# Probabilistic Coherence Spaces are Fully Abstract for Probabilistic PCF *

Thomas Ehrhard Christine Tasson

Laboratoire PPS - CNRS - Université Paris Diderot

thomas.ehrhard@pps.univ-paris-diderot.fr
christine.tasson@pps.univ-paris-diderot.fr

Michele Pagani

Laboratoire LIPN - Université Paris 13

michele.pagani@lipn.univ-paris13.fr

## Abstract

Probabilistic coherence spaces (**PCoh**) yield a semantics of higher-order probabilistic computation, interpreting types as convex sets and programs as power series. We prove that the equality of interpretations in **PCoh** characterizes the operational indistinguishability of programs in PCF with a random primitive.

This is the first result of full abstraction for a semantics of probabilistic PCF. The key ingredient relies on the regularity of power series and introduces, in denotational semantics, techniques coming from Calculus.

Along the way to the theorem, we design a weighted intersection type assignment system giving a logical presentation of **PCoh**.

***Categories and Subject Descriptors*** CR-number [*subcategory*]: third-level

## 1. Introduction

Probabilistic behaviors appear in many places in the study of programing languages, for instance if the environment of a program behaves randomly or if the program uses probabilistic constructs.

To understand how the introduction of probabilities changes the computational landscape, we use *Semantics*. Indeed, in the last decades [24, 27, 30], semantics has succeeded in giving insights on the way programs compute. More precisely, *operational semantics* allows one to formalize a program by the sequence of its execution steps, while *denotational semantics* represents programs by functions in some mathematical space relating the interpretations of inputs and outputs in a compositional way. If this last mathematical representation is correct and accurate enough, then denotational properties lead to computational features. A key example is *full abstraction* [23], stating that operational indistinguishability (i.e. behaving in the same way in any context) is characterized by denotational equivalence (i.e. having the same interpretation). So semantics is useful both to give a precise meaning to syntactical constructs and to separate non-equivalent programs.

Of course, probabilistic semantics have already been investigated. First, the domain-theoretic approach has led to a *probabilistic powerdomain* [19, 29] which is a sibling of the non-deterministic power domain [27]. This approach follows the computational monad method [25]: programs are interpreted as functions from the input domain to the powerdomain of the output domain. Intuitively, a program takes an input and returns a probability distribution on outputs. This line of work has been continued by the continuous random variable construction [15], introducing standard tools of probability theory into semantics. Secondly, the game-theoretic approach [1, 18] has been extended with probabilistic features [5]. Intuitively, probabilistic programs are interpreted as probabilistic strategies, that are stochastic processes on the plays of the games associated with the types of the programs.

*Quantitative semantics* follows another tradition stemming from Linear Logic models [10, 11]. From the very beginning, Linear Logic has been associated with intuitions coming from calculus and linear algebra [7, 8, 12]. Indeed, programs are interpreted as entire series between mathematical spaces or as analytic functors between sets [17], and programs that use their resources only once are interpreted as linear functions. This connection with resource consumption has been fruitful in the last decade with the introduction of differential nets and resource calculus [2, 9]. Recently, quantitative semantics has been explicitly related to the study of quantitative properties such as time or space consumed by a computation [22]. This illustrates a new paradigm where a semiring of scalars allows one to encode non-deterministic or probabilistic computations as opposed to the domain paradigm where monads are used. Another important tenet of Linear Logic is the perfect duality between programs and environments since a program can be seen as the environment of other programs. Therefore, representing probabilistic environments or programs will boil down to the same study. *Probabilistic coherence spaces* provide a quantitative semantics [4, 14] which model probabilistic computations.

The main contribution of the present work is to show that probabilistic coherence spaces provide a fully abstract model of PPCF, a probabilistic extension of the functional programing language PCF [28]. Although the proof follows the general pattern that consists in finding a *definable* context that separates two terms, the key ingredient is based on Calculus (see Lemma 25) since programs are interpreted as power series.

To our knowledge, no known model of probabilistic PCF has yet been proved to be fully abstract. Games semantics provide fully abstract models of standard PCF via an extensional collapse [1, 18]. This technique has been adapted to probabilistic game semantics, providing a fully abstract model of probabilistic idealized ALGOL [5] (an extension of probabilistic PCF with references). Our result characterizes the operational indistinguishability without the

* Partially founded by French ANR project COQUAS (number 12 JS02 006 01) and CNRS chair “Logique linéaire et calcul”.



*POPL '14*, January 22–24, 2014, San Diego, CA, USA.

ACM 978-1-4503-2544-8/14/01…$15.00.
http://dx.doi.org/10.1145/2535838.2535865

need of an extensional collapse and deals with a functional probabilistic language with no references.

Section 2 is devoted to an insight on the way programs and data are interpreted in probabilistic coherence spaces. Section 3 is devoted to the syntax and the operational semantics. Next, in Section 4, we describe the key notions of probabilistic coherence spaces that will be useful to prove, in Section 5, full abstraction. Along the way to the theorem, we define an intersection type assignment system (Figure 3) giving a logical presentation to the model. This system has an interest by its own, allowing one to turn a question of computing the semantics of a term (and hence its operational behavior) into a proof search problem. Finally, in Section 6, we show that inequational full abstraction fails, i.e. the semantical order does not coincide with the operational one. For this, we achieve a context lemma for probabilistic PCF (Proposition 31).

**Notation 1.** We write $\mathbb{N}$ for the set of non negative *integers*, $\mathbb{N}^*$ for the set of positive integers ($\mathbb{N}^* \triangleq \mathbb{N}\backslash\{0\}$), $\mathbb{R}_+$ for the set of non negative *real* numbers and $\overline{\mathbb{R}}_+ \triangleq \mathbb{R}_+ \cup \{\infty\}$ for the completed real half line. Let $S$ be a set, $\sharp S$ denotes its cardinality. *Multisets* of elements of $S$ are identified with functions $S \to \mathbb{N}$. If $m$ is such a multiset, $\mathrm{Supp}\,(m)$ denotes its *support* set $\{a \in S \text{ s.t. } m(a) \neq 0\}$. A *finite multiset* is a multiset with a finite support. We write $\mathcal{M}_{\mathrm{f}}\,(S)$ for the set of all finite multisets of elements of $S$. We enumerate $m$ by using $(a,i) \in m$ to denote $a \in \mathrm{Supp}\,(m)$ and $1 \leq i \leq m(a)$. Whenever $(a_1,\dots,a_n) \in S^n$, we write $[a_1,\dots,a_n]$ for the finite multiset: $a \in S \mapsto \sharp\{i \text{ s.t. } a_i = a\}$. The empty multiset is $[\,]$ and $\uplus$ is the multiset union: $(m \uplus p)(a) \triangleq m(a) + p(a)$. A *vector* $v \in \mathbb{R}_+^S$ is given by its values $v_a$ on any index $a \in S$. Given a multiset $m \in \mathcal{M}_{\mathrm{f}}\,(S)$, we define the power $v^m \triangleq \prod_{a \in \mathrm{Supp}(m)} v_a^{m(a)}$. For any $a \in S$, let $e_a \in \mathbb{R}_+^S$ be the base vector $(e_a)_b \triangleq \delta_{a,b}$, with $\delta_{a,b}$ denoting the Kronecker symbol.

Let us fix our typographic conventions: $a, b, c$ range over web elements and $m, p, q$ over multisets. $v, w, u$ range over vectors, $\phi, \psi, \xi$ over matrices, $\alpha, \beta, \gamma$ over monomials. $A, B, C$ range over simple types, $\mathtt{Int}$ being the integer type. $x, y, z$ range over term variables. $M, N, P$ range over terms. Finally, $\kappa, \mu, \nu$ range over scalars in $\mathbb{R}_+$ and $\vec{\kappa}$ denotes a list of scalars and similarly for the other metavariables.

## 2. Denoting Probabilistic Programing

Probabilistic programs use two levels of randomness: the first one on data (i.e. terms of ground type), the second one on programs (i.e. terms of higher-order type).

A *probabilistic datum* is a random variable whose outcome is given to the program. Thus, a probabilistic datum will be characterized by its law, that will be its *interpretation*. Then, a *probabilistic program* uses random instructions and behaves like a function from probabilistic data to probabilistic data. Moreover, a program can call several times its argument $x$. Each of its occurrences is represented by the outcome of an independent random variable with the same distribution as $x$.

We consider two examples of probabilistic data of type $\mathtt{Int}$. The first one, $\mathtt{Coin}$, is the toss of a $0/1$ fair coin. The second one, $\mathtt{Rand}(n)$, follows the discrete uniform distribution with outcomes between $0$ and $n-1$.

If we are only interested in resulting values, then $\mathtt{Int}$ is interpreted by the set of non negative integers $|\mathtt{Int}| = \mathbb{N}$. First, we consider $\mathtt{Coin}$ and $\mathtt{Rand}(n)$ as non deterministic data, and interpret them by the range of the corresponding random variables:

$$|\mathtt{Coin}| = \{0,1\} \quad \text{and} \quad |\mathtt{Rand}(n)| = \{0,\dots,n-1\}.$$

Then, to take into account the randomized behavior of the datum, we associate a coefficient to each outcome: its probability to happen. The interpretation of a datum is now given by a sequence of non-negative integers indexed by the possible outcomes:

$$[\![\mathtt{Coin}]\!] = (\tfrac{1}{2}, \tfrac{1}{2}, 0, \dots) \text{ and } [\![\mathtt{Rand}(n)]\!] = (\underbrace{\tfrac{1}{n}, \dots, \tfrac{1}{n}}_{n}, 0 \dots).$$

In general, we will interpret the integer type by subprobability[1] distributions over $\mathbb{N}$:

$$\mathrm{P}\,(\mathtt{Int}) = \left\{(\kappa_n)_{n\in\mathbb{N}} \in \mathbb{R}_+^{\mathbb{N}} \text{ s.t. } \textstyle\sum_{n\in\mathbb{N}} \kappa_n \leq 1\right\}.$$

For probabilistic programs, we follow the same pattern as for probabilistic data. First, their *non deterministic* behavior is described. Then, coefficients are introduced in order to render their *quantitative* behavior.

As an example, let us consider the program $\mathtt{Rand} : \mathtt{Int} \Rightarrow \mathtt{Int}$ that takes an input value $n$, and returns any non negative integer strictly less than $n$ with equal chances and make it interact with probabilistic data (e.g. a probabilistic distribution over $\mathbb{N}$).

Focusing on the association between input and output values, its non deterministic behavior is described as a relation: $|\mathtt{Rand}| \subseteq |\mathtt{Int}| \times |\mathtt{Int}|$

$$|\mathtt{Rand}| = \left\{(n,a) \text{ s.t. } n \in \mathbb{N},\ a \in \{0,\dots,n-1\}\right\}.$$

Then we associate a coefficient to each pair input-output. It represents the quantitative account of getting the given output knowing the input. The interpretation of the program is now turned into a matrix indexed by the values interpreting inputs (column indices) and outputs (raw indices)[2]: $[\![\mathtt{Rand}]\!] \in (\mathbb{R}_+)^{|\mathtt{Int}|\times|\mathtt{Int}|}$

$$[\![\mathtt{Rand}]\!] = \begin{array}{c} \begin{array}{cccccc} \overset{0}{\downarrow} & \overset{1}{\downarrow} & \overset{2}{\downarrow} & \cdots & \overset{n}{\downarrow} & \cdots \end{array} \\ \left(\begin{array}{cccccc} 0 & 1 & \frac{1}{2} & \cdots & \frac{1}{n} & \cdots \\ 0 & 0 & \frac{1}{2} & \cdots & \frac{1}{n} & \cdots \\ \vdots & \vdots & 0 & \ddots & \vdots & \\ & & \vdots & 0 & \frac{1}{n} & \\ & & & \vdots & \ddots & \ddots \end{array}\right) \end{array} \begin{array}{l} \\ \to 0 \\ \to 1 \\ \vdots \\ \to n-1 \\ \vdots \end{array} \tag{1}$$

The interaction between the program $\mathtt{Rand}$ and a probabilistic datum $x$ is then given by the product of the matrix interpreting the program and of the sequence interpreting the datum. Besides, this probabilistic program preserves subprobability distributions.

Actually, this approach is valid only if the program uses exactly once its argument. Indeed, as a side effect of the call-by-name[3] execution strategy, each occurrence of a probabilistic datum behaves as an independent sample of a random variable. So, if a program makes several calls to a given probabilistic datum, then the outcomes of the calls may differ due to the randomized setting.Thus, we gather the input values into a finite multiset: finite, since if the execution of a program terminates, then the number of resources effectively used is finite; multiset rather than a sequence since this model is not accurate enough to distinguish the order of the inputs.

To illustrate this point, we examine the probabilistic programs:

$$\begin{aligned} \mathtt{Once} &\triangleq \lambda x^{\mathtt{Int}}.\mathsf{if}(x, \mathtt{Coin}, \underline{42}), \\ \mathtt{Twice} &\triangleq \lambda x^{\mathtt{Int}}.\mathsf{if}(x, \mathsf{if}(x, \mathtt{Coin}, \underline{42}), \mathsf{if}(x, \underline{42}, \underline{0})), \end{aligned}$$

where $\mathsf{if}(x, \_, \_)$ branches depending whether $x$ evaluates to zero or not. Notice that the two programs uses, among others, probabilistic

[1] Since a call to a datum can fail, the total probability distribution over the possible values may be less than 1.

[2] $[\![\mathtt{Rand}]\!]$ is the transpose of a stochastic matrix and the image of a subprobability distribution is the matrix product $[\![\mathtt{Rand}]\!] \cdot v$.

