, \tag{3}$$

and the reader may already match the tree structure of this term with Figure 17. However, as for configurations, augmentations are too rigid and carry explicit names for all events which should be quotiented out before we establish the link with extensional resource terms. This is done next.

**Isogmentations.** An **isomorphism** of augmentations $\varphi : \mathsf{q} \cong \mathsf{p}$ is a bijection $|\mathsf{q}| \cong |\mathsf{p}|$, preserving and reflecting all structure. An **isogmentation** is an isomorphism class of augmentations, ranged over by $\mathbf{q}, \mathbf{p}$, *etc.*: we write $\mathsf{Isog}(A)$ (resp. $\mathsf{Isog}_\bullet(A)$) for isogmentations (resp. *pointed* isogmentations). If $\mathsf{q} \in \mathsf{Aug}(A)$, we write $\overline{\mathsf{q}} \in \mathsf{Isog}(A)$ for its isomorphism class; reciprocally, if $\mathbf{q} \in \mathsf{Isog}(A)$, we fix a representative $\underline{\mathbf{q}} \in \mathbf{q}$.

Clearly, the size of an augmentation is invariant under isomorphism: the **size** of an isogmentation $\mathbf{q}$, written $\#\mathbf{q}$, is the size of any representative. Similarly, if $\mathbf{q} \in \mathsf{Aug}(\Gamma \vdash \mathbf{U}^{\mathbb{N}})$, we set the **range** of $\mathbf{q}$ to be the range of any of its representatives.

### 8.4.2 Constructions on Isogmentations

Next, we provide a few constructions on augmentations and isogmentations.

**Tupling.** Consider negative arenas $\Gamma, A_1, \dots, A_k$, and $\mathsf{q}_i \in \mathsf{Aug}(\Gamma \vdash A_i)$ for $1 \leq i \leq k$. We set $\mathsf{q} = \langle\!\langle \mathsf{q}_i \mid 1 \leq i \leq k \rangle\!\rangle \in \mathsf{Aug}(\Gamma \vdash \otimes_{1 \leq i \leq k} A_i)$ with

$$|\mathsf{q}| = \sum_{i=1}^{k} |\mathsf{q}_i|\,, \qquad \left\{ \begin{array}{rcll} \partial_{\mathsf{q}}(i,m) & = & \langle 1, g\rangle & \text{if } \partial_{\mathsf{q}_i}(m) = \langle 1, g\rangle, \\ \partial_{\mathsf{q}}(i,m) & = & \langle 2, \langle i, a\rangle\rangle & \text{if } \partial_{\mathsf{q}_i}(m) = \langle 2, a\rangle, \end{array} \right.$$

with the two orders $\leq_{\mathsf{q}}$ and $\leq_{(\!|\mathsf{q}|\!)}$ inherited.

**PROPOSITION 8.10.** *For any negative arenas $\Gamma, A_1, \dots, A_k$, this yields a bijection*

$$\langle\!\langle -, \dots, - \rangle\!\rangle : \prod_{i=1}^{k} \mathsf{Isog}(\Gamma \vdash A_i) \simeq \mathsf{Isog}(\Gamma \vdash \otimes_{1 \leq i \leq k} A_i)\,.$$

**PROOF.** First, it is direct that this construction preserves symmetry so that it extends to isogmentations. We show that it is a bijection.

*Injective.* As an iso must preserve $\rightarrowtriangle$ and display maps, any $\varphi : \langle\!\langle \mathsf{q}_i \mid i \in I \rangle\!\rangle \cong \langle\!\langle \mathsf{p}_i \mid i \in I \rangle\!\rangle$ decomposes uniquely into a sequence of $\varphi_i : \mathsf{q}_i \cong \mathsf{p}_i$, as required.

*Surjective.* Consider $\mathsf{q} \in \mathsf{Aug}(\Gamma \vdash \otimes_{1 \leq i \leq k} A_i)$. Since $\mathsf{q}$ is a forest, any $m \in |\mathsf{q}|$ has a unique minimal antecedent, called $\mathsf{init}(m)$, sent by the display map (via *negative*) to one of the $A_i$s — we say that $m$ is *above* $A_i$. This allows us to assign every move of $\mathsf{q}$ to exactly one of the $A_i$s, so that moves connected by (the transitive symmetric closure of) $\leq_{\mathsf{q}}$ are in the same component. This determines a partition of $\mathsf{q}$ into a family of $\mathsf{q}_i \in \mathsf{Aug}(\Gamma \vdash A_i)$, one for each component – and it is then a straightforward verification that $\mathsf{q} \cong \langle\!\langle \mathsf{q}_i \mid 1 \leq i \leq k \rangle\!\rangle$. ■

In particular, we obtain a bijection

$$\begin{aligned} \mathsf{Isog}(\Gamma \vdash \mathbf{U}) \times \mathsf{Isog}(\Gamma \vdash \mathbf{U}^{\mathbb{N}}) &\simeq \mathsf{Isog}(\Gamma \vdash \mathbf{U}^{\mathbb{N}}) \\ \langle \mathbf{p}, \mathbf{q} \rangle &\mapsto \mathbf{p} \mathbin{\substack{\circ\circ\\ \circ\circ}} \mathbf{q} \end{aligned}$$

by setting $\mathbf{p} \mathbin{\substack{\circ\circ\\ \circ\circ}} \mathbf{q} := \mathsf{pack}_\Gamma(\langle\!\langle \mathbf{p}, \mathbf{q} \rangle\!\rangle)$ where

$$\mathsf{pack}_\Gamma : \mathsf{Isog}(\Gamma \vdash \mathbf{U} \otimes \mathbf{U}^{\mathbb{N}}) \simeq \mathsf{Isog}(\Gamma \vdash \mathbf{U}^{\mathbb{N}})$$

is induced by the isomorphism of arenas $\mathsf{pack}_{\mathbf{U}} : \mathbf{U} \otimes \mathbf{U}^{\mathbb{N}} \simeq \mathbf{U}^{\mathbb{N}}$.

**Bags.** Consider negative arenas $\Gamma$ and $A$, and $\mathsf{q}_1, \mathsf{q}_2 \in \mathsf{Aug}(\Gamma \vdash A)$. We set $\mathsf{q}_1 * \mathsf{q}_2 \in \mathsf{Aug}(\Gamma \vdash A)$ with events $|\mathsf{q}_1 * \mathsf{q}_2| = |\mathsf{q}_1| + |\mathsf{q}_2|$, and display $\partial_{\mathsf{q}_1 * \mathsf{q}_2}(i, m) = \partial_{\mathsf{q}_i}(m)$, and the two orders $\leq_{\mathsf{q}_1 * \mathsf{q}_2}$ and $\leq_{(\!|\mathsf{q}_1 * \mathsf{q}_2|\!)}$ inherited. This generalizes to an $n$-ary operation $\Pi_{i \in I} \mathsf{q}_i$ in the obvious way, which preserves isomorphisms. The operation induced on isogmentations is associative and admits as neutral element the empty isogmentation $\overline{\varnothing} \in \mathsf{Isog}(\Gamma \vdash A)$ (with a unique representative $\varnothing$) with no event. If $\bar{\mathbf{q}} = [\mathbf{q}_1, \dots, \mathbf{q}_k] \in \mathfrak{M}_{\mathrm{f}}(\mathsf{Isog}_\bullet(\Gamma \vdash A))$, then we write $[\![\mathbf{q}_1, \dots, \mathbf{q}_k]\!] := \Pi \bar{\mathbf{q}}$.

