Effects in Call-By-Push-Value, from a Linear Logic point of view

Thomas Ehrhard

Effects in Call-By-Push-Value, from a Linear Logic point of view

Thomas Ehrhard
CNRS, IRIF, UMR 8243,
Univ Paris Diderot, Sorbonne Paris Cité, F-75205 Paris, France
thomas.ehrhard@pps.univ-paris-diderot.fr

October 3, 2016

Abstract

We define and study a non deterministic extension of Call-By-Push-Value (CBPV) for which we prove an Adequacy and Full Abstraction theorem with respect to a Scott semantics of classical Linear Logic (LL). We also consider an extension of CBPV with a general notion of global state for which we propose a simple LL-based Scott denotational semantics and prove an adequacy result.

Introduction

The Call-By-Push-Value (CBPV) lambda-calculus was introduced in [10] as a a common framework for call-by-name (CBN) and call-by-value (CBV). CBPV takes place in a line of research mainly inspired by Moggi's computational lambda-calculus [14].

We propose to view CBPV as a refinement of CBN and CBV lambda-calculus factorizing the standard encodings of these calculi in Linear Logic (LL). Girard's CBN and CBV translation[1] of lambda-calculus into LL mainly differ by the way the exponential promotion rule is used. In CBN, an application $(s)t$ is defined as the linear application of s to the promotion of t and the abstraction $\lambda x\, s$ is simply interpreted as an abstraction, whereas in CBV, $(s)t$ is the linear application of the dereliction of s to t and $\lambda x\, s$ is interpreted as the promotion of the abstraction of s. In CBN a β-redex can be immediately reduced (the argument, being promoted, is freely discardable and duplicable) whereas in CBV one needs first to reduce the argument to a "value", that is, to a term whose translation is a promotion (abstraction or variable).

This suggests to understand a value as a term which can be freely duplicated and discarded. A way to implement this idea is to require values to have types which are interpreted in LL by formulas equipped with structural rules. This idea is used in [5] for representing classical logic in LL, identifying two dual classes of LL formulas: the positive and the negative formulas. A positive formula is equipped with a structure of !-coalgebra as explained in [9] (and a negative one is the linear negation of a positive formula) and morphisms of !-coalgebras are duplicable and discardable. $!A$ is a positive formula, and these formulas are closed under $\otimes$ and $\oplus$ (remark: products and sums are basic data type constructions and "!" can be used as a lazy type constructor). More precisely, given a categorical model $\mathcal{L}$ of LL, the Eilenberg-Moore category $\mathcal{L}^!$ of the "!" comonad is cartesian (with $\otimes$ as product) and cocartesian (with $\oplus$ as coproduct).

We base our version of CBPV on that idea: the type system features positive types $\varphi, \psi \dots$ and general types $\sigma, \tau \dots$. If σ is a general type then $!\sigma$ is a positive types and if φ_1, φ_2 are positive types, then $\varphi_1 \otimes \varphi_2$ and $\varphi_1 \oplus \varphi_2$ are positive types (we also admit a recursive positive type construction). Any positive type is a general type and if φ is a positive type and σ is a general type then $\varphi \multimap \sigma$ is a general type. Given a model $\mathcal{L}$ of LL, positive types are interpreted as objects of $\mathcal{L}^!$ and general types are interpreted in $\mathcal{L}$. We can define many data types as recursive types: unary natural number ($\iota = 1 \oplus \iota$ where 1 is a unit type which is also definable), streams of natural numbers ($\rho = \iota \otimes !\rho$) etc. Since we allow only to form function types of shape $\varphi \multimap \sigma$ with φ positive, even though this type is interpreted as a linear implication, there are no linearity restrictions on the use of variables in the calculus. It is not

[1] The CBN translation is central in [4] and the CBV translation is only alluded to and rejected as "boring".

true that all terms of positive types are interpreted as coalgebra morphisms, but this property holds for a particular class of terms: the values. So β-reduction can be performed only if the argument term is a value. Similar restrictions apply to the other reduction rules. The rewriting rules allow in particular to reduce terms of positive types to values.

This calculus, as well as its LL-based denotational semantics, is presented in [3]. We also refer to this paper for more references to related works. Our purpose here is to study its extensions with two kinds of effects.

The first effect we consider is non-determinism.

We extend the language with a construction $\mathsf{choose}(M, N)$ which reduces to M or N, non deterministically. We define an operational semantics by reducing terms to finite multisets of terms (corresponding to the various non-deterministic choices). We provide a denotational semantics for this calculus based on a Scott model of LL where types are interpreted as pre-ordered sets (such a pre-ordered set induces a complete lattice: the set of its downwards-closed subsets, and morphisms are the linear maps, that is the maps which commute with arbitrary unions). Interpreting multisets of terms as the union of their interpretations, one can prove a soundness property. By a realizability technique one proves a "sensibility" result, showing that if a term has a non-empty semantics then one of its non-deterministic reductions terminates. This property implies adequacy: if two terms have the same semantics then they are observationally equivalent. We prove then the converse (Full Abstraction), associating a term $a^- : !\sigma \multimap 1$ with any element a of the pre-ordered set $[\sigma]$ interpreting a type σ in such a way that, if u is a downwards-closed subset in $[\sigma]$, one has $a \in u$ iff the term $\langle a^- \rangle u^!$ terminates.

Last we consider a global states effect. We define a notion of "transition system with parameters" (TSP) which is a transition system where transitions are labeled by instructions applied to integer parameters, yielding integer results. A typical example is global memory: a state is a store and there is a read operations which takes one integer parameter (the address to be read) and yields the stored value, and a write operation which takes two integers (address and value to be stored) and returns 0.

Given a TSP $\mathcal{Q}$, we extend our CBPV with monadic type and term construction and with instructions associated with the transitions of $\mathcal{Q}$. We define a concrete denotational model for this extension of CBPV by considering the set of states of $\mathcal{Q}$ as an object of the Scott model of LL. We extend the sensibility result to this new syntax by extending our realizability method to states.

This collection of results shows that, though they are particular cases of the general notion of adjunction models of CBPV [11], the LL-based models are quite expressive.

1 Purely functional CBPV

We introduce our version of CBPV called Λ_{HP} (HP stands for "half-polarized"). Types are given by the following BNF syntax. We define by mutual induction two kinds of types: *positive types* and *general types*, given type variables ζ, $\xi \ldots$:

$$\text{positive} \quad \varphi, \psi, \ldots := !\sigma \mid \varphi \otimes \psi \mid \varphi \oplus \psi \mid \zeta \mid \mathsf{Fix}\,\zeta \cdot \varphi \tag{1}$$

$$\text{general} \quad \sigma, \tau \ldots := \varphi \mid \varphi \multimap \sigma \mid \top \tag{2}$$

More constructions could be added for general types, as in CBPV (such as products or recursive types), we do not so here by lack of space. We consider the types up to the equation $\mathsf{Fix}\,\zeta \cdot \varphi = \varphi\,[(\mathsf{Fix}\,\zeta \cdot \varphi)/\zeta]$.

Terms are given by the following BNF syntax, given variables $x, y, \ldots$:

$$\begin{aligned} M, N \ldots := {} & x \mid M^! \mid \langle M, N \rangle \mid \mathsf{in}_1 M \mid \mathsf{in}_2 M \\ & \mid \lambda x^{\varphi}\, M \mid \langle M \rangle N \mid \mathsf{case}(M, x_1 \cdot N_1, x_2 \cdot N_2) \\ & \mid \mathsf{pr}_1 M \mid \mathsf{pr}_2 M \mid \mathsf{der}(M) \mid \mathsf{fix}\, x^{!\sigma}\, M \end{aligned}$$

This calculus can be seen as a version of Levy's CBPV [11] in which the type constructor F is kept implicit (and U is "!"). It is close to SFPL [13]. We use LL inspired notations; $M^!$ corresponds to $\mathsf{thunk}(M)$ and $\mathsf{der}(M)$ to $\mathsf{force}(M)$.

Figure 1 provides the typing rules for these terms. A typing context is an expression $\mathcal{P} = (x_1 : \varphi_1, \ldots, x_k : \varphi_k)$ where all types are positive and the x_is are pairwise distinct variables.

$$\frac{\mathcal{P}\vdash M:\sigma}{\mathcal{P}\vdash M^!:!\sigma}\qquad \frac{\mathcal{P}\vdash M_1:\varphi_1 \qquad \mathcal{P}\vdash M_2:\varphi_2}{\mathcal{P}\vdash \langle M_1,M_2\rangle:\varphi_1\otimes\varphi_2}\qquad \frac{\mathcal{P}\vdash M:\varphi_i}{\mathcal{P}\vdash \mathsf{in}_i M:\varphi_1\oplus\varphi_2}\qquad \frac{}{\mathcal{P},x:\varphi\vdash x:\varphi}$$

$$\frac{\mathcal{P},x:\varphi\vdash M:\sigma}{\mathcal{P}\vdash \lambda x^{\varphi} M:\varphi\multimap\sigma}\qquad \frac{\mathcal{P}\vdash M:\varphi\multimap\sigma \qquad \mathcal{P}\vdash N:\varphi}{\mathcal{P}\vdash \langle M\rangle N:\sigma}\qquad \frac{\mathcal{P}\vdash M:!\sigma}{\mathcal{P}\vdash \mathsf{der}(M):\sigma}\qquad \frac{\mathcal{P},x:!\sigma\vdash M:\sigma}{\mathcal{P}\vdash \mathsf{fix}\,x^{!\sigma} M:\sigma}$$

$$\frac{\mathcal{P}\vdash M:\varphi_1\oplus\varphi_2 \qquad \mathcal{P},x_1:\varphi_1\vdash M_1:\sigma \qquad \mathcal{P},x_2:\varphi_2\vdash M_2:\sigma}{\mathcal{P}\vdash \mathsf{case}(M,x_1\cdot M_1,x_2\cdot M_2):\sigma}\qquad \frac{\mathcal{P}\vdash M:\varphi_1\otimes\varphi_2}{\mathcal{P}\vdash \mathsf{pr}_i M:\varphi_i}$$

Figure 1: Typing system for Λ_{HP}

$$\frac{}{\mathsf{der}(M^!)\to_{\mathsf{w}} M}\qquad \frac{}{\langle\lambda x^{\varphi} M\rangle V\to_{\mathsf{w}} M\,[V/x]}\qquad \frac{}{\mathsf{pr}_i\langle V_1,V_2\rangle\to_{\mathsf{w}} V_i}$$

$$\frac{}{\mathsf{fix}\,x^{!\sigma} M\to_{\mathsf{w}} M\left[(\mathsf{fix}\,x^{!\sigma} M)^!/x\right]}\qquad \frac{M\to_{\mathsf{w}} M'}{\mathsf{der}(M)\to_{\mathsf{w}}\mathsf{der}(M')}\qquad \frac{M\to_{\mathsf{w}} M'}{\langle M\rangle N\to_{\mathsf{w}}\langle M'\rangle N}$$

$$\frac{N\to_{\mathsf{w}} N'}{\langle M\rangle N\to_{\mathsf{w}}\langle M\rangle N'}\qquad \frac{M\to_{\mathsf{w}} M'}{\mathsf{pr}_i M\to_{\mathsf{w}}\mathsf{pr}_i M'}\qquad \frac{M_1\to_{\mathsf{w}} M_1'}{\langle M_1,M_2\rangle\to_{\mathsf{w}}\langle M_1',M_2\rangle}$$

$$\frac{M_2\to_{\mathsf{w}} M_2'}{\langle M_1,M_2\rangle\to_{\mathsf{w}}\langle M_1,M_2'\rangle}\qquad \frac{}{\mathsf{case}(\mathsf{in}_i V,x_1\cdot M_1,x_2\cdot M_2)\to_{\mathsf{w}} M_i\,[V/x_i]}\qquad \frac{M\to_{\mathsf{w}} M'}{\mathsf{in}_i M\to_{\mathsf{w}}\mathsf{in}_i M'}$$

$$\frac{M\to_{\mathsf{w}} M'}{\mathsf{case}(M,x_1\cdot M_1,x_2\cdot M_2)\to_{\mathsf{w}}\mathsf{case}(M',x_1\cdot M_1,x_2\cdot M_2)}$$

Figure 2: Weak reduction axioms and rules for Λ_{HP}

We define now a *weak* reduction relation on terms, meaning that we never reduce within a "box" $M^!$ or under a λ. *Values* are particular Λ_{HP} terms (they are not a new syntactic category) defined by the following BNF syntax:

$$V,W\ldots := x \mid M^! \mid \langle V,W\rangle \mid \mathsf{in}_1 V \mid \mathsf{in}_2 V\,.$$

Figure 2 defines weak reduction $\to_{\mathsf{w}}$. It clearly enjoys subject reduction and confluence (actually the diamond property holds).