[3] This phenomenon will also appear in a call-by-value setting, if every probabilistic datum $x$ is replaced by its CPS translation $\lambda a^{\mathtt{Int}}.x$.

instructions and that they call once or twice their input. Their interpretation is given by a matrix indexed by finite multisets (the input) and integers (the output). Coefficients are non zero only if the multiset is of size one or two:

$$[\![\texttt{Once}]\!]: \quad \begin{cases} ([0],\ 0\ ) \mapsto \frac{1}{2} \\ ([0],\ 1\ ) \mapsto \frac{1}{2} \\ ([a], 42) \mapsto 1 \ \text{ if } a \neq 0 \\ (m,\ a\ ) \mapsto 0 \ \text{ otherwise.} \end{cases} \qquad [\![\texttt{Twice}]\!]: \quad \begin{cases} ([0,0],\ 0\ ) \mapsto \frac{1}{2} \\ ([0,0],\ 1\ ) \mapsto \frac{1}{2} \\ ([0,a], 42) \mapsto 2 \ \text{ if } a \neq 0 \\ ([a,b],\ 0\ ) \mapsto 1 \ \text{ if } a \neq 0, b \neq 0 \\ (\ m\ ,\ a\ ) \mapsto 0 \ \text{ otherwise.} \end{cases}$$

On the second example, two quantitative phenomena are mixed up. The first one is probabilistic: coefficient $\frac{1}{2}$ comes from the use of `coin`. The second one stems from branching construction: two different execution traces are leading to $\underline{42}$, either the two outcomes of the random variable $x$ are first 0, and then $a \neq 0$, or its outcomes are first $a$, and then 0. In our model, these traces are gathered in the same multiset $[0, a]$ and the coefficient 2 is the combinatorial counterpart. This gives a hint of why programs are not represented as random variables.

Now, the interaction of a probabilistic program with a probabilistic data boils down to a matrix product adapted to take into account multiplicities.

Let us consider a probabilistic datum $x : \texttt{Int}$ and a program $M : \texttt{Int} \Rightarrow \texttt{Int}$. As a reminder, the $a$-th coefficient $[\![x]\!]_a$ is the probability that $a$ is the outcome of $x$. The probability $[\![(M)\,x]\!]_b$, that $(M)\,x$ returns $b$, is given by decomposing the computation in pairwise disjoint events. Each event corresponds to an intermediate multiset $m$ gathering the input values that are effectively used during computation:

$$[\![(M)\,x]\!]_b = \sum_{m \in \mathcal{M}_f(\mathbb{N})} [\![M]\!]_{(m,b)}\ [\![x]\!]^m\,, \tag{2}$$

(see Notation 1 for $[\![x]\!]^m$) is the probability that independent calls to $x$ returns the values gathered in $m$. In this way, every coefficient $[\![(M)\,x]\!]_b$ is a power series with infinitely many variables, that are the $[\![x]\!]_a$ for $a \in \mathbb{N}$.

Applying Formula (2), we compute the non zero coefficients:

$$[\![(\texttt{Once})\,x]\!]_0 = [\![(\texttt{Once})\,x]\!]_1 = \tfrac{1}{2}\,[\![x]\!]_0 \qquad [\![(\texttt{Once})\,x]\!]_{42} = \sum_{a \geq 1} [\![x]\!]_a$$

$$[\![(\texttt{Twice})\,x]\!]_0 = \tfrac{1}{2}\,[\![x]\!]_0^2 + \sum_{a,b \geq 1} [\![x]\!]_a\,[\![x]\!]_b$$

$$[\![(\texttt{Twice})\,x]\!]_1 = \tfrac{1}{2}\,[\![x]\!]_0^2 \qquad [\![(\texttt{Twice})\,x]\!]_{42} = 2\sum_{a \geq 1} [\![x]\!]_0\,[\![x]\!]_a$$

Notice that every coefficient of $[\![(\texttt{Once})\,x]\!]$ is a linear function and every coefficient of $[\![(\texttt{Twice})\,x]\!]$ is a polynomial of degree 2. Besides, `Once` and `Twice` preserve subprobability distributions:

$$\sum_b [\![(\texttt{Once})\,x]\!]_b = \tfrac{1}{2}\,[\![x]\!]_0 + \tfrac{1}{2}\,[\![x]\!]_0 + \sum_{a \geq 1} [\![x]\!]_a \leq 1$$

$$\begin{aligned} \sum_b [\![(\texttt{Twice})\,x]\!]_b &= [\![x]\!]_0^2 + 2\,[\![x]\!]_0 \sum_{a \geq 1} [\![x]\!]_a + (\sum_{a \geq 1} [\![x]\!]_a)^2, \\ &= (\sum_{a \geq 0} [\![x]\!]_a)^2 \leq 1. \end{aligned}$$

## 3. Probabilistic PCF

We define the language PPCF in Figure 1. This system is a minor variant with respect to the probabilistic extension of PCF presented in [4] (see Remark 6). The grammar of terms is obtained by adding to standard PCF a random number generator `rand`.The typing rules (Figure 1(b)) are the usual ones, with `rand` of type $\texttt{Int} \Rightarrow \texttt{Int}$. From now on, we call *program* a closed term of type `Int`. Figure 1(c) gives the one step *reduction relation* $\xrightarrow{\kappa}$, implementing a weak head-reduction (i.e. a lazy call-by-name strategy). The relation is weighted with a probability $\kappa \in [0, 1]$ equal to 1 except for steps firing a $(\texttt{rand})\,\underline{n}$ redex, reducing to any numeral in $\{\underline{0}, \ldots, \underline{n-1}\}$ with equal probability $\frac{1}{n}$. The probability of a reduction sequence is the product of the weights of all its steps. Some notational conventions are introduced in Figure 1(d) and will be used henceforth.

**Example 2.** Notice (by induction on $n \geq 1$) that the term $\texttt{choose}(M_i)_{i=1}^n$, for $1 \leq i \leq n$ reduces (in several steps) to any $M_i$ with equal probability $\frac{1}{n}$. In general, any reduction sequence has a probability given by a rational number in $[0, 1]$, since $\texttt{rand}\,n$ introduces just fractions.

**Example 3.** The fix-point constructor yields infinite reduction sequences, like the typical loop $\Omega_A \xrightarrow{1} (\lambda x^A.x)\,\Omega_A \xrightarrow{1} \Omega_A$. However, we can have terms with infinite reduction sequences but converging to a normal form with probability 1. Indeed, take $M \triangleq \mathsf{fix}(\lambda x^{\texttt{Int}}.x \oplus \underline{0})$ (see notations in Figure 1(d)) we have:

$$\begin{tikzcd} M \arrow[r, "1"] & (\lambda x^{\texttt{Int}}.x \oplus \underline{0})\,M \arrow[r, "1"] & M \oplus \underline{0} \arrow[r, "\frac{1}{2}"] \arrow[ll, bend left, "\frac{1}{2}"] & \underline{0} \end{tikzcd}$$

The probability that $M$ reduces to $\underline{0}$ is equal to the sum of the probabilities of all finite reductions sequences from $M$ to $\underline{0}$, which is $\sum_{n=0}^{\infty} \frac{1}{2^{n+1}} = 1$, $n$ being the number of loops taken by a sequence.

A way for precisely defining the probability of convergence of a term to a normal form is by giving the operational semantics as a Markov process over the set of PPCF terms, following [4]. The transition matrix $\textsc{Proba} \in [0, 1]^{\text{PPCF} \times \text{PPCF}}$ is defined by:

$$\textsc{Proba}_{M,N} \triangleq \begin{cases} \kappa & \text{if } M \xrightarrow{\kappa} N, \\ 1 & \text{if } M = N \text{ is a normal form,} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Notice that $\textsc{Proba}_{M,N}$ is well-defined since there exists at most one reduction step $M \xrightarrow{\kappa} N$, once fixed $M$ and $N$. $\textsc{Proba}$ is a stochastic matrix (i.e. for all $M$, $\sum_{N \in \text{PPCF}} \textsc{Proba}_{M,N} = 1$). The value of $\textsc{Proba}_{M,N}$ intuitively describes the probability of evolving from the state $M$ to the state $N$ in one step.

A term $M$ is *absorbing* whenever $\textsc{Proba}_{M,M} = 1$: the absorbing states are those which are invariant under the transition matrix. Notice that the normal forms are all absorbing, but the converse is false, e.g. $\Omega$ is an absorbing term.

The $n$-th power $\textsc{Proba}^n$ of the matrix $\textsc{Proba}$ is a stochastic matrix on PPCF (in case $n = 0$, we have the identity matrix on PPCF). Intuitively, the value of $\textsc{Proba}^n_{M,N}$ is the probability of evolving from the state $M$ to the state $N$ in exactly $n$ steps.

**Proposition 4** ([4, Lemma 32])**.** *Let $M \in$ PPCF and $N$ absorbing, the sequence $\{\textsc{Proba}^n_{M,N}\}_{n \in \mathbb{N}}$ is monotonic.*

We can thus define, for every program $M$ and $n \in \mathbb{N}$:

$$\textsc{Proba}^{\infty}_{M,\underline{n}} \triangleq \sup_{k=0}^{\infty} (\textsc{Proba}^k_{M,\underline{n}}) \tag{4}$$

Intuitively, $\textsc{Proba}^{\infty}_{M,\underline{n}}$ defines the probability that $M$ reaches a numeral $\underline{n}$ in an arbitrary number of steps.

In standard PCF the observational pre-order is defined with respect to the termination of a term in a context of type `Int`. In a probabilistic framework like PPCF, one can refine such a pre-order

Types $A, B, C ::= \mathtt{Int} \mid A \Rightarrow B$ Terms $M, N, P ::= x \mid \lambda x^A.M \mid (M)\,N \mid \mathsf{fix}(M) \mid \underline{0} \mid \mathsf{s}(M) \mid \mathsf{p}(M) \mid \mathsf{if}(M, N, P) \mid \mathtt{rand}$

(a) Grammar of types and terms. The constant $\mathtt{Int}$ is the base type of integers. Given $n \in \mathbb{N}$, $\underline{n}$ will denote its associated *numeral*, defined as $\mathsf{s}^n(\underline{0})$.

$$\frac{}{\Gamma, x : A \vdash x : A} \qquad \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x^A.M : A \Rightarrow B} \qquad \frac{\Gamma \vdash M : A \Rightarrow B \qquad \Gamma \vdash N : A}{\Gamma \vdash (M)\,N : B} \qquad \frac{\Gamma \vdash M : A \Rightarrow A}{\Gamma \vdash \mathsf{fix}(M) : A}$$

$$\frac{}{\Gamma \vdash \underline{0} : \mathtt{Int}} \qquad \frac{\Gamma \vdash P : \mathtt{Int}}{\Gamma \vdash \mathsf{s}(P) : \mathtt{Int}} \qquad \frac{\Gamma \vdash P : \mathtt{Int}}{\Gamma \vdash \mathsf{p}(P) : \mathtt{Int}} \qquad \frac{\Gamma \vdash M : \mathtt{Int} \qquad \Gamma \vdash N : A \qquad \Gamma \vdash P : A}{\Gamma \vdash \mathsf{if}(M, N, P) : A} \qquad \frac{}{\Gamma \vdash \mathtt{rand} : \mathtt{Int} \Rightarrow \mathtt{Int}}$$

(b) Simple type assignment system.

$$\left(\lambda x^A.M\right) N \xrightarrow{1} M\,[N/x] \qquad \mathsf{if}(\underline{0}, N, P) \xrightarrow{1} N \qquad \mathsf{p}(\underline{n+1}) \xrightarrow{1} \underline{n}$$

$$\mathsf{fix}(M) \xrightarrow{1} (M)\,\mathsf{fix}(M) \qquad \mathsf{if}(\underline{n+1}, N, P) \xrightarrow{1} P \qquad (\mathtt{rand})\,\underline{n} \xrightarrow{\frac{1}{n}} \underline{k}, \quad \text{for } k < n$$

$$\frac{M \xrightarrow{\kappa} M',\ M \text{ not abstract.}}{(M)\,N \xrightarrow{\kappa} (M')\,N} \qquad \frac{M \xrightarrow{\kappa} M'}{\mathsf{s}(M) \xrightarrow{\kappa} \mathsf{s}(M')} \qquad \frac{M \xrightarrow{\kappa} M'}{\mathsf{p}(M) \xrightarrow{\kappa} \mathsf{p}(M')} \qquad \frac{M \xrightarrow{\kappa} M'}{\mathsf{if}(M, N, P) \xrightarrow{\kappa} \mathsf{if}(M', N, P)} \qquad \frac{M \xrightarrow{\kappa} M'}{(\mathtt{rand})\,M \xrightarrow{\kappa} (\mathtt{rand})\,M'}$$

(c) Reduction rules. In case $n = 0$, $(\mathtt{rand})\,\underline{n}$ is a normal form.

$$\Omega_A \triangleq \mathsf{fix}(\lambda x^A.x) \qquad M \oplus N \triangleq \mathsf{if}((\mathtt{rand})\,\underline{2}, M, N)$$

$$\mathtt{choose}(\,) \triangleq \Omega_A \qquad \mathtt{choose}(M_1, \ldots, M_{n+1}) \triangleq \mathsf{if}((\mathtt{rand})\,\underline{n+1}, M_{n+1}, \mathtt{choose}(M_1, \ldots, M_n))$$

$$\mathsf{if}(\wedge_{i=1}^{0} M_i, N, P) \triangleq N \qquad \mathsf{if}(\wedge_{i=1}^{n+1} M_i, N, P) \triangleq \mathsf{if}(M_{n+1}, \mathsf{if}(\wedge_{i=1}^{n} M_i, N, P), P)$$

$$\mathsf{if}(M = \underline{0}, N, P) \triangleq \mathsf{if}(M, N, P) \qquad \mathsf{if}(M = \underline{n+1}, N, P) \triangleq \mathsf{if}(\mathsf{p}(M) = \underline{n}, N, P)$$

(d) Notational conventions. Also, we can use $\mathtt{choose}(M_i)_{i=1}^{n}$ as a shortcut for $\mathtt{choose}(M_1, \ldots, M_n)$.

**Figure 1:** Probabilistic extension of PCF.

by comparing the probability of convergence. Let $\mathcal{C}^{\Gamma,A}$ be the set of contexts $Q$ mapping terms $M$ of type $A$ in the environment $\Gamma$, into programs $Q[M]$, i.e. closed terms of type $\mathtt{Int}$.

**Definition 5** (Observational pre-order)**.** Given $\Gamma \vdash M : A$, and $\Gamma \vdash N : A$, we define[4]:

$$M \sqsubseteq_\Gamma N \ \text{ iff } \ \forall Q \in \mathcal{C}^{\Gamma,A}, \mathrm{PROBA}^{\infty}_{Q[M],\underline{0}} \le \mathrm{PROBA}^{\infty}_{Q[N],\underline{0}}\,.$$

Let $\equiv_\Gamma$ be the equivalence induced by $\sqsubseteq_\Gamma$.

