

# The Benefit of Being Non-Lazy in Probabilistic $\lambda$ -calculus

Applicative Bisimulation is Fully Abstract for Non-Lazy Probabilistic Call-by-Name

Gianluca Curzi  
University of Turin  
Turin, Italy  
gcurzi@unito.it

Michele Pagani  
IRIF UMR 8243, Université de Paris, CNRS  
Paris, France  
pagani@irif.fr

## Abstract

We consider the probabilistic applicative bisimilarity (PAB) — a coinductive relation comparing the applicative behaviour of probabilistic untyped  $\lambda$ -terms according to a specific operational semantics. This notion has been studied by Dal Lago et al. with respect to the two standard parameter passing policies, call-by-value (cbv) and call-by-name (cbn), using a lazy reduction strategy not reducing within the body of a function. In particular, PAB has been proven to be fully abstract with respect to the contextual equivalence in cbv [6] but not in lazy cbn [16].

We overcome this issue of cbn by relaxing the laziness constraint: we prove that PAB is fully abstract with respect to the standard head reduction contextual equivalence. Our proof is based on Leventis' Separation Theorem [19], using probabilistic Nakajima trees as a tree-like representation of the contextual equivalence classes.

Finally, we prove also that the inequality full abstraction fails, showing that the probabilistic applicative similarity is strictly contained in the contextual preorder.

**CCS Concepts:** • Software and its engineering → Semantics; • Theory of computation → Program semantics.

**Keywords:** Probabilistic lambda calculus, Bisimilarity, Full abstraction, Observational equivalence, Separation

## ACM Reference Format:

Gianluca Curzi and Michele Pagani. 2020. The Benefit of Being Non-Lazy in Probabilistic  $\lambda$ -calculus: Applicative Bisimulation is Fully Abstract for Non-Lazy Probabilistic Call-by-Name. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
*LICS '20, July 8–11, 2020, Saarbrücken, Germany*  
© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-7104-9/20/07...\$15.00

DOI: 10.1145/3373718.3394806

*(LICS '20), July 8–11, 2020, Saarbrücken, Germany. ACM, New York, NY, USA, 14 pages.*

## 1 Introduction

The probabilistic  $\lambda$ -calculus  $\Lambda_{\oplus}$  extends the pure untyped  $\lambda$ -calculus with a sum  $M \oplus N$ , evaluating to  $M$  or  $N$  with equal probability 0.5. The operational semantics gives then a function mapping a term  $M$  to a probability distribution  $\llbracket M \rrbracket$  of values. Exactly as in standard  $\lambda$ -calculus, different design choices may affect the meaning  $\llbracket M \rrbracket$  of a term.

First, one has to decide how to evaluate a  $\beta$ -redex, i.e. the application of a function  $\lambda x.M$  to an argument  $N$ . There are two main evaluation mechanisms: the *call-by-value policy* (cbv) consists first in evaluating  $N$  to some value  $V$  and then replacing the parameter  $x$  in  $M$  with  $V$ , while the *call-by-name policy* (cbn) replaces  $x$  with  $N$  as it is, before any evaluation. It is well-known that the two policies give rise to different results, especially in a probabilistic setting. Consider for example the term  $(\lambda vz.vv)(T \oplus F)$ , where  $T = \lambda xy.x$  and  $F = \lambda xy.y$ . In cbv, we first evaluate  $T \oplus F$ , yielding either  $T$  or  $F$  with equal probability, and then we pass the result to the function  $\lambda vz.vv$ , producing either  $\lambda z.TT$  or  $\lambda z.FF$ , both with probability 0.5. By contrast, in cbn we pass the whole term  $T \oplus F$  to the function before evaluating it, obtaining  $\lambda z.(T \oplus F)(T \oplus F)$  with probability 1.

Second, one has to define which redexes to evaluate in a term, if any. Also in this case, there are various choices in  $\lambda$ -calculus: the *lazy strategy*, forbidding any reduction in the body of a function, so that  $\lambda x.M$  is a value whatever  $M$  is, or the *head reduction*, consisting in reducing the redex in head position, which is at the left of any application. Again, the choice gives rise to different meanings, the meaning of a term w.r.t. the head reduction is a distribution of head normal forms.

By the way, let us remark here that some variants of the standard head reduction have been considered in the literature, as for example the *head spine reduction* that, given a  $\beta$ -redex  $(\lambda x.M)N$ , first evaluates the body of  $M$  and then evaluates the outermost redex according to cbn. A side result of our paper is that the head and head spine strategies are actually equivalent, even in a probabilistic setting (Theorem 2).

Comparing terms by their operational semantics is too narrow, as higher-order normal forms differ often by syntactical details that are inessential with respect to their computational behaviour. Contextual equivalence is usually considered: two terms  $M, N$  are contextually equivalent ( $M =_{\text{cxt}} N$  in symbols) whenever they “behave” the same in any possible “programming context”. This definition depends on the notion of *context* and on that of *observable behavior*. In  $\Lambda_{\oplus}$ , a context  $C$  is a term with a special variable  $[\cdot]$ , the *hole*, and what we observe is the total mass of the distribution  $\llbracket C[M] \rrbracket$ , i.e. the total probability of getting a result from the evaluation of the term  $C[M]$  obtained by replacing the hole with  $M$ . The definition of  $=_{\text{cxt}}$  depends therefore on the chosen operational semantics but it is more canonical than the latter.

Proving that two terms are contextually equivalent is rather difficult since we have to consider *all* contexts, hence the quest for more tractable equivalences comparable with  $=_{\text{cxt}}$ . We say in particular that an equivalence  $\equiv$  over  $\lambda$ -terms is *sound* with respect to  $=_{\text{cxt}}$  whenever the former implies the latter (i.e.  $\equiv \subseteq =_{\text{cxt}}$ ), it is *complete* if the converse holds (i.e.  $=_{\text{cxt}} \subseteq \equiv$ ) and it is *fully abstract* if it is both sound and complete, i.e. the two relations coincide.

In probabilistic  $\lambda$ -calculus, the first results in this line of research have been achieved in the setting of the denotational semantics of the  $\Lambda_{\oplus}$  head reduction. In particular, Ehrhard et al. prove that the equivalence  $\equiv_{\mathcal{D}^\infty}$  induced by the reflexive object  $\mathcal{D}^\infty$  of the cartesian closed category of probabilistic coherence spaces [11] (as well as of the weighted relations [17]) is sound. More recently, Leventis proves a fundamental separation theorem, giving as a consequence that the probabilistic Nakajima tree equality is complete [19]. From the latter result, Clairambault and Paquet derive a fully abstract game model of  $\Lambda_{\oplus}$  and as a corollary also the full abstraction of  $\mathcal{D}^\infty$  [5]. The latter result has been also achieved independently by Leventis and Pagani [20].

All the above results deal with the head reduction, i.e. a non-lazy cbn operational semantics. For lazy strategies, a different approach is available, based on the notion of *applicative bisimulation*, which is the true object of this paper. The idea dates back to [1] and consists in looking at the operational semantics as a transition system having  $\lambda$ -terms as states and transitions given by the evaluation of the application between  $\lambda$ -terms. The benefit of this setting is to transport into  $\lambda$ -calculus the whole theory of bisimilarity and its associated coinductive reasoning, which is a fundamental tool for comparing processes in concurrency theory. Basically, two terms  $M$  and  $N$  are applicative bisimilar (in symbols  $M \sim N$ ) whenever their applications  $MP$  and  $NP$  reduce to applicative bisimilar values for any argument  $P$ .

This approach has been lifted to the probabilistic  $\lambda$ -calculus in a series of works by Dal Lago et al. [6, 7, 16], introducing the notion of *probabilistic applicative bisimilarity* (PAB) for lazy semantics. In particular, PAB is proven to be sound with the contextual equivalence in both cbv and cbn, but only cbv

PAB is fully abstract. In case of lazy cbn, we have terms like:

$$M \triangleq \lambda xy.(x \oplus y) \quad N \triangleq (\lambda xy.x) \oplus (\lambda xy.y) \quad (1)$$

such that  $M =_{\text{cxt}} N$  but  $M \not\sim N$ . In fact, lazy PAB is able to discriminate between a term where a choice can be performed *before* any interaction, like  $N$ , and a term that needs to interact in order to trigger a choice, like  $M$ . Notice that this difference is caught also by cbv contextual semantics, as these two terms are distinguished by the context  $C = (\lambda v.(v\mathbf{I}\Omega)(v\mathbf{I}\Omega))[\cdot]$  in cbv (the total mass of  $\llbracket C[M] \rrbracket_{\text{cbv}}$  is 0.25, while that of  $\llbracket C[N] \rrbracket_{\text{cbv}}$  is 0.5), but not in cbn (namely,  $\llbracket C[M] \rrbracket_{\text{cbn}} = \llbracket C[N] \rrbracket_{\text{cbn}}$  has mass 0.25).

In [16] the authors analyse this example remarking that the cbn policy misses the “capability to copy a term *after* having evaluated it”. This is indeed a fundamental primitive in probabilistic programming: when implementing a probabilistic algorithm we need often to toss a coin and then to pass *the result* of this tossing to several subroutines. It is so common to extend a probabilistic language with a *let-in* constructor, often called *sampling*, evaluating a choice *before* passing it to a function even in a cbn semantics. As expected, it is shown [15] that such an extension recovers cbn PAB full abstraction, as terms like (1) become contextually different.

Let us remark that we are here in front of two disconcerting facts. First, it has been proven that in simply typed languages the presence of the *let-in* constructor does not affect the discriminating power of the contextual equivalence, for example in probabilistic PCF the lazy cbn contextual equivalence coincides with the equality in the model of probabilistic coherence spaces [12, 13], with or without a sampling primitive. Why this striking difference with an untyped framework? Second, we have already mentioned several denotational models of  $\Lambda_{\oplus}$  which are fully abstract with respect to a pure cbn contextual equivalence, so without this “capability to copy a term *after* having evaluated it”. Is it really so necessary for getting a fully abstract PAB?

The first question can be easily answered by focussing on the laziness constraint of the operational semantics. Every  $\lambda$ -abstraction is a value for a lazy semantics. This does not affect the set of observables in a simply typed setting (as PCF), because this is defined on ground types (booleans, numerals, etc). By contrast, every term is a function in an untyped setting, so the laziness radically changes what we can observe in the behaviour of a term. The goal of this paper is to show that also the second question deals with laziness: we prove that PAB is fully abstract for the head reduction (Theorem 23). This is unexpected: non-lazy semantics seems not needing the sampling primitives in order to have fully abstract PAB, even with a cbn policy and an untyped setting.

On a more technical side, we stress that our proofs of soundness and completeness follow a different reasoning than the one used in probabilistic lazy semantics [6, 7, 15]. First, the soundness ( $\sim \subseteq =_{\text{cxt}}$ ) does not need an Howe lifting [14], as we prove a Context Lemma (Lemma 9) for  $=_{\text{cxt}}$  and

an applicative property of  $\sim$  (Lemma 15), the latter using the notion of probabilistic assignments as in [16]. Second, and more fundamental, the proof of completeness ( $=_{\text{ctx}} \subseteq \sim$ ) is not achieved by transforming PAB into a testing equivalence using a theorem by van Breugel et al. [25]. Rather, we use Leventis' Separation property [19] to prove that the contextual equivalence is a probabilistic applicative bisimulation and so contained in PAB by definition (Theorem 23).

What about inequalities? All equivalences so far introduced have an asymmetric version: the contextual preorder and the probabilistic applicative similarity (PAS). We prove also that PAS is sound but not complete with respect to the contextual inequality. A counterexample to the full abstraction in the asymmetric case is given in Section 4 and it is further discussed in the Conclusion.

We refer to the extended version [8] for the details of many proofs.

**Notation.** We write  $\mathbb{N}$  for the set of natural numbers,  $\mathbb{R}$  for the set of real numbers and  $[0, 1]$  for the unit interval of  $\mathbb{R}$ .

A *subprobability distribution over a countable set  $X$*  is a function  $f : X \rightarrow [0, 1]$  such that  $\sum_{x \in X} f(x) \leq 1$ . Distributions are ranged over by  $\mathcal{D}, \mathcal{E}, \mathcal{F}, \dots$  and  $\mathcal{D}(X)$  denotes the set of all subprobability distributions over  $X$ . Given a distribution  $\mathcal{D} \in \mathcal{D}(X)$ , its *support*  $\text{supp}(\mathcal{D})$  is the subset of all elements in  $X$  such that  $\mathcal{D}(x) > 0$ , its *mass*  $\sum \mathcal{D}$  is simply  $\sum_{x \in X} \mathcal{D}(x)$ . Given  $x_1, \dots, x_n \in X$ , the expression  $p_1x_1 + \dots + p_nx_n$  is used to denote the distribution  $\mathcal{D} \in \mathcal{D}(X)$  with finite support  $\{x_1, \dots, x_n\}$  such that  $\mathcal{D}(x_i) = p_i$ , for every  $i \leq n$ . Notice that, in this case,  $\sum \mathcal{D} = \sum_{i=1}^n p_i$ . The symbol  $\perp$  denotes the empty distribution and  $x$  can denote both an element in  $X$  and the distribution having all its mass on  $x$ . Given a (possibly infinite) index set  $I$ , a family  $\{r_i\}_{i \in I}$  of positive real numbers such that  $\sum_{i \in I} r_i \leq 1$  and a family  $\{\mathcal{D}_i\}_{i \in I}$  of distributions, the distribution  $\sum_{i \in I} r_i \cdot \mathcal{D}_i$  is defined, for all  $x \in X$ , by  $(\sum_{i \in I} r_i \cdot \mathcal{D}_i)(x) = \sum_{i \in I} r_i \cdot \mathcal{D}_i(x)$ .

