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

Abstract. We present and study a simple Call-By-Push-Value lambda-calculus with fix-points and recursive types. We explain its connection with Linear Logic by presenting a denotational interpretation of the language in any model of Linear Logic equipped with a notion of embedding retraction pairs. We consider the particular case of the Scott model of Linear Logic from which we derive an intersection type system for our calculus and prove an adequacy theorem. Last, we introduce a fully polarized version of this calculus which turns out to be a term language for a large fragment of LLP and refines lambda-mu.

Introduction

Linear Logic (LL) has been introduced as a refinement of Intuitionistic Logic: in [14], Girard proposed a very simple and natural translation of intuitionistic logic and of the lambda-calculus in LL. From a categorical point of view, as explained in [28], this translation corresponds to the construction of the Kleisli category of the exponential comonad "!" of LL. An adequate categorical axiomatization of the denotational models of LL has been then provided in [3], see also [26] for a very complete and detailed picture.

In [14], another possible translation of intuitionistic formulas strangely called "boring" is mentioned. It appeared later that, just as the original Girard's translation corresponds to the call-by-name (CBN) evaluation strategy of the lambda-calculus, the "boring" translation corresponds to the call-by-value (CBV) reduction strategy, see in particular [24]. Indeed, a first observation is that this latter translation does not preserve all β-reductions, but only those respecting a CBV discipline. More deeply, domain-theoretic denotational models of the lambda-calculus arising through the original Girard translation, that is, arising as Kleisli categories of the exponential comonad, enjoy an adequacy property expressing that a term reduces to a "value" (a head-normal term, say) iff its interpretation is different from $\bot$. A similar property holds for the CBV translation (now, a closed value is an abstraction) with respect to the CBV reduction strategy.

Both translations give a particularly preeminent role to the Kleisli category of the "!" comonad. This is obvious for the original CBN translation but it is also true for the CBV translation if we consider that the "!" functor defines a

strong monad on the cartesian closed Kleisli category: then the CBV translation coincides with Moggi's interpretation of the CBV lambda-calculus in a CCC equipped with a computational monad [27].

So LL provides a common setting where both CBN and CBV can be faithfully interpreted. In spite of its appealing symmetries and its high degree of asynchrony, the syntax of LL proof nets is complex and does not seem to be a convenient starting point for the design of programming languages; it is rather a powerful tool for analyzing the operational and denotational properties of programming languages. It seems therefore natural to look for lambda-calculi admitting a translation in LL and where both CBN and CBV can be embedded, factorizing the two translations mentioned above.

Levy introduced a lambda-calculus subsuming both CBN and CBV: the *Call-By-Push-Value* lambda-calculus. We propose a similar functional calculus where both CBN and CBV can be encoded and features a simple connection with LL. For reasons which will become clear later, we call it *half-polarized lambda-calculus* (Λ_{HP}). Our calculus is actually isomorphic to (a sub-calculus) of SFPL [25]. It bears some similarities with the adjoint calculus of [2], though we do not need linear variables. It is also close to the Enriched Effect Calculus [10].

The LL exponential "!" allows to turn a term of type A into a term of type $!A$ which is duplicable and discardable, by means of an operation called *promotion*. This discardability and duplicability is made possible by the structural rules $!A$ is equipped with. It was already observed by Girard in [15] (and probably much earlier) that the property of "being equipped with structural rules" is preserved by the $\otimes$ and $\oplus$ connectives of LL. This observation can be made more accurate as follows: a "type equipped with structural rules" is a coalgebra for the "!" connective, that is, an object of the Eilenberg-Moore category of "!", and this category admits $\oplus$ as coproduct and $\otimes$ as cartesian product. This category is not a CCC in general but contains the CCC Kleisli category of "!" as a full sub-category (remember that the Kleisli category is the category of free !-coalgebras). The Eilenberg-Moore category of "!" was used crucially by Girard to give a semantics to a classical sequent calculus and by other authors (see for instance [21]) to interpret classical extensions of the lambda-calculus such as Parigot's $\lambda\mu$-calculus.

Replacing the Kleisli category with this larger Eilenberg-Moore category when interpreting the lambda-calculus has other major benefits. Consider for instance the interpretation of ordinary PCF in an LL-induced categorical model, that is, in the Kleisli category of the "!" comonad of a categorical model $\mathcal{L}$ of LL. The simplest and most natural interpretation for the type of natural numbers is $\mathsf{N} = 1 \oplus 1 \oplus \cdots$ (ω copies of the unit 1 of the tensor product). But 1 has a canonical structure of !-coalgebra (because $1 = !\top$ where $\top$ is the terminal object of $\mathcal{L}$) and this is therefore also true of N, as a coproduct of coalgebras. This means that we have a well behaved morphism $\mathrm{h}_{\mathsf{N}} \in \mathcal{L}(\mathsf{N}, !\mathsf{N})$ which allows to turn any morphism $f \in \mathcal{L}_!(\mathsf{N}, X)$ of the Kleisli CCC $\mathcal{L}_!$ where we interpret PCF into a *linear* morphism $f\,\mathrm{h}_{\mathsf{N}} \in \mathcal{L}(\mathsf{N}, X)$. Operationally, this means that, in spite of the fact that PCF is a CBN language and that its interpretation in $\mathcal{L}_!$

is a CBN interpretation, we can deal with the terms of ground type in a CBV fashion. For instance we can replace the ordinary PCF "if zero" conditional with the following more sensible one:

$$\frac{\mathcal{P} \vdash M : \iota \qquad \mathcal{P} \vdash N : \sigma \qquad \mathcal{P}, x : \iota \vdash N' : \sigma}{\mathcal{P} \vdash \mathsf{if}(M, N, x \cdot N') : \sigma}$$

with the reduction rules

$$\mathsf{if}(\underline{0}, N, x \cdot N') \to N \qquad \mathsf{if}(\underline{n+1}, N, x \cdot N') \to N'\,[\underline{n}/x]$$
$$M \to M' \Rightarrow \mathsf{if}(M, N, x \cdot N') \to \mathsf{if}(M', N, x \cdot N')$$

The denotational interpretation of $\mathsf{if}(M, N, x \cdot N')$ uses crucially the coalgebra structure h_{N} of N. This idea is also reminiscent of Krivine's *storage operators* [20] which allow a CBV discipline for data types in a globally CBN calculus.

Call-By-Push-Value and Λ_{HP} generalize this idea. We consider two classes of types: the *positive types* $\varphi, \psi \ldots$ and the larger class of *general types* $\sigma, \tau, \ldots$ Just as in [15], positive types correspond to objects of $\mathcal{L}^!$ whereas general types are just objects of $\mathcal{L}$. Of course there is an obvious way of considering a positive type as a general one by simply forgetting its coalgebra structure. There is also a way of turning a general type into a positive one using "!", and positive types are stable under sums $\oplus$ and product $\otimes$. In Girard's CBN translation, $\sigma \Rightarrow \tau$ (where $\Rightarrow$ is intuitionistic implication) becomes $!\sigma \multimap \tau$: *the main idea of* Λ_{HP} *is to generalize this idea by allowing to replace* $!\sigma$ *with an arbitrary positive type* φ. Therefore, the implication of Λ_{HP} is linear: all the required non-linearity is provided by the positivity of the premise. Accordingly all variables have positive types. This is also why we use a new notation $\langle M \rangle N$ for the application of M to N, to stress the fact that this application is linear.

There is however a subtlety which does not occur in CBN: consider a term M of type σ with one free variable x of positive type φ. Consider also a closed term N of type φ. The interpretation of M is a morphism $f \in \mathcal{L}(\varphi, \sigma)$ (identifying types with their interpretation) and the interpretation of N is a morphism $g \in \mathcal{L}(1, \varphi)$. There is no reason however for g to belong to $\mathcal{L}^!(1, \varphi)$ and so we cannot be sure that $f\,g$ (we use simple juxtaposition to denote composition of morphisms in $\mathcal{L}$) will coincide with the interpretation of $M\,[N/x]$. Indeed, for that substitutivity property to hold, we would need g to be duplicable and discardable which is the case if we can make sure that $g \in \mathcal{L}^!(1, \varphi)$, but does not hold in general. It is here that the syntactic notion of *value* comes in: if $\varphi = !\sigma$ then N is a value if $N = R^!$ (a promotion of a term R of type σ), if $\varphi = \varphi_1 \otimes \varphi_2$ then N is a value if $N = \langle V_1, V_2 \rangle$ where V_i is a value of type φ_i, and similarly for sums. The main property of values is that their interpretations are coalgebra morphisms: if N is a value then $g \in \mathcal{L}^!(1, \varphi)$. This is why, in Λ_{HP}, β reduction is restricted to the case where the argument is a value. This also means that in $\langle \lambda x^{\varphi}\, M \rangle N$ we need to reduce N to a value before reducing the β-redex. If, for instance, the reduction of N diverges without reaching a value then the reduction of $\langle \lambda x^{\varphi}\, M \rangle N$ diverges even if M does not use x, just as in CBV.

Positive types are also stable under fix-points if we assume that the objects of $\mathcal{L}$ can be equipped with a notion of embedding-retraction pairs, which is usually the case in categories of domains. Under the assumption that this new category has all countable filtered colimits and that the functors interpreting types are continuous, it is easy to prove that all "positive types with parameters" have fix-points which are themselves positive types. The presence of "!" as an explicit type constructor in Λ_{HP} allows to define lazy recursive types such as $\rho = \varphi \otimes !\rho$ where φ is a given type: ρ is the type of streams of elements of type φ. At the level of terms, the construction which introduces an "!" is the aforementioned construction $R^!$ which corresponds to the well known (generalized) promotion of LL aka *exponential box*; this construction corresponds here to *thunks* or *suspensions*. In this stream type ρ the box construction is crucially used to postpone the evaluation of the tail of the stream. The box can be opened (that is, the thunk can be forced) by means of an explicit *dereliction* syntactic construct $\mathsf{der}(M)$ which can be applied to any term M of type $!\sigma$ for some σ and which corresponds exactly to the usual dereliction rule of LL.

