

Citation for published version:

Laird, JD, Manzonetto, G, Mccusker, G & Pagani, M 2013, Weighted relational models of typed Lambda-Calculi. in *2013 28th Annual IEEE/ACM Symposium on Logic in Computer Science (LICS)*. Annual IEEE/ACM Symposium on Logic in Computer Science (LICS), IEEE, Los Alamitos, California, pp. 301-310, 2013 Twenty-Eighth Annual IEEE/ACM Symposium on Logic in Computer Science (LICS 2013), New Orleans, LA, USA, UK United Kingdom, 25/06/13. <https://doi.org/10.1109/LICS.2013.36>

DOI:

[10.1109/LICS.2013.36](https://doi.org/10.1109/LICS.2013.36)

Publication date:

2013

Document Version

Peer reviewed version

[Link to publication](#)

© 2013 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works

University of Bath

Alternative formats

If you require this document in an alternative format, please contact:
openaccess@bath.ac.uk

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Weighted relational models of typed lambda-calculi

Jim Laird*, Giulio Manzonetto†, Guy McCusker* and Michele Pagani†,‡

*Department of Computer Science, University of Bath, Bath, BA2 7AY, UK

†Université Paris 13, Sorbonne Paris Cité, LIPN, F-93430, Villetaneuse, France

‡CNRS, UMR 7030, F-93430, Villetaneuse, France

Abstract—The category \mathbf{Rel} of sets and relations yields one of the simplest denotational semantics of Linear Logic (LL). It is known that \mathbf{Rel} is the biproduct completion of the Boolean ring. We consider the generalization of this construction to an arbitrary continuous semiring \mathbf{R} , producing a cpo-enriched category which is a semantics of LL, and its (co)Kleisli category is an adequate model of an extension of PCF, parametrized by \mathbf{R} . Specific instances of \mathbf{R} allow us to compare programs not only with respect to “what they can do”, but also “in how many steps” or “in how many different ways” (for non-deterministic PCF) or even “with what probability” (for probabilistic PCF).

I. INTRODUCTION

Since the pioneering work of Scott, Strachey, Milner, Plotkin and others in the 1970s [1], [2], [3], a rich theory of programming languages has been developed in which programs have both a *denotational semantics*, with programs denoting values of some mathematical structure, and an *operational semantics*, an abstract description of their execution. Typically, there is some notion of correctness connecting the two, the strongest being Milner’s notion of *full abstraction* which places the two characterizations of program behaviour in precise agreement.

Both the operational and denotational approaches have been undeniably successful at developing our understanding of how programs behave and how to reason about them, and it has become standard to regard programs as equivalent when they are *contextually equivalent*: program phrases M and N are considered equivalent if every program of the form $C[M]$ (a program containing M as a subphrase) computes the same answer as $C[N]$ (the same program, with N replacing M). However, this notion of equivalence, and all the attendant operational and denotational theory, usually overlooks *quantitative* notions such as the time, space, or energy consumed by a computation, or the probability of successful computation. This simplification was made with good reason and to great results: the theory has exposed powerful logical techniques, such as relational reasoning [4], [5], and uncovered some of the essential mathematical structure of programs, such as continuity and monads [6]. Nevertheless, the lack of attention paid to quantitative notions in the semantics literature is perhaps surprising, and stands in some contrast to the field of program verification [7], [8], [9].

There are, of course, examples of quantitative operational and denotational semantics. Sands’s theory of *improvements* is an operational account of costs with a refined notion of program equivalence, and Ghica has shown how to refine game semantics to bring its theory of program equivalence in line

with that of Sands [10], [11]. The use of game semantics, rather than a Scott-Strachey denotational model, is revealing: in order to capture *intensional* notions such as the cost of a computation, a model must of course record more detail than simply the input-output behaviour of a program, as is typical of denotational models. Perhaps the most significant step in exposing such detail was the introduction by Girard of *linear logic* [12]: using linear logic, rather than intuitionistic logic, to structure a type system or denotational model immediately reveals information about resource usage. It should come as no surprise that models of linear logic often contain quantitative information. Indeed, even the simple relational model of linear logic uses multisets to keep track of how many times a resource is used. The path to discovery of linear logic took in another quantitative model, the normal functors [13], and coherence spaces; subsequently, Girard showed how to refine coherence spaces to give an account of probabilistic computation, analysed more deeply in [14], [15].

Our purpose in this paper is to give a uniform denotational account of a range of quantitative notions, using a simple refinement of the relational model. Relations between sets A and B can be seen as matrices indexed by A and B , populated by Boolean values. Replacing the Booleans by elements of an arbitrary continuous semiring, we arrive at a new *weighted relations* model embodying some quantitative information; but what does that information tell us? We consider \mathbf{PCF}^{or} , the extension of Plotkin’s \mathbf{PCF} with a nondeterministic choice operator which can naturally be interpreted in our models by addition of matrices. The interpretation of a closed term of ground type is then a vector of scalars from \mathcal{R} . To understand their meaning, we consider a further extended language $\mathbf{PCF}^{\mathcal{R}}$, in which terms can be instrumented with elements of \mathcal{R} . We demonstrate that our weighted relations correctly model execution in this language, and go on to use $\mathbf{PCF}^{\mathcal{R}}$ as a metalanguage for quantitative modelling of the execution of programs in \mathbf{PCF}^{or} : by varying our choice of \mathcal{R} , and of how terms are instrumented, we show in Section VI that our models can capture, e.g., may- and must- convergence for nondeterministic programs; probability of convergence; and minimum and maximum number of reduction steps to convergence.

Related and future work

The models we describe in this paper are in some sense the simple cousins of a range of models studied by Ehrhard and co-authors: finiteness spaces, Köthe spaces, as well as probabilistic coherence spaces [16], [17], [14]. In all cases,

the coherence structure serves to constrain morphisms so that the quantities in the model can remain finite. Our models sacrifice this property in return for simplicity and generality. Nevertheless, it would be instructive to study the extent to which such coherence-like structures can be deployed when working with arbitrary semirings.

Though our focus in this paper is on weighted models generalising relations, we believe that the key step — replacing matrices over Booleans with matrices over arbitrary \mathcal{R} — is more widely applicable. Indeed, we discovered these models while considering a quantitative version of the constructions described in [18], which allow us to build not only relational models but also games models. We believe that, for instance, Danos and Harmer’s probabilistic games model [19] can be recovered by an analogous construction. We also think that Ghica’s notion of *slot* might be generalized to more abstract algebraic structures, like semirings. The advantage of these game semantics is that they can model Erratic Idealized Algor, which is significantly richer than probabilistic PCE.

II. PRELIMINARIES

Let us fix some notation. We denote by \mathbb{N} the set of natural numbers and by \mathbb{R}^+ the set of positive real numbers. Given two sets A, B , we write $A \subseteq_f B$ if A is a finite subset of B .

A. Category Theory

Given a category \mathbf{C} and objects A, B we denote by $\mathbf{C}(A, B)$ the corresponding hom-set and by $\varphi, \psi, \vartheta, \dots$ its elements. We write the identity morphism on A as id^A , or simply A . Composition is written using infix $;$ in diagram order.

In a *symmetric monoidal category* (*smc*) \mathbf{C} , we denote by \otimes the tensor product and by 1 its unit. When \mathbf{C} is monoidal closed (*smcc*), the monoidal exponential object is denoted as $A \multimap B$. We use $\text{eval}^{A,B} \in \mathbf{C}((A \multimap B) \otimes A, B)$ for the monoidal evaluation morphism and $\lambda(\varphi) \in \mathbf{C}(A, B \multimap C)$ for the monoidal currying of a morphism $\varphi \in \mathbf{C}(A \otimes B, C)$. When \mathbf{C} is moreover \star -autonomous with respect to a dualizing object \perp , we indicate by A^\perp the dual object $A \multimap \perp$.

We will elide all associativity and unit isomorphisms associated with monoidal categories.

In a *cartesian closed category* (*ccc*) \mathbf{C} , we write \top for the terminal object and \top^A for the unique morphism in $\mathbf{C}(A, \top)$. We use $\langle \varphi, \psi \rangle$ to denote the pairing of maps $\varphi \in \mathbf{C}(A, B)$ and $\psi \in \mathbf{C}(A, C)$, and π^1, π^2 for the corresponding projections. In presence of biproducts, we denote by ι^1, ι^2 the corresponding injections. The exponential object is denoted by $A \Rightarrow B$, the evaluation map by $\text{Eval}^{A,B} \in \mathbf{C}((A \Rightarrow B) \times A, B)$ and the currying of $\varphi \in \mathbf{C}(A \times B, C)$ by $\Lambda(\varphi) \in \mathbf{C}(A, B \Rightarrow C)$.

An *object of numerals* \mathfrak{N} is an object N equipped with maps $z \in \mathbf{C}(\top, N)$, $\text{succ}, \text{pred} \in \mathbf{C}(N, N)$, and $\text{zero}^? \in \mathbf{C}(N \times (N \times N), N)$ such that $(\forall n \in \mathbb{N}, \forall \varphi, \psi \in \mathbf{C}(A, N))$:

$$\begin{aligned} \tilde{0}; \text{pred} &= \tilde{0}, & \widetilde{n+1}; \text{pred} &= \tilde{n}, \\ (\tilde{0} \times \langle \varphi, \psi \rangle); \text{zero}^? &= \varphi, & ((\widetilde{n+1}) \times \langle \varphi, \psi \rangle); \text{zero}^? &= \psi, \end{aligned}$$

where $\tilde{n} \in \mathbf{C}(\top, N)$ is defined by $\tilde{0} = z$ and $\widetilde{n+1} = \tilde{n}; \text{succ}$.

B. Lafont Categories

We now describe in a nutshell the categorical semantics of linear logic (LL) as formulated in Lafont’s thesis [20]. This is not the most general definition of a LL model, but it has the advantage of being simple and general enough to encompass the class of models that will be defined in Section III. Our main reference for categorical models of LL is the paper [21].

Recall that an object A of an smcc \mathbf{C} is a (*commutative*) *comonoid* if it is equipped with a multiplication $c \in \mathbf{C}(A, A \otimes A)$ and a unit $w \in \mathbf{C}(A, 1)$ satisfying the usual associativity (commutativity) and unit equations. A *comonoid morphism* φ from (A_1, c_1, w_1) to (A_2, c_2, w_2) is defined as a morphism $\varphi \in \mathbf{C}(A_1, A_2)$ such that $\varphi; c_2 = c_1; (\varphi \otimes \varphi)$ and $\varphi; w_2 = w_1$.

Definition II.1. An smcc \mathbf{C} is a Lafont category if:

- (i) it has finite products and,
- (ii) for every object A , there exists an object $!A$ being the free commutative comonoid generated by A .

Condition (ii) asks that for every A , there is an object $!A$ endowed with a commutative comonoid structure:

$$\text{contr}^A \in \mathbf{C}(!A, !A \otimes !A), \quad \text{weak}^A \in \mathbf{C}(!A, 1),$$

and a morphism $\text{der}^A \in \mathbf{C}(!A, A)$ satisfying the following universality property: for every commutative comonoid B and for every morphism $\varphi \in \mathbf{C}(B, A)$ there exists a unique comonoid morphism $\varphi^\dagger \in \mathbf{C}(B, !A)$ satisfying $\varphi^\dagger; \text{der}^A = \varphi$. The multiplication and the unit of $!A$ are called respectively *contraction* and *weakening*, while der is called *dereliction*.