**PROPOSITION 8.11.** *For any negative arenas* $\Gamma, A$, *this yields a bijection*

$$\begin{aligned} \mathfrak{M}_{\mathrm{f}}(\mathsf{Isog}_\bullet(\Gamma \vdash A)) &\simeq \mathsf{Isog}(\Gamma \vdash A) \\ [\mathbf{q}_1, \dots, \mathbf{q}_k] &\mapsto [\![\mathbf{q}_1, \dots, \mathbf{q}_k]\!] \,. \end{aligned}$$

**PROOF.** *Injective.* Assuming the $\mathsf{q}_i$'s and $\mathsf{p}_i$'s are pointed, an iso $\varphi : \Pi_{1 \leq i \leq k} \mathsf{q}_i \cong \Pi_{1 \leq j \leq l} \mathsf{p}_j$ forces $k = l$ and induces a permutation $\pi$ on $k$ and a family of isos $(\varphi_i : \mathsf{q}_i \simeq \mathsf{p}_{\pi(i)})_{1 \leq i \leq k}$, which implies $[\overline{\mathsf{q}_i} \mid 1 \leq i \leq k] = [\overline{\mathsf{p}_i} \mid 1 \leq i \leq k]$ as bags.

*Surjective.* As any $\mathsf{q} \in \mathsf{Aug}(\Gamma \vdash A)$ is finite, it has a finite set $I$ of initial moves. As $\mathsf{q}$ is a forest, any $m \in |\mathsf{q}|$ is above exactly one initial move. For $i \in I$, write $\mathsf{q}_i \in \mathsf{Aug}_\bullet(\Gamma \vdash A)$ the restriction of $\mathsf{q}$ above $i$; then $\mathsf{q} \cong \Pi_{i \in I} \mathsf{q}_i$. ■

**Currying.** For negative arenas $\Gamma$, $A$ and $B$, with $B$ pointed, we have

$$\Lambda_{\Gamma, A, B} : \mathsf{Aug}(\Gamma \otimes A \vdash B) \simeq \mathsf{Aug}(\Gamma \vdash A \Rightarrow B)$$

a bijection compatible with isos, which leaves the core of the augmentation unchanged and only reassigns the display map, following the associativity up to isomorphism of the tagged disjoint union $(X + Y) + Z \simeq X + (Y + Z)$. Additionally, this bijection preserves symmetry, and hence extends to isogmentations. In linking resource terms and isogmentation, we shall only use the instance:

$$\Lambda_{V, \vec{x}} : \mathsf{Isog}_\bullet((\mathbf{U}^{\mathbb{N}})^{V, \vec{x}} \vdash o) \simeq \mathsf{Isog}_\bullet((\mathbf{U}^{\mathbb{N}})^{V} \vdash \mathbf{U}^{\mathbb{N}} \Rightarrow o)$$

where $V$ is a finite set of sequence variables and $\vec{x} \notin V$ is a sequence variable.

**Head occurrence.** We are missing one last bijection, corresponding to normal base terms. We start with the construction on augmentations. Fix for now a negative arena $\Gamma = (\mathbf{U}^{\mathbb{N}})^V$ for $V$ a finite set of sequence variables. Consider also $\mathsf{q} \in \mathsf{Aug}(\Gamma \vdash \mathbf{U}^{\mathbb{N}})$ an augmentation, a sequence variable $\vec{x} \in V$, and a rank $i \in \mathbb{N}$. With this, we form

$$\Box_{\vec{x},i}(\mathsf{q}) \in \mathsf{Aug}_\bullet(\Gamma \vdash o)$$

the $(\vec{x}, i)$**-lifting** of $\mathsf{q}$: it calls the value variable $\vec{x}(i)$, giving it $\mathsf{q}$ as a stream of arguments, mimicking a base term $\vec{x}(i)\,\mathsf{q}$. Concretely, $\Box_{\vec{x},i}(\vec{\mathsf{q}})$ starts by playing the initial move in the component $\langle \vec{x}, i\rangle$ of $\Gamma$, then proceeds as $\mathsf{q}$. More precisely:

**DEFINITION 8.12.** Consider $\Gamma = (\mathbf{U}^{\mathbb{N}})^V$ for $V$ a finite set of sequence variables, $\mathsf{q} \in \mathsf{Aug}(\Gamma \vdash \mathbf{U}^{\mathbb{N}})$ an augmentation, $\vec{x} \in V$, and $i \in \mathbb{N}$. The $(\vec{x}, i)$**-lifting** of $\mathsf{q}$, written $\Box_{\vec{x},i}(\mathsf{q}) \in \mathsf{Aug}_\bullet(\Gamma \vdash o)$, has dynamic partial order $\leq_{\mathsf{q}}$ prefixed with two additional moves, *i.e.* $\ominus \rightarrowtriangle \oplus \rightarrowtriangle \mathsf{q}$. Its static causality is the least partial order comprising the dependencies of the form

$$\begin{array}{lllll} m & \leq_{(\!|\Box_{\vec{x},i}(\mathsf{q})|\!)} & n & & \text{for } m, n \in |\mathsf{q}| \text{ with } m \leq_{(\!|\mathsf{q}|\!)} n, \\ \oplus & \leq_{(\!|\Box_{\vec{x},i}(\mathsf{q})|\!)} & m & & \text{for all } m \in |\mathsf{q}| \text{ with } \partial_{\mathsf{q}}(m) = \langle 2, -\rangle, \end{array}$$

and with display map given by the following clauses:

$$\begin{array}{lcll} \partial_{\Box_{\vec{x},i}(\mathsf{q})}(\ominus) & = & \langle 2, q\rangle & \\ \partial_{\Box_{\vec{x},i}(\mathsf{q})}(\oplus) & = & \langle 1, \langle \vec{x}, \langle i, \varepsilon\rangle\rangle\rangle & \\ \partial_{\Box_{\vec{x},i}(\mathsf{q})}(m) & = & \langle 1, a\rangle & \text{if } \partial_{\mathsf{q}}(m) = \langle 1, a\rangle, \\ \partial_{\Box_{\vec{x},i}(\mathsf{q})}(m) & = & \langle 1, \langle \vec{x}, \langle i, k :: l\rangle\rangle\rangle & \text{if } \partial_{\mathsf{q}}(m) = \langle 2, \langle k, l\rangle\rangle, \end{array}$$

altogether defining $\Box_{\vec{x},i}(\mathsf{q}) \in \mathsf{Aug}_\bullet(\Gamma \vdash o)$ as required.