Contents. We define a version of Λ_{HP} featuring positive and ordinary types, recursive positive types and a fix-point operator for terms. Since many data types can be defined easily in this language (ordinary integers, lazy integers, lists, streams, various kinds of finite and infinite trees...), it widely encompasses PCF and is closer to FPC [13], with the additional feature of allowing to freely combine CBV and CBN. We define the syntax of the language, provide a typing system and a simple operational semantics which is "weak" in the sense that reduction is forbidden under λs and within boxes (more general reductions can of course be defined but, just as in CBV, a let construction or new reduction rules as in [4] should be added).

We do not describe the embeddings of the CBN and CBV lambda-calculi into Λ_{HP} because they are very similar to the corresponding embeddings into CBPV, described in [22]. It will be fully described in a longer version of this paper.

Then we recall the general definition of a categorical model of LL (with fix-point operators), of its Eilenberg-Moore category and we describe the additional categorical structure which will allow to interpret recursive positive types. To illustrate the connection of Λ_{HP} with LL without introducing a syntax for LL proofs[1], we describe the interpretation of Λ_{HP} in such a categorical model of LL and state a Soundness Theorem.

This categorical semantics is a special case of Levy's adjunction semantics of [22] for CBPV. One of the interests of this LL-based semantics is that it admits extensions in various exciting directions (Ludics, Differential LL, Light LL etc) on which we think that Λ_{HP} will shed a new light. There is another good reason for considering this semantics: it gives a standard way of building the "category of values" as $\mathcal{L}^!$, starting from a "linear" category $\mathcal{L}$ which is usually much simpler. This is typically the case when $\mathcal{L}$ is the category of coherence spaces and linear maps; in this case it is not easy to give a direct, domain-theoretic description of

[1] This would be very interesting but would require more space.

$\mathcal{L}^!$. The Scott semantics described in Section 3 is a remarkable exception where $\mathcal{L}^!$ can be seen as a category of predomains and Scott continuous functions.

We also provide a description of this Scott semantical interpretation as a very simple "intersection" typing system and prove an adequacy theorem (which can be understood as a normalization theorem for this typing system). This result is important as it shows that our weak reduction semantics for Λ_{HP} is complete in the sense that if a closed term is equivalent to a value in any equivalence relation compatible with the denotational semantics, then it weakly reduces to a value. Next we introduce a fully polarized version Λ_{LLP} of our calculus, a system which is closer to Levy's original CBPV although it generalizes it by allowing "classical" constructions borrowed from Parigot's $\lambda\mu$-calculus and is deeply related with Laurent's Polarized Linear Logic LLP. Last we define an encoding of Λ_{HP} into Λ_{LLP} and outline its basic features.

Our initial motivation for introducing Λ_{LLP} was to combine our representation of data as !-coalgebra morphisms in Λ_{HP} with the representation of stacks (continuations) as !-coalgebra morphisms in the semantics of classical calculi such as the CBN $\lambda\mu$-calculus. Λ_{LLP} is presented in the $\lambda\mu\tilde{\mu}$ style [7] and further developed in [8]. We think that it provides a satisfactory answer to our quest and deserves further study. Independently, Pierre-Louis Curien introduced recently in the unpublished note [5] a similar formalism for representing Levy's original CBPV (Curien's calculus however is intuitionistic whereas ours is classical).

1 Syntax

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 $\zeta, \xi \ldots$:

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

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

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 special case of Levy's CBPV [22] in which the type constructor F is kept implicit (and U is "!"). Section 4 proposes a version of CBPV where F is "?", taking benefit of this for introducing constructs related to classical logic.

The notion of substitution is defined as usual. 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.

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

$$\frac{}{\mathcal{P}, x : \varphi \vdash x : \varphi} \qquad \frac{\mathcal{P}, x : \varphi \vdash M : \sigma}{\mathcal{P} \vdash \lambda x^{\varphi} M : \varphi \multimap \sigma} \qquad \frac{\mathcal{P} \vdash M : \varphi \multimap \sigma \qquad \mathcal{P} \vdash N : \varphi}{\mathcal{P} \vdash \langle M \rangle N : \sigma}$$

$$\frac{\mathcal{P} \vdash M : !\sigma}{\mathcal{P} \vdash \mathsf{der}(M) : \sigma} \qquad \frac{\mathcal{P}, x : !\sigma \vdash M : \sigma}{\mathcal{P} \vdash \mathsf{fix}\, x^{!\sigma} M : \sigma}$$

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

$$\frac{\mathcal{P} \vdash M : \varphi_1 \otimes \varphi_2}{\mathcal{P} \vdash \mathsf{pr}_i M : \varphi_i}$$

Fig. 1. Typing system for Λ_{HP}

Remark 1. It might seem strange to the reader acquainted with LL that the rules introducing the $\otimes$ connective and eliminating the $\multimap$ connective have an "additive" handling of typing contexts (the same typing context $\mathcal{P}$ occurs in both premises): our contexts are not linear. The reason for this becomes clear in Section 2 where positive types are interpreted as !-coalgebras, which are equipped with morphisms allowing to interpret the structural rules of weakening and contraction. This is why typing contexts involve positive types only.

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:

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

Remark 2. A closed value is simply a tree whose leaves are "boxes" or "thunks" $M^!$ (where the M's are arbitrary well typed closed terms) and whose internal nodes are either unary nodes bearing an index 1 or 2, or ordered binary nodes.

Figure 2 defines weak reduction $\to_{\mathsf{w}}$.

Proposition 1. *The reduction $\to_{\mathsf{w}}$ enjoys subject reduction and Church-Rosser.*

The first statement is a straightforward verification using a Substitution Lemma that we do not state. The second one is easy: $\to_{\mathsf{w}}$ has the diamond property.

Proposition 2. *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.*

This is easy. In the second statement M has to be closed (the term $\langle \mathsf{der}(x) \rangle V$ is normal, is not a value and can be given a positive type).

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

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

$$\frac{M \to_{\mathsf{w}} M'}{\langle M\rangle N \to_{\mathsf{w}} \langle M'\rangle N} \qquad \frac{N \to_{\mathsf{w}} N'}{\langle M\rangle N \to_{\mathsf{w}} \langle M\rangle N'}$$

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

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

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

Fig. 2. Weak reduction axioms and rules for Λ_{HP}

1.1 Examples

Given any type σ, we define $\Omega^{\sigma} = \mathsf{fix}\, x^{!\sigma}\, \mathsf{der}(x)$ which satisfies $\vdash \Omega^{\sigma} : \sigma$. It is clear that $\Omega^{\sigma} \to_{\mathsf{w}} \mathsf{der}((\Omega^{\sigma})^!) \to_{\mathsf{w}} \Omega^{\sigma}$ so that we can consider Ω^{σ} as the ever-looping program of type σ.

Unit type and natural numbers. We define a unit type 1 by $1 = !\top$, and we set $* = (\Omega^{\top})^!$. We define the type ι of 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 $\underline{n+1} = \mathsf{in}_2 \underline{n}$ so that we have $\mathcal{P} \vdash \underline{n} : \iota$ for each $n \in \mathbb{N}$.

Then, given a term M, we define the term $\mathsf{suc}(M) = \mathsf{in}_2 M$, so that we have

$$\frac{\mathcal{P} \vdash M : \iota}{\mathcal{P} \vdash \mathsf{suc}(M) : \iota}$$

Last, given terms M, N_1 and N_2 and a variable x, we define an "ifz" conditional by $\mathsf{if}(M, N_1, x \cdot N_2) = \mathsf{case}(M, z \cdot N_1, x \cdot N_2)$ where z is not free in N_1, so that

$$\frac{\mathcal{P} \vdash M : \iota \qquad \mathcal{P} \vdash N_1 : \sigma \qquad \mathcal{P}, x : \iota \vdash N_2 : \sigma}{\mathcal{P} \vdash \mathsf{if}(M, N_1, x \cdot N_2) : \sigma}$$

Streams. Let φ be a positive type and S_{φ} be the positive type defined by $\mathsf{S}_{\varphi} = \varphi \otimes !\mathsf{S}_{\varphi}$, that is $\mathsf{S}_{\varphi} = \mathsf{Fix}\,\zeta \cdot (\varphi \otimes !\zeta)$. We can define a term M such that $\vdash M : \mathsf{S}_{\varphi} \multimap \iota \multimap \varphi$ which computes the nth element of a stream:

$$M = \mathsf{fix}\, f^{!(\mathsf{S}_{\varphi} \multimap \iota \multimap \varphi)}\, \lambda x^{\mathsf{S}_{\varphi}}\, \lambda y^{\iota}\, \mathsf{if}(y, \mathsf{pr}_1 x, z \cdot \langle \mathsf{der}(f)\rangle \mathsf{der}(\mathsf{pr}_2 x)\, z)$$

Conversely, we can define a term N such that $\vdash N : !(\iota \multimap \varphi) \multimap \mathsf{S}_{\varphi}$ which turns a function into a stream.

$$N = \mathsf{fix}\, F^{!(!(\iota \multimap \varphi) \multimap \mathsf{S}_{\varphi})}\, \lambda f^{!(\iota \multimap \varphi)}\, \langle \langle \mathsf{der}(f)\rangle \underline{0}, (\langle \mathsf{der}(F)\rangle (\lambda x^{\iota}\, \langle \mathsf{der}(f)\rangle \mathsf{suc}(x))^!)^!\rangle$$

Observe that the recursive call of F is encapsulated into a box, which makes the construction lazy.

Lists. There are various possibilities for defining a type of lists of elements of a positive type φ. The simplest definition is $\lambda_0 = 1 \oplus (\varphi \otimes \lambda_0)$. This corresponds to the ordinary ML type of lists. But we can also define $\lambda_1 = 1 \oplus (\varphi \otimes !\lambda_1)$ and then we have a type of lazy lists (or terminable streams) where the tail of the list is computed only when required.