A relation  $\mathcal{R}$  over a set  $X$  is a subset of  $X \times X$ . Given a relation  $\mathcal{R}$  over a set  $X$  and  $Y \subseteq X$ ,  $\mathcal{R}(Y)$  denotes the image of  $Y$  under  $\mathcal{R}$ , i.e. the set  $\{x \mid \exists y \in Y (y, x) \in \mathcal{R}\}$ ,  $\mathcal{R}^{op}$  represents the converse of  $\mathcal{R}$ , i.e.  $\{(x, y) \mid (y, x) \in \mathcal{R}\}$ , and  $\mathcal{R}^*$  the reflexive and transitive closure of  $\mathcal{R}$ . Moreover, if  $\mathcal{R}$  is an equivalence relation, then  $X/\mathcal{R}$  stands for the set of all equivalence classes of  $X$  modulo  $\mathcal{R}$ .

## 2 Preliminaries

This section introduces the fundamental notions of the paper. We first present the syntax and the operational semantics of the probabilistic  $\lambda$ -calculus  $\Lambda_{\oplus}$ , on top of which we shall consider the contextual equivalence and the contextual preorder relations. Then, we recall Larsen and Skou's probabilistic (bi)similarity on labelled Markov chains [18] and, in the spirit

of Abramsky's work on applicative (bi)similarity [1] and following [6, 7, 15, 16], we apply it to the operational semantics of  $\Lambda_{\oplus}$ , getting the probabilistic applicative (bi)similarity.

### 2.1 The Probabilistic $\lambda$ -calculus $\Lambda_{\oplus}$

The set  $\Lambda_{\oplus}$  of *probabilistic  $\lambda$ -terms* over a given set  $\mathcal{V}$  of variables is generated by the following grammar:

$$M, N := x \mid \lambda x.M \mid (MN) \mid M \oplus N \quad (2)$$

where  $x \in \mathcal{V}$ . We consider the usual conventions as in [3], so for example application is left-associative and has higher precedence than  $\lambda$ -abstraction. Parenthesis can be omitted when clear from the context. A term is in (or is a) *head normal form*, or *hnf* for short, if it is of the form  $\lambda x_1 \dots x_n.yN_1 \dots N_m$ , for some  $n, m \in \mathbb{N}$ . If  $n = 0$  then the term is also called *neutral*. Head normal forms are ranged over by metavariables like  $H$ . The set of all hnfs will be denoted by HNF, the set of all neutral terms will be denoted by NEUT.

Terms are considered modulo renaming of bound variables. The set  $FV(M)$  of the free variables of a term  $M$  and the capture-free substitution  $M[N/x]$  of  $N$  for the free occurrences of  $x$  in  $M$  are defined in the standard way. Finite subsets of  $\mathcal{V}$  are ranged over by  $\Gamma$ . Given  $\Gamma$ , the set of terms (resp. head normal forms) whose free variables are within  $\Gamma$  is denoted  $\Lambda_{\oplus}^{\Gamma}$  (resp.  $\text{HNF}^{\Gamma}$ ).

**Example 1.** Useful terms are the identity  $\mathbf{I} \triangleq \lambda x.x$ , the boolean values  $\mathbf{T} \triangleq \lambda xy.x$  and  $\mathbf{F} \triangleq \lambda xy.y$ , the duplicator  $\Delta \triangleq \lambda x.xx$ , the Turing fixed-point combinator  $\Theta \triangleq (\lambda x.\lambda y.(y(xxy)))(\lambda x.\lambda y.(y(xxy)))$  and the ever looping term  $\Omega \triangleq \Delta\Delta$ . An example of probabilistic  $\lambda$ -term that does not belong to the standard  $\lambda$ -calculus is  $\text{hid} \triangleq \mathbf{I} \oplus \Omega$ .

Let  $\mathfrak{D}(\text{HNF})$  be the set of all subprobability distributions over HNF, called *head distributions*. Let  $\mathcal{D} \in \mathfrak{D}(\text{HNF})$ , we define  $\lambda x.\mathcal{D}$  as  $(\lambda x.\mathcal{D})(H) \triangleq \mathcal{D}(H')$ , if  $H = \lambda x.H'$ , for some  $H' \in \text{HNF}$ , otherwise  $(\lambda x.\mathcal{D})(H) \triangleq 0$ . If  $X \subseteq \text{HNF}$ , we let  $\mathcal{D}(X) \triangleq \sum_{H \in X} \mathcal{D}(H)$ . We may also write  $\mathcal{D}(X)$  for a generic subset  $X \subseteq \Lambda_{\oplus}$  of terms, meaning in fact  $\mathcal{D}(X \cap \text{HNF})$ .

Subprobability distributions allow us to model divergence and to look at some distributions as “approximations” of others. To formally define this, we lift the canonical order on  $\mathbb{R}$  pointwise: we set  $\mathcal{D} \leq_{\mathfrak{D}} \mathcal{E}$  if and only if  $\forall H \in \text{HNF}, \mathcal{D}(H) \leq \mathcal{E}(H)$ . Notice that  $\leq_{\mathfrak{D}}$  is a directed-complete partial order over  $\mathfrak{D}(\text{HNF})$ ,  $\perp$  being the least element.

We now endow  $\Lambda_{\oplus}$  with a big-step probabilistic operational semantics in two stages, following Dal Lago and Zorzi [9]. First, the rules of Figure 1 define a big-step approximation relation  $M \Downarrow \mathcal{D}$  between a term  $M$  and a head distribution  $\mathcal{D}$ . This relation is not a function: many different head distributions can be put in correspondence with the same term  $M$ , because of the rule  $s1$  that allows one to “give up” while looking for a distribution of a term. The big-step semantics is then the supremum of all such finite

$$\begin{array}{c}
\frac{}{M \Downarrow \perp} \text{ s1} \qquad \frac{}{x \Downarrow x} \text{ s2} \qquad \frac{M \Downarrow \mathcal{D}}{\lambda x.M \Downarrow \lambda x.\mathcal{D}} \text{ s3} \\
\frac{M \Downarrow \mathcal{D} \quad \{H[N/x] \Downarrow \mathcal{E}_{H,N}\}_{\lambda x.H \in \text{supp}(\mathcal{D})}}{MN \Downarrow \sum_{\lambda x.H \in \text{supp}(\mathcal{D})} \mathcal{D}(\lambda x.H) \cdot \mathcal{E}_{H,N} + \sum_{H \in \text{supp}(\mathcal{D}) \cap \text{NEUT}} \mathcal{D}(H) \cdot HN} \text{ s4} \qquad \frac{M \Downarrow \mathcal{D} \quad N \Downarrow \mathcal{E}}{M \oplus N \Downarrow \frac{1}{2} \cdot \mathcal{D} + \frac{1}{2} \cdot \mathcal{E}} \text{ s5}
\end{array}$$

Figure 1. Big-step approximation.

approximations:

$$\llbracket M \rrbracket \triangleq \sup\{\mathcal{D} \mid M \Downarrow \mathcal{D}\} \quad (3)$$

Observe that this supremum is guaranteed to exist since  $\{\mathcal{D} \in \mathfrak{D}(\text{HNF}) \mid M \Downarrow \mathcal{D}\}$  is a directed set, as can be proved by induction on  $M$ .

**Example 2.** Consider the term  $M \triangleq \Delta(\mathbf{T} \oplus \mathbf{F})$ . One can easily check that the rules in Figure 1 allow us to derive  $M \Downarrow \mathcal{D}$  for any  $\mathcal{D}$  in the following set  $\left\{ \perp, \frac{1}{4} \cdot \lambda y. \mathbf{T}, \frac{1}{4} \cdot \lambda y. \mathbf{F}, \frac{1}{2} \cdot \mathbf{I}, \frac{1}{4} \cdot \lambda y. \mathbf{T} + \frac{1}{4} \cdot \lambda y. \mathbf{F}, \frac{1}{4} \cdot \lambda y. \mathbf{T} + \frac{1}{2} \cdot \mathbf{I}, \frac{1}{4} \cdot \lambda y. \mathbf{F} + \frac{1}{2} \cdot \mathbf{I}, \frac{1}{4} \cdot \lambda y. \mathbf{T} + \frac{1}{4} \cdot \lambda y. \mathbf{F} + \frac{1}{2} \cdot \mathbf{I} \right\}$ . The latter head distribution is the supremum of this set and so it defines the semantics of  $M$ .

Example 2 is about normalizing terms, which means here terms  $M$  with semantics of total mass  $\sum \llbracket M \rrbracket = 1$  and such that there exists a unique finite derivation giving  $M \Downarrow \llbracket M \rrbracket$ . Standard non-converging terms gives partiality:

**Example 3.** By inspection on the rule s4 in Figure 1, one can check that  $\Omega \Downarrow \mathcal{D}$  only if  $\mathcal{D} = \perp$ , so  $\llbracket \Omega \rrbracket = \perp$ . As a consequence we also have, e.g.  $\llbracket \Omega \oplus \mathbf{I} \rrbracket = \frac{1}{2} \cdot \mathbf{I}$ .

The probabilistic  $\lambda$ -calculus allows us also for *almost sure terminating* terms, namely terms  $M$  such that  $\sum \llbracket M \rrbracket = 1$  but *without* finite derivations of  $M \Downarrow \llbracket M \rrbracket$ :

**Example 4.** Consider the derivation of  $MM \Downarrow \sum_{i=1}^n \frac{1}{2^i} \cdot y$  depicted in Figure 2, where  $M \triangleq \lambda x.(y \oplus xx)$ . Any such finite approximation of  $\llbracket MM \rrbracket$  gives a head distribution of the form  $\sum_{i=1}^n \frac{1}{2^i} \cdot y$ , for some  $n \geq 1$ , but only the limit sum  $\sup_{i=1}^n \sum \frac{1}{2^i} \cdot y$  is equal to  $y$ , thus yielding  $\llbracket MM \rrbracket = y$ .

The operational semantics can be defined inductively as follows:

**Proposition 1.** For every  $M, N \in \Lambda_{\oplus}$  and  $H \in \text{HNF}$ :

- (1)  $\llbracket (\lambda x.H)N \rrbracket = \llbracket H[N/x] \rrbracket$ .
- (2)  $\llbracket \lambda x.M \rrbracket = \lambda x. \llbracket M \rrbracket$ .
- (3)  $\llbracket MN \rrbracket$  is equal to the following distribution:

$$\begin{aligned}
& \sum_{\lambda x.H \in \text{supp}(\llbracket M \rrbracket)} \llbracket M \rrbracket(\lambda x.H) \cdot \llbracket H[N/x] \rrbracket \\
& + \sum_{H \in \text{supp}(\llbracket M \rrbracket) \cap \text{NEUT}} \llbracket M \rrbracket(H) \cdot HN.
\end{aligned}$$

$$(4) \llbracket M \oplus N \rrbracket = \frac{1}{2} \llbracket M \rrbracket + \frac{1}{2} \llbracket N \rrbracket.$$

Moreover, for every  $H \in \text{HNF}$ ,  $\llbracket H \rrbracket = H$ .

Note that, if  $M$  is deterministic, i.e. a term without the probabilistic sum  $\oplus$ , then either  $M$  has a unique head normal form  $H$  and  $\llbracket M \rrbracket(H) = 1$ , or  $M$  is a diverging term and  $\llbracket M \rrbracket = \perp$ . So  $\llbracket \cdot \rrbracket$  generalises the usual deterministic semantics.

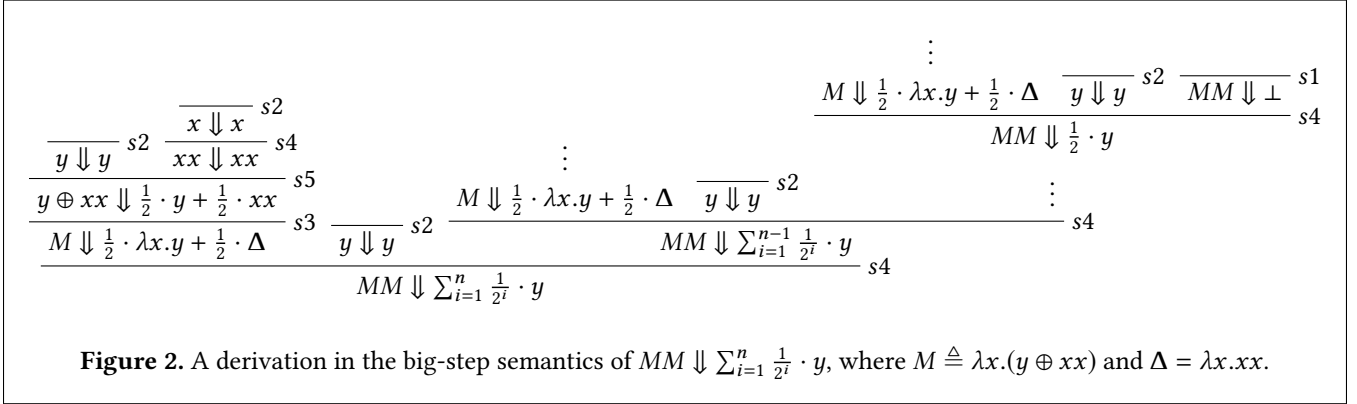
## 2.2 The Head Spine Reduction is Equivalent to the Head Reduction

The rules in Figure 1 do not correspond to the standard head reduction of the  $\lambda$ -calculus, but implement a variant of it, called *head spine* reduction in [24]. Let us see the difference on a deterministic  $\lambda$ -term, e.g.  $M \triangleq (\lambda x.(\lambda y.x)y)z$ . The (small-step) head reduction first evaluates the outermost redex of  $M$ , getting  $(\lambda y.z)y$ , and then the latter term, terminating in the hnf  $z$ . The small-step reduction relation associated with Figure 1 is detailed in [8], but just the inspection of the rule s4 may convince the reader that this reduction will first evaluate the body of  $\lambda x.(\lambda y.x)y$  to an hnf, so getting the term  $\lambda x.x$  and then it fires the application of the latter to the variable  $z$ , getting  $z$ . The two reduction sequences are different but they give the same result (and actually with the same number of reduction steps). We prove in Theorem 2 that this is always the case, even in a probabilistic setting<sup>1</sup>.