*Remark* 6. There are slightly different probabilistic primitives that can be added to PCF. We have chosen a quite standard one, $\mathtt{rand}$, implementing CAML function `Random.int`. In [5, 21] it is also considered the more basic $\mathtt{Coin} \triangleq (\mathtt{rand})\,\underline{2}$. Another possibility is to allow an arbitrary superposition of two terms $M \oplus_\kappa N$, for $\kappa \in [0, 1]$, evaluating to $M$ with probability $\kappa$ and to $N$ with probability $1 - \kappa$, see e.g. [15, 19]. Concerning $n$-ary distributions, there is $\oplus_{i=1}^{n} \kappa_i M_i$, evaluating to any $M_i$ with probability $\kappa_i$ [26]. Finally, [4] allows any probabilistic distribution over the whole set of numerals $(\kappa_n)_{n\in\mathbb{N}}$, evaluating to $\underline{n}$ with probability $\kappa_n$.

Obviously, $\mathtt{Coin}$ can be defined as $\oplus_{\frac{1}{2}}$ as well as $(\mathtt{rand})\,\underline{2}$. Also, both $\oplus_\kappa$ and $\mathtt{rand}$ are definable as $\oplus_{i=1}^{n} \kappa_i M_i$, and the latter as $(\kappa_n)_{n\in\mathbb{N}}$. However, the converses are less trivial. In [5, Figure 6] the authors show how to define $\mathtt{rand}$ as a recursive program using just $\mathtt{Coin}$. On the contrary, $\oplus_\kappa$ is strictly more expressive than $\mathtt{rand}$ and $\mathtt{Coin}$, since it is not true that all real numbers can be approximated by series of rational numbers generated from PPCF terms (hence recursively enumerable series). One can then wonder whether the observational equivalence on PCF terms depends on the chosen probabilistic primitive. The full abstraction gives a negative answer to the question: all such primitives induce the same observational equivalence on PCF terms, which is in fact the equivalence induced by **Pcoh** (Corollary 28).

[4] The numeral $\underline{0}$ chosen for testing the equality is not significant. Indeed, from a context $Q$ semi-separating two terms $M$, $N$ on a numeral $\underline{n}$, i.e. such that $\mathrm{PROBA}^{\infty}_{Q[M],\underline{n}} \not\le \mathrm{PROBA}^{\infty}_{Q[N],\underline{n}}$, one can get a context semi-separating $M$ and $N$ on another numeral $\underline{m}$ by applying a suitable number of $\mathsf{p}(\,)$ or $\mathsf{s}(\,)$ constructors.

## 4. Probabilistic Coherence Spaces

We present probabilistic coherence spaces [4, 14] and recall the main results needed to state and prove the full abstraction theorem.

The category **PCoh** of *probabilistic coherence spaces* is a model of PPCF. As usual, types are interpreted as objects of the category and programs as maps between the interpretation of the input and output types.

Actually, **PCoh** is underlied by a model of linear logic that we do not make explicit for the sake of brevity, but we refer to [4].

### 4.1 Probabilistic Coherence Spaces

A probabilistic coherence space is a pair $(|\mathcal{A}|, \mathrm{P}(\mathcal{A}))$ of a set $|\mathcal{A}|$ and a set $\mathrm{P}(\mathcal{A})$ of vectors in the module $\mathbb{R}_+^{|\mathcal{A}|}$. Intuitively, the elements in $|\mathcal{A}|$ represent atomic data, while the vectors in $\mathrm{P}(\mathcal{A})$

express probabilistic data. At the ground type, $\mathrm{P}(\texttt{Int})$ is in fact the convex set of the subprobability distributions of the elements in $|\texttt{Int}|$. However, at higher-order types, this intuition is lost[5] and the definition of $\mathrm{P}(\mathcal{A})$ depends on a duality condition describing the probability of having an interaction between a datum of type $A$ (which is a program) and an environment.

Consider a program and an environment interacting on atomic data $a$ included in a set $|\mathcal{A}|$ such that their respective interpretation are positive real vectors: $v, w \in \mathbb{R}_+^{|\mathcal{A}|}$. Their pairing gives a quantitative estimation of the interaction success

$$\langle v, w\rangle \triangleq \sum_{a\in|\mathcal{A}|} v_a\, w_a \in \overline{\mathbb{R}}_+. \tag{5}$$

We say that their interaction is probabilistic whenever $\langle v, w\rangle \leq 1$. We then define a polar operation on sets of vectors $\mathrm{P} \subseteq \mathbb{R}_+^{|\mathcal{A}|}$ as

$$\mathrm{P}^\perp \triangleq \{w \in \mathbb{R}_+^{|\mathcal{A}|} \text{ s.t. } \forall v \in \mathrm{P}\ \ \langle v, w\rangle \leq 1\}. \tag{6}$$

The probabilistic duality environment/program is then enforced in our model by the closedness condition $P^{\perp\perp} = P$.

**Definition 7** ([4, 14])**.** A *probabilistic coherence space*, or PCS for short, is a pair $\mathcal{A} = (|\mathcal{A}|, \mathrm{P}(\mathcal{A}))$ where $|\mathcal{A}|$ is a countable set called the *web* of $\mathcal{A}$ and $\mathrm{P}(\mathcal{A})$ is a subset of $\mathbb{R}_+^{|\mathcal{A}|}$ satisfying:

1. $\mathrm{P}(\mathcal{A})^{\perp\perp} = \mathrm{P}(\mathcal{A})$,
2. $\forall a \in |\mathcal{A}|, \exists\kappa > 0, \forall v \in \mathrm{P}(\mathcal{A}), v_a \leq \kappa$,
3. $\forall a \in |\mathcal{A}|, \exists\kappa > 0, \kappa e_a \in \mathrm{P}(\mathcal{A})$.

As a side effect, Condition (1) forces $\mathrm{P}(\mathcal{A})$ to be a convex set. Condition (2) requires the projection of $\mathrm{P}(\mathcal{A})$ in any direction to be bounded, while (3) forces $\mathrm{P}(\mathcal{A})$ to cover every direction.[6]

**Example 8.** The set $\mathrm{P}(\texttt{Int})$ of subprobability distributions over $\mathbb{N}$ yields a PCS. In particular, $\mathrm{P}(\texttt{Int})^{\perp\perp} = \mathrm{P}(\texttt{Int})$, its polar being $\mathrm{P}(\texttt{Int})^\perp = [0,1]^{\mathbb{N}}$.

### 4.2 The category PCoh

The *objects* of **PCoh** are the PCSs and the set $\mathbf{PCoh}(\mathcal{A},\mathcal{B})$ of *morphisms from* $\mathcal{A}$ *to* $\mathcal{B}$ is the set of matrices $\phi \in \mathbb{R}_+^{\mathcal{M}_{\mathrm{f}}(|\mathcal{A}|)\times|\mathcal{B}|}$ such that $\forall v \in \mathrm{P}(\mathcal{A}), \phi(v) \in \mathrm{P}(\mathcal{B})$, where

$$\forall b \in |\mathcal{B}|\,,\ (\phi(v))_b \triangleq \sum_{m\in\mathcal{M}_{\mathrm{f}}(|\mathcal{A}|)} \phi_{m,b}\cdot v^m, \tag{7}$$

(see Notation 1 for the definition of $v^m$). Following Section 2, a morphism is presented as a matrix that gathered the coefficients of the power series described by the Equation (7).

The *identity on* $\mathcal{A}$ is given by the matrix

$$\mathrm{Id}^{\mathcal{A}}_{m,a} \triangleq \begin{cases} 1 & \text{if } m = [a],\\ 0 & \text{otherwise.}\end{cases}$$

In fact, we have $\mathrm{Id}^{\mathcal{A}}(v) = v$, for every $v \in \mathrm{P}(\mathcal{A})$.

Let $\phi \in \mathbf{PCoh}(\mathcal{A},\mathcal{B})$ and $\psi \in \mathbf{PCoh}(\mathcal{B},\mathcal{C})$, their composition must satisfy: $\forall v \in \mathrm{P}(\mathcal{A})\,,\ (\psi\circ\phi)(v) = \psi(\phi(v))$. In matricial terms, this comes out as: $\forall c \in |\mathcal{C}|$,

$$\begin{aligned}\sum_{m\in\mathcal{M}_{\mathrm{f}}(|\mathcal{A}|)} (\psi\circ\phi)_{m,c}\cdot v^m &= \sum_{p\in\mathcal{M}_{\mathrm{f}}(|\mathcal{B}|)} \psi_{p,c}\cdot\left(\phi(v)\right)^p\\ &= \sum_{p\in\mathcal{M}_{\mathrm{f}}(|\mathcal{B}|)} \psi_{p,c}\cdot \prod_{b\in\mathrm{Supp}(p)} \Big(\sum_{q\in\mathcal{M}_{\mathrm{f}}(|\mathcal{A}|)} \phi_{q,b}\cdot v^q\Big)^{p(b)}.\end{aligned}$$

[5] $\mathrm{P}(\mathcal{A})$ is not anymore a subset of $[0,1]^{|\mathcal{A}|}$, see discussion in Section 2.

[6] These conditions are introduced in [4] for keeping finite all the scalars involved, yet they are not explicitly stated in the definition of PCS in [14].

Now, to extract the coefficient of the monomial $v^m$, we distribute product over sum. This amounts to choosing a partition of $m = \biguplus(q_{(b,i)})_{(b,i)\in p}$ matching the enumeration $\{(b,i) \text{ s.t. } b \in \mathrm{Supp}(p)\,, i \leq p(b)\}$ of $p$.

Therefore, the *composition* $\psi\circ\phi$ is defined[7] as the matrix coefficients, for $m \in \mathcal{M}_{\mathrm{f}}(|\mathcal{A}|)$ and $c \in |\mathcal{C}|$:

$$(\psi\circ\phi)_{m,c} \triangleq \sum_{p\in\mathcal{M}_{\mathrm{f}}(|\mathcal{B}|)} \psi_{p,c} \sum_{\substack{(m_{(b,i)})_{(b,i)\in p}\\ \text{s.t.}\\ m=\biguplus(m_{(b,i)})}} \prod_{(b,i)\in p} \phi_{m_{(b,i)},b}. \tag{8}$$

### 4.3 PCoh is Cartesian Closed

The *cartesian product* of any countable family $(\mathcal{A}_i)_{i\in I}$ of PCSs is:

$$\left|\prod_{i\in I}\mathcal{A}_i\right| \triangleq \bigcup_{i\in I}(\{i\}\times|\mathcal{A}_i|),$$

$$\mathrm{P}\left(\prod_{i\in I}\mathcal{A}_i\right) \triangleq \left\{ v \in (\mathbb{R}_+)^{|\prod_{i\in I}\mathcal{A}_i|} \text{ s.t. } \begin{array}{l}\forall i\in I,\\ \pi_i(v) \in \mathrm{P}(\mathcal{A}_i)\end{array}\right\},$$

where $\pi_i(v)$ is the vector in $\mathbb{R}_+^{|\mathcal{A}_i|}$ denoting the $i$-th component of $v$, i.e. $\left(\pi_i(v)\right)_a \triangleq v_{(i,a)}$.

The *$j$-th projection* $\mathrm{Pr}^j \in \mathbf{PCoh}(\prod_{i\in I}\mathcal{A}_i, \mathcal{A}_j)$ and the *product* $\langle\phi_i\rangle_{i\in I} \in \mathbf{PCoh}(\mathcal{B}, \prod_{i\in I}\mathcal{A}_i)$ are given by:

$$\mathrm{Pr}^j_{m,a} \triangleq \begin{cases}1 & \text{if } m = [(j,a)],\\ 0 & \text{otherwise.}\end{cases} \qquad (\langle\phi_i\rangle_{i\in I})_{p,(j,a)} = (\phi_j)_{p,a}$$

The *terminal object* $\mathbf{1}$ is given by the empty product $(\emptyset, \{\mathbf{0}\})$. Notice that the set of *points* of a PCS $\mathcal{A}$, i.e. the set $\mathbf{PCoh}(\mathbf{1},\mathcal{A})$, is isomorphic to the convex set $\mathrm{P}(\mathcal{A})$.

**Notation 9.** We write $\mathcal{A}_1\times\mathcal{A}_2$ for the binary product: in the sequel, we present any $v \in \mathrm{P}(\mathcal{A}_1\times\mathcal{A}_2)$ as the pair $(\pi_1(v),\pi_2(v)) \in \mathrm{P}(\mathcal{A}_1)\times\mathrm{P}(\mathcal{A}_2)$ of its components.

Notice that the set $\mathcal{M}_{\mathrm{f}}\left(\left|\prod_{i\in I}\mathcal{A}_i\right|\right)$ is isomorphic to the set-theoretic cartesian product $\prod_{i\in I}\left(\mathcal{M}_{\mathrm{f}}(|\mathcal{A}_i|)\right)$, via the map associating any $m \in \mathcal{M}_{\mathrm{f}}\left(\left|\prod_{i\in I}\mathcal{A}_i\right|\right)$ with the $I$-indexed family $(m_i)_{i\in I}$ defined as $m_i(a) \triangleq m(i,a)$. This means that any morphism $\phi \in \mathbf{PCoh}(\prod_{i\in I}\mathcal{A}_i, \mathcal{B})$ can be presented as a matrix indexed by sequences in $\left(\prod_{i\in I}\mathcal{M}_{\mathrm{f}}(|\mathcal{A}_i|)\right)\times|\mathcal{B}|$.

The *object of morphisms* is defined as

$$|\mathcal{A}\Rightarrow\mathcal{B}| \triangleq \mathcal{M}_{\mathrm{f}}(|\mathcal{A}|)\times|\mathcal{B}|\,, \qquad \mathrm{P}(\mathcal{A}\Rightarrow\mathcal{B}) \triangleq \mathbf{PCoh}(\mathcal{A},\mathcal{B}).$$

**PCoh** is then turned into a cartesian closed category by the *evaluation* $\mathrm{Ev} \in \mathbf{PCoh}((\mathcal{A}\Rightarrow\mathcal{B})\times\mathcal{A},\mathcal{B})$ and the *curryfication* $\mathrm{Cur}(\phi) \in \mathbf{PCoh}(\mathcal{C},\mathcal{A}\Rightarrow\mathcal{B})$, for every $\phi \in \mathbf{PCoh}(\mathcal{C}\times\mathcal{A},\mathcal{B})$, defined as:

$$\mathrm{Ev}_{(m,p),a} \triangleq \begin{cases}1 & \text{if } m = [(p,v)],\\ 0 & \text{otherwise,}\end{cases} \qquad \mathrm{Cur}(\phi)_{m,(p,b)} \triangleq \phi_{(m,p),b}$$

Notice that the above equations use Notation 9, e.g. representing a multiset in $\mathcal{M}_{\mathrm{f}}(|(\mathcal{A}\Rightarrow\mathcal{B})\times\mathcal{A}|)$ as a pair $(m,p)$ of multisets in $\mathcal{M}_{\mathrm{f}}(|\mathcal{A}\Rightarrow\mathcal{B}|)\times\mathcal{M}_{\mathrm{f}}(|\mathcal{A}|)$.