We could also consider $\lambda_2 = 1 \oplus (!\sigma \otimes \lambda_2)$ which allows to manipulate lists of objects of type σ (which can be a general type) without accessing their elements.

2 Denotational Semantics

The kind of denotational models we are interested in in this paper are those induced by a model of LL, in the spirit of Girard's seminal work [15] on the semantics of the classical system LC where positive formulas are interpreted as $\otimes$-comonoids; this interpretation is further developed *e.g.* in [21]. We use here exactly the same idea for interpreting positive types.

We first recall the general categorical definition of a model of LL implicit in [14], our main reference here is [26] to which we also refer for the rich bibliography on this general topic.

2.1 Models of Linear Logic

A model of LL consists of the following data.

- A category $\mathcal{L}$.
- A symmetric monoidal structure $(\otimes, 1, \lambda, \rho, \alpha, \sigma)$ which is assumed to be closed: $\otimes$ 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 isomorphisms satisfying coherence diagrams that we do not record 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 linear curryfication map $\mathcal{L}(Z \otimes X, Y) \to \mathcal{L}(Z, X \multimap Y)$.
- An object $\bot$ of $\mathcal{L}$ such that the natural morphism $\eta_X = \mathrm{cur}(\mathrm{ev}\ \sigma_{X \multimap \bot, X}) \in \mathcal{L}(X, (X \multimap \bot) \multimap \bot)$ is an iso for each object X (one says that $\mathcal{L}$ is a $*$-autonomous category[2]). We use $X^\bot$ for the object $X \multimap \bot$ of $\mathcal{L}$.
- 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 use 0 for the initial object, $\oplus$ for the coproduct and in_i for the injections. Given an object X of $\mathcal{L}$, we use in^X for the unique element of $\mathcal{L}(0, X)$.

[2] Though not essential for interpreting Λ_{HP}, this assumption is natural as it holds in the concrete models we have in mind and it simplifies the interpretation of Λ_{LLP}, the calculus of Section 4

– We are also given a comonad $!_- : \mathcal{L} \to \mathcal{L}$ with counit $\mathrm{der}_X \in \mathcal{L}(!X, X)$ (called dereliction) and comultiplication $\mathrm{dig}_X \in \mathcal{L}(!X, !!X)$ (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 $\mathrm{m}^0 \in \mathcal{L}(1, !\top)$ and a natural iso $\mathrm{m}^2_{X,Y} \in \mathcal{L}(!X \otimes !Y, !(X \,\&\, Y))$ which satisfy commutations that we do not record here. We also require a coherence condition relating m^2 and dig.

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.

Lax monoidality. It follows that we can define a lax symmetric monoidal structure for the functor $!_-$ from the symmetric monoidal category $(\mathcal{L}, \otimes)$ to itself. This means that we can define a morphism $\mu^0 \in \mathcal{L}(1, !1)$ and a natural transformation $\mu^2_{X,Y} \in \mathcal{L}(!X \otimes !Y, !(X \otimes Y))$ which satisfy some coherence diagrams whose main consequence is that we can canonically extend this natural transformation to the case of n-ary tensors:

$$\mu^{(n)}_{X_1,\ldots,X_N} \in \mathcal{L}(!X_1 \otimes \cdots \otimes !X_n, !(X_1 \otimes \cdots \otimes X_n))$$

in a way which is compatible with the symmetric monoidal structure of $\mathcal{L}$ (and allows us to write things just as if $\otimes$ were strictly associative).

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 $\mathrm{dig}_{\underline{P}}\, \mathrm{h}_P = !\mathrm{h}_P\, \mathrm{h}_P$.

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 $\mathrm{h}_Q\, f = !f\, \mathrm{h}_P$. 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 $(!X, \mathrm{dig}_X)$ and a morphism $f \in \mathcal{L}(X, Y)$ to the coalgebra morphism $!f \in \mathcal{L}^!((!X, \mathrm{dig}_X), (!Y, \mathrm{dig}_Y))$. This functor 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 $f \in \mathcal{L}(\underline{P}, X)$, 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

$$f^!\, g = (f\, g)^! \tag{3}$$

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 $\mathcal{L}^!$, one defines an object $P \otimes Q$ of $\mathcal{L}^!$ by setting $\underline{P \otimes Q} = \underline{P} \otimes \underline{Q}$ and by defining $\mathrm{h}_{P\otimes Q}$ as a composition of morphisms:

$$\underline{P} \otimes \underline{Q} \xrightarrow{\mathrm{h}_P \otimes \mathrm{h}_Q} !\underline{P} \otimes !\underline{Q} \xrightarrow{\mu^2_{\underline{P},\underline{Q}}} !(\underline{P} \otimes \underline{Q})$$

Any object P of $\mathcal{L}^!$ can be equipped with a canonical structure of commutative comonoid. This means that we can define a morphism $\mathrm{w}_P \in \mathcal{L}^!(P, 1)$ and a morphism $\mathrm{c}_P \in \mathcal{L}^!(P, P \otimes P)$ which satisfy the commutations of Figure 3.

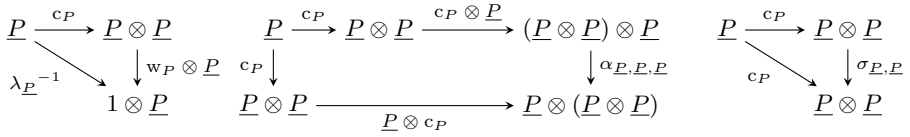


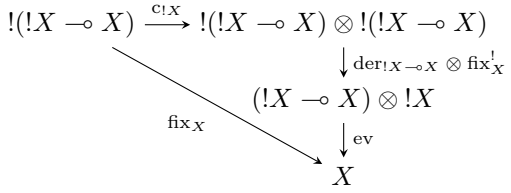
Fig. 3. Commutative $\otimes$-comonoid

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 surprisingly long computations for which we refer again to [26].

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 $(\bigoplus_{i\in I} \underline{P_i}, (\mathrm{in}_i)_{i\in I})$ in $\mathcal{L}$, then it admits a coproduct in $\mathcal{L}^!$. This coproduct $P = \bigoplus_{i\in I} P_i$ is defined by $\underline{P} = \bigoplus_{i\in I} \underline{P_i}$, with a structure map h_P defined by the fact that, for each $i \in I$, $\mathrm{h}_P \,\mathrm{in}_i$ is the following composition of morphisms:

$$\underline{P_i} \xrightarrow{\mathrm{h}_{P_i}} !\underline{P_i} \xrightarrow{!\mathrm{in}_i} !\underline{P}$$

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



Remark 3. It might seem natural to require stronger uniformity conditions inspired by the notion of *Conway operator* [30]. 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.

2.2 Embedding-retraction pairs

We introduce the categorical assumptions used to interpret fix-points of types.

We assume that 0 and $\top$ are isomorphic; these isos being unique, we assume that 0 and $\top$ are the same objects[3].

[3] This is true in many concrete models. It implies that any hom-set $\mathcal{L}(X, Y)$ has a distinguished element which coincides with the least element $\bot$ in denotational models based on domains or games. So this identification is typical of models featuring partial morphisms, which is required here because of the availability of fix-point operators for types and for programs.

We assume to be given a category $\mathcal{L}_\subseteq$ such that $\mathsf{Obj}(\mathcal{L}_\subseteq) = \mathsf{Obj}(\mathcal{L})$ together 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 $(\varphi^-, \varphi^+) = \mathsf{F}(\varphi)$. We assume that $\varphi^- \, \varphi^+ = \mathrm{Id}_X$. We define E (for *embedding*) as the functor $\mathrm{pr}_2 \, \mathsf{F} : \mathcal{L}_\subseteq \to \mathcal{L}$. We assume moreover that the following properties hold.

- The category $\mathcal{L}_\subseteq$ has all countable filtered colimits and these colimits are preserved by E.
- 0 is initial in $\mathcal{L}_\subseteq$ with $\mathsf{E}\theta^X = \mathsf{in}^X$ if θ^X is the unique element of $\mathcal{L}_\subseteq(0, X)$.
- There is a continuous functor[4] $\otimes_\subseteq : \mathcal{L}_\subseteq^2 \to \mathcal{L}_\subseteq$ which behaves as $\otimes$ on objects and satisfies $(\varphi \otimes_\subseteq \psi)^+ = \varphi^+ \otimes \psi^+$ and $(\varphi \otimes_\subseteq \psi)^- = \varphi^- \otimes \psi^-$. We use the same notation $\otimes$ for the functor $\otimes_\subseteq$. We make similar assumptions for $\oplus$ and $!\_$.
- There is a continuous functor $\mathcal{N} : \mathcal{L}_\subseteq \to \mathcal{L}_\subseteq$ such that $\mathcal{N}(X) = X^\perp$, $\mathcal{N}(\varphi)^+ = (\varphi^-)^\perp$ and $\mathcal{N}(\varphi)^- = (\varphi^+)^\perp$. We simply denote $\mathcal{N}(\varphi)$ as $\varphi^\perp$; remember that this operation is covariant.

So we can define a continuous covariant functor $\multimap : \mathcal{L}_\subseteq^2 \to \mathcal{L}_\subseteq$ by $X \multimap Y = (X \otimes Y^\perp)^\perp$ and $\varphi \multimap \psi = (\varphi \otimes \psi^\perp)^\perp$, so that $(\varphi \multimap \psi)^+ = \varphi^- \multimap \psi^+$ and $(\varphi \multimap \psi)^- = \varphi^+ \multimap \psi^-$ in $\mathcal{L}$.