We decided to consider the head spine reduction as it has a compact big-step presentation and it fits perfectly into the  $\Lambda_{\oplus}$ -Markov chain definition (see Remark 1). Also, it allows us for a simpler proof of the soundness property (Theorem 16). On the other side, the equivalence with the head reduction makes available the separation property (here Theorem 18) that Leventis proved for the head reduction strategy [19] and that will play a crucial role for completeness.

In order to state Theorem 2 let us define precisely the probabilistic head reduction operational semantics  $\mathcal{H}^{\infty}$ . Following [10, 11], we define it as the limit of the small-step

<sup>1</sup>To the best of our knowledge, this result does not appear in the earlier literature, even in the deterministic case.



**Figure 2.** A derivation in the big-step semantics of  $MM \Downarrow \sum_{i=1}^n \frac{1}{2^i} \cdot y$ , where  $M \triangleq \lambda x.(y \oplus xx)$  and  $\Delta = \lambda x.xx$ .

transition matrix  $\mathcal{H}$  over  $\Lambda_{\oplus}$ . For  $M, N \in \Lambda_{\oplus}$  we set:

$$\mathcal{H}(M, N) \triangleq \begin{cases} 1 & \text{if } M = \mathcal{E}[(\lambda y.P)Q] \text{ and } N = \mathcal{E}[P[Q/y]] \\ \frac{1}{2} & \text{if } M = \mathcal{E}[P_1 \oplus P_2], P_1 \neq P_2 \text{ and } N = \mathcal{E}[P_i] \\ 1 & \text{if } M = \mathcal{E}[P \oplus P], \text{ and } N = \mathcal{E}[P] \\ 1 & \text{if } M = N \in \text{HNF} \\ 0 & \text{otherwise} \end{cases}$$

where  $\mathcal{E}$  is a *head context*, i.e. a special one-hole context of the form  $\lambda x_1 \dots x_n. [\cdot] L_1 \dots L_m$ , with  $n, m \geq 0$  and  $L_i \in \Lambda_{\oplus}$  (we slightly anticipate from Subsection 2.3). The matrix  $\mathcal{H}$  is stochastic, i.e. for any  $M, \sum_{N \in \Lambda_{\oplus}} \mathcal{H}(M, N) = 1$ .

Intuitively, the entry  $\mathcal{H}^n(M, N)$  of the  $n$ -th power  $\mathcal{H}^n$  of the matrix  $\mathcal{H}$  describes the probability that  $M$  reduces to  $N$  after at most  $n$  steps of head reduction. Notice that the head normal forms are absorbing states of the process, so for  $M \in \Lambda_{\oplus}$  and  $H \in \text{HNF}$ , the sequence  $(\mathcal{H}^n(M, H))_{n \in \mathbb{N}}$  is monotone increasing and bounded by 1, so it converges. We define its limit by:

$$\mathcal{H}^{\infty}(M, H) \triangleq \sup_{n \in \mathbb{N}} \mathcal{H}^n(M, H) \quad \forall M \in \Lambda_{\oplus}, \forall H \in \text{HNF}. \quad (4)$$

This quantity gives the total probability of  $M$  to reduce to the hnf  $H$  in an arbitrary number of head reduction steps.

**Theorem 2.** *Let  $M \in \Lambda_{\oplus}, H \in \text{HNF}$ , we have:*

$$\llbracket M \rrbracket(H) = \mathcal{H}^{\infty}(M, H).$$

Hence, our definition of  $\llbracket \cdot \rrbracket$  is just another way of presenting the operational semantics generated by the head reduction and discussed, for example, in [11, 19, 20]

### 2.3 Contextual Equivalence

A *context* of  $\Lambda_{\oplus}$  is a term containing a unique occurrence of a special variable  $[\cdot]$ , called the hole. This is generated by:

$$C := [\cdot] \mid \lambda x.C \mid CM \mid MC \mid C \oplus M \mid M \oplus C \quad (5)$$

We denote by  $\text{CA}_{\oplus}$  the set of all contexts. Given  $C \in \text{CA}_{\oplus}$  and  $M \in \Lambda_{\oplus}$ , then  $C[M]$  denotes a term obtained by substituting the unique hole in  $C$  with  $M$  allowing the possible capture of free variables of  $M$ .

The typical observation in  $\Lambda_{\oplus}$  is the probability of converging to a value. Since values are hnfs, *contextual preorder*  $\leq_{\text{cxt}}$  and *contextual equivalence*  $=_{\text{cxt}}$  can be defined as follows:

$$M \leq_{\text{cxt}} N \text{ iff } \forall C \in \text{CA}_{\oplus}, \sum \llbracket C[M] \rrbracket \leq \sum \llbracket C[N] \rrbracket, \quad (6)$$

$$M =_{\text{cxt}} N \text{ iff } \forall C \in \text{CA}_{\oplus}, \sum \llbracket C[M] \rrbracket = \sum \llbracket C[N] \rrbracket. \quad (7)$$

Note that  $M =_{\text{cxt}} N$  if and only if  $M \leq_{\text{cxt}} N$  and  $N \leq_{\text{cxt}} M$ .

**Example 5.** Consider the terms  $M \triangleq \lambda xyz.z(x \oplus y)$  and  $N \triangleq \lambda xyz.(zx \oplus zy)$ . They can be discriminated by the context  $C \triangleq [\cdot] \Omega \mathbf{I} \Delta$ , where  $\Omega, \mathbf{I}$ , and  $\Delta$  are as in Example 1. In Figure 3 we show that  $\sum \llbracket C[M] \rrbracket = \frac{1}{4}$  and  $\sum \llbracket C[N] \rrbracket = \frac{1}{2}$ .

Contexts enjoy the following monotonicity property:

**Lemma 3.** *Let  $M, N \in \Lambda_{\oplus}$ . If  $\llbracket M \rrbracket \leq_{\mathfrak{D}} \llbracket N \rrbracket$  then  $\forall C \in \text{CA}_{\oplus} \llbracket C[M] \rrbracket \leq_{\mathfrak{D}} \llbracket C[N] \rrbracket$ .*

An immediate consequence of Lemma 3 is the soundness of the operational semantics:

**Proposition 4.** *Let  $M, N \in \Lambda_{\oplus}$ : if  $\llbracket M \rrbracket \leq_{\mathfrak{D}} \llbracket N \rrbracket$  (resp.  $\llbracket M \rrbracket = \llbracket N \rrbracket$ ) then  $M \leq_{\text{cxt}} N$  (resp.  $M =_{\text{cxt}} N$ ).*

Thanks to Proposition 4, one can prove that quite different terms are indeed contextually equivalent, as the following example shows:

**Example 6.** The term  $MM$  in Example 4 and  $y$  are contextually equivalent, i.e.  $MM =_{\text{cxt}} y$ , since  $\llbracket MM \rrbracket = y$ .

However, not all contextually equivalent terms have the same semantics: the term  $\lambda x.x$  and its  $\eta$ -expansion  $\lambda xy.xy$  are contextually equivalent but  $\llbracket \lambda x.x \rrbracket = \lambda x.x \neq \lambda xy.xy = \llbracket \lambda xy.xy \rrbracket$ .

Proving contextual equivalence might be rather difficult since its definition quantifies over the set of *all* contexts. Fortunately, various other tools can be deployed to show the equivalence of terms. An example is bisimilarity, we shall discuss in the next subsection. Checking that two terms are bisimilar requires the *existence* of a particular relation, called “bisimulation”. Proving that bisimilarity and contextual equivalence actually coincide would imply that the latter

$$\begin{array}{c}
\begin{array}{c}
\vdots \\
\frac{\overline{\Omega \Downarrow \perp} \text{ s1} \quad \overline{\mathbf{I} \Downarrow \mathbf{I}} \text{ s5} \quad \vdots}{\text{hid} \Downarrow \frac{1}{2} \cdot \mathbf{I}} \text{ s4} \\
\vdots \\
\frac{\Delta \Downarrow \Delta}{\text{hid hid} \Downarrow \frac{1}{4} \cdot \mathbf{I}} \text{ s4} \\
\vdots \\
\frac{\lambda z.z \text{ hid} \Downarrow \lambda z.z \text{ hid}}{\Delta \text{ hid} \Downarrow \frac{1}{4} \cdot \mathbf{I}} \text{ s4} \\
\vdots \\
\frac{\lambda yz.z(\Omega \oplus y) \Downarrow \lambda yz.z(\Omega \oplus y)}{(\lambda yz.z(\Omega \oplus y))\mathbf{I}\Delta \Downarrow \frac{1}{4} \cdot \mathbf{I}} \text{ s4} \\
\vdots \\
\frac{M \Downarrow M}{M\Omega\mathbf{I}\Delta \Downarrow \frac{1}{4} \cdot \mathbf{I}} \text{ s4}
\end{array} \\
\\
\begin{array}{c}
\vdots \\
\frac{\overline{z \Downarrow z} \text{ s2} \quad \overline{z \Downarrow z} \text{ s2}}{zx \Downarrow zx} \text{ s4} \quad \frac{\overline{z \Downarrow z} \text{ s2} \quad \overline{z \Downarrow z} \text{ s2}}{zy \Downarrow zy} \text{ s4} \\
\vdots \\
\frac{\overline{zx \oplus zy} \Downarrow \frac{1}{2} \cdot zx + \frac{1}{2} \cdot zy}{N \Downarrow \frac{1}{2} \cdot \lambda xyz.zx + \frac{1}{2} \cdot \lambda xyz.zy} \text{ s3} \\
\vdots \\
\frac{\lambda yz.z\mathbf{I} \Downarrow \lambda yz.z\mathbf{I}}{(\lambda yz.z\mathbf{I})\Delta \Downarrow \mathbf{I}} \text{ s4} \\
\vdots \\
\frac{\lambda yz.z\Omega \text{I}\Delta \Downarrow \perp}{N\Omega\mathbf{I}\Delta \Downarrow \frac{1}{2} \cdot \mathbf{I}} \text{ s1} \quad \frac{\lambda yz.zy \Downarrow \lambda yz.zy}{(\lambda yz.zy)\mathbf{I}\Delta \Downarrow \mathbf{I}} \text{ s4}
\end{array}
\end{array}$$

**Figure 3.** The derivations in the big-step semantics of  $M\Omega\mathbf{I}\Delta \Downarrow \frac{1}{4} \cdot \mathbf{I}$  and  $N\Omega\mathbf{I}\Delta \Downarrow \frac{1}{2} \cdot \mathbf{I}$ , where  $M \triangleq \lambda xyz.z(x \oplus y)$ ,  $N \triangleq \lambda xyz.(zx \oplus zy)$ ,  $\Delta = \lambda x.xx$ , and  $\text{hid} = \Omega \oplus \mathbf{I}$ . The double inference line means multiple applications of the same rule.

can be established using the much more tractable operational techniques coming from bisimilarity.

## 2.4 Probabilistic Applicative (Bi)Similarity

We recall here the main definitions and basic properties given in [16], as these do not depend on a specific operational semantics. First, we introduce labelled Markov chains and its associated probabilistic (bi)similarity [18]. Then, we apply these notions to the operational semantics of  $\Lambda_{\oplus}$ , getting the probabilistic applicative (bi)similarity.

A *labelled Markov chain* is a triple  $\mathcal{M} = (\mathcal{S}, \mathcal{L}, \mathcal{P})$ , where  $\mathcal{S}$  is a countable set of states,  $\mathcal{L}$  is a set of labels (actions) and  $\mathcal{P}$  is a transition probability matrix, i.e. a function  $\mathcal{P} : \mathcal{S} \times \mathcal{L} \times \mathcal{S} \rightarrow [0, 1]$  satisfying the following condition:

$$\forall s \in \mathcal{S}, \forall l \in \mathcal{L} : \sum_{t \in \mathcal{S}} \mathcal{P}(s, l, t) \leq 1 .$$

If  $X \subseteq \mathcal{S}$ , we let  $\mathcal{P}(s, l, X)$  denote  $\sum_{t \in X} \mathcal{P}(s, l, t)$ .