It is a routine exercise to check that this yields a well-defined augmentation and that it preserves isomorphisms, hence extends to isogmentations, to give:

**PROPOSITION 8.13.** *Let* $\Gamma = (\mathbf{U}^{\mathbb{N}})^V$ *for* $V$ *a finite set of sequence variables. We obtain a bijection*

$$\begin{array}{rcl} \Box : V \times \mathbb{N} \times \mathsf{Isog}(\Gamma \vdash \mathbf{U}^{\mathbb{N}}) & \simeq & \mathsf{Isog}_\bullet(\Gamma \vdash o) \\ \langle \vec{x}, i, \mathbf{q}\rangle & \mapsto & \Box_{\vec{x},i}(\mathbf{q})\,. \end{array}$$

**PROOF.** *Injective.* Assume $\varphi : \Box_{\vec{x},i}\mathsf{q} \cong \Box_{\vec{y},j}\mathsf{p}$ an iso. From the commutation with the display map, the first Player moves must be in the same component, hence $\vec{x} = \vec{y}$ and $i = j$. Removing the first two events, $\varphi$ restricts to $\psi : \mathsf{q} \cong \mathsf{p}$.

*Surjective.* Any $\mathsf{q} \in \mathsf{Aug}_\bullet(\Gamma \vdash o)$ has a unique initial move (write it $\ominus$), which cannot be maximal by $+$-*covered*. By *determinism*, there is a unique subsequent Player move (write it $\oplus$), displayed to the initial move of some occurrence of $\mathbf{U}$ in $\Gamma = (\mathbf{U}^{\mathbb{N}})^V$, determining some $\vec{x} \in V$ and $i \in \mathbb{N}$. We obtain $\mathsf{p} \in \mathsf{Aug}(\Gamma \vdash \mathbf{U}^{\mathbb{N}})$ by removing $\ominus$ and $\oplus$, and redisplaying on the right the events statically depending on $\oplus$, reversing the reassignment of Definition 8.12. We obtain $\mathsf{p} \in \mathsf{Aug}(\Gamma \vdash \mathbf{U}^{\mathbb{N}})$ and $\mathsf{q} \cong \Box_{\vec{x},i}\mathsf{p}$ by a routine inspection of the definitions. ■

## 8.5 Isogmentations as normal resource terms

We spell out the bijection between normal resource terms and isogmentations. If $\Gamma$ is a finite set of sequence variables, write $\Delta^{\mathrm{n}}_{\mathrm{v}}(\Gamma)$ (*resp.* $\Delta^{\mathrm{n}}_{\mathrm{b}}(\Gamma), \Delta^{\mathrm{n}}_{!}(\Gamma), \Delta^{\mathrm{n}}_{\mathrm{s}}(\Gamma)$) for the *normal* value terms $m \in \Delta_{\mathrm{v}}$ s.t. $\mathcal{V}_{\mathrm{s}}(m) \subseteq \Gamma$ (*resp.* likewise for base, bag, and stream terms). A finite set of sequence variables is interpreted as an arena:

$$\|\Gamma\| := (\mathbf{U}^{\mathbb{N}})^{\Gamma}\,.$$

We shall in fact establish *four* mutually inductive bijections for value, base, bag and stream normal extensional resource terms, presenting them as isogmentations on appropriate arenas.

**DEFINITION 8.14.** We define

$$\begin{aligned}
\|-\|^{\Gamma}_{\mathrm{b}} &: \Delta^{\mathrm{n}}_{\mathrm{b}}(\Gamma) \longrightarrow \mathsf{Isog}_{\bullet}(\|\Gamma\| \vdash o)\\
\|-\|^{\Gamma}_{\mathrm{v}} &: \Delta^{\mathrm{n}}_{\mathrm{v}}(\Gamma) \longrightarrow \mathsf{Isog}_{\bullet}(\|\Gamma\| \vdash \mathbf{U})\\
\|-\|^{\Gamma}_{!} &: \Delta^{\mathrm{n}}_{!}(\Gamma) \longrightarrow \mathsf{Isog}(\|\Gamma\| \vdash \mathbf{U})\\
\|-\|^{\Gamma}_{\mathrm{s}} &: \Delta^{\mathrm{n}}_{\mathrm{s}}(\Gamma) \longrightarrow \mathsf{Isog}(\|\Gamma\| \vdash \mathbf{U}^{\mathbb{N}})
\end{aligned}$$

by mutual induction as follows:

$$\begin{aligned}
\|\vec{x}(i)\,\vec{n}\|^{\Gamma}_{\mathrm{b}} &:= \Box_{\vec{x},i}(\|\vec{n}\|^{\Gamma}_{\mathrm{s}})\\
\|\lambda\vec{x}.a\|^{\Gamma}_{\mathrm{v}} &:= \Lambda_{\|\Gamma\|,\vec{x}}(\|a\|^{\Gamma,\vec{x}}_{\mathrm{b}})\\
\|[m_1,\ldots,m_k]\|^{\Gamma}_{!} &:= [\![\|m_1\|^{\Gamma}_{\mathrm{v}},\ldots,\|m_k\|^{\Gamma}_{\mathrm{v}}]\!]\\
\|\iota\|^{\Gamma}_{\mathrm{s}} &:= \overline{\varnothing}\\
\|\bar{m} :: \vec{n}\|^{\Gamma}_{\mathrm{s}} &:= \|\bar{m}\|^{\Gamma}_{\mathrm{b}} \mathbin{\overset{\circ\circ}{\circ\circ}} \|\vec{n}\|^{\Gamma}_{\mathrm{s}} \quad \text{if } \bar{m} :: \vec{n} \neq \iota.
\end{aligned}$$

Note that $\|\vec{m}\|^{\Gamma}_{\mathrm{s}} = \iota$ iff $\vec{m} = \iota$, and that the identity $\|\bar{m} :: \vec{n}\|^{\Gamma}_{\mathrm{s}} = \|\bar{m}\|^{\Gamma}_{\mathrm{b}} \mathbin{\overset{\circ\circ}{\circ\circ}} \|\vec{n}\|^{\Gamma}_{\mathrm{s}}$ always holds.

Unlike in the typed setting [5], the proof that these define bijections cannot be done by induction on types. As we did in the proof of Theorem 8.8, we shall reason by induction on terms for injectivity, and by induction on the size and range of isogmentations for surjectivity. Note that $\#\mathbf{q} = 0$ iff $\mathbf{q}$ is empty. By a straightforward inspection of the definitions, we moreover obtain:

$$\begin{aligned}
\#\Box_{\vec{x},i}(\mathbf{p}) &= \#\mathbf{p} + 2 & \qquad \#\Lambda_{\Gamma,\vec{x}}(\mathbf{p}) &= \#\mathbf{p}\\
\#[\![\mathbf{q}_i \mid i \in I]\!] &= \sum_{i\in I} \#\mathbf{q}_i & \qquad \#(\mathbf{q} \mathbin{\overset{\circ\circ}{\circ\circ}} \mathbf{p}) &= \#\mathbf{q} + \#\mathbf{p}
\end{aligned}$$

and $\mathsf{r}(\mathbf{p} \mathbin{\overset{\circ\circ}{\circ\circ}} \mathbf{q}) = \mathsf{r}(\mathbf{q}) + 1$ if at least one of $\mathbf{p}$ and $\mathbf{q}$ is non-empty.