Actually, the category **PCoh** is *well-pointed* in the sense that the equality of morphisms is extensional, i.e. given two matrices $\phi,\psi \in \mathbf{PCoh}(\mathcal{A},\mathcal{B})$, if for every $v \in \mathrm{P}(\mathcal{A})$, $\phi(v) = \psi(v)$, then $\phi = \psi$. To sum up, we have:

[7] The definition of Equation (8) is due to [17]. On a side note, remark that in [4], the authors use another formulation in which identical summands produced by different partitions are gathered. This gives rise to multinomial coefficients that are hidden in the sums of the present formulation. However, the two definitions give rise to the same matrix.

**Proposition 10** ([4, §1.6])**.** ***PCoh*** *is a well pointed cartesian closed category.*

### 4.4 Object of numerals and Cpo-Enrichement

The *object of numerals* of **PCoh** is the PCS $\mathcal{I}nt \triangleq (\mathbb{N}, \mathrm{P}(\mathtt{Int}))$ equipped with the morphisms $\mathrm{z} \in \mathbf{PCoh}(\mathbf{1}, \mathcal{I}nt) \simeq \mathrm{P}(\mathcal{I}nt)$, pred, succ $\in \mathbf{PCoh}(\mathcal{I}nt, \mathcal{I}nt)$, and ifz $\in \mathbf{PCoh}(\mathcal{I}nt \times \mathcal{I}nt \times \mathcal{I}nt, \mathcal{I}nt)$ defined as

$$\mathrm{z}_n = \delta_{0,n}, \qquad \mathrm{pred}_{m,n} = \delta_{m,[n+1]}, \qquad \mathrm{succ}_{m,n+1} = \delta_{m,[n]},$$

$$\mathrm{ifz}_{(m,p,q),n} = \begin{cases} 1 & \text{if } (m,p,q) = ([0],[n],[]), \\ & \text{or } (m,p,q) = ([k+1],[],[n]), \\ 0 & \text{otherwise.} \end{cases}$$

The natural order on $\mathbb{R}_+$ enriches **PCoh** with a *cpo-structure*, defined componentwise on morphisms: i.e., given $\phi, \psi \in \mathbf{PCoh}(\mathcal{A}, \mathcal{B})$

$$\phi \le \psi \quad \text{iff} \quad \forall m \in \mathcal{M}_{\mathrm{f}}(|\mathcal{A}|), \forall b \in \mathcal{M}_{\mathrm{f}}(|\mathcal{B}|), \phi_{m,b} \le \psi_{m,b}.$$

The matrix $\mathbf{0}$ is the minimum element and the lub of a directed net $(\phi_d)_{d \in D}$ is then given by

$$\big(\sup_{d \in D}(\phi_d)\big)_{m,b} \triangleq \sup_{d \in D}\big((\phi_d)_{m,b}\big),$$

*Remark* 11. The componentwise order on matrices is not extensional, i.e. there are $\phi, \psi \in \mathbf{PCoh}(\mathcal{A}, \mathcal{B})$ such that $\forall v \in \mathrm{P}(\mathcal{A})$, $\phi(v) \le \psi(v)$, but $\phi \not\le \psi$. For example, take $\phi, \psi \in \mathbf{PCoh}(\mathcal{I}nt, \mathcal{I}nt)$:

$$\phi_{m,n} \triangleq \begin{cases} 1 & \text{if } m = [k], n = 0, \\ 0 & \text{otherwise.} \end{cases} \quad \psi_{m,n} \triangleq \begin{cases} 1 & \text{if } m = [], n = 0, \\ 0 & \text{otherwise.} \end{cases}$$

Although $\phi \not\le \psi$, for every subprobability distribution of natural numbers $v \in \mathrm{P}(\mathcal{I}nt)$, we get $\phi(v) = (\sum_k v_k, 0, 0, \dots) \le (1, 0, 0, \dots) = \psi(v)$. In fact, we will use this mismatch for disproving the inequality full abstraction in Section 6.

### 4.5 PCoh is an Adequate Model of PPCF

The model of PPCF is obtained by extending the usual categorical interpretation of PCF to `rand`.

With a type $A$, we associate a PCS $\mathcal{A}$, by induction on the type: $\mathtt{Int} \mapsto \mathcal{I}nt$ and $A \Rightarrow B \mapsto \mathcal{A} \Rightarrow \mathcal{B}$.

Let $\Gamma = x_1 : A_1, \dots, x_n : A_n$. The interpretation of a judgment $\Gamma \vdash M : B$ is a morphism $[\![M]\!]^\Gamma$ of $\mathbf{PCoh}(\prod_{i=1}^n \mathcal{A}_i, \mathcal{B})$, defined in Figure 2 by structural induction on the unique derivation of $\Gamma \vdash M : B$.

The fix-point operator $\mathrm{fix}(M)$ is the lub of its approximants, given by induction by

$$\mathrm{fix}^0 \phi \triangleq \mathbf{0}, \qquad \mathrm{fix}^{n+1} \phi \triangleq \mathrm{Ev} \circ \langle \phi, \mathrm{fix}^n \phi \rangle.$$

The operator `rand` is defined by using (a multiset variant of) the matrix `Rand` of Equation (1):

$$[\![\mathtt{rand}]\!]_{m,k} \triangleq \begin{cases} \frac{1}{n} & \text{if } m = [n], k < n, \\ 0 & \text{otherwise.} \end{cases}$$

Together with the categorical interpretation of a term, we describe its action on the vectors in the convex set associated with its input type. Notice that, since the category is well pointed of the category, this action univocally determines the interpretation of the term. Notice also that the matrices interpreting the basic constructs in Figure 2 have 0, 1 coefficients, except for $[\![\mathtt{rand}]\!]$ which is interpreted as the random function that introduces rational numbers in $[0, 1]$. Coefficients greater than 1 may be produced by composition of morphisms (Equation 8).

Thanks to Notation 9, $[\![M]\!]^\Gamma$ can be described as a vector indexed by a tuple $(\vec{m}, b)$ of $\vec{m} \in \mathcal{M}_{\mathrm{f}}(|\Gamma|)$ and a web element $b \in |\mathcal{B}|$. This convention will be used hereafter.

**Example 12.** If `Coin` is identified with $(\mathtt{rand})\, \underline{2}$, then the interpretation of `Once` and `Twice` are the matrices given in Section 2.

Consider the term $\Omega_A \triangleq \mathsf{fix}(\lambda x^A.x)$. Notice that $[\![\lambda x^A.x]\!]$ is different from zero only on the web elements of the form $([a], a)$, so that $\mathrm{fix}^n\, [\![\lambda x^A.x]\!]\, (\mathbf{0}) = \mathbf{0}$ for any natural number $n$. We conclude $[\![\Omega_A]\!] = \mathbf{0}$, as expected.

Consider now the term $P \triangleq \mathsf{fix}(\lambda x^{\mathtt{Int}}.x \oplus \underline{0})$. We have that $[\![\lambda x^{\mathtt{Int}}.x \oplus \underline{0}]\!]_{(m,a)} = \frac{1}{2}[\![x]\!]^m_a + \frac{1}{2}[\![\underline{0}]\!]^m_a$, i.e. it is equal to $\frac{1}{2}$ when $m = [a]$, or when $m = [\,]$ and $a = 0$, otherwise it is equal to 0. This means that $\mathrm{fix}^n\, [\![\lambda x^{\mathtt{Int}}.x \oplus \underline{0}]\!]\, (\mathbf{0}) = \frac{1}{2} + \frac{1}{4} + \dots + \frac{1}{2^n}$, for any $n > 0$. We conclude that $[\![P]\!] \triangleq \sup_n \mathrm{fix}^n\, [\![\lambda x^{\mathtt{Int}}.x \oplus \underline{0}]\!]\, (\mathbf{0}) = 1$.

**Proposition 13** (Soundness [4])**.** *The semantics is invariant under reduction, i.e. for every* $\Gamma \vdash M : B$*:*

$$[\![M]\!]^\Gamma = \sum_N \mathrm{P{\scriptsize ROBA}}_{M,N}\, [\![N]\!]^\Gamma.$$

*Proof (Sketch).* The invariance under reduction rules of the standard PCF redexes follows by cartesian closedness and the cpo-enrichement of **PCoh**. The soundness of the reduction of $(\mathtt{rand})\, \underline{n}$ is straight from the definition of $[\![\mathtt{Rand}]\!]$. The soundness of the context rules depends on the fact that the interpretation of a context is linear in the argument associated with the fired redex. For example, the soundness of the context rule associated with application depends on the equality: $\mathrm{Ev} \circ \langle \kappa\phi + \mu\psi, \xi \rangle = \kappa(\mathrm{Ev} \circ \langle \phi, \xi \rangle) + \mu(\mathrm{Ev} \circ \langle \psi, \xi \rangle)$ ☐

**Theorem 14** (Adequacy [4])**.** *Let $M$ be a closed term of type* `Int`*. Then, $[\![M]\!]$ is the sub-probability distribution on $\mathbb{N}$ such that*

$$\forall n \in \mathbb{N},\ [\![M]\!]_n = \mathrm{P{\scriptsize ROBA}}^\infty_{M,\underline{n}}.$$

*Remark* 15. Adequacy allows one to prove that specific primitives are not definable in the language. A noteworthy example is the *parallel or* function [27]. In our setting, this should be a closed term $\mathtt{por} : \mathtt{Int} \Rightarrow \mathtt{Int} \Rightarrow \mathtt{Int}$ such that $\mathrm{P{\scriptsize ROBA}}^\infty_{((\mathtt{por})\underline{0})\underline{0},\underline{0}}$, $\mathrm{P{\scriptsize ROBA}}^\infty_{((\mathtt{por})\underline{1})\Omega,\underline{1}}$ and $\mathrm{P{\scriptsize ROBA}}^\infty_{((\mathtt{por})\Omega)\underline{1},\underline{1}}$ are equal to 1. By adequacy, $[\![\mathtt{por}]\!]\,(e_0)(e_0)_1 = [\![\mathtt{por}]\!]\,(e_1)(\mathbf{0})_0 = [\![\mathtt{por}]\!]\,(\mathbf{0})(e_1)_0 = 1$. By definition of a morphism in **PCoh**, $[\![\mathtt{por}]\!]\,(e_0)(e_0)$ must be a subprobability distribution, hence $[\![\mathtt{por}]\!]\,(e_0)(e_0)_0 = 0$. That implies $[\![\mathtt{por}]\!]_{([],[]),0} = 0$. On the other hand, $[\![\mathtt{por}]\!]\,(e_1)(e_1)_0 \ge [\![\mathtt{por}]\!]\,(e_1)(\mathbf{0})_0 + [\![\mathtt{por}]\!]\,(\mathbf{0})(e_1)_0 - [\![\mathtt{por}]\!]_{[],[],0} = 2$, which contradicts that $[\![\mathtt{por}]\!]\,(e_1)(e_1)$ is a subprobability distribution. We conclude that `por` is not a term of PPCF.

However, let us mention that the *Gustave* function is a valid morphism of **PCoh** (see [13]).

Full abstraction extends to higher-order types the perfect matching syntax / semantics stated by adequacy on `Int`. One direction of full abstraction is indeed a consequence of Theorem 14.

**Corollary 16** (Abstraction)**.** *Given $\Gamma \vdash M : A$, and $\Gamma \vdash N : A$, we have that $[\![M]\!] \le [\![N]\!]$ implies $M \sqsubseteq_\Gamma N$. In particular, $[\![M]\!] = [\![N]\!]$ implies $M \equiv_\Gamma N$.*

*Proof.* By induction on $C[\,]$, one proves that $[\![M]\!] \le [\![N]\!]$ implies $[\![C[M]]\!] \le [\![C[N]]\!]$. Then the result follows from Theorem 14. ☐

## 5. Full Abstraction

We prove *equational full abstraction* (Theorem 27), that is the converse of the part of Corollary 16 dealing with equality. This is a straightforward consequence of Lemma 26 stating that for any

$$\Gamma \xrightarrow{\mathrm{Pr}^i} A_i \qquad \vec{v} \longmapsto \vec{v}_i \qquad [\![x_i]\!]^\Gamma$$

$$\Gamma \xrightarrow{\mathrm{Cur}([\![M]\!]^{\Gamma,x:B})} A \Rightarrow B \qquad \vec{v} \longmapsto \mathrm{Cur}(u \mapsto \phi(\vec{v},u)) \qquad [\![\lambda x^B.M]\!]^\Gamma, \text{ with } \phi = [\![M]\!]^{\Gamma,x:B}$$

$$\Gamma \xrightarrow{\langle [\![M]\!]^\Gamma, [\![N]\!]^\Gamma \rangle} (A \Rightarrow B) \,\&\, A \xrightarrow{\mathrm{Ev}} B \qquad \vec{v} \longmapsto \phi(\vec{v})(\psi(\vec{v})) \qquad [\![(M)\,N]\!]^\Gamma, \text{ with } \phi = [\![M]\!]^\Gamma, \psi = [\![N]\!]^\Gamma$$

$$\Gamma \xrightarrow{\mathbb{T}} \mathbf{1} \xrightarrow{\mathrm{z}} \mathtt{Int} \qquad \vec{v} \longmapsto (1,0,0,\ldots) \qquad [\![\underline{0}]\!]^\Gamma$$

$$\Gamma \xrightarrow{[\![M]\!]^\Gamma} \mathtt{Int} \xrightarrow{\mathrm{succ}} \mathtt{Int} \qquad \vec{v} \longmapsto (0,w_0,w_1,\ldots) \qquad [\![\mathsf{s}(M)]\!]^\Gamma, \text{ with } w = [\![M]\!]^\Gamma(\vec{v})$$

$$\Gamma \xrightarrow{[\![M]\!]^\Gamma} \mathtt{Int} \xrightarrow{\mathrm{pred}} \mathtt{Int} \qquad \vec{v} \longmapsto (w_1,w_2,\ldots) \qquad [\![\mathsf{p}(M)]\!]^\Gamma, \text{ with } w = [\![M]\!]^\Gamma(\vec{v})$$

$$\Gamma \xrightarrow{\mathbb{T}} \mathbf{1} \xrightarrow{\mathtt{rand}} \mathtt{Int} \Rightarrow \mathtt{Int} \qquad \vec{v} \longmapsto [\![\mathtt{rand}]\!] \qquad [\![\mathtt{rand}]\!]^\Gamma$$

$$\Gamma \xrightarrow{\langle [\![M]\!]^\Gamma, [\![N]\!]^\Gamma, [\![N']\!]^\Gamma \rangle} \mathtt{Int} \times A \times A \xrightarrow{\mathrm{ifz}} \mathtt{Int} \qquad \vec{v} \longmapsto w_0 u + \sum_{n=1}^{\infty} w_n u' \qquad [\![\mathsf{if}(M,N,P)]\!]^\Gamma, \text{ with } w = [\![M]\!]^\Gamma(\vec{v}), u = [\![N]\!]^\Gamma(\vec{v}), u' = [\![N']\!]^\Gamma(\vec{v})$$

$$\Gamma \xrightarrow{\sup_n \mathrm{fix}^n([\![M]\!]^\Gamma)} A \qquad \vec{v} \longmapsto \sup_n \mathrm{fix}^n(w) \qquad [\![\mathsf{fix}(M)]\!]^\Gamma, \text{ with } w = [\![M]\!]^\Gamma(\vec{v})$$

**Figure 2:** The standard semantics of PPCF terms together with its action on $\vec{v} \in \mathrm{P}(\Gamma)$.

closed terms $M$ and $N$ having different interpretations in **Pcoh**, there is a *testing* term $P$ such that $(P)\,M$ and $(P)\,N$ reduce to $\underline{0}$ with different probabilities. Let us outline the path to this result.