A *probabilistic simulation*  $\mathcal{R}$  in  $\mathcal{M}$  is a preorder over  $\mathcal{S}$  s.t.:

$$\forall (s, t) \in \mathcal{R}, \forall X \subseteq \mathcal{S}, \forall l \in \mathcal{L}, \mathcal{P}(s, l, X) \leq \mathcal{P}(t, l, \mathcal{R}(X)) \quad (8)$$

A *probabilistic bisimulation*  $\mathcal{R}$  is an equivalence over  $\mathcal{S}$  s.t.:

$$\forall (s, t) \in \mathcal{R}, \forall E \in \mathcal{S}/\mathcal{R}, \forall l \in \mathcal{L}, \mathcal{P}(s, l, E) = \mathcal{P}(t, l, E) \quad (9)$$

The *probabilistic similarity*  $\lesssim$  (resp. *probabilistic bisimilarity*  $\sim$ ) is the union of all probabilistic simulations (resp. bisimulations). For all  $s, t \in \mathcal{S}$ :

$$s \leq t \Leftrightarrow \exists \mathcal{R} \text{ probabilistic simulation s.t. } s \mathcal{R} t, \quad (10)$$

$$s \sim t \Leftrightarrow \exists \mathcal{R} \text{ probabilistic bisimulation s.t. } s \mathcal{R} t. \quad (11)$$

**Proposition 5** (e.g. [16]). *The relation  $\lesssim$  (resp.  $\sim$ ) is a probabilistic simulation (resp. bisimulation). Moreover, it holds that  $\sim = \lesssim \cap \lesssim^{op}$ .*

In order to apply these notions to  $\Lambda_{\oplus}$ , we need to preset its operational semantics as a labelled Markov chain (Definition 1). Intuitively, terms are seen as states, while labels are of two kinds: one can either *evaluate* a term (this kind of transition will be labelled by  $\tau$ ), obtaining a distribution of hnf's, or *apply* a hnf to a term  $M$  (this kind of transition will be labelled by  $M$ ). For technical reasons, it is useful to consider only closed terms and to consider for each closed hnf  $H = \lambda x.H'$  two distinct representations, depending on the way we consider it: either as a term or properly as a normal form, and in the latter case we indicate it as  $\overline{H} \triangleq vx.H'$  to stress the difference. Consequently, we define  $\overline{\text{HNF}}$  as the set of all “distinguished” closed hnf's, namely  $\{\overline{H} \mid H \in \text{HNF}^0\}$ . More in general, if  $X \subseteq \text{HNF}^0$ , we define  $\overline{X} \triangleq \{\overline{H} \mid H \in X\}$ .

**Definition 1.** The  $\Lambda_{\oplus}$ -Markov chain is the triple  $(\Lambda_{\oplus}^0 \uplus \overline{\text{HNF}}, \Lambda_{\oplus}^0 \uplus \{\tau\}, \mathcal{P}_{\oplus})$ , where the set of states is the disjoint

union of the set of closed terms and the set of “distinguished” closed hnf’s, labels (actions) are either closed terms or the  $\tau$  action, and the transition probability matrix  $\mathcal{P}_\oplus$  is defined in the following way:

(i) for every closed term  $M$  and distinguished hnf  $vx.H$ :

$$\mathcal{P}_\oplus(M, \tau, vx.H) \triangleq \llbracket M \rrbracket(\lambda x.H) ,$$

(ii) for every closed term  $M$  and distinguished hnf  $vx.H$ :

$$\mathcal{P}_\oplus(vx.H, M, H[M/x]) \triangleq 1 ,$$

(iii) in all other cases,  $\mathcal{P}_\oplus$  returns 0.

A *probabilistic applicative (bi)simulation* is a probabilistic (bi)simulation of the  $\Lambda_\oplus$ -Markov chain. The *probabilistic applicative similarity*, PAS for short, and the *probabilistic applicative bisimilarity*, PAB for short, are defined as in (10) and (11). From now on, with  $\lesssim$  (resp.  $\sim$ ) we mean probabilistic applicative similarity (resp. bisimilarity).

*Remark 1.* In the  $\Lambda_\oplus$ -Markov chain, a term  $M$  can be thought at the head of a (potentially infinite) stack of applications, where at each time we first evaluate the head of the stack until we reach a head normal form  $H$  (point (i)), and then we apply  $H$  to the next term of the stack (point (ii)). This is exactly the behaviour of the head spine reduction on an application  $MN_1 \dots N_n$ . Lemma 15 formalizes these intuitions.

The notions of PAS and PAB are defined on closed terms. We extend them to open terms  $M, N \in \Lambda_\oplus^{\{x_1, \dots, x_n\}}$ , by:

$$M \lesssim N \Leftrightarrow \lambda x_1 \dots x_n. M \lesssim \lambda x_1 \dots x_n. N , \quad (12)$$

$$M \sim N \Leftrightarrow \lambda x_1 \dots x_n. M \sim \lambda x_1 \dots x_n. N . \quad (13)$$

One can notice that the order of the abstractions in the term closure does not affect the obtained relation.

The following proposition is analogous to Proposition 4, stating the soundness of the operational semantics with respect to both PAS and PAB.

**Proposition 6.** *Let  $M, N \in \Lambda_\oplus$ : if  $\llbracket M \rrbracket \leq_{\mathfrak{D}} \llbracket N \rrbracket$  (resp.  $\llbracket M \rrbracket = \llbracket N \rrbracket$ ) then  $M \lesssim N$  (resp.  $M \sim N$ ).*

*Proof.* We prove only the inequality soundness, as the equality one is an immediate consequence by Proposition 5. Moreover, the proof is for closed terms, as the case of open terms follows from Proposition 1.(2).

Let  $M, N \in \Lambda_\oplus^0$  be such that  $\llbracket M \rrbracket \leq_{\mathfrak{D}} \llbracket N \rrbracket$ , and consider the relation  $\mathcal{R} = \{(P, Q) \in \Lambda_\oplus^0 \times \Lambda_\oplus^0 \mid \llbracket P \rrbracket \leq_{\mathfrak{D}} \llbracket Q \rrbracket\} \cup \{(vx.H, vx.H) \in \overline{\text{HNF}} \times \overline{\text{HNF}}\}$ . If we show that  $\mathcal{R}$  is a PAS, then  $\mathcal{R} \subseteq \lesssim$ , and hence  $M \lesssim N$ . Clearly,  $\mathcal{R}$  is a preorder. Now, let  $(P, Q), (vx.H, vx.H) \in \mathcal{R}$ , and let  $X \subseteq \Lambda_\oplus^0 \cup \overline{\text{HNF}}$ . It is straightforward that  $\mathcal{P}_\oplus(vx.H, l, X) \leq \mathcal{P}_\oplus(vx.H, l, \mathcal{R}(X))$ , for all  $l \in \Lambda_\oplus^0 \cup \{\tau\}$ . Moreover, for all  $F \in \Lambda_\oplus^0$  we have  $0 = \mathcal{P}_\oplus(P, F, X) \leq \mathcal{P}_\oplus(Q, F, \mathcal{R}(X))$ . Last:

$$\begin{aligned} \mathcal{P}_\oplus(P, \tau, X) &= \sum_{vx.H \in X} \mathcal{P}_\oplus(P, \tau, vx.H) = \llbracket P \rrbracket(X \cap \overline{\text{HNF}}) \\ &\leq \llbracket Q \rrbracket(X \cap \overline{\text{HNF}}) = \mathcal{P}_\oplus(Q, \tau, \mathcal{R}(X)). \end{aligned}$$

Hence, for all  $l \in \Lambda_\oplus^0 \cup \{\tau\}$  and  $X \subseteq \Lambda_\oplus^0 \cup \overline{\text{HNF}}$ , we have  $\mathcal{P}_\oplus(P, l, X) \leq \mathcal{P}_\oplus(Q, l, \mathcal{R}(X))$ .  $\square$

**Example 7.** Let us show that  $I \sim \lambda xy.xy$  so that, from the soundness (Theorem 16), one can infer  $I =_{\text{cxt}} \lambda xy.xy$ .

Let us define  $\mathcal{R}_1 \triangleq \{(I, \lambda xy.xy), (\lambda xy.xy, I)\}$ , as well as  $\mathcal{R}_2 \triangleq \{(\tilde{I}, vx.\lambda y.xy), (vx.\lambda y.xy, \tilde{I})\}$  and  $\mathcal{R}_3 \triangleq \sim$ . Let  $\mathcal{R} \triangleq (\mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3)^*$ . Since  $\mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3$  is a symmetric relation, then its reflexive and transitive closure  $\mathcal{R} \triangleq (\mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3)^*$  is an equivalence. Let us prove that it is a probabilistic bisimulation.

We have to prove that  $\mathcal{P}_\oplus(M, l, E) = \mathcal{P}_\oplus(N, l, E), \forall (M, N) \in \mathcal{R}, \forall E \in (\Lambda_\oplus^0 \cup \overline{\text{HNF}})/\mathcal{R}, \forall l \in \Lambda_\oplus^0 \cup \{\tau\}$ . Notice that, if this holds for  $(M, N) \in (\mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3)$ , then we are done. Indeed, suppose  $(M, N) \in \mathcal{R}$ . Then there exists  $n \geq 0$  and  $P_0, \dots, P_n \in \Lambda_\oplus^0 \cup \overline{\text{HNF}}$  such that  $P_0 = M, P_n = N$  and  $P_{i-1} \mathcal{R}_j P_i$  for every  $1 \leq i \leq n$ , where  $1 \leq j_i \leq 3$ . Hence, we have  $\mathcal{P}_\oplus(M, l, E) = \mathcal{P}_\oplus(P_0, l, E) = \dots = \mathcal{P}_\oplus(P_n, l, E) = \mathcal{P}_\oplus(N, l, E), \forall E \in (\Lambda_\oplus^0 \cup \overline{\text{HNF}})/\mathcal{R}, \forall l \in \Lambda_\oplus^0 \cup \{\tau\}$ .

Let us now show the case  $(M, N) \in (\mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3)$ . If  $(M, N) \in \mathcal{R}_3$  we just apply Proposition 5. Otherwise, it suffices to consider  $(I, \lambda xy.xy)$  and  $(\tilde{I}, vx.\lambda y.xy)$ . Recall that, by Definition 1,  $\mathcal{P}_\oplus(M, N, E) = 0$  and  $\mathcal{P}_\oplus(\tilde{I}, \tau, E) = 0$ , for all  $M, N \in \Lambda_\oplus^0, \tilde{I} \in \overline{\text{HNF}}$  and  $E \in (\Lambda_\oplus^0 \cup \overline{\text{HNF}})/\mathcal{R}$ . On the one hand, since  $(\tilde{I}, vx.\lambda y.xy) \in \mathcal{R}$ , we have  $\tilde{I} \in E$  if and only if  $vx.\lambda y.xy \in E$ , for all  $E \in (\Lambda_\oplus^0 \cup \overline{\text{HNF}})/\mathcal{R}$ . This implies  $\mathcal{P}_\oplus(I, \tau, E) = \mathcal{P}_\oplus(\lambda xy.xy, \tau, E)$ , for all  $E \in (\Lambda_\oplus^0 \cup \overline{\text{HNF}})/\mathcal{R}$ . On the other hand, since terms are considered modulo renaming of bound variables, by Proposition 1 we have  $\llbracket N \rrbracket = \llbracket \lambda y.Ny \rrbracket$ , for all  $N \in \Lambda_\oplus^0$  (notice that this equality may fail if  $N$  has free variables). By Proposition 6,  $N \sim \lambda y.Ny$ , and hence  $N \in E$  if and only if  $\lambda y.Ny \in E$ , for all  $E \in (\Lambda_\oplus^0 \cup \overline{\text{HNF}})/\mathcal{R}$ . This implies  $\mathcal{P}_\oplus(\tilde{I}, N, E) = \mathcal{P}_\oplus(vx.\lambda y.xy, N, E)$ , for all  $N \in \Lambda_\oplus^0$  and for all  $E \in (\Lambda_\oplus^0 \cup \overline{\text{HNF}})/\mathcal{R}$ .

**Example 8.** We show that the terms  $M \triangleq \lambda xyz.z(x \oplus y)$  and  $N \triangleq \lambda xyz.(zx \oplus zy)$  in Example 5 are not bisimilar. Indeed, suppose for the sake of contradiction that a probabilistic bisimulation  $\mathcal{R}$  such that  $(M, N) \in \mathcal{R}$  exists. By definition  $\mathcal{R}$  is an equivalence relation. Let  $E \in (\Lambda_\oplus^0 \cup \overline{\text{HNF}})/\mathcal{R}$  be such that  $vx.\lambda yz.z(x \oplus y) \in E$ . Then it must be that  $\mathcal{P}_\oplus(M, \tau, E) = 1 = \mathcal{P}_\oplus(N, \tau, E)$ , and it follows that both  $vx.\lambda yz.zx$  and  $vx.\lambda yz.zy$  are in  $E$ , so that  $(vx.\lambda yz.z(x \oplus y), vx.\lambda yz.zx) \in \mathcal{R}$ . Then it must be that  $\mathcal{P}_\oplus(vx.\lambda yz.z(x \oplus y), \Omega, E_1) = 1 = \mathcal{P}_\oplus(vx.\lambda yz.zx, \Omega, E_1)$ , for some  $E_1 \in (\Lambda_\oplus^0 \cup \overline{\text{HNF}})/\mathcal{R}$  containing both  $\lambda yz.z(\Omega \oplus y)$  and  $\lambda yz.z\Omega \in E_1$ , which implies  $(\lambda yz.z(\Omega \oplus y), \lambda yz.z\Omega) \in \mathcal{R}$ . By a similar reasoning, we get that  $\mathcal{R}$  contains the pairs  $(vy.\lambda z.z(\Omega \oplus y), vy.\lambda z.z\Omega), (\lambda z.z(\Omega \oplus I), \lambda z.z\Omega)$ , and  $(vz.z(\Omega \oplus I), vz.z\Omega)$ . Now, let  $E_2$  be an equivalence class containing  $\Omega \oplus I$ . From  $\mathcal{P}_\oplus(vz.z(\Omega \oplus I), I, E_2) = 1 = \mathcal{P}_\oplus(vz.z\Omega, I, E_2)$  we get that  $\Omega \in E_2$ , i.e.  $(\Omega \oplus I, \Omega) \in \mathcal{R}$ . Finally, if  $E_3$  is an equivalence class such that  $vx.x \in E_3$ , then  $\mathcal{P}_\oplus(\Omega \oplus I, \tau, E_3) = \frac{1}{2} = \mathcal{P}_\oplus(\Omega, \tau, E_3)$ . This is

a contradiction, since  $\mathcal{P}_\oplus(\Omega, \tau, E_3) = 0$ . Therefore, the terms  $M$  and  $N$  are not bisimilar.