Every Lafont category \mathbf{C} is equipped with a comonad $(!, \text{der}, \text{dig})$ defined as follows:

- the endofunctor $!$ sends every object A into the free commutative comonoid $!A$ and every morphism $\varphi \in \mathbf{C}(A, B)$ into $(\text{der}^A; \varphi)^\dagger \in \mathbf{C}(!A, !B)$,
- the multiplication is called *digging* and defined as $\text{dig}^A := (\text{id}^{!A})^\dagger \in \mathbf{C}(!A, !!A)$,
- the unit is the morphism $\text{der}^A \in \mathbf{C}(!A, A)$ given above.

The functor $!$ is equipped with a monoidal structure turning it into a strong symmetric monoidal functor from the smc (\mathbf{C}, \otimes) to the smc (\mathbf{C}, \times) : the corresponding two isomorphisms are given by $m^\top := (\top^1)^\dagger \in \mathbf{C}(1, !\top)$ and $m^{A,B} := ((\text{der}^A \otimes \text{weak}^B), (\text{weak}^A \otimes \text{der}^B))^\dagger \in \mathbf{C}(!A \otimes !B, !(A \times B))$.

As usual, the (co)Kleisli category $\mathbf{C}_!$ over the comonad $(!, \text{dig}, \text{der})$ is defined to have the same objects as \mathbf{C} and $\mathbf{C}_!(A, B) := \mathbf{C}(!A, B)$. Composition in $\mathbf{C}_!$ is denoted by $;$ and defined as $\varphi; \psi := \text{dig}; !\varphi; \psi$ and identities $A := \text{der}^A$.

Theorem II.2. The Kleisli category $\mathbf{C}_!$ of a Lafont category \mathbf{C} is cartesian closed.

Indeed, the structure of cartesian smcc of \mathbf{C} is lifted to a cartesian closed structure in $\mathbf{C}_!$ by the isomorphisms m . The exponential object $A \Rightarrow B$ is defined as $!A \multimap B$ and the morphism $\text{Eval}^{A,B} \in \mathbf{C}_!((A \Rightarrow B) \times A, B)$ is given by $(m^{!A \multimap B, A})^{-1}; (\text{der}^{!A \multimap B} \otimes !A); \text{eval}^{!A, B}$. This defines an exponentiation since for every $\varphi \in \mathbf{C}_!(C \times A, B)$ there is a unique morphism $\Lambda(\varphi) := \lambda(m^{C, A}; \varphi) \in \mathbf{C}_!(C, A \Rightarrow B)$ satisfying $\Lambda(\varphi) \times A; \text{Eval} = \varphi$.

C. Constructing Lafont Categories

It is known in the *folklore*, and not difficult to check, that an smcc is endowed with the free commutative comonoids generated by its objects, as soon as the following conditions hold. First, the category has countable biproducts, so the monoidal structure distributes over them. Second, for every object A and $n \in \mathbb{N}$ the *symmetric* tensor power A^n exists, the intuition being that A^n provides the n -th layer of $!A$.

Proposition II.3 (Folklore, cf. [22]). *An smcc \mathbf{C} with countable biproducts is a Lafont category whenever:*

- there is the equalizer (A^n, eq^{A^n}) of the $n!$ symmetries of the n -fold tensor $A^{\otimes n}$, for every $n \in \mathbb{N}$ and object A ;
- such an equalizer is preserved by the tensor product, i.e., for every B , $(A^n \otimes B, \text{eq}^{A^n \otimes B})$ is the equalizer of the diagram made of all morphisms $\sigma \otimes \text{id}^B$, where σ a symmetry of $A^{\otimes n}$.

Indeed, following the recipe in [22], one constructs the free commutative comonoid as $!A := \prod_{n \in \mathbb{N}} A^n$, with multiplication and unit given by:

$$\text{contr}^A := \langle \langle \pi^{n+m}; c^{n,m} \rangle_{m \in \mathbb{N}}; \cong \rangle_{n \in \mathbb{N}}; \cong, \quad \text{weak}^A := \pi^0,$$

where \cong is the distributivity of the tensor over countable (bi)products and $c^{n,m}$ is the unique morphism such that $c^{n,m}; (\text{eq}^{A^n} \otimes \text{eq}^{A^m}) = \text{eq}^{A^{n+m}}$. The dereliction is given by $\text{der}^A := \pi^1$. The following lemma describes more concretely the action of $!$ on morphisms.

Lemma II.4. *For every $\varphi \in \mathbf{C}(A, B)$, we have that $!\varphi = \langle \pi^n; \varphi^n \rangle_{n \in \mathbb{N}}$, where φ^n is the unique morphism such that $\varphi^n; \text{eq}^{B^n} = \text{eq}^{A^n}; \varphi^{\otimes n}$, which exists by applying the universal property of the equalizer (B^n, eq^{B^n}) to $\text{eq}^{A^n}; \varphi^{\otimes n}$.*

D. Continuous \mathcal{R} -Categories

Continuous semirings have been introduced in [23] and are instances of continuous algebras (see e.g. [24]). In this section we consider categories whose hom-sets have the structure of continuous modules over continuous semirings.

Recall that a *complete partial order* (cpo) is a partially ordered set (X, \preceq) having a bottom element and such that any directed subset $D \subseteq X$ has a supremum $\bigvee D$.

A (unary) operator F on cpo's is *continuous* if it is monotone and preserves directed suprema, i.e. $F(\bigvee_{i \in I} x_i) = \bigvee_{i \in I} F(x_i)$. Similarly, we say that an n -ary operator F is *continuous* if it is continuous in each component.

Definition II.5. *A continuous semiring \mathcal{R} is a semiring $(|\mathcal{R}|, +, \cdot, \mathbf{0}, \mathbf{1})$ equipped with a partial order \preceq such that:*

- $(|\mathcal{R}|, \preceq)$ is a cpo having $\mathbf{0}$ as bottom element,
- the operators $+$ and \cdot are continuous.

When $p \preceq q$ if and only if there is $r \in |\mathcal{R}|$ such that $p+r = q$, we say that \preceq is *natural* and that \mathcal{R} is *naturally ordered*.

We will often confuse \mathcal{R} with its underlying set $|\mathcal{R}|$.

Lemma II.6. *Given a continuous semiring \mathcal{R} and a (possibly infinite) subset $S \subseteq \mathcal{R}$, the set $\{\sum_{p \in F} p \mid F \subseteq_f S\}$ is directed, hence its supremum is defined.*

Therefore, we can define the *I-indexed sum* over \mathcal{R} as

$$\sum_{p \in I} p := \bigvee_{F \subseteq_f I} \left(\sum_{p \in F} p \right).$$

Note that every continuous semiring \mathcal{R} has a top element $\infty := \sum_{p \in \mathcal{R}} p$. In particular, $p + \infty = \infty$ for every $p \in \mathcal{R}$.

Given a set X we write \overline{X} for $X \cup \{\infty\}$ and X_\perp for $X \cup \{-\infty\}$, where $\infty, -\infty$ are fresh elements.

Example II.7. *The following semirings, endowed with the natural ordering, are continuous.*

- Boolean semiring: $\mathcal{B} := (\{\mathbf{t}, \mathbf{f}\}, \vee, \wedge, \mathbf{f}, \mathbf{t})$ where $\mathbf{f} < \mathbf{t}$.
- \mathbb{N} completed: $\mathcal{N} := (\overline{\mathbb{N}}, +, \cdot, 0, 1, \leq)$ where $+, \cdot$ are defined in the obvious way (in particular $0 \cdot \infty = 0 = \infty \cdot 0$). Note that for every infinite $S \subseteq \overline{\mathbb{N}}$ we have $\bigvee S = \infty$.
- Tropical semiring: $\mathcal{T} := (\overline{\mathbb{N}}, \min, +, \infty, 0, \geq)$. Note that the order is reversed so that 0 is the top element.
- Arctic semiring: $\mathcal{A} := (\overline{\mathbb{N}}_\perp, \max, +, -\infty, 0, \leq)$ where $\max, +$ are extended as usual (e.g. $(-\infty) + \infty = -\infty$).
- \mathbb{R}^+ completed: $\mathcal{P} := (\overline{\mathbb{R}^+}, +, \cdot, 0, 1, \leq)$.

A *continuous module* $(\mathcal{M}, +, 0)$ over a continuous semiring \mathcal{R} is a module over \mathcal{R} having a cpo structure such that 0 is the bottom and addition and scalar multiplication are continuous.

Definition II.8. *We call a category \mathbf{C} a continuous \mathcal{R} -category if every hom-set is endowed with a structure of continuous module over \mathcal{R} and the composition is continuous. So \mathbf{C} is a cpo-enriched category, and moreover each hom-cpo is a continuous module over \mathcal{R} .*

Let \mathbf{C} be a continuous \mathcal{R} -category. A (unary) operator $F(-)$ on hom-sets of \mathbf{C} is *linear* if it preserves the structure of continuous module over \mathcal{R} , that is:

$$F(0) = 0, \quad F(p\varphi) = pF(\varphi), \quad F(\varphi + \psi) = F(\varphi) + F(\psi).$$

An n -ary operator F is *multilinear*, if it is linear in each component. A morphism $\varphi \in \mathbf{C}(A, B)$ is called: *pre-linear* when the operator $-; \varphi$ is linear; *post-linear* when the operator $\varphi; -$ is linear; *linear* when it is both pre- and post-linear.

If \mathbf{C} is moreover cartesian and has an object of numerals \mathfrak{N} , we say that \mathfrak{N} is *linear* if pred and succ are linear, and *zero?* is linear in its first component (i.e. $(-\times \varphi); \text{zero?}$ is linear).

Definition II.9. *A continuous \mathcal{R} -category \mathbf{C} is called pre-linear (resp. post-linear, linear) whenever all its morphisms are pre-linear (resp. post-linear, linear).*

For ccc's, Definition II.8 is extended as follows.

Definition II.10. *A post-linear continuous \mathcal{R} -ccc is a ccc \mathbf{C} that satisfies the conditions of Definition II.8, is post-linear and moreover is such that the pairing is continuous and the currying is continuous and linear.*

Therefore, a post-linear continuous \mathcal{R} -ccc is not just a post-linear \mathcal{R} -category that happens to be cartesian closed.

Remark II.11. *Since $\langle \varphi, \psi \rangle; \text{Eval} = \langle \text{id}, \psi \rangle; \Lambda^{-1}(\varphi)$ in every post-linear continuous \mathcal{R} -ccc Eval is linear in its first component (i.e. $\langle -, \psi \rangle; \text{Eval}$ is linear).*

III. THE CATEGORY \mathcal{R}^Π

Let us consider fixed an (arbitrary) continuous semiring $\mathcal{R} = (|\mathcal{R}|, \mathbf{0}, \mathbf{1}, +, \cdot, \preceq)$, whose product \cdot is commutative (as in Example II.7). Note that \mathcal{R} can be seen as a one-object category whose morphisms are the elements of \mathcal{R} , composition is the product \cdot , and the identity is given by $\mathbf{1}$.

Given a set A and $a, a' \in A$, define the *Kronecker symbol* $\delta_{a,a'} \in \mathcal{R}$ which takes value $\mathbf{1}$ if $a = a'$ and $\mathbf{0}$ if $a \neq a'$.

The *free biproduct completion* of the category \mathcal{R} , denoted by \mathcal{R}^Π , is defined as follows (cf. [25, §VIII.2 Exercise 6]).

Definition III.1. *The objects of \mathcal{R}^Π are sets and the morphisms from A to B are the matrices in $\mathcal{R}^{A \times B}$. Identity over A is the diagonal matrix defined as $\text{id}_{a,a'}^A := \delta_{a,a'}$ for all $a, a' \in A$. The composition of $\varphi \in \mathcal{R}^\Pi(A, B)$ and $\psi \in \mathcal{R}^\Pi(B, C)$ is the morphism $\varphi; \psi$ given by the usual matrix composition $(\varphi; \psi)_{a,c} := \sum_{b \in B} \varphi_{a,b} \cdot \psi_{b,c}$ for all $a \in A, c \in C$.*

Note that, despite the fact that $(\varphi; \psi)_{a,c}$ can be an infinite sum, it is always well-defined by Lemma II.6.

By construction, the category \mathcal{R}^Π has (countable) biproducts, represented by disjoint union and indicated as $\&$. Indeed, given a (possibly infinite) set I of indices we have:

$$\&_{i \in I} A_i := \bigcup_{i \in I} \{i\} \times A_i, \quad \pi_{(i,a),a'}^j := \iota_{a,(i,a')}^j := \delta_{(i,a),(j,a')}$$

where π^j (resp. ι^j) stands for the canonical projection on A_j (resp. injection from A_j). Moreover, given $\varphi_j \in \mathcal{R}^\Pi(B, A_j)$ and $\psi_j \in \mathcal{R}^\Pi(A_j, B)$ we have that

$$\langle \langle \varphi_i \rangle_{i \in I} \rangle_{b,(j,a)} := (\varphi_j)_{b,a}, \quad \langle \langle \psi_i \rangle_{i \in I} \rangle_{(j,a),b} := (\psi_j)_{a,b},$$

are the unique morphisms satisfying $\langle \varphi_i \rangle_{i \in I}; \pi^j = \varphi_j$ and $\iota^j; \langle \psi_i \rangle_{i \in I} = \psi_j$. The terminal (actually null) object \top is \emptyset .

We now show that the hom-sets of \mathcal{R}^Π inherit from \mathcal{R} the structure of continuous module.

Definition III.2. *Given two sets A, B , define for all matrices $\varphi, \psi \in \mathcal{R}^{A \times B}$ and scalars $p \in \mathcal{R}$ the following operations:*

$$0_{a,b} := \mathbf{0}, \quad (\varphi + \psi)_{a,b} := \varphi_{a,b} + \psi_{a,b}, \quad (p\varphi)_{a,b} := p \cdot \varphi_{a,b}.$$

Moreover, we set $\varphi \preceq \psi$ iff $\varphi_{a,b} \preceq \psi_{a,b}$ for all $a \in A, b \in B$.

Proposition III.3. *\mathcal{R}^Π , endowed with the operations and the ordering of Definition III.2, is a linear continuous \mathcal{R} -category.*

A. The Linear Structure

We briefly present the monoidal structure of \mathcal{R}^Π , showing that it is a \star -autonomous category (actually, compact closed).

The bifunctor $\otimes : \mathcal{R}^\Pi \times \mathcal{R}^\Pi \rightarrow \mathcal{R}^\Pi$ acts on objects like the cartesian product and on morphisms like the Kronecker product, that is (for every $\varphi \in \mathcal{R}^\Pi(A, B)$, $\psi \in \mathcal{R}^\Pi(C, D)$):

$$A \otimes B := A \times B, \quad (\varphi \otimes \psi)_{(a,c),(b,d)} := \varphi_{a,b} \cdot \psi_{c,d}$$

Bifunctoriality of this operation follows from commutativity of the \mathcal{R} -product \cdot . The unit of the tensor is the singleton set $1 := \{*\}$. Usual calculations show that $\alpha_{((a,b),c),(a',(b',c'))}^{A,B,C} := \delta_{(a,b,c),(a',b',c')}$ is a natural isomorphism giving the associativity of \otimes , while $\rho_{(*,a),a'}^A := \delta_{a,a'}$ and

$\lambda_{(a,*) , a'}^A := \delta_{a,a'}$ give the neutrality of 1 . The tensor product is moreover continuous, bilinear and symmetric, thanks to the symmetries $\sigma_{(a,b),(b',a')}^{A,B} := \delta_{(a,b),(a',b')}$.

The category \mathcal{R}^Π is monoidal closed. The monoidal exponential object and the monoidal evaluation are defined as:

$$A \multimap B := A \times B, \quad \text{eval}_{((a,b),a'),b'}^{A,B} := \delta_{(a,b),(a',b')}, \\ \lambda(\varphi)_{c,(a,b)} := \varphi_{(c,a),b}.$$

It is easy to check that $\lambda(-)$ is continuous and linear.

Notice that the object $\perp := \{*\}$ is dualizing since, for every object A , the morphism $\partial^{A,\perp} \in \mathcal{R}^\Pi(A, A^{\perp\perp})$ defined as $\partial_{a,((a',*),*)}^{A,\perp} := \delta_{a,a'}$ is an isomorphism whose inverse is $\partial_{((a,*) ,*),a'}^- := \delta_{a,a'}$, therefore \mathcal{R}^Π is \star -autonomous.

Proposition III.4. *The linear continuous \mathcal{R} -category \mathcal{R}^Π is \star -autonomous and has countable biproducts. The tensor product and monoidal currying are both continuous and (bi)linear.*

B. Constructing Lafont Exponentials in \mathcal{R}^Π

In this section we show that \mathcal{R}^Π has all symmetric tensor powers A^n . In order to describe them concretely, we need to introduce some notions and notations concerning multisets.

Let A be a set. We represent a finite multiset m over A as an unordered list $[a_1, \dots, a_n]$ with repetitions and say that n is its cardinality. The union of two multisets m_1, m_2 is written as $m_1 + m_2$. For every $n \in \mathbb{N}$, we denote by $\mathcal{M}_n(A)$ the set of all multisets over A of cardinality n . The set of all finite multisets over A is then defined as $\mathcal{M}_f(A) := \bigcup_{n \in \mathbb{N}} \mathcal{M}_n(A)$.

Lemma III.5. *For every $n \in \mathbb{N}$ and object A , the equalizer (A^n, eq^{A^n}) of the symmetries of $A^{\otimes n}$ exists and is defined by*

$$A^n := \mathcal{M}_n(A), \quad \text{eq}_{m,(a_1, \dots, a_n)}^{A^n} := \delta_{m,[a_1, \dots, a_n]}.$$

These equalizers are preserved by the tensor products.

From the above lemma and Proposition III.4, we get the following corollary of Proposition II.3.

Corollary III.6. *\mathcal{R}^Π is a Lafont category.*

Therefore we can build the exponential as in Subsection II-C:

$$!A := \&_{n \in \mathbb{N}} A^n \cong \mathcal{M}_f(A), \quad \text{der}_{m,a}^A := \delta_{m,[a]}, \\ \text{contr}_{m,(m_1, m_2)}^A := \delta_{m, m_1 + m_2}, \quad \text{weak}_{m,*}^A := \delta_{m, []}.$$

Let $\varphi \in \mathcal{R}^\Pi(A, B)$. From Lemma II.4 we get the following description of the matrix $!\varphi$ ($\forall m \in !A, \forall [b_1, \dots, b_n] \in !B$):

$$!\varphi_{m,[b_1, \dots, b_n]} = \sum_{\substack{(a_1, \dots, a_n) \text{ s.t.} \\ m = [a_1, \dots, a_n]}} \prod_{i=1}^n \varphi_{a_i, b_i}.$$

The concrete presentation of the digging is given by ($\forall m \in !A, \forall [m_1, \dots, m_n] \in !!A$):

$$\text{dig}_{m,[m_1, \dots, m_n]}^A = \delta_{m, m_1 + \dots + m_n}.$$

This matrix is actually the digging, since it is the unique comonoid morphism satisfying $\text{dig}^A; \text{der}^{!A} = \text{id}^{!A}$.

The canonical isomorphism $m^{A,B}$ between $!A \otimes !B$ and $!(A \& B)$ maps the pair $([a_1, \dots, a_n], [b_1, \dots, b_k])$ to the multiset $[(1, a_1), \dots, (1, a_n), (2, b_1), \dots, (2, b_k)]$. Analogously, the isomorphism m^\top between $! \top$ and $! \top$ sends $*$ to the multiset $[\]$. We treat these bijections as equalities, for instance we still denote by (m_1, m_2) the corresponding element of $!(A \& B)$.

C. The Kleisli Category $\mathcal{R}_!^\Pi$

The Kleisli category of \mathcal{R}^Π over the comonad $!$ can be directly described as follows. The objects of $\mathcal{R}_!^\Pi$ are all the sets, a morphism from A to B is a matrix in $\mathcal{R}^{\mathcal{M}_f(A) \times B}$, that is $\mathcal{R}_!^\Pi(A, B) := \mathcal{R}^\Pi(\mathcal{M}_f(A), B)$. The composition of morphisms $\varphi \in \mathcal{R}_!^\Pi(A, B)$ and $\psi \in \mathcal{R}_!^\Pi(B, C)$ is given by:

$$(\varphi;_! \psi)_{m,c} := \sum_{[b_1, \dots, b_n] \in !B} \sum_{\substack{(m_1, \dots, m_n) \text{ s.t.} \\ m = m_1 + \dots + m_n}} \psi_{[b_1, \dots, b_n], c} \cdot \prod_{i=1}^n \varphi_{m_i, b_i}$$

The identity on A is given by $A_{m,a} := \delta_{m, [a]}$.

For the sake of simplicity the points of A , which are the maps in $\mathcal{R}_!^\Pi(\top, A)$, will be represented as vectors in \mathcal{R}^A .

From Proposition III.3 and the Kleisli construction, it follows that $\mathcal{R}_!^\Pi$, endowed with the operations and the ordering of Definition III.2, is a post-linear continuous \mathcal{R} -category in the sense of Definition II.8. In particular, every $\varphi \in \mathcal{R}_!^\Pi(A, B)$ can be seen as a continuous map from \mathcal{R}^A to \mathcal{R}^B by setting $\varphi(\vartheta) := \vartheta;_! \varphi$ for all vectors $\vartheta \in \mathcal{R}^A$.

The cartesian structure of \mathcal{R}^Π is preserved in $\mathcal{R}_!^\Pi$, therefore the product of an indexed family $(A_i)_{i \in I}$ is still $\&_{i \in I} A_i$, while the j -th projection is $\pi_{m,a}^j = \delta_{m, [(j,a)]}$. The exponential object $A \Rightarrow B$ is $\mathcal{M}_f(A) \times B$, the evaluation morphism $\text{Eval} \in \mathcal{R}_!^\Pi((A \Rightarrow B) \& A, B)$ is defined as $\text{Eval}_{(m,m'), b} = \delta_{m, [(m', b)]}$ and the currying $\Lambda(\varphi) \in \mathcal{R}_!^\Pi(C, A \Rightarrow B)$ of a morphism $\varphi \in \mathcal{R}_!^\Pi(C \& A, B)$ is given by $\Lambda(\varphi)_{m, (m', b)} = \varphi_{(m, m'), b}$.

The tuple $\mathfrak{N} = (\mathbb{N}, z, \text{succ}, \text{pred}, \text{zero}?)$ defined as:

$$\begin{aligned} z_n &:= \delta_{n, 0}, \\ \text{pred}_{m,n} &:= \delta_{n, 0} \cdot \delta_{m, [0]} + \delta_{m, [n+1]}, \\ \text{succ}_{m,n} &:= \sum_{k \in \mathbb{N}} \delta_{m, [k]} \cdot \delta_{n, k+1}, \\ \text{zero}?(m, m_1, m_2), n &:= \begin{cases} 1 & \text{if } (m, m_1, m_2) = ([0], [n], [\]), \\ & \text{or } (m, m_1, m_2) = ([k+1], [\], [n]), \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

is an object of numerals living in $\mathcal{R}_!^\Pi$.

Theorem III.7. *The category $\mathcal{R}_!^\Pi$ is a post-linear continuous \mathcal{R} -ccc. Moreover \mathfrak{N} is linear.*

Clearly **PCF** can be interpreted in $\mathcal{R}_!^\Pi$ since it is a cpo-enriched ccc having an object of numerals. In the interpretation of a **PCF** term in $\mathcal{R}_!^\Pi$ several scalars in \mathcal{R} appear. The next section is devoted to investigating the meaning of such scalars.

IV. THE LANGUAGE **PCF** ^{\mathcal{R}}

We now define **PCF** ^{\mathcal{R}} , a prototypical programming language extending **PCF** [2] with a nondeterministic choice operator “or” and scalars from \mathcal{R} . This opens the way for modeling quantitative effects.

Typing Rules of **PCF** ^{\mathcal{R}}

$$\begin{array}{c} \frac{}{\Gamma, x : A \vdash x : A} \quad \frac{\Gamma \vdash M : A}{\Gamma \vdash pM : A} \quad \frac{\Gamma \vdash M : A \quad \Gamma \vdash P : A}{\Gamma \vdash M \text{ or } P : A} \\ \\ \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x^A. M : A \rightarrow B} \quad \frac{\Gamma \vdash M : A \rightarrow B \quad \Gamma \vdash P : A}{\Gamma \vdash MP : B} \\ \\ \frac{}{\Gamma \vdash \underline{0} : \text{int}} \quad \frac{\Gamma \vdash M : \text{int}}{\Gamma \vdash \text{pred } M : \text{int}} \quad \frac{\Gamma \vdash M : \text{int}}{\Gamma \vdash \text{succ } M : \text{int}} \\ \\ \frac{\Gamma \vdash M : \text{int} \quad \Gamma \vdash P : \text{int} \quad \Gamma \vdash L : \text{int}}{\Gamma \vdash \text{ifz}(M, P, L) : \text{int}} \quad \frac{\Gamma \vdash M : A \rightarrow A}{\Gamma \vdash \mathbf{Y}M : A} \end{array}$$

(a) The typing rules of **PCF** ^{\mathcal{R}} . The type annotation on the lambda-abstraction ensures that the derivation is unique, given a context Γ and term M .

Reduction Rules

$$\begin{array}{ll} \beta : & (\lambda x. M)P \xrightarrow{1} M[P/x] \\ \text{fix} : & \mathbf{Y}M \xrightarrow{1} M(\mathbf{Y}M) \\ \text{scal} : & pM \xrightarrow{p} M \\ \text{or}_l : & M \text{ or } P \xrightarrow{1} M \\ \text{or}_r : & M \text{ or } P \xrightarrow{1} P \\ \text{pred} : & \text{pred } \underline{n} \xrightarrow{1} \underline{n-1} \\ \text{if}_0 : & \text{ifz}(\underline{0}, P, L) \xrightarrow{1} P \\ \text{if}_s : & \text{ifz}(\underline{n+1}, P, L) \xrightarrow{1} L \end{array}$$

(b) Redex-to-contractum rules. In the rule **pred** we suppose that $0 - 1 = 0$. We write $M \xrightarrow{\ell} P$ to mean that M reduces to P using the rule (ℓ) .

Contextual Rules

$$\begin{array}{ll} MP \xrightarrow{p} M'P & \text{pred } M \xrightarrow{p} \text{pred } M' \\ \text{ifz}(M, P, L) \xrightarrow{p} \text{ifz}(M', P, L) & \text{succ } M \xrightarrow{p} \text{succ } M' \end{array}$$

(c) Contextual rules. Supposing $M \xrightarrow{p} M'$ using the rule (ℓ) .

Fig. 1. Typing rules and operational semantics of **PCF** ^{\mathcal{R}} .

Definition IV.1. *(The language **PCF** ^{\mathcal{R}}) The set of types contains all arrow types built from the ground type **int**.*

The set of terms is generated by (for $p \in \mathcal{R}$):

$$L, M, P ::= x \mid \lambda x^A. M \mid MP \mid \mathbf{Y}M \mid \underline{0} \mid \text{pred } M \mid \text{succ } M \mid \text{ifz}(M, P, L) \mid pM \mid M \text{ or } P$$

For all $n \in \mathbb{N}$ we write $M^n(P)$ for $M(M(\dots(MP)\dots))$ (n times) and \underline{n} for $\text{succ}^n(\underline{0})$.

The notions of α -conversion, free and bound variable, and substitution $M[P/x]$ are defined as usual in λ -calculus [26, §2]. Hereafter, terms are considered up to α -conversion.

Example IV.2. *Concerning specific **PCF** ^{\mathcal{R}} terms, we set:*

- $\Phi := \lambda x^{\text{int}}. \text{ifz}(x, \text{succ } x, \underline{0})$,
- $\Omega := \mathbf{Y}(\lambda x^{\text{int}}. x)$,
- $\Psi := \mathbf{Y}(\lambda x^{\text{int}}. (x \text{ or } \underline{0}))$.

These terms will be used as examples throughout the paper.

A context Q is a **PCF** ^{\mathcal{R}} term having a single occurrence of a “hole”, denoted by $[-]$, inside. Given a context $Q[-]$ and

$$\Phi P \xrightarrow[\beta]{1} \text{ifz}(P, \text{succ } P, \underline{0}) \begin{cases} \xrightarrow[\text{or}_1]{1} \text{ifz}(p\underline{0}, \text{succ } P, \underline{0}) \xrightarrow[\text{scal}]{p} \text{ifz}(\underline{0}, \text{succ } P, \underline{0}) \xrightarrow[\text{if}_0]{1} \text{succ } P \xrightarrow[\text{or}_1]{1} \text{succ}(p\underline{0}) \xrightarrow[\text{scal}]{p} \text{succ } \underline{0} := \underline{1} \\ \xrightarrow[\text{or}_r]{1} \text{ifz}(q\underline{1}, \text{succ } P, \underline{0}) \xrightarrow[\text{scal}]{q} \text{ifz}(\underline{1}, \text{succ } P, \underline{0}) \xrightarrow[\text{if}_s]{1} \underline{0} \xrightarrow[\text{or}_r]{1} \text{succ}(q\underline{1}) \xrightarrow[\text{scal}]{q} \text{succ } \underline{1} := \underline{2} \end{cases}$$

Fig. 2. Example of reduction sequences starting from ΦP , where P is the weighted nondeterministic numeral $p\underline{0}$ or $q\underline{1}$.

a term M we write $Q[M]$ for the result of substituting M for the hole $[-]$ in Q , possibly with capture of free variables.

(Type) environments are finite maps from variables to types. We write $x_1 : A_1, \dots, x_n : A_n$ to denote the environment Γ such that $\text{dom}(\Gamma) = \{x_1, \dots, x_n\}$ and $\Gamma(x_i) = A_i$ for all i .

(Type) judgements are denoted by $\Gamma \vdash M : A$ and can be inferred using the typing rules of Figure 1(a).

Remark IV.3. The terms of Example IV.2 are well-typed: Ω and Ψ are of type int , while Φ is of type $\text{int} \rightarrow \text{int}$.

Hereafter, we only consider well-typed terms.

Definition IV.4. The operational semantics of $\text{PCF}^{\mathcal{R}}$ is defined in Figures 1(b), 1(c).

- The reduction rules defined in Figure 1(b) are treated as relations between terms, decorated with a weight $p \in \mathcal{R}$ and a label $\ell \in \{\beta, \text{fix}, \text{scal}, \text{or}_1, \text{or}_r, \text{pred}, \text{if}_0, \text{if}_s\}$. In each rule (ℓ), the term at the left-hand side is a redex, while the term at the right-hand side is its contractum.
- The elementary reduction step (ers) $M \xrightarrow[p]{\ell} P$ is the least (quaternary) relation closed under the above reduction rules and the contextual rules of Figure 1(c).
- A term M is a normal form whenever there are no weight p , term P and label ℓ such that $M \xrightarrow[p]{\ell} P$.

The operational semantics implements the leftmost-outermost reduction strategy. The label ℓ is needed in the ers relation to ensure that there are two distinct reductions from M or M to M .

We write $M \xrightarrow{p} P$ to mean that $M \xrightarrow[p]{\ell} P$ for some label ℓ .

Example IV.5. Consider the terms of Example IV.2.

1. The behaviour of Φ on numerals is easy to determine, indeed $\Phi \underline{0} \xrightarrow[\beta]{1} \text{ifz}(\underline{0}, \underline{1}, \underline{0}) \xrightarrow[\text{if}_0]{1} \underline{1}$ and, for all $n > 0$, $\Phi \underline{n} \xrightarrow[\beta]{1} \text{ifz}(\underline{n}, \underline{n+1}, \underline{0}) \xrightarrow[\text{if}_s]{1} \underline{0}$.
2. The reduction of Φ is more interesting on weighted nondeterministic numerals, like $P := p\underline{0}$ or $q\underline{1}$ (see Figure 2).
3. Clearly, we have $\Omega \xrightarrow[\text{fix}]{1} (\lambda x^{\text{int}}.x)\Omega \xrightarrow[\beta]{1} \Omega \xrightarrow{1} \dots$.
4. $\Psi \xrightarrow[\text{fix}]{1} (\lambda x^{\text{int}}.(x \text{ or } \underline{0}))\Psi \xrightarrow[\beta]{1} \Psi \text{ or } \underline{0}$ which reduces with weight 1 using the or -rules either to Ψ itself or to $\underline{0}$.

Remark that every term has at most one redex that reduces, moreover the reduction is deterministic except for the or -constructor. By induction one proves the following lemma.

Lemma IV.6 (Subject reduction). If $M \xrightarrow{p} P$ and $\Gamma \vdash M : A$, then $\Gamma \vdash P : A$.

Definition IV.7. Let M, P be two terms.

- A reduction sequence π from M to P is a finite sequence $(M_i \xrightarrow{p_i} M_{i+1})_{i < k}$ of elementary reduction steps such that

$M_0 = M$ and $M_k = P$. In particular, for all M , there is an empty reduction sequence ϵ from M to itself.

- The set of all reduction sequences from M to P of length at most k is denoted by $M \Rightarrow^{\leq k} P$.
- The set $M \Rightarrow P$ of all reduction sequences from M to P is defined as $\bigcup_{k \in \mathbb{N}} (M \Rightarrow^{\leq k} P)$.

As elementary reduction steps are weighted, it makes sense to define the weight of a (set of) reduction sequence(s).

Definition IV.8. Let M, P be two terms.

- The weight of a reduction sequence $\pi \in M \Rightarrow P$ where $\pi := (M_i \xrightarrow{p_i} M_{i+1})_{i < k}$ is defined as $w(\pi) := \prod_{i < k} p_i \in \mathcal{R}$. Note that $w(\epsilon) = 1$.
- The above operation is extended to a subset $A \subseteq M \Rightarrow P$ by setting $w(A) := \sum_{\pi \in A} w(\pi)$.

Remark that $w(M \Rightarrow P)$ is always defined by Lemma II.6.

Example IV.9. Consider the terms of Example IV.2.

1. From Example IV.5.1 we have that $w(\Phi \underline{n} \Rightarrow \underline{k})$ is equal to 1 if either $n = 0$ and $k = 1$, or $n > 0$ and $k = 0$; otherwise it is equal to 0.
2. Weights can be used to carry information on resource consumption. For instance, Figure 2 gives (for $P := p\underline{0}$ or $q\underline{1}$): $w(\Phi P \Rightarrow \underline{1}) = p^2$, $w(\Phi P \Rightarrow \underline{0}) = q$ and $w(\Phi P \Rightarrow \underline{2}) = p \cdot q$. The degree of the parameter p (resp. q) corresponds to the number of times the term ΦP uses the resource $p\underline{0}$ (resp. $q\underline{1}$) during the reduction to a numeral.
3. From Example IV.5.3 it follows that, for all $n \in \mathbb{N}$, we have $\Omega \Rightarrow \underline{n} = \emptyset$ and therefore $w(\Omega \Rightarrow \underline{n}) = 0$.

A. Abstract Denotational Semantics

Let us fix a post-linear continuous \mathcal{R} -ccc \mathbf{C} with a linear object of numerals \mathfrak{N} . We interpret $\text{PCF}^{\mathcal{R}}$ in \mathbf{C} by extending the standard interpretation of PCF [27, §6].

As usual, types are interpreted by:

$$\llbracket \text{int} \rrbracket := N, \quad \llbracket A \rightarrow B \rrbracket := \llbracket A \rrbracket \Rightarrow \llbracket B \rrbracket.$$

Given an environment $\Gamma = x_1 : A_1, \dots, x_n : A_n$, its interpretation is $\llbracket \Gamma \rrbracket := \prod_{i=1}^n \llbracket A_i \rrbracket$. To lighten the notations we will confuse types and environments with their interpretations.

Definition IV.10. The interpretation of a term M having type B in an environment Γ , is the morphism $\llbracket M \rrbracket^\Gamma \in \mathbf{C}(\llbracket \Gamma \rrbracket, \llbracket B \rrbracket)$ defined by induction as follows:

- $\llbracket x_i \rrbracket^\Gamma := \pi^i$,
- $\llbracket \lambda x^A. M \rrbracket^\Gamma := \Lambda(\llbracket M \rrbracket^{\Gamma, x:A})$ where $x \notin \text{dom}(\Gamma)$,
- $\llbracket MP \rrbracket^\Gamma := \langle \llbracket M \rrbracket^\Gamma, \llbracket P \rrbracket^\Gamma \rangle; \text{Eval}$,
- $\llbracket \mathbf{Y}M \rrbracket^\Gamma := \bigvee_{n \in \mathbb{N}} \text{fix}^n(\llbracket M \rrbracket^\Gamma)$,

- $\llbracket 0 \rrbracket^\Gamma := \top^\Gamma; z,$
- $\llbracket \text{pred } M \rrbracket^\Gamma := \llbracket M \rrbracket^\Gamma; \text{pred},$
- $\llbracket \text{succ } M \rrbracket^\Gamma := \llbracket M \rrbracket^\Gamma; \text{succ},$
- $\llbracket \text{ifz}(M, P, L) \rrbracket := \langle \llbracket M \rrbracket^\Gamma, \langle \llbracket P \rrbracket^\Gamma, \llbracket L \rrbracket^\Gamma \rangle \rangle; \text{zero?},$
- $\llbracket pM \rrbracket^\Gamma := p\llbracket M \rrbracket^\Gamma,$
- $\llbracket M \text{ or } P \rrbracket^\Gamma := \llbracket M \rrbracket^\Gamma + \llbracket P \rrbracket^\Gamma.$

where $\text{fix}^n(\varphi)$ is defined by induction on $n \in \mathbb{N}$ as

$$\text{fix}^0(\varphi) := 0, \quad \text{fix}^{n+1}(\varphi) := \langle \varphi, \text{fix}^n(\varphi) \rangle; \text{Eval}.$$

Remark that $B := N$ in the rules for pred , succ and ifz .

The fact that the family $(\text{fix}^n(\llbracket M \rrbracket^\Gamma))_{n \in \mathbb{N}}$ is increasing follows from the assumptions of continuity in Definition II.10. By induction one proves that the substitution lemma holds.

Lemma IV.11 (Substitution). $\Gamma, x : A \vdash M : B$ and $\Gamma \vdash P : A$ entail $\llbracket M[P/x] \rrbracket^\Gamma = \langle \Gamma, \llbracket P \rrbracket^\Gamma \rangle; \llbracket M \rrbracket^{\Gamma, x:A}.$

Proposition IV.12 (Soundness). For every term M which is not a normal form, we have:

$$\llbracket M \rrbracket^\Gamma = \sum_{M \xrightarrow{\epsilon} L} p\llbracket L \rrbracket^\Gamma.$$

B. Denotational Semantics in \mathcal{R}_\dagger^Π

We now describe the interpretation of terms in \mathcal{R}_\dagger^Π . From Theorem III.7 and Proposition IV.12 it follows that \mathcal{R}_\dagger^Π is a sound model of $\mathbf{PCF}^{\mathcal{R}}$.

Notice that, up to isomorphism, the interpretation $\llbracket M \rrbracket^\Gamma \in \mathcal{R}_\dagger^\Pi(\Gamma, B)$, where $\Gamma = x_1 : A_1, \dots, x_n : A_n$, is a matrix

$$\llbracket M \rrbracket^\Gamma \in \mathcal{R}^{\mathcal{M}_f(A_1) \times \dots \times \mathcal{M}_f(A_n) \times B}.$$

When the underlying category is not clear from the context we write $\llbracket M \rrbracket^{\mathcal{R}, \Gamma}$ to emphasize that $\llbracket M \rrbracket^\Gamma$ lives in \mathcal{R}_\dagger^Π .

Some interpretations in Definition IV.10 admit a more concrete description which is given in Figure 3. For every closed term M of type A , $\llbracket \mathbf{Y}M \rrbracket$ is the least fixed point of $\llbracket M \rrbracket$, seen as a continuous map from \mathcal{R}^A to itself. Using these characterizations we can compute the following examples.

Example IV.13. Consider the terms of Example IV.2.

- $\llbracket \Phi \rrbracket_{m,n} = \begin{cases} \mathbf{1} & \text{if either } m = [0, k] \text{ and } n = k + 1 \\ & \text{or } m = [k + 1] \text{ and } n = 0, \\ \mathbf{0} & \text{otherwise.} \end{cases}$
- From the definition of Ω and the fact that the least fixed point of the identity is 0 we obtain $\llbracket \Omega \rrbracket = 0$.
- Hence $\llbracket M \text{ or } \Omega \rrbracket = \llbracket M \rrbracket$, for every term M .
- $\llbracket \lambda x^{\text{int}}.(x \text{ or } \underline{0}) \rrbracket_{m,n} = \delta_{m,[n]} + \delta_{(m,n),(\llbracket, 0 \rrbracket)}$.
- To compute $\llbracket \Psi \rrbracket$ it is enough to take the supremum for $n \in \mathbb{N}$ of $\text{fix}^n(\llbracket \lambda x^{\text{int}}.(x \text{ or } \underline{0}) \rrbracket) = z + \dots + z$ (n times).

Corollary IV.14. For every closed term M of type int we have $w(M \Rightarrow \underline{n}) \preceq \llbracket M \rrbracket_n$, for all $n \in \mathbb{N}$.

Proof: We prove by induction on k that $w(M \Rightarrow^{\leq k} \underline{n}) \preceq \llbracket M \rrbracket_n$, which implies $w(M \Rightarrow \underline{n}) \preceq \llbracket M \rrbracket_n$ since $w(M \Rightarrow \underline{n}) = \bigvee_{k \in \mathbb{N}} w(M \Rightarrow^{\leq k} \underline{n})$. In the base case, either $M = \underline{n}$ and $w(M \Rightarrow^{\leq 0} \underline{n}) = \mathbf{1} = \llbracket M \rrbracket_n$, or $M \neq \underline{n}$ and $w(M \Rightarrow^{\leq 0} \underline{n}) = \mathbf{0} \preceq \llbracket M \rrbracket_n$. The induction step follows by Proposition IV.12 and $w(M \Rightarrow^{\leq k+1} \underline{n}) = \sum_{M \xrightarrow{\epsilon} L} p \cdot w(L \Rightarrow^{\leq k} \underline{n})$. ■

V. ADEQUACY OF \mathcal{R}_\dagger^Π FOR $\mathbf{PCF}^{\mathcal{R}}$

We prove the adequacy of the model \mathcal{R}_\dagger^Π , a result relating denotational and operational semantics on closed terms of type int . More precisely, we prove that not only Corollary IV.14 holds but actually, for all $n \in \mathbb{N}$, we have $\llbracket M \rrbracket_n = w(M \Rightarrow \underline{n})$ (Theorem V.6, below). The new inequality is achieved following the lines of the adequacy proof in [14], i.e. by using logical relations (Definition V.1 and Proposition V.5, below).

Definition V.1 (Logical relations). For every type A , let \triangleleft^A be the relation between vectors in \mathcal{R}^A and closed terms of type A , defined by induction on A as follows:

$$\begin{aligned} \varphi \triangleleft^{\text{int}} M &\iff \forall n \in \mathbb{N}, \varphi_n \preceq w(M \Rightarrow \underline{n}), \\ \varphi \triangleleft^{B \rightarrow C} M &\iff \forall \psi, P, \psi \triangleleft^B P \text{ entails } \langle \varphi, \psi \rangle; \dagger \text{Eval} \triangleleft^C MP. \end{aligned}$$

Lemmas V.2, V.3 and V.4 state standard closure properties of the logical relations.

Lemma V.2. For every closed term M of type A , we have:

- $0 \triangleleft^A M,$
- if $\psi \preceq \varphi \triangleleft^A M$, then $\psi \triangleleft^A M,$
- if $\varphi_i \triangleleft^A M$ for all $i \in I$, then $\bigvee_{i \in I} \varphi_i \triangleleft^A M.$

Lemma V.3. Let M, M_i, P, P_i for $i = 1, 2$ be closed terms.

- If $M \xrightarrow{\epsilon} P$ and $\varphi \triangleleft^A P$ then $p\varphi \triangleleft^A M.$
- If $M \xrightarrow{1}_{\text{or}_1} P_1$ and $M \xrightarrow{1}_{\text{or}_2} P_2$, and $\varphi_1 \triangleleft^A P_1$ and $\varphi_2 \triangleleft^A P_2$, then $\varphi_1 + \varphi_2 \triangleleft^A M.$

Lemma V.4. Let M, P, L be closed terms such that $\varphi \triangleleft^{\text{int}} M$, $\psi \triangleleft^{\text{int}} P$ and $\vartheta \triangleleft^{\text{int}} L$. Then we have:

- $\langle \varphi, \langle \psi, \vartheta \rangle \rangle; \dagger \text{zero?} \triangleleft^{\text{int}} \text{ifz}(M, P, L),$
- $\varphi; \dagger \text{pred} \triangleleft^{\text{int}} \text{pred } M,$
- $\varphi; \dagger \text{succ} \triangleleft^{\text{int}} \text{succ } M.$

Proposition V.5. Let M be a term such that $\Gamma \vdash M : B$ where $\Gamma = x_1 : A_1, \dots, x_k : A_k$. For all maps φ_i and closed terms P_i such that $\varphi_i \triangleleft^{A_i} P_i$ (for $1 \leq i \leq k$), we have

$$\langle \varphi_1, \dots, \varphi_k \rangle; \dagger \llbracket M \rrbracket^\Gamma \triangleleft^B M[P_1/x_1, \dots, P_k/x_k].$$

Proof: To shorten the notation, we write $\vec{\varphi}$ for $\langle \varphi_i \rangle_{i \leq k}$ and \overline{M} for $M[P_1/x_1, \dots, P_k/x_k]$.

We proceed by structural induction on M . In case M is a variable x_i or the constant $\underline{0}$ the result follows trivially.

In case $M = \lambda x.M'$, we have $B = C \rightarrow D$. Let us take $\psi \triangleleft^C L$ and prove that $\langle \vec{\varphi}; \dagger \llbracket M \rrbracket^\Gamma, \psi \rangle; \dagger \text{Eval} \triangleleft^D \overline{M}L$. By induction hypothesis, we have $\langle \vec{\varphi}, \psi \rangle; \dagger \llbracket M' \rrbracket^{\Gamma, x:C} \triangleleft^D \overline{M}'L/x$. Notice that $\langle \vec{\varphi}, \psi \rangle; \dagger \llbracket M' \rrbracket^{\Gamma, x:C} = \langle \vec{\varphi}; \dagger \llbracket M \rrbracket^\Gamma, \psi \rangle; \dagger \text{Eval}$, hence we conclude by Lemma V.3 and the fact that $\overline{M}L \xrightarrow{1} \overline{M}'L/x$.

In case $M = LP$ then there exists a type C such that $\Gamma \vdash L : C \rightarrow B$ and $\Gamma \vdash P : C$. By induction hypothesis we have $\vec{\varphi}; \dagger \llbracket L \rrbracket^\Gamma \triangleleft^{C \rightarrow B} \overline{L}$, and $\vec{\varphi}; \dagger \llbracket P \rrbracket^\Gamma \triangleleft^C \overline{P}$. Hence $\langle \vec{\varphi}; \dagger \llbracket L \rrbracket^\Gamma, \vec{\varphi}; \dagger \llbracket P \rrbracket^\Gamma \rangle; \dagger \text{Eval} \triangleleft^B \overline{L}\overline{P}$ and we conclude remarking that $\langle \vec{\varphi}; \dagger \llbracket L \rrbracket^\Gamma, \vec{\varphi}; \dagger \llbracket P \rrbracket^\Gamma \rangle; \dagger \text{Eval} = \vec{\varphi}; \dagger \llbracket M \rrbracket^\Gamma$ and $\overline{L}\overline{P} = \overline{M}$.

In case $M = \mathbf{Y}L$, then the induction hypothesis gives $\vec{\varphi}; \dagger \llbracket L \rrbracket^\Gamma \triangleleft^{B \rightarrow B} \overline{L}$. By induction on n one establishes that $\vec{\varphi}; \dagger \text{fix}^n(\llbracket L \rrbracket^\Gamma) \triangleleft^B \overline{L}(\mathbf{Y}\overline{L})$, where the base of induction follows from Lemma V.2(i). From Lemma V.2(iii) we then get

$$\begin{aligned}
\llbracket x_i \rrbracket_{\vec{m},b}^\Gamma &= \delta_{m_i,[b]} \cdot \prod_{j \neq i} \delta_{m_j,[]}, & \llbracket \lambda x^A . M \rrbracket_{\vec{m},(m',b)}^\Gamma &= \llbracket M \rrbracket_{(\vec{m},m'),b}^{\Gamma,x:A}, & \llbracket MP \rrbracket_{\vec{m},b}^\Gamma &= \sum_{\substack{m' = \\ [a_1, \dots, a_k]}} \sum_{\substack{(\vec{m}_0, \dots, \vec{m}_k) \\ \sum_{i=0}^k \vec{m}_i = \vec{m}}} \llbracket M \rrbracket_{\vec{m}_0,(m',b)}^\Gamma \cdot \prod_{i=1}^k \llbracket P \rrbracket_{\vec{m}_i,a_i}^\Gamma, \\
\llbracket 0 \rrbracket_{\vec{m},n}^\Gamma &= \delta_{0,n} \cdot \prod_i \delta_{[],m_i}, & \llbracket \text{pred } M \rrbracket_{\vec{m},n}^\Gamma &= \delta_{n,0} \cdot \llbracket M \rrbracket_{\vec{m},0}^\Gamma + \llbracket M \rrbracket_{\vec{m},n+1}^\Gamma, & \llbracket \text{succ } M \rrbracket_{\vec{m},0}^\Gamma &= \mathbf{0}, & \llbracket \text{succ } M \rrbracket_{\vec{m},n+1}^\Gamma &= \llbracket M \rrbracket_{\vec{m},n}^\Gamma, \\
\llbracket \text{ifz}(M, P, L) \rrbracket_{\vec{m},n}^\Gamma &= \sum_{\substack{(\vec{m}_0, \vec{m}_1) \text{ s.t.} \\ \vec{m}_0 + \vec{m}_1 = \vec{m}}} \left(\llbracket M \rrbracket_{\vec{m}_0,0}^\Gamma \cdot \llbracket P \rrbracket_{\vec{m}_1,n}^\Gamma + \left(\sum_{k=1}^{\infty} \llbracket M \rrbracket_{\vec{m}_0,k}^\Gamma \right) \cdot \llbracket L \rrbracket_{\vec{m}_1,n}^\Gamma \right).
\end{aligned}$$

Fig. 3. Explicit characterizations of the interpretation of some terms. We suppose $\vec{m} \in !\Gamma$, $m' \in !A$, $b \in B$, $n \in \mathbb{N}$.

$\bigvee_{n \in \mathbb{N}} \bar{\varphi}; \text{fix}^n(\llbracket L \rrbracket^\Gamma) \triangleleft^B \bar{L}(\mathbf{Y}\bar{L})$ and since $\bar{M} \xrightarrow{1} \bar{L}(\mathbf{Y}\bar{L})$ we get $\bigvee_{n \in \mathbb{N}} \bar{\varphi}; \text{fix}^n(\llbracket L \rrbracket^\Gamma) \triangleleft^B \bar{M}$, by Lemma V.3(i). We conclude by remarking that $\bigvee_{n \in \mathbb{N}} \bar{\varphi}; \text{fix}^n(\llbracket L \rrbracket^\Gamma) = \bar{\varphi}; \llbracket M \rrbracket^\Gamma$.

The cases $M = \text{ifz}(M', L, P)$, $M = \text{pred } L$ and $M = \text{succ } L$ follow straightforwardly using Lemma V.4.

If $M = L$ or P , then by induction hypothesis $\bar{\varphi}; \llbracket L \rrbracket^\Gamma \triangleleft^A \bar{L}$ and $\bar{\varphi}; \llbracket P \rrbracket^\Gamma \triangleleft^A \bar{P}$. Since $\bar{M} \xrightarrow{1} \bar{L}$ and $\bar{M} \xrightarrow{1} \bar{P}$, we use Lemma V.3(ii) to get $\bar{\varphi}; \llbracket M \rrbracket^\Gamma = \bar{\varphi}; \llbracket L \rrbracket^\Gamma + \bar{\varphi}; \llbracket P \rrbracket^\Gamma \triangleleft^A \bar{M}$.

The case $M = pL$ is similar. \blacksquare

Theorem V.6 (Adequacy). *For every closed term M of type int and $n \in \mathbb{N}$ we have $\llbracket M \rrbracket_n = w(M \Rightarrow \underline{n})$*

Proof: From Corollary IV.14 and Proposition V.5. \blacksquare

A. Failure of Full Abstraction

We now show that, for every choice of \mathcal{R} , the model \mathcal{R}_1^Π is not fully abstract for $\text{PCF}^\mathcal{R}$ — it does not capture exactly the observational pre-order on terms induced by \mathcal{R} .

Let $\mathcal{C}_B^{\Gamma,A}$ be the set of contexts Q mapping terms M of type A in Γ , into terms $Q[M]$ of type B in the empty environment.

Definition V.7 (Observational pre-order). *Given $\Gamma \vdash M : A$ and $\Gamma \vdash P : A$, define*

$$M \sqsubseteq^\Gamma P \iff \forall Q \in \mathcal{C}_{\text{int}}^{\Gamma,A}, w(Q[M] \Rightarrow \underline{0}) \leq w(Q[P] \Rightarrow \underline{0}).$$

Let \equiv^Γ be the equivalence induced by \sqsubseteq^Γ .

Remark that the numeral $\underline{0}$ chosen for testing the equality is not significant. Indeed, from a context Q semi-separating M and P , i.e. such that $w(Q[M] \Rightarrow \underline{0}) \not\leq w(Q[P] \Rightarrow \underline{0})$, one can define the context $Q'[-] := \text{succ}^n(Q[-])$ satisfying $w(Q'[M] \Rightarrow \underline{n}) \not\leq w(Q'[P] \Rightarrow \underline{n})$.

Remark V.8. *By structural induction it is possible to show that $\llbracket M \rrbracket^\Gamma \leq \llbracket P \rrbracket^\Gamma$ entails $\llbracket Q[M] \rrbracket \leq \llbracket Q[P] \rrbracket$, for all $Q \in \mathcal{C}_{\text{int}}^{\Gamma,A}$.*

The model \mathcal{R}_1^Π would be (inequationally) fully abstract if, for all terms M, P : $\llbracket M \rrbracket^\Gamma \leq \llbracket P \rrbracket^\Gamma$ if and only if $M \sqsubseteq^\Gamma P$. As a corollary of the adequacy, we get the ‘only if’ direction.

Corollary V.9. *If $\llbracket M \rrbracket^\Gamma \leq \llbracket P \rrbracket^\Gamma$, then $M \sqsubseteq^\Gamma P$.*

We now show that the other implication does not hold. Let

$$\Xi := \lambda y^{\text{int}}. \infty \underline{0}, \quad \Upsilon := \lambda y^{\text{int}}. (\infty \underline{0} \text{ or } \text{ifz}(y, \underline{0}, \Omega)). \quad (1)$$

where Ω is defined in Example IV.2 and ∞ in Section II-D.

Both terms have type $\text{int} \rightarrow \text{int}$. By using the rules of Figure 3 one can easily compute their interpretations:

- $\llbracket \Xi \rrbracket_{[],0} = \infty$ and $\llbracket \Xi \rrbracket_{m,n} = \mathbf{0}$ otherwise,
- $\llbracket \Upsilon \rrbracket_{[],0} = \infty$, $\llbracket \Upsilon \rrbracket_{[0],0} = \mathbf{1}$ and $\llbracket \Upsilon \rrbracket_{m,n} = \mathbf{0}$ otherwise.

Note that $\llbracket \Xi \rrbracket \prec \llbracket \Upsilon \rrbracket$, indeed $\llbracket \Xi \rrbracket_{[0],0} = \mathbf{0} \prec \mathbf{1} = \llbracket \Upsilon \rrbracket_{[0],0}$. However, the two terms are observationally equivalent, as proven in Proposition V.11. The reasoning is standard and uses the logical relation \triangleleft^A (Definition V.1) to shrink the set of the contexts observing the operational behaviour of Ξ and Υ .

Lemma V.10. *We have $\llbracket \Upsilon \rrbracket \triangleleft^{\text{int} \rightarrow \text{int}} \Xi$.*

Proof: Let $\varphi \in \mathcal{R}^\mathbb{N}$ and P be a closed term of type int . Since $\Xi P \xrightarrow{1} \infty \underline{0} \xrightarrow{\infty} \underline{0}$ and ∞ is the top element, we have $w(\Xi P \Rightarrow \underline{0}) = \infty \succeq \langle \llbracket \Upsilon \rrbracket, \varphi \rangle; \text{Eval}_0$.

For $n > 0$ we have $\llbracket \Upsilon \rrbracket_{m,n} = \mathbf{0}$, hence $\langle \llbracket \Upsilon \rrbracket, \varphi \rangle; \text{Eval}_n = \sum_{m=[n_1, \dots, n_k]} \llbracket \Upsilon \rrbracket_{m,n} \cdot \prod_{i=1}^k \varphi_{n_i} = \mathbf{0} \preceq w(\Xi P \Rightarrow \underline{n})$.

So we obtain $\langle \llbracket \Upsilon \rrbracket, \varphi \rangle; \text{Eval} \triangleleft^{\text{int}} \Xi P$ and since φ and P are arbitrary, we conclude $\llbracket \Upsilon \rrbracket \triangleleft^{\text{int} \rightarrow \text{int}} \Xi$. \blacksquare

Proposition V.11. *Υ and Ξ are observationally equivalent.*

Proof: For any context $Q \in \mathcal{C}_{\text{int}}^{\text{int} \rightarrow \text{int}}$ and closed term M of type $\text{int} \rightarrow \text{int}$, we have $(\lambda x^{\text{int} \rightarrow \text{int}}. Q[x])M \xrightarrow{1} Q[M]$. Therefore $M \sqsubseteq M'$ if and only if $w(LM \Rightarrow \underline{0}) \leq w(LM' \Rightarrow \underline{0})$, for every closed term $L : (\text{int} \rightarrow \text{int}) \rightarrow \text{int}$.

From Proposition V.5 we get $\llbracket L \rrbracket \triangleleft^{(\text{int} \rightarrow \text{int}) \rightarrow \text{int}} L$, hence by Lemma V.10 and Theorem V.6, we obtain $w(L\Upsilon \Rightarrow \underline{0}) = \llbracket L\Upsilon \rrbracket_0 = \langle \llbracket L \rrbracket, \llbracket \Upsilon \rrbracket \rangle; \text{Eval}_0 \preceq w(L\Xi \Rightarrow \underline{0})$. This gives $\Upsilon \sqsubseteq \Xi$, the converse follows by Corollary V.9 and $\llbracket \Xi \rrbracket \preceq \llbracket \Upsilon \rrbracket$. \blacksquare

This is even a counterexample to equational full abstraction as we found two terms Ξ, Υ such that $\llbracket \Xi \rrbracket \neq \llbracket \Upsilon \rrbracket$ but $\Xi \equiv \Upsilon$.

Remark V.12. *Counterexample (1) can be rephrased without using scalar multiplication as soon as \mathcal{R} is such that $\infty = \sum_{n \in \mathbb{N}} \mathbf{1} + \dots + \mathbf{1}$ (n times). (This is the case for all semirings in Example II.7.) Indeed, under this hypothesis, the term Ψ has the same observational and denotational semantics of $\infty \underline{0}$.*

VI. APPLICATIONS

In this section we show how, choosing appropriate continuous semirings \mathcal{R} , it is possible to capture semantically several quantitative operational properties of programs.

We analyse PCF^{or} , the restriction of $\text{PCF}^\mathcal{R}$ obtained by forbidding the rule pM in the grammar of Definition IV.1, so

that the weight of any reduction sequence is 1. This has a natural translation into $\mathbf{PCF}^{\mathcal{R}}$, of course, since it is merely a restriction of that language. Here we shall see that other translations, obtained by *instrumenting* \mathbf{PCF}^{or} terms with elements of \mathcal{R} using the pM rule, allow us to refine the semantics to various quantitative purposes. Thus $\mathbf{PCF}^{\mathcal{R}}$ is used as a semantic metalanguage, capable of describing a range of different quantitative models of \mathbf{PCF}^{or} .

A. May/Must Non-Deterministic Convergence

The most basic behaviour to observe is whether a \mathbf{PCF}^{or} program (closed term of type int) M *may-converges* to a numeral \underline{n} , that is whether there exists a reduction sequence from M to \underline{n} . (For instance Ψ may-converges to $\underline{0}$, while Ω does not.) To observe such a behaviour it is enough to consider the simplest (non-trivial) continuous semiring, that is the Boolean semiring \mathcal{B} (Example II.7.1). Theorem V.6 specializes to the following characterization of may-convergence.

Corollary VI.1. *For every program M of \mathbf{PCF}^{or} , $\llbracket M \rrbracket_n^{\mathcal{B}} = \mathbf{t}$ if and only if M may-converges to \underline{n} .*

Note that $\mathcal{B}_I^{\text{II}}$ is isomorphic to the category \mathbf{MRel} , known as the *relational semantics*. Therefore, this first result is not very surprising as \mathbf{MRel} has been proved to characterize may-convergence for a resource sensitive extension of \mathbf{PCF}^{or} [18].

Starting from the standard semiring \mathcal{N} (Example II.7.1) we already get a much finer observation on programs. Indeed $w(M \Rightarrow \underline{n})$ becomes equal to the number of paths in $M \Rightarrow \underline{n}$. This means that $\mathcal{N}_I^{\text{II}}$ is able to compare programs depending on how many reduction sequences lead to a certain numeral.

Corollary VI.2. *For every program M of \mathbf{PCF}^{or} , $\llbracket M \rrbracket_n^{\mathcal{N}}$ is the number of reduction sequences from M to \underline{n} .*

For instance, we have $\llbracket \Psi \rrbracket_0^{\mathcal{N}} = \infty$ and $\llbracket \Phi(\underline{1} \text{ or } \underline{1}) \rrbracket_0^{\mathcal{N}} = 2$, so $\mathcal{N}_I^{\text{II}}$ separates the two terms, while $\mathcal{B}_I^{\text{II}}$ gives the same interpretation to both.

The characterization of *must-convergence* (i.e. the convergence to a numeral \underline{n} regardless of the erratic choices taken during the evaluation) requires a more complex translation of \mathbf{PCF}^{or} into $\mathbf{PCF}^{\mathcal{N}}$, allowing detection of potentially infinite reductions. For instance, the programs $\Phi \underline{1}$ or Ω and $\Phi \underline{1}$ have the same interpretation for any choice of \mathcal{R} (Example IV.13), but the first term is not must-convergent while the second is.

Let us consider the translation $(-)_\Gamma^\circ$ mapping judgments $\Gamma \vdash_{\mathbf{PCF}^{\text{or}}} - : A$ into judgments $\Gamma \vdash_{\mathbf{PCF}^{\mathcal{N}}} - : A$ which is generated by (assuming M of type $B \rightarrow B$ and L of type B , with $B = B_1 \rightarrow \dots \rightarrow B_k \rightarrow \text{int}$):

$$\begin{aligned} (\mathbf{Y}M)_\Gamma^\circ &:= \mathbf{Y}(\lambda x^B.((M)_\Gamma^\circ x \text{ or } \lambda y_1^{B_1} \dots \lambda y_k^{B_k}.\underline{0})), \\ (\lambda x^C.L)_\Gamma^\circ &:= \lambda x^C.((L)_{\Gamma;x:C}^\circ \text{ or } \lambda y_1^{B_1} \dots \lambda y_k^{B_k}.\underline{0}), \end{aligned}$$

where *generated by* means that $(-)_\Gamma^\circ$ commutes with all other constructors of \mathbf{PCF}^{or} . From now on we will consider \mathbf{PCF}^{or} programs, so the environment will be omitted.

Lemma VI.3. *For all programs M, P of \mathbf{PCF}^{or} , we have $M \rightarrow_\ell P$ if and only if one of the following conditions holds:*

- $\ell = \text{fix}$ and $M^\circ \xrightarrow{1}_{\text{fix}} \xrightarrow{1}_\beta \xrightarrow{1}_{\text{or}_1} P^\circ$,
- $\ell = \beta$ and $M^\circ \xrightarrow{1}_\beta \xrightarrow{1}_{\text{or}_1} P^\circ$,
- $\ell \notin \{\text{fix}, \beta\}$ and $M^\circ \xrightarrow{1}_\ell P^\circ$.

Lemma VI.4. *For every \mathbf{PCF}^{or} program M , there exists a reduction sequence from M° to \underline{n} , for some $n \in \mathbb{N}$.*

As a first corollary we obtain a characterization of strong convergence — a \mathbf{PCF}^{or} program M is *strongly converging* if there is no infinite reduction sequence starting from M .

Corollary VI.5. *A \mathbf{PCF}^{or} program M is strongly converging if and only if $\sum_{n \in \mathbb{N}} \llbracket M^\circ \rrbracket_n^{\mathcal{N}} < \infty$.*

For instance, $\Omega^\circ = \mathbf{Y}(\lambda x^{\text{int}}.((\lambda x^{\text{int}}.(x \text{ or } \underline{0}))x \text{ or } \underline{0}))$, and $\sum_{n \in \mathbb{N}} \llbracket \Omega^\circ \rrbracket_n^{\mathcal{N}} = \infty$ as $\llbracket \Omega^\circ \rrbracket_0^{\mathcal{N}} = \infty$.

Finally, from Corollaries VI.1 and VI.5, we obtain the following characterization of must-convergence.

Corollary VI.6. *A \mathbf{PCF}^{or} program M must-converges to a numeral \underline{n} if and only if $\sum_{k \in \mathbb{N}} \llbracket M^\circ \rrbracket_k^{\mathcal{N}} < \infty$, $\llbracket M \rrbracket_n^{\mathcal{N}} > 0$ and $\llbracket M \rrbracket_k^{\mathcal{N}} = 0$ for all $k \neq n$.*

B. Probabilistic Convergence

Let us now determine the probability that a \mathbf{PCF}^{or} program reduces to a numeral \underline{n} , supposing that the probability of applying or_1 or or_r when firing an or -redex is uniformly distributed. In the spirit of [14], this amounts to define its operational semantics through a Markov system having the terms as states, and the normal forms as absorbing states.

The Markov matrix describing such a process is given by:

$$\text{Red}_{M,P} := \begin{cases} 1 & \text{if } P = M \text{ is a normal form,} \\ 1 & \text{if } M \rightarrow_\ell P \text{ with } \ell \notin \{\text{or}_1, \text{or}_r\}, \\ 1 & \text{if } M \rightarrow_{\text{or}_1} P \text{ and } M \rightarrow_{\text{or}_r} P, \\ 0.5 & \text{if } M \rightarrow_{\text{or}_1} P \text{ but } M \not\rightarrow_{\text{or}_r} P \text{ or viceversa,} \\ 0 & \text{otherwise.} \end{cases}$$

Note that Red is a stochastic matrix (i.e. $\sum_P \text{Red}_{M,P} = 1$), and that $\text{Red}_{M,P}$ describes the probability of evolving from M to P in one ers. Similarly, the k -th fold matrix product Red^k , which is still a stochastic matrix, gives the evolution of the system after k steps. Since \underline{n} is absorbing, $\text{Red}_{M,\underline{n}}^k$ is monotone in k and bounded by 1, so $\text{Red}_{M,\underline{n}}^\infty := \sup_{k \in \mathbb{N}} \text{Red}_{M,\underline{n}}^k$ is well-defined and gives the probability that M reduces to \underline{n} in finitely many elementary reduction steps.

To capture this probabilistic feature in our semantic framework, consider the semiring \mathcal{P} (Example II.7.5) and the translation $(-)^\circ : \mathbf{PCF}^{\text{or}} \rightarrow \mathbf{PCF}^{\mathcal{P}}$ generated by:

$$(M \text{ or } P)^\circ := (0.5 M^\circ) \text{ or } (0.5 P^\circ).$$

Note that a reduction step $M \rightarrow_\ell P$ can be simulated by $M^\circ \xrightarrow{1}_\ell P^\circ$ when ℓ is not an or -rule, otherwise we need two steps $M^\circ \xrightarrow{1}_\ell \xrightarrow{0.5}_{\text{sca1}} P^\circ$.

Lemma VI.7. *For every program M of \mathbf{PCF}^{or} and $n \in \mathbb{N}$, we have $w(M^\circ \Rightarrow \underline{n}) = \text{Red}_{M,\underline{n}}^\infty$.*

As a corollary we get the following result, restating for $\mathcal{P}_!^\Pi$ the adequacy theorem proved in [14] for the category $\mathbf{PCoh}_!$ of probabilistic coherence spaces and entire functions.

Corollary VI.8. *For every program M of \mathbf{PCF}^{or} , $\llbracket M^\circ \rrbracket_n^{\mathcal{P}} = \text{Red}_{M, \underline{n}}^\infty$ which is the probability that M reduces to \underline{n} .*

For example, $\llbracket (\Phi \perp)^\circ \rrbracket^{\mathcal{P}} = \llbracket \Psi^\circ \rrbracket^{\mathcal{P}}$, both giving 1 on the web element 0. Notice also that, omitting the translation, $\llbracket \Phi \perp \rrbracket_0^{\mathcal{P}} = 1$ while $\llbracket \Psi \rrbracket_0^{\mathcal{P}} = \infty$.

The two models $\mathcal{P}_!^\Pi$ and $\mathbf{PCoh}_!$ share the same interpretations on probabilistic programs (i.e. on the image of the translation), since there is a faithful forgetful functor from $\mathbf{PCoh}_!$ to $\mathcal{P}_!^\Pi$ which acts like the identity on morphisms. These categories however differ in a crucial property, namely the fact that $\mathbf{PCoh}_!$ is well-pointed, while $\mathcal{P}_!^\Pi$ is not (the counterexample being given by the maps $\llbracket \Xi \rrbracket$ and $\llbracket \Upsilon \rrbracket$).

C. Resource Analysis.

We wish now to determine the minimum number of times that a β - or a fix -redex is contracted during an evaluation of a \mathbf{PCF}^{or} program M (*best case analysis*), or the maximum number (*worst case analysis*). These are indeed the two most critical redexes from the point of view of resource consumption, as their contraction may increase the size of M .

The model built from the tropical semiring \mathcal{T} (Example II.7.3) computes the best case analysis, through the translation $(-)^{\circ} : \mathbf{PCF}^{\text{or}} \rightarrow \mathbf{PCF}^{\mathcal{T}}$ generated by:

$$(\lambda x^A. M)^{\circ} := \lambda x^A. 1M^{\circ}, \quad (\mathbf{Y}M)^{\circ} := \mathbf{Y}(1M^{\circ}).$$

Recall that in \mathcal{T} the product is $+$ and $1 := 0$, so $1 \neq 1$.

Lemma VI.9. *For all \mathbf{PCF}^{or} terms M, P we have $M \rightarrow_{\ell} P$ if and only if either $\ell \in \{\beta, \text{fix}\}$ and $M^{\circ} \xrightarrow{0}_{\ell} \xrightarrow{1}_{\text{scal}} P^{\circ}$ or $\ell \notin \{\beta, \text{fix}\}$ and in that case $M^{\circ} \xrightarrow{0}_{\ell} P^{\circ}$.*

Therefore, given a reduction sequence $\pi \in M^{\circ} \Rightarrow \underline{n}$, its weight $w(\pi)$ gives the number of β - and fix -redexes contracted in π . Since the addition of \mathcal{T} is \min (with respect to the standard order on $\overline{\mathbb{N}}$), we have the following corollary.

Corollary VI.10. *For every program M of \mathbf{PCF}^{or} , $\llbracket M^\circ \rrbracket_n^{\mathcal{T}}$ is the minimum number of β - and fix -redexes reduced in a reduction sequence from M to \underline{n} .*

For the worst case analysis, consider the model built from the arctic semiring \mathcal{A} (Example II.7.4), where the addition is \max , and the translation $(-)^{\circ} : \mathbf{PCF}^{\text{or}} \rightarrow \mathbf{PCF}^{\mathcal{A}}$ is defined as before. An analogous reasoning gives the next corollary.

Corollary VI.11. *For every program M of \mathbf{PCF}^{or} , $\llbracket M^\circ \rrbracket_n^{\mathcal{A}}$ is the maximum number of β - and fix -redexes reduced in a reduction sequence from M to \underline{n} .*

For instance, we have $\llbracket (\Phi((\lambda x^{\text{int}}.x)\underline{0}))^\circ \rrbracket^{\mathcal{T}} > \llbracket (\text{succ } \Psi)^\circ \rrbracket^{\mathcal{T}}$, namely $\llbracket (\Phi((\lambda x^{\text{int}}.x)\underline{0}))^\circ \rrbracket_1^{\mathcal{T}} = 3$ and $\llbracket (\text{succ } \Psi)^\circ \rrbracket_1^{\mathcal{T}} = 2$, while $\llbracket (\Phi((\lambda x^{\text{int}}.x)\underline{0}))^\circ \rrbracket^{\mathcal{A}} < \llbracket (\text{succ } \Psi)^\circ \rrbracket^{\mathcal{A}}$, in fact $\llbracket (\Phi((\lambda x^{\text{int}}.x)\underline{0}))^\circ \rrbracket_1^{\mathcal{A}} = 3$ and $\llbracket (\text{succ } \Psi)^\circ \rrbracket_1^{\mathcal{A}} = \infty$.

Acknowledgements. Work partly supported by ANR Coquas 12JS0200601 and CNRS chaire “Logique linéaire et calcul”.

REFERENCES

- [1] R. Milner and C. Strachey, *A Theory of Programming Language Semantics*. Chapman and Hall, London, 1976.
- [2] G. D. Plotkin, “LCF considered as a programming language,” *Theor. Comput. Sci.*, vol. 5, no. 3, pp. 223–255, 1977.
- [3] R. Milner, “Fully abstract models of typed lambda-calculi,” *Theoretical Computer Science*, vol. 4, pp. 1–22, 1977.
- [4] A. M. Pitts, “Operationally-based theories of program equivalence,” in *Semantics and Logics of Computation*, P. Dybjer and A. M. Pitts, Eds. Cambridge University Press, 1997, pp. 241–298.
- [5] A. Ahmed, “Step-indexed syntactic logical relations for recursive and quantified types,” in *Proceedings of the 15th European conference on Programming Languages and Systems*, ser. ESOP’06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 69–83. [Online]. Available: http://dx.doi.org/10.1007/11693024_6
- [6] E. Moggi, “Notions of computation and monads,” *Information and Computation*, vol. 93, pp. 55–92, 1991.
- [7] M. Kwiatkowska, “On quantitative software verification,” in *Proc. 16th International SPIN Workshop*, ser. LNCS, C. Pasareanu, Ed., vol. 5578. Springer, 2009, pp. 2–3.
- [8] M. Kwiatkowska, G. Norman, and D. Parker, “PRISM 4.0: Verification of probabilistic real-time systems,” in *Proc. 23rd International Conference on Computer Aided Verification (CAV’11)*, ser. LNCS, G. Gopalakrishnan and S. Qadeer, Eds., vol. 6806. Springer, 2011, pp. 585–591.
- [9] K. Chatterjee, L. Doyen, and T. A. Henzinger, “Quantitative languages,” *ACM Trans. Comput. Logic*, vol. 11, no. 4, pp. 23:1–23:38, Jul. 2010. [Online]. Available: <http://doi.acm.org/10.1145/1805950.1805953>
- [10] D. Sands, “Operational theories of improvement in functional languages (extended abstract),” in *Functional Programming*, ser. Workshops in Computing, R. Heldal, C. K. Holst, and P. Wadler, Eds. Springer, 1991, pp. 298–311.
- [11] D. R. Ghica, “Slot games: a quantitative model of computation,” in *Proc. of the 32nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL’05)*, J. Palsberg and M. Abadi, Eds. ACM, 2005, pp. 85–97.
- [12] J.-Y. Girard, “Linear logic,” *Th. Comp. Sc.*, vol. 50, pp. 1–102, 1987.
- [13] —, “Normal functors, power series and lambda-calculus,” *Ann. Pure Appl. Logic*, vol. 37, no. 2, pp. 129–177, 1988.
- [14] V. Danos and T. Ehrhard, “Probabilistic coherence spaces as a model of higher-order probabilistic computation,” *Inf. Comput.*, vol. 209, no. 6, pp. 966–991, 2011.
- [15] T. Ehrhard, M. Pagani, and C. Tasson, “The Computational Meaning of Probabilistic Coherence Spaces,” in *Proceedings of the 26th Annual IEEE Symposium on Logic in Computer Science (LICS 2011)*, ser. IEEE Computer Society Press, M. Grohe, Ed., 2011, pp. 87–96.
- [16] T. Ehrhard, “Finiteness spaces,” *Math. Structures Comput. Sci.*, vol. 15, no. 4, pp. 615–646, 2005.
- [17] —, “On K othe sequence spaces and linear logic,” *MSCS*, vol. 12, pp. 579–623, 2002.
- [18] J. Laird, G. Manzonetto, and G. McCusker, “Constructing differential categories and deconstructing categories of games,” *Information and Computation*, vol. 222, no. C, pp. 247–264, 2013.
- [19] V. Danos and R. Harmer, “Probabilistic game semantics,” *ACM Transactions on Computational Logic*, vol. 3, no. 3, pp. 359–382, Jul. 2002.
- [20] Y. Lafont, “Logiques, cat egories et machines,” Ph.D. dissertation, Universit e Paris 7, 1988.
- [21] P.-A. Melli s, “Categorical semantics of linear logic,” *Panoramas et Synth eses*, vol. 27, 2009.
- [22] P.-A. Melli s, N. Tabareau, and C. Tasson, “An explicit formula for the free exponential modality of linear logic,” in *Int. Coll. Aut., Lang. and Prog. (ICALP’09)*, ser. LNCS, vol. 5556. Springer, 2009, pp. 247–260.
- [23] M. Droste and W. Kuich, “Semirings and formal power series,” in *Handbook of Weighted Automata*, M. Droste, W. Kuich, and H. Vogler, Eds. Springer-Verlag, 2009, ch. 1.
- [24] I. Guessarian, *Algebraic Semantics*, ser. Lecture Notes in Computer Science. Springer, 1981, vol. 99.
- [25] S. Mac Lane, *Categories for the Working Mathematician*. Berlin: Springer-Verlag, 1971.
- [26] H. Barendregt, *The Lambda-Calculus, its Syntax and Semantics*, ser. Stud. Log. F. Math., vol. 103. North-Holland, 1984.
- [27] R. Amadio and P.-L. Curien, *Domains and Lambda Calculi*, ser. Cambridge tracts in theoretical computer science. Cambridge University Press, Jul. 1998.