Proposition 1 *Any value is $\to_{\mathsf{w}}$-normal. If φ is a positive type, $\vdash M:\varphi$ and M is $\to_{\mathsf{w}}$-normal, then M is a value.*

1.1 LL-based denotational semantics, in a nutshell

The kind of denotational models we are interested in in this paper are those induced by a model of LL, as explained in [3]. We record the basic definitions and notations, referring to that paper for more details.

1.1.1 Models of Linear Logic.

A model of LL consists of the following data.

A symmetric monoidal closed category $(\mathcal{L},\otimes,1,\lambda,\rho,\alpha,\sigma)$. We use $X\multimap Y$ for the object of linear morphisms from X to Y, $\mathrm{ev}\in\mathcal{L}((X\multimap Y)\otimes X,Y)$ for the evaluation morphism and cur for the linear curryfication map $\mathcal{L}(Z\otimes X,Y)\to\mathcal{L}(Z,X\multimap Y)$. For convenience, and because it is the case in the concrete models we consider, we assume this SMCC to be a $*$-autonomous category with dualizing object $\bot$. We use $X^{\bot}$ for the object $X\multimap\bot$ of $\mathcal{L}$ (the dual, or linear negation, of X).

$\mathcal{L}$ is cartesian with terminal object $\top$, product $\&$, projections pr_i. By $*$-autonomy $\mathcal{L}$ is cocartesian with initial object 0, coproduct $\oplus$ and injections in_i.

We are given a comonad $!\_:\mathcal{L}\to\mathcal{L}$ with counit $\mathrm{der}_X\in\mathcal{L}(!X,X)$ (*dereliction*) and comultiplication $\mathrm{dig}_X\in\mathcal{L}(!X,!!X)$ (*digging*) together with a strong symmetric monoidal structure (Seely isos m^0 and m^2) for the functor $!\_$, from the symmetric monoidal category $(\mathcal{L},\&)$ to the symmetric monoidal category $(\mathcal{L},\otimes)$ satisfying an additional coherence condition wrt. dig.

We use ?_ for the "De Morgan dual" of !_: $?X = (!(X^\perp))^\perp$ and similarly for morphisms. It is a monad on $\mathcal{L}$.

1.1.2 The Eilenberg-Moore category. It is then standard to define the category $\mathcal{L}^!$ of !-coalgebras. An object of this category is a pair $P = (\underline{P}, \mathrm{h}_P)$ where $\underline{P} \in \mathsf{Obj}(\mathcal{L})$ and $\mathrm{h}_P \in \mathcal{L}(\underline{P}, !\underline{P})$ is such that $\mathrm{der}_{\underline{P}}\, \mathrm{h}_P = \mathrm{Id}$ and $\mathrm{dig}_{\underline{P}}\, \mathrm{h}_P = !\mathrm{h}_P\, \mathrm{h}_P$. Then $f \in \mathcal{L}^!(P, Q)$ iff $f \in \mathcal{L}(\underline{P}, \underline{Q})$ such that $\mathrm{h}_Q\, f = !f\, \mathrm{h}_P$. The functor !_ can be seen as a functor from $\mathcal{L}$ to $\mathcal{L}^!$ mapping X to $(!X, \mathrm{dig}_X)$ and $f \in \mathcal{L}(X, Y)$ to $!f$. It is right adjoint to the forgetful functor $\mathsf{U} : \mathcal{L}^! \to \mathcal{L}$. Given $f \in \mathcal{L}(\underline{P}, X)$, we use $f^! \in \mathcal{L}^!(P, !X)$ for the morphism associated with f by this adjunction, one has $f^! = !f\, \mathrm{h}_P$. If $g \in \mathcal{L}^!(Q, P)$, we have $f^!\, g = (f\, g)^!$.

Then $\mathcal{L}^!$ is cartesian (with product of shape $P \otimes Q = (\underline{P} \otimes \underline{Q}, \mathrm{h}_{P\otimes Q})$ and terminal object $(1, \mathrm{h}_1)$, still denoted as 1). This category is also cocartesian with coproduct of shape $P \oplus Q = (\underline{P} \oplus \underline{Q}, \mathrm{h}_{P\oplus Q})$ and initial object $(0, \mathrm{h}_0)$ still denoted as 0. The complete definitions can be found in [3]. We use $\mathrm{c}_P \in \mathcal{L}^!(P, P \otimes P)$ (*contraction*) for the diagonal and $\mathrm{w}_P \in \mathcal{L}^!(P, 1)$ (*weakening*) for the unique morphism to the terminal object.

1.1.3 Fix-points. For any object X, we assume to be given a morphism $\mathrm{fix}_X \in \mathcal{L}(!(!X \multimap X), X)$ such that[2] $\mathrm{ev}\, (\mathrm{der}_{!X\multimap X} \otimes \mathrm{fix}^!_X) \circ \mathrm{c}_{(!!X\multimap X)}$ which will allow to interpret term fix-points.

In order to interpret fix-points of types, we assume that the category $\mathcal{L}$ is equipped with a notion of embedding-retraction pairs, following a standard approach. We denote as $\mathcal{L}_\subseteq$ for the corresponding category. It is equipped with a functor $\mathsf{F} : \mathcal{L}_\subseteq \to \mathcal{L}^{\mathsf{op}} \times \mathcal{L}$ such that $\mathsf{F}(X) = (X, X)$ and for which we use the notation $(\varphi^-, \varphi^+) = \mathsf{F}(\varphi)$ and assume that $\varphi^-\, \varphi^+ = \mathrm{Id}_X$. We assume furthermore that $\mathcal{L}_\subseteq$ has all countable directed colimits and that the functor $\mathsf{E} = \mathrm{pr}_2\, \mathsf{F} : \mathcal{L}_\subseteq \to \mathcal{L}$ is continuous. We also assume that all the basic operations on objects ($\otimes$, $\oplus$, $(\_)^\perp$ and $!\_$) are continuous functors from $\mathcal{L}_\subseteq$ to itself[3].

Then it is easy to carry this notion of embedding-retraction pairs to $\mathcal{L}^!$, to show that this category has all countable directed colimits and that the functors $\otimes$ and $\oplus$ are continuous on this category. One checks also that !_ define a continuous functor from $\mathcal{L}_\subseteq$ to $\mathcal{L}^!_\subseteq$. This allows to interpret recursive types.

1.1.4 Interpreting types and terms. We simply outline the way types and terms of Λ_{HP} are interpreted in this kind of model. Again, we refer to [3] for precise definitions.

With any positive type φ and any repetition-free list $\vec{\zeta} = (\zeta_1, \dots, \zeta_n)$ of type variables containing all free variables of φ we associate a continuous functor $[\varphi]^!_{\vec{\zeta}} : (\mathcal{L}^!_\subseteq)^n \to \mathcal{L}^!_\subseteq$ and with any general type σ and any list $\vec{\zeta} = (\zeta_1, \dots, \zeta_n)$ of pairwise distinct type variables containing all free variables of σ we associate a continuous functor $[\sigma]_{\vec{\zeta}} : (\mathcal{L}^!_\subseteq)^n \to \mathcal{L}_\subseteq$.

When we write $[\sigma]$ or $[\varphi]^!$ (without subscript), we assume implicitly that the types σ and φ have no free type variables. Then $[\sigma]$ is an object of $\mathcal{L}$ and $[\varphi]^!$ is an object of $\mathcal{L}^!$. We have $[\varphi] = \underline{[\varphi]^!}$ that is, considered as a generalized type, the semantics of a positive type φ is the carrier of the coalgebra $[\varphi]^!$.

Given a typing context $\mathcal{P} = (x_1 : \varphi_1, \dots, x_k : \varphi_k)$, we define $[\mathcal{P}]^! = [\varphi_1]^! \otimes \cdots \otimes [\varphi_k]^! \in \mathcal{L}^!$.

From now on, we assume that the only isos of $\mathcal{L}_\subseteq$ are the identity maps. This implies that the types $\mathsf{Fix}\, \zeta \cdot \varphi$ and $\varphi\, [(\mathsf{Fix}\, \zeta \cdot \varphi)/\zeta]$ are interpreted as the same object (or functor). It will be the case in the models we consider in this paper and in many other models.

Using these categorical structures, given a term M such that $\mathcal{P} \vdash M : \sigma$, one defines a morphism $[M]_\mathcal{P} \in \mathcal{L}([\mathcal{P}], [\sigma])$ in such a way that, if σ is a positive type φ and if M is a value V, then $[V]_\mathcal{P} \in \mathcal{L}^!([\mathcal{P}]^!, [\varphi]^!)$.

One can prove that this interpretation is sound: if $\mathcal{P} \vdash M : \sigma$ and $M \to_{\mathsf{w}} M'$ then $[M]_\mathcal{P} = [M']_\mathcal{P}$. The proof relies on a Substitution Lemma which uses in an essential way the fact that values are interpreted as coalgebra morphisms.

[2] It might seem natural to require the stronger uniformity conditions of *Conway operator* [16]. This does not seem to be necessary as far as soundness of our semantics is concerned even if the fix-point operators arising in concrete models satisfy these further properties.

[3] This is a rough statement; one has to say that *e.g.* that if $\varphi_i \in \mathcal{L}_\subseteq(X_i, Y_i)$ for $i = 1, 2$ then $(\varphi_1 \otimes \varphi_2)^- = \varphi_1^- \otimes \varphi_2^-$ etc. The details can be found in [3].

$$\lambda x^\sigma\, \mathcal{M} = \sum_{i=1}^{k} \lambda x^\sigma\, M_i \quad \langle \mathcal{M} \rangle \mathcal{N} = \sum_{i=1}^{k} \sum_{j=1}^{n} \langle M_i \rangle N_j \quad \mathsf{der}(\mathcal{M}) = \sum_{i=1}^{k} \mathsf{der}(M_i) \quad \mathsf{pr}_l \mathcal{M} = \sum_{i=1}^{k} \mathsf{pr}_l M_i$$

$$\langle \mathcal{M}, \mathcal{N} \rangle = \sum_{i=1}^{k} \sum_{j=1}^{n} \langle M_i, N_j \rangle \quad \mathsf{in}_l \mathcal{M} = \sum_{i=1}^{k} \mathsf{in}_l M_i \quad \mathsf{case}(\mathcal{M}, x_1 \cdot R_1, x_2 \cdot R_2) = \sum_{i=1}^{k} \mathsf{case}(M_i, x_1 \cdot R_1, x_2 \cdot R_2)$$

Figure 3: Non-deterministic extended syntax

$$\frac{}{\mathsf{der}(M^!) \to^1_{\mathsf{m}} M} \qquad \frac{}{\langle \lambda x^\varphi\, M \rangle V \to^1_{\mathsf{m}} M\,[V/x]} \qquad \frac{}{\mathsf{pr}_i \langle V_1, V_2 \rangle \to^1_{\mathsf{m}} V_i}$$

$$\frac{}{\mathsf{fix}\, x^{!\sigma}\, M \to^1_{\mathsf{m}} M\left[(\mathsf{fix}\, x^{!\sigma}\, M)^!/x\right]} \qquad \frac{M \to^1_{\mathsf{m}} \mathcal{M}}{\mathsf{der}(M) \to^1_{\mathsf{m}} \mathsf{der}(\mathcal{M})} \qquad \frac{M \to^1_{\mathsf{m}} \mathcal{M}}{\langle M \rangle N \to^1_{\mathsf{m}} \langle \mathcal{M} \rangle N}$$

$$\frac{N \to^1_{\mathsf{m}} \mathcal{N}}{\langle M \rangle N \to^1_{\mathsf{m}} \langle M \rangle \mathcal{N}} \qquad \frac{M \to^1_{\mathsf{m}} \mathcal{M}}{\mathsf{pr}_i M \to^1_{\mathsf{m}} \mathsf{pr}_i \mathcal{M}} \qquad \frac{M_1 \to^1_{\mathsf{m}} \mathcal{M}_1}{\langle M_1, M_2 \rangle \to^1_{\mathsf{m}} \langle \mathcal{M}_1, M_2 \rangle}$$

$$\frac{M_2 \to^1_{\mathsf{m}} \mathcal{M}_2}{\langle M_1, M_2 \rangle \to^1_{\mathsf{m}} \langle M_1, \mathcal{M}_2 \rangle} \qquad \frac{}{\mathsf{case}(\mathsf{in}_i V, x_1 \cdot M_1, x_2 \cdot M_2) \to^1_{\mathsf{m}} M_i\,[V/x_i]} \qquad \frac{M \to^1_{\mathsf{m}} \mathcal{M}}{\mathsf{in}_i M \to^1_{\mathsf{m}} \mathsf{in}_i \mathcal{M}}$$

$$\frac{M \to^1_{\mathsf{m}} \mathcal{M}}{\mathsf{case}(M, x_1 \cdot M_1, x_2 \cdot M_2) \to^1_{\mathsf{m}} \mathsf{case}(\mathcal{M}, x_1 \cdot M_1, x_2 \cdot M_2)} \qquad \frac{}{\mathsf{choose}(M_1, M_2) \to^1_{\mathsf{m}} M_1 + M_2}$$