### 3 Soundness

A fundamental technique to establish the soundness of applicative (bi)similarity is based on *Howe's lifting* [14]. This method shows that applicative bisimilarity is a *congruence*, i.e. an equivalence relation that respects the structure of terms, which is the hard part in the soundness proof. This technique has been used in e.g. [6, 16] for, respectively, the lazy cbn and cbv semantics of  $\Lambda_\oplus$ . We consider here a different approach. Following the reasoning by Abramsky and Ong [2], we shall first prove that  $\lesssim$  is included in  $\leq_{\text{app}}$  (Lemma 15), which requires a technical Key Lemma (Lemma 14) specific to the probabilistic framework and then we conclude by applying a Context Lemma (Lemma 9). The latter result says that the computational behaviour of the contextual semantics is *functional*. This property has also been called *operational extensionality* in Bloom [4]. Milner [21] proved a similar result in the case of simply typed combinatory algebra. To the best of our knowledge, the Context Lemma lacks a corresponding formulation in the probabilistic  $\lambda$ -calculus  $\Lambda_\oplus$ , so we prove it in the following subsection.

#### 3.1 Context Lemma

The Context Lemma states that only the subset of applicative contexts “really matter” in establishing contextual equivalence. We define an *applicative context* as a context  $\mathcal{E} \in \mathcal{C}\Lambda_\oplus$  of the form  $(\lambda x_1 \dots \lambda x_n. [\cdot])P_1 \dots P_m$ , where  $n, m \in \mathbb{N}$  and  $P_1 \dots P_m \in \Lambda_\oplus^0$ . We denote by  $\mathbb{A}\Lambda_\oplus$  the set of all applicative contexts.

The *applicative contextual preorder*  $\leq_{\text{app}}$  (resp. *applicative contextual equivalence*  $=_{\text{app}}$ ) is defined by restricting the quantifier  $\forall C$  to the subset  $\mathbb{A}\Lambda_\oplus$  of  $\mathcal{C}\Lambda_\oplus$  in the contextual preorder (resp. equivalence) definition (6) (resp. (7)).

**Lemma 7.** *Let  $M, N \in \Lambda_\oplus^{\Gamma \cup \{x\}}$ . Then:*

- (1) *If  $M \leq_{\text{app}} N$  then  $\lambda x.M \leq_{\text{app}} \lambda x.N$ .*
- (2) *If  $\lambda x.M \leq_{\text{cxt}} \lambda x.N$  then  $M \leq_{\text{cxt}} N$ .*
- (3) *If  $M \leq_{\text{cxt}} N$  then, for all  $L \in \Lambda_\oplus$ ,  $ML \leq_{\text{cxt}} NL$ .*

In order to prove the Context Lemma more easily, we shall adopt a slightly more general notion of context, allowing multiple holes. A *generalized context* of  $\Lambda_\oplus$  is a term containing holes  $[\cdot]$ , generated by the following grammar:

$$C := x \mid [\cdot] \mid \lambda x.C \mid CC \mid C \oplus C . \quad (14)$$

We denote by  $\mathbb{G}\Lambda_\oplus$  the set of all generalized contexts. If  $C \in \mathbb{G}\Lambda_\oplus$  and  $M \in \Lambda_\oplus$ , then  $C[M]$  denotes the term obtained by substituting every hole in  $C$  with  $M$  allowing the possible capture of free variables of  $M$ .

**Lemma 8.** *Let  $M, N \in \Lambda_\oplus^0$  be such that  $M \leq_{\text{app}} N$ . Then  $\sum \llbracket C[M] \rrbracket \leq \sum \llbracket C[N] \rrbracket$ , for all  $C \in \mathbb{G}\Lambda_\oplus$ .*

*Proof (sketch).* By Theorem 2 it is enough to show that, for all  $n \in \mathbb{N}$  and for all generalized contexts  $C \in \mathbb{G}\Lambda_\oplus$ :

$$\sum_{H \in \text{HNF}} \mathcal{H}^n(C[M], H) \leq \sum_{H \in \text{HNF}} \mathcal{H}^n(C[N], H) . \quad (15)$$

The proof is by induction on  $(n, |C|)$ , where  $|C|$  is the size of  $C \in \mathbb{G}\Lambda_\oplus$ , i.e. the number of nodes in the syntax tree of  $C$ . Since  $C$  must be of the form  $C_0 C_1 \dots C_k$ , for some  $k \in \mathbb{N}$ , we proceed by case analysis, looking at the structure of  $C_0$ .  $\square$

**Lemma 9** (Context Lemma). *Let  $M, N \in \Lambda_\oplus$ . Then:*

- (1)  *$M \leq_{\text{cxt}} N$  if and only if  $M \leq_{\text{app}} N$ .*
- (2)  *$M =_{\text{cxt}} N$  if and only if  $M =_{\text{app}} N$ .*

*Proof.* Point (2) follows directly from point (1). Lemma 8 gives us point (1) for  $M, N \in \Lambda_\oplus^0$ . We extend it to open terms by applying Lemma 7.(1) and Lemma 7.(2).  $\square$

#### 3.2 The Soundness Theorem

We start with some preliminary lemmas.

**Lemma 10.** *Let  $H, H' \in \text{HNF}^{\{x\}}$ . Then, the following are equivalent statements:*

- (1)  $\lambda x.H \lesssim \lambda x.H'$ ,
- (2)  $\nu x.H \lesssim \nu x.H'$ ,
- (3)  $\forall P \in \Lambda_\oplus^0, H[P/x] \lesssim H'[P/x]$  .

*Proof (sketch).* The implication (1)  $\Rightarrow$  (2)  $\Rightarrow$  (3) is by definition and by Proposition 5. To prove (3)  $\Rightarrow$  (2), it suffices to show that the relation  $\mathcal{R} \triangleq \{(\nu x.H, \nu x.H') \in \overline{\text{HNF}} \times \overline{\text{HNF}} \mid \forall P \in \Lambda_\oplus^0, H[P/x] \lesssim H'[P/x]\} \cup \lesssim$  is a probabilistic applicative simulation. Similarly, (2)  $\Rightarrow$  (1) holds by showing that the relation  $\mathcal{R} \triangleq \{(\lambda x.H, \lambda x.H') \in \overline{\text{HNF}} \times \overline{\text{HNF}} \mid \nu x.H \lesssim \nu x.H'\} \cup \lesssim$  is a probabilistic applicative simulation.  $\square$

Let us recall that, given  $X \subseteq \text{HNF}$ ,  $\lesssim(X)$  denotes the image of  $X$  under  $\lesssim$ . Moreover, given  $X \subseteq \text{HNF}^{\{x\}}$ ,  $\nu x.\lesssim(X)$  denotes the set of distinguished hnfs  $\{\nu x.H \mid H \in \lesssim(X)\}$ , while  $\lambda x.\lesssim(X)$  denotes the set of terms  $\{\lambda x.M \mid M \in \lesssim(X)\}$ .

**Lemma 11.** *Let  $X \subseteq \text{HNF}^{\{x\}}$ . We have:*

$$\begin{aligned} \lesssim(\lambda x.X) \cap \text{HNF}^0 &= \lambda x.\lesssim(X) \cap \text{HNF}^0, \\ \lesssim(\nu x.X) &= \nu x.\lesssim(X) . \end{aligned}$$

**Lemma 12.** *Let  $M, N \in \Lambda_\oplus^0$ . For all  $X \subseteq \text{HNF}^0$ ,  $\llbracket M \rrbracket(X) \leq \llbracket N \rrbracket(\lesssim(X))$  if and only if  $M \lesssim N$ .*

The forthcoming Lemma 14 describes the applicative behaviour of  $\lesssim$  and it requires an auxiliary result about the so-called “probabilistic assignments”. Probabilistic assignments were first introduced in this setting by [16] to prove the soundness of PAS in the lazy cbn.

**Definition 2** (Probabilistic assignments). A *probabilistic assignment* is defined as a pair  $(\{p_i\}_{1 \leq i \leq n}, \{r_I\}_{I \subseteq \{1, \dots, n\}})$ , with all  $p_i, r_I$  in  $[0, 1]$ , such that, for all  $I \subseteq \{1, \dots, n\}$ :

$$\sum_{i \in I} p_i \leq \sum_{\substack{J \subseteq \{1, \dots, n\} \\ \text{s.t. } J \cap I \neq \emptyset}} r_J . \quad (16)$$



**Lemma 13** ([16]). *Let  $(\{p_i\}_{1 \leq i \leq n}, \{r_I\}_{I \subseteq \{1, \dots, n\}})$  be a probabilistic assignment. Then for every  $I \subseteq \{1, \dots, n\}$  and for every  $k \in I$  there is  $s_{k,I} \in [0, 1]$  such that:*

$$(1) \forall j \in \{1, \dots, n\}, p_j \leq \sum_{\substack{J \subseteq \{1, \dots, n\} \\ s.t. j \in J}} s_{j,J} \cdot r_J.$$

$$(2) \forall J \subseteq \{1, \dots, n\}, \sum_{\substack{j \in \{1, \dots, n\} \\ s.t. j \in J}} s_{j,J} \leq 1.$$

Following essentially the same ideas of [16], we shall use the above property to decompose and recombine distributions in the proof of the following lemma.

**Lemma 14** (Key Lemma). *Let  $M, N \in \Lambda_{\oplus}^0$ . If  $M \lesssim N$  then, for all  $P \in \Lambda_{\oplus}^0$ ,  $MP \lesssim NP$ .*

*Proof (sketch).* By Lemma 12 it suffices to prove that, for all  $X \subseteq \text{HNF}^0$  and for all  $\mathcal{D} \in \mathfrak{D}(\text{HNF})$  such that  $MP \Downarrow \mathcal{D}$ , it holds that  $\mathcal{D}(X) \leq \llbracket NP \rrbracket(\lesssim(X))$ . The non-trivial case is when the last rule of  $MP \Downarrow \mathcal{D}$  is  $s_4$ , i.e. when:

$$\mathcal{D}(X) = \sum_{\lambda x.H \in \text{supp}(\mathcal{E})} \mathcal{E}(\lambda x.H) \cdot \mathcal{F}_{H,P}(X) \quad (17)$$

for  $M \Downarrow \mathcal{E}$  and  $H[P/x] \Downarrow \mathcal{F}_{H,P}$ . Notice that  $\text{supp}(\mathcal{E})$  is finite, say  $\text{supp}(\mathcal{E}) = \{\lambda z.H_1, \dots, \lambda z.H_n\}$ .

Proposition 1 gives us:

$$\llbracket NP \rrbracket(\lesssim(X)) = \sum_{\lambda x.H} \llbracket N \rrbracket(\lambda x.H) \cdot \llbracket H[P/x] \rrbracket(\lesssim(X)) \quad (18)$$

One would be then tempted to compare the sums (17) and (18) term by term. In fact, by hypothesis we know that for every  $\lambda x.H$ ,  $\mathcal{E}(\lambda x.H) \leq \llbracket N \rrbracket(\lesssim\{\lambda x.H\})$ . This gives that every term  $\mathcal{E}(\lambda x.H) \cdot \mathcal{F}_{H,P}(X)$  of (17) is smaller than  $\sum_{\lambda x.H' \in \lesssim(\lambda x.H)} \llbracket N \rrbracket(\lambda x.H') \cdot \llbracket H[P/x] \rrbracket(\lesssim(X))$ . Unfortunately we cannot conclude, as different hnfs  $\lambda x.H$  do not always generate disjoint  $\lesssim(\lambda x.H)$  (e.g. think about  $\eta$ -equivalent hnfs), so that we cannot factor (18) according to  $\lesssim(\lambda x.H_1), \dots, \lesssim(\lambda x.H_n)$ . Here is where Lemma 13 on probabilistic assignments plays a role, permitting to “disentangle” the different quantities  $\llbracket N \rrbracket(\lesssim\{\lambda x.H_1\}), \dots, \llbracket N \rrbracket(\lesssim\{\lambda x.H_n\})$ . In fact, one can prove that for all  $\lambda z.H' \in \bigcup_{i \in I} \lesssim\{\lambda z.H_i\}$  (notice that, since  $N \in \Lambda_{\oplus}^0$ ,  $\llbracket N \rrbracket(\bigcup_{i \in I} \lesssim\{\lambda z.H_i\}) = \llbracket N \rrbracket(\bigcup_{i \in I} \lesssim\{\lambda z.H_i\} \cap \text{HNF}^0)$ ), we can apply Lemma 13 and get  $s_1^{H'}, \dots, s_n^{H'} \in [0, 1]$  such that:

$$(1) \forall i \leq n, \mathcal{E}(\lambda z.H_i) \leq \sum_{\lambda z.H' \in \lesssim(\lambda z.H_i)} s_i^{H'},$$

$$(2) \forall \lambda z.H' \in \bigcup_{i \in I} \lesssim(\lambda z.H_i), \llbracket N \rrbracket(\lambda z.H') \geq \sum_{i=1}^n s_i^{H'}.$$

From this, we have:

$$\mathcal{D}(X) \leq \sum_{i=1}^n \left( \sum_{\lambda z.H' \in \lesssim(\lambda z.H_i)} s_i^{H'} \right) \cdot \mathcal{F}_{H_i,P}(X)$$

$$\leq \sum_{i=1}^n \sum_{\lambda z.H' \in \lesssim(\lambda z.H_i)} s_i^{H'} \cdot \llbracket H'[P/z] \rrbracket(\lesssim(X))$$

$$\leq \sum_{\lambda z.H' \in \bigcup_{i=1}^n \lesssim(\lambda z.H_i)} \left( \sum_{i=1}^n s_i^{H'} \right) \cdot \llbracket H'[P/z] \rrbracket(\lesssim(X))$$

$$\leq \sum_{\lambda z.H'} \llbracket N \rrbracket(\lambda z.H') \cdot \llbracket H'[P/z] \rrbracket(\lesssim(X)) = \llbracket NP \rrbracket(\lesssim(X))$$

and hence  $\mathcal{D}(X) \leq \llbracket NP \rrbracket(\lesssim(X))$ .  $\square$

**Lemma 15.** *Let  $M, N \in \Lambda_{\oplus}^0$ . If  $M \lesssim N$  then  $M \leq_{\text{app}} N$ .*

*Proof.* We have to show that  $M \lesssim N$  implies  $\sum \llbracket MP_1 \dots P_n \rrbracket \leq \sum \llbracket NP_1 \dots P_n \rrbracket$ , for any sequence  $P_1, \dots, P_n \in \Lambda_{\oplus}^0$ . The proof is by induction on  $n$ , using Lemma 12 for the base case and Lemma 14 for the induction step.  $\square$

**Theorem 16** (Soundness). *Let  $M, N \in \Lambda_{\oplus}$ . Then:*

- (1)  $M \lesssim N$  implies  $M \leq_{\text{cxt}} N$ .
- (2)  $M \sim N$  implies  $M =_{\text{cxt}} N$ .