With any web element $a$, we associate a testing term $\mathcal{P}(a)$ that is described in Figure 5 and that reminds the contexts used in [3]. $\mathcal{P}(a)$ is not an ordinary term of PPCF since its construction uses a random operator, weighted by a list $\vec{X}$ of *formal parameters*. However, a parameterized term becomes an ordinary PPCF term when we substitute $\vec{X}$ by a list $\vec{\kappa}$ of rationals in $[0,1]$. Following [6], we introduce in Figure 3 an *intersection type system* that defines semantics of parameterized terms. This interpretation is a formal power series over $\vec{X}$ (Definition 21). Moreover, parameterized semantics is compatible with **Pcoh** through substitution of parameters (Lemma 20).

Now, assume that $M$ and $N$ are interpreted by different matrices and pick a web element $a$ such that $[\![M]\!]_a \neq [\![N]\!]_a$. Then, the semantics of the parameterized terms $(\mathcal{P}(a))\,M$ and $(\mathcal{P}(a))\,N$ are distinct formal power series. Indeed, Lemma 24 implies that they differ at least on one coefficient.

Finally, Lemma 25 ensures the existence of a list $\vec{\kappa}$ of rationals in $[0,1]$ on which these power series disagree. So, the substitution of $\vec{X}$ by $\vec{\kappa}$ in $\mathcal{P}(a)$ produces a testing context separating $M$ and $N$.

We first describe *parameterized* PPCF in Subsection 5.1. Then, testing terms are presented in Subsection 5.2. We conclude in Subsection 5.3 with the full abstraction theorem.

### 5.1 Parameterized PPCF

Let $\mathfrak{P}$ be a denumerable set of formal parameters. $X, Y, Z$ range over parameters in $\mathfrak{P}$.

The *grammar* of parameterized PPCF is an extension of PPCF (Figure 1(a)) by multiplication of terms by parameters:

$$\mathrm{PPCF}^{\mathfrak{P}} ::= \cdots \mid X \cdot M, \text{ where } X \in \mathfrak{P}.$$

The *simple type* of $X \cdot M$ under a context $\Gamma$ is $A$ whenever $\Gamma \vdash M : A$ in PPCF (Figure 1(b)).

The *substitution* of parameters by scalars let us recover ordinary terms from parameterized ones. More precisely, let $M \in \mathrm{PPCF}^{\mathfrak{P}}$ and $\frac{n}{m} \in [0,1]$ be a rational number. We define $M\left[\frac{n}{m}/X\right]$ as the term obtained by replacing in $M$ any subterm of shape $X \cdot N$ with

$$\mathtt{choose}(N^n, \Omega^{m-n}) \triangleq \mathtt{choose}(\underbrace{N,\ldots,N}_{n \text{ times}}, \underbrace{\Omega,\ldots,\Omega}_{m-n \text{ times}}) \tag{9}$$

(see Figure 1(d) for $\mathtt{choose}$ definition). Substitution is then generalized to lists $\vec{X}$ of parameters and $\vec{\kappa}$ of rationals as $M\left[\vec{\kappa}/\vec{X}\right]$.

**Fact 17.** *If $\vec{X}$ is the list of all parameters in $M$ and $\vec{\kappa}$ a list of rational numbers in $[0,1]$, then $M\left[\vec{\kappa}/\vec{X}\right]$ is a term of* PPCF.

The *semantics* of $\mathrm{PPCF}^{\mathfrak{P}}$ is a refinement of PPCF semantics taking into account parameters. Yet, for the sake of the full abstraction proof, we give a different presentation and use a *weighted intersection type system*. Roughly speaking, types are *web elements* and with each type derivation $\pi$, we associate a *weight* $\omega(\pi)$ which is a positive *monomial*, i.e. a product of rationals in $[0,1]$ and of finitely many parameters. Then, the interpretation $[\![M]\!]$ of a $\mathrm{PPCF}^{\mathfrak{P}}$ term $M$ is a matrix indexed by web elements. For each web element, there can be several type derivations and the corresponding coefficient is the sum of their weights.

More precisely, Figure 3 describes the rules for constructing a derivation $\pi :: \Gamma^\bullet \vdash_\alpha M : a$ of what we call a *web judgment* $\Gamma^\bullet \vdash_\alpha M : a$. Notice that $\Gamma \vdash M : A$ is a valid simple type judgment, $a \in |\mathcal{A}|$ and $\alpha$ is a monomial. Besides, a *web context* $\Gamma^\bullet$ is defined as a function mapping any typed variable $x^C$ occurring in $\Gamma$ to a finite multiset $m \in \mathcal{M}_{\mathrm{f}}(|\mathcal{C}|)$ of web elements and mapping variables non-appearing in $\Gamma$ to the empty multiset. For instance, for any $\Gamma = x_1 : C_1 \ldots x_n : C_n$ and $\vec{m} \in \mathcal{M}_{\mathrm{f}}(|\Gamma|)$, $\Gamma^{\vec{m}}$ denotes the web context $x_i^{C_i} \mapsto \vec{m}_i$ for $1 \leq i \leq n$. Disjoint unions $\Gamma^\bullet \uplus \Delta^\bullet$ of web contexts are defined pointwise.

**Fact 18.** *If $\Gamma^\bullet \vdash_\alpha M : a$ is derivable and $\Gamma^\bullet(x)$ is a non-empty multiset, then $x$ is free in $M$.*

Rules $\mathtt{app}$ and $\mathtt{fix}$ deserve some comments. Application of $\mathrm{PPCF}^{\mathfrak{P}}$ terms is interpreted following Equation (8) that defines composition. Remark that indices of $\mathtt{app}$ mainly coincide with the indices of the sums in (8), with one more difficulty since we have to split contexts. Now, $\mathtt{fix}$ is derived from the rule $\mathtt{app}$ and from

$$\frac{}{x^A : [a] \vdash_1 x : a}\ \texttt{var} \qquad \frac{}{\vdash_1 \underline{n} : n}\ \texttt{nat} \qquad \frac{k < n}{\vdash_{\frac{1}{n}} \texttt{rand} : ([n], k)}\ \texttt{rand} \qquad \frac{\Gamma^\bullet, x^A : m \vdash_\alpha M : a}{\Gamma^\bullet \vdash_\alpha \lambda x^A.M : (m, a)}\ \texttt{abs}$$

$$\frac{\Gamma^{\bullet\prime} \vdash_\alpha M : (m, b) \qquad \forall (a,i) \in m,\quad \Gamma^\bullet_{(a,i)} \vdash_{\beta_{(a,i)}} N : a}{\Gamma^\bullet \vdash_{\alpha \prod_{(a,i)\in m} \beta_{(a,i)}} (M)\, N : b}\ \texttt{app}_{(m, (\Gamma^\bullet_{(a,i)})_{(a,i)\in m})} \quad \text{s.t.} \begin{cases} m \in \mathcal{M}_{\mathrm{f}}(|\mathcal{A}|) \\ \Gamma^{\bullet\prime} \uplus \biguplus_{(a,i)\in m} \Gamma^\bullet_{(a,i)} = \Gamma^\bullet \end{cases}$$

$$\frac{\Gamma^{\bullet\prime} \vdash_\alpha M : (m, b) \qquad \forall (a,i) \in m,\quad \Gamma^\bullet_{(a,i)} \vdash_{\beta_{(a,i)}} \mathsf{fix}(M) : a}{\Gamma^\bullet \vdash_{\alpha \prod_{(a,i)\in m} \beta_{(a,i)}} \mathsf{fix}(M) : b}\ \texttt{fix}_{(m, (\Gamma^\bullet_{(a,i)})_{(a,i)\in m})} \quad \text{s.t.} \begin{cases} m \in \mathcal{M}_{\mathrm{f}}(|\mathcal{A}|) \\ \Gamma^{\bullet\prime} \uplus \biguplus_{(a,i)\in m} \Gamma^\bullet_{(a,i)} = \Gamma^\bullet \end{cases}$$

$$\frac{\Gamma^\bullet \vdash_\alpha M : n+1}{\Gamma^\bullet \vdash_\alpha \mathsf{p}(M) : n}\ \texttt{pred} \qquad \frac{\Gamma^\bullet \vdash_\alpha M : n}{\Gamma^\bullet \vdash_\alpha \mathsf{s}(M) : n+1}\ \texttt{succ}$$

$$\frac{\Gamma^\bullet \vdash_\beta M : 0 \qquad \Delta^\bullet \vdash_\alpha N : a}{\Gamma^\bullet \uplus \Delta^\bullet \vdash_{\beta\alpha} \mathsf{if}(M, N, P) : a}\ \texttt{if}_0 \qquad \frac{\Gamma^\bullet \vdash_\beta M : n+1 \qquad \Delta^\bullet \vdash_\alpha P : a}{\Gamma^\bullet \uplus \Delta^\bullet \vdash_{\beta\alpha} \mathsf{if}(M, N, P) : a}\ \texttt{if}_s \qquad \frac{\Gamma^\bullet \vdash_\alpha M : a}{\Gamma^\bullet \vdash_{\alpha X} X \cdot M : a}\ \texttt{par}$$

**Figure 3:** Semantics of parameterized PPCF.

$\mathsf{fix}(M) \xrightarrow{1} (M)\,\mathsf{fix}(M)$. To illustrate this point and the followings, we detail Examples 22 and 23 at the end of this subsection.

**Fact 19.** *If $\Gamma \vdash M : A$ then $\sum_{\pi :: \Gamma^{\vec{m}} \vdash M : a} \omega(\pi)$ is a power series with finitely many variables in $\mathfrak{P}$ and non negative coefficients. Thus, for any list $\vec{\kappa}$ of non negative reals, $\big(\sum_{\pi :: \Gamma^{\vec{m}} \vdash M : a} \omega(\pi)\big)(\vec{\kappa})$ is well defined.*

The following fundamental lemma ensures the soundness of the weighted type system that computes the semantics of $\mathrm{PPCF}^{\mathfrak{P}}$.

**Lemma 20** (Soundness)**.** *Let $M$ be a term of $\mathrm{PPCF}^{\mathfrak{P}}$ such that $\Gamma \vdash M : B$ and $\vec{X}$ lists its parameters. For any $\vec{p} \in \mathcal{M}_f(|\Gamma|)$, $b \in |\mathcal{B}|$, and any list $\vec{\kappa}$ of rational numbers in $[0,1]$,*

$$\left[\!\left[ M \left[\vec{\kappa}/\vec{X}\right] \right]\!\right]^{\Gamma}_{\vec{p},b} = \Big( \sum_{\pi :: \Gamma^{\vec{p}} \vdash M : b} \omega(\pi) \Big)(\vec{\kappa}) \tag{10}$$

*Proof.* We prove Equation (10) by structural induction on $M$, using rules of Figure 3 and the definition of the categorical interpretation of a term in PPCF. We consider the term $\mathsf{fix}(M)$ greater than $(M^n)\, y$ for any natural number $n > 0$ and variable $y$. In fact, we are using the induction on the ordinal $\omega^2$.

Most cases follow directly from induction hypothesis. We only detail three cases: (i) multiplication with a parameter, (ii) application, (iii) fix-point.

(i) Assume $M = X_i \cdot N$ and $\kappa_i = \frac{n}{m}$. Then (using Equation (9)) we get: $M\left[\vec{\kappa}/\vec{X}\right] = \texttt{choose}\big(N\left[\vec{\kappa}/\vec{X}\right]^n, \Omega^{m-n}\big)$. Besides, $[\![\Omega]\!] = \mathbf{0}$ (Example 12), so $\left[\!\left[ M \left[\vec{\kappa}/\vec{X}\right] \right]\!\right]^{\Gamma}_{\vec{p},b} = \frac{n}{m} \left[\!\left[ N \left[\vec{\kappa}/\vec{X}\right] \right]\!\right]^{\Gamma}_{\vec{p},b}$.

Now, any derivation $\tau :: \Gamma^{\vec{p}} \vdash M\left[\vec{\kappa}/\vec{X}\right] : b$ consists of a sequence of $h \leq n$ rules $\texttt{if}_s$ with on top, the rule $\texttt{if}_0$ with a derivation $\pi :: \Gamma^{\vec{p}} \vdash N\left[\vec{\kappa}/\vec{X}\right] : b$ as right premise. Indeed, the subterm $\Omega$ has no web type (see Example 23). When $\pi$ is fixed, $\omega(\tau) = \frac{\omega(\pi)}{m}$ and there are as many derivations $\tau$ as many choices of $h \in \{1, \dots, n\}$. So, we compute

$$\sum_{\tau :: \Gamma^{\vec{p}} \vdash M : b} \omega(\tau) \left[\vec{\kappa}/\vec{X}\right] = \frac{n}{m} \Big( \sum_{\pi :: \Gamma^{\vec{p}} \vdash N : b} \omega(\pi) \left[\vec{\kappa}/\vec{X}\right] \Big),$$

which is equal to $\frac{n}{m} \left( \left[\!\left[ N \left[\vec{\kappa}/\vec{X}\right] \right]\!\right]^{\Gamma}_{\vec{p},b} \right)$ by induction hypothesis. We conclude by gathering the equalities of the two paragraphs.