Figure 4: Definition of the reduction relation $\to^1_{\mathsf{m}}$ for $\Lambda^{\mathsf{nd}}_{\mathsf{HP}}$

2 Non-deterministic CBPV

The first effect we want to deal with is *non determinism*. We do not extend the typing system, we simply extend the syntax of Λ_{HP} by adding the following construct, where "$\cdots$" stands for the constructions introduced in Section 1.

$$M := \cdots \mid \mathsf{choose}(M_1, M_2)$$

Accordingly, we add one typing rule

$$\frac{\mathcal{P} \vdash M_1 : \sigma \qquad \mathcal{P} \vdash M_2 : \sigma}{\mathcal{P} \vdash \mathsf{choose}(M_1, M_2) : \sigma}$$

2.1 Operational semantics

We use letters $\mathcal{M}, \mathcal{N}, \ldots$ to denote finite multisets of terms. In this context, given a term M, we also denote as M the multiset consisting of 1 copy of M. We use additive notations to deal with these multisets that we consider as finite formal sums of terms.

We extend the term syntax to these multisets: given multisets $\mathcal{M} = \sum_{i=1}^{k} M_i$ and $\mathcal{N} = \sum_{j=1}^{n} N_j$ of terms, we use the *notations* of Figure 3. Using these notations, we define in Figure 4 a rewriting relation $\to^1_{\mathsf{m}}$ from terms to multisets of terms.

We define then a rewriting relation $\to_{\mathsf{m}}$ on multisets of terms.

$$\frac{}{\mathcal{M} \to_{\mathsf{m}} \mathcal{M}} \qquad \frac{M \to^1_{\mathsf{m}} \mathcal{M}}{M \to_{\mathsf{m}} \mathcal{M}} \qquad \frac{\mathcal{M}_1 \to_{\mathsf{m}} \mathcal{M}'_1 \qquad \mathcal{M}_2 \to_{\mathsf{m}} \mathcal{M}'_2}{\mathcal{M}_1 + \mathcal{M}_2 \to_{\mathsf{m}} \mathcal{M}'_1 + \mathcal{M}'_2}$$

The usual non-deterministic reduction $\to_{\mathsf{nd}}$ can now be defined as follows: $M \to_{\mathsf{nd}} M'$ means that $M \to_{\mathsf{m}} M' + \mathcal{M}$ for some multiset of terms $\mathcal{M}$ (observe that $M \to_{\mathsf{m}} M' + \mathcal{M}$ iff $M \to^1_{\mathsf{m}} M' + \mathcal{M}$). It is clear that $M \to^*_{\mathsf{nd}} M'$ iff $M \to^*_{\mathsf{m}} M' + \mathcal{M}$ for some multiset of terms $\mathcal{M}$.

Given a multiset of terms $\mathcal{M} = \sum_{i=1}^{k} M_i$, we write $\mathcal{P} \vdash \mathcal{M} : \sigma$ if we have $\mathcal{P} \vdash M_i : \sigma$ for $i = 1, \ldots, k$. Then Subject Reduction extends easily to this non-deterministic setting: if $\mathcal{P} \vdash \mathcal{M} : \sigma$ and $\mathcal{M} \to_{\mathsf{m}} \mathcal{M}'$ then $\mathcal{P} \vdash \mathcal{M}' : \sigma$.

Let M be such that $\mathcal{P} \vdash M : \varphi$ for some positive type φ. We say that M converges (notation $M \downarrow_{\mathsf{nd}}$) if $M \to_{\mathsf{m}}^{*} V + \mathcal{M}$ for some value V and some multiset of terms $\mathcal{M}$.

2.2 Scott semantics

Usually, in a model $\mathcal{L}$ of LL, an object X of $\mathcal{L}$ can be endowed with several different structures of !-coalgebras which makes the category $\mathcal{L}^!$ difficult to describe simply (in contrast with the Kleisli category used for interpreting PCF; its objects are those of $\mathcal{L}$). In the Scott model of LL however, every object of the linear category has exactly one structure of !-coalgebra as we shall see. This is a distinctive feature of this model. A nice outcome of these observations will be a very simple intersection typing system.

Remark: As explained in [1], this semantics of LL has been independently discovered by several authors [7, 18]. The same model is also considered in [12] Section 5.5.4 to interpret may non-determinism in CBPV. This is essentially this model that we are considering here.

2.2.1 Objects and morphisms. We introduce a "linear" category **Polr** of preorders and relations. A preorder is a pair $X = (|X|, \leq_X)$ where $|X|$ is a countable set and $\leq_X$ is a preorder relation on $|X|$. A morphism from X to Y is a $f \subseteq |X| \times |Y|$ such that, if $(a, b) \in f$ and $a \leq_X a'$ and $b' \leq_Y b$, then $(a', b') \in f$. The relational composition of two morphisms is still a morphism and the identity morphism at X is $\mathrm{Id}_X = \{(a, a') \mid a' \leq_X a\}$.

Given an object X in **Polr**, the set $\mathsf{Ini}(X)$ of downwards closed subsets of $|X|$, ordered by inclusion, is a complete lattice which is ω-prime-algebraic (and all such lattices are of that shape up to iso). **Polr** is equivalent to the category of ω-prime algebraic complete lattices and linear maps (functions preserving all lubs).

2.2.2 Monoidal structure and cartesian product. The tensor unit 1 is $(\{*\}, =)$ and $X \otimes Y = (|X| \times |Y|, \leq_X \times \leq_Y)$. The tensor product of morphisms is defined in the obvious way, as well as the coherence isos. Then one defines $X \multimap Y$ by $|X \multimap Y| = |X| \times |Y|$ and $(a', b') \leq_{X \multimap Y} (a, b))$ if $a \leq a'$ and $b' \leq b$. The linear evaluation morphism $\mathrm{ev} \in \mathbf{Polr}((X \multimap Y) \otimes X, Y)$ is given by $\mathrm{ev} = \{(((a', b), a), b') \mid b' \leq b \text{ and } a' \leq a\}$. If $f \in \mathbf{Polr}(Z \otimes X, Y)$ then $\mathrm{cur}(f) \in \mathbf{Polr}(Z, X \multimap Y)$ is defined by moving parentheses. This shows that **Polr** is closed. It is *-autonomous, with $\bot = 1$ as dualizing object. Observe that $X^{\bot}$ is simply $|X|$ equipped with $\geq_X$ as preorder relation $\leq_{X^{\bot}}$.

Given a countable family of objects $(X_i)_{i \in I}$, the cartesian product X is defined by $|X| = \bigcup_{i \in I} \{i\} \times |X_i|$ with $(i, a) \leq (j, b)$ if $i = j$ and $a \leq b$. Projections are defined by $\mathrm{pr}_i = \{((i, a), a') \mid a' \leq a\}$. Tupling of morphisms is defined as in **Rel**. Coproducts are defined similarly.

2.2.3 Exponential. One sets $!X = (\mathcal{P}_{\mathrm{fin}}(|X|), \leq)$ with $u \leq u'$ if $\forall a \in u \, \exists a' \in u' \; a \leq_X a'$ (where $\mathcal{P}_{\mathrm{fin}}(E)$ is the set of all finite subsets of E). Given $f \in \mathbf{Polr}(X, Y)$, one defines $!f$ as $\{(u, v) \in |!X| \times |!Y| \mid \forall b \in v \, \exists a \in u \; (a, b) \in f\}$. This defines a functor $\mathbf{Polr} \to \mathbf{Polr}$. Then one sets $\mathrm{der}_X = \{(u, a) \mid \exists a' \in u \; a \leq a'\} \in \mathbf{Polr}(!X, X)$ and $\mathrm{dig}_X = \{(u, \{u_1, \ldots, u_n\}) \mid u_1 \cup \cdots \cup u_n \leq_{!X} u\} \in \mathbf{Polr}(!X, !!X)$. This defines a comonad $\mathbf{Polr} \to \mathbf{Polr}$. The Seely isos are given by $\mathrm{m}^0 = \{(*, \emptyset)\} \in \mathbf{Polr}(1, !\top)$ and $\mathrm{m}^2_{X,Y} = \{((u, v), w) \mid w \leq_{!(X \& Y)} \{1\} \times u \cup \{2\} \times v\} \in \mathbf{Polr}(!X \otimes !Y, !(X \,\&\, Y))$.

Each X has a fix-point operator $\mathrm{fix}_X \in \mathbf{Polr}(!(!X \multimap X), X)$ which is defined as a least fix-point: $\mathrm{fix}_X = \{(w, a) \mid \exists (u', a') \in w \; a \leq a' \text{ and } \forall a'' \in u' \; (w, a'') \in \mathrm{fix}_X\}$.

2.3 The category of !-coalgebras

The first main observation is that each object of **Polr** has exactly one structure of !-coalgebra. Proofs: see [3].

Theorem 2 *Let X be an object of* **Polr**. *Then (X, p_X) is a !-coalgebra, where $\mathrm{p}_X = \{(a, u) \in |X| \times |!X| \mid \forall a' \in u \; a' \leq a\}$. Moreover, if P is a !-coalgebra, then $\mathrm{h}_P = \mathrm{p}_{\underline{P}}$.*

2.3.1 Morphisms of !-coalgebras. So we consider the objects of $\mathbf{Polr}^!$ as simple preorders. When we consider X as an object of $\mathbf{Polr}^!$, we always mean the object (X, p_X) described above.

With any preorder X we associate an ω-algebraic cpo $\mathsf{Idl}(X)$ which is its ideal completion: an element of $\mathsf{Idl}(X)$ is a $\xi \subseteq |X|$ such that $\xi \neq \emptyset$, ξ is downwards closed and directed. We equip $\mathsf{Idl}(X)$ with the inclusion partial order relation.

Lemma 3 *The poset $\mathsf{Idl}(X)$ is a cpo which has countably many compact elements. For any $\xi \in \mathsf{Idl}(X)$, the set of compact elements $\xi_0 \in \mathsf{Idl}(X)$ such that $\xi_0 \subseteq \xi$ is directed and ξ is the lub of that set. In general, $\mathsf{Idl}(X)$ has no minimum element.*

Theorem 4 *There is a bijective and functorial correspondence between $\mathbf{Polr}^!(X, Y)$ and the set of Scott continuous functions from $\mathsf{Idl}(X) \to \mathsf{Idl}(Y)$. This correspondence is an order isomorphism when Scott-continuous functions are equipped with the pointwise order and $\mathbf{Polr}^!(S, T)$ is equipped with inclusion.*

Let **Predom** be the category whose objects are the preorders and where a morphism from X to Y is a Scott continuous function from $\mathsf{Idl}(X)$ to $\mathsf{Idl}(Y)$. We have seen that $\mathbf{Polr}^!$ and **Predom** are equivalent categories (isomorphic indeed). It is easy to retrieve directly the fact that **Predom** has products and sums: the product of X and Y is $X \otimes Y$ (and indeed, it is easy to check that $\mathsf{Idl}(X \otimes Y) \simeq \mathsf{Idl}(X) \times \mathsf{Idl}(Y)$) and their sum is $X \oplus Y$ and indeed $\mathsf{Idl}(X \oplus Y) \simeq \mathsf{Idl}(X) + \mathsf{Idl}(Y)$, the disjoint union of the predomains $\mathsf{Idl}(X)$ and $\mathsf{Idl}(Y)$. This predomain has no minimum element as soon as $|X|$ and $|Y|$ are non-empty. Observe also that $\mathsf{Idl}(!X) = \mathsf{Ini}(X)$: one retrieves the fact that the Kleisli category of the ! comonad is the category of preorders and Scott continuous functions between the associated lattices[4].