*Proof.* Point (2) follows from point (1) since it holds that  $\sim = \lesssim \cap \lesssim^{op}$  (Proposition 5) and  $=_{\text{cxt}}$  is  $\leq_{\text{cxt}} \cap (\leq_{\text{cxt}})^{op}$ . Concerning point (1), we first prove it for closed terms. So, let  $M, N \in \Lambda_{\oplus}^0$  be such that  $M \lesssim N$ . By Lemma 15, it holds that  $M \leq_{\text{app}} N$ . By Lemma 9, this implies  $M \leq_{\text{cxt}} N$ . Now, let  $M, N \in \Lambda_{\oplus}^{\{x_1, \dots, x_n\}}$  be such that  $M \lesssim N$ . From (12), we have that  $\lambda x_1 \dots x_n.M \lesssim \lambda x_1 \dots x_n.N$ . Because these are closed terms, we obtain  $\lambda x_1 \dots x_n.M \leq_{\text{cxt}} \lambda x_1 \dots x_n.N$ . By repeatedly applying Lemma 7.(2), we conclude  $M \leq_{\text{cxt}} N$ .  $\square$

## 4 Full Abstraction

We prove that PAB is complete, hence fully abstract (Theorem 23), while PAS is not, giving a counterexample to PAS completeness in Section 4.3.

As mentioned in the Introduction, the completeness property is usually achieved by transforming PAB into a testing semantics defined by Larsen and Skou [18], proven equivalent to probabilistic bisimulation by van Breugel et al. [25], and then showing that every test is definable by a context in the language, see e.g. [6, 15]. This reasoning is not so simple to implement in our setting, as the testing definability needs a kind of sampling primitive, which is not clear if representable in a call-by-name semantics (see the discussion in the Introduction).

Fortunately, we succeed in following a different path, based on Leventis’ Separation Theorem [19]. The idea is to prove that (a trivial extension of) the contextual equivalence is a probabilistic applicative bisimulation, hence contained in  $\sim$  by definition (Eq. (11)). Basically, this amounts to check that for any contextual equivalence class  $E$  of hnfs and any  $M =_{\text{cxt}} N$ , we have  $\llbracket M \rrbracket(E) = \llbracket N \rrbracket(E)$  (see Eq. (9)). How to prove it? We associate terms with a kind of infinitary, extensional normal forms, the so-called probabilistic Nakajima trees (Section 4.1). The Separation Theorem states that two terms  $M$  and  $N$  share the same Nakajima tree whenever they are contextually equivalent (Theorem 18), so that we can use such trees as representatives of the contextual equivalence

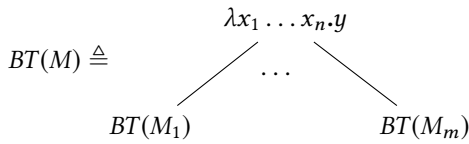
classes. Lemma 21 shows that the quantity  $\llbracket M \rrbracket(E)$  depends only on the Nakajima tree of  $M$  and that of  $E$ , so we can conclude with Lemma 22 giving  $\llbracket M \rrbracket(E) = \llbracket N \rrbracket(E)$  and hence the full abstraction result Theorem 23.

On the other hand, the counterexample to the completeness of PAS (Eq. (21)) uses the Context Lemma.

#### 4.1 Probabilistic Nakajima Trees

A *Böhm tree* [3] is a labelled tree describing a kind of infinitary normal form of a deterministic  $\lambda$ -term. In more details, the Böhm tree  $BT(M)$  of a  $\lambda$ -term  $M$  can be given co-inductively as follows:

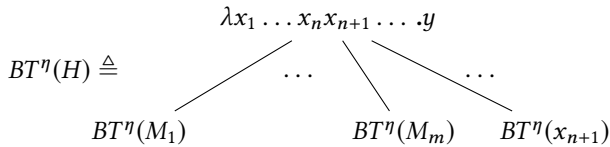
- If the head reduction of  $M$  terminates into the hnf  $\lambda x_1 \dots x_n . y M_1 \dots M_m$ , then:



where  $BT(M_1), \dots, BT(M_m)$  are the Böhm trees of the subterms  $M_1, \dots, M_m$  of the hnf of  $M$ .

- Otherwise, the tree is a node labelled by  $\Omega$ .

The notion of Böhm tree is not sufficient to characterize contextual equivalence because it lacks extensionality: the terms  $y$  and  $\lambda z . yz$  have different Böhm trees and yet  $y =_{\text{ctx}} \lambda z . yz$  holds. To recover extensionality, we need the so-called *Nakajima trees* [22], which are infinitely  $\eta$ -expanded representations of the Böhm trees. The Nakajima tree  $BT^\eta(H)$  of a hnf  $H = \lambda x_1 \dots x_n . y M_1 \dots M_m$  is the infinitely branching tree:



where  $x_1 \dots x_n x_{n+1} \dots$  is an infinite sequence of pairwise distinct variables and, for  $i > n$ , the  $x_i$ 's are fresh.

Nakajima trees represent infinitary  $\eta$ -long hnf's. Every hnf  $H = \lambda x_1 \dots x_n . y M_1 \dots M_m$   $\eta$ -expands into the head normal form  $\lambda x_1 \dots x_{n+k} . y M_1 \dots M_m x_{n+1} \dots x_{n+k}$  for any  $k \in \mathbb{N}$  and  $x_{n+1} \dots x_{n+k}$  fresh: Nakajima trees are, intuitively, the asymptotical representations of these  $\eta$ -expansions.

To generalize such a construction to probabilistic terms we define by mutual recursion the tree associated with a hnf and the tree of an arbitrary term  $M$  as a subprobability distribution over the trees of the hnf's  $M$  reduces to. Hence, strictly speaking, a probabilistic Nakajima tree is not properly a tree.

Following Leventis [19] we shall give an inductive, "level-by-level" definition of the probabilistic Nakajima trees.

The set  $\mathcal{PT}_\ell^\eta$  of *probabilistic Nakajima trees* with level at most  $\ell \in \mathbb{N}$  is the set of subprobability distributions over

value Nakajima trees  $\mathcal{VT}_\ell^\eta$ . These sets are defined by mutual recursion as follows:

$$\mathcal{VT}_0^\eta \triangleq \emptyset$$

$$\mathcal{VT}_{\ell+1}^\eta \triangleq \{\lambda x_1 x_2 \dots . y T_1, T_2, \dots \mid T_i \in \mathcal{PT}_\ell^\eta, \forall i \geq 1\},$$

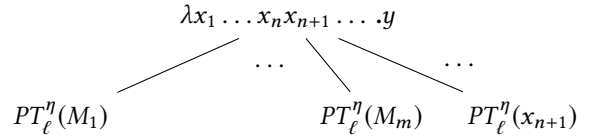
$$\mathcal{PT}_0^\eta \triangleq \{\perp\},$$

$$\mathcal{PT}_{\ell+1}^\eta \triangleq \{T : \mathcal{VT}_{\ell+1}^\eta \rightarrow [0, 1] \mid \sum_{t \in \mathcal{VT}_{\ell+1}^\eta} T(t) \leq 1\}.$$

where  $\perp$  represents the zero distribution. Value Nakajima trees are ranged over by  $t$ , and probabilistic Nakajima trees are ranged over by  $T$ .

Let  $\ell \in \mathbb{N}$ . By mutual recursion we define a function  $VT_{\ell+1}^\eta$  associating with each  $H \in \text{HNF}$  its value Nakajima tree  $VT_{\ell+1}^\eta(H)$  of level  $\ell + 1$ , and a function  $PT_\ell^\eta$  associating with each  $M \in \Lambda_\oplus$  its probabilistic Nakajima tree  $PT_\ell^\eta(M)$  of level  $\ell$ :

- If  $H = \lambda x_1 \dots x_n . y M_1 \dots M_m$ , then  $VT_{\ell+1}^\eta(H)$  is:



where  $x_1 \dots x_n x_{n+1} \dots$  is an infinite sequence of pairwise distinct variables and, for  $i > n$ , the  $x_i$ 's are fresh;

- $PT_\ell^\eta(M) \triangleq \begin{cases} t \mapsto \sum_{H \in (\mathcal{VT}_\ell^\eta)^{-1}(t)} \llbracket M \rrbracket(H) & \text{if } \ell > 0 \\ \perp & \text{otherwise.} \end{cases}$

We say that  $M$  and  $N$  have the same Nakajima tree, and we write  $M =_{\text{PT}^\eta} N$ , if  $PT_\ell^\eta(M) = PT_\ell^\eta(N)$  holds for all  $\ell \in \mathbb{N}$ .

Theorem 2 assures that the above definition based on the operational semantics  $\llbracket \cdot \rrbracket$  given in (3) is equivalent to the one given by Leventis in [19], based on the head reduction.

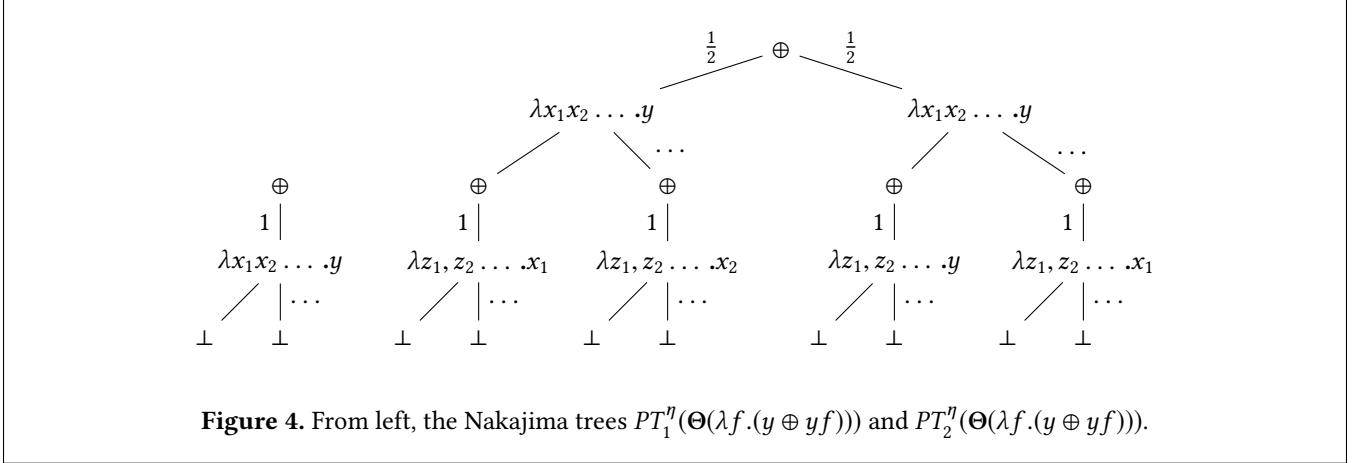
**Example 9.** Figure 4 depicts the Nakajima trees of level, respectively, 1 and 2 associated with term  $\Theta(\lambda f . (y \oplus yf))$ , where  $\Theta$  is the Turing fixed-point combinator (Example 1). Distributions are represented by barycentric sums, depicted as  $\oplus$  nodes whose outgoing edges are weighted by probabilities. Notice that the more the level  $\ell$  increases, the more the top-level distribution's support grows.

**Proposition 17** ([19]). *Let  $M, N \in \Lambda_\oplus$ . If  $PT_\ell^\eta(M) = PT_\ell^\eta(N)$  for some  $\ell \in \mathbb{N}$ , then  $PT_{\ell'}^\eta(M) = PT_{\ell'}^\eta(N)$  for all  $\ell' \leq \ell$ .*

**Theorem 18** (Separation [19]). *Let  $M, N \in \Lambda_\oplus$ . If  $M =_{\text{ctx}} N$  then  $M =_{\text{PT}^\eta} N$ .*

#### 4.2 The Completeness Theorem

In the previous subsection probabilistic Nakajima trees have been inductively presented by introducing "level-by-level" their finite representations. To recover the full quantitative information of a Nakajima tree we shall need a notion of approximation together with some general properties.



**Figure 4.** From left, the Nakajima trees  $PT_1^\eta(\Theta(\lambda f.(y \oplus yf)))$  and  $PT_2^\eta(\Theta(\lambda f.(y \oplus yf)))$ .

**Definition 3.** Let  $r, r' \in \mathbb{R}$  and  $\epsilon > 0$ . We say that  $r$   $\epsilon$ -approximates  $r'$ , and we write  $r \approx_\epsilon r'$ , if  $|r - r'| < \epsilon$ .