(ii) Assume $M = (N)\, P$. Thanks to the categorical semantics, $[\![(N)\, P]\!]^{\Gamma}_{\vec{p},b} = \big(\mathrm{Ev} \circ \langle [\![N]\!]^{\Gamma}, [\![P]\!]^{\Gamma} \rangle\big)_{\vec{p},b}$. Then by definition of composition (Equation (8)) and left linearity of Ev, $[\![(N)\, P]\!]^{\Gamma}_{\vec{p},b}$ is

$$\sum_{\substack{m \in \mathcal{M}_{\mathrm{f}}(|\mathcal{A}|), \\ (\vec{p}_{(a,i)})_{(a,i)\in m} \text{ s.t.} \\ \biguplus_{(a,i)\in m} \vec{p}_{(a,i)} \subseteq \vec{p}}} [\![N]\!]^{\Gamma}_{\vec{p} \setminus \biguplus \vec{p}_{(a,i)}, (m,b)} \prod_{(a,i)\in m} [\![P]\!]^{\Gamma}_{\vec{p}_{(a,i)}, a}.$$

By induction hypothesis and distributing product over sum, we get

$$\sum_{\substack{m \in \mathcal{M}_{\mathrm{f}}(|\mathcal{A}|), \\ (\vec{p}_{(a,i)})_{(a,i)\in m} \\ \text{s.t. } \biguplus \vec{p}_{(a,i)} \subseteq \vec{p}}} \ \sum_{\substack{\tau :: \vec{q} \vdash N : (m,b) \\ \text{where} \\ \vec{q} = \vec{p} \setminus \biguplus \vec{p}_{(a,i)}}} \omega(\tau) \sum_{\substack{(\pi_{(a,i)})_{(a,i)\in m} \\ \text{s.t. } \forall (a,i) \in m, \\ \pi_{(a,i)} :: \vec{p}_{(a,i)} \vdash P : a}} \ \prod_{(a,i)\in m} \omega(\pi_{(a,i)})$$

Although the indices of the sums might be frightening, they precisely describe all possible derivations of $\Gamma^{\vec{p}} \vdash (N)\, P : b$. Namely, the first sum defines the label of the terminal application rule, while the other sums give the choices of the derivations of their premises. The total weight is then the product of premise weights.

(iii) Assume $M = \mathsf{fix}(N)$. Any derivation $\pi :: \Gamma^{\vec{p}} \vdash \mathsf{fix}(N) : b$ ends with a cluster of $\texttt{fix}$ rules (see Example 23). For $n \in \mathbb{N}^*$, let $\Pi^n$ be the set of derivations whose cluster height is at most $n$.

Now, remark that a derivation in $\Pi^n$ can be transformed into a derivation $\tau :: \Gamma^{\vec{p}}, y^B : [\,] \vdash (N^n)\, y : b$ (where $y$ is fresh) by replacing $\texttt{fix}$ rules with $\texttt{app}$ rules (keeping labels) and each occurrence of $\mathsf{fix}(N)$ with $\big(N^h\big)\, y$ for the suitable $h \leq n$. Besides, this transformation preserves weights. Therefore:

$$\sum_{\pi \in \Pi^n} \omega(\pi) = \sum_{\tau :: \Gamma^{\vec{p}}, y^B : [\,] \vdash (N^n) y : b} \omega(\tau)$$

Since $[\![\mathsf{fix}(N)]\!]^{\Gamma}_{\vec{p},b} = \bigvee_{n\in\mathbb{N}} [\![(N^n)\,y]\!]^{\Gamma,y}_{(\vec{p},[\,]),b}$, and by induction hypothesis: $[\![(N^n)\,y]\!]^{\Gamma,y}_{(\vec{p},[\,]),b} = \bigvee_{n\in\mathbb{N}} \sum_{\tau::\Gamma^{\vec{p}},y^B:[\,]\vdash(N^n)y:b}\omega(\tau)$,

$$[\![\mathsf{fix}(N)]\!]^{\Gamma}_{\vec{p},b} = \bigvee_{n\in\mathbb{N}}\sum_{\pi\in\Pi^n}\omega(\pi) = \sum_{\pi\in\bigcup_n\Pi^n}\omega(\pi) = \sum_{\pi::\Gamma^{\vec{p}}\vdash\mathsf{fix}(N):b}\omega(\pi)$$

□

According to this result, we generalize the semantics notation to parameterized PPCF:

**Definition 21.** For any term $M$ in PPCF$^{\mathfrak{P}}$ with $n$ parameters,

$$[\![M]\!]^{\Gamma^\bullet}_a \triangleq \sum_{\pi::\Gamma^\bullet\vdash M:a}\omega(\pi) \tag{11}$$

It is a power series whose domain of convergence contains $[0,1]^n$.

The weighted type assignment system of Figure 3 has an interest by its own. It turns computation of semantics of terms (and hence its observational behavior) into a proof search problem.

The reader can convince himself that this approach is useful by computing the semantics of the following examples by applying directly rules of Figure 2.

**Example 22.** Let $k$ and $k'$ be non negative integers. Let us consider the possible derivations of the application $(M)\,N$ with

$$M \triangleq \lambda x^{\mathtt{Int}}.\mathsf{if}(x,\Omega_{\mathtt{Int}},\mathsf{if}(x,\Omega_{\mathtt{Int}},\underline{42})),\qquad N\triangleq \mathsf{if}(y,\underline{k},\underline{k'}).$$

The derivable judgments on the term $N$ are of the form $y^{\mathtt{Int}} : [\overline{n}] \vdash_1 N : \overline{k}$, with $\overline{k}\in\{k,k'\}$. In fact, once fixed $\overline{n}$, the derivation is unique and is given at the left-hand side of Figure 4. As for the term $M$, the derivable judgments are of the form $y^{\mathtt{Int}} : [\,] \vdash_1 M : ([h,h'],42)$, with $h,h' > 0$. However, in this case we have two different derivations whenever $h\neq h'$: one derivation is given at the right-hand side of Figure 4, while the other one is obtained by swapping the order between the $\mathtt{var}$ rules on $h$ and $h'$.

The weight of a derivation of $(M)\,N$ is 1. However, the number of derivation of a fix judgment $y^{\mathtt{Int}} : [n,n'] \vdash_1 (M)\,N : 42$ depends on $n,n',k,k'$ following cases:

- if $k=k'$ and $n=n'$, then there is exactly one derivation;
- if $k=k'$ and $n\neq n'$, then there are two derivations, one ending with the rule $\mathtt{app}_{([k,k],\{(k,1)\mapsto[n],(k,2)\mapsto[n']\})}$ and the other one with the rule $\mathtt{app}_{([k,k],\{(k,1)\mapsto[n'],(k,2)\mapsto[n]\})}$;
- if $k\neq k'$ and $n=n'$, then there are two possible derivations, depending on which is the derivation used for inferring the judgment on $M$; notice that the final rule typing $(M)\,N$ must be $\mathtt{app}_{([k,k'],\{(k,1)\mapsto[n],(k',1)\mapsto[n]\})}$;
- if $k\neq k'$ and $n\neq n'$, then there are four possible derivations, depending on the last rule (i.e. $\mathtt{app}_{([k,k'],\{(k,1)\mapsto[n],(k',1)\mapsto[n']\})}$ or $\mathtt{app}_{([k,k'],\{(k,1)\mapsto[n'],(k',1)\mapsto[n]\})}$) and on the derivation for inferring $M$.

Actually, by Lemma 20, weights of possible derivations sums into $[\![(M)\,N]\!]^y$, i.e.

$$[\![(M)\,N]\!]^y_{m,h} = \begin{cases} 1 & \text{if } h=42,\ k=k',\ m=[n,n],\\ 2 & \text{if } h=42,\ k=k',\ m=[n\neq n'],\\ & \text{or } h=42,\ k\neq k',\ m=[n,n],\\ 4 & \text{if } h=42,\ k\neq k',\ m=[n\neq n'],\\ 0 & \text{otherwise.}\end{cases}$$

**Example 23.** Any derivation of a fix-point term $\mathsf{fix}(M)$ ends with a cluster of $\mathtt{fix}$ rules, each rule has one premise typing $M$ and a number of premises typing $\mathsf{fix}(M)$ and along which the cluster grows. Since the derivation must be finite, the cluster eventually ends with $\mathtt{fix}$ rules of label $([\,],(\,))$. They have exactly one premise typing $M$ with a web element of shape $([\,],a)$. As a consequence, the term $\Omega_A = \mathsf{fix}(\lambda x^A.x)$ has no web type, as $\lambda x^A.x$ cannot have a web type of shape $([\,],a)$. We find again $[\![\Omega_A]\!] = \mathbf{0}$.

On the other hand, any derivation of $\mathsf{fix}(\lambda x^{\mathtt{Int}}.x\oplus\underline{0})$ will end with a branch of $n>0$ $\mathtt{fix}$ rules: $n-1$ rules labelled by $([0],(\,))$, and the rule at the top of the branch labelled by $([\,],(\,))$ and having as premise the unique derivation of $\vdash_{\frac{1}{2}} \lambda x^{\mathtt{Int}}.x\oplus\underline{0} : ([\,],0)$. The whole derivation has conclusion $\vdash_{\frac{1}{2^n}} \mathsf{fix}(\lambda x^{\mathtt{Int}}.x\oplus\underline{0}) : 0$ and the sum of these weights for $n>0$ yields $[\![\mathsf{fix}(\lambda x^{\mathtt{Int}}.x\oplus\underline{0})]\!]_0 = 1$. We find again the results of Examples 3 and 12.

$$\begin{aligned}
\mathcal{P}(n) &= \lambda x^{\mathtt{Int}}.\mathsf{if}(x=\underline{n},\underline{0},\Omega_{\mathtt{Int}})\\
\mathcal{N}(n) &= \underline{n}\\
\mathcal{P}(a) &= \lambda z^{B\Rightarrow C}.(\mathcal{P}(c))\,((z)\,\mathtt{choose}(X_i\cdot\mathcal{N}(b_i))_{i=1}^n)\\
\mathcal{N}(a) &= \lambda x^B.\mathsf{if}(\wedge_{i=1}^n\,(\mathcal{P}(b_i))\,x,\mathcal{N}(c),\Omega_C)
\end{aligned}$$

**Figure 5:** testing terms. In the higher-order case, i.e. $A=B\Rightarrow C$, we suppose $a=([b_1,\dots,b_n],c)$, with $b_i\in|\mathcal{B}|$ and $c\in|\mathcal{C}|$. Subtesting terms are supposed to have disjoint parameters and the $X_i$'s occurring in the definition of $\mathcal{P}(a)$ are assumed to be fresh.

### 5.2 Testing terms

Figure 5 associates with every web element $a\in|\mathcal{A}|$ two closed terms of PPCF$^{\mathfrak{P}}$: one term $\mathcal{P}(a)$ of type $A\Rightarrow\mathtt{Int}$ and one term $\mathcal{N}(a)$ of type $A$, defined by mutual induction on $A$.

$\mathcal{P}(a)$ and $\mathcal{N}(a)$ are named *testing terms*. The parameters occurring in this terms are in a bijective correspondence with the elements of the multisets appearing in $a$ (at all depths).

In the following, we focus on $[\![\mathcal{P}(a)]\!]_{(m,0)}$ and $[\![\mathcal{N}(a)]\!]_{a'}$, for any $m\in\mathcal{M}_{\mathrm{f}}(|\mathcal{A}|)$ and $a'\in\mathcal{A}$, which are formal series in the parameters occurring in the terms (Fact 19). More precisely, we are interested in the coefficient of the monomial having each parameter of the term occurring with degree 1. This monomial is called the *skeleton of* $a$ and is denoted by $\mathtt{sk}(a)$.