2.3.2 Inclusions and embedding-retraction pairs. We define a category $\mathbf{Polr}_\subseteq$ as follows: the objects are those of **Polr** and $\mathbf{Polr}_\subseteq(X, Y)$ is a singleton $\{\varphi_{X,Y}\}$ if $|X| \subseteq |Y|$ and $\forall a, a' \in |X|\ a \leq_X a' \Leftrightarrow a \leq_Y a'$ (and then we write $X \subseteq Y$) and is empty otherwise. So $\mathbf{Polr}_\subseteq$ is a partially ordered class.

The functor $\mathsf{F} : \mathbf{Polr}_\subseteq \to \mathbf{Polr}^{\mathsf{op}} \times \mathbf{Polr}$ is defined as follows: if $X \subseteq Y$ then $\varphi^+_{X,Y} = \{(a, b) \in |X| \times |Y| \mid b \leq_Y a\}$ and $\varphi^-_{X,Y} = \{(b, a) \in |Y| \times |X| \mid a \leq_Y b\}$; this definition is functorial and $\varphi^-_{X,Y}\, \varphi^+_{X,Y} = \mathrm{Id}_X$. The partially ordered class $\mathbf{Polr}_\subseteq$ is complete in the sense that any countable directed family of objects. $(X_i)_{i\in J}$ has a lub X given by $|X| = \cup_{i\in J}|X_i|$ and $a \leq_X a'$ if $a \leq_{X_i} a'$ for some i; we denote this preorder as $\cup_{i\in J} X_i$. The operations $\otimes$, $\oplus$ and $!\_$ are monotone and Scott-continuous operations on this partially ordered class.

2.3.3 Interpreting types and terms. As explained in Section 1.1.4, with any closed type σ we associate an object $[\sigma]$ of **Polr** and with any positive type φ we associate an object $[\varphi]^!$ of $\mathbf{Polr}^!$ (of which Theorem 4 offers a simple presentation). With any typing context $\mathcal{P} = (x_1 : \varphi_1, \ldots, x_k : \varphi_k)$ we associate an object $[\mathcal{P}]^! = [\varphi_1]^! \otimes \cdots \otimes [\varphi_k]^!$ of $\mathbf{Polr}^!$.

Given a term M such that $\mathcal{P} \vdash M : \sigma$ we associate a morphism $[M]_\mathcal{P} \in \mathbf{Polr}([\mathcal{P}], [\sigma])$. If σ is a positive type φ and if M is a value V, we have moreover $[V]_\mathcal{P} \in \mathbf{Polr}^!([\mathcal{P}]^!, [\varphi]^!)$ (this hom-set indeed is a subset of $\mathbf{Polr}([\mathcal{P}], [\varphi])$).

This interpretation extends easily to our non-deterministic primitive: if $\mathcal{P} \vdash M_i : \sigma$ for $i = 1, 2$ then we set

$$[\mathsf{choose}(M_1, M_2)]_\mathcal{P} = [M_1]_\mathcal{P} \cup [M_2]_\mathcal{P} \in \mathbf{Polr}([\mathcal{P}], [\sigma])\,.$$

Remark: It is crucial to observe that if $\mathcal{P} \vdash V_i : \varphi$ for $i = 1, 2$ and the V_i's are values, then we have $\mathcal{P} \vdash \mathsf{choose}(V_1, V_2) : \varphi$ but this term $\mathsf{choose}(V_1, V_2)$ is *not a value*. And indeed, if $f_1, f_2 \in \mathbf{Polr}^!([\mathcal{P}]^!, [\varphi]^!)$ (which means that the f_i's are continuous maps $\mathsf{Idl}([\mathcal{P}]) \to \mathsf{Idl}([\varphi])$) then $f_1 \cup f_2$ is not a coalgebra morphism in general simply because ideals are not closed under unions.

We extend this semantics straightforwardly to multisets of terms: given terms $(M_i)_{i=1}^k$ such that $\mathcal{P} \vdash M_i : \sigma$ for each i, we set $[\sum_{i=1}^k M_i]_\mathcal{P} = \bigcup_{i=1}^k [M_i]_\mathcal{P}$. Then we can express the soundness of our calculus.

[4] We retrieve as an Eilenberg-Moore category of a model of LL the cartesian and cocartesian category of predomains and continuous functions singled out in [17] as a canonical category on top of which a continuation semantics for classical lambda calculi can be built. This is perfectly compatible with [9] where it is shown that this continuation semantics can be performed on top of any Eilenberg-Moore category of a model of LL.

$$\frac{a' \leq_{[\varphi]} a}{\Phi, x : a : \varphi \vdash x : a' : \varphi} \qquad \frac{u \in \mathcal{P}_{\mathrm{fin}}(|[\sigma]|) \qquad \forall a \in u \quad \Phi \vdash M : a : \sigma}{\Phi \vdash M^! : u : !\sigma}$$

$$\frac{\Phi \vdash M_1 : a_1 : \varphi_1 \qquad \Phi \vdash M_2 : a_2 : \varphi_2}{\Phi \vdash \langle M_1, M_2 \rangle : (a_1, a_2) : \varphi_1 \otimes \varphi_2} \qquad \frac{\Phi \vdash M : \{a\} : !\sigma}{\Phi \vdash \mathsf{der}(M) : a : \sigma} \qquad \frac{\Phi \vdash M : a : \varphi_i}{\Phi \vdash \mathsf{in}_i M : (i, a) : \varphi_1 \oplus \varphi_2}$$

$$\frac{\Phi \vdash M : (a, b) : \varphi \multimap \sigma \qquad \Phi \vdash N : a : \varphi}{\Phi \vdash \langle M \rangle N : b : \sigma}$$

$$\frac{\Phi \vdash M : (1, a) : \varphi_1 \oplus \varphi_2 \qquad \Phi, x_1 : a : \varphi_1 \vdash N_1 : b : \sigma \qquad \underline{\Phi}, x_2 : \varphi_2 \vdash N_2 : \sigma}{\Phi \vdash \mathsf{case}(M, x_1 \cdot N_1, x_2 \cdot N_2) : b : \sigma}$$

$$\frac{\Phi \vdash M : (2, a) : \varphi_1 \oplus \varphi_2 \qquad \underline{\Phi}, x_1 : \varphi_1 \vdash N_1 : \sigma \qquad \Phi, x_2 : a : \varphi_2 \vdash N_2 : b : \sigma}{\Phi \vdash \mathsf{case}(M, x_1 \cdot N_1, x_2 \cdot N_2) : b : \sigma}$$

$$\frac{\Phi \vdash M : (a_1, a_2) : \varphi_1 \otimes \varphi_2}{\Phi \vdash \mathsf{pr}_i M : a_i : \varphi_i}$$

$$\frac{\Phi, x : u : !\sigma \vdash M : a : \sigma \qquad \forall b \in u \ \Phi \vdash \mathsf{fix}\, x^{!\sigma}\, M : b : \sigma}{\Phi \vdash \mathsf{fix}\, x^{!\sigma}\, M : a : \sigma} \qquad \frac{\Phi \vdash M_i : a : \sigma \qquad \underline{\Phi} \vdash M_{2-i} : \sigma}{\Phi \vdash \mathsf{choose}(M_1, M_2) : a : \sigma}$$

Figure 5: Scott Semantics as a Typing System

Theorem 5 *If $\mathcal{P} \vdash \mathcal{M} : \sigma$ and $\mathcal{M} \to_{\mathsf{m}}^* \mathcal{M}'$ then $[\mathcal{M}]_{\mathcal{P}} = [\mathcal{M}']_{\mathcal{P}}$.*

It is interesting to present this Scott semantics of terms as a typing system, in the spirit of Coppo-Dezani Intersection Types, see [8]. A *semantic context* is a sequence $\Phi = (x_1 : a_1 : \varphi_1, \ldots, x_n : a_n : \varphi_n)$ where $a_i \in [\varphi_i]$ for each i, its *underlying typing context* is $\underline{\Phi} = (x_1 : \varphi_1, \ldots, x_n : \varphi_n)$ and its *underlying tuple* is $\langle \Phi \rangle = (a_1, \ldots, a_k) \in [\underline{\Phi}]$. The typing rules are given in Figure 5, the intended meaning of these rules is made clear in Proposition 7.

A simple induction on typing derivation trees shows that this typing system is "monotone" as usually for intersection type systems. We write $\Phi \leq \Phi'$ if $\Phi = (x_1 : a_1 : \varphi_1, \ldots, x_n : a_n : \varphi_n)$, $\Phi' = (x_1 : a'_1 : \varphi_1, \ldots, x_n : a'_n : \varphi_n)$ and $a_i \leq_{[\varphi_i]} a'_i$ for $i = 1, \ldots, n$.

Proposition 6 *If $\Phi \vdash M : a : \sigma$, $a' \leq_{[\sigma]} a$ and $\Phi \leq \Phi'$ then $\Phi' \vdash M : a' : \sigma$.*

Using this property, one can prove that this deduction system describes exactly the Scott denotational semantics of Λ_{HP}.

Proposition 7 *Given $a_1 \in [\varphi_1], \ldots, a_n \in [\varphi_n]$ and $a \in [\sigma]$, one has $(a_1, \ldots, a_n, a) \in [M]_{x_1 : \sigma_n, \ldots, x_1 : \sigma_n}$ iff $x_1 : a_1 : \varphi_1, \ldots, x_n : a_n : \varphi_n \vdash M : a : \sigma$.*

The proof also uses crucially the fact that all structural operations (weakening, contraction, dereliction, promotion) admit a very simple description in terms of the preorder relation on objects by Theorem 2; for instance the contraction morphism of X (seen as an object of $\mathbf{Polr}^!$) is $\mathrm{c}_X = \{(a, (a_1, a_2)) \mid a_i \leq_X a \text{ for } i = 1, 2\}$.

2.3.4 Adequacy. Our goal now is to prove that, if a closed term M of positive type φ has a non-empty interpretation, that is, if there is $a \in |[\varphi]|$ such that $\vdash M : a : \varphi$, then the reduction $\to_{\mathsf{m}}$ starting from M terminates in the sense that one of the "branches" of the non-deterministic reduction of M reaches a value.

We use a semantic method adapted from the presentation of the *reducibility* method in [8].

Given a type σ and an $a \in [\sigma]$, we define a set $|a|^\sigma$ of terms M such that $\vdash M : \sigma$ (so these terms are all closed). The definition is by induction on the structure of the point a.

Given a positive type φ and $a \in [\varphi]$, we define $|a|^\varphi_{\mathsf{v}}$ as a set of *closed values* V such that $\vdash V : \varphi$ and given a general type σ and $a \in [\sigma]$, we define $|a|^\sigma$ as a set of closed terms M such that $\vdash M : \sigma$. The

$$
\begin{aligned}
|u|_{\mathsf{v}}^{!\sigma} &= \{N^! \mid N \in \bigcap_{a\in u} |a|^\sigma\}\\
|(a_1,a_2)|_{\mathsf{v}}^{\varphi_1\otimes\varphi_2} &= \{\langle V_1,V_2\rangle \mid V_i \in |a_i|_{\mathsf{v}}^{\varphi_i} \text{ for } i=1,2\}\\
|(i,a)|_{\mathsf{v}}^{\varphi_1\oplus\varphi_2} &= \{\mathsf{in}_i V \mid V \in |a|_{\mathsf{v}}^{\varphi_i}\}\\
|a|^{\varphi} &= \{M \mid \vdash M:\varphi \text{ and } \exists V \in |a|_{\mathsf{v}}^{\varphi}\ M \to_{\mathsf{nd}}^* V\}\\
|(a,b)|^{\varphi\multimap\sigma} &= \{M \mid \vdash M:\varphi\multimap\sigma \text{ and } \forall V \in |a|_{\mathsf{v}}^{\varphi}\ \langle M\rangle V \in |b|^{\sigma}\}
\end{aligned}
$$

Figure 6: Interpretation of points as sets of terms in Λ_{HP}

definitions are by mutual induction and are given in Figure 6. Observe that if $\vdash V:\varphi$ and if $a\in[\varphi]$, the statements $V\in|a|^\varphi$ and $V\in|a|_{\mathsf{v}}^\varphi$ are equivalent because V is $\to_{\mathsf{m}}$-normal.