We need to extend this notion of embedding-retraction pair to !-coalgebras because we want to define fix-points of positive types. Let $\mathcal{L}_\subseteq^!$ be the category whose objects are those of $\mathcal{L}^!$ and where

$$\mathcal{L}_\subseteq^!(P, Q) = \{\varphi \in \mathcal{L}_\subseteq(\underline{P}, \underline{Q}) \mid \varphi^+ \in \mathcal{L}^!(P, Q)\}\,.$$

In this definition, it is important *not to require* φ^- to be a coalgebra morphism. We still use U for the obvious forgetful functor $\mathcal{L}_\subseteq^! \to \mathcal{L}_\subseteq$. Observe that $\otimes$ and $\oplus$ define functors $(\mathcal{L}_\subseteq^!)^2 \to \mathcal{L}_\subseteq^!$ and that $!\_$ defines a functor $\mathcal{L}_\subseteq \to \mathcal{L}_\subseteq^!$.

Let J be a countable filtered category and let $\mathcal{E} : J \to \mathcal{L}_\subseteq^!$ be a functor. Let X be the colimit of the functor $\mathsf{U}\,\mathcal{E}$ in $\mathcal{L}_\subseteq$ and let $(\varphi_i \in \mathcal{L}_\subseteq(\mathsf{U}\mathcal{E}(i), X))_{i \in J}$ be the corresponding colimit cocone. We know that $(\varphi_i^+ \in \mathcal{L}(\mathsf{U}\mathcal{E}(i), X))_{i \in J}$ is a colimit cocone in $\mathcal{L}$. In particular, to prove that two morphisms $g, g' \in \mathcal{L}(X, Y)$ are equal, it suffices to prove that $g \, \varphi_i^+ = g' \, \varphi_i^+$ for each $i \in J$.

We want to equip X with a coalgebra structure $h \in \mathcal{L}(X, !X)$. For this, due to this universal property, it suffices to define a cocone $(f_i \in \mathcal{L}(\mathsf{U}\mathcal{E}(i), !X))_{i \in J}$. We set $f_i = !\varphi_i^+ \, \mathrm{h}_{\mathcal{E}(i)}$ and h is completely characterized by the fact that $h \, \varphi_i^+ = f_i$ for each $i \in J$.

Let us prove that $\mathrm{der}_X \; h = \mathrm{Id}_X$. We have $\mathrm{der}_X \; h \, \varphi_i^+ = \mathrm{der}_X \; !\varphi_i^+ \, \mathrm{h}_{\mathcal{E}(i)} = \varphi_i^+ \, \underline{\mathrm{der}_{\mathcal{E}(i)}} \; h_{\mathcal{E}(i)} = \varphi_i^+$ and the result follows from the universal property. The equation $\mathrm{dig}_X \; h = !h \, h$ is proven similarly: we have $\mathrm{dig}_X \; h \, \varphi_i^+ = \mathrm{dig}_X \; !\varphi_i^+ \, \mathrm{h}_{\mathcal{E}(i)} = !!\varphi_i^+ \, \underline{\mathrm{dig}_{\mathcal{E}(i)}} \; \mathrm{h}_{\mathcal{E}(i)} = !!\varphi_i^+ \, !\mathrm{h}_{\mathcal{E}(i)} \, \mathrm{h}_{\mathcal{E}(i)} = !f_i \, \mathrm{h}_{\mathcal{E}(i)}$ and $!h \, h \, \varphi_i^+ = !h \, !\varphi_i^+ \, \mathrm{h}_{\mathcal{E}(i)} = !f_i \, \mathrm{h}_{\mathcal{E}(i)}$.

[4] That is, a directed colimits preserving functor.

So we have proven that $\mathcal{E}$ has a colimit in the category $\mathcal{L}^!$ of coalgebras. A functor $\Phi : \mathcal{M}_1 \times \cdots \times \mathcal{M}_n \to \mathcal{M}$ (where $\mathcal{M}$ and the $\mathcal{M}_i$s belong to $\{\mathcal{L}_\subseteq, \mathcal{L}^!_\subseteq\}$) is continuous if it commutes with countable filtered colimits.

Proposition 3. *The functors $\otimes$ and $\oplus$ from $(\mathcal{L}^!_\subseteq)^2$ to $\mathcal{L}^!_\subseteq$ are continuous. The functor $!_-$: $\mathcal{L}_\subseteq \to \mathcal{L}^!_\subseteq$ is continuous. The functor $\multimap : (\mathcal{L}_\subseteq)^2 \to \mathcal{L}_\subseteq$ is continuous.*

Immediate consequence of our hypotheses and of the above considerations.

Theorem 1. *Let $\Phi : (\mathcal{L}^!_\subseteq)^{n+1} \to \mathcal{L}^!_\subseteq$ be continuous. There is a continuous $Fix(\Phi) : (\mathcal{L}^!_\subseteq)^n \to \mathcal{L}^!_\subseteq$ which is naturally isomorphic to $\Phi \circ \langle \mathrm{Id}^n, Fix(\Phi)\rangle$.*

Proof. Let $\boldsymbol{P} = (P_1, \ldots, P_n)$ be a tuple of objects of $\mathcal{L}^!$. Consider the functor $\Phi_{\boldsymbol{P}} : \mathcal{L}^!_\subseteq \to \mathcal{L}^!_\subseteq$ defined by $\Phi_{\boldsymbol{P}}(P) = \Phi(\boldsymbol{P}, P)$ and similarly for morphisms. Consider the set of natural numbers equipped with the usual order relation as a filtered category ($\mathbb{N}(n,m)$ has one element $l_{n,m}$ if $n \leq m$ and is empty otherwise). We define a functor $\mathcal{E} : \mathbb{N} \to \mathcal{L}^!_\subseteq$ as follows. First, we set $\mathcal{E}(i) = \Phi^i_{\boldsymbol{P}}(0)$. For each i, we define $\varphi_0 = \theta^{\mathcal{E}(1)}$ and then we set $\varphi_{i+1} = \Phi_{\boldsymbol{P}}(\varphi_i)$. Then, given $i, j \in \mathbb{N}$ such that $i \leq j$, we set $\mathcal{E}(l_{i,j}) = \varphi_{j-1} \cdots \varphi_i$. We define $\mathrm{Fix}(\Phi)(\boldsymbol{P})$ as the colimit of this functor $\mathcal{E}$ in $\mathcal{L}^!_\subseteq$. By standard categorical methods using the universal property of colimits, we extend this operation to a continuous functor $\mathrm{Fix}(\Phi) : (\mathcal{L}^!_\subseteq)^n \to \mathcal{L}^!_\subseteq$ which satisfies the required condition by continuity of Φ. □

2.3 Interpreting types and terms

With any positive type φ and any repetition-free list $\boldsymbol{\zeta} = (\zeta_1, \ldots, \zeta_n)$ of type variables containing all free variables of φ we associate a continuous functor $[\varphi]^!_{\boldsymbol{\zeta}} : (\mathcal{L}^!_\subseteq)^n \to \mathcal{L}^!_\subseteq$ and with any general type σ and any list $\boldsymbol{\zeta} = (\zeta_1, \ldots, \zeta_n)$ of pairwise distinct type variables containing all free variables of σ we associate a continuous functor $[\sigma]_{\boldsymbol{\zeta}} : (\mathcal{L}^!_\subseteq)^n \to \mathcal{L}_\subseteq$. We give the definition on objects, the definition on morphisms being similar.

$$
\begin{array}{rr}
[\zeta_i]^!_{\boldsymbol{\zeta}}(\boldsymbol{P}) = P_i & [!\sigma]^!_{\boldsymbol{\zeta}}(\boldsymbol{P}) = !([\sigma]_{\boldsymbol{\zeta}}(\boldsymbol{P})) \\
[\varphi \otimes \psi]^!_{\boldsymbol{\zeta}}(\boldsymbol{P}) = [\varphi]^!_{\boldsymbol{\zeta}}(\boldsymbol{P}) \otimes [\psi]^!_{\boldsymbol{\zeta}}(\boldsymbol{P}) & \\
[\varphi \oplus \psi]^!_{\boldsymbol{\zeta}}(\boldsymbol{P}) = [\varphi]^!_{\boldsymbol{\zeta}}(\boldsymbol{P}) \oplus [\psi]^!_{\boldsymbol{\zeta}}(\boldsymbol{P}) & \\
[\mathsf{Fix}\,\zeta \cdot \varphi]^!_{\boldsymbol{\zeta}} = \mathrm{Fix}([\varphi]^!_{\boldsymbol{\zeta},\zeta}) & [\top]_{\boldsymbol{\zeta}}(\boldsymbol{P}) = \top \quad \text{(the terminal object of } \mathcal{L}) \\
[\varphi]_{\boldsymbol{\zeta}} = \mathsf{U}\,[\varphi]^!_{\boldsymbol{\zeta}} & [\varphi \multimap \sigma]_{\boldsymbol{\zeta}}(\boldsymbol{P}) = (\underline{[\varphi]^!{}_{\boldsymbol{\zeta}}(\boldsymbol{P})}) \multimap ([\sigma]_{\boldsymbol{\zeta}}(\boldsymbol{P}))
\end{array}
$$

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 $\mathcal{L}^!$.

From now on, we assume that the only isos of $\mathcal{L}_\subseteq$ are the identity maps. This means in particular that the isos resulting from Theorem 1 are identities. This assumption is necessary because we have no roll/unroll syntactic constructs of terms associated with fixpoints of types. It holds in many concrete models, and in particular in the Scott model of Section 3.

Interpreting terms. Given a typing context $\mathcal{P} = (x_1 : \varphi_1, \ldots, x_n : \varphi_n)$, we define $[\mathcal{P}]^!$ as the object $[\varphi_1]^! \otimes \cdots \otimes [\varphi_n]^!$ of $\mathcal{L}^!$. Notice that $\underline{[\mathcal{P}]^!} = [\varphi_1] \otimes \cdots \otimes [\varphi_n]$. We denote this object of $\mathcal{L}$ as $[\mathcal{P}]$.

Given a term M, a typing context $\mathcal{P} = (x_1 : \varphi_1, \ldots, x_n : \varphi_n)$ and a type σ such that $\mathcal{P} \vdash M : \sigma$, we define $[M]_{\mathcal{P}} \in \mathcal{L}([\mathcal{P}], [\sigma])$ by induction on the typing derivation of M, that is, on M. Indeed, given a term M and a typing context $\mathcal{P}$, there is at most one type σ and one derivation of $\mathcal{P} \vdash M : \sigma$ (and this derivation is isomorphic to M).