**Lemma 24.** *Let $A$ be a type, $a,a'\in|\mathcal{A}|$ and $m\in\mathcal{M}_f(|\mathcal{A}|)$. In the series $[\![\mathcal{P}(a)]\!]_{(m,0)}$ (resp. $[\![\mathcal{N}(a)]\!]_{a'}$), the monomial $\mathtt{sk}(a)$ has a non zero coefficient if and only if $m=[a]$ (resp. $a'=a$).*

*Proof.* By definition, the coefficient of the monomial $\mathtt{sk}(a)$ in $[\![\mathcal{P}(a)]\!]_{(m,0)}$ can be recovered from the sum of the weights of the derivations of shape $P ::\vdash_{\kappa\mathtt{sk}(a)} \mathcal{P}(a) : (m,0)$, and similarly for $[\![\mathcal{N}(a)]\!]_{a'}$. Hence, the statement is equivalent to the following property $\mathtt{IH}(A)$, which will be proved by induction on type $A$:

- ⋆ *For any $a\in|\mathcal{A}|$, there is a derivation of $\vdash_{\kappa\mathtt{sk}(a)} \mathcal{P}(a) : (m,0)$, with $\kappa\neq 0$, if and only if $m=[a]$*
- ⋆ *For any $a\in|\mathcal{A}|$, there is a derivation of $\vdash_{\kappa\mathtt{sk}(a)} \mathcal{N}(a) : a'$, with $\kappa\neq 0$, if and only if $a'=a$.*

For the ground type $\mathtt{Int}$, testing terms have no parameters, so that the monomial $\mathtt{sk}(n)$ is the constant term of the series. The derivations of $\mathcal{P}(n)$ and $\mathcal{N}(n)$ are unique and of shape:

$$\dfrac{\dfrac{\dfrac{m=[n]}{x:m\vdash_1 x:n}\qquad \dfrac{}{\vdash_1\underline{0}:0}}{x:m\vdash_1 \mathsf{if}(x=\underline{n},0,\Omega_\iota):0}}{\vdash_1 \lambda x^\iota.\mathsf{if}(x=\underline{n},0,\Omega_\iota):(m,0)}\qquad\qquad \dfrac{a'=n}{\vdash_1\underline{n}:a'}$$

$$
\dfrac{\dfrac{}{y^{\mathtt{Int}} : [\overline{n}] \vdash_1 y : \overline{n}} \qquad \dfrac{}{y^{\mathtt{Int}} : [\,] \vdash_1 \underline{k} : \overline{k}}}{y^{\mathtt{Int}} : [\overline{n}] \vdash_1 N : \overline{k}}
\qquad
\dfrac{\dfrac{\dfrac{}{y^{\mathtt{Int}} : [\,], x^{\mathtt{Int}} : [h] \vdash_1 x : h} \qquad \dfrac{\dfrac{}{y^{\mathtt{Int}} : [\,], x^{\mathtt{Int}} : [h'] \vdash_1 x : h'} \quad \dfrac{}{y^{\mathtt{Int}} : [\,], x^{\mathtt{Int}} : [\,] \vdash_1 \underline{42} : 42}}{y^{\mathtt{Int}} : [\,], x^{\mathtt{Int}} : [h'] \vdash_1 \mathsf{if}(x, \Omega_{\mathtt{Int}}, \underline{42}) : 42}}{y^{\mathtt{Int}} : [\,], x^{\mathtt{Int}} : [h, h'] \vdash_1 \mathsf{if}(x, \Omega_{\mathtt{Int}}, \mathsf{if}(x, \Omega_{\mathtt{Int}}, \underline{42})) : 42}}{y^{\mathtt{Int}} : [\,] \vdash_1 M : ([h, h'], 42)}
$$

**Figure 4:** Weighted derivations discussed in the Example 22

$$
\dfrac{\dfrac{\dfrac{\mathtt{IH}(C)\text{: } q = [c]}{\vdash_\gamma \mathcal{P}(c) : (q, 0)}\,\text{③} \qquad \dfrac{\dfrac{\begin{array}{c} m = [(p', c)] \\ p' = [b'_1, \ldots, b'_{k'}] \end{array}}{z : m \vdash_1 z : (p', c)}\,\text{⑤} \qquad \forall j \in \{1, \ldots, k'\},\ \dfrac{\dfrac{\dfrac{\mathtt{IH}(B)\text{: } b_{i_j} = b'_j}{\vdash_{\beta_{i_j}} \mathcal{N}(b_{i_j}) : b'_j}\,\text{⑧}}{\vdash_{X_{i_j}\beta_{i_j}} X_{i_j} \cdot \mathcal{N}(b_{i_j}) : b'_j}\,\text{⑦}}{\vdash_{\frac{1}{k} X_{i_j}\beta_{i_j}} \mathtt{choose}(X_i \cdot \mathcal{N}(b_i))_{i=1}^k : b'_j}\,\text{⑥}}{z : m \vdash_\beta (z)\, \mathtt{choose}(X_i \cdot \mathcal{N}(b_i))_{i=1}^k : c}\,\text{④}}{z : m \vdash_{\omega(P)} (\mathcal{P}(c))\, (z)\, \mathtt{choose}(X_i \cdot \mathcal{N}(b_i))_{i=1}^k : 0}\,\text{②}}{\vdash_{\omega(P)} \lambda z^{B \Rightarrow C}.(\mathcal{P}(c))\, (z)\, \mathtt{choose}(X_i \cdot \mathcal{N}(b_i))_{i=1}^k : (m, 0)}\,\text{①}
$$

**Figure 6:** Proof derivation contributing in the $\mathtt{sk}(a)$ monomial in $\mathcal{P}(a)$.

We conclude that the coefficient is non zero iff $m = [n]$ and $a' = n$.

For the inductive case, let us assume that the result holds for type $B$ and $C$ and that $a = ([b_1, \ldots, b_k], c) \in |\mathcal{B} \Rightarrow \mathcal{C}|$. Notice that $\mathtt{sk}(a) = \mathtt{sk}(c) \prod_{i=1}^k X_i \mathtt{sk}(b_i)$. A derivation of $\vdash_{\kappa \mathtt{sk}(a)} \mathcal{P}(a) : (m, 0)$, with $\kappa \neq 0$, must be as described in Figure 6, as justified below:

- ① The only possible rule is the abstraction one.
- ②,③ For using the application rule, we have to choose an intermediate multiset $q \in \mathcal{M}_{\mathrm{f}}(|\mathcal{B} \Rightarrow \mathcal{C}|)$ and a context $\Gamma^{\bullet\prime}$ such that $\Gamma^{\bullet\prime} \vdash_\gamma \mathcal{P}(c) : (q, 0)$ is derivable. Since $\mathcal{P}(c)$ is a closed term, by Fact 18, $\Gamma^{\bullet\prime}$ must be empty (i.e. mapping all variables to the empty multiset). Moreover, by definition, the parameters occurring in $\mathcal{P}(c)$ must be different from those occurring in each $X_i \cdot \mathcal{N}(b_i)$. Therefore, $\gamma = \nu \mathtt{sk}(c)$, with $\nu \neq 0$. By induction hypothesis in ③, we infer that $q = [c]$. We then deduce that ② is a binary applicative rule of label $\mathtt{app}_{([c],(z:m))}$. Hence, the right premise is the sequent $z : m \vdash_\beta (z)\, \mathtt{choose}(X_i \cdot \mathcal{N}(b_i))_{i=1}^k : c$, for a $\beta$ of the shape $\mu \prod_{i=1}^k X_i \cdot \mathtt{sk}(b_i)$.
- ④,⑤ The shape of the term forces us to use the application rule again: let $p' = [b'_1, \ldots, b'_{k'}]$ denote the intermediate multiset. Since the left premise is the only one in which $z$ appears, there is only one possible way of separating the context: $\Gamma^{\bullet\prime} = z : m$ and $\Gamma^\bullet_{b'_j} = \emptyset$. Since the only possible rule for ⑤ is the $\mathtt{var}$ rule, we get $m = [(p', c)]$. We conclude that ④ is an applicative rule with $k' + 1$ premises, the $k'$ right premises being of shape $\vdash_{\beta_j} \mathtt{choose}(X_i \cdot \mathcal{N}(b_i))_{i=1}^k : b'_j$.
- ⑥,⑦,⑧ For any $j \in \{1 \ldots k'\}$, there is a cluster of $h < k$ rules $\mathtt{if}_s$ and at the top of it a rule $\mathtt{if}_0$ with, as right premise, a weighted typing of a summand $X_{i_j} \cdot \mathcal{N}(b_{i_j})$. From that we must apply a $\mathtt{par}$ rule (numbered ⑦) and get as premise a sequent of shape $\vdash_{\beta_{i_j}} \mathcal{N}(b_{i_j}) : b'_j$, for $i_j \in 1 \ldots k$ and weight $\beta_{i_j}$. Then, notice that the weight of the conclusion of ⑦ is $X_{i_j}\beta_{i_j}$ and that of the conclusion of ⑥ is $\frac{1}{k} X_{i_j}\beta_{i_j}$. On the one hand, the reasoning on ② gives us $\beta = \mu \prod_{i=1}^k X_i \cdot \mathtt{sk}(b_i)$, while the definition of the weight of the conclusion of ④ gives us $\beta = \prod_{j=1}^{k'} \frac{1}{k} X_{i_j} \cdot \beta_{i_j}$. Since by definition no parameter $X_i$ occurs in $\mathcal{N}(b_{i_j})$, hence no $X_i$ appears in $\beta_{i_j}$, we have that the mapping $j \mapsto i_j$ is a bijective correspondence between $\{1, \ldots, k'\}$ and $\{1, \ldots, k\}$, so $k = k'$. Finally, since the sets of parameters occurring in the $\mathcal{N}(b_{i_j})$ are pairwise disjoint, we deduce that $\beta_{i_j}$ is proportional to $\mathtt{sk}(b_{i_j})$. We can then apply the induction hypothesis on ⑧ and deduce that $b_{i_j} = b_i$, and so $p' = [b_1, \ldots, b_k]$.

Combining all these results, we eventually get that $m = [(p', c)] = [([b_1, \ldots, b_k], c)] = [a]$. We conclude that there is a derivation of $\vdash_{\kappa \mathtt{sk}(a)} \mathcal{P}(a) : (m, 0)$ if and only if $m = [a]$.

Now, let us consider a derivation of $\vdash_{\kappa \mathtt{sk}(a)} \mathcal{N}(a) : a'$, which must have the following form:

$$
\dfrac{\dfrac{\vdots}{x : p' \vdash_\alpha \mathsf{if}(\wedge_{i=1}^k (\mathcal{P}(b_i))\, x, \mathcal{N}(c), \Omega_C) : c'}}{\vdash_\alpha \lambda x^B.\mathsf{if}(\wedge_{i=1}^k (\mathcal{P}(b_i))\, x, \mathcal{N}(c), \Omega_C) : (p', c')}
$$

We prove by recursion on $k - j$ the property $\mathtt{RH}(j)$:

- ⋆ *if $x : p' \vdash_\alpha \mathsf{if}(\wedge_{i=j}^k (\mathcal{P}(b_i))\, x, \mathcal{N}(c), \Omega_C) : c'$ is derivable, then $\alpha$ is proportional to $\mathtt{sk}(c) \prod_{i=j}^k \mathtt{sk}(b_i)$ if and only if $c = c'$ and $p' = [b_j, \ldots, b_k]$.*

The base case ($j = k$) is the induction hypothesis $\mathtt{IH}(C)$. Indeed, if $x : p' \vdash_\alpha \mathcal{N}(c) : c'$ is derivable, then, by Fact 18, $p'$ is empty, so that the weight $\alpha$ is proportional to $\mathtt{sk}(c)$ if and only if $c = c'$. The inductive case is proved accordingly to Figure 7:

- ① Since $\Omega_C$ has no web type (Example 23), the only possible rule is $\mathtt{if}_0$. The choice of how to partition the context $m_1 \uplus m_2$ will be deduced from the type derivation. Since by definition the set of parameters in $\mathcal{P}(b_j)$ and the set of parameters in $\mathsf{if}(\wedge_{i=j+1}^k (\mathcal{P}(b_i))\, x, \mathcal{N}(c), \Omega_C)$ are disjoint, we have that $\beta$ is proportional to $\mathtt{sk}(b_j)$ and $\gamma$ is proportional to $\prod_{i=j+1}^k \mathtt{sk}(b_i)$.
- ② For the right premise we can then apply the hypothesis $\mathtt{RH}(j{+}1)$ and deduce that $c = c'$ and $m_2 = [b_{j+1}, \ldots, b_k]$.
- ③,④ For the left premise, the only possible rule is $\mathtt{app}$. We have then to choose the intermediate multiset $m'$. As in the previous cases, one can argue that the left premise has empty context

$$\dfrac{\dfrac{\dfrac{\mathtt{IH}(A)\colon m_1 = [b_j]}{\vdash_\beta \mathcal{P}(b_j) : (m_1, 0)}④ \quad \dfrac{}{x : m_1 \vdash_1 x : m_1}}{x : m_1 \vdash_\beta (\mathcal{P}(b_j))\, x : 0}③ \quad \dfrac{\mathtt{RH}(j+1)\colon c = c', m_2 = [b_{j+1}, \dots, b_k]}{x : m_2 \vdash_\gamma \mathsf{if}(\wedge_{i=j+1}^{k} (\mathcal{P}(b_i))\, x, \mathcal{N}(c), \Omega_C) : c'}②}{x : m_1 \uplus m_2 \vdash_{\beta\gamma} \mathsf{if}(\wedge_{i=j}^{k} (\mathcal{P}(b_i))\, x, \mathcal{N}(c), \Omega_C) : c'}①$$

**Figure 7:** Proof derivation contributing in the $\mathtt{sk}(a)$ monomial in $\mathcal{N}(a)$.

since $\mathcal{P}(b_j)$ is closed and then the induction hypothesis $\mathtt{IH}(C)$ gives $m' = [b_j]$. This means that ③ has only one premise at the right-hand side, which is a conclusion of a $\mathtt{var}$ rule, so that $m' = m_1$.

We conclude by combining all these results: we get that the context $m_1 \uplus m_2 = [b_j, \dots, b_k]$ and $c = c'$.

We have proved that $\mathtt{RH}(1)$ holds: $c = c'$ and $p' = [b_1, \dots, b_k]$, hence $a' = a$. □

### 5.3 Main Result

The theory of multi-variables analytic functions can be subtler than single variable ones (for instance zeros of functions are not isolated in general). In the following Lemma 25, we underline that the coefficients of a power series vanishing on a neighborhood of zero have to be null. Indeed, coefficients are computed by successive derivations as limits of difference quotients. We then apply this result to power series with possibly negative coefficients.

**Lemma 25.** *[16] Let $f$ be a power series from $\mathbb{R}^n$ to $\mathbb{R}$ absolutely converging on $[0,1]^n$. If $f$ vanishes on a dense subset of $[0,1]^n$, then the coefficients of the power series $f$ are zero.*

**Lemma 26.** *Let $M$ and $N$ be two closed terms of* PPCF *with the same type $A$. If $[\![M]\!] \neq [\![N]\!]$, then there is a* PPCF *term $P$ of type $A \Rightarrow \mathtt{Int}$ such that $\mathrm{PROBA}^{\infty}_{(P)M,\underline{0}} \neq \mathrm{PROBA}^{\infty}_{(P)N,\underline{0}}$.*

*Proof.* Let $a \in |\mathcal{A}|$ such that $[\![M]\!]_a \neq [\![N]\!]_a$. Consider the testing term $\mathcal{P}(a)$ as defined in Figure 5.