Lemma 8 *Assume that $\vdash M:\sigma$. If $M\to_{\mathsf{nd}}^* M'$ and $M'\in|a|^\sigma$ then $M\in|a|^\sigma$.*

Proof. By induction on the structure of a. If $\sigma=\varphi$, the property follows readily from the definition. Assume that $\sigma=\varphi\multimap\tau$ and $a=(b,c)$. Assume that $M\to_{\mathsf{m}} M'+\mathcal{M}$ with $M'\in|a|^\sigma$. Let $V\in|b|_{\mathsf{v}}^\varphi$, we have $\langle M\rangle V\to_{\mathsf{m}}\langle M'\rangle V+\langle\mathcal{M}\rangle V$ and $\langle M'\rangle V\in|c|^\tau$ by definition of $|\varphi\multimap\tau|^{(b,c)}$. The announced property follows by inductive hypothesis. □

Lemma 9 *Let σ be a type and let $a,a'\in|[\sigma]|$ be such that $a\leq_{[\sigma]}a'$. Then $|a|^\sigma\supseteq|a'|^\sigma$. If σ is positive, we have $|a|_{\mathsf{v}}^\sigma\supseteq|a'|_{\mathsf{v}}^\sigma$*

Theorem 10 (Interpretation Lemma) *Let $\Phi=(x_1:a_1:\varphi_1,\dots,x_k:a_k:\varphi_k)$ and assume that $\Phi\vdash M:a:\sigma$. Then for any family of closed values $(V_i)_{i=1}^k$ such that $V_i\in|a_i|^{\varphi_i}$ one has*

$$M\,[V_1/x_1,\dots,V_k/x_k]\in|a|^\sigma\,.$$

Proof. By induction on the Scott typing derivation of M. Let V_i be values such that $V_i\in|a_i|^{\varphi_i}$ for $i=1,\dots,k$. For any term R, we use R' for $R\,[V_1/x_1,\dots,V_k/x_k]$. This proof uses the definition of $\to_{\mathsf{nd}}$ in Section 2.1. The basic notion of reduction in this non-deterministic context is $\to_{\mathsf{m}}$ defined in the same section, the reduction $\to_{\mathsf{nd}}$ is only a derived notion, but a convenient one. We deal only with a few cases.

Assume that $M=N^!$ with $\sigma=!\tau$, $a=u\in\mathcal{P}_{\mathrm{fin}}(|[\tau]|)$, $\Phi\vdash N:b:\tau$ for each $b\in u$. By inductive hypothesis, we have $N'\in\bigcap_{b\in u}|b|^\tau$. Since $M'=N'^!$, and hence $M'\to_{\mathsf{nd}}^* N'^!$ in 0 steps, the announced property holds.

Assume that $M=\mathsf{case}(N,x_1\cdot N_1,x_2\cdot N_2)$ with $\Phi\vdash N:(1,b):\varphi_1\oplus\varphi_2$ and $\Phi,x_1:b:\varphi_1\vdash N_1:a:\sigma$ (and also $\underline{\Phi},x_2:\varphi_2\vdash N_2:\sigma$). By inductive hypothesis we have $N'\in|(1,b)|^{\varphi_1\oplus\varphi_2}$. This means that there is $V\in|b|_{\mathsf{v}}^{\varphi_1}$ such that $N'\to_{\mathsf{nd}}^*\mathsf{in}_1V$. Therefore we have $M'=\mathsf{case}(N',x_1\cdot N_1',x_2\cdot N_2')\to_{\mathsf{nd}}^*\mathsf{case}(\mathsf{in}_1V,x_1\cdot N_1',x_2\cdot N_2')\to_{\mathsf{nd}}N_1'\,[V/x_1]$. By inductive hypothesis applied to N_1, and because $V\in|b|_{\mathsf{v}}^{\varphi_1}$, we have $N_1'\,[V/x_1]\in|a|^\sigma$ and hence $M'\in|a|^\sigma$ as expected.

Assume that $M=\langle N\rangle R$ with $\Phi\vdash N:(b,a):\varphi\multimap\sigma$ and $\Phi\vdash R:b:\varphi$. By inductive hypothesis we have $N'\in|(b,a)|^{\varphi\multimap\sigma}$ and $R'\in|b|^\varphi$. Therefore there is $V\in|b|_{\mathsf{v}}^\varphi$ such that $R'\to_{\mathsf{nd}}^*V$. Hence $M'=\langle N'\rangle R'\to_{\mathsf{nd}}^*\langle N'\rangle V\in|a|^\sigma$ by definition of $|(b,a)|^{\varphi\multimap\sigma}$ and hence $M'\in|a|^\sigma$ by Lemma 8.

Assume that $M=\lambda x^\varphi N$ with $\sigma=\varphi\multimap\tau$, $a=(b,c)$ and $\Phi,x:b:\varphi\vdash N:c:\tau$. We must prove that $\lambda x^\varphi N'\in|(b,c)|^{\varphi\multimap\tau}$. So let $V\in|b|_{\mathsf{v}}^\varphi$, we must check that $\langle\lambda x^\varphi N'\rangle V\in|c|^\tau$ which results from the fact that $\langle\lambda x^\varphi N'\rangle V\to_{\mathsf{nd}}N'\,[V/x]\in|c|^\tau$ by inductive hypothesis and from Lemma 8.

Assume last that $M=\mathsf{choose}(M_1,M_2)$ with $\Phi\vdash M_1:a:\varphi$ and $\underline{\Phi}\vdash M_2:\sigma$. By inductive hypothesis we have $M_1'\in|a|^\varphi$ and hence by Lemma 8 $M'\in|a|^\varphi$ since, clearly, $M'=\mathsf{choose}(M_1',M_2')\to_{\mathsf{nd}}M_1'$. □

Let $1=!\top$, which is a positive type. One has $\vdash(\Omega^\top)^!:1$. This type is similar to the type unit of ML and $(\Omega^\top)^!$ is similar to $()$, we use it as our basic type of observations (what we observe is termination). We have $[1]=1$ (up to trivial iso) and $[(\Omega^\top)^!]=\{\emptyset\}$. If $\vdash M:\varphi$ and $[M]\neq\emptyset$ we have $M\to_{\mathsf{w}}^*V$ for a value V by Theorem 10.

Let us say that two closed terms M_1, M_2 such that $\vdash M_i:\sigma$ for $i=1,2$ are observationally equivalent, notation $M_1\sim M_2$, if for all closed term C of type $!\sigma\multimap 1$, $\langle C\rangle M_1^!\to_{\mathsf{w}}^*()$ iff $\langle C\rangle M_2^!\to_{\mathsf{w}}^*()$.

Theorem 11 (Adequacy) *If* $\vdash M_i : \sigma$ *for* $i = 1, 2$ *satisfy* $[M_1] = [M_2]$ *then* $M_1 \sim M_2$.

Easy consequence of Theorems 10 and 5.

Another important consequence of Theorem 10 is a "completeness" property of the weak reduction of Figure 2 which can be stated as follows, for the language Λ_{HP} (it is not completely clear what the analogue statement for the non-deterministic extension of Λ_{HP} should be). Let $\simeq$ be an equivalence relation on terms which contains $\to_{\mathsf{w}}^*$ and is compatible with the semantics of Section 1.1.4 in the sense that, if $\mathcal{P} \vdash M_i : \sigma$ for $i = 1, 2$ and $M_1 \simeq M_2$, then $[M_1]_{\mathcal{P}} = [M_2]_{\mathcal{P}}$ in any model $\mathcal{L}$. Assume that $\vdash M : \varphi$ and $\vdash V : \varphi$ for a closed term M and a closed value V, and assume that $M \simeq V$. Then $M \to_{\mathsf{w}}^* V'$ where V' is a value such that $V \simeq V'$.

2.4 Full Abstraction

We prove now the converse of Theorem 11. For this purpose we associate terms with points of the model. The proof use a technique similar to the one developed in [15]. More precisely:

- Given a positive type φ and $a \in |[\varphi]|$, we define a term a^0 such that $\vdash a^0 : \varphi \multimap 1$.
- Given a general type σ and $a \in |[\sigma]|$, we define terms a^- such that $\vdash a^- : !\sigma \multimap 1$ and a^+ such that $\vdash a^+ : \sigma$. Moreover if σ is positive, the term a^+ is a value.

We give the definition of these terms, by induction on the structure of the point a. For this we use the term conj_φ such that $\vdash \mathsf{conj} : 1 \multimap \varphi \multimap \varphi$ defined by $\mathsf{conj} = \lambda x^1\, \lambda y^\varphi\, y$. It is easy to check that $[\mathsf{conj}_\varphi] = \{(\emptyset, (a, a')) \mid a' \leq_{[\varphi]} a\}$. Given terms M_i such that $\vdash M_i : 1$ for $i = 1, \dots, k$, we use the notation $M_1 \wedge \dots \wedge M_k$ for the term $M = \langle \mathsf{conj}_1 \rangle M_1 \cdots \langle \mathsf{conj}_1 \rangle M_{k-1} M_k$ so that $\vdash M : 1$. For $k = 0$, we set $M = () = (\Omega^\top)^!$.

For terms $M_1, \dots, M_k$ such that $\vdash M_i : \sigma$ for $i = 1, \dots, k$, we set

$$M_1 \vee \dots \vee M_k = \mathsf{choose}(M_1, \dots, \mathsf{choose}(M_{k-1}, M_k))\,.$$

If $\varphi = !\sigma$ and $a = [b_1, \dots, b_k]$ with $b_i \in |[\sigma]|$. By inductive hypothesis, we have terms $\vdash b_i^- : \varphi \multimap 1$ and $\vdash b_i^+ : \sigma$. Then we set

$$a^0 = \lambda x^\varphi\, \langle b_1^- \rangle x \wedge \dots \wedge \langle b_k^- \rangle x \quad a^+ = (b_1^+ \vee \dots \vee b_k^+)^!\,.$$

If $\varphi = \varphi_1 \otimes \varphi_2$ and $a = (b_1, b_2)$ with $b_i \in |[\varphi_i]|$ for $i = 1, 2$, then we set

$$a^0 = \lambda x^{\varphi_1 \otimes \varphi_2}\, \langle b_1^0 \rangle \mathsf{pr}_1 x \wedge \langle b_2^0 \rangle \mathsf{pr}_2 x \quad a^+ = \langle b_1^+, b_2^+ \rangle\,.$$

If $\varphi = \varphi_1 \oplus \varphi_2$ and $a = (1, b)$ with $b \in |[\varphi_1]|$ then we set

$$a^0 = \lambda x^{\varphi_1 \oplus \varphi_2}\, \mathsf{case}(x, y_1 \cdot \langle b^0 \rangle y_1, y_2 \cdot \Omega^1) \quad a^+ = \mathsf{in}_1 b^+$$

and when $a = (2, b)$ the definition is similar.

For a general type σ and $a \in |[\sigma]|$, we define now a^- and a^+. If $\sigma = \varphi$ is positive, then we have defined a^0 and we set

$$a^- = \lambda x^{!\varphi}\, \langle a^0 \rangle \mathsf{der}(x)$$

and a^+ is already defined. If $\sigma = \varphi \multimap \tau$ and $a = (b, c)$ with $b \in |[\varphi]|$ and $c \in |[\tau]|$ then we set

$$a^- = \lambda f^{!(\varphi \multimap \tau)}\, \langle c^- \rangle (\langle \mathsf{der}(f) \rangle b^+)^!$$
$$a^+ = \lambda x^\varphi\, \mathsf{der}(\mathsf{conj}_{!\tau}(\langle b^0 \rangle x, (c^+)^!))\,.$$

It is easy to check that these terms satisfy the announced typing judgments. It is also clear that the term a^+ is always a value.

Given an object X of $\mathbf{Polr}$, an element u of $\mathsf{Ini}(X)$ is the same thing as a morphism in $\mathbf{Polr}(1, X)$ and an element of $\mathsf{Idl}(X)$ is the same thing as a morphism in $\mathbf{Polr}^!(1, X)$ (X being equipped with its unique structure of !-coalgebra). Remember also that morphisms $f \in \mathbf{Polr}(X, Y)$ are in bijective correspondence with linear functions $\mathsf{Ini}(X) \to \mathsf{Ini}(Y)$; the function associated with f is defined by $f(u) = \{b \mid \exists a\ (a, b) \in f\}$.