Remark 4. A crucial observation is that $\mathcal{L}^!([\mathcal{P}]^!, [\varphi]^!) \subseteq \mathcal{L}([\mathcal{P}], [\varphi])$ for any positive type φ. Hence, for a term M such that $\mathcal{P} \vdash M : \varphi$, it may happen, *but it is not necessarily the case*, that $[M]_{\mathcal{P}} \in \mathcal{L}^!([\mathcal{P}]^!, [\varphi]^!)$. The terms M which have this property are duplicable and discardable, and the main property of values is that they belong to this semantically defined class of terms. Let us call such terms *$\mathcal{L}$-regular*.

We define $[M]_{\mathcal{P}}$ by induction on the typing derivation, that is, on M.

If $M = x_i$ for $1 \leq i \leq n$, then $[M]_{\mathcal{P}} = \mathrm{pr}_i \in \mathcal{L}^!([\mathcal{P}]^!, [P_i]^!)$. Remember indeed that $[\mathcal{P}]^!$ is the cartesian product of $[P_1]^!$,...,$[P_n]^!$ in $\mathcal{L}^!$. Observe that M is $\mathcal{L}$-regular.

Assume that $M = N^!$ and $\sigma = !\tau$ with $\mathcal{P} \vdash N : \tau$. By inductive hypothesis we have $[N]_{\mathcal{P}} \in \mathcal{L}(\underline{[\mathcal{P}]^!}, [\tau])$ and hence we can set $[M]_{\mathcal{P}} = [N]^!_{\mathcal{P}} \in \mathcal{L}^!([\mathcal{P}]^!, ![\tau])$ so that M is $\mathcal{L}$-regular.

Assume that $M = \langle M_1, M_2 \rangle$ and $\sigma = \varphi_1 \otimes \varphi_2$ with $\mathcal{P} \vdash M_i : \varphi_i$ for $i = 1, 2$. By inductive hypothesis we have defined $[M_i]_{\mathcal{P}} \in \mathcal{L}([\mathcal{P}], [\varphi_i])$ for $i = 1, 2$. Since $[\mathcal{P}] = \underline{[\mathcal{P}]^!}$ we have a contraction morphism $\mathrm{c}_{[\mathcal{P}]^!} \in \mathcal{L}^!([\mathcal{P}]^!, [\mathcal{P}]^! \otimes [\mathcal{P}]^!)$ so that we can set $[M] = ([M_1]_{\mathcal{P}} \otimes [M_2]_{\mathcal{P}})\ \mathrm{c}_{[\mathcal{P}]^!} \in \mathcal{L}([\mathcal{P}], [\sigma])$. Hence if M_1 and M_2 are $\mathcal{L}$-regular, then M is $\mathcal{L}$-regular.

Assume that $M = \mathsf{in}_i N$ (for $i = 1$ or $i = 2$)) and $\sigma = \varphi_1 \oplus \varphi_2$. By inductive hypothesis we have $[N]_{\mathcal{P}} \in \mathcal{L}([\mathcal{P}], [\varphi_i])$ and since we have $\mathrm{in}_i \in \mathcal{L}^!([\varphi_i]^!, [\varphi_1]^! \oplus [\varphi_2]^!)$ it makes sense to set $[M]_{\mathcal{P}} = \mathrm{in}_i\, [N]_{\mathcal{P}} \in \mathcal{L}([\mathcal{P}], [\sigma])$. Observe that if N is $\mathcal{L}$-regular then so is M.

Assume that $M = \lambda x^{\varphi} N$ and $\sigma = \varphi \multimap \tau$ with $\mathcal{P}, x : \varphi \vdash N : \tau$. By inductive hypothesis we have $[N]_{\mathcal{P},x:\varphi} \in \mathcal{L}([\mathcal{P}] \otimes [\varphi], [\tau])$ and we set $[M]_{\mathcal{P}} = \mathrm{cur}\,([N]_{\mathcal{P},x:\varphi}) \in \mathcal{L}([\mathcal{P}], [\varphi] \multimap [\tau])$. Of course, even if τ is positive and N is $\mathcal{L}$-regular, M is not $\mathcal{L}$-regular, simply because its type is not positive.

Assume that $M = \langle N \rangle R$ with $\mathcal{P} \vdash N : \varphi \multimap \sigma$ and $\mathcal{P} \vdash R : \varphi$ for some positive type φ. By inductive hypothesis we have $[N]_{\mathcal{P}} \in \mathcal{L}([\mathcal{P}], [\varphi] \multimap [\sigma])$ and $[R]_{\mathcal{P}} \in \mathcal{L}([\mathcal{P}], [\varphi])$. Since $[\mathcal{P}] = \underline{[\mathcal{P}]^!}$ we have a contraction morphism $\mathrm{c}_{[\mathcal{P}]^!} \in \mathcal{L}^!([\mathcal{P}]^!, [\mathcal{P}]^! \otimes [\mathcal{P}]^!)$ so that we can set $[M]_{\mathcal{P}} = \mathrm{ev}\,([N]_{\mathcal{P}} \otimes [R]_{\mathcal{P}})\ \mathrm{c}_{[\mathcal{P}]^!} \in \mathcal{L}([\mathcal{P}], [\sigma])$.

Assume that $M = \mathsf{case}(N, x_1 \cdot R_1, x_2 \cdot R_2)$ with $\mathcal{P} \vdash N : \varphi_1 \oplus \varphi_2$ and $\mathcal{P}, x_i : \varphi_i \vdash R_i : \sigma$ for $i = 1, 2$. By inductive hypothesis we have $[M]_{\mathcal{P}} \in \mathcal{L}([\mathcal{P}], [\varphi_1] \oplus [\varphi_2])$ and $[R_i]_{\mathcal{P},x_i:\varphi_i} \in \mathcal{L}([\mathcal{P}] \otimes [\varphi_i], [\sigma])$ for $i = 1, 2$. By the universal property of the coproduct $\oplus$ in $\mathcal{L}$ and by the fact that the functor $[\mathcal{P}] \otimes \_$ is a left adjoint, there is exactly one morphism $f \in \mathcal{L}([\mathcal{P}] \otimes ([\varphi_1] \oplus [\varphi_2]), [\sigma])$ such that $f\,([\mathcal{P}] \otimes \mathrm{in}_i) = [R_i]_{\mathcal{P},x_i:\varphi_i}$ for $i = 1, 2$. Then we set $[M]_{\mathcal{P}} = f\,([\mathcal{P}] \otimes [N]_{\mathcal{P}})\ \mathrm{c}_{[\mathcal{P}]^!}$.

Assume that $M = \mathsf{pr}_i N$ and $\sigma = \varphi_i$ for $i = 1$ or $i = 2$, with $\mathcal{P} \vdash N : \varphi_1 \otimes \varphi_2$. By inductive hypothesis we have $[N]_{\mathcal{P}} \in \mathcal{L}([\mathcal{P}], [\varphi_1] \otimes [\varphi_2])$. Then remember that we have the projection $\mathrm{pr}_i \in \mathcal{L}^!([\varphi_1]^! \otimes [\varphi_2]^!, [\varphi_i]^!)$ so that we can set $[M]_{\mathcal{P}} = \mathrm{pr}_i\, [N]_{\mathcal{P}} \in \mathcal{L}([\mathcal{P}], [\varphi_i])$.

Assume that $M = \mathsf{der}(N)$ with $\mathcal{P} \vdash N : !\sigma$. Then we have $\mathrm{der}_{[\sigma]} \in \mathcal{L}(![\sigma], [\sigma])$ so that we can set $[M]_{\mathcal{P}} = \mathrm{der}_{[\sigma]}\, [N]_{\mathcal{P}} \in \mathcal{L}([\mathcal{P}], [\sigma])$.

Assume that $M = \mathsf{fix}\, x^{!\sigma}\, N$ so that $\mathcal{P}, x : !\sigma \vdash N : \sigma$. By inductive hypothesis we have $\mathrm{cur}([N]_{\mathcal{P},x:!\sigma}) \in \mathcal{L}(\underline{[\mathcal{P}]^!}, ![\sigma] \multimap [\sigma])$ and hence $(\mathrm{cur}([N]_{\mathcal{P},x:!\sigma}))^! \in \mathcal{L}^!([\mathcal{P}], !(![\sigma] \multimap [\sigma]))$ so that we can set $[M]_{\mathcal{P}} = \mathrm{fix}\, (\mathrm{cur}([N]_{\mathcal{P},x:!\sigma}))^! \in \mathcal{L}([\mathcal{P}], [\sigma])$.

Proposition 4. *If $\mathcal{P} \vdash V : \varphi$ and V is a value, then V is $\mathcal{L}$-regular, that is $[V]_{\mathcal{P}} \in \mathcal{L}^!([\mathcal{P}]^!, [\varphi]^!)$.*

The proof is a straightforward verification (in the definition of the interpretation of terms we have singled out the constructions which preserve $\mathcal{L}$-regularity). The main operational feature of $\mathcal{L}$-regular terms is that they enjoy the following substitutivity property.