**THEOREM 8.15.** *Consider* $\Gamma$ *any finite set of sequence variables. Then* $\|-\|^{\Gamma}_{\mathsf{b}}$, $\|-\|^{\Gamma}_{\mathsf{v}}$, $\|-\|^{\Gamma}_{!}$, *and* $\|-\|^{\Gamma}_{\mathsf{s}}$ *are bijections.*

**PROOF.** For injectivity, we reason by induction on terms.

If $\|\vec{x}(i)\,\vec{n}\|^{\Gamma}_{\mathsf{b}} = \|\vec{y}(j)\,\vec{p}\|^{\Gamma}_{\mathsf{b}}$ then $\vec{x} = \vec{y}$, $i = j$ and $\|\vec{n}\|^{\Gamma}_{\mathsf{s}} = \|\vec{p}\|^{\Gamma}_{\mathsf{s}}$, by the injectivity of $\square$. We moreover obtain $\vec{n} = \vec{p}$ by induction hypothesis.

If $\|\lambda\vec{x}.a\|^{\Gamma}_{\mathsf{v}} = \|\lambda\vec{y}.b\|^{\Gamma}_{\mathsf{v}}$, we can always assume $\vec{x} = \vec{y}$ by $\alpha$-equivalence, hence $\|a\|^{\Gamma,\vec{x}}_{\mathsf{b}} = \|b\|^{\Gamma,\vec{x}}_{\mathsf{b}}$ by the injectivity of $\Lambda_{\Gamma,\vec{x}}$. Then we obtain $a = b$ by induction hypothesis.

If $\|[m_1,\ldots,m_k]\|^{\Gamma}_{!} = \|[n_1,\ldots,n_l]\|^{\Gamma}_{!}$, then by the injectivity of $[\![-]\!]$, we have $[\|m_1\|^{\Gamma}_{!},\ldots,\|m_k\|^{\Gamma}_{!}] = [\|n_1\|^{\Gamma}_{!},\ldots,\|n_l\|^{\Gamma}_{!}]$, hence $k = l$ and, up to reordering the $n_i$'s, $\|m_i\|^{\Gamma}_{!} = \|n_i\|^{\Gamma}_{!}$ for $1 \leq i \leq k$. The induction hypothesis yields $m_i = n_i$ for $1 \leq i \leq k$.

We have already observed that $\|\vec{m}\|^{\Gamma}_{\mathsf{s}} = \iota$ iff $\vec{m} = \iota$. If $\mathbf{q} = \|\bar{m} :: \vec{n}\|^{\Gamma}_{\mathsf{s}} = \|\bar{m}' :: \vec{n}'\|^{\Gamma}_{\mathsf{s}}$ with $\bar{m} :: \vec{n} \neq \iota$, then $\mathbf{q} \neq \overline{\varnothing}$, hence $\bar{m}' :: \vec{n}' \neq \iota$. The injectivity of $\mathbin{\substack{\circ\circ\\\circ\circ}}$ then yields $\|\bar{m}\|^{\Gamma}_{!} = \|\bar{m}'\|^{\Gamma}_{!}$ and $\|\vec{n}\|^{\Gamma}_{\mathsf{s}} = \|\vec{n}'\|^{\Gamma}_{\mathsf{s}}$.

For surjectivity, we prove by induction on $d \in \mathbb{N}$ that, for each $\mathbf{q} \in \mathrm{Isog}_{\bullet}(\|\Gamma\| \vdash o)$ (resp. $\mathbf{q} \in \mathrm{Isog}_{\bullet}(\|\Gamma\| \vdash \mathbf{U})$, $\mathbf{q} \in \mathrm{Isog}(\|\Gamma\| \vdash \mathbf{U})$, or $\mathbf{q} \in \mathrm{Isog}(\|\Gamma\| \vdash \mathbf{U}^{\mathbb{N}})$) such that $\#\mathbf{q} \leq d$, there exists a term $a \in \Delta^{\mathsf{n}}_{\mathsf{b}}(\Gamma)$ (resp. $m \in \Delta^{\mathsf{n}}_{\mathsf{v}}(\Gamma)$, $\bar{m} \in \Delta^{\mathsf{n}}_{!}(\Gamma)$, or $\vec{m} \in \Delta^{\mathsf{n}}_{\mathsf{s}}(\Gamma)$) such that $\|a\|^{\Gamma}_{\mathsf{b}} = \mathbf{q}$ (resp. $\|m\|^{\Gamma}_{\mathsf{v}} = \mathbf{q}$, $\|\bar{m}\|^{\Gamma}_{!} = \mathbf{q}$, or $\|\vec{m}\|^{\Gamma}_{\mathsf{s}} = \mathbf{q}$).

For each $\mathbf{q} \in \mathrm{Isog}_{\bullet}(\|\Gamma\| \vdash o)$, the surjectivity of $\square$ yields $\vec{x} \in \Gamma$, $i \in \mathbb{N}$ and $\mathbf{p} \in \mathrm{Isog}(\|\Gamma\| \vdash \mathbf{U}^{\mathbb{N}})$ such that $\mathbf{q} = \square_{\vec{x},i}(\mathbf{p})$. In particular, $\#\mathbf{p} < \#\mathbf{q}$, and the induction hypothesis gives $\vec{m}$ with $\|\vec{m}\|^{\Gamma}_{\mathsf{s}} = \mathbf{p}$. We obtain $\mathbf{q} = \|\vec{x}(i)\,\vec{m}\|^{\Gamma}_{\mathsf{b}}$.

For each $\mathbf{q} \in \mathrm{Isog}_{\bullet}(\|\Gamma\| \vdash \mathbf{U})$ and sequence variable $\vec{x} \notin \Gamma$, the surjectivity of $\Lambda_{\Gamma,\vec{x}}$ yields $\mathbf{p} \in \mathrm{Isog}_{\bullet}(\|\Gamma,\vec{x}\| \vdash o)$ such that $\mathbf{q} = \Lambda_{\Gamma,\vec{x}}(\mathbf{p})$. In particular, $\#\mathbf{p} \leq \#\mathbf{q}$, and the previous paragraph gives $a$ with $\|a\|^{\Gamma,\vec{x}}_{\mathsf{b}} = \mathbf{p}$. We obtain $\mathbf{q} = \|\lambda\vec{x}.a\|^{\Gamma}_{\mathsf{v}}$.

For each $\mathbf{q} \in \mathrm{Isog}(\|\Gamma\| \vdash \mathbf{U})$, the surjectivity of $[\![-]\!]$ yields $\mathbf{p}_1,\ldots,\mathbf{p}_k \in \mathrm{Isog}_{\bullet}(\|\Gamma\| \vdash \mathbf{U})$ such that $\mathbf{q} = [\![\mathbf{p}_1,\ldots,\mathbf{p}_k]\!]$. In particular, $\#\mathbf{p}_i \leq \#\mathbf{q}$, and the previous paragraph gives $m_i$ with $\|m_i\|^{\Gamma}_{\mathsf{v}} = \mathbf{p}_i$. We obtain $\mathbf{q} = \|[m_1,\ldots,m_k]\|^{\Gamma}_{!}$.