Proposition 12 *Let* σ *be a type and* $a \in |[\sigma]|$, *then* $[a^+] = \downarrow_{[\sigma]} a$ *and, given* $u \in \mathsf{Ini}([\sigma])$, *one has* $[a^-](u^!) \neq 0$ *iff* $a \in u$. *If* σ *is a positive type* φ *and if* $u \in \mathsf{Idl}([\varphi])$, *then* $[a^0](u) \neq \emptyset$ *iff* $a \in u$.

Proof. By mutual induction on the structure of a. If $\varphi = !\sigma$ and $a = \{b_1, \dots, b_k\}$, we have

$$\begin{aligned}[a^+] &= [(b_1^+ \vee \cdots \vee b_k^+)^!] = ([b_1^+] \cup \cdots \cup [b_k^+])^! \\ &= \mathcal{P}_{\mathrm{fin}}((\underset{[\sigma]}{\downarrow} b_1 \cup \cdots \cup \underset{[\sigma]}{\downarrow} b_k)) = \underset{![\sigma]}{\downarrow} \{b_1, \dots, b_k\}\,.\end{aligned}$$

Let $u \in \mathsf{Idl}([\varphi])$, that is $u \in \mathbf{Polr}^!(1, [\varphi]^!)$, then

$$[a^0](u) = [b_1^-](u) \cap \cdots \cap [b_k^-](u)$$

because u commutes with contraction and weakening on $[\varphi]$, as a morphism of $\mathbf{Polr}^!$. For the same reason we know that $u = v^!$ where $v = \mathrm{der}_{[\sigma]}(u) = \bigcup u$ (using the definition of der in $\mathbf{Polr}$ and the downwards closure of u). By inductive hypothesis it follows that $[a^0](u) \neq \emptyset$ holds iff $\forall i\ b_i \in v$, that is $\forall i\, \exists a_i \in u\ b_i \in a_i$ and since u is an ideal for the preorder of $![\sigma]$ this latter condition is equivalent to $\{b_1, \dots, b_k\} \in u$, that is $a \in u$ as contended.

If $\varphi = \varphi_1 \otimes \varphi_2$ and $a = (a_1, a_2)$ with $a_i \in |[\varphi_i]|$, then

$$[a^+] = [\langle a_1^+, a_2^+\rangle] = [a_1^+] \times [a_2^+] = \underset{[\varphi_2]}{\downarrow} a_1 \times \underset{[\varphi_1]}{\downarrow} a_2 = \underset{[\varphi]}{\downarrow} a\,.$$

If $u \in \mathsf{Idl}([\varphi])$, then $u = u_1 \times u_2$ where $u_i = \mathrm{pr}_i(u) \in \mathsf{Idl}([\varphi_i])$ for $i = 1, 2$ because u is an ideal. Therefore

$$\begin{aligned}[a^0](u) &= [\lambda x^{\varphi_1 \otimes \varphi_2}\, \langle b_1^0\rangle \mathsf{pr}_1 x \wedge \langle b_2^0\rangle \mathsf{pr}_2 x](u) \\ &= [a_1^0](u_1) \cap [a_2^0](u_2)\end{aligned}$$

and hence by inductive hypothesis $[a^0](u) \neq \emptyset$ holds iff $a_i \in u_i$ for $i = 1, 2$, that is, iff $a \in u$.

If $\varphi = \varphi_1 \oplus \varphi_2$ and $a = (1, a_1)$ with $a_1 \in |[\varphi_1]|$ (the case $a = (2, a_2)$ being similar), then we have

$$[a^+] = [\mathsf{in}_1 a_1^+] = \{1\} \times [a_1^+] = \{1\} \times \underset{[\varphi_1]}{\downarrow} a_1 = \underset{[\varphi]}{\downarrow} a\,.$$

Let $u \in \mathsf{Idl}([\varphi])$. Then there is $i \in \{1, 2\}$ such that $u = \{i\} \times u_i$ with $u_i \in \mathsf{Idl}([\varphi_i])$. If $i = 1$ we have

$$[a^0](u) = [\lambda x^{\varphi_1 \oplus \varphi_2}\, \mathsf{case}(x, y_1 \cdot \langle a_1^0\rangle y_1, y_2 \cdot \Omega^1)](u) = [a_1^0](u_1)$$

and if $i = 2$ then $[a^0](u) = [\Omega^1] = \emptyset$ by Theorem 10. If follows that $[a^0](u) \neq \emptyset$ holds iff $a_1 \in u_1$.

Let now φ be a positive type and let $a \in |[\varphi]|$. Let $u \in \mathsf{Ini}([\varphi])$, we have

$$[a^-](u^!) = [\lambda x^{!\varphi}\, \langle a^0\rangle \mathsf{der}(x)](u^!) = [a^0](u)$$

so that $[a^-](u^!) \neq \emptyset$ iff $a \in u$ by what we have seen above.

Last, let $\sigma = \varphi \multimap \tau$ and let $a = (b, c) \in |[\sigma]|$. Then $a^+ = \lambda x^{\varphi}\, \mathsf{der}(\mathsf{conj}_{!_\tau}(\langle b^0\rangle x, (c^+)^!))$ satisfies by inductive hypothesis, for each $u \in \mathsf{Idl}([\varphi])$,

$$[a^+](u) = \begin{cases} \downarrow_{[\tau]} c & \text{if } b \in u \\ \emptyset & \text{otherwise}\,. \end{cases}$$

It follows that $[a^+] = \downarrow_{[\varphi \multimap \tau]}(b, c)$. Let now $u \in \mathsf{Ini}([\sigma])$, we have

$$[a^-](u^!) = [\lambda f^{!(\varphi \multimap \tau)}\, \langle c^-\rangle(\langle \mathsf{der}(f)\rangle b^+)^!](u^!) = [c^-](u([b^+])^!)$$

so that $[a^-](u^!) \neq \emptyset$ holds iff $c \in u(\downarrow_{[\varphi]} b)$ and this latter property is equivalent to $(b, c) \in u$ (identifying elements of $\mathbf{Polr}([\varphi], [\sigma])$ with linear functions as explained above). □

Theorem 13 (Full Abstraction) *If* $\vdash M_i : \sigma$ *for* $i = 1, 2$ *satisfy* $M_1 \sim M_2$ *then* $[M_1] = [M_2]$.

Proof. Towards a contradiction, assume that $[M_1] \neq [M_2]$. Wlog assume that a is an element of $[M_1]$ such that $a \notin [M_2]$. Then by Proposition 12 we have $[\langle a^-\rangle M_1^!] = [a^-]([M_1]^!) = \{\emptyset\}$ and $[\langle a^-\rangle M_2^!] = [a^-]([M_2]^!) = \emptyset$. By Theorem 10 it follows that $\langle a^-\rangle M_1^! \downarrow_{\mathsf{nd}}$ and we have not $\langle a^-\rangle M_2^! \downarrow_{\mathsf{nd}}$. It follows that $M_1 \not\sim M_2$. □

3 Global states

3.1 Transition system with parameters

We base our CBPV with global states on a very general enriched notion notion of transition system. A *transition system with parameters* (TSP) is a tuple

$$\mathcal{Q} = (\mathsf{S}_{\mathcal{Q}}, \mathsf{I}_{\mathcal{Q}}, \mathsf{ar}_{\mathcal{Q}}, \mathsf{T}_{\mathcal{Q}})$$

where $\mathsf{S}_{\mathcal{Q}}$ is a countable set of states, $\mathsf{I}_{\mathcal{Q}}$ is a countable set of instructions, $\mathsf{ar}_{\mathcal{Q}} : \mathsf{I}_{\mathcal{Q}} \to \mathbb{N}$ is an arity map and $\mathsf{T}_{\mathcal{Q}}$ is a transition relation. More precisely, setting $\mathsf{I}_{\mathcal{Q}}(k) = \{\theta \in \mathsf{I}_{\mathcal{Q}} \mid \mathsf{ar}_{\mathcal{Q}}(\theta) = k\}$ (the set of instructions or arity k), we have

$$\mathsf{T}_{\mathcal{Q}} \subseteq \bigcup_{k\in\mathbb{N}} (\mathsf{S}_{\mathcal{Q}} \times \mathbb{N}^k) \times \mathsf{I}_{\mathcal{Q}}(k) \times (\mathsf{S}_{\mathcal{Q}} \times \mathbb{N})\,.$$

Intuitively, $((q, \vec{n}), \theta, (q', n)) \in \mathsf{T}_{\mathcal{Q}}$ with $\theta \in \mathsf{I}_{\mathcal{Q}}(k)$ means that, at state q and given k parameters $\vec{n} = (n_1, \dots, n_k)$, the instruction θ leads to state q' with output n. The fact that $\mathsf{T}_{\mathcal{Q}}$ is a relation and not a function means that we also admit non-deterministic and partial behaviors. For such a transition we also use the more familiar notation $(q, \vec{n}) \xrightarrow{\theta}_{\mathcal{Q}} (q', n)$.

Example. There are of course many useful instances of such general structures. Let us mention two of them.

- *Global memory.* Given $N \in \mathbb{N}$, we define a TSP $\mathcal{Q}$ as follows. We set $\mathsf{S}_{\mathcal{Q}} = \mathbb{N}^{\{0,\dots,N-1\}}$, $\mathsf{I}_{\mathcal{Q}} = \{\mathsf{rd}, \mathsf{wr}\}$, $\mathsf{ar}_{\mathcal{Q}}(\mathsf{rd}) = 1$, $\mathsf{ar}_{\mathcal{Q}}(\mathsf{wr}) = 2$ and

$$(q, i) \xrightarrow{\mathsf{rd}}_{\mathcal{Q}} (q', n) \text{ if } 0 \le i < N,\ q(i) = n \text{ and } q' = q$$
$$(q, i, n) \xrightarrow{\mathsf{wr}}_{\mathcal{Q}} (q', n') \text{ if } 0 \le i < N,\ q' = q\,[n/i] \text{ and } n' = 0$$

  where $q' = q\,[n/i]$ is the function $\{0, \dots, N-1\} \to \mathbb{N}$ such that $q'(j) = n$ if $j = i$ and $q'(j) = q(j)$ otherwise. The choice taken that the output integer n' is 0 for the write instruction is of course arbitrary. We could modify these instructions for allowing them to yield error messages in case of out of range error, here we have decided that the instruction would not return in such a case.

- *Non determinism.* We define a TSP $\mathcal{Q}$ as follows: $\mathsf{S}_{\mathcal{Q}} = \{0,1\}^{<\infty}$ (the set of finite sequences of booleans), $\mathsf{I}_{\mathcal{Q}} = \{\mathsf{or}\}$ with $\mathsf{ar}_{\mathcal{Q}}(\mathsf{or}) = 0$ (this should be understood as an oracle delivering a new boolean (or nothing) each time it is called. Then

$$q \xrightarrow{\mathsf{or}}_{\mathcal{Q}} (q', n) \text{ if } q = nq'\,.$$

Such TSP's $\mathcal{Q}$ and $\mathcal{R}$ can easily combined when they have disjoint sets of instructions: one defines $\mathcal{Q} + \mathcal{R}$ by $\mathsf{S}_{\mathcal{Q}+\mathcal{R}} = \mathsf{S}_{\mathcal{Q}} \times \mathsf{S}_{\mathcal{R}}$, $\mathsf{I}_{\mathcal{Q}+\mathcal{R}} = \mathsf{I}_{\mathcal{Q}} \cup \mathsf{I}_{\mathcal{R}}$, $\mathsf{ar}_{\mathcal{Q}+\mathcal{R}}$ defined in the obvious way, and given $\theta \in \mathsf{I}_{\mathcal{Q}}(k)$, we have $(q, r, \vec{n}) \xrightarrow{\theta}_{\mathcal{Q}+\mathcal{R}} (q', r', n)$ if $(q, \vec{n}) \xrightarrow{\theta}_{\mathcal{Q}} (q', n)$ and $r = r'$, and symmetrically for $\theta \in \mathsf{I}_{\mathcal{R}}$.

3.2 Syntax

Given a TSP $\mathcal{Q}$, we define a CBPV lambbda-calculus $\Lambda^{\mathcal{Q}}_{\mathsf{HP}}$ on top of it. More precisely, we extend the syntax of Section 1 as follows.

The definition of positive types is unchanged, for general types, we set:

$$\sigma, \tau \dots := \cdots \mid \mathsf{T}(\varphi) \tag{3}$$

So we add a monadic construction for dealing with effects, in the spirit of Moggi's computational lambda calculus [14]. From a semantical viewpoint one could admit a more general type $\mathsf{T}(\sigma)$ but actually only $\mathsf{T}(\varphi)$ seems syntactically useful.