Let $f$ denote the series $[\![(\mathcal{P}(a))\,M]\!]_0$, which, by Definition 21 is a power series from $[0,1]^n$ to $[0,1]$, where $n$ is the number of parameters in $\mathcal{P}(a)$. Let $\mathtt{cf}(\mathtt{sk}(a), f)$ denote the coefficient of the monomial $\mathtt{sk}(a)$ in $f$. By definition (Equation (11)) this coefficient is given by the sum of the weights of the possible derivations of $\vdash_\alpha (\mathcal{P}(a))\,M : 0$, for $\alpha$ proportional to $\mathtt{sk}(a)$. The last rule of this derivation must be an $\mathtt{app}$ rule of shape:

$$\dfrac{\vdash_\gamma \mathcal{P}(a) : (m, 0) \qquad \forall (a', i) \in m, \quad \vdash_{\beta_{(a',i)}} M : a'}{\vdash_{\gamma \prod_{(a',i)\in m} \beta_{(a',i)}} (\mathcal{P}(a))\,M : 0}$$

Since $M$ has no free parameter (it is a term of PPCF), each $\beta_{(a',i)}$ is a positive real number. Hence, $\alpha = \gamma \prod_{(a',i)\in m} \beta_{(a',i)}$ is proportional to $\mathtt{sk}(a)$ iff $\gamma$ is proportional to $\mathtt{sk}(a)$. We can then apply Lemma 24 and get $m = [a]$. Moreover, by Lemma 20, the sum of the weights of the derivations of $\vdash M : a$ is equal to the scalar $[\![M]\!]_a$. To sum up, we conclude:

$$\mathtt{cf}(\mathtt{sk}(a), f) = \mathtt{cf}(\mathtt{sk}(a), [\![\mathcal{P}(a)]\!]_{([a],0)})\, [\![M]\!]_a$$

with $\mathtt{cf}(\mathtt{sk}(a), [\![\mathcal{P}(a)]\!]_{([a],0)}) \neq 0$. By an analogous reasoning:

$$\mathtt{cf}(\mathtt{sk}(a), g) = \mathtt{cf}(\mathtt{sk}(a), [\![\mathcal{P}(a)]\!]_{([a],0)})\, [\![N]\!]_a$$

where $g$ denotes the power series $[\![(\mathcal{P}(a))\,N]\!]_0$. We can conclude that $f$ and $g$ have a different coefficient for the monomial $\mathtt{sk}(a)$ as soon as $[\![M]\!]_a \neq [\![N]\!]_a$.

Now, we apply Lemma 25 to the series $f - g$, which is not the zero power series because of the coefficient of $\mathtt{sk}(a)$. Therefore, as rational numbers are dense in $[0,1]$, there is a list $\vec{\kappa}$ of rational numbers in $[0,1]$ such that $f(\vec{\kappa}) \neq g(\vec{\kappa})$. By Lemma 20, this is equivalent to $\left[\!\left[\left(\mathcal{P}(a)\left[\vec{\kappa}/\vec{X}\right]\right) M\right]\!\right]_0 \neq \left[\!\left[\left(\mathcal{P}(a)\left[\vec{\kappa}/\vec{X}\right]\right) N\right]\!\right]_0$. We conclude by setting $P = \mathcal{P}(a)\left[\vec{\kappa}/\vec{X}\right]$. Actually, $P$ is a well-defined PPCF term by Fact 17. So, by adequacy (Theorem 14), $\mathrm{PROBA}^{\infty}_{(P)M,\underline{0}} = [\![(P)\,M]\!]_0 \neq [\![(P)\,N]\!]_0 = \mathrm{PROBA}^{\infty}_{(P)N,\underline{0}}$. □

**Theorem 27** (Full Abstraction)**.** *Given $\Gamma \vdash M : A$, and $\Gamma \vdash N : A$, we have:*

$$M \equiv_\Gamma N \textit{ iff } [\![M]\!]^\Gamma = [\![N]\!]^\Gamma .$$

*Proof.* By Corollary 16 and Lemma 26. □

As a consequence, the observational equivalence does not depend on the chosen probabilistic primitive (recall Remark 6):

**Corollary 28.** *For any $\mathtt{p} \in \{\mathtt{Coin}, \mathtt{rand}, \oplus_\kappa, \oplus_i \kappa_i, (\kappa_n)_{n\in\mathbb{N}}\}$, let $\equiv_{\mathtt{p}}$ denote the observational equivalence induced by the extension of* PCF *with the probabilistic primitive $\mathtt{p}$. The equivalence $\equiv_{\mathtt{p}}$ coincides with the equality of the interpretations in **PCoh**. In particular, $\equiv_{\mathtt{p}}$ gives the same equivalence on* PCF *terms, for any choice of $\mathtt{p}$.*

*Proof (Sketch).* We have that $\equiv_{\mathtt{Coin}} = \equiv_{\mathtt{rand}}$ since the two primitives $\mathtt{Coin}$ and $\mathtt{rand}$ are interdefinable by [5]. Similarly, by the discussion of Remark 6, we have: $\equiv_{\mathtt{rand}} \supseteq \equiv_{\oplus_\kappa} \supseteq \equiv_{\oplus_i \kappa_i} \supseteq \equiv_{(\kappa_n)_{n\in\mathbb{N}}}$. In [4], the adequacy property of **PCoh** is established for the language containing the the $(\kappa_n)_{n\in\mathbb{N}}$ primitive, hence $\equiv_{(\kappa_n)_{n\in\mathbb{N}}}$ contains the equality $=_{\mathbf{Pcoh}}$ of the interpretations in **PCoh**. Finally, by Theorem 27, we have $=_{\mathbf{Pcoh}} \supseteq \equiv_{\mathtt{rand}}$ and we can conclude. □

## 6. A counter-example to inequational FA

We prove that inequational full abstraction fails for **PCoh**. The proof consists in: (i) showing that the morphisms of Remark 11, that proves the non-extensionality of the **PCoh** order, are definable in PPCF (Equation 12); (ii) achieving a context lemma (Proposition 31) allowing us to infer the observational inequality by just observing the behavior of closed terms in applicative contexts.

If one would like to enrich the language in order to make observable also the componentwise order of **PCoh**, one should add a kind of differential operator to PPCF, in the spirit of [9]. However, such an extension is not trivial, since **PCoh** is known not to be sound for the whole differential $\lambda$-calculus.

Let us define $M_1$ and $M_2$ as follows:

$$M_1 \triangleq \lambda x^{\mathtt{Int}}.\mathsf{if}(x, \underline{0}, \underline{0}), \qquad M_2 \triangleq \lambda x^{\mathtt{Int}}.\underline{0}. \qquad (12)$$

By using rules of Figure 3 and Lemma 20, one can check that the semantics of the two terms is given by the morphisms of Example 11, i.e. $[\![M_1]\!] = \phi$ and $[\![M_2]\!] = \psi$. Hence, the two matrices $[\![M_1]\!]$ and $[\![M_2]\!]$ are incomparable in **Pcoh**.

In order to prove $M_1 \sqsubseteq M_2$, we use a standard reasoning and introduce a logical relation (Definition 29) allowing us to shrink the set of contexts (Proposition 31). We conclude with Corollary 32.

**Definition 29.** By structural induction on a type $A$, we define the binary relation $\lhd_A$ over closed PPCF terms of type $A$, as follows: $N \lhd_{\mathtt{Int}} N'$ iff $\forall n \in \mathbb{N}$, $[\![N]\!]_n \leq [\![N']\!]_n$, and $M \lhd_{A \Rightarrow B} M'$ iff $\forall N \lhd_A N'$, $(M)\, N \lhd_B (M')\, N'$.

**Lemma 30.** *Let $M$ be a term of type $x_1 : A_1, \ldots, x_n : A_n \vdash M : A$. If for all $i$, $N_i \lhd_{A_i} N_i'$, then $M\left[\vec{N}/\vec{x}\right] \lhd_A M\left[\vec{N}'/\vec{x}\right]$.*

**Proposition 31.** *Let $M$ and $N$ be closed terms of type $A$. Then, $M \lhd_A N$ iff $M \sqsubseteq_A N$.*

**Corollary 32.** *The terms $M_1$ and $M_2$ of Equation (12) are logically related: $M_1 \lhd_{\mathtt{Int} \Rightarrow \mathtt{Int}} M_2$. Hence, $M_1 \sqsubseteq M_2$.*

## 7. Conclusion

We show that the observational equality over probabilistic programs is faithfully described by **PCoh**, a relatively abstract model based on convex sets and extensional functions. The proof uses innovative tools which might be useful to study probabilistic programing from a semantical viewpoint. In a language with random functions, two programs should be considered different not only when they give different results, but also when they give the same result but with different probabilities. Showing this difference can be much harder than in a deterministic language. Indeed, it requires a sharp control over coefficients expressing probabilities. **PCoh** denotes programs with power series, this allows us to use standard tools of Calculus for handling probabilities.

Although we chose one probabilistic primitive (see Remark 6), Corollary 28 shows that our result does not depend on this choice. Moreover, we focused on PPCF, a call-by-name functional language, but our result might be extended to other frameworks. In fact, **PCoh** comes from a model of linear logic: it is the cokleisli category associated with the exponential comonad, that corresponds to the translation of the functional arrow $A \Rightarrow B$ into the linear logic formula $!A \multimap B$. Call-by-value can be obtained by using the Eilenberg-Moore construction, i.e. translating $A \Rightarrow B$ into $!(A \multimap B)$. Control operators can be introduced considering a polarized fragment of linear logic. Yet, extending this model to concurrent systems or references is certainly more challenging.

A crucial tool in the proof of full abstraction is the use of the type system of Figure 3, which is a kind of intersection type system. In fact, a web element $([a_1, \ldots, a_n], b)$ can be seen as a type $(a_1 \wedge \cdots \wedge a_n) \to b$, where the intersection is non-idempotent. This system is a quantitative refinement of De Carvalho's [6] and yields the first logical presentation of a vectorial based semantics. Clearly, both type inference and type checking are undecidable. However, one can look for interesting restrictions of PPCF where the system becomes decidable, in the spirit of [20].

Last, it should be noticed that, unlike most full abstraction models for PCF, our model has no simple PPCF-definability properties. Our full abstraction proof builds applicative contexts using terms which belong to a very small subset of the domains associated with types. Their discriminating power relies on a strong regularity property of power series (unlike smooth functions, a power series which vanishes on an open set must be equal to 0, see for example to the function defined by $e^{-1/x^2}$ for $x > 0$ and 0 for $x \leq 0$). In contrast, most full abstraction proofs build applicative contexts using terms which belong to a *dense* subset of the corresponding domains.

## References


[1] S. Abramsky, R. Jagadeesan, and P. Malacaria. Full abstraction for PCF. *Information and Computation*, 163(2):409–470, Dec. 2000.

[2] G. Boudol, P.-L. Curien, and C. Lavatelli. A semantics for lambda calculi with resources. *Math. Struct. Comp. Sci.*, 9(4):437–482, 1999.

[3] A. Bucciarelli, A. Carraro, T. Ehrhard, G. Manzonetto, et al. Full abstraction for resource calculus with tests. In *CSL11 – Annual Conference of the EACSL*, volume 12, pages 97–111, 2011.

[4] V. Danos and T. Ehrhard. Probabilistic coherence spaces as a model of higher-order probabilistic computation. *Inf. Comput.*, 209(6):966–991, 2011.

[5] V. Danos and R. Harmer. Probabilistic game semantics. *ACM Transactions on Computational Logic*, 3(3):359–382, July 2002.

[6] D. de Carvalho. Execution Time of $\lambda$-Terms via Denotational Semantics and Intersection Types. Preprint, 2009.

[7] T. Ehrhard. On Köthe sequence spaces and linear logic. *Math. Struct. Comput. Sci.*, 12:579–623, 2002.

[8] T. Ehrhard. Finiteness spaces. *Math. Struct. Comput. Sci.*, 15(4):615–646, 2005.

[9] T. Ehrhard and L. Regnier. The Differential Lambda-Calculus. *Theor. Comput. Sci.*, 309(1):1–41, 2003.

[10] J.-Y. Girard. The system F of variable types, fifteen years later. *Theor. Comput. Sci.*, 45:159–192, 1986.

[11] J.-Y. Girard. Linear logic. *Theor. Comput. Sci.*, 50:1–102, 1987.

[12] J.-Y. Girard. Normal functors, power series and lambda-calculus. *Ann. Pure Appl. Logic*, 37(2):129–177, 1988.

[13] J.-Y. Girard. Coherent banach spaces: a continuous denotational semantics. *Theor. Comput. Sci.*, 227:297, 1999.

[14] J.-Y. Girard. Between logic and quantic: a tract. In T. Ehrhard, J.-Y. Girard, P. Ruet, and P. Scott, editors, *Linear Logic in Computer Science*, volume 316 of *London Math. Soc. Lect. Notes Ser.* CUP, 2004.

[15] J. Goubault-Larrecq and D. Varacca. Continuous random variables. In *LICS*, pages 97–106. IEEE Computer Society, 2011.

[16] É. Goursat. *Cours d'analyse mathématique. Tome I.* Gauthier-Villars, 1918.

[17] R. Hasegawa. Two applications of analytic functors. *Theor. Comput. Sci.*, 272(1-2):113–175, 2002.

[18] M. Hyland and L. Ong. On full abstraction for PCF. *Information and Computation*, 163(2):285–408, Dec. 2000.

[19] C. Jones and G. Plotkin. A probabilistic powerdomains of evaluation. In *LICS*. IEEE Computer Society, 1989.

[20] A. J. Kfoury and J. B. Wells. Principality and decidable type inference for finite-rank intersection types. In *POPL*, pages 161–174, 1999.

[21] U. D. Lago and M. Zorzi. Probabilistic operational semantics for the lambda calculus. *RAIRO - Theor. Inf. and Applic.*, 46(3):413–450, 2012.

[22] J. Laird, G. Manzonetto, G. McCusker, and M. Pagani. Weighted relational models of typed lambda-calculi. In *LICS 2013*. IEEE Press, June 2013.

[23] R. Milner. Fully abstract models of typed lambda-calculi. *Theor. Comput. Sci.*, 4:1–22, 1977.

[24] R. Milner and C. Strachey. *A Theory of Programming Language Semantics*. Chapman and Hall, London, 1976.

[25] E. Moggi. Computational lambda-calculus and monads. In *LICS*, pages 14–23. IEEE Computer Society, 1989.

[26] A. D. Pierro, C. Hankin, and H. Wiklicky. Probabilistic lambda-calculus and quantitative program analysis. *J. Log. Comput.*, 15(2): 159–179, 2005.

[27] G. D. Plotkin. A powerdomain construction. *SIAM J. Comput.*, 5(3): 452–487, 1976.

[28] G. D. Plotkin. LCF considered as a programming language. *Theor. Comput. Sci.*, 5(3):225–255, 1977.

[29] N. Saheb-Djahromi. Cpo's of measures for nondeterminism. *Theor. Comput. Sci.*, 12:19–37, 1980.

[30] D. Scott. Continuous lattices. In Lawvere, editor, *Toposes, Algebraic Geometry and Logic*, volume 274 of *Lecture Notes in Math.*, pages 97–136. Springer, 1972.