Proposition 5 (Substitution Lemma). *Assume that $\mathcal{P}, x : \varphi \vdash M : \sigma$ and $\mathcal{P} \vdash N : \varphi$, and assume that N is $\mathcal{L}$-regular. Then we have*

$$[M\,[N/x]]_{\mathcal{P}} = [M]_{\mathcal{P},x:\varphi}\, ([\mathcal{P}] \otimes [N]_{\mathcal{P}})\ \mathrm{c}_{[\mathcal{P}]^!}$$

Proof. Induction on M using in an essential way the $\mathcal{L}$-regularity of N. Let us consider two cases to illustrate this point. Assume first that $M = R^!$ and $\sigma = !\tau$, with $\mathcal{P}, x : \varphi \vdash R : \tau$ so that $[R]_{\mathcal{P},x:\varphi} \in \mathcal{L}([\mathcal{P}] \otimes [\varphi], [\sigma])$. Then we have $[M\,[N/x]]_{\mathcal{P}} = [R\,[N/x]^!]_{\mathcal{P}} = ([R\,[N/x]]_{\mathcal{P}})^! = ([R]_{\mathcal{P},x:\varphi}\, ([\mathcal{P}] \otimes [N]_{\mathcal{P}})\ \mathrm{c}_{[\mathcal{P}]})^!$ by inductive hypothesis. We obtain the contended equation by applying Equation (3) and the fact that $[\mathcal{P}] \otimes [N]_{\mathcal{P}}$ is a coalgebra morphism since N is $\mathcal{L}$-regular, and the fact that $\mathrm{c}_{[\mathcal{P}]}$ is also a coalgebra morphism.

Let us also consider the case $M = \langle M_1, M_2 \rangle$ and $\sigma = \varphi_1 \otimes \varphi_2$ with $\mathcal{P}, x : \varphi \vdash M_i : \varphi_i$ for $i = 1, 2$ so that $[M_i]_{\mathcal{P},x:\varphi} \in \mathcal{L}([\mathcal{P}] \otimes [\varphi], [\varphi_i])$ for $i = 1, 2$. We have

$$\begin{aligned}
[M\,[N/x]]_{\mathcal{P}} &= [\langle M_1\,[N/x]\,, M_2\,[N/x]\rangle]_{\mathcal{P}} \\
&= ([M_1\,[N/x]]_{\mathcal{P}} \otimes [M_2\,[N/x]]_{\mathcal{P}})\ \mathrm{c}_{[\mathcal{P}]^! \otimes [\varphi]^!} \\
&= (([M_1]_{\mathcal{P},x:\varphi}\, ([\mathcal{P}] \otimes [N]_{\mathcal{P}})) \otimes ([M_2]_{\mathcal{P},x:\varphi}\, ([\mathcal{P}] \otimes [N]_{\mathcal{P}}))) \\
&\qquad\qquad \mathrm{c}_{[\mathcal{P}]^! \otimes [\varphi]^!} \\
&= ([M_1]_{\mathcal{P}} \otimes [M_2]_{\mathcal{P}})\ \mathrm{c}_{[\mathcal{P}]^!}\, ([\mathcal{P}] \otimes [N]_{\mathcal{P}})
\end{aligned}$$

using the inductive hypothesis, naturality of contraction in $\mathcal{L}^!$ and the fact that $([\mathcal{P}] \otimes [N]_{\mathcal{P}})$ is a coalgebra morphism since N is $\mathcal{L}$-regular. The other cases are handled similarly. □

Theorem 2 (Soundness). *If $\mathcal{P} \vdash M : \sigma$ and $M \to_{\mathsf{w}} M'$ then $[M]_{\mathcal{P}} = [M']_{\mathcal{P}}$.*

Proof. By induction on the derivation that $M \to_{\mathsf{w}} M'$, using the Substitution Lemma and the $\mathcal{L}$-regularity of values. □

3 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 now. This is certainly a distinctive feature of this model. Such a property does not hold for instance in coherence spaces. A nice outcome of these observations will be a very simple intersection typing system for Λ_{HP}.

Remark 5. As explained in [11], this semantics of LL has been independently discovered by several authors, see for instance [18, 31]. The same model is also considered in [23] Section 5.5.4 to interpret may non-determinism in CBPV. Our model allows therefore to interpret the corresponding non-deterministic extension of Λ_{HP} and the proof of adequacy of Section 3.4[5] can easily be extended to this language. This will be explained thoroughly in a forthcoming paper.

3.1 The Scott semantics of LL

We introduce a "linear" category **Polr** of preorders and relations. A preorder is a pair $S = (|S|, \leq_S)$ where $|S|$ is an at most countable set and $\leq_S$ (written $\leq$ when no confusion is possible) is a preorder relation on $|S|$. Given two preorders S and T, a morphism from S to T is a $f \subseteq |S| \times |T|$ such that, if $(a, b) \in f$ and $(a', b') \in |S| \times |T|$ satisfy $a \leq_S a'$ and $b' \leq_S b$, then $(a', b') \in f$. The relational composition of two morphisms is still a morphism and the identity morphism at S is $\mathrm{Id}_S = \{(a, a') \mid a' \leq_S a\}$.

Given an object S in **Polr**, the set $\mathsf{Ini}(S)$ of downwards closed subsets of $|S|$, 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).

Monoidal structure and cartesian product. The object 1 is $(\{*\}, =)$ and given preorders S and T we set $S \otimes T = (|S| \times |T|, \leq_S \times \leq_T)$. The tensor product of morphisms is defined in the obvious way. The isos defining the monoidal structure are easy to define. Then one defines $S \multimap T$ by $|S \multimap T| = |S| \times |T|$ and $(a', b') \leq_{S \multimap T} (a, b))$ if $a \leq a'$ and $b' \leq b$. The linear evaluation morphism $\mathrm{ev} \in \mathbf{Polr}((S \multimap T) \otimes S, T)$ is given by $\mathrm{ev} = \{(((a', b), a), b') \mid b' \leq b \text{ and } a' \leq a\}$.

[5] This adequacy result can probably be seen as a special case of the general adequacy results of [23].

If $f \in \mathbf{Polr}(U \otimes S, T)$ then $\mathrm{cur}(f) \in \mathbf{Polr}(U, S \multimap T)$ is defined by moving parentheses. This shows that **Polr** is closed. It is *-autonomous, with $\bot = 1$ as dualizing object. Observe that $S^\bot$ is simply $|S|$ equipped with $\geq_S$ as preorder relation $\leq_{S^\bot}$.

Given a countable family of objects $(S_i)_{i\in I}$, the cartesian product S is defined by $|S| = \bigcup_{i\in I}\{i\} \times |S_i|$ 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.

Exponential. One sets $!S = (\mathcal{P}_{\mathrm{fin}}(|S|), \leq)$ with $u \leq u'$ if $\forall a \in u\, \exists a' \in u'\ a \leq_S a'$ (where $\mathcal{P}_{\mathrm{fin}}(E)$ is the set of all finite subsets of E). Given $f \in \mathbf{Polr}(S,T)$, one defines $!f$ as $\{(u,v) \in |!S| \times |!T| \mid \forall b \in v\, \exists a \in u\ (a,b) \in f\}$. It is easy to prove that this defines a functor $\mathbf{Polr} \to \mathbf{Polr}$. Then one sets $\mathrm{der}_S = \{(u,a) \mid \exists a' \in u\ a \leq a'\} \in \mathbf{Polr}(!S, S)$ and $\mathrm{dig}_S = \{(u, \{u_1, \ldots, u_n\}) \mid u_1 \cup \cdots \cup u_n \leq_{!S} u\} \in \mathbf{Polr}(!S, !!S)$. 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 $\mathrm{m}^2_{S,T} = \{((u,v),w) \mid w \leq_{!(S\&T)} \{1\} \times u \cup \{2\} \times v\} \in \mathbf{Polr}(!S \otimes !T, !(S \,\&\, T))$.

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

3.2 The category of !-coalgebras

The first main observation is that each object of **Polr** has exactly one structure of !-coalgebra. The proofs will be given in a longer version of this paper.

Theorem 3. *Let S be an object of* **Polr**. *Then (S, p_S) is a !-coalgebra, where $\mathrm{p}_S = \{(a,u) \in |S| \times |!S| \mid \forall a' \in u\ a' \leq a\}$. Moreover, if P is a !-coalgebra, then $\mathrm{h}_P = \mathrm{p}_{\underline{P}}$.*

Morphisms of !-coalgebras. Now that we know that the objects of $\mathbf{Polr}^!$ are those of **Polr**, we turn our attention to morphisms. When we consider a preorder S as an object of $\mathbf{Polr}^!$, we always mean the object (S, p_S) described above.

With any preorder S, we have associated an ω-prime algebraic complete lattice $\mathsf{Ini}(S)$. We associate now with such a preorder an ω-algebraic cpo $\mathsf{Idl}(S)$ which is the ideal completion of S: an element of $\mathsf{Idl}(S)$ is a subset ξ of $|S|$ such that ξ is non-empty, downwards closed and directed (meaning that if $a, a' \in \xi$ then there is $a'' \in \xi$ such that $a, a' \leq_S a''$). We equip $\mathsf{Idl}(S)$ with the inclusion partial order relation.

Lemma 1. *For any preorder S, the partially ordered set $\mathsf{Idl}(S)$ is a cpo which has countably many compact elements. Moreover, for any $\xi \in \mathsf{Idl}(S)$, the set of compact elements $\xi_0 \in \mathsf{Idl}(S)$ such that $\xi_0 \subseteq \xi$ is directed and ξ is the lub of that set. In general, $\mathsf{Idl}(S)$ has no minimum element.*

Theorem 4. *Given preorders S and T, there is a bijective and functorial correspondence between $\mathbf{Polr}^!(S,T)$ and the set of Scott continuous functions from $\mathsf{Idl}(S)$ to $\mathsf{Idl}(T)$. Moreover, this correspondence is an order isomorphism when Scott-continuous functions are equipped with the usual pointwise ordering relation and $\mathbf{Polr}^!(S,T)$ is equipped with the inclusion order on relations.*

Let **Predom** be the category whose objects are the preorders and where a morphism from S to T is a Scott continuous function from $\mathsf{Idl}(S)$ to $\mathsf{Idl}(T)$. 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 S and T is $S \otimes T$ (and indeed, it is easy to check that $\mathsf{Idl}(S \otimes T) \simeq \mathsf{Idl}(S) \times \mathsf{Idl}(T)$) and their sum is $S \oplus T$ and indeed $\mathsf{Idl}(S \oplus T) \simeq \mathsf{Idl}(S) + \mathsf{Idl}(T)$, the disjoint union of the predomains $\mathsf{Idl}(S)$ and $\mathsf{Idl}(T)$. This predomain has no minimum element as soon as $|S|$ and $|T|$ are non-empty. Observe also that $\mathsf{Idl}(!S) = \mathsf{Ini}(S)$: one retrieves the fact that the Kleisli category of the ! comonad is the category of preorders and Scott continuous functions between the associated lattices.