Concerning terms, the syntax is extended as follows:

$$M, N \dots := \cdots \mid \mathsf{e}(M) \mid \mathsf{seq}(M, x \cdot N) \mid \underline{\theta}(M_1, \dots, M_k)$$

$$\frac{\mathcal{P} \vdash M : \varphi}{\mathcal{P} \vdash \mathsf{e}(M) : \mathsf{T}(\varphi)} \qquad \frac{\mathcal{P} \vdash M : \mathsf{T}(\varphi) \qquad \mathcal{P}, x : \varphi \vdash N : \mathsf{T}(\psi)}{\mathcal{P} \vdash \mathsf{seq}(M, x \cdot N) : \mathsf{T}(\psi)}$$

$$\frac{\theta \in \mathsf{I}_{\mathcal{Q}}(k) \qquad \mathcal{P} \vdash M_i : \iota \text{ for } i = 1, \dots, k}{\mathcal{P} \vdash \underline{\theta}(M_1, \dots, M_k) : \mathsf{T}(\iota)}$$

Figure 7: Typing system for $\Lambda^{\mathcal{Q}}_{\mathsf{HP}}$

$$\frac{}{\mathsf{seq}(\mathsf{e}(V), x \cdot N) \to_{\mathsf{w}} N\,[V/x]} \qquad \frac{M \to_{\mathsf{w}} M'}{\mathsf{e}(M) \to_{\mathsf{w}} \mathsf{e}(M')} \qquad \frac{M_i \to_{\mathsf{w}} M_i' \qquad \forall j \neq i \ M_i' = M_i}{(q, \underline{\theta}(M_1, \dots, M_k)) \to_{\mathsf{w}} (q, \underline{\theta}(M_1', \dots, M_k'))}$$

$$\frac{M \to_{\mathsf{w}} M'}{(q, M) \to_{\mathsf{s}} (q, M')} \qquad \frac{(q, M) \to_{\mathsf{s}} (q', M')}{(q, \mathsf{seq}(M, x \cdot N)) \to_{\mathsf{s}} (q', \mathsf{seq}(M', x \cdot N))}$$

$$\frac{\theta \in \mathsf{I}_{\mathcal{Q}}(k) \qquad (q, n_1, \dots, n_k) \xrightarrow{\theta}_{\mathcal{Q}} (q', n)}{(q, \underline{\theta}(\underline{n_1}, \dots, \underline{n_k})) \to_{\mathsf{s}} (q', \mathsf{e}(\underline{n}))}$$

Figure 8: Stateful reduction rules

for $k \in \mathbb{N}$ and $\theta \in \mathsf{I}_{\mathcal{Q}}(k)$. Our construction $\mathsf{seq}(M, x \cdot N)$ is similar to the more usual $\mathsf{let}\, x = M \,\mathsf{in}\, N$, but we prefer a notation which stresses the order of evaluation.

We define the type ι of flat (unary) natural numbers by $\iota = 1 \oplus \iota$ (by this we mean that $\iota = \mathsf{Fix}\, \zeta \cdot (1 \oplus \zeta)$). We define $\underline{0} = \mathsf{in}_1()$ and $\underline{n+1} = \mathsf{in}_2 \underline{n}$. Observe that $\mathcal{P} \vdash \underline{n} : \iota$ for each $n \in \mathbb{N}$.

Figure 7 provides the additional typing rules.

3.3 Operational semantics

As in Section 1 we define a weak reduction on terms based on a notion of value; the definitions are exactly the same and we use the same notational conventions so we do not repeat them here, although they apply now to an extended syntax. So our calculus is equipped with the weak reduction relation $\to_{\mathsf{w}}$ defined in Figure 2.

Concerning the "stateful" part of the calculus, we introduce a notion of computation $\to_{\mathsf{s}}$ which applies to pairs (q, M) where $q \in \mathsf{S}_{\mathcal{Q}}$ and M is a term. The rules are given in Figure 8.

Proposition 14 (Stateful Subject Reduction) *Assume that $\mathcal{P} \vdash M : \sigma$. If $M \to_{\mathsf{w}} M'$ then $\mathcal{P} \vdash M' : \sigma$. If $(q, M) \to_{\mathsf{s}} (q', M')$ then $\sigma = \mathsf{T}(\varphi)$ for some positive type φ, and we have $\mathcal{P} \vdash M' : \sigma$.*

The proof is quite easy (again, a substitution lemma is needed).

Remark: The second statement of Proposition 14 is particularly important as it justifies the definition of $\to_{\mathsf{w}}$ for $\Lambda^{\mathcal{Q}}_{\mathsf{HP}}$. It is absolutely crucial to observe that, in all of the deduction rules defining $\to_{\mathsf{w}}$ in Figure 2 the terms occurring in the premises have types which are not of shape $\mathsf{T}(\varphi)$. For instance, in the 6th rule, if $\mathcal{P} \vdash M : \sigma$, we must have $\sigma = \varphi \multimap \tau$ for some types φ and τ and hence σ is not of shape $\mathsf{T}(\psi)$. In the 7th rule, if $\mathcal{P} \vdash N : \sigma$ then σ must be positive and is therefore not of shape $\mathsf{T}(\psi)$. Similar considerations apply to all deduction rules of Figure 2. It is only for this reason that the definition of $\to_{\mathsf{w}}$ does not involve states since the only terms whose "reduction modify states" have type of shape $\mathsf{T}(\varphi)$.

3.4 Denotational semantics

We first explain how to interpret the purely monadic constructions $\mathsf{e}(M)$ and $\mathsf{seq}(M, x \cdot N)$ in the general categorical setting of Section 1.1. So we assume to be given a category $\mathcal{L}$ equipped with all the structures and satisfying all the properties outlined in that section.

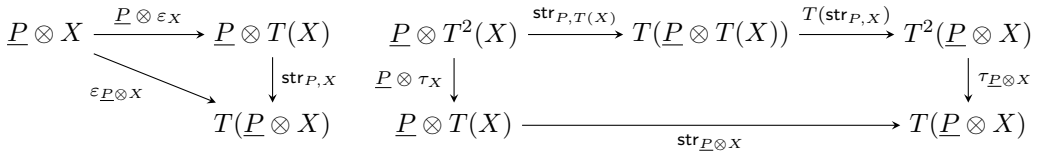

Figure 9: Two commutations satisfied by a computational monad

3.4.1 Computational monad. Like [6], we assume furthermore to be given a monad $T : \mathcal{L} \to \mathcal{L}$ with unit ε and multiplication τ. Moreover we assume this monad to be equipped with a natural transformation $\mathsf{str} : T_1 \to T_2$ where $T_1, T_2 : \mathcal{L}^! \times \mathcal{L} \to \mathcal{L}$ are the functors defined by: $T_1 = \mathsf{U} \otimes T$ and $T_2 = T(\mathsf{U} \otimes \mathcal{L})$ where $\mathsf{U} : \mathcal{L}^! \to \mathcal{L}$ is the forgetful functor. This natural transformation is called the *strength* of T. In other words, for each P object of $\mathcal{L}^!$ and each X object of $\mathcal{L}$, we have a morphism $\mathsf{str}_{P,X} \in \mathcal{L}(\underline{P} \otimes T(X), T(\underline{P} \otimes X))$, natural in P and X. As usual this natural transformation is subject to coherence diagrams of which we give two examples in Figure 9 (the other ones involve the symmetric monoidal structure of $\mathcal{L}$).

Remark: Requiring the strength $\mathsf{str}_{P,X}$ only for "contexts" P taken in $\mathcal{L}^!$ (we do not ask for a general strength $X \otimes T(Y) \to T(X \otimes Y)$) is important as it allows the monad $?_- : \mathcal{L} \to \mathcal{L}$ to have a strength in our sense whereas it wouldn't have with the stronger definition. Remember that "?" is the monad of linear continuations, used non-linearly.

We also assume that the functor T acts as a functor $\mathcal{L}_\subseteq \to \mathcal{L}_\subseteq$ in such a way that, if $\varphi \in \mathcal{L}_\subseteq(X, Y)$, then $(T\varphi)^+ = T(\varphi^+)$ and $(T\varphi)^- = T(\varphi^-)$, and that $T : \mathcal{L}_\subseteq \to \mathcal{L}_\subseteq$ is continuous. In most cases, this condition will trivially be satisfied because T will be defined using basic LL constructs ($\otimes$, $\multimap$, $!_-$ etc).

We extend the inductive definitions of types and term interpretation of [3] to these monadic constructions. Concerning types, we set of course $[\mathsf{T}(\varphi)] = T([\varphi])$.

Assume first that $\mathcal{P} \vdash M : \varphi$, so that by inductive hypothesis $[M]_\mathcal{P} \in \mathcal{L}([\mathcal{P}], [\varphi])$, we set $[\mathsf{e}(M)]_\mathcal{P} = \varepsilon_{[\varphi]}\,[M]_\mathcal{P}$. Assume next that $\mathcal{P}, x : \varphi \vdash M : \mathsf{T}(\psi)$ and $\mathcal{P} \vdash N : \mathsf{T}(\varphi)$. By inductive hypothesis we have $[M]_{\mathcal{P},x:\varphi} \in \mathcal{L}([\mathcal{P}] \otimes [\varphi], T[\psi])$ and $[N]_\mathcal{P} \in \mathcal{L}([\mathcal{P}], T[\varphi])$. We define $[\mathsf{seq}(N, x \cdot M)]_\mathcal{P}$ as the following composition of morphisms in $\mathcal{L}$.

$$\begin{array}{ccccc} [\mathcal{P}] & \xrightarrow{\mathrm{c}_{[\mathcal{P}]}} & [\mathcal{P}] \otimes [\mathcal{P}] & \xrightarrow{[\mathcal{P}] \otimes [N]_\mathcal{P}} & [\mathcal{P}] \otimes T([\varphi]) \\ & & & & \downarrow \mathsf{str}_{[\mathcal{P}]^!,[\varphi]} \\ T([\psi]) & \xleftarrow{\tau_{[\psi]}} & T^2([\psi]) & \xleftarrow{T([M]_{\mathcal{P},x:\varphi})} & T([\mathcal{P}] \otimes [\varphi]) \end{array}$$

This interpretation is sound wrt. our operational semantics:

Proposition 15 *If $\mathcal{P}, x : \varphi \vdash M : \mathsf{T}(\psi)$ and $\mathcal{P} \vdash V : \varphi$, then $[\mathsf{seq}(\mathsf{e}(V), x \cdot M)]_\mathcal{P} = [M\,[V/x]]_\mathcal{P}$.*

This is a direct consequence of the axioms of strong monad, and of the Substitution Lemma which states that $[M\,[V/x]]_\mathcal{P} = [M]_{\mathcal{P},x:\varphi}\,([\mathcal{P}] \otimes [V]_\mathcal{P})\,\mathrm{c}_{[\mathcal{P}]^!}$, and holds because $[V]_\mathcal{P}$ is a coalgebra morphism. This lemma is proven by induction on M.

3.4.2 States in the Scott model of LL. Given an object S in $\mathcal{L}$, it is standard to define a *linear state monad* (T, ε, τ) on $\mathcal{L}$ which satisfies $T(X) = S \multimap S \otimes X$. We will recall the precise definition in a special case.

From now on we take for $\mathcal{L}$ the category **Polr** of Section 2.2. Let S be the object of **Polr** such that $|S| = \mathsf{S}_\mathcal{Q}$ and $\leq_S$ is the diagonal (that is $q \leq_S q'$ iff $q = q'$). The corresponding linear state monad is described in Figure 10. Remember that the objects of $\mathbf{Polr}^!$ can be identified with those of **Polr**. For this particular monad, the strength can be seen as a natural transformation $T'_1 \to T'_2$ where $T'_1, T'_2 : \mathbf{Polr}^2 \to \mathbf{Polr}$ are defined by $T'_1(X, Y) = X \otimes T(Y)$ and $T'_2(X, Y) = T(X \otimes Y)$ and similarly for morphisms.