Finally, if $\mathbf{q} \in \mathrm{Isog}(\|\Gamma\| \vdash \mathbf{U}^{\mathbb{N}})$ we prove by a further induction on $\mathsf{r}(\mathbf{q})$ that there exists $\vec{m} \in \Delta^{\mathsf{n}}_{\mathsf{s}}(\Gamma)$ such that $\|\vec{m}\|^{\Gamma}_{\mathsf{s}} = \mathbf{q}$. Since $\overline{\varnothing} = \|\iota\|^{\Gamma}_{\mathsf{s}}$ we can assume that $\mathbf{q} \neq \overline{\varnothing}$. By the surjectivity of $\mathbin{\substack{\circ\circ\\\circ\circ}}$, we can write $\mathbf{q} = \mathbf{q}_1 \mathbin{\substack{\circ\circ\\\circ\circ}} \mathbf{q}_2$ with $\mathbf{q}_1 \in \mathrm{Isog}(\|\Gamma\| \vdash \mathbf{U})$ and $\mathbf{q}_2 \in \mathrm{Isog}(\|\Gamma\| \vdash \mathbf{U}^{\mathbb{N}})$. Then $\#\mathbf{q}_1 \leq \#\mathbf{q}$, $\#\mathbf{q}_2 \leq \#\mathbf{q}$, and $\mathsf{r}(\mathbf{q}_2) < \mathsf{r}(\mathbf{q})$ since $\mathbf{q}$ is not empty. The previous paragraph gives $\bar{m}_1 \in \Delta^{\mathsf{n}}_{!}(\Gamma)$ such that $\|\bar{m}_1\|^{\Gamma}_{!} = \mathbf{q}_1$, and the induction hypothesis gives $\vec{m}_2 \in \Delta^{\mathsf{n}}_{\mathsf{s}}(\Gamma)$ such that $\|\vec{m}_2\|^{\Gamma}_{\mathsf{s}} = \mathbf{q}_2$. We obtain $\mathbf{q} = \|\bar{m}_1 :: \vec{m}_2\|^{\Gamma}_{\mathsf{s}}$. ■

As claimed, normal extensional resource terms are a syntax for isogmentations on the universal arena. The reader may apply the implicit algorithm on the augmentation of Figure 17, and observe that we indeed get the term (3).

## 8.6 Compatibility with the bijection on positions

Each isogmentation yields (forgetting the dynamic causal links) a position. In Section 8.3, we established a bijection between positions of the universal arena and the relational types of Section 7. In Section 7.2, we proved that each normal resource term admits *exactly one* typing derivation. In this final technical section, we tie everything together by proving that these correspondences are compatible – that is, that the following diagram commutes:

$$\begin{tikzcd}
\text{Resource terms} \arrow[r, "\|-\|"] \arrow[d, "\text{typing}"'] & \text{Isogmentations} \arrow[d, "\text{desequentialization}"] \\
\text{Type terms} \arrow[r, "\|-\|"'] & \text{Positions}
\end{tikzcd}$$

To express this more formally, we need to introduce a few additional notations.

**From normal terms to types.** In Lemma 7.1, we introduced the notations $\mathsf{ctx}(u)$ and $\mathsf{type}(u)$ respectively for the unique context and type term such that $\mathsf{ctx}(u) \vdash u : \mathsf{type}(u)$ is derivable. Given a finite set $\Gamma$ of sequence variables, we moreover write $\Delta^{\mathrm{n}}(\Gamma) = \Delta^{\mathrm{n}}_{\mathrm{v}}(\Gamma) \uplus \Delta^{\mathrm{n}}_{\mathrm{b}}(\Gamma) \uplus \Delta^{\mathrm{n}}_{!}(\Gamma) \uplus \Delta^{\mathrm{n}}_{\mathrm{s}}(\Gamma)$ for the set of normal resource terms with free sequence variables in $\Gamma$, and we define a function $\mathsf{ctx}_\Gamma : \Delta^{\mathrm{n}}(\Gamma) \to \mathscr{D}^{\Gamma}_{\mathrm{s}}$, using the notations of Section 7: $\mathsf{ctx}_\Gamma(u)(\vec{x}) := \left\langle \mathsf{ctx}(u)(\vec{x}(i)) \right\rangle_{i \in \mathbb{N}}$.

**From isogmentations to positions.** Recall that any augmentation $\mathsf{q} \in \mathsf{Aug}(A)$ is based on a configuration $(\!|\mathsf{q}|\!) \in \mathscr{C}(A)$, its *desequentialization*. In turn, this yields a position by taking the symmetry class. Composing with the constructions (tensor and $\vdash$) of Lemma 8.6, we get functions:

$$\begin{array}{rrcl}
(\!|-|\!)^{\Gamma}_{\mathrm{b}} : & \mathsf{Isog}_{\bullet}(\|\Gamma\| \vdash o) & \to & \mathsf{Pos}(\mathbf{U}^{\mathbb{N}})^{\Gamma} \\
(\!|-|\!)^{\Gamma}_{\mathrm{v}} : & \mathsf{Isog}_{\bullet}(\|\Gamma\| \vdash \mathbf{U}) & \to & \mathsf{Pos}(\mathbf{U}^{\mathbb{N}})^{\Gamma} \times \mathsf{Pos}_{\bullet}(\mathbf{U}) \\
(\!|-|\!)^{\Gamma}_{!} : & \mathsf{Isog}(\|\Gamma\| \vdash \mathbf{U}) & \to & \mathsf{Pos}(\mathbf{U}^{\mathbb{N}})^{\Gamma} \times \mathsf{Pos}(\mathbf{U}) \\
(\!|-|\!)^{\Gamma}_{\mathrm{s}} : & \mathsf{Isog}(\|\Gamma\| \vdash \mathbf{U}^{\mathbb{N}}) & \to & \mathsf{Pos}(\mathbf{U}^{\mathbb{N}})^{\Gamma} \times \mathsf{Pos}(\mathbf{U}^{\mathbb{N}})\,.
\end{array}$$

**Compatibility.** We are now equipped to state and prove:

**THEOREM 8.16.** *The correspondences of Theorem 8.8 and Theorem 8.15 are compatible. More formally, the diagrams of Figure 18 commute.*

**PROOF.** For any arena $A$, $x \in \mathscr{C}(A)$ and $i \in \mathbb{N}$, we write $i \cdot x \in \mathscr{C}(A^{\mathbb{N}})$ for the configuration with the same events and order, and such that $\partial_{i \cdot x}(a) = \langle i, \partial_x(a) \rangle$ for each $a \in |x|$. This operation obviously preserves symmetry, thus extends to positions. Moreover, for any $\vec{x} \in \Gamma$ and $\mathsf{x} \in \mathsf{Pos}(A)$, we write $(\vec{x} \mapsto \mathsf{x}) \in \mathsf{Pos}(A)^{\Gamma}$ for the $\Gamma$-indexed family sending $\vec{x}$ to $\mathsf{x}$ and each $\vec{y} \neq \vec{x}$ to $\overline{\varnothing}$.