Inclusions and embedding-retraction pairs. We define a category $\mathbf{Polr}_\subseteq$ as follows: the objects are those of **Polr** and $\mathbf{Polr}_\subseteq(S,T)$ is a singleton $\{\varphi_{S,T}\}$ if $|S| \subseteq |T|$ and $\forall a, a' \in |S|\ a \leq_S a' \Leftrightarrow a \leq_T a'$ (and then we write $S \subseteq T$) and is empty otherwise. So $(\mathbf{Polr}_\subseteq, \subseteq)$ is a partially ordered class. The functor F is defined as follows: if $S \subseteq T$ then $\varphi^+_{S,T} = \{(a,b) \in |S| \times |T| \mid b \leq_T a\}$ and $\varphi^-_{S,T} = \{(b,a) \in |T| \times |S| \mid a \leq_T b\}$; this definition is functorial and $\varphi^-_{S,T}\, \varphi^+_{S,T} = \mathrm{Id}_S$. The partially ordered class $\mathbf{Polr}_\subseteq$ is complete in the sense that any directed family of objects[6] $(S_i)_{i \in J}$ has a lub S given by $|S| = \cup_{i \in J} |S_i|$ and $a \leq_S a'$ if $a \leq_{S_i} a'$ for some i; we denote this preorder as $\cup_{i \in J} S_i$. The operations $\otimes$, $\oplus$ and $!\_$ are monotone and Scott-continuous operations on this partially ordered class.

Lemma 2. *Assume that $S_1 \subseteq S_2$ and let $f_i \in \mathbf{Polr}(S_i, T)$ for $i = 1, 2$. Then $f_1 = f_2\, \varphi^+_{S_1,S_2}$ iff $f_1 = f_2 \cap |S_1 \multimap T|$.*

Given a directed family $(S_i)_{i \in J}$ in $\mathbf{Polr}_\subseteq$ and setting $S = \cup_{i \in J} S_i$, one proves easily using Lemma 2 that the cone $(\varphi^+_{S_i,S} \in \mathbf{Polr}(S_i, S))_{i \in J}$ is a colimit cone in **Polr**. Consider indeed a family of morphisms $(f_i \in \mathbf{Polr}(S_i, T))_{i \in J}$ such that $i \leq j \Rightarrow f_i = f_j\, \varphi^+_{S_i,S_j}$, that is $f_i = f_j \cap |S_i \multimap T|$. Then $f = \cup_{i \in J} f_i$ is the unique element of $\mathbf{Polr}(S,T)$ such that $f\, \varphi^+_{S_i,S} = f_i$ for each $i \in J$. So the category $\mathbf{Polr}_\subseteq$ satisfies all the axioms of Section 2.2.

As explained in that section, this allows to define fixpoints of positive types. As a first example consider the type of flat natural numbers $\iota = \mathsf{Fix}\,\zeta \cdot (1 \oplus \zeta)$ where $1 = !\top$, so that $|1| = \{\emptyset\}$. Up to renaming we have $|[\iota]| = \mathbb{N}$ and $n \leq_{[\iota]} n'$ iff $n = n'$. The coalgebraic structure of this positive type is given by $\mathrm{h}_{[\iota]} =$

[6] Because we are dealing with a partially ordered class, we can replace general filtered categories with directed posets.

$\{(n,\emptyset) \mid n \in \mathbb{N}\} \cup \{(n,\{n\}) \mid n \in \mathbb{N}\}$. Consider now the type $\rho = \mathsf{Fix}\,\zeta \cdot (1 \oplus (\iota \otimes !\zeta))$ of lazy lists of flat natural numbers. The interpretation S of this type is the least fix-point of the continuous functor (that is, the Scott continuous functional) $[1 \oplus (\iota \otimes !\zeta)]_\zeta : \mathbf{Polr}_\subseteq \to \mathbf{Polr}_\subseteq$. So $|S| = \cup_{n=0}^{\infty} U_n$ where $(U_n)_{n\in\mathbb{N}}$ is the monotone sequence of sets defined by $U_0 = \emptyset$ and $U_{n+1} = \{\emptyset\} \cup (\mathbb{N} \times \mathcal{P}_{\mathrm{fin}}(U_n))$ (this is a disjoint union). The preorder relation on $|S|$ is given by: $\emptyset \leq_S a$ iff $a = \emptyset$ and $(n,u) \leq_S a$ iff $a = (n,u')$ and $\forall b \in u \exists b' \in u'\ b \leq_S b'$. This preorder relation defines the coalgebraic structure of this positive type.

3.3 Scott semantics as a typing system

It is interesting to present the Scott semantics of terms as a typing system, in the spirit of Coppo-Dezani Intersection Types, see [19]. 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 4, the intended meaning of these rules is made clear in Proposition 7.

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

$$\frac{\Phi \vdash M_1 : a_1 : \varphi_1 \qquad \Phi \vdash M_2 : a_2 : \varphi_2}{\Phi \vdash \langle M_1, M_2 \rangle : (a_1, a_2) : \varphi_1 \otimes \varphi_2}$$

$$\frac{\Phi \vdash M : a : \varphi_i}{\Phi \vdash \mathsf{in}_i M : (i,a) : \varphi_1 \oplus \varphi_2} \qquad \frac{\Phi \vdash M : (a,b) : \varphi \multimap \sigma \qquad \Phi \vdash N : a : \varphi}{\Phi \vdash \langle M \rangle N : b : \sigma}$$

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

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

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

$$\frac{\Phi, x : u : !\sigma \vdash M : a : \sigma \qquad \forall b \in u \quad \Phi \vdash \mathsf{fix}\, x^{!\sigma}\, M : b : \sigma}{\Phi \vdash \mathsf{fix}\, x^{!\sigma}\, M : a : \sigma}$$

Fig. 4. Scott Semantics as a Typing System

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 $i = 1, \ldots, n$.

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

Fig. 5. Interpretation of points as sets of terms in Λ_{HP}

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

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 thanks to Theorem 3; for instance the contraction morphism of an object S (seen as an object of $\mathbf{Polr}^!$) is $\mathrm{c}_S = \{(a,(a_1,a_2)) \mid a_i \leq a \text{ for } i = 1,2\}$.

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 $\to_{\mathsf{w}}$ starting from M terminates. We use a semantic method adapted *e.g.* from the presentation of the *reducibility* method in [19].

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 (and not on the type σ, whose definition involves fixpoints and is therefore not well-founded in general).

Given a positive type φ and $a \in [\varphi]$, we define $|a|_{\mathsf{v}}^{\varphi}$ 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 definitions are by mutual induction and are given in Figure 5. Observe that for a value V such that $\vdash V : \varphi$ and for $a \in [\varphi]$, the statements $V \in |a|^{\varphi}$ and $V \in |a|_{\mathsf{v}}^{\varphi}$ are equivalent because V is normal for the $\to_{\mathsf{w}}$ reduction.

Lemma 3. *If $M \to_{\mathsf{w}} M' \in |a|^\sigma$ 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 $M \to_{\mathsf{w}} M' \in |a|^{\sigma}$. Let $V \in |b|^{\varphi}_{\mathsf{v}}$, we have $\langle M\rangle V \to_{\mathsf{w}} \langle M'\rangle V$ and $\langle M'\rangle V \in |c|^{\tau}$ by definition of $|\varphi \multimap \tau|^{(b,c)}$. The announced property follows by inductive hypothesis. □

Lemma 4. *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 $|a|^{\sigma}_{\mathsf{v}} \supseteq |a'|^{\sigma}_{\mathsf{v}}$*

Theorem 5. *Let $\Phi = (x_1 : a_1 : \varphi_1, \ldots, x_k : a_k : \varphi_k)$ 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 $M\,[V_1/x_1, \ldots, V_k/x_k] \in |a|^{\sigma}$.*

So if $\vdash M : \varphi$ and $[M] \neq \emptyset$ we have $M \to_{\mathsf{w}}^{*} V$ for a value V. Let us say that two closed terms M_1, M_2 such that $\vdash M_i : \sigma$ for $i = 1, 2$ are observationally equivalent if for all closed term C of type $!\sigma \multimap 1$, $\langle C\rangle M_1^{!} \to_{\mathsf{w}}^{*} *$ iff $\langle C\rangle M_2^{!} \to_{\mathsf{w}}^{*} *$. As usual, Theorem 5 allows to prove that if $[M_1] = [M_2]$ then M_1 and M_2 are observationally equivalent.

Another important consequence of Theorem 5 is a "completeness" property of the weak reduction of Figure 2 which can be stated as follows. Let $\simeq$ be an equivalence relation on terms which contains $\to_{\mathsf{w}}^{*}$ and is compatible with the semantics of Section 2.3 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 $V \simeq V'$. Indeed $[V]$ is non-empty in the Scott model of Section 3, let $a \in [V]$. By our hypothesis we have $a \in [M]$ and hence $M \in |a|^{\varphi}$, meaning that $M \to_{\mathsf{w}}^{*} V'$ for some $V' \in |\varphi|^{a}_{\mathsf{v}}$. By our assumptions we have $M \simeq V'$ and hence $V' \simeq V$.

4 A fully polarized calculus

In Λ_{HP} positive types are interpreted as !-coalgebras, and general types are simply interpreted as objects of the underlying linear category: in some sense, this system is half-polarized and is intuitionistic for that reason. In a fully polarized system we would expect non-positive types to be negative, that is, linear duals of !-coalgebras. This system would feature syntactic constructions related to classical logic such as $\mathsf{call/cc}$, the price to pay being a more complicated encoding of data-types.