**Fact 19.** Let  $r, r', r'' \in \mathbb{R}$  and  $\epsilon, \epsilon' > 0$ . If  $r \approx_\epsilon r'$  and  $r' \approx_{\epsilon'} r''$  then  $r \approx_{\epsilon+\epsilon'} r''$ .

**Lemma 20.** Let  $\{A_n\}_{n \in \mathbb{N}}$  be a descending chain of countable sets of positive real numbers satisfying  $\sum_{r \in A_n} r < \infty$ , for all  $n \in \mathbb{N}$ . Then:

$$\sum_{r \in \bigcap_{n \in \mathbb{N}} A_n} r = \inf_{n \in \mathbb{N}} \left( \sum_{r \in A_n} r \right). \quad (19)$$

A consequence of Theorem 18 is that for every contextual equivalence class  $E \in \Lambda_\oplus^0 / =_{\text{cxt}}$  and for every level  $\ell \in \mathbb{N}$  there exists a *unique* value Nakajima tree  $t$  of that level such that  $VT_\ell^\eta(H) = t$  for all  $H \in E$ . Let  $t_{E, \ell}$  denote such a tree.

**Lemma 21.** Let  $M \in \Lambda_\oplus^0$  and  $E \in \Lambda_\oplus^0 / =_{\text{cxt}}$ . We have:

- (1)  $\llbracket M \rrbracket(E) = \inf_{\ell \in \mathbb{N}} (PT_\ell^\eta(M)(t_{E, \ell}))$ .
- (2)  $\forall \epsilon > 0 \exists \ell \in \mathbb{N} \forall \ell' \geq \ell: \llbracket M \rrbracket(E) \approx_\epsilon PT_{\ell'}^\eta(M)(t_{E, \ell'})$ .

*Proof.* Let  $E_V \triangleq E \cap \text{HNF}^0$ , notice that  $\llbracket M \rrbracket(E) = \llbracket M \rrbracket(E_V)$ . As for point (1), we have  $H \in E_V$  if and only if  $\forall \ell \in \mathbb{N} VT_\ell^\eta(H) = t_{E, \ell}$  if and only if  $\forall \ell \in \mathbb{N} H \in (VT_\ell^\eta)^{-1}(t_{E, \ell})$ , so that  $E_V = \bigcap_{\ell \in \mathbb{N}} (VT_\ell^\eta)^{-1}(t_{E, \ell})$ . Moreover, by Proposition 17, for all  $\ell \in \mathbb{N}$  it holds that:

$$\begin{aligned} (VT_{\ell+1}^\eta)^{-1}(t_{E, \ell+1}) &= \{H \in \text{HNF}^0 \mid VT_{\ell+1}^\eta(H) = t_{E, \ell+1}\} \\ &\subseteq \{H \in \text{HNF}^0 \mid VT_\ell^\eta(H) = t_{E, \ell}\} \\ &= (VT_\ell^\eta)^{-1}(t_{E, \ell}). \end{aligned} \quad (20)$$

Therefore,  $((VT_\ell^\eta)^{-1}(t_{E, \ell}))_{\ell \in \mathbb{N}}$  is a descending chain, so that  $\{\llbracket M \rrbracket(H) \mid H \in (VT_\ell^\eta)^{-1}(t_{E, \ell})\}_{\ell \in \mathbb{N}}$  is. Moreover, by definition we have  $\sum_{H \in (VT_\ell^\eta)^{-1}(t_{E, \ell})} \llbracket M \rrbracket(H) \leq \sum \llbracket M \rrbracket \leq 1$ , for all  $\ell \in \mathbb{N}$ . Hence, by applying Lemma 20 and by definition of Nakajima tree equality, we have:

$$\llbracket M \rrbracket(E) = \sum_{H \in E_V} \llbracket M \rrbracket(H) = \sum_{H \in \bigcap_{\ell \in \mathbb{N}} (VT_\ell^\eta)^{-1}(t_{E, \ell})} \llbracket M \rrbracket(H)$$

$$\begin{aligned} &= \inf_{\ell \in \mathbb{N}} \sum_{H \in (VT_\ell^\eta)^{-1}(t_{E, \ell})} \llbracket M \rrbracket(H) \\ &= \inf_{\ell \in \mathbb{N}} (PT_\ell^\eta(M)(t_{E, \ell})). \end{aligned}$$

Let us prove point (2). On the one hand,  $(PT_\ell^\eta(M)(t_{E, \ell}))_{\ell \in \mathbb{N}}$  is clearly a bounded below sequence. On the other hand, from (20) it is also monotone decreasing. Indeed, for all  $\ell \in \mathbb{N}$ :

$$\begin{aligned} PT_{\ell+1}^\eta(M)(t_{E, \ell+1}) &= \sum_{H \in (VT_{\ell+1}^\eta)^{-1}(t_{E, \ell+1})} \llbracket M \rrbracket(H) \\ &\leq \sum_{H \in (VT_\ell^\eta)^{-1}(t_{E, \ell})} \llbracket M \rrbracket(H) = PT_\ell^\eta(M)(t_{E, \ell}). \end{aligned}$$

Thus,  $\lim_{\ell \rightarrow \infty} (PT_\ell^\eta(M)(t_{E, \ell}))_{\ell \in \mathbb{N}} = \inf_{\ell \in \mathbb{N}} (PT_\ell^\eta(M)(t_{E, \ell})) = \llbracket M \rrbracket(E)$ , and point (2) follows by definition of limit.  $\square$

**Lemma 22.** Let  $M, N \in \Lambda_\oplus^0$ . If  $M =_{\text{cxt}} N$  then  $\llbracket M \rrbracket(E) = \llbracket N \rrbracket(E)$ , for all  $E \in \Lambda_\oplus^0 / =_{\text{cxt}}$ .

*Proof.* Suppose toward contradiction that  $\llbracket M \rrbracket(E) \neq \llbracket N \rrbracket(E)$  and consider  $\epsilon > 0$  such that  $2\epsilon \leq |\llbracket M \rrbracket(E) - \llbracket N \rrbracket(E)|$ . By Lemma 21.(2) there exist  $\ell \in \mathbb{N}$  such that:

$$\llbracket M \rrbracket(E) \approx_\epsilon PT_\ell^\eta(M)(t_{E, \ell}) \quad \llbracket N \rrbracket(E) \approx_\epsilon PT_\ell^\eta(N)(t_{E, \ell}).$$

By Theorem 18, from  $M =_{\text{cxt}} N$  we obtain  $M =_{\text{PT}^\eta} N$ , and hence  $PT_\ell^\eta(M) = PT_\ell^\eta(N)$ . By Fact 19,  $\llbracket M \rrbracket(E) \approx_{2\epsilon} \llbracket N \rrbracket(E)$ , i.e.  $|\llbracket M \rrbracket(E) - \llbracket N \rrbracket(E)| < 2\epsilon$ . A contradiction.  $\square$

*Remark 2.* Observe that the statement of Lemma 22 may fail when  $\Lambda_\oplus$  is endowed with a different operational semantics than head reduction. As an example, recall the terms  $M \triangleq \lambda xy.(x \oplus y)$  and  $N \triangleq (\lambda xy.x) \oplus (\lambda xy.y)$  discussed in the Introduction (Eq. (1)). In the lazy cbn,  $M$  and  $N$  are contextually equivalent [16]. Moreover,  $M$  is a value for lazy cbn, while  $N$  reduces with equal probability  $\frac{1}{2}$  to  $T = \lambda xy.x$  and  $F = \lambda xy.y$ . However,  $M$ ,  $T$  and  $F$  are pairwise contextually inequivalent since, by setting  $C = [\cdot] \mathbf{I} \Omega$ , we have that  $C[M]$ ,  $C[T]$ , and  $C[F]$  converge with probability  $\frac{1}{2}$ , 1, and 0, respectively. Therefore, by setting  $E$  as the lazy cbn contextual

equivalence class containing  $M$ , we have  $\llbracket M \rrbracket(E) = 1$ , while  $\llbracket N \rrbracket(E) = 0$ .

**Theorem 23** (Full abstraction). *For all  $M, N \in \Lambda_{\oplus}$ :*

$$M =_{\text{cxt}} N \Leftrightarrow M \sim N .$$

*Proof.* The right-to-left direction is Theorem 16.(2). Concerning the converse, we first consider the case of closed terms. So, let  $M, N \in \Lambda_{\oplus}^0$  be such that  $M =_{\text{cxt}} N$ . We prove that there exists probabilistic applicative bisimulation  $\mathcal{R}$  containing  $=_{\text{cxt}}$ . We define  $\mathcal{R}$  as follows:

$$\begin{aligned} & \{(P, Q) \in \Lambda_{\oplus}^0 \times \Lambda_{\oplus}^0 \mid P =_{\text{cxt}} Q\} \\ & \cup \{(vx.H, vx.H') \in \widehat{\text{HNF}} \times \widehat{\text{HNF}} \mid \lambda x.H =_{\text{cxt}} \lambda x.H'\} . \end{aligned}$$

Let us prove that  $\mathcal{R}$  is a probabilistic applicative bisimulation. Since  $=_{\text{cxt}}$  is an equivalence relation, then  $\mathcal{R}$  is. Now, let  $(vx.H, vx.H'), (P, Q) \in \mathcal{R}$ ,  $E \in (\Lambda_{\oplus}^0 \cup \widehat{\text{HNF}})/\mathcal{R}$ , and let  $l \in \Lambda_{\oplus}^0 \cup \{\tau\}$ . We have to show that:

- (1)  $\mathcal{P}_{\oplus}(P, l, E) = \mathcal{P}_{\oplus}(Q, l, E)$ ,
- (2)  $\mathcal{P}_{\oplus}(vx.H, l, E) = \mathcal{P}_{\oplus}(vx.H', l, E)$ .

Let us prove point (1). If  $l \in \Lambda_{\oplus}^0$  then  $\mathcal{P}_{\oplus}(P, l, E) = 0 = \mathcal{P}_{\oplus}(Q, l, E)$ . If  $l = \tau$  we define  $\widehat{E} \triangleq \{\lambda x.H \in \widehat{\text{HNF}}^0 \mid vx.H \in E\} \cup \{P' \in \Lambda_{\oplus}^0 \mid P' \in E\}$ . Then, by definition:

$$\mathcal{P}_{\oplus}(P, \tau, E) = \llbracket P \rrbracket(\widehat{E}) \quad \mathcal{P}_{\oplus}(Q, \tau, E) = \llbracket Q \rrbracket(\widehat{E}) .$$

Since  $(P, Q) \in \mathcal{R}$  and  $E \in (\Lambda_{\oplus}^0 \cup \widehat{\text{HNF}})/\mathcal{R}$ , it holds that  $P =_{\text{cxt}} Q$  and  $\widehat{E} \in \Lambda_{\oplus}^0 / =_{\text{cxt}}$ . By applying Lemma 22 we have  $\llbracket P \rrbracket(\widehat{E}) = \llbracket Q \rrbracket(\widehat{E})$ , and hence  $\mathcal{P}_{\oplus}(P, \tau, E) = \mathcal{P}_{\oplus}(Q, \tau, E)$ .

Let us now prove point (2). If  $l = \tau$  then  $\mathcal{P}_{\oplus}(vx.H, \tau, E) = 0 = \mathcal{P}_{\oplus}(vx.H', \tau, E)$ . Otherwise, let  $l = L \in \Lambda_{\oplus}^0$ . Since  $=_{\text{cxt}}$  is  $\leq_{\text{cxt}} \cap (\leq_{\text{cxt}})^{op}$ , by Lemma 7.(3) we have that  $\lambda x.H =_{\text{cxt}} \lambda x.H'$  implies  $(\lambda x.H)L =_{\text{cxt}} (\lambda x.H')L$ . From Proposition 1.(1) and Proposition 4 we have:

$$H[L/x] =_{\text{cxt}} (\lambda x.H)L =_{\text{cxt}} (\lambda x.H')L =_{\text{cxt}} H'[L/x] .$$

Therefore,  $H[L/x] \in E$  if and only if  $H'[L/x] \in E$ , and hence  $\mathcal{P}_{\oplus}(vx.H, L, E) = \mathcal{P}_{\oplus}(vx.H', L, E)$ .

Now, let  $M, N \in \Lambda_{\oplus}^{\{x_1, \dots, x_n\}}$  be such that  $M =_{\text{cxt}} N$ . Since  $=_{\text{cxt}}$  is  $\leq_{\text{cxt}} \cap (\leq_{\text{cxt}})^{op}$ , by repeatedly applying Lemma 7.(1) and Lemma 9.(1),  $\lambda x_1 \dots x_n.M =_{\text{cxt}} \lambda x_1 \dots x_n.N$ . Since these terms are closed, we obtain  $\lambda x_1 \dots x_n.M \sim \lambda x_1 \dots x_n.N$ . Finally, from (13) we conclude  $M \sim N$ .  $\square$

### 4.3 PAS is Not Complete

Theorem 23 establishes a precise correspondence between PAB and contextual equivalence. But what about PAS and contextual preorder? The soundness theorem (Theorem 16) states that the former implies the latter, so that it is natural to wonder whether the converse holds as well. Surprisingly enough, as in the case of the lazy reduction strategies (see [16] and [6]), the answer is negative.