$$\begin{tikzcd}
\Delta^{\mathrm{n}}_{\mathrm{b}}(\Gamma) \arrow[r, "\|-\|^{\Gamma}_{\mathrm{b}}"] \arrow[d, "\mathrm{ctx}_{\Gamma}"'] & \mathrm{Isog}_{\bullet}(\|\Gamma\| \vdash o) \arrow[d, "\llparenthesis - \rrparenthesis^{\Gamma}_{\mathrm{b}}"] & \Delta^{\mathrm{n}}_{\mathrm{v}}(\Gamma) \arrow[r, "\|-\|^{\Gamma}_{\mathrm{v}}"] \arrow[d, "\langle \mathrm{ctx}_{\Gamma}, \mathrm{type} \rangle"'] & \mathrm{Isog}_{\bullet}(\|\Gamma\| \vdash \mathbf{U}) \arrow[d, "\llparenthesis - \rrparenthesis^{\Gamma}_{\mathrm{v}}"] \\
\mathcal{D}^{\Gamma}_{\mathrm{s}} \arrow[r, "\kappa^{\Gamma}_{\mathrm{s}}"'] & \mathrm{Pos}(\mathbf{U}^{\mathbb{N}})^{\Gamma} & \mathcal{D}^{\Gamma}_{\mathrm{s}} \times \mathcal{D}_{\mathrm{v}} \arrow[r, "\kappa^{\Gamma}_{\mathrm{s}} \times \kappa_{\mathrm{v}}"'] & \mathrm{Pos}(\mathbf{U}^{\mathbb{N}})^{\Gamma} \times \mathrm{Pos}_{\bullet}(\mathbf{U}) \\
\Delta^{\mathrm{n}}_{!}(\Gamma) \arrow[r, "\|-\|^{\Gamma}_{!}"] \arrow[d, "\langle \mathrm{ctx}_{\Gamma}, \mathrm{type} \rangle"'] & \mathrm{Isog}(\|\Gamma\| \vdash \mathbf{U}) \arrow[d, "\llparenthesis - \rrparenthesis^{\Gamma}_{!}"] & \Delta^{\mathrm{n}}_{\mathrm{s}}(\Gamma) \arrow[r, "\|-\|^{\Gamma}_{\mathrm{s}}"] \arrow[d, "\langle \mathrm{ctx}_{\Gamma}, \mathrm{type} \rangle"'] & \mathrm{Isog}(\|\Gamma\| \vdash \mathbf{U}^{\mathbb{N}}) \arrow[d, "\llparenthesis - \rrparenthesis^{\Gamma}_{\mathrm{s}}"'] \\
\mathcal{D}^{\Gamma}_{\mathrm{s}} \times \mathcal{D}_{!} \arrow[r, "\kappa^{\Gamma}_{\mathrm{s}} \times \kappa_{!}"'] & \mathrm{Pos}(\mathbf{U}^{\mathbb{N}})^{\Gamma} \times \mathrm{Pos}(\mathbf{U}) & \mathcal{D}^{\Gamma}_{\mathrm{s}} \times \mathcal{D}_{\mathrm{s}} \arrow[r, "\kappa^{\Gamma}_{\mathrm{s}} \times \kappa_{\mathrm{s}}"'] & \mathrm{Pos}(\mathbf{U}^{\mathbb{N}})^{\Gamma} \times \mathrm{Pos}(\mathbf{U}^{\mathbb{N}})
\end{tikzcd}$$

**Figure 18.** Compatibility of the bijections

A careful analysis of the action on configurations of the constructions on isogmentations involved in the bijection of Theorem 8.15 then yields the following equalities:

$$\begin{array}{rcll}
\llparenthesis \square_{\vec{x},i}(\mathbf{q}) \rrparenthesis^{\Gamma}_{\mathrm{b}} & = & \gamma + (\vec{x} \mapsto i \cdot (\mathrm{y} \Rightarrow o)) & \text{for } \mathbf{q} \in \mathrm{Isog}(\|\Gamma\| \vdash \mathbf{U}^{\mathbb{N}}), \text{ with } \llparenthesis \mathbf{q} \rrparenthesis = \langle \gamma, \mathrm{y} \rangle; \\
\llparenthesis \Lambda_{\|\Gamma\|,\vec{x}}(\mathbf{q}) \rrparenthesis^{\Gamma}_{\mathrm{v}} & = & \left\langle \gamma' \setminus \vec{x}, \gamma'(\vec{x}) \Rightarrow o \right\rangle & \text{for } \mathbf{q} \in \mathrm{Isog}_{\bullet}(\|\Gamma, \vec{x}\| \vdash o), \text{ with } \llparenthesis \mathbf{q} \rrparenthesis = \gamma'; \\
\llparenthesis [\![ \mathbf{q}_1, \dots, \mathbf{q}_k ]\!] \rrparenthesis^{\Gamma}_{!} & = & \langle \sum \gamma_i, [\![ \mathrm{x}_i \mid 1 \leq i \leq k ]\!] \rangle & \text{for } \mathbf{q}_i \in \mathrm{Isog}_{\bullet}(\|\Gamma\| \vdash \mathbf{U}), \text{ with } \llparenthesis \mathbf{q}_i \rrparenthesis = \langle \gamma_i, \mathrm{x}_i \rangle; \\
\llparenthesis \mathbf{q} \mathbin{⁘} \mathbf{p} \rrparenthesis^{\Gamma}_{\mathrm{s}} & = & \langle \gamma_1 + \gamma_2, \mathrm{x} \mathbin{⁘} \mathrm{y} \rangle & \text{for } \mathbf{q} \in \mathrm{Isog}(\|\Gamma\| \vdash \mathbf{U}) \text{ and } \mathbf{p} \in \mathrm{Isog}(\|\Gamma\| \vdash \mathbf{U}^{\mathbb{N}}), \\
 & & & \text{with } \llparenthesis \mathbf{q} \rrparenthesis = \langle \gamma_1, \mathrm{x} \rangle \text{ and } \llparenthesis \mathbf{p} \rrparenthesis = \langle \gamma_2, \mathrm{y} \rangle;
\end{array}$$

and $\llparenthesis \iota \rrparenthesis^{\Gamma}_{\mathrm{s}} = \langle \langle \overline{\varnothing} \rangle_{\vec{y} \in \Gamma}, \overline{\varnothing} \rangle$. Here the sum of $\Gamma$-indexed families of positions is defined point-wise, $\mathrm{x} \in \mathrm{Pos}(\mathbf{U})$, $\mathrm{y} \in \mathrm{Pos}(\mathbf{U}^{\mathbb{N}})$, $\mathrm{x}_i \in \mathrm{Pos}_{\bullet}(\mathbf{U})$ for $1 \leq i \leq k$, $\gamma, \gamma_1, \gamma_2 \in \mathrm{Pos}(\mathbf{U}^{\mathbb{N}})^{\Gamma}$, $\gamma' \in \mathrm{Pos}(\mathbf{U}^{\mathbb{N}})^{\Gamma,\vec{x}}$, and $\gamma' \setminus \vec{x}$ denotes the restriction of $\gamma'$ to $\Gamma$.

We then show that the diagrams of Figure 18 commute, by mutual induction on type terms, observing that the above equations match the rules of the intersection type system in Section 7.2. ■

## 9. Directions for future work

As stated in our introduction, although we consider extensional Taylor expansion as a valuable contribution in itself, it is but a first step in a ongoing program to exhibit a precise, compositional relationship between Taylor expansion and game semantics.