It is quite easy to turn our hierarchy of types (1), (2) into a polarized one:

$$\varphi, \psi, \ldots := !\sigma \mid \varphi \otimes \psi \mid \varphi \oplus \psi \mid \zeta \mid \mathsf{Fix}\,\zeta \cdot \varphi \quad \text{(positive)} \tag{4}$$

$$\sigma, \tau \ldots := ?\varphi \mid \varphi \multimap \sigma \mid \top \quad \text{(negative)} \tag{5}$$

Accordingly we introduce a polarized syntax for expressions featuring five mutually recursive syntactic categories.

$$
\begin{aligned}
P, Q \ldots &:= x \mid N^{!} \mid \langle P_1, P_2 \rangle \mid \mathsf{in}_i P \quad \text{(positive terms)} \\
M, N, \ldots &:= \mathsf{der}\, P \mid \lambda x^{\varphi}\, M \mid \mu\alpha^{\sigma}\, c \mid \mathsf{fix}\, x^{!\sigma}\, M \quad \text{(negative terms)} \\
\pi, \rho \ldots &:= \alpha \mid \eta^{!} \mid P \cdot \pi \quad \text{(positive contexts)} \\
\eta, \theta \ldots &:= \mathsf{der}\, \pi \mid \mathsf{pr}_i\, \eta \mid [\eta_1, \eta_2] \mid \tilde{\mu} x^{\varphi}\, c \quad \text{(negative contexts)} \\
c, d \ldots &:= P * \eta \mid M * \pi \quad \text{(commands, cuts)}
\end{aligned}
$$

Intuitively, positive terms correspond to data, negative terms to programs, negative contexts to patterns (apart from the negative context $\tilde{\mu} x^{\varphi}\, c$ which generalizes the concept of "closure") and positive contexts to evaluation environments.

The typing rules correspond to a large fragment of LLP, see [21][7], and are given in Figure 6.

Let us say that an expression e is well typed in typing contexts $\mathcal{P} = (x_1 : \varphi_1, \ldots, x_n : \varphi_n)$, $\mathcal{N} = (\alpha_1 : \sigma_1, \ldots, \alpha_k : \sigma_k)$ if e is a positive term P and $\mathcal{P} \vdash P : \varphi \,|\, \mathcal{N}$ for some type φ, if e is a negative term M and $\mathcal{P} \vdash M : \sigma \,|\, \mathcal{N}$ for some type σ, if e is a positive context π and $\mathcal{P} \,|\, \pi : \sigma \vdash \mathcal{N}$ for some type σ, if e is a negative context η and $\mathcal{P} \,|\, \eta : \varphi \vdash \mathcal{N}$ for some type φ and if e is a command c and $\mathcal{P} \vdash c \,|\, \mathcal{N}$. In the four first cases, φ (resp. σ) is *the type of* e. Observe that, when it exists, this type is completely determined by e, $\mathcal{P}$ and $\mathcal{N}$ (the typing rules are syntax-directed).

4.1 Operational semantics

The weak reduction rules are given in Figure 7. All redexes are commands and it is crucial to observe that there are no critical pairs. Specifically, there is no command which is simultaneously of shape $M * \pi$ and $P * \tilde{\mu} x^{\varphi}\, c$ because in the former the term is negative whereas it is positive in the latter. In particular the "Lafont critical pair" $\mu\alpha^{\theta}\, c * \tilde{\mu} x^{\theta}\, d$ cannot occur since θ cannot be positive and negative!.

Remark 6. In this weak reduction paradigm, we only reduce commands. A sequence of reduction alternates therefore sequences of *positive commands* of shape $P * \eta$ where a piece of data P is explored by a pattern η with sequences of *negative commands* $M * \pi$ where a program M is executed in an evaluation context π. The transition from the execution phase to the pattern-matching phase is realized by the reduction rule (6) and the converse by (10). We retrieve the basic idea of *focusing* of [1] and of Ludics, [16], that "positive" means passive and "negative", active (many other authors should be mentioned here of course).

[7] In a Sequent Calculus presentation, with double-sided sequents contrarily to most presentations of LLP in the literature. Our syntax is based on the $\lambda\mu\tilde{\mu}$ presentation of sequent calculus-oriented classical lambda-calculi due to [7].

$$\frac{}{\mathcal{P}, x : \varphi \vdash x : \varphi \,|\, \mathcal{N}} \qquad \frac{\mathcal{P} \vdash N : \sigma \,|\, \mathcal{N}}{\mathcal{P} \vdash N^{!} : !\sigma \,|\, \mathcal{N}} \qquad \frac{\mathcal{P} \vdash P_1 : \varphi_1 \,|\, \mathcal{N} \qquad \mathcal{P} \vdash P_1 : \varphi_2 \,|\, \mathcal{N}}{\mathcal{P} \vdash \langle P_1, P_2 \rangle : \varphi_1 \otimes \varphi_2 \,|\, \mathcal{N}}$$

$$\frac{\mathcal{P} \vdash P : \varphi_i \,|\, \mathcal{N}}{\mathcal{P} \vdash \mathsf{in}_i P : \varphi_1 \oplus \varphi_2 \,|\, \mathcal{N}} \qquad \frac{\mathcal{P} \vdash P : \varphi \,|\, \mathcal{N}}{\mathcal{P} \vdash \mathsf{der}\, P : ?\varphi \,|\, \mathcal{N}} \qquad \frac{\mathcal{P}, x : \varphi \vdash M : \sigma \,|\, \mathcal{N}}{\mathcal{P} \vdash \lambda x^{\varphi} M : \varphi \multimap \sigma \,|\, \mathcal{N}}$$

$$\frac{\mathcal{P} \vdash c \,|\, \alpha : \sigma, \mathcal{N}}{\mathcal{P} \vdash \mu\alpha^{\sigma} c : \sigma \,|\, \mathcal{N}} \qquad \frac{\mathcal{P}, x : !\sigma \vdash M : \sigma \,|\, \mathcal{N}}{\mathcal{P} \vdash \mathsf{fix}\, x^{!\sigma} M : \sigma \,|\, \mathcal{N}}$$

$$\frac{}{\mathcal{P} \,|\, \alpha : \sigma \vdash \alpha : \sigma, \mathcal{N}} \qquad \frac{\mathcal{P} \,|\, \eta : \varphi \vdash \mathcal{N}}{\mathcal{P} \,|\, \eta^{!} : ?\varphi \vdash \mathcal{N}} \qquad \frac{\mathcal{P} \vdash P : \varphi \,|\, \mathcal{N} \qquad \mathcal{P} \,|\, \pi : \sigma \vdash \mathcal{N}}{\mathcal{P} \,|\, P \cdot \pi : \varphi \multimap \sigma \vdash \mathcal{N}}$$

$$\frac{\mathcal{P} \,|\, \pi : \sigma \vdash \mathcal{N}}{\mathcal{P} \,|\, \mathsf{der}\, \pi : !\sigma \vdash \mathcal{N}} \qquad \frac{\mathcal{P} \,|\, \eta : \varphi_i \vdash \mathcal{N}}{\mathcal{P} \,|\, \mathsf{pr}_i \eta : \varphi_1 \otimes \varphi_2 \vdash \mathcal{N}}$$

$$\frac{\mathcal{P} \,|\, \eta_1 : \varphi_1 \vdash \mathcal{N} \qquad \mathcal{P} \,|\, \eta_2 : \varphi_2 \vdash \mathcal{N}}{\mathcal{P} \,|\, [\eta_1, \eta_2] : \varphi_1 \oplus \varphi_2 \vdash \mathcal{N}} \qquad \frac{\mathcal{P}, x : \varphi \vdash c \,|\, \mathcal{N}}{\mathcal{P} \,|\, \tilde{\mu} x^{\varphi} c : \varphi \vdash \mathcal{N}}$$

$$\frac{\mathcal{P} \vdash P : \varphi \,|\, \mathcal{N} \qquad \mathcal{P} \,|\, \eta : \varphi \vdash \mathcal{N}}{\mathcal{P} \vdash P * \eta \,|\, \mathcal{N}} \qquad \frac{\mathcal{P} \vdash M : \sigma \,|\, \mathcal{N} \qquad \mathcal{P} \,|\, \pi : \sigma \vdash \mathcal{N}}{\mathcal{P} \vdash M * \pi \,|\, \mathcal{N}}$$

Fig. 6. Typing rules for Λ_{LLP}: positive terms, negative terms, positive contexts, negative contexts and commands

$$\mathsf{der}\, P * \eta^{!} \to_{\mathsf{w}} P * \eta \tag{6}$$
$$\lambda x^{\varphi} M * P \cdot \pi \to_{\mathsf{w}} M\,[P/x] * \pi \tag{7}$$
$$\mu\alpha^{\sigma} c * \pi \to_{\mathsf{w}} c\,[\pi/\alpha] \tag{8}$$
$$\mathsf{fix}\, x^{!\sigma} M * \pi \to_{\mathsf{w}} M \left[(\mathsf{fix}\, x^{!\sigma} M)^{!} / x \right] * \pi \tag{9}$$
$$M^{!} * \mathsf{der}\, \pi \to_{\mathsf{w}} M * \pi \tag{10}$$
$$\langle P_1, P_2 \rangle * \mathsf{pr}_i \eta \to_{\mathsf{w}} P_i * \eta \tag{11}$$
$$\mathsf{in}_i P * [\eta_1, \eta_2] \to_{\mathsf{w}} P * \eta_i \tag{12}$$
$$P * \tilde{\mu} x^{\varphi} c \to_{\mathsf{w}} c\,[P/x] \tag{13}$$

Fig. 7. Reduction rules for Λ_{LLP}

We also consider a general reduction relation $\to$ on expressions which is defined by allowing the application of the rules of Figure 7 *anywhere* in an expression as well as the two following $\mu\eta$ reduction rules: $\mu\alpha^{\sigma}\,(M * \alpha) \to M$ if α does not occur free in M and $\tilde{\mu} x^{\varphi}\,(x * \eta) \to \eta$