To avoid considering multisets of terms as in Section 2.1, we assume that the TSP $\mathcal{Q}$ is deterministic in the sense that if $\theta \in \mathsf{I}_\mathcal{Q}(k)$, the relation $\xrightarrow{\theta}_\mathcal{Q}$ is a partial function from $\mathsf{T}_\mathcal{Q} \times \mathbb{N}^k$ to $\mathsf{T}_\mathcal{Q} \times \mathbb{N}$.

$$
\begin{aligned}
|TX| &= \mathsf{S}_{\mathcal{Q}}^2 \times |X| \\
T(f) &= \{((q,q',a),(q,q',b)) \mid q,q' \in \mathsf{S}_{\mathcal{Q}} \text{ and } (a,b) \in f\} \text{ if } f \in \mathbf{Polr}(X,Y) \\
\varepsilon_X &= \{(a,(q,q,a')) \mid q \in \mathsf{S}_{\mathcal{Q}} \text{ and } a' \leq_X a\} \\
\tau_X &= \{(q,q',(q',q'',a)),(q,q'',a')) \mid q,q',q'' \in \mathsf{S}_{\mathcal{Q}} \text{ and } a' \leq_X a\} \\
\mathsf{str}_{X,Y} &= \{((a,(q,q',b)),(q,q',(a',b'))) \mid q,q' \in \mathsf{S}_{\mathcal{Q}} \text{ and } a' \leq_X a,\ b' \leq_Y b\}
\end{aligned}
$$

Figure 10: The linear state monad in **Polr**

$$
\frac{\Phi \vdash M : a : \varphi}{\Phi \vdash \mathsf{e}(M) : (q,q,a) : \mathsf{T}(\varphi)}
\qquad
\frac{\Phi, x : a : \varphi \vdash M : (q',q'',b) : \mathsf{T}(\psi) \qquad \Phi \vdash N : (q,q',a) : \mathsf{T}(\varphi)}{\Phi \vdash \mathsf{seq}(N, x \cdot M) : (q,q'',b) : \mathsf{T}(\psi)}
$$

$$
\frac{\Phi \vdash M_i : n_i : \iota \text{ for } i = 1,\dots,k \qquad (q,n_1,\dots,n_k) \xrightarrow{\theta}_{\mathcal{Q}} (q',n)}{\Phi \vdash \underline{\theta}(M_1,\dots,M_k) : (q,q',n) : \mathsf{T}(\iota)}
$$

Figure 11: Scott typing rules for states

Up to isomorphism we have $[\iota] = \mathsf{N}$ where $\mathsf{N} = (\mathbb{N},=)$ is the object of "flat natural numbers" in **Polr**. With $\theta \in \mathsf{I}_{\mathcal{Q}}(k)$ we associate $\overline{\theta} \in \mathbf{Polr}(\mathsf{N}^{\otimes k}, T(\mathsf{N}))$ given by $\overline{\theta} = \{(\vec{n},(q,q',n)) \mid (q,\vec{n}) \xrightarrow{\theta}_{\mathcal{Q}} (q',n)\}$. Then, given terms M_i such that $\mathcal{P} \vdash M_i : \iota$ for $i = 1,\dots,k$, we set $[\underline{\theta}(M_1,\dots,M_k)]_{\mathcal{P}} = \overline{\theta}\,([M_1]_{\mathcal{P}} \otimes \cdots \otimes [M_k]_{\mathcal{P}}) \circ \mathrm{c}$ where $\mathrm{c} \in \mathbf{Polr}^!([\varphi]^!, ([\varphi]^!)^{\otimes k})$ is the diagonal morphism in the cartesian category $\mathbf{Polr}^!$ (k-ary contraction).

As explained in Section 2.2, the interpretation of terms can be represented as a typing system, see Figure 5. We extend this typing system[5] with the rules of Figure 11 which are quite intuitive if we interpret $M : (q,q',a) : \mathsf{T}(\varphi)$ as "in state q, M reduces to the value a in the new state q'". Then Proposition 7 extends easily to these new constructions: the typing system of Figure 5 and 11 describes exactly the Scott semantics of $\Lambda^{\mathcal{Q}}_{\mathsf{HP}}$.

This interpretation is sound wrt. the operational semantics of Section 3.3. To express this property we need an auxiliary notation. Let $f \in \mathbf{Polr}(X,T(Y))$ and let $q \in \mathsf{S}_{\mathcal{Q}}$, we set $f^{(q)} = \{(a,(q',b)) \mid (a,(q,q',b)) \in f\} \in \mathbf{Polr}(X, S \otimes Y)$.

Proposition 16 *If $\mathcal{P} \vdash M : \sigma$ and $M \to_{\mathsf{w}} M'$ then $[M]_{\mathcal{P}} = [M']_{\mathcal{P}}$. If $\sigma = \mathsf{T}(\varphi)$ and $(q,M) \to_{\mathsf{s}} (q',M')$ then $[M]^{(q)}_{\mathcal{P}} = [M']^{(q')}_{\mathcal{P}}$.*

The proof is easy and uses our hypothesis that $\mathcal{Q}$ is deterministic to deal with the case where $M = \underline{\theta}(\underline{n_1},\dots,\underline{n_k})$. Using multisets of terms and states as in Section 2.1, it could easily be extended to the non-deterministic case.

To prove adequacy, we extend first the definition given in Section 2.3.4 for the interpretation of points of the model as sets of terms, see Figure 6. We set

$$
|(q,q',a)|^{\mathsf{T}(\varphi)} = \{M \mid \vdash M : \mathsf{T}(\varphi) \text{ and } \exists V \in |a|^{\varphi}_{\mathsf{v}}\ (q,M) \to^*_{\mathsf{s}} (q',\mathsf{e}(V))\}
$$

Then we can adapt the reasoning of that section. Since our operational semantics is deterministic (thanks to our hypothesis on $\mathcal{Q}$) we can replace everywhere $\to_{\mathsf{nd}}$ with $\to_{\mathsf{w}}$. Lemma 9 is easily extended. Lemma 8 is rephrased as follows:

Lemma 17 *Assume that $\vdash M : \sigma$. If $M \to_{\mathsf{w}} M'$ and $M' \in |a|^{\sigma}$ then $M \in |a|^{\sigma}$. If $\sigma = \mathsf{T}(\varphi)$, $(q,M) \to_{\mathsf{s}} (q',M')$ and $M' \in |(q',q'',a)|^{\mathsf{T}(\varphi)}$ then $M \in |(q,q'',a)|^{\mathsf{T}(\varphi)}$.*

Again, the proof is easy. Last one has to prove Theorem 10 for this extended language. The statement of the Theorem is unchanged, one has just to consider a few more cases in the proof. They are all easy, we just deal with one of them to give a flavor of the reasoning, using the same notational conventions.

[5] Of course, we drop the rules for $\mathsf{choose}(M,N)$ which is not part of our syntax.

Assume that $\sigma = \mathsf{T}(\psi)$, $a = (q, q'', c)$, $M = \mathsf{seq}(N, x \cdot R)$ with $\Phi \vdash N : (q, q', b) : \mathsf{T}(\varphi)$ and $\Phi, x : b : \varphi \vdash R : (q', q'', c) : \mathsf{T}(\psi)$. By inductive hypothesis we have $N' \in |(q, q', b)|^{\mathsf{T}(\varphi)}$, so let $V \in |b|^{\varphi}_{\mathsf{v}}$ be such that $(q, N) \to_{\mathsf{s}}^{*} (q', \mathsf{e}(V))$. By the reduction rules of Figure 8 we have $(q, M') \to_{\mathsf{s}}^{*} (q', \mathsf{seq}(\mathsf{e}(V), x \cdot R')) \to_{\mathsf{s}} (q', R'\,[V/x])$. By inductive hypothesis applied to R (using the fact that $V \in |a|^{\varphi}_{\mathsf{v}}$) we have $R'\,[V/x] \in |(q', q'', c)|^{\mathsf{T}(\psi)}$, that is $(q', R'\,[V/x]) \to_{\mathsf{s}}^{*} (q'', \mathsf{e}(W))$ for some $W \in |c|^{\psi}_{\mathsf{v}}$. Therefore $(q, M') \to_{\mathsf{s}}^{*} (q'', \mathsf{e}(W))$ which shows that $M' \in |(q, q'', c)|^{\mathsf{T}(\psi)}$ as contended.

3.4.3 Consequences. Assume that $\vdash M : \mathsf{T}(\varphi)$ and let $q \in \mathsf{S}_{\mathcal{Q}}$. By Theorem 10 for $\Lambda^{\mathcal{Q}}_{\mathsf{HP}}$ and by soundness, we have that $(q, M) \to_{\mathsf{s}} (q', \mathsf{e}(V))$ for some value V and some state q' (that is, the computation of M in state q terminates on some value) iff $[M]^{(q)} \neq \emptyset$. This provides us with a purely denotational characterization of termination (independent of any choice of reduction rules) which shows in particular that our definition of $\to_{\mathsf{w}}$ and $\to_{\mathsf{s}}$ is sufficient for evaluating all terminating programs.

One can also derive from this theorem an adequacy theorem similar to Theorem 11. This will be presented in a further paper.

Conclusion. These global state constructions can be performed in many other models of LL: relational semantics, coherence spaces, probabilistic coherence spaces etc. This offers interesting perspectives for combining states with differential LL [2] and probabilistic computing.

References

[1] Thomas Ehrhard. The Scott model of Linear Logic is the extensional collapse of its relational model. *Theoretical Computer Science*, 424:20–45, 2012.

[2] Thomas Ehrhard. An introduction to Differential Linear Logic: proof-nets, models and antiderivatives. *Mathematical Structures in Computer Science*, 2015. To appear.

[3] Thomas Ehrhard. Call-By-Push-Value from a Linear Logic Point of View. In Peter Thiemann, editor, *Programming Languages and Systems - 25th European Symposium on Programming, ESOP 2016, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2016, Eindhoven, The Netherlands, April 2-8, 2016, Proceedings*, volume 9632 of *Lecture Notes in Computer Science*, pages 202–228. Springer, 2016.

[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] Ryu Hasegawa. Two applications of analytic functors. *Theoretical Computer Science*, 272(1-2):113–175, 2002.

[7] Michael Huth. Linear Domains and Linear Maps. In Stephen D. Brookes, Michael G. Main, Austin Melton, Michael W. Mislove, and David A. Schmidt, editors, *MFPS*, volume 802 of *Lecture Notes in Computer Science*, pages 438–453. Springer-Verlag, 1993.

[8] Jean-Louis Krivine. *Lambda-Calculus, Types and Models*. Ellis Horwood Series in Computers and Their Applications. Ellis Horwood, 1993. Translation by René Cori from French 1990 edition (Masson).

[9] 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.

[10] Paul Blain Levy. Call-by-push-value: A subsuming paradigm. In Jean-Yves Girard, editor, *Typed Lambda Calculi and Applications, 4th International Conference, TLCA'99, L'Aquila, Italy, April 7-9, 1999, Proceedings*, volume 1581 of *Lecture Notes in Computer Science*, pages 228–242. Springer, 1999.

[11] Paul Blain Levy. Adjunction Models For Call-By-Push-Value With Stacks. *Electronic Notes in Theoretical Computer Science*, 69:248–271, 2002.

[12] Paul Blain Levy. *Call-By-Push-Value: A Functional/Imperative Synthesis*, volume 2 of *Semantics Structures in Computation*. Springer-Verlag, 2004.

[13] Michael Marz, Alexander Rohr, and Thomas Streicher. Full Abstraction and Universality via Realisability. In *14th Annual IEEE Symposium on Logic in Computer Science, Trento, Italy, July 2-5, 1999*, pages 174–182. IEEE Computer Society, 1999.

[14] Eugenio Moggi. Computational lambda-calculus and monads. In *Proceedings of the 4th Annual IEEE Symposium on Logic in Computer Science*. IEEE Computer Society, 1989.

[15] Mikkel Nygaard and Glynn Winskel. Full abstraction for HOPLA. In Roberto M. Amadio and Denis Lugiez, editors, *CONCUR 2003 - Concurrency Theory, 14th International Conference, Marseille, France, September 3-5, 2003, Proceedings*, volume 2761 of *Lecture Notes in Computer Science*, pages 378–392. Springer-Verlag, 2003.

[16] Alex K. Simpson and Gordon D. Plotkin. Complete Axioms for Categorical Fixed-Point Operators. In *15th Annual IEEE Symposium on Logic in Computer Science, Santa Barbara, California, USA, June 26-29, 2000*, pages 30–41. IEEE Computer Society, 2000.

[17] Thomas Streicher and Bernhard Reus. Classical logic, continuation semantics and abstract machines. *Journal of Functional Programming*, 8(6):543–572, 1998.

[18] Glynn Winskel. A linear metalanguage for concurrency. In Armando Martin Haeberer, editor, *AMAST*, volume 1548 of *Lecture Notes in Computer Science*, pages 42–58. Springer-Verlag, 1998.