We have made significant progress in the typed case by describing the game semantics interpretation $\mathcal{G}(m)$ of a resource term $m$ in a compositional way, as is exposed by the structure of a resource category, which ensures its compatibility with resource reduction [5]. Moreover, on $\eta$-long resource terms in normal form, this interpretation coincides with the simply typed

$$\begin{tikzcd}
m \arrow[rrrr, mapsto, "\mathcal{N}"] \arrow[dd, mapsto, "\mathcal{G}"'] & & & & \mathcal{N}(m) \arrow[dd, phantom, "\wr"] \\
& & (1) & & \\
\mathcal{G}(m) & = & \mathcal{G}(\mathcal{N}(m)) & = & \|\mathcal{N}(m)\|_{\mathrm{v}}
\end{tikzcd}
\qquad
\begin{tikzcd}
M \arrow[rr, mapsto, "\mathcal{T}"] \arrow[dd, mapsto, "\mathcal{G}"'] & & \mathcal{T}(M) \arrow[rr, mapsto, "\mathcal{N}"] \arrow[dd, mapsto, "\mathcal{G}"] & & \mathcal{N}\mathcal{T}_\eta(M) \arrow[dd, phantom, "\wr"] \\
& (2) & & (1') & \\
\mathcal{G}(M) & = & \mathcal{G}(\mathcal{T}(M)) & = & \|\mathcal{N}\mathcal{T}_\eta(M)\|_{\mathrm{v}}
\end{tikzcd}$$

**Figure 19.** A correspondence between extensional Taylor expansion and game semantics

version of the bijection $\|-\|$: this makes the diagram (1) of Figure 19 commute – the commutation of diagram (1′) follows directly, by linearity. We are convinced that the extensional resource calculus provides the appropriate syntax and dynamics to adapt this approach in the untyped setting.

The next step, that is the subject of ongoing work, is to check that the game semantics $\mathcal{G}(M)$ of a typed $\lambda$-term $M$ coincides with that of $\mathcal{T}(M)$. More precisely we aim to establish how, given a resource category with enough structure (satisfied by the category of games interpreting the resource calculus), one can construct a cartesian closed category (which, in the case of games, is the usual interpretation of the $\lambda$-calculus), in such a way that Taylor expansion describes exactly the relation between both interpretations. This would make the diagram (2) of Figure 19 commute, altogether establishing that $\mathcal{G}(M)$ and $\mathcal{N}\mathcal{T}_\eta(M)$ are isomorphic interpretations. Again, we believe that extensional Taylor expansion can provide the appropriate framework to adapt this line of research in the untyped setting.

Another direction for future work is to lift the completeness condition on the semiring of coefficients, by means of uniformity. Indeed, ordinary Taylor expansion is amenable to a more general quantitative setting:

- it is easy to check that the definition of $\mathcal{T}(M)$ uses finite sums of non-zero coefficients only;
- it is possible to show, relying on a suitable notion of parallel resource reduction, that the compatibility of Taylor expansion with $\beta$-reduction does not require infinite sums either [39];
- and Ehrhard and Regnier have shown that, due to the uniformity properties of the Taylor expansion of pure $\lambda$-terms, no sum of coefficients is actually performed during normalization, in the computation of $\mathcal{N}\mathcal{T}_\eta(M)$ [19].

The first observation is easy to reproduce in the extensional setting, and we are confident that the third one can be adapted, because the very same uniformity arguments apply (in fact, we already know that the coefficients are finite, both in the extensional Taylor expansion of a term and in its normal form, thanks to Lemmas 6.8 and 5.10). The compatibility with $\beta$-reduction needs more care, though. Indeed, our simulation results (Theorems 5.12 and 5.14) rely on unbounded iterations of resource reductions (see Section 4.3). This was essential, in particular to deal with the $\eta$-expansion of variables: Lemma 5.4 does require the iterated reduction of possibly created redexes, and could not be reproduced with a single step of parallel reduction.

Nonetheless, we believe that one could adapt the approach of Cerda, in his study of Taylor expansion for the infinitary $\lambda$-calculus [9, esp. Section 4.4]: there, he considers a restricted version of parallel reduction, that is sufficient to capture $\beta$-reduction, yet preserves uniformity, which allows him to safely consider iterated reductions and normalization, without completeness hypothesis.

---

## References

[1] **Davide Barbarossa and Giulio Manzonetto**. Taylor subsumes Scott, Berry, Kahn and Plotkin. *Proc. ACM Program. Lang.* 4(POPL):1:1–1:23, 2020. DOI (8)

[2] **Hendrik Pieter Barendregt**. The Lambda Calculus - Its Syntax and Semantics, volume 103 of *Studies in logic and the foundations of mathematics*. North-Holland, 1985. (2, 7, 12, 14, 58, 62)

[3] **Henk Barendregt and Giulio Manzonetto**. A Lambda Calculus Satellite. College Publications, 2022. URL (2, 4)

[4] **Lison Blondeau-Patissier and Pierre Clairambault**. Positional injectivity for innocent strategies. *6th International Conference on Formal Structures for Computation and Deduction, FSCD 2021, July 17-24, 2021, Buenos Aires, Argentina (Virtual Conference)*, volume 195 of *LIPIcs*, 17:1–17:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. DOI (15, 72)

[5] **Lison Blondeau-Patissier, Pierre Clairambault, and Lionel Vaux Auclair**. Strategies as resource terms, and their categorical semantics. *Logical Methods in Computer Science*, Volume 21, Issue 4, 9, October 2025. DOI (9, 14, 16, 72, 73, 84, 87)

[6] **Pierre Boudes**. Thick subtrees, games and experiments. *Typed Lambda Calculi and Applications, 9th International Conference, TLCA 2009, Brasilia, Brazil, July 1-3, 2009. Proceedings*, volume 5608 of *Lecture Notes in Computer Science*, pages 65–79. Springer, 2009. DOI (75)

[7] **Antonio Bucciarelli, Alberto Carraro, Thomas Ehrhard, and Giulio Manzonetto**. Full abstraction for the resource lambda calculus with tests, through Taylor expansion. *Log. Methods Comput. Sci.* 8(4), 2012. DOI (15, 34–36)

[8] **Antonio Bucciarelli, Thomas Ehrhard, and Giulio Manzonetto**. Not enough points is enough. *Computer Science Logic, 21st International Workshop, CSL 2007, 16th Annual Conference of the EACSL, Lausanne, Switzerland, September 11-15, 2007, Proceedings*, volume 4646 of *Lecture Notes in Computer Science*, pages 298–312. Springer, 2007. DOI (14, 34, 64, 72, 78)

[9] **Rémy Cerda**. Taylor Approximation and Infinitary $\lambda$-Calculi. Thèse de doctorat, Aix-Marseille Université, June 2024. URL (89)

[10] **Jules Chouquet**. Taylor expansion, finiteness and strategies. *Proceedings of the Thirty-Fifth Conference on the Mathematical Foundations of Programming Semantics, MFPS 2019, London, UK, June 4-7, 2019*, volume 347 of *Electronic Notes in Theoretical Computer Science*, pages 65–85. Elsevier, 2019. DOI (9)

[11] **Jules Chouquet and Christine Tasson**. Taylor expansion for call-by-push-value. *28th EACSL Annual Conference on Computer Science Logic, CSL 2020, January 13-16, 2020, Barcelona, Spain*, volume 152 of *LIPIcs*, 16:1–16:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. DOI (9)

[12] **M. Coppo, M. Dezani-Ciancaglini, and S. Ronchi della Rocca**. (Semi)-separability of finite sets of terms in Scott's $D_\infty$-models of the $\lambda$-calculus. *Automata, Languages and Programming, ICALP 1978*, pages 142–164, Berlin, Heidelberg. Springer Berlin Heidelberg, 1978. DOI (63)

[13] **Ugo Dal Lago and Thomas Leventis**. On the Taylor expansion of probabilistic $\lambda$-terms. *4th International Conference on Formal Structures for Computation and Deduction, FSCD 2019, June 24-30, 2019, Dortmund, Germany*, volume 131 of *LIPIcs*, 13:1–13:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. DOI (8)

[14] **Aloÿs Dufour and Damiano Mazza**. Böhm and Taylor for all! *9th International Conference on Formal Structures for Computation and Deduction, FSCD 2024, July 10-13, 2024, Tallinn, Estonia*, volume 299 of *LIPIcs*, 29:1–29:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. DOI (9)

[15] **Thomas Ehrhard**. Finiteness spaces. *Math. Struct. Comput. Sci.* 15(4):615–646, 2005. DOI (8)

[16] **Thomas Ehrhard and Laurent Regnier**. Böhm trees, Krivine's machine and the Taylor expansion of lambda-terms. *Logical Approaches to Computational Barriers, Second Conference on Computability in Europe, CiE 2006, Swansea, UK, June 30-July 5, 2006, Proceedings*, volume 3988 of *Lecture Notes in Computer Science*, pages 186–197. Springer, 2006. DOI (2, 7)

[17] **Thomas Ehrhard and Laurent Regnier**. Differential interaction nets. *Theor. Comput. Sci.* 364(2):166–195, 2006. DOI (8)

[18] **Thomas Ehrhard and Laurent Regnier**. The differential lambda-calculus. *Theor. Comput. Sci.* 309(1-3):1–41, 2003. DOI (6, 15)

[19] **Thomas Ehrhard and Laurent Regnier**. Uniformity and the Taylor expansion of ordinary lambda-terms. *Theor. Comput. Sci.* 403(2-3):347–372, 2008. DOI (2, 6–8, 88)

[20] **Samuel Eilenberg**. Automata, Languages, and Machines, volume 59-A of *Pure and Applied Mathematics*. Academic Press, 1974. (37)

[21] **Jean-Yves Girard**. Linear logic. *Theor. Comput. Sci.* 50:1–102, 1987. DOI (8)

[22] **Jean-Yves Girard**. Normal functors, power series and $\lambda$-calculus. *Ann. Pure Appl. Log.* 37(2):129–177, 1988. DOI (8)

[23] **J. Martin E. Hyland**. A survey of some useful partial order relations on terms of the lambda calculus. *Lambda-Calculus and Computer Science Theory, Proceedings of the Symposium Held in Rome, Italy, March 25-27, 1975*, volume 37 of *Lecture Notes in Computer Science*, pages 83–95. Springer, 1975. DOI (9)

[24] **J. Martin E. Hyland and C.-H. Luke Ong**. On full abstraction for PCF: I, II, and III. *Inf. Comput.* 163(2):285–408, 2000. DOI (14)

[25] **Richard Kennaway**, **Jan Willem Klop**, **M. Ronan Sleep**, **and Fer-Jan de Vries**. Infinitary lambda calculus. *Theor. Comput. Sci.* 175(1):93–125, 1997. DOI (4, 12)

[26] **Andrew D. Ker**, **Hanno Nickau**, **and C.-H. Luke Ong**. Innocent game models of untyped lambda-calculus. *Theor. Comput. Sci.* 272(1-2):247–292, 2002. DOI (15, 73)

[27] **Axel Kerinec**, **Giulio Manzonetto**, **and Michele Pagani**. Revisiting call-by-value Böhm trees in light of their Taylor expansion. *Log. Methods Comput. Sci.* 16(3), 2020. DOI (9)

[28] **Jean-Louis Krivine**. Lambda-calculus types and models. Université Paris 7, February 2002. DEA. URL (2, 7, 14, 63)

[29] **Thomas Leventis**. Probabilistic Böhm trees and probabilistic separation. *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018*, pages 649–658. ACM, 2018. DOI (8)

[30] **Giulio Manzonetto**. A general class of models of $\mathcal{H}^*$. *Mathematical Foundations of Computer Science 2009, 34th International Symposium, MFCS 2009, Novy Smokovec, High Tatras, Slovakia, August 24-28, 2009. Proceedings*, volume 5734 of *Lecture Notes in Computer Science*, pages 574–586. Springer, 2009. DOI (14, 65, 72)

[31] **Giulio Manzonetto**. What is a categorical model of the differential and the resource $\lambda$-calculi? *Math. Struct. Comput. Sci.* 22(3):451–520, 2012. DOI (65)

[32] **Giulio Manzonetto and Domenico Ruoppolo**. Relational graph models, Taylor expansion and extensionality. *Proceedings of the 30th Conference on the Mathematical Foundations of Programming Semantics, MFPS 2014, Ithaca, NY, USA, June 12-15, 2014*, volume 308 of *Electronic Notes in Theoretical Computer Science*, pages 245–272. Elsevier, 2014. DOI (9)

[33] **Paul-André Melliès**. Asynchronous games 2: the true concurrency of innocence. *Theor. Comput. Sci.* 358(2-3):200–228, 2006. DOI (14)

[34] **Reiji Nakajima**. Infinite normal forms for the lambda-calculus. *Lambda-Calculus and Computer Science Theory, Proceedings of the Symposium Held in Rome, Italy, March 25-27, 1975*, volume 37 of *Lecture Notes in Computer Science*, pages 62–82. Springer, 1975. DOI (12, 56, 58)

[35] **Federico Olimpieri and Lionel Vaux Auclair**. On the Taylor expansion of $\lambda$-terms and the groupoid structure of their rigid approximants. *Log. Methods Comput. Sci.* 18(1), 2022. DOI (41)

[36] **Masako Takahashi**. Parallel reductions in lambda-calculus. *Inf. Comput.* 118(1):120–127, 1995. DOI (7)

[37] **Takeshi Tsukada**, **Kazuyuki Asada**, **and C.-H. Luke Ong**. Generalised species of rigid resource terms. *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*, pages 1–12. IEEE Computer Society, 2017. DOI (9)

[38] **Takeshi Tsukada and C.-H. Luke Ong**. Plays as resource terms via non-idempotent intersection types. *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, pages 237–246. ACM, 2016. DOI (9, 14–16)

[39] **Lionel Vaux**. Normalizing the Taylor expansion of non-deterministic $\lambda$-terms, via parallel reduction of resource vectors. *Log. Methods Comput. Sci.* 15(3), 2019. DOI (2, 6, 7, 41, 43, 88)