A counterexample to PAS completeness is given by:

$$M \triangleq \lambda x.x(\Omega \oplus \mathbf{I}), \quad N \triangleq \lambda x.(x\Omega \oplus x\mathbf{I}). \quad (21)$$

whose Markov chain is sketched in Figure 5. First, observe that  $M$  and  $N$  are incomparable with respect to PAS:

**Lemma 24.** *Neither  $M \lesssim N$  nor  $N \lesssim M$  hold.*

*Proof.* Let  $M \lesssim N$ . Then,  $\mathcal{P}_{\oplus}(M, \tau, \widetilde{M}) \leq \mathcal{P}_{\oplus}(N, \tau, \lesssim(\widetilde{M}))$ , so that  $vx.x\Omega \in \lesssim(\widetilde{M})$ , and  $\widetilde{M} \lesssim vx.x\Omega$ . Hence,  $\mathcal{P}_{\oplus}(\widetilde{M}, \mathbf{I}, \mathbf{I}(\Omega \oplus \mathbf{I})) \leq \mathcal{P}_{\oplus}(vx.x\Omega, \mathbf{I}, \lesssim(\mathbf{I}(\Omega \oplus \mathbf{I})))$ . This means that  $\mathbf{I}\Omega \in \lesssim(\mathbf{I}(\Omega \oplus \mathbf{I}))$ , so that  $\mathbf{I}(\Omega \oplus \mathbf{I}) \lesssim \mathbf{I}\Omega$ . So  $\frac{1}{2} = \mathcal{P}_{\oplus}(\mathbf{I}(\Omega \oplus \mathbf{I}), \tau, \widetilde{\mathbf{I}}) \leq \mathcal{P}_{\oplus}(\mathbf{I}\Omega, \tau, \lesssim(\widetilde{\mathbf{I}})) = 0$ . A contradiction.

Now, suppose  $N \lesssim M$ . Then we have  $\mathcal{P}_{\oplus}(N, \tau, vx.x\mathbf{I}) \leq \mathcal{P}_{\oplus}(M, \tau, \lesssim(vx.x\mathbf{I}))$ , so that  $\widetilde{M} \in \lesssim(vx.x\mathbf{I})$ , and  $vx.x\mathbf{I} \lesssim \widetilde{M}$ . Hence,  $\mathcal{P}_{\oplus}(vx.x\mathbf{I}, \mathbf{I}, \mathbf{I}) \leq \mathcal{P}_{\oplus}(\widetilde{M}, \mathbf{I}, \lesssim(\mathbf{I}))$ . This means that  $\mathbf{I}(\Omega \oplus \mathbf{I}) \in \lesssim(\mathbf{I})$ , so that  $\mathbf{I} \lesssim \mathbf{I}(\Omega \oplus \mathbf{I})$ . Therefore,  $1 = \mathcal{P}_{\oplus}(\mathbf{I}, \tau, \widetilde{\mathbf{I}}) \leq \mathcal{P}_{\oplus}(\mathbf{I}(\Omega \oplus \mathbf{I}), \tau, \lesssim(\widetilde{\mathbf{I}})) = \frac{1}{2}$ . A contradiction.  $\square$

However, the two terms can be compared through the contextual preorder relation:

**Lemma 25.** *It holds that  $M \leq_{\text{cxt}} N$ .*

*Proof (sketch).* By Lemma 9 it is enough to show that  $M \leq_{\text{app}} N$ . Since  $M, N \in \Lambda_{\oplus}^0$ , this amounts to check that for all finite sequences  $L_1, \dots, L_n \in \Lambda_{\oplus}^0$ , it holds that  $\sum \llbracket ML_1 \dots L_n \rrbracket \leq \sum \llbracket NL_1 \dots L_n \rrbracket$ . The proof easily follows once one has:

$$\llbracket L[\Omega/x] \rrbracket \leq_{\mathfrak{D}} \llbracket L[\mathbf{I}/x] \rrbracket, \quad (22)$$

$$\sum \llbracket L[(\Omega \oplus \mathbf{I})/x] \rrbracket \leq \frac{1}{2} \cdot \sum \llbracket L[\Omega/x] \rrbracket + \frac{1}{2} \cdot \sum \llbracket L[\mathbf{I}/x] \rrbracket, \quad (23)$$

for any term  $L$ . The first inequation is an easy consequence of Proposition 4, while the second one can be proven by induction on an approximation of  $\llbracket \cdot \rrbracket$ .  $\square$

**Theorem 26.** *PAS is not complete (hence fully abstract) with respect to contextual preorder.*

## 5 Conclusion

We have considered the untyped probabilistic  $\lambda$ -calculus  $\Lambda_{\oplus}$  endowed with an operational semantics based on the head spine reduction, a variant of the head reduction strategy giving rise to the same big-step semantics (Theorem 2). We have proven that probabilistic applicative bisimilarity is fully abstract with respect to contextual equivalence (Theorem 23). The soundness part is a consequence of a Context Lemma (Lemma 9). The completeness proof relies on the Separation Theorem, introducing probabilistic Nakajima trees [19].

Our result completes the picture about fully abstract descriptions of the probabilistic head reduction contextual equivalence, adding finally a coinductive characterisation. To the best of our knowledge, this picture can be resumed by the equivalences of all the following items, for  $M$  and  $N$  probabilistic  $\lambda$ -terms:

1.  $M$  and  $N$  are contextually equivalent,

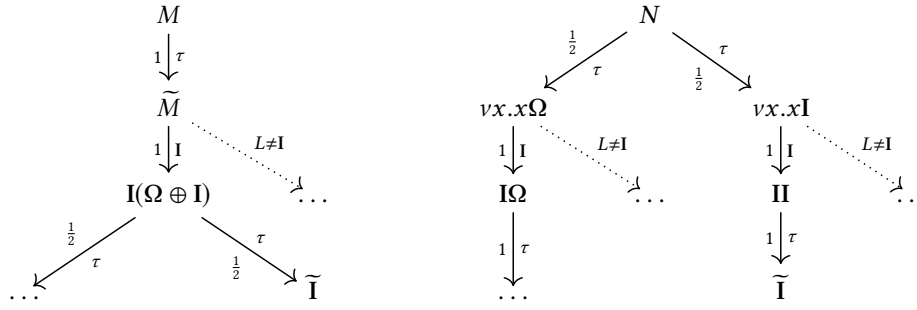


Figure 5. Markov chain for  $M = \lambda x.x(\Omega \oplus I)$  and  $N = \lambda x.(x\Omega \oplus xI)$ .

2.  $M$  and  $N$  have the same probabilistic Nakajima tree [19, 20],
3.  $M$  and  $N$  have the same denotation in the reflexive arena  $\mathcal{U}$  of the cartesian closed category of probabilistic concurrent game semantics [5],
4.  $M$  and  $N$  have the same denotation in the reflexive object  $\mathcal{D}^\infty$  of the cartesian closed category of probabilistic coherence spaces or of the  $\mathbb{R}^+$ -weighted relations [5, 20],
5.  $M$  and  $N$  are applicatively bisimilar (this paper),
6.  $M$  and  $N$  are testing equivalent according to the testing language  $T_0$  (a consequence of [25] and this paper).

Last, we have shown a counterexample to the full abstraction problem for probabilistic applicative similarity (Equation (21)). We conjecture that extending the calculus with Plotkin’s parallel disjunction [23], as done in [7], is enough to restore this property. This is left to future work.

## Acknowledgments

We would like to thank T. Ehrhard, R. Crubillé, V. Vignudelli and the anonymous reviewers for useful comments and discussions. This material is based upon work supported by the French ANR : “Agence Nationale de Recherche” under Grant “PPS: Probabilistic Program Semantics”, No. ANR-19-CE48-0014.

## References

- [1] Samson Abramsky. 1990. *The Lazy Lambda Calculus*. Addison-Wesley Longman Publishing Co., Inc., USA, 65–116.
- [2] Samson Abramsky and C-H Luke Ong. 1993. Full abstraction in the lazy lambda calculus. *Information and Computation* 105, 2 (1993), 159–267.
- [3] Hendrik P Barendregt et al. 1984. *The lambda calculus*. Vol. 3. North-Holland Amsterdam.
- [4] Bard Bloom. 1990. Can LCF be topped? Flat lattice models of typed  $\lambda$ -calculus. *Information and Computation* 87, 1-2 (1990), 264–301.
- [5] Pierre Clairambault and Hugo Paquet. 2018. Fully Abstract Models of the Probabilistic lambda-calculus. In *27th EACSL Annual Conference on Computer Science Logic, CSL 2018, September 4-7, 2018, Birmingham, UK (LIPIcs)*, Dan R. Ghica and Achim Jung (Eds.), Vol. 119. Schloss

- Dagstuhl - Leibniz-Zentrum fuer Informatik, 16:1–16:17. <https://doi.org/10.4230/LIPIcs.CSL.2018.16>
- [6] Raphaëlle Crubillé and Ugo Dal Lago. 2014. On Probabilistic Applicative Bisimulation and Call-by-Value  $\lambda$ -Calculi. In *Programming Languages and Systems*, Zhong Shao (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 209–228.
- [7] Raphaëlle Crubillé, Ugo Dal Lago, Davide Sangiorgi, and Valeria Vignudelli. 2015. On applicative similarity, sequentiality, and full abstraction. In *Correct System Design*. Springer, 65–82.
- [8] Gianluca Curzi and Michele Pagani. 2020. The Benefit of Being Non-Lazy in Probabilistic Lambda-calculus. arXiv:cs.LO/2004.12891
- [9] Ugo Dal Lago and Margherita Zorzi. 2012. Probabilistic operational semantics for the lambda calculus. *RAIRO-Theoretical Informatics and Applications* 46, 3 (2012), 413–450.
- [10] Vincent Danos and Thomas Ehrhard. 2011. Probabilistic coherence spaces as a model of higher-order probabilistic computation. *Information and Computation* 209, 6 (2011), 966–991.
- [11] Thomas Ehrhard, Michele Pagani, and Christine Tasson. 2011. The Computational Meaning of Probabilistic Coherence Spaces. In *Proceedings of the 26th Annual IEEE Symposium on Logic in Computer Science, LICS 2011, June 21-24, 2011, Toronto, Ontario, Canada*. IEEE Computer Society, 87–96. <https://doi.org/10.1109/LICS.2011.29>
- [12] Thomas Ehrhard, Michele Pagani, and Christine Tasson. 2018. Full Abstraction for Probabilistic PCF. *J. ACM* 65, 4 (2018), 23:1–23:44. <https://doi.org/10.1145/3164540>
- [13] Thomas Ehrhard, Christine Tasson, and Michele Pagani. 2014. Probabilistic coherence spaces are fully abstract for probabilistic PCF. In *The 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '14, San Diego, CA, USA, January 20-21, 2014*, Suresh Jagannathan and Peter Sewell (Eds.). ACM, 309–320. <https://doi.org/10.1145/2535838.2535865>
- [14] Douglas J Howe. 1996. Proving congruence of bisimulation in functional programming languages. *Information and Computation* 124, 2 (1996), 103–112.
- [15] Simona Kasterovic and Michele Pagani. 2019. The Discriminating Power of the Let-In Operator in the Lazy Call-by-Name Probabilistic lambda-Calculus. In *4th International Conference on Formal Structures for Computation and Deduction, FSCD 2019, June 24-30, 2019, Dortmund, Germany (LIPIcs)*, Herman Geuvers (Ed.), Vol. 131. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 26:1–26:20. <https://doi.org/10.4230/LIPIcs.FSCD.2019.26>

- [16] Ugo Dal Lago, Davide Sangiorgi, and Michele Alberti. 2014. On coinductive equivalences for higher-order probabilistic functional programs. In *The 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '14, San Diego, CA, USA, January 20–21, 2014*, Suresh Jagannathan and Peter Sewell (Eds.). ACM, 297–308. <https://doi.org/10.1145/2535838.2535872>
- [17] Jim Laird, Giulio Manzonetto, Guy McCusker, and Michele Pagani. 2013. Weighted Relational Models of Typed Lambda-Calculi. In *28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25–28, 2013*. IEEE Computer Society, 301–310. <https://doi.org/10.1109/LICS.2013.36>
- [18] Kim G Larsen and Arne Skou. 1991. Bisimulation through probabilistic testing. *Information and computation* 94, 1 (1991), 1–28.
- [19] Thomas Leventis. 2018. Probabilistic Böhm Trees and Probabilistic Separation. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09–12, 2018*, Anuj Dawar and Erich Grädel (Eds.). ACM, 649–658. <https://doi.org/10.1145/3209108.3209126>
- [20] Thomas Leventis and Michele Pagani. 2019. Strong Adequacy and Untyped Full-Abstraction for Probabilistic Coherence Spaces. In *Foundations of Software Science and Computation Structures - 22nd International Conference, FOSSACS 2019, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2019, Prague, Czech Republic, April 6–11, 2019, Proceedings (Lecture Notes in Computer Science)*, Mikolaj Bojanczyk and Alex Simpson (Eds.), Vol. 11425. Springer, 365–381. [https://doi.org/10.1007/978-3-030-17127-8\\_21](https://doi.org/10.1007/978-3-030-17127-8_21)
- [21] Robin Milner. 1977. Fully abstract models of typed  $\lambda$ -calculi. *Theoretical Computer Science* 4, 1 (1977), 1–22.
- [22] Reiji Nakajima. 1975. Infinite normal forms for the  $\lambda$ -calculus. In  *$\lambda$ -Calculus and Computer Science Theory*, C. Böhm (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 62–82.
- [23] Gordon D. Plotkin. 1977. LCF considered as a programming language. *Theoretical computer science* 5, 3 (1977), 223–255.
- [24] Peter Sestoft. 2002. Demonstrating lambda calculus reduction. In *The essence of computation*. Springer, 420–435.
- [25] Franck van Breugel, Michael Mislove, Joel Ouaknine, and James Worrel. 2005. Domain theory, testing and simulation for labelled Markov processes. *Theoretical Computer Science* 333, 1 (2005), 171 – 197. <https://doi.org/10.1016/j.tcs.2004.10.021> Foundations of Software Science and Computation Structures.