On Classical PCF, Linear Logic and the MIX Rule*

Shahin Amini[1] and Thomas Ehrhard[2]

1 **PPS, UMR 7126, Université Paris Diderot, Sorbonne Paris Cité**
F-75205 Paris, France
shahin.a.amini@gmail.com

2 **CNRS, PPS, UMR 7126, Université Paris Diderot, Sorbonne Paris Cité**
F-75205 Paris, France
thomas.ehrhard@pps.univ-paris-diderot.fr

Abstract

We study a classical version of PCF from a semantic point of view. We define a general notion of model based on categorical models of Linear Logic, in the spirit of earlier work by Girard, Regnier and Laurent. We give a concrete example based on the relational model of Linear Logic, that we present as a non-idempotent intersection type system, and we prove an Adequacy Theorem using ideas introduced by Krivine. Following Danos and Krivine, we also consider an extension of this language with a MIX construction introducing a form of must non-determinism; in this language, a program of type integer can have more than one value (or no value at all, raising an error). We propose a refinement of the relational model of classical PCF in which programs of type integer are single valued; this model rejects the MIX syntactical constructs (and the MIX rule of Linear Logic).

1998 ACM Subject Classification F.3.2 Semantics of Programming Languages, F.3.3 Studies of Program Constructs, F.4.1 Mathematical Logic

Keywords and phrases lambda-calculus, linear logic, classical logic, denotational semantics

Digital Object Identifier 10.4230/LIPIcs.CSL.2015.582

1 Introduction

Since the fundamental discovery by Timothy Griffin [7] that the call/cc primitive of the programming language *Scheme* can be typed by a non intuitionistic classical tautology (the Law of Peirce), much work has been dedicated to the study of the computational content of classical proofs. A particular attention has been devoted to the denotational semantics of classically extended lambda-calculi. Among the notions of categorical model defined for this purpose, we can isolate two main concepts.

- Models extending the standard categorical setting of CCC's for interpreting usual λ-calculi through a CPS translation. In these models of classical λ-calculi, terms are interpreted in a CCC of *negated objects*, that is objects of shape Σ^P where P is an object of a cartesian and cocartesian category $\mathcal{P}$ where Σ is a distinguished baseable object, see [15] and its generalization [14] where the category of negated objects is axiomatized. In this latter paper, it is also shown that Parigot's $\lambda\mu$-calculus [13] is the internal language of such categories and completeness of this notion of model for $\lambda\mu$ theories is proven in [9], enforcing further its universality.

* This work was partially supported by the French-Chinese ANR-NSFC project Locali.

24th EACSL Annual Conference on Computer Science Logic (CSL 2015).
Editor: Stephan Kreutzer; pp. 582–596
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

- Models based on Linear Logic (LL) and polarities. The basic idea of such models is to divide objects (formulas) in two categories, exchanged by linear negation: negative objects and positive ones: this is the basic idea of Girard's LC logical system. The main feature of these polarized objects is that each one carries its own structural morphisms (weakening and contraction). In [5], positive objects are correlation spaces (commutative $\otimes$-comonoids), and it is crucially used that all such comonoids are coalgebras for the $!\_$ functor, because, in the considered coherence space model, this exponential is the free commutative $\otimes$-comonoid functor. An obvious generalization, implicitly considered in [11], is to interpret directly positive formulas as !-coalgebras instead of $\otimes$-comonoids without making further assumptions on the ! comonad.

As recalled in Section 3.2, all models of the second class can be seen as models of the first one, but it is not clear that such a presentation is always particularly enlightening. We rather believe that, depending on the considered concrete model, one presentation might be more convenient than the other; for example, the classical PCF game model of [8] and the polarized HO game model of [10] are suitably described using the first notion. In the present paper, we focus on models for which the second presentation is more convenient.

To define the general interpretation of classical PCF, we assume therefore to be given a categorical model of classical LL $\mathcal{L}$ with a few additional features: an object N for natural numbers which is the coproduct of ω copies of 1 (the tensor unit) in $\mathcal{L}$ as well as a fix-point operator at each object, in the Kleisli category of !, which is a CCC.

Our classical version of PCF is based on the $\overline{\lambda}\mu$-calculus described in [1] which features three kinds of expressions: terms, stacks (or continuations) and commands which are pairings of terms and stacks. The operational semantics is given as a rewriting system on commands (it can easily be extended to terms and stacks but we do not do it for lack of space).

Stacks can be duplicated or erased during computations, hence types are interpreted as !-coalgebras and stacks as coalgebra morphisms. Notice that the interpretation of types corresponds to the linear negation of the usual PCF interpretation of types: roughly speaking, we interpret $\sigma \Rightarrow \tau$ as $!(P^{\perp}) \otimes Q$ where P and Q are the interpretations of σ and τ ($P^{\perp}$ is the linear negation of P). We retrieve the ordinary interpretation simply by taking the linear negation of this *positive translation*[1].

In particular, the positive interpretation of the ground type of natural numbers has to be such a coalgebra. Therefore, the most tempting choice, which would be to take $[\![\iota]\!] = \mathsf{N}^{\perp}$, is not possible (*Warning:* N is canonically a !-coalgebra, but $[\![\iota]\!] = \mathsf{N}^{\perp}$ is not!). So we simply set $[\![\iota]\!] = !(\mathsf{N}^{\perp})$, the free !-coalgebra generated by $\mathsf{N}^{\perp}$. We do not know if, depending on the concrete model under consideration, more "economical" choices of !-coalgebras would have been possible; this is certainly an interesting research direction. We describe the corresponding interpretation of expressions and state a general soundness theorem: this interpretation of commands is invariant under reduction (of course this could be extended to terms and stacks in a setting where the reduction would be extended to these expressions).

Then we consider the simplest example of this situation, where we take for $\mathcal{L}$ the category of sets and relations, which is a well known model of LL. We provide a description of the interpretation of expressions in this particular model by means of an intersection typing

[1] Due to the symmetries of a categorical model of LL, this is more an aesthetic choice of design than anything else. We could have preferred a negative interpretation, representing stacks as ?-algebras and using $⅋$ instead of $\otimes$ for interpreting contexts. The two interpretations would have been the same, up to linear transposition. The positive interpretation is in some sense closer to usual λ-calculus intuitions because, when interpreting expressions, the context remains on the argument side of morphisms.

system, in the spirit of [3, 16]. We then prove an adequacy theorem: if a command has a non-empty interpretation (that is, if it is typable in this intersection typing system) then its reduction terminates. The proof is based on a standard reducibility method (see [2] for instance). This model accommodates a natural extension of classical PCF by a parallel composition of commands corresponding to the MIX rule of LL as in [2].

In classical PCF with MIX, a normalizing command without free variables but with free names of ground type ι can yield an arbitrary amount of unrelated natural numbers on each of its free names (outputs). Without MIX syntactical constructs such a command will produce exactly one natural number on exactly one of its outputs. This can be checked syntactically, but we also build a simple refinement of the relational model of LL which does not accommodate the MIX rule and gives a direct semantic account of this uniqueness of values for classical PCF without MIX. This means that this crucial property will remain true in any extension of classical PCF that can be interpreted in this model.

2 Classical PCF

Types are given by the following BNF syntax: $\sigma := \iota \mid \sigma \Rightarrow \sigma \mid \sigma \times \sigma$.

The expressions of our language are those of the $\overline{\lambda}\mu$-calculus [1], extended with fix-points and primitives for dealing with integers. Let $x, y \dots$ be variables and $\alpha, \beta \dots$ be names. Terms t, commands c and stacks π are defined as follows (with $n \in \mathbb{N}$):

$$t := x \mid \underline{n} \mid \lambda x^\sigma\, t \mid \langle t, t\rangle \mid \mu\alpha^\sigma\, c \mid \mathsf{fix}\, x^\sigma\, t \qquad\qquad c := t * \pi$$
$$\pi := \alpha \mid \mathsf{arg}(t)\cdot\pi \mid \mathsf{pr}_1\cdot\pi \mid \mathsf{pr}_2\cdot\pi \mid \mathsf{succ}\cdot\pi \mid \mathsf{pred}\cdot\pi \mid \mathsf{if}(t,t)\cdot\pi$$

We give now the typing rules, which correspond to a sequent calculus. Γ's are typing variable contexts and Δ's are typing name contexts. We give rules for term typing judgments $\Gamma \vdash t : \sigma \mid \Delta$, stack typing judgments $\Gamma \mid \pi : \sigma \vdash \Delta$ and command typing judgments $\Gamma \vdash c \mid \Delta$.

$$\frac{}{\Gamma, x : \sigma \vdash x : \sigma \mid \Delta} \qquad \frac{}{\Gamma \mid \alpha : \sigma \vdash \alpha : \sigma, \Delta} \qquad \frac{\Gamma \vdash t : \sigma \mid \Delta \qquad \Gamma \mid \pi : \sigma \vdash \Delta}{\Gamma \vdash t * \pi \mid \Delta}$$

$$\frac{\Gamma \vdash s : \sigma \mid \Delta \qquad \Gamma \vdash t : \tau \mid \Delta}{\Gamma \vdash \langle s, t\rangle : \sigma \times \tau \mid \Delta} \qquad \frac{\Gamma \mid \pi : \sigma \vdash \Delta}{\Gamma \mid \mathsf{pr}_1\cdot\pi : \sigma \times \tau \vdash \Delta} \qquad \frac{\Gamma \mid \pi : \tau \vdash \Delta}{\Gamma \mid \mathsf{pr}_2\cdot\pi : \sigma \times \tau \vdash \Delta}$$

$$\frac{\Gamma, x : \sigma \vdash t : \tau \mid \Delta}{\Gamma \vdash \lambda x^\sigma\, t : \sigma \Rightarrow \tau \mid \Delta} \qquad \frac{\Gamma \vdash t : \sigma \mid \Delta \qquad \Gamma \mid \pi : \tau \vdash \Delta}{\Gamma \mid \mathsf{arg}(t)\cdot\pi : \sigma \Rightarrow \tau \vdash \Delta} \qquad \frac{\Gamma \vdash c \mid \alpha : \sigma, \Delta}{\Gamma \vdash \mu\alpha^\sigma\, c : \sigma \mid \Delta}$$

$$\frac{}{\Gamma \vdash \underline{n} : \iota \mid \Delta} \qquad \frac{\Gamma \mid \pi : \iota \vdash \Delta}{\Gamma \mid \mathsf{succ}\cdot\pi : \iota \vdash \Delta} \qquad \frac{\Gamma \mid \pi : \iota \vdash \Delta}{\Gamma \mid \mathsf{pred}\cdot\pi : \iota \vdash \Delta}$$

$$\frac{\Gamma \vdash t_1 : \sigma \mid \Delta \qquad \Gamma \vdash t_2 : \sigma \mid \Delta \qquad \Gamma \mid \pi : \sigma \vdash \Delta}{\Gamma \mid \mathsf{if}(t_1, t_2)\cdot\pi : \iota \vdash \Delta} \qquad \frac{\Gamma, x : \sigma \vdash t : \sigma \mid \Delta}{\Gamma \vdash \mathsf{fix}\, x^\sigma\, t : \sigma \mid \Delta}$$

We define a deterministic reduction relation $\to$ on processes.

$$(\lambda x^\sigma\, s) * \mathsf{arg}(t)\cdot\pi \to s\,[t/x] * \pi \qquad \langle s, t\rangle * \mathsf{pr}_1\cdot\pi \to s * \pi \qquad \langle s, t\rangle * \mathsf{pr}_2\cdot\pi \to t * \pi$$
$$(\mu\alpha^\sigma\, c) * \pi \to c\,[\pi/\alpha] \qquad (\mathsf{fix}\, x^\sigma\, t) * \pi \to t\,[\mathsf{fix}\, x^\sigma\, t/x] * \pi \qquad \underline{n} * \mathsf{succ}\cdot\pi \to \underline{n+1} * \pi$$
$$\underline{0} * \mathsf{pred}\cdot\pi \to \underline{0} * \pi \qquad \underline{n+1} * \mathsf{pred}\cdot\pi \to \underline{n} * \pi$$
$$\underline{0} * \mathsf{if}(t_1, t_2)\cdot\pi \to t_1 * \pi \qquad \underline{n+1} * \mathsf{if}(t_1, t_2)\cdot\pi \to t_2 * \pi$$

▶ **Proposition 1** (Subject Reduction). *Assume that* $\Gamma \vdash c \mid \Delta$ *and* $c \to c'$. *Then* $\Gamma \vdash c' \mid \Delta$.

The proof is a straightforward case analysis involving a Substitution Lemma.

A typical example of classical PCF program is the $\mathsf{call/cc}$ operator $t = \lambda f^{(\iota\Rightarrow\sigma)\Rightarrow\iota}\,\mu\alpha^{\iota}\,(f * \mathsf{arg}(\lambda x^{\iota}\,\mu\beta^{\sigma}\,(x * \alpha)) \cdot \alpha)$ which satisfies $\vdash t : ((\iota \Rightarrow \sigma) \Rightarrow \iota) \Rightarrow \iota \mid$ (its type is an instance of the well known Peirce classical tautology). When fed with an argument s such that $\vdash s : (\iota \Rightarrow \sigma) \Rightarrow \iota \mid$ (a functional), t tests whether this functional returns directly a value (and then t returns that value) or uses its argument (a function) by providing it with a natural number $\underline{n}$, and in that case t returns $\underline{n}$. This choice between two options is implemented by a contraction (the two occurrences of α).

The MIX extension. We consider also an extension of this classical version of PCF where we add two new constructs: a command err and, given commands c and d, a command $c\|d$. A similar extension of an untyped classical calculus has already been considered in [2]. It is more naturally introduced at the level of commands in the present $\overline{\lambda}\mu$ setting. These constructions obey the following typing rules

$$\frac{}{\Gamma \vdash \mathsf{err} \mid \Delta} \qquad \frac{\Gamma \vdash c \mid \Delta \qquad \Gamma \vdash d \mid \Delta}{\Gamma \vdash (c\|d) \mid \Delta}$$

We then extend the operational semantics of the calculus by adding the following reduction rules for commands.

$$\frac{}{\mathsf{err}\|c \to c} \qquad \frac{}{c\|\mathsf{err} \to c} \qquad \frac{c \to c'}{c\|d \to c'\|d} \qquad \frac{d \to d'}{c\|d \to c\|d'}$$

The resulting calculus on commands (with the other reduction rules given in Section 2) clearly satisfies the diamond property, the strongest form of confluence. These constructions can be extended as term constructions, available at all types. Simply set $\mathsf{err}^{\sigma} = \mu\alpha^{\sigma}\,\mathsf{err}$ and $(s\|t) = \mu\alpha^{\sigma}\,(s * \alpha\|t * \alpha)$. The term $s\|t$ is as a parallel composition of s and t enriching the language with a form of *must non-determinism*. It allows *eg.* to write $\underline{3}\|\underline{7}$, a closed term of type ι, whose value is *at the same time* 3 and 7.

Almost closed commands. We come back to our initial version of classical PCF, without the MIX constructs. A name context $\Delta = (\alpha_1 : \tau_1, \dots, \alpha_k : \tau_k)$ is *ground* if $\tau_j = \iota$ for each j. We say that a command c is *almost closed* if $\vdash c \mid \Delta$ for some ground Δ. An almost closed command is very similar to a closed term of type ι in ordinary PCF. The difference is twofold: first an almost closed command can have more than one output (one for each name in the name context), and second its outputs are named, simply to make them usable.

▶ **Proposition 2.** *Let c be an almost closed and normal command. Then $c = \underline{n} * \alpha$ for some $n \in \mathbb{N}$ and name α.*

The proof is a simple case analysis.

So, consider an almost closed command c such that, say, $\vdash c \mid \alpha_1 : \iota, \dots, \alpha_k : \iota$. Then either the $\to$ reduction of c does not terminate, or it ends with a normal almost closed command, which must be of shape $\underline{n} * \alpha_i$ for uniquely determined $i \in \{1, \dots, k\}$ and $n \in \mathbb{N}$: the reduction of c computes the value $\underline{n}$ and chooses the output on which it is issued.

The notion of almost closed command still makes sense in classical PCF with MIX. The difference is that normal forms are now MIX compositions of elementary command $\underline{n} * \alpha_i$. One can obtain for instance $(\underline{0} * \alpha_1)\|((\underline{3} * \alpha_2)\|(\underline{7} * \alpha_1))$ whose effect is to produce the value $\underline{3}$ on output α_2, values $\underline{0}$ and $\underline{7}$ on output α_1 and nothing on the other outputs.

3 Linear logic based denotational semantics

The kind of denotational models in which we are interested in this paper are those induced by a model of LL, in the spirit of Girard's seminal work [5] further developed *eg.*. in [11]. We first recall the general definition of a model of LL implicit in [4], our main reference here is [12] to which we also refer for the rich bibliography on this general topic. A model of LL consists of:

- A category $\mathcal{L}$.
- A symmetric monoidal closed structure $(\otimes, 1, \lambda, \rho, \alpha, \sigma)$: $\otimes$ is a functor $\mathcal{L}^2 \to \mathcal{L}$, 1 an object of $\mathcal{L}$, $\lambda_X \in \mathcal{L}(1 \otimes X, X)$, $\rho_X \in \mathcal{L}(X \otimes 1, X)$, $\alpha_{X,Y,Z} \in \mathcal{L}((X \otimes Y) \otimes Z, X \otimes (Y \otimes Z))$ and $\sigma_{X,Y} \in \mathcal{L}(X \otimes Y, Y \otimes X)$ are natural isos satisfying coherence diagrams that we do not recall here. We use $X \multimap Y$ for the object of linear morphisms from X to Y, ev for the evaluation morphism which belongs to $\mathcal{L}((X \multimap Y) \otimes X, Y)$ and cur for the map $\mathcal{L}(Z \otimes X, Y) \to \mathcal{L}(Z, X \multimap Y)$.
- An object $\bot$ of $\mathcal{L}$ such that $\eta_X = \mathsf{cur}(\mathsf{ev}\, \sigma_{X \multimap \bot, X}) \in \mathcal{L}(X, (X \multimap \bot) \multimap \bot)$ be an iso for each object X (one says that $\mathcal{L}$ is a $*$-autonomous category); we use $X^\perp$ for $X \multimap \bot$.
- The category $\mathcal{L}$ is assumed to be cartesian. We use $\top$ for the terminal object, & for the cartesian product and pr^i for the projections. It follows by $*$-autonomy that $\mathcal{L}$ has also all finite coproducts.
- We are also given a comonad $!\_ : \mathcal{L} \to \mathcal{L}$ with counit $\mathsf{der}_X \in \mathcal{L}(!X, X)$ (called dereliction) and comultiplication $\mathsf{dig}_X \in \mathcal{L}(!X, !!X)$ (called digging).
- And a strong symmetric monoidal structure for the functor $!\_$, from the symmetric monoidal category $(\mathcal{L}, \&)$ to the symmetric monoidal category $(\mathcal{L}, \otimes)$. This means that we are given an iso $\mathsf{m}^{(0)} \in \mathcal{L}(1, !\top)$ and a natural iso $\mathsf{m}^{(2)}_{X,Y} \in \mathcal{L}(!X \otimes !Y, !(X \mathbin{\&} Y))$ which satisfy a series of commutations that we do not recall here (they are often called *Seely isos*). We also require a coherence condition relating $\mathsf{m}^{(2)}$ and dig.

It follows that we can define a lax symmetric monoidal structure for the functor $!\_$ from the symmetric monoidal category $(\mathcal{L}, \otimes)$ to itself, that is a natural morphism $\mu^{(n)}_{X_1,\dots,X_N} \in \mathcal{L}(!X_1 \otimes \cdots \otimes !X_n, !(X_1 \otimes \cdots \otimes X_n))$ satisfying some coherence conditions.

We use $?\_$ for the "De Morgan dual" of $!\_$: $?X = (!(X^\perp))^\perp$ and similarly for morphisms. It is a monad on $\mathcal{L}$ with unit der'_X and multiplication dig'_X defined straightforwardly, using der_Y and dig_Y.

The Eilenberg-Moore category. It is then standard to define the category $\mathcal{L}^!$ of !-coalgebras. An object of this category is a pair $P = (\underline{P}, \mathsf{h}_P)$ where $\underline{P} \in \mathsf{Obj}(\mathcal{L})$ and $\mathsf{h}_P \in \mathcal{L}(\underline{P}, !\underline{P})$ is such that $\mathsf{der}_{\underline{P}}\, \mathsf{h}_P = \mathsf{Id}$ and $\mathsf{dig}_{\underline{P}}\, \mathsf{h}_P = !\mathsf{h}_P\, \mathsf{h}_P$.

Given two such coalgebras P and Q, an element of $\mathcal{L}^!(P, Q)$ is an $f \in \mathcal{L}(\underline{P}, \underline{Q})$ such that $\mathsf{h}_Q\, f = !f\, \mathsf{h}_P$. Identities and composition are defined in the obvious way. The functor $!\_$ can then be seen as a functor from $\mathcal{L}$ to $\mathcal{L}^!$: this functor maps X to the coalgebra $(!X, \mathsf{dig}_X)$ and a morphism $f \in \mathcal{L}(X, Y)$ to the coalgebra morphism $!f \in \mathcal{L}^!((!X, \mathsf{dig}_X), (!Y, \mathsf{dig}_Y))$. It is right adjoint to the forgetful functor $\mathsf{U} : \mathcal{L}^! \to \mathcal{L}$ which maps a !-coalgebra P to $\underline{P}$ and a morphism f to itself. Given $f \in \mathcal{L}(\underline{P}, X)$ (where X is an object of $\mathcal{L}$ and P an object of $\mathcal{L}^!$), we use $f^!$ for the corresponding element of $\mathcal{L}^!(P, !X)$, called *generalized promotion* of f.

The object 1 of $\mathcal{L}$ induces an object of $\mathcal{L}^!$, still denoted as 1, namely $(1, \mu^{(0)})$.

Given two objects P and Q of $\mathcal{L}^!$, we can define an object $P \otimes Q$ of $\mathcal{L}^!$ setting $\underline{P \otimes Q} = \underline{P} \otimes \underline{Q}$ and $\mathsf{h}_{P \otimes Q} = \mu^{(2)}_{\underline{P},\underline{Q}}\, (\mathsf{h}_P \otimes \mathsf{h}_Q)$.

Any object P of $\mathcal{L}^!$ can be equipped with a canonical structure of commutative comonoid. This means that we can define a morphism $\mathsf{w}_P \in \mathcal{L}^!(P, 1)$ and a morphism $\mathsf{c}_P \in \mathcal{L}^!(P, P \otimes P)$

$$\begin{tikzcd}
\underline{P} \arrow[r, "\mathsf{c}_P"] \arrow[dr, "{\lambda_{\underline{P}}}^{-1}"'] & \underline{P} \otimes \underline{P} \arrow[d, "\mathsf{w}_P \otimes \underline{P}"] \\
& 1 \otimes \underline{P}
\end{tikzcd}
\qquad
\begin{tikzcd}
\underline{P} \arrow[r, "\mathsf{c}_P"] \arrow[d, "\mathsf{c}_P"'] & \underline{P} \otimes \underline{P} \arrow[r, "\mathsf{c}_P \otimes \underline{P}"] & (\underline{P} \otimes \underline{P}) \otimes \underline{P} \arrow[d, "\alpha_{\underline{P},\underline{P},\underline{P}}"] \\
\underline{P} \otimes \underline{P} \arrow[rr, "\underline{P} \otimes \mathsf{c}_P"'] & & \underline{P} \otimes (\underline{P} \otimes \underline{P})
\end{tikzcd}
\qquad
\begin{tikzcd}
\underline{P} \arrow[r, "\mathsf{c}_P"] \arrow[dr, "\mathsf{c}_P"'] & \underline{P} \otimes \underline{P} \arrow[d, "\sigma_{\underline{P},\underline{P}}"] \\
& \underline{P} \otimes \underline{P}
\end{tikzcd}$$

Figure 1 Comonoid properties of a coalgebra.

$$\begin{tikzcd}
1 \otimes !X \otimes !X \arrow[r, "1 \otimes \mathsf{der}_X \otimes \mathsf{w}_{!X}"] \arrow[d, "\overline{0} \otimes !X \otimes !X"'] & 1 \otimes X \otimes 1 \arrow[d, "\varphi"] \\
\mathsf{N} \otimes !X \otimes !X \arrow[r, "\overline{\mathsf{if}}"'] & X
\end{tikzcd}
\qquad
\begin{tikzcd}
1 \otimes !X \otimes !X \arrow[r, "1 \otimes \mathsf{w}_{!X} \otimes \mathsf{der}_X"] \arrow[d, "\overline{n+1} \otimes !X \otimes !X"'] & 1 \otimes 1 \otimes X \arrow[d, "\psi"] \\
\mathsf{N} \otimes !X \otimes !X \arrow[r, "\overline{\mathsf{if}}"'] & X
\end{tikzcd}$$

Figure 2 Categorical properties of the conditional.

which satisfy the commutations of Figure 1.

One can check a stronger property, namely that 1 is the terminal object of $\mathcal{L}^!$ and that $P \otimes Q$ (equipped with projections defined in the obvious way using w_Q and w_P) is the cartesian product of P and Q in $\mathcal{L}^!$; the proof consists of rather long computations, see [12].

It is also important to notice that, if the family $(P_i)_{i\in I}$ of objects of $\mathcal{L}^!$ is such that the family $(\underline{P_i})_{i\in I}$ admits a coproduct $(\bigoplus_{i\in I} \underline{P_i}, (\mathsf{in}^i)_{i\in I})$ in $\mathcal{L}$, then it admits a coproduct in $\mathcal{L}^!$. This coproduct $P = \bigoplus_{i\in I} P_i$ is defined as $\underline{P} = \bigoplus_{i\in I} \underline{P_i}$, with a structure map h_P defined by the fact that, for each $i \in I$, $\mathsf{h}_P\,\mathsf{in}^i = !\mathsf{in}^i\,\mathsf{h}_{P_i}$.

Object of natural numbers and conditional. We assume also that in $\mathcal{L}$, the family of objects $(X_n)_{n\in\mathbb{N}}$ such that $X_n = 1$ for each n, has a coproduct N. For each $n \in \mathbb{N}$, we use $\overline{n}$ for the nth injection $\overline{n} \in \mathcal{L}(1, \mathsf{N})$. Using the obvious iso between N and $1 \oplus \mathsf{N}$, we define two morphisms $\overline{\mathsf{succ}}, \overline{\mathsf{pred}} \in \mathcal{L}(\mathsf{N}, \mathsf{N})$ such that $\overline{\mathsf{succ}}\,\overline{n} = \overline{n+1}$, $\overline{\mathsf{pred}}\,\overline{0} = \overline{0}$ and $\overline{\mathsf{pred}}\,\overline{n+1} = \overline{n}$. Let X be an object of $\mathcal{L}$. Let $\overline{\mathsf{if}}_0 \in \mathcal{L}(1 \otimes !X \otimes !X, X)$ be defined as the following composition in $\mathcal{L}$: $1 \otimes !X \otimes !X \xrightarrow{1 \otimes \mathsf{der}_X \otimes \mathsf{w}_{!X}} 1 \otimes X \otimes 1 \xrightarrow{\varphi} X$, where φ is the obvious iso. Let $\overline{\mathsf{if}}_+ \in \mathcal{L}(\mathsf{N} \otimes !X \otimes !X, X)$ be defined as the following composition of morphisms in $\mathcal{L}$: $\mathsf{N} \otimes !X \otimes !X \xrightarrow{\mathsf{w}_\mathsf{N} \otimes \mathsf{w}_{!X} \otimes \mathsf{der}_X} 1 \otimes 1 \otimes X \xrightarrow{\psi} X$, where ψ is the obvious iso. Observe that we use the fact that N has a canonical structure of !-coalgebra (as a sum of coalgebras) inducing the weakening morphism w_N. It is the only place where this property is used. Using these two morphisms, the iso between N and $1 \oplus \mathsf{N}$ and the fact that $\otimes$ commutes with sums (because it is a left adjoint), we define a morphism $\overline{\mathsf{if}} \in \mathcal{L}(\mathsf{N} \otimes !X \otimes !X, X)$ such that the two diagrams of Figure 2 commute.

Fix-point operators. For any object X, we assume to be given a morphism $\overline{\mathsf{fix}}_X \in \mathcal{L}(!(!X \multimap X), X)$ such that the following diagram commutes in $\mathcal{L}$:

$$\begin{tikzcd}
!(!X \multimap X) \arrow[r, "\mathsf{c}_{!X}"] \arrow[dr, "\overline{\mathsf{fix}}_X"'] & !(!X \multimap X) \otimes !(!X \multimap X) \arrow[r, "\mathsf{der}_{!X \multimap X} \otimes \overline{\mathsf{fix}}^!_X"] & (!X \multimap X) \otimes !X \arrow[dl, "\mathsf{ev}"] \\
& X &
\end{tikzcd}$$

MIX in Linear Logic The categorical setting introduced so far allows to interpret the MIX-free version of classical PCF. In order to interpret the MIX extension of Section 2, it suffices to assume that $\perp$ is equipped with a structure of commutative $\otimes$-monoid (this is an additional structure of the model). If we think of $\perp$ as an object of scalars, which is a natural intuition since $\perp$ is the dualizing object, this means that these scalars have a multiplication, a natural intuition again if we have linear algebra in mind. We use $\mathsf{mix}^0 \in \mathcal{L}(1,\perp)$ for the unit of this monoid and $\mathsf{mix}^2 \in \mathcal{L}(\perp\otimes\perp,\perp)$ for its multiplication. When this structure is added, we say that $\mathcal{L}$ is a model of LL with MIX.

3.1 Interpreting classical PCF

The semantics of a type σ is an object $\llbracket\sigma\rrbracket$ of $\mathcal{L}^!$. We set $\llbracket\iota\rrbracket = !(\mathsf{N}^\perp)$, $\llbracket\sigma\Rightarrow\tau\rrbracket = !(\underline{\llbracket\sigma\rrbracket}^\perp)\otimes\llbracket\tau\rrbracket$ and $\llbracket\sigma\times\tau\rrbracket = \llbracket\sigma\rrbracket\oplus\llbracket\tau\rrbracket$. So $\underline{\llbracket\iota\rrbracket}^\perp = ?\mathsf{N}$, which will be the target object for the interpretation of terms of type ι. Let $\Gamma = (x_1:\sigma_1,\dots,x_n:\sigma_n)$ be a variable context and $\Delta = (\alpha_1:\tau_1,\dots,\alpha_k;\tau_k)$ be a name context, then we define two objects of $\mathcal{L}^!$ by $\llbracket\Gamma\rrbracket = !(\underline{\llbracket\sigma_1\rrbracket}^\perp)\otimes\cdots\otimes !(\underline{\llbracket\sigma_n\rrbracket}^\perp)$ and $\llbracket\Delta\rrbracket = \llbracket\tau_1\rrbracket\otimes\cdots\otimes\llbracket\tau_k\rrbracket$. With any term t such that $\Gamma\vdash t:\sigma\mid\Delta$, we associate $\llbracket t\rrbracket_{\Gamma,\Delta}\in\mathcal{L}(\underline{\llbracket\Gamma\rrbracket}\otimes\underline{\llbracket\Delta\rrbracket},\underline{\llbracket\sigma\rrbracket}^\perp)$, with any command c such that $\Gamma\vdash c\mid\Delta$ we associate $\llbracket c\rrbracket_{\Gamma,\Delta}\in\mathcal{L}(\underline{\llbracket\Gamma\rrbracket}\otimes\underline{\llbracket\Delta\rrbracket},\perp)$ and with any stack π such that $\Gamma\mid\pi:\sigma\vdash\Delta$ we associate $\llbracket\pi\rrbracket_{\Gamma,\Delta}\in\mathcal{L}^!(\llbracket\Gamma\rrbracket\otimes\llbracket\Delta\rrbracket,\llbracket\sigma\rrbracket)$. This latter has to be a coalgebra morphism because stacks must be duplicable and discardable (think of the reduction rule $(\mu\alpha\, c)*\pi\to c\,[\pi/\alpha]$).

We give now the interpretation of expressions, starting with terms. In these definitions, the symbol φ stands for an iso which can be deduced from the context. The interpretation of a variable $\llbracket x\rrbracket_{\Gamma,x:\sigma,\Delta}$ is defined as the following composition of morphisms:

$$\underline{\llbracket\Gamma\rrbracket}\otimes !(\underline{\llbracket\sigma\rrbracket}^\perp)\otimes\underline{\llbracket\Delta\rrbracket}\xrightarrow{\mathsf{w}_{\llbracket\Gamma\rrbracket}\otimes\mathsf{der}_{\underline{\llbracket\sigma\rrbracket}^\perp}\otimes\mathsf{w}_{\llbracket\Delta\rrbracket}} 1\otimes\underline{\llbracket\sigma\rrbracket}^\perp\otimes 1\xrightarrow{\varphi}\underline{\llbracket\sigma\rrbracket}^\perp$$

Let $n\in\mathbb{N}$, remember that $\overline{n}\in\mathcal{L}(1,\mathsf{N})$ so that $?\overline{n}\in\mathcal{L}(?1,?\mathsf{N})$. We define $\llbracket\underline{n}\rrbracket_{\Gamma,\Delta}$ as the following composition of morphisms in $\mathcal{L}$:

$$\underline{\llbracket\Gamma\rrbracket}\otimes\underline{\llbracket\Delta\rrbracket}\xrightarrow{\varphi\,(\mathsf{w}_{\llbracket\Gamma\rrbracket}\otimes\mathsf{w}_{\llbracket\Delta\rrbracket})} 1\xrightarrow{\mathsf{d}'_1} ?1\xrightarrow{?\overline{n}} ?\mathsf{N}$$

Assume next that $\Gamma,x:\sigma\vdash t:\tau\mid\Delta$ so that we have $\llbracket t\rrbracket_{\Gamma,x:\sigma,\Delta}\,\varphi\in\mathcal{L}(\underline{\llbracket\Gamma\rrbracket}\otimes\underline{\llbracket\Delta\rrbracket}\otimes !(\underline{\llbracket\sigma\rrbracket}^\perp),\underline{\llbracket\tau\rrbracket}^\perp)$. We set $\llbracket\lambda x^\sigma\, t\rrbracket_{\Gamma,\Delta} = \mathsf{cur}(\llbracket t\rrbracket_{\Gamma,x:\sigma,\Delta}\,\varphi)\in\mathcal{L}(\underline{\llbracket\Gamma\rrbracket}\otimes\underline{\llbracket\Delta\rrbracket}, !(\underline{\llbracket\sigma\rrbracket}^\perp)\multimap\underline{\llbracket\tau\rrbracket}^\perp)$ and we have $!(\underline{\llbracket\sigma\rrbracket}^\perp)\multimap\underline{\llbracket\tau\rrbracket}^\perp = (!(\underline{\llbracket\sigma\rrbracket}^\perp)\otimes\underline{\llbracket\tau\rrbracket})^\perp = \underline{\llbracket\sigma\Rightarrow\tau\rrbracket}^\perp$ up to canonical isos.

Assume that $\Gamma\vdash s:\sigma\mid\Delta$ and $\Gamma\vdash t:\tau\mid\Delta$ so that we have $\llbracket s\rrbracket_{\Gamma,\Delta}\in\mathcal{L}(\underline{\llbracket\Gamma\rrbracket}\otimes\underline{\llbracket\Delta\rrbracket},\underline{\llbracket\sigma\rrbracket}^\perp)$ and $\llbracket t\rrbracket_{\Gamma,\Delta}\in\mathcal{L}(\underline{\llbracket\Gamma\rrbracket}\otimes\underline{\llbracket\Delta\rrbracket},\underline{\llbracket\tau\rrbracket}^\perp)$. So we set $\llbracket\langle s,t\rangle\rrbracket_{\Gamma,\Delta} = \langle\llbracket s\rrbracket_{\Gamma,\Delta},\llbracket t\rrbracket_{\Gamma,\Delta}\rangle\in\mathcal{L}(\underline{\llbracket\Gamma\rrbracket}\otimes\underline{\llbracket\Delta\rrbracket},\underline{\llbracket\sigma\rrbracket}^\perp\ \&\ \underline{\llbracket\tau\rrbracket}^\perp)$ which has the prescribed codomain since $\underline{\llbracket\sigma\rrbracket}^\perp\ \&\ \underline{\llbracket\tau\rrbracket}^\perp = \underline{\llbracket\sigma\times\tau\rrbracket}^\perp$.

Assume that $\Gamma\vdash c\mid\alpha:\sigma,\Delta$ so that we have $\llbracket c\rrbracket_{\Gamma,\alpha:\sigma,\Delta}\,\varphi\in\mathcal{L}(\underline{\llbracket\Gamma\rrbracket}\otimes\underline{\llbracket\Delta\rrbracket}\otimes\underline{\llbracket\sigma\rrbracket},\perp)$. Then we set $\llbracket\mu\alpha^\sigma\, c\rrbracket_{\Gamma,\Delta} = \mathsf{cur}(\llbracket c\rrbracket_{\Gamma,\alpha:\sigma,\Delta}\,\varphi)\in\mathcal{L}(\underline{\llbracket\Gamma\rrbracket}\otimes\underline{\llbracket\Delta\rrbracket},\underline{\llbracket\sigma\rrbracket}^\perp)$.

Assume that $\Gamma,x:\sigma\vdash t:\sigma\mid\Delta$ so that we have $\llbracket t\rrbracket_{\Gamma,x:\sigma,\Delta}\,\varphi\in\mathcal{L}(\underline{\llbracket\Gamma\rrbracket}\otimes\underline{\llbracket\Delta\rrbracket}\otimes !(\underline{\llbracket\sigma\rrbracket}^\perp),\underline{\llbracket\sigma\rrbracket}^\perp)$. We set $\llbracket\mathsf{fix}\,x^\sigma\, t\rrbracket_{\Gamma,\Delta} = \overline{\mathsf{fix}}_{\underline{\llbracket\sigma\rrbracket}^\perp}\,\mathsf{cur}\llbracket t\rrbracket_{\Gamma,x:\sigma,\Delta}\,\varphi\in\mathcal{L}(\underline{\llbracket\Gamma\rrbracket}\otimes\underline{\llbracket\Delta\rrbracket},\underline{\llbracket\sigma\rrbracket}^\perp)$.

Concerning commands, assume that $\Gamma\vdash t:\sigma\mid\Delta$ and that $\Gamma\mid\pi:\sigma\vdash\Delta$ so that we have $\llbracket t\rrbracket_{\Gamma,\Delta}\in\mathcal{L}(\underline{\llbracket\Gamma\rrbracket}\otimes\underline{\llbracket\Delta\rrbracket},\underline{\llbracket\sigma\rrbracket}^\perp)$ and $\llbracket\pi\rrbracket_{\Gamma,\Delta}\in\mathcal{L}^!(\llbracket\Gamma\rrbracket\otimes\llbracket\Delta\rrbracket,\llbracket\sigma\rrbracket)$ and therefore $\llbracket\pi\rrbracket_{\Gamma,\Delta}\in\mathcal{L}(\underline{\llbracket\Gamma\rrbracket}\otimes\underline{\llbracket\Delta\rrbracket},\underline{\llbracket\sigma\rrbracket})$. We define $\llbracket t*\pi\rrbracket_{\Gamma,\Delta}$ as the following composition of morphisms in $\mathcal{L}$

$$\underline{\llbracket\Gamma\rrbracket}\otimes\underline{\llbracket\Delta\rrbracket}\xrightarrow{\mathsf{c}_{\llbracket\Gamma\rrbracket\otimes\llbracket\Delta\rrbracket}}\underline{\llbracket\Gamma\rrbracket}\otimes\underline{\llbracket\Delta\rrbracket}\otimes\underline{\llbracket\Gamma\rrbracket}\otimes\underline{\llbracket\Delta\rrbracket}\xrightarrow{\llbracket t\rrbracket\otimes\llbracket\pi\rrbracket}\underline{\llbracket\sigma\rrbracket}^\perp\otimes\underline{\llbracket\sigma\rrbracket}\xrightarrow{\mathsf{ev}}\perp$$

Let us come now to stacks. The morphism $\llbracket\alpha\rrbracket_{\Gamma,\alpha:\sigma,\Delta}$ is defined as the following composition of morphisms in $\mathcal{L}^!$

$$\llbracket\Gamma\rrbracket\otimes\llbracket\sigma\rrbracket\otimes\llbracket\Delta\rrbracket\xrightarrow{\mathsf{w}_{\llbracket\Gamma\rrbracket}\otimes\llbracket\sigma\rrbracket\otimes\mathsf{w}_{\llbracket\Delta\rrbracket}} 1\otimes\llbracket\sigma\rrbracket\otimes 1\xrightarrow{\varphi}\llbracket\sigma\rrbracket$$

Remember that we have defined $\overline{\mathsf{succ}}, \overline{\mathsf{pred}} \in \mathcal{L}(\mathsf{N}, \mathsf{N})$, so that we have $!(\overline{\mathsf{succ}}^{\perp}), !(\overline{\mathsf{pred}}^{\perp}) \in \mathcal{L}^{!}(!(\underline{\mathsf{N}}^{\perp}), !(\underline{\mathsf{N}}^{\perp}))$. Assume that $\Gamma \mid \pi : \iota \vdash \Delta$ so that $\llbracket \pi \rrbracket_{\Gamma,\Delta} \in \mathcal{L}^{!}(\llbracket \Gamma \rrbracket \otimes \llbracket \Delta \rrbracket, \llbracket \iota \rrbracket)$, and we set $\llbracket \mathsf{succ} \cdot \pi \rrbracket_{\Gamma,\Delta} = !(\overline{\mathsf{succ}}^{\perp})\, \llbracket \pi \rrbracket_{\Gamma,\Delta}$ and $\llbracket \mathsf{pred} \cdot \pi \rrbracket_{\Gamma,\Delta} = !(\overline{\mathsf{pred}}^{\perp})\, \llbracket \pi \rrbracket_{\Gamma,\Delta}$; both morphisms belong to $\mathcal{L}^{!}(\llbracket \Gamma \rrbracket \otimes \llbracket \Delta \rrbracket, \llbracket \iota \rrbracket)$.

Remember also that, for any object X of $\mathcal{L}$, we have defined $\overline{\mathsf{if}} \in \mathcal{L}(\mathsf{N} \otimes !X \otimes !X, X)$. Using $*$-autonomy and isos induced by the monoidal structure of $\mathcal{L}$, we can canonically turn this morphism into $\overline{\mathsf{if}}' \in \mathcal{L}(X^{\perp} \otimes !X \otimes !X, \mathsf{N}^{\perp})$. Assume that $X = \underline{P}^{\perp}$ where P is an object of $\mathcal{L}^{!}$. Then we can set $\overline{\mathsf{If}} = \overline{\mathsf{if}}'^{!} \in \mathcal{L}^{!}(P \otimes !(\underline{P}^{\perp}) \otimes !(\underline{P}^{\perp}), !(\mathsf{N}^{\perp}))$. Assume that $\Gamma \mid \pi : \sigma \vdash \Delta$ and $\Gamma \vdash t_i : \sigma \mid \Delta$ for $i = 1, 2$. Then we have $\llbracket \pi \rrbracket_{\Gamma,\Delta} \in \mathcal{L}^{!}(\llbracket \Gamma \rrbracket \otimes \llbracket \Delta \rrbracket, \llbracket \sigma \rrbracket)$ and $\llbracket t_1 \rrbracket^{!}_{\Gamma,\Delta}, \llbracket t_2 \rrbracket^{!}_{\Gamma,\Delta} \in \mathcal{L}^{!}(\llbracket \Gamma \rrbracket \otimes \llbracket \Delta \rrbracket, !\underline{\llbracket \sigma \rrbracket}^{\perp})$, and we define $\llbracket \mathsf{if}(t_1, t_2, \pi) \rrbracket_{\Gamma,\Delta}$ as the following composition of morphisms in $\mathcal{L}^{!}$, using a ternary version of the contraction morphism

$$\llbracket \Gamma \rrbracket \otimes \llbracket \Delta \rrbracket \xrightarrow{\mathsf{c}^{(3)}_{\llbracket \Gamma \rrbracket \otimes \llbracket \Delta \rrbracket}} (\llbracket \Gamma \rrbracket \otimes \llbracket \Delta \rrbracket)^{\otimes 3} \xrightarrow{\llbracket \pi \rrbracket \otimes \llbracket t_1 \rrbracket^{!} \otimes \llbracket t_2 \rrbracket^{!}} \llbracket \sigma \rrbracket \otimes !(\underline{\llbracket \sigma \rrbracket}^{\perp}) \otimes !(\underline{\llbracket \sigma \rrbracket}^{\perp}) \xrightarrow{\overline{\mathsf{If}}} !(\mathsf{N}^{\perp})$$

Assume that $\Gamma \vdash \pi : \tau \mid \Delta$ and that $\Gamma \vdash t : \sigma \mid \Delta$ so that $\llbracket \pi \rrbracket_{\Gamma,\Delta} \in \mathcal{L}^{!}(\llbracket \Gamma \rrbracket \otimes \llbracket \Delta \rrbracket, \llbracket \tau \rrbracket)$ and $\llbracket t \rrbracket^{!}_{\Gamma,\Delta} \in \mathcal{L}^{!}(\llbracket \Gamma \rrbracket \otimes \llbracket \Delta \rrbracket, !(\llbracket \sigma \rrbracket^{\perp}))$, we set $\llbracket \mathsf{arg}(t) \cdot \pi \rrbracket_{\Gamma,\Delta} = (\llbracket t \rrbracket^{!}_{\Gamma,\Delta} \otimes \llbracket \pi \rrbracket_{\Gamma,\Delta})\, \mathsf{c}_{\llbracket \Gamma \rrbracket \otimes \llbracket \Delta \rrbracket} \in \mathcal{L}^{!}(\llbracket \Gamma \rrbracket \otimes \llbracket \Delta \rrbracket, \llbracket \sigma \Rightarrow \tau \rrbracket)$.

Assume last that $\Gamma \mid \pi : \sigma \vdash \Delta$ so that $\llbracket \pi \rrbracket_{\Gamma,\Delta} \in \mathcal{L}^{!}(\llbracket \Gamma \rrbracket \otimes \llbracket \Delta \rrbracket, \llbracket \sigma \rrbracket)$ and we can set $\llbracket \mathsf{pr}_1 \cdot \pi \rrbracket = \mathsf{in}^1\, \llbracket \pi \rrbracket_{\Gamma,\Delta} \in \mathcal{L}^{!}(\llbracket \Gamma \rrbracket \otimes \llbracket \Delta \rrbracket, \llbracket \sigma \rrbracket \oplus \llbracket \tau \rrbracket)$ and $\llbracket \mathsf{pr}_2 \cdot \pi \rrbracket$ is defined similarly.

Assume now that $\mathcal{L}$ is a model of LL with MIX, see Section 3. Here is the interpretation of the MIX constructs of Section 2. If $c = \mathsf{err}$, with $\Gamma \vdash \mathsf{err} \mid \Delta$, then $\llbracket c \rrbracket_{\Gamma,\Delta} = \mathsf{mix}^0\, \mathsf{w}_{\llbracket \Gamma \rrbracket \otimes \llbracket \Delta \rrbracket}$. If $c = c_1 \| c_2$ with $\Gamma \vdash c_i \mid \Delta$ for $i = 1, 2$, we set $\llbracket c \rrbracket_{\Gamma,\Delta} = \mathsf{mix}^2\, (\llbracket c_1 \rrbracket \otimes \llbracket c_2 \rrbracket)\, \mathsf{c}_{\llbracket \Gamma \rrbracket \otimes \llbracket \Delta \rrbracket}$.

▶ **Theorem 3** (Soundness). *Assume that $\Gamma \vdash c \mid \Delta$ and that $c \to c'$. Then $\llbracket c \rrbracket_{\Gamma,\Delta} = \llbracket c' \rrbracket_{\Gamma,\Delta}$.*

3.2 A continuation category

We recall briefly the connection between this LL-based approach and the Lafont-Reus-Streicher (LRS) [15] approach of continuation categories, see [11] for more details[2]. Let $\mathcal{P} = \mathcal{L}^{!}$, we have seen that $\mathcal{P}$ is a cocartesian and cartesian category, with $\oplus$ as coproduct and $\otimes$ as product. As object of responses, we take $\Sigma = !\perp$. Let P and Q be objects of $\mathcal{P}$. Then we have $\mathcal{P}(P \otimes Q, \Sigma) = \mathcal{L}^{!}(P \otimes Q, !\perp) \simeq \mathcal{L}(\underline{P} \otimes \underline{Q}, \perp)$ because $!\_$ is right adjoint to U. Hence $\mathcal{P}(P \otimes Q, \Sigma) \simeq \mathcal{L}(\underline{P}, \underline{Q}^{\perp}) \simeq \mathcal{L}^{!}(P, !(\underline{Q}^{\perp}))$ by the same adjunction. So setting $\Sigma^{Q} = !(\underline{Q}^{\perp})$ we have $\mathcal{P}(P \otimes Q, \Sigma) \simeq \mathcal{P}(P, \Sigma^{Q})$. Hence Σ is a baseable object of $\mathcal{P}$.

The category $\Sigma^{\mathcal{P}}$ of *negated objects* has the same objects as $\mathcal{P}$, and $\Sigma^{\mathcal{P}}(P, Q) = \mathcal{P}(\Sigma^{P}, \Sigma^{Q})$. It is a cartesian closed category with product $P \times Q = P \oplus Q$ and object of morphisms $P \Rightarrow Q = \Sigma^{P} \otimes Q$ as can easily be checked, using the fact that Σ is baseable. In the LRS setting, interpretation of types is done in $\mathcal{P}$, setting $\llbracket \sigma \Rightarrow \tau \rrbracket = \Sigma^{\llbracket \sigma \rrbracket} \otimes \llbracket \tau \rrbracket$ and, given contexts $\Gamma = (x_1 : \sigma_1, \dots, x_n : \sigma_n)$ and $\Delta = (\alpha_1 : \tau_1, \dots, \alpha_k : \tau_k)$, a term t such that $\Gamma \vdash t : \sigma \mid \Delta$ is interpreted as $\llbracket t \rrbracket_{\Gamma,\Delta} \in \mathcal{P}(\Sigma^{\llbracket \sigma_1 \rrbracket} \times \cdots \times \Sigma^{\llbracket \sigma_n \rrbracket} \times \llbracket \tau_1 \rrbracket \times \cdots \times \llbracket \tau_k \rrbracket, \Sigma^{\llbracket \sigma \rrbracket})$, a command c such that $\Gamma \vdash c \mid \Delta$ is interpreted as $\llbracket c \rrbracket_{\Gamma,\Delta} \in \mathcal{P}(\Sigma^{\llbracket \sigma_1 \rrbracket} \times \cdots \times \Sigma^{\llbracket \sigma_n \rrbracket} \times \llbracket \tau_1 \rrbracket \times \cdots \times \llbracket \tau_k \rrbracket, \Sigma)$ and a stack π such that $\Gamma \mid \pi : \tau \vdash \Delta$ is interpreted as $\llbracket \pi \rrbracket_{\Gamma,\Delta} \in \mathcal{P}(\Sigma^{\llbracket \sigma_1 \rrbracket} \times \cdots \times \Sigma^{\llbracket \sigma_n \rrbracket} \times \llbracket \tau_1 \rrbracket \times \cdots \times \llbracket \tau_k \rrbracket, \llbracket \tau \rrbracket)$ and it is easily checked again that this interpretation is exactly the same as the one described above, up to the identification of $\mathcal{P}(P, !X)$ with $\mathcal{L}(\underline{P}, X)$.

[2] This paper establishes the correspondence with Selinger control categories [14] which are equivalent to continuation categories. They use therefore a negative translation whereas we use a positive one.

4 Relational semantics

In this most simple and canonical interpretation of LL, $\mathcal{L}$ is the category $\mathbf{Rel}$ whose objects are sets[3] and where $\mathbf{Rel}(X, Y) = \mathcal{P}(X \times Y)$, composition being defined as the usual composition of relations. We recall that the tensor unit is $1 = \{*\}$ (arbitrary one-point set), that $X \otimes Y = X \times Y$ with tensor product of morphisms defined accordingly, that $X \multimap Y = X \times Y$ (and evaluation defined in the obvious way), that $\perp = 1$ so that $X^{\perp} = X$ up to canonical iso. This category is countably cartesian, with cartesian product $\&_{i \in I} X_i = \bigcup_{i \in I}(\{i\} \times X_i)$ (disjoint union) and projections defined in the obvious way ($\mathsf{pr}^i = \{((i, a), a) \mid a \in X_i\}$). It is cocartesian with coproducts defined exactly as products and injections given by $\mathsf{in}^i = \{(a, (i, a)) \mid a \in X_i\}$. It has an exponential functor defined on objects by $!X = \mathcal{M}_{\mathrm{fin}}(X)$, the set of all finite multisets[4] of elements of X. On morphisms, this functor is defined by $!f = \{([a_1, \dots, a_n], [b_1, \dots, b_n]) \mid (a_i, b_i) \in f \text{ for each } i\}$. Dereliction (counit) is given by $\mathsf{der}_X = \{([a], a) \mid a \in X\}$ and digging (comultiplication) is given by $\mathsf{dig}_X = \{(m_1 + \dots + m_k, [m_1, \dots, m_k]) \mid m_1, \dots, m_k \in !X\}$. The symmetric monoidality isos are given by $\mathsf{m}^{(0)} = \{(*, [])\}$ and

$$\mathsf{m}^{(2)}_{X,Y} = \{(([a_1, \dots, a_n], [b_1, \dots, b_k]), [(1, a_1), \dots, (1, a_n), (2, b_1), \dots, (2, b_k)]) \mid \\ a_1, \dots, a_n \in X \text{ and } b_1, \dots, b_k \in Y\}$$

Let $P = (\underline{P}, \mathsf{h}_P)$ be an object of $\mathbf{Rel}^!$ and X be an object of $\mathbf{Rel}$. Given $f \in \mathbf{Rel}(\underline{P}, X)$, the generalized promotion $f^! \in \mathbf{Rel}^!(P, !X)$ is given by $f^! = \{(b, [a_1, \dots, a_n]) \mid \exists b_1, \dots, b_n \in \underline{P}\ (b, [b_1, \dots, b_n]) \in \mathsf{h}_P \text{ and } (b_i, a_i) \in f \text{ for each } i\}$. The n-ary contraction $\mathsf{c}^{(n)}_P \in \mathbf{Rel}^!(P, P^{\otimes n})$ is given by $\mathsf{c}^{(n)}_P = \{(a, (a_1, \dots, a_n)) \mid (a, [a_1, \dots, a_n]) \in \mathsf{h}_P\}$. In particular (0-ary case) we have $\mathsf{w}_P = \{(a, *) \mid (a, []) \in \mathsf{h}_P\}$. The next easy lemma is essential for computing the interpretation of expressions, using *eg.* the formalism of Section 4.1.

▶ **Lemma 4.** *Let P_1 and P_2 be objects of $\mathbf{Rel}^!$. One has $((a, b), [(a_1, b_1), \dots, (a_n, b_n)]) \in \mathsf{h}_{P_1 \otimes P_2}$ iff $(a, [a_1, \dots, a_n]) \in \mathsf{h}_{P_1}$ and $(b, [b_1, \dots, b_n]) \in \mathsf{h}_{P_2}$. And, given $l \in \{1, 2\}$, one has $((l, a), [b_1, \dots, b_n]) \in \mathsf{h}_{P_1 \oplus P_2}$ iff, for each $i = 1, \dots, n$, one has $b_i = (l, a_i)$ for some a_i, and moreover $(a, [a_1, \dots, a_n]) \in \mathsf{h}_{P_l}$.*

For each set X, we can define a fix-point operator as a least fix-point wrt. morphism inclusion as $\overline{\mathsf{fix}}_X = \{(m_1 + \dots + m_k + [([a_1, \dots, a_k], a)], a) \mid \forall i\ (m_i, a_i) \in \overline{\mathsf{fix}}_X\}$.

The object of natural numbers is the set $\mathbb{N}$, the morphisms $\overline{\mathsf{succ}}$ and $\overline{\mathsf{pred}}$ are given by $\overline{\mathsf{succ}} = \{(n, n{+}1) \mid n \in \mathbb{N}\}$, $\overline{\mathsf{pred}} = \{(0, 0\} \cup \{(n{+}1, n) \mid n \in \mathbb{N}\}$. When $X = \underline{P}^{\perp}$ where P is an object of $\mathbf{Rel}^!$, the corresponding coalgebra morphism $\overline{\mathsf{If}}_X \in \mathbf{Rel}^!(P \otimes !\underline{P}^{\perp} \otimes !\underline{P}^{\perp}, !\mathsf{N}^{\perp})$ is given by $\overline{\mathsf{If}}_X = \{(a, [a_1, \dots, a_k], [a_{k+1}, \dots, a_l], [n_1, \dots, n_l]) \mid n_1 = \dots = n_k = 0 \text{ and } n_{k+1}, \dots, n_l \neq 0 \text{ and } (a, [a_1, \dots, a_l]) \in \mathsf{h}_P\}$. This model of LL is also a model of MIX. It suffices to take $\mathsf{mix}^0 = \{(*, *)\}$ and $\mathsf{mix}^2 = \{((*, *), *)))\}$ and these morphisms define clearly a structure of commutative $\otimes$-monoid on $\perp$.

So a typing variable context $\Gamma = (x_1 : \sigma_1, \dots, x_n : \sigma_n)$ is interpreted as a set of tuples $(m_1, \dots, m_n)$ where $m_i \in \mathcal{M}_{\mathrm{fin}}([\![\sigma_i]\!])$ for each i, a typing name context $\Delta = (\alpha_1 : \tau_1, \dots, \alpha_k : \tau_k)$ is interpreted as a set of tuples $(a_1, \dots, a_k)$ where $a_j \in [\![\tau_j]\!]$ for each j.

[3] All sets can be assumed to be at most countable, this is a very reasonable assumption which is preserved by all the constructions that we introduce.

[4] We use $[a_1, \dots, a_k]$ for the multiset whose elements are $a_1, \dots, a_n$, taking multiplicities into account and we use $m + m'$ for the disjoint union of the multiset m and m' which is a natural notation since multisets are $\mathbb{N}$-valued functions. Similarly if $k \in \mathbb{N}$ and m is a multiset, $km = m + \dots + m$ (k times).

With these notations for Γ and Δ, if $\Gamma \vdash M : \tau \mid \Delta$ then $\llbracket M \rrbracket_{\Gamma,\Delta}$ is a set of tuples $(m_1, \dots, m_n, a_1, \dots, a_k, a)$ where $a \in \llbracket \tau \rrbracket$, if $\Gamma \mid \pi : \sigma \vdash \Delta$ then $\llbracket \pi \rrbracket_{\Gamma,\Delta}$ is a set of tuples $(m_1, \dots, m_n, a_1, \dots, a_k, a)$ where $a \in \llbracket \sigma \rrbracket$, and if $\Gamma \vdash c \mid \Delta$ then $\llbracket c \rrbracket_{\Gamma,\Delta}$ is a set of tuples $(m_1, \dots, m_n, a_1, \dots, a_k)$. In all cases $m_i \in \mathcal{M}_{\mathrm{fin}}(\llbracket \sigma_i \rrbracket)$ and $a_j \in \llbracket \tau_j \rrbracket$ for each i and j.

4.1 Interpretation as a type deduction system

We introduce a typing system extending the one of [16] to represent the relational denotational semantics described above. A *semantic variable context* is a sequence $\Phi = (x_1 : m_1 : \sigma_1, \dots, x_n : m_n : \sigma_n)$ where $m_i \in !\underline{\llbracket \sigma_i \rrbracket}^{\perp}$ for each i and variables are pairwise distinct. A *semantic name context* is a sequence $\Psi = (\alpha_1 : a_1 : \tau_1, \dots, \alpha_k : a_k : \tau_k)$ where $a_i \in \underline{\llbracket \tau_i \rrbracket}$ for each i and the names are pairwise distinct. We also define the underlying typing contexts $\mathsf{u}(\Phi) = (x_1 : \sigma_1, \dots, x_n : \sigma_n)$ and $\mathsf{u}(\Psi) = (\alpha_1 : \tau_1, \dots, \alpha_k : \tau_k)$ as well as the underlying tuples $\langle \Phi \rangle = (m_1, \dots, m_n)$ and $\langle \Psi \rangle = (a_1, \dots, a_k)$. We extend multiset addition to tuples of multisets componentwise.

One has $\langle \Phi \rangle \in \underline{\llbracket \mathsf{u}(\Phi) \rrbracket}$ and similarly for Ψ. Given a variable context $\Gamma = (x_1 : \sigma_1, \dots, x_n : \sigma_n)$ one defines the corresponding *zero semantic context* $0_\Gamma = (x_1 : [] : \sigma_1, \dots, x_n : [] : \sigma_n)$.

We define now this typing system. Its main property (proven by an easy induction on expressions) is that $\Phi \vdash t : a : \sigma \mid \Psi$ iff $(\langle \Phi \rangle, \langle \Psi \rangle, a) \in \llbracket t \rrbracket_{\mathsf{u}(\Phi),\mathsf{u}(\Psi)}$ and $\Phi \mid \pi : a : \sigma \vdash \Psi$ iff $(\langle \Phi \rangle, \langle \Psi \rangle, a) \in \llbracket \pi \rrbracket_{\mathsf{u}(\Phi),\mathsf{u}(\Psi)}$, and also $\Phi \vdash c \mid \Psi$ iff $(\langle \Phi \rangle, \langle \Psi \rangle) \in \llbracket c \rrbracket_{\mathsf{u}(\Phi),\mathsf{u}(\Psi)}$. Here are the axioms and deduction rules:

$$\frac{}{0_\Gamma, x : [a] : \sigma \vdash x : a : \sigma \mid \Psi} \qquad \frac{}{0_\Gamma \mid \alpha : a : \sigma \vdash \alpha : a : \sigma, \Psi}$$

if $(\langle \Psi \rangle, 0_{\llbracket \mathsf{u}(\Psi) \rrbracket}) \in \mathsf{h}_{\llbracket \mathsf{u}(\Psi) \rrbracket}$.

$$\frac{\Phi_1 \vdash t : a : \sigma \mid \Psi_1 \qquad \Phi_2 \mid \pi : a : \sigma \vdash \Psi_2}{\Phi \vdash t * \pi \mid \Psi}$$

if $\mathsf{u}(\Phi_i) = \mathsf{u}(\Phi)$ and $\mathsf{u}(\Psi_i) = \mathsf{u}(\Psi)$ for $i = 1, 2$, $\langle \Phi \rangle = \langle \Phi_1 \rangle + \langle \Phi_2 \rangle$ and $(\langle \Psi \rangle, [\langle \Psi_1 \rangle, \langle \Psi_2 \rangle]) \in \mathsf{h}_{\llbracket \Delta \rrbracket}$.

$$\frac{\Phi, x : m : \sigma \vdash t : b : \tau \mid \Psi}{\Phi \vdash \lambda x^\sigma\, t : (m, b) : \sigma \Rightarrow \tau \mid \Psi} \qquad \frac{\Phi \vdash s : a : \sigma \mid \Psi \qquad \mathsf{u}(\Phi) \vdash t : \tau \mid \mathsf{u}(\Psi)}{\Phi \vdash \langle s, t \rangle : (1, a) : \sigma \times \tau \mid \Psi}$$

$$\frac{\mathsf{u}(\Phi) \vdash s : \sigma \mid \mathsf{u}(\Psi) \qquad \Phi \vdash t : b : \sigma \mid \Psi}{\Phi \vdash \langle s, t \rangle : (2, b) : \sigma \times \tau \mid \Psi} \qquad \frac{\Phi_0 \mid \pi : b : \tau \vdash \Psi_0 \qquad (\Phi_i \vdash t : a_i : \sigma \mid \Psi_i)_{i=1}^{k}}{\Phi \mid \mathsf{arg}(t) \cdot \pi : ([a_1, \dots, a_k], b) : \sigma \Rightarrow \tau \vdash \Psi}$$

$$\frac{\Phi_0, x : [a_1, \dots, a_k] : \sigma \vdash t : a : \sigma \mid \Psi_0 \qquad (\Phi_i \vdash \mathsf{fix}\, x^\sigma\, t : a_i : \sigma \mid \Psi_i)_{i=1}^{k}}{\Phi \vdash \mathsf{fix}\, x^\sigma\, t : a : \sigma \mid \Psi}$$

if $\mathsf{u}(\Phi_i) = \mathsf{u}(\Phi)$, $\mathsf{u}(\Psi_i) = \mathsf{u}(\Psi)$ for each $i = 0, \dots, k$, $\langle \Phi \rangle = \langle \Phi_0 \rangle + \dots + \langle \Phi_k \rangle$ and $(\langle \Psi \rangle, [\langle \Psi_0 \rangle, \dots, \langle \Psi_k \rangle]) \in \mathsf{h}_{\llbracket \mathsf{u}(\Psi) \rrbracket}$, for the two last deduction rules.

$$\frac{\Phi \mid \pi : a : \sigma \vdash \Psi}{\Phi \mid \mathsf{pr}_1 \cdot \pi : (1, a) : \sigma \times \tau \vdash \Psi} \qquad \frac{\Phi \mid \pi : b : \tau \vdash \Psi}{\Phi \mid \mathsf{pr}_2 \cdot \pi : (2, b) : \sigma \times \tau \vdash \Psi}$$

$$\frac{\Phi \vdash c \mid \alpha : a : \sigma, \Psi}{\Phi \vdash \mu\alpha^\sigma\, c : a : \sigma \mid \Psi} \qquad \frac{(\langle \Psi \rangle, 0_{\llbracket \mathsf{u}(\Psi) \rrbracket}) \in \mathsf{h}_{\llbracket \mathsf{u}(\Psi) \rrbracket} \qquad n \in \mathbb{N}}{0_\Gamma \vdash \underline{n} : n : \iota \mid \Psi}$$

$$\frac{\Phi \mid \pi : [n_1 + 1, \dots, n_k + 1] : \iota \vdash \Psi}{\Phi \mid \mathsf{succ} \cdot \pi : [n_1, \dots, n_k] : \iota \vdash \Psi} \qquad \frac{\Phi \mid \pi : k[0] + [n_1, \dots, n_l] : \iota \vdash \Psi}{\Phi \mid \mathsf{pred} \cdot \pi : k[0] + [n_1 + 1, \dots, n_l + 1] : \iota \vdash \Psi}$$

$$\frac{\Phi_0 \mid \pi : a : \sigma \vdash \Psi_0 \qquad (\Phi_i \vdash t_1 : a_i : \sigma \mid \Psi_i)_{i=1}^{k} \qquad (\Phi_i \vdash t_2 : a_i : \sigma \mid \Psi_i)_{i=k+1}^{l}}{\Phi \mid \mathsf{if}(t_1, t_2) \cdot \pi : \iota : k[0] + [n_1 + 1, \ldots, n_{l-k} + 1] \vdash \Psi}$$

if $(a, [a_1, \ldots, a_l]) \in \mathsf{h}_{[\![\sigma]\!]}$, $\mathsf{u}(\Phi_i) = \mathsf{u}(\Phi)$, $\mathsf{u}(\Psi_i) = \mathsf{u}(\Psi)$ for each i, $\langle\Phi\rangle = \langle\Phi_0\rangle + \cdots + \langle\Phi_l\rangle$ and $(\langle\Psi\rangle, [\langle\Psi_0\rangle, \ldots, \langle\Psi_l\rangle]) \in \mathsf{h}_{[\![\mathsf{u}(\Psi)]\!]}$. For classical PCF with MIX, we add the rules:

$$\frac{(\langle\Psi\rangle, 0_{[\![\mathsf{u}(\Psi)]\!]}) \in \mathsf{h}_{[\![\mathsf{u}(\Psi)]\!]}}{0_\Gamma \vdash \mathsf{err} \mid \Psi} \qquad \frac{\Phi_1 \vdash c_1 \mid \Psi_1 \qquad \Phi_2 \vdash c_2 \mid \Psi_2}{\Phi \vdash c_1 \| c_2 \mid \Psi}$$

if $\mathsf{u}(\Phi_i) = \mathsf{u}(\Phi)$, $\mathsf{u}(\Psi_i) = \mathsf{u}(\Psi)$ for $i = 1, 2$, $\langle\Phi\rangle = \langle\Phi_1\rangle + \langle\Phi_2\rangle$ and $(\langle\Psi\rangle, [\langle\Psi_1\rangle, \langle\Psi_2\rangle]) \in \mathsf{h}_{[\![\mathsf{u}(\Psi)]\!]}$.

5 Adequacy

Our goal here is to prove that, in the full calculus (including the MIX constructions), if an almost closed command has a non-empty relational semantics, then its $\to$-reduction terminates. In other words, an almost closed command typable in the semantic typing system is $\to$-normalizing. Let $\mathcal{N}$ be the set of all $\to$-normalizing almost closed commands.

Let us say that a term t (resp. a stack π) is almost closed of type σ if $\vdash t : \sigma \mid \Delta$ (resp. $\mid \pi : \sigma \vdash \Delta$) for some ground name context Δ (that is, for any ground name context where all free names appear). Observe that if t and π are an almost closed term and an almost closed stack of the same type, then $t * \pi$ is an almost closed command.

By induction on σ, we define, for each $a \in [\![\sigma]\!]$, a set $\|a\|^\sigma$ of almost closed stacks of type σ. We use the notation $|a|^\sigma$ for the set of all almost closed terms t of type σ such that $t * \pi \in \mathcal{N}$ for all $\pi \in \|a\|^\sigma$. Given $a_1, \ldots, a_n \in [\![\sigma]\!]$, we set $|[a_1, \ldots, a_n]|^\sigma = \bigcap_{i=1}^{n} |a_i|^\sigma$.

The most important part of the definition is the base case: given $m = [n_1, \ldots, n_k] \in [\![\iota]\!] = !(\mathsf{N}^\perp)$, we define $\|m\|^\iota$ as the set of all almost closed stacks π of type ι such that $\forall i \in \{1, \ldots, k\}\ \underline{n_i} * \pi \in \mathcal{N}$. This set contains all names (considered of type ι) and hence is never empty.

The inductive step follows the general pattern of classical reducibility. Let σ and τ be types, let $a_1, \ldots, a_n \in [\![\sigma]\!]$ and $b \in [\![\tau]\!]$. We set

$$\|([a_1, \ldots, a_n], b)\|^{\sigma \Rightarrow \tau} = \left\{ \mathsf{arg}(t) \cdot \pi \mid t \in |[a_1, \ldots, a_n]|^\sigma \text{ and } \pi \in \|b\|^\tau \right\}$$

Let $a \in [\![\sigma]\!]$, we set $\|(1, a)\|^{\sigma \times \tau} = \{\mathsf{pr}_1 \cdot \pi \mid \pi \in \|a\|^\sigma\}$ and $\|(2, b)\|^{\sigma \times \tau}$ is defined similarly for $b \in [\![\tau]\!]$.

▶ **Theorem 5** (Adequacy). *Let $\Phi = (x_1 : m_1 : \sigma_1, \ldots, x_n : m_n : \sigma_n)$ and $\Psi = (\alpha_1 : a_1 : \tau_1, \ldots, \alpha_k : a_k : \tau_k)$ be semantic contexts. Let σ be a type, t be a term, c be a command and π be a stack such that*

- $\Phi \vdash t : a : \sigma \mid \Psi$
- *resp.* $\Phi \vdash c \mid \Psi$,
- *resp.* $\Phi \mid \pi : a : \sigma \vdash \Psi$.

Then, for all $t_1 \in |m_1|^{\sigma_1}, \ldots, t_n \in |m_n|^{\sigma_n}$ and all $\pi_1 \in \|a_1\|^{\tau_1}, \ldots, \pi_k \in \|a_k\|^{\tau_k}$, one has

- $t\,[t_1/x_1, \ldots, t_n/x_n]\,[\pi_1/\alpha_1, \ldots, \pi_k/\alpha_k] \in |a|^\sigma$
- *resp.* $c\,[t_1/x_1, \ldots, t_n/x_n]\,[\pi_1/\alpha_1, \ldots, \pi_k/\alpha_k] \in \mathcal{N}$,
- *resp.* $\pi\,[t_1/x_1, \ldots, t_n/x_n]\,[\pi_1/\alpha_1, \ldots, \pi_k/\alpha_k] \in \|a\|^\sigma$.

So, if an almost closed command c has a non-empty interpretation, it normalizes for the $\to$-reduction to a uniquely defined normal almost closed command which can easily be retrieved from the semantics of c. For instance if c satisfies $\vdash c \mid \Delta$ where $\Delta = (\alpha_1 : \iota, \alpha_2 : \iota, \alpha_3 : \iota)$, and if we have $\vdash c \mid \alpha_1 : [0,7] : \iota, \alpha_2 : [3] : \iota, \alpha_3 : [] : \iota$, then c normalizes by Theorem 5. Its normal form c_0 satisfies $\vdash c_0 \mid \alpha_1 : \iota, \alpha_2 : \iota, \alpha_3 : \iota$ by Proposition 1 and hence must be of shape $(n_1^1 * \alpha_1) \| \cdots \| (n_1^{l_1} * \alpha_1) \| (n_2^1 * \alpha_2) \| \cdots \| (n_2^{l_2} * \alpha_2) \| (n_3^1 * \alpha_3) \| \cdots \| (n_3^{l_3} * \alpha_3)$ up to associativity and commutativity of $\|$, by the final considerations of Section 2. By definition of the interpretation, $[\![c_0]\!]_{(),\Delta} = \{([n_1^1, \dots, n_1^{l_1}], [n_2^1, \dots, n_2^{l_2}], [n_3^1, \dots, n_3^{l_3}])\}$. But by Theorem 3 we have $[\![c]\!]_{(),\Delta} = [\![c_0]\!]_{(),\Delta}$ and hence *we must have* $l_1 = 2$, $l_2 = 1$, $l_3 = 0$, $n_1^1 = 0$, $n_1^2 = 7$ (or conversely) and $n_1^2 = 3$. This adequacy property entails that denotational equivalence of terms implies their observational equivalence (to be suitably defined)[5].

The considerations above show that the interpretation of an almost closed command contains at most one element. This can also be proved purely semantically, endowing the relational semantics with a binary *coherence relation*.

If c does not contain MIX constructs, we know that it will reduce to a normal command of shape $\underline{n} * \alpha$, but the model does not reflect this property that we proved syntactically in Section 2. We introduce now a light refinement of the relational model which takes this *uniqueness of values* property into account, and therefore rejects the MIX constructs.

6 A semantic account of uniqueness of values

This model originates from the observation made independently by several authors[6] at an early stage of the development of LL that, in a multiplicative proof-net, there is a simple relation between the number of $\otimes$'s and of $⅋$'s.

A *weighted set* is a pair $X = (|X|, \gamma_X)$ where $|X|$ is a set and $\gamma_X : |X| \to \mathbb{Z}$ is a function. If we think of a as a proof tree in (constant-free) multiplicative LL (MLL) with only one conclusion (the root of the tree), then $\gamma_X(a) = p - t$ where p is the number of $⅋$ and t is the number of $\otimes$ binary connectives occurring in a. If such a multiplicative proof tree can be sequentialized into a sequent calculus proof in MLL, then $p - t = 1$, see *eg.* [6], pages 250-251 (the converse is not true). This intuition explains the next definitions. One sets $\mathsf{C}(X) = \{x \subseteq |X| \mid \forall a \in x \; \gamma_X(a) = 1\}$.

Let **RelW** be the category of weighted sets and such that $\mathbf{RelW}(X,Y) = \{t \subseteq |X| \times |Y| \mid \forall (a,b) \in t \; \gamma_X(a) = \gamma_Y(b)\}$. Then $\mathsf{Id}_X = \{(a,a) \mid a \in |X|\} \in \mathbf{RelW}(X,X)$ and the relational composition of two morphisms is a morphism, so **RelW** is a category.

One defines the weighted set 1 by $|1| = \{*\}$ (a singleton) and $\gamma_1(*) = 1$. Given two weighted sets X_1 and X_2, one defines $X_1 \otimes X_2$ by $|X_1 \otimes X_2| = |X_1| \times |X_2|$ and $\gamma_{X_1 \otimes X_2}(a_1, a_2) = \gamma_{X_1}(a_1) + \gamma_{X_2}(a_2) - 1$. Given $t_i \in \mathbf{RelW}(X_i, Y_i)$ for $i = 1, 2$, one defines $t_1 \otimes t_2$ as in **Rel**, then it is clear that $t_1 \otimes t_2 \in \mathbf{RelW}(X_1 \otimes X_2, Y_1 \otimes Y_2)$ and that this operation is a functor. Moreover, the usual bijections $|1 \otimes X| \to |X|$, $|X \otimes 1| \to |X|$ and $|(X_1 \otimes X_2) \otimes X_3| \to |X_1 \otimes (X_2 \otimes X_3)|$ are isos in **RelW**. Indeed we have $\gamma_{1 \otimes X}(*, a) = 1 + \gamma_X(a) - 1 = \gamma_X(a)$ and $\gamma_{(X_1 \otimes X_2) \otimes X_3}((a_1, a_2), a_3) = \gamma_{X_1}(a_1) + \gamma_{X_2}(a_2) + \gamma_{X_3}(a_3) - 2 = \gamma_{X_1 \otimes (X_2 \otimes X_3)}(a_1, (a_2, a_3))$.

In that way, we have equipped **RelW** with a structure of symmetric monoidal category. We check that it is closed. Given two weighted sets X and Y, let $X \multimap Y = (|X| \times |Y|, \gamma_{X \multimap Y})$ and $\gamma_{X \multimap Y}(a, b) = \gamma_Y(b) - \gamma_X(a) + 1$. Observe that $\mathsf{C}(X \multimap Y) = \mathbf{RelW}(X, Y)$.

Then $\mathsf{ev} = \{(((a,b),a),b) \mid a \in |X| \text{ and } b \in |Y|\}$ belongs to $\mathbf{RelW}((X \multimap Y) \otimes X, Y)$. Indeed we have $\gamma_{(X \multimap Y) \otimes X}((a,b),a) = \gamma_Y(b) - \gamma_X(a) + 1 + \gamma_X(a) - 1 = \gamma_Y(b)$.

[5] The converse implication (full abstraction) is far from being true.

[6] At least: Girard, Danos and Regnier, Métayer, Fleury and Rétoré, Guerrini...

Let Z be another weighted set and let $((c,a),b) \in |(Z \otimes X) \multimap Y|$. Then we have $\gamma_{(Z\otimes X)\multimap Y}((c,a),b) = \gamma_Y(b) - (\gamma_Z(c) + \gamma_X(a) - 1) + 1 = \gamma_Y(b) - \gamma_Z(c) - \gamma_X(a) + 2$. On the other hand we have $\gamma_{Z\multimap(X\multimap Y)}(c,(a,b)) = (\gamma_Y(b) - \gamma_X(a) + 1) - \gamma_Z(c) + 1 = \gamma_Y(b) - \gamma_Z(c) - \gamma_X(a) + 2$ and therefore, given $t \in \mathbf{RelW}(Z \otimes X, Y)$, we have $\mathsf{cur}(t) = \{(c,(a,b)) \mid ((c,a),b) \in t\} \in \mathbf{RelW}(Z, X \multimap Y)$. This shows that $\mathbf{RelW}$ is closed.

Let $\bot = (\{*\}, \gamma_\bot)$ with $\gamma_\bot(*) = -1$. Then we have $\gamma_{X\multimap\bot}(a,*) = -1 - \gamma_X(a) + 1 = -\gamma_X(a)$. It follows that the canonical morphism $\eta_X \in \mathbf{RelW}(X, (X \multimap \bot) \multimap \bot)$ given by $\eta_X = \mathsf{cur}(\mathsf{ev}\ \sigma_{X,X\multimap\bot})$ (where σ is the symmetry natural iso associated with the symmetric monoidal closed structure of $\mathbf{RelW}$) is an iso in $\mathbf{RelW}$. This shows that, equipped with $\bot$ as dualizing object, the symmetric monoidal closed category $\mathbf{RelW}$ is *-autonomous.

The co-tensor product, called *par*, is the operation defined by $X \parr Y = (X^\perp \otimes Y^\perp)^\perp$ and is characterized by $|X \parr Y| = |X| \times |Y|$ and $\gamma_{X\parr Y}(a,b) = \gamma_X(a) + \gamma_Y(b) + 1$.

Let $X^\perp = (|X|, -\gamma_X)$. Then $X^\perp$ is naturally isomorphic to $X \multimap \bot$ and defines a strictly involutive functor $\mathbf{RelW} \to \mathbf{RelW}^{\mathsf{op}}$. Its action on morphisms is contraposition: $t^\perp = \{(b,a) \mid (a,b) \in t\} \in \mathbf{RelW}(Y^\perp, X^\perp)$ for any $t \in \mathbf{RelW}(X,Y)$.

The category $\mathbf{RelW}$ is cartesian and cocartesian. Given a family $(X_i)_{i\in I}$ of objects, let $X = \&_{i\in I} X_i$ be defined by $|X| = \bigcup_{i\in I}\{i\} \times |X_i|$ and $\gamma_X(i,a) = \gamma_{X_i}(a)$. Let $\mathsf{pr}^i = \{((i,a),a) \mid a \in |X_i|\} \in \mathbf{RelW}(X, X_i)$. Then $(X, (\mathsf{pr}^i)_{i\in I})$ is a cartesian product of the family $(X_i)_{i\in I}$. The coproduct is defined in a completely similar way. Observe that the product of the empty family (the terminal object) is $\top = (\emptyset, \emptyset)$, which is also the initial object of $\mathbf{RelW}$.

Let $!X = (\mathcal{M}_{\mathrm{fin}}(|X|), \gamma_{!X})$ where

$$\gamma_{!X}([a_1,\dots,a_n]) = \gamma_{X^{\otimes n}}(a_1,\dots,a_n) = -n + 1 + \sum_{i=1}^{n} \gamma_X(a_i) = 1 + \sum_{i=1}^{n} (\gamma_X(a_i) - 1)\,.$$

Given $t \in \mathbf{RelW}(X,Y)$, it is clear that $!t \in \mathbf{RelW}(!X, !Y)$ where $!t$ is defined as in $\mathbf{Rel}$.

So $!\_$ is a functor $\mathbf{RelW} \to \mathbf{RelW}$. We equip this functor with a structure of comonad. For each object X, let $\mathsf{der}_X = \{([a], a) \mid a \in |X|\}$. Since $\gamma_{!X}([a]) = \gamma_X(a)$, we have $\mathsf{der}_X \in \mathbf{RelW}(!X, X)$. The naturality of der_X is obvious (it already holds in $\mathbf{Rel}$).

One defines also $\mathsf{dig}_X = \{(m_1 + \cdots + m_k, [m_i, \dots, m_k]) \mid k \in \mathbb{N} \text{ and } \forall i\, m_i \in \mathcal{M}_{\mathrm{fin}}(|X|)\}$. Let $m_1, \dots, m_k \in \mathcal{M}_{\mathrm{fin}}(|X|)$, and let us write $m_i = [a^i_1, \dots, a^i_{k_i}]$. We have

$$\begin{aligned}\gamma_{!!X}([m_1,\dots,m_k]) &= 1 + \sum_{i=1}^{k}(\gamma_{!X}(m_i) - 1) = 1 + \sum_{i=1}^{k}\big(1 + \big(\sum_{j=1}^{k_i}\gamma_X(a^i_j) - 1\big) - 1\big)\\ &= \gamma_{!X}(m_1 + \cdots + m_k)\end{aligned}$$

and therefore $\mathsf{dig}_X \in \mathbf{RelW}(!X, !!X)$. One proves easily that $(!X, \mathsf{der}_X, \mathsf{dig}_X)$ defines a comonad (the definition of this structure is the same as in $\mathbf{Rel}$).

Last we check that the standard Seely isos of $\mathbf{Rel}$ are morphisms in $\mathbf{RelW}$. The 0-ary iso is $\mathsf{m}^{(0)} = \{(*, [])\}$ and belongs to $\mathbf{RelW}(1, !\top)$ since $\gamma_{!\top}([]) = 1$. The binary version is $\mathsf{m}^{(2)}_{X,Y} = \{(([a_1,\dots,a_n],[b_1,\dots,b_p]),\ [(1,a_1),\dots,(1,a_n),(2,b_1),\dots,(2,b_p)]) \mid \forall i\, a_i \in |X| \text{ and } \forall j\, b_j \in |Y|\}$ and we prove that $\mathsf{m}^{(2)}_{X,Y} \in \mathbf{RelW}(!X \otimes !Y, !(X \,\&\, Y))$: indeed, with the notations of this definition, we have

$$\gamma_{!X\otimes !Y}([a_1,\dots,a_n],[b_1,\dots,b_p]) = 1 + \sum_{i=1}^{n}(\gamma_X(a_i) - 1) + 1 + \sum_{j=1}^{p}(\gamma_Y(b_j) - 1) - 1$$

$$= 1 + \sum_{i=1}^{n}(\gamma_X(a_i) - 1) + \sum_{j=1}^{p}(\gamma_Y(b_j) - 1) = \gamma_{!(X\&Y)}([(1,a_1),\dots,(1,a_n),(2,b_1),\dots,(2,b_p)]))$$

This ends the description of the purely logical structures of the model.

The object N of natural numbers (in the sense of Section 3) is the coproduct of ω copies of 1, so $|\mathsf{N}| = \mathbb{N}$ and $\gamma_{\mathsf{N}}(n) = 1$ for each $n \in \mathbb{N}$. The morphisms $\overline{\mathsf{succ}}$, $\overline{\mathsf{pred}}$ and $\overline{\mathsf{if}}$ as defined in Section 4 in the category **Rel** are also morphisms in **RelW** simply because they are defined using the universal property of N. Last, for any object X of **RelW**, the fix-point operator $\overline{\mathsf{fix}}_{|X|}$ as defined in Section 4 is also a morphism $!(!X \multimap X) \to X$ in **RelW**.

So **RelW** is a model of classical PCF in the sense of Section 3.1. Let $\llbracket\sigma\rrbracket^{\mathsf{w}}$ be the interpretation of the type σ in the category $\mathbf{RelW}^!$, we have $|\llbracket\sigma\rrbracket^{\mathsf{w}}| = \llbracket\sigma\rrbracket$ and $\mathsf{h}_{\llbracket\sigma\rrbracket^{\mathsf{w}}} = \mathsf{h}_{\llbracket\sigma\rrbracket}$ (as relations). If e is an expression typable in contexts Γ, Δ, we denote with $\llbracket e\rrbracket^{\mathsf{w}}_{\Gamma,\Delta}$ the interpretation of e in **RelW** (if e is a command or a term) or in $\mathbf{RelW}^!$ (if e is a stack).

▶ **Proposition 6.** *For any expression of classical PCF e typable in contexts Γ and Δ, one has $\llbracket e\rrbracket^{\mathsf{w}}_{\Gamma,\Delta} = \llbracket e\rrbracket_{\Gamma,\Delta}$ (as relations).*

This is due to the fact that the basic LL constructs are interpreted by the same relations in both models.

Rejection of MIX and uniqueness of values. Observe that $\mathsf{mix}^0 = \{(*, *)\} \in \mathbf{Rel}(1, \perp)$ and $\mathsf{mix}^2 = \{((*, *), *)\} \in \mathbf{Rel}(\perp \otimes \perp, \perp)$ *are not* morphisms in **RelW**: for mix^0, this is due to the fact that $\gamma_1(*) = 1$ and $\gamma_\perp(*) = -1$ and for mix^2, this is due to the fact that $\gamma_{\perp\otimes\perp}(*, *) = -3$.

Let $\Delta = (\alpha_1 : \iota, \dots, \alpha_k : \iota)$ be a ground name context. Then an almost closed command c such that $\vdash c \mid \Delta$ has interpretation $\llbracket c\rrbracket_{(),\Delta} \in \mathbf{RelW}(\llbracket\iota\rrbracket^{\mathsf{w}} \otimes \cdots \otimes \llbracket\iota\rrbracket^{\mathsf{w}}, \perp)$, that is $\llbracket c\rrbracket_{(),\Delta} \in \mathbf{RelW}(1, ?\mathsf{N} \parr \cdots \parr ?\mathsf{N})$. Let $m_1, \dots, m_k \in |?\mathsf{N}|$, then we have $\gamma_{?\mathsf{N}\parr\cdots\parr?\mathsf{N}}(m_1, \dots, m_k) = \gamma_{?N}(m_1) + \cdots + \gamma_{?N}(m_k) + k - 1 = 2(\#m_1) - 1 + \cdots + 2(\#m_k) - 1 + k - 1 = 2(\#m_1 + \cdots + \#m_k) - 1$. So any element $(m_1, \dots, m_k)$ of $\llbracket c\rrbracket_{(),\Delta}$ must satisfy $2(\#m_1 + \cdots + \#m_k) - 1 = 1$, that is $\#m_1 + \cdots + \#m_k = 1$. Hence there must exist $i \in \{1, \dots, k\}$ such that $\#m_i = 1$ and $m_j = []$ for $j \neq i$. We retrieve semantically the fact that c is single valued.

7 Conclusion

We have developed a semantic investigation of classical PCF, presented in Herbelin's very pleasant $\overline{\lambda}\mu$ format. We have recalled the general LL semantic framework for this calculus, based on Girard's categorical semantics of LC, and its connection with Lafont-Reus-Streicher continuation categories. We have outlined a simple adequacy proof for the relational model and proposed a model which enforces uniqueness of values, rejecting the extension of classical PCF by a parallel composition construct based on the MIX rule of LL. In a longer version of this paper, we shall show that the Eilenberg-Moore category of the Scott semantics of LL admits a very simple description. The relational model of LL is also deeply related with useful extensions of LL (systems with bounded complexity, differential LL etc) which could suggest interesting extensions of classical PCF. For these reasons, we think that the LL-based semantics of classical PCF is worth being further studied.

Acknowledgments. We would like to thank the referees for their careful reading of this paper and their insightful remarks, as well as Thomas Streicher for a long and very useful discussion at an early stage of this work.

References

1 Pierre-Louis Curien and Hugo Herbelin. The duality of computation. In Martin Odersky and Philip Wadler, editors, *Proceedings of the Fifth ACM SIGPLAN International Confer-*

ence on Functional Programming (ICFP'00), Montreal, Canada, September 18-21, 2000., pages 233–243. ACM, 2000.

2 Vincent Danos and Jean-Louis Krivine. Disjunctive Tautologies as Synchronisation Schemes. In Peter Clote and Helmut Schwichtenberg, editors, *CSL*, volume 1862 of *Lecture Notes in Computer Science*, pages 292–301. Springer-Verlag, 2000.

3 Daniel De Carvalho. Execution Time of λ-Terms via Denotational Semantics and Intersection Types. Research Report RR-6638, INRIA, 2008. To appear in Mathematical Structures in Computer Sciences.

4 Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

5 Jean-Yves Girard. A new constructive logic: classical logic. *Mathematical Structures in Computer Science*, 1(3):225–296, 1991.

6 Jean-Yves Girard. *The blind spot: lectures on logic.* European Mathematical Society, 2011. 537 pages.

7 Timothy G. Griffin. The Formulae-as-Types Notion of Control. In *Proceedings of the 17h ACM Symposium on Principles of Programming Languages (POPL)*, pages 47–57. Association for Computing Machinery, January 1990.

8 Hugo Herbelin. Games and Weak-Head Reduction for Classical PCF. In Philippe de Groote, editor, *TLCA*, volume 1210 of *Lecture Notes in Computer Science*, pages 214–230. Springer-Verlag, 1997.

9 Martin Hofmann and Thomas Streicher. Completeness of Continuation Models for lambda-mu-Calculus. *Information and Computation*, 179(2):332–355, 2002.

10 Olivier Laurent. Polarized games. *Annals of Pure and Applied Logic*, 130(1-3):79–123, 2004.

11 Olivier Laurent and Laurent Regnier. About Translations of Classical Logic into Polarized Linear Logic. In *18th IEEE Symposium on Logic in Computer Science (LICS 2003), 22-25 June 2003, Ottawa, Canada, Proceedings*, pages 11–20. IEEE Computer Society, 2003.

12 Paul-André Melliès. Categorical Semantics of Linear Logic. *Panoramas et Synthèses*, 27, 2009.

13 Michel Parigot. $\lambda\mu$-Calculus: An Algorithmic Interpretation of Classical Natural Deduction. In Andrei Voronkov, editor, *Logic Programming and Automated Reasoning,International Conference LPAR'92, St. Petersburg, Russia, July 15-20, 1992, Proceedings*, volume 624 of *Lecture Notes in Computer Science*, pages 190–201. Springer-Verlag, 1992.

14 Peter Selinger. Control Categories and Duality: on the Categorical Semantics of the Lambda-Mu Calculus. *Mathematical Structures in Computer Science*, 11(2):207–260, 2001.

15 Thomas Streicher and Bernhard Reus. Classical Logic, Continuation Semantics and Abstract Machines. *Journal of Functional Programming*, 8(6):543–572, 1998.

16 Lionel Vaux. Convolution lambda-bar-mu-calculus. In Simona Ronchi Della Rocca, editor, *Typed Lambda Calculi and Applications, 8th International Conference, TLCA 2007, Paris, France, June 26-28, 2007, Proceedings*, volume 4583 of *Lecture Notes in Computer Science*, pages 381–395. Springer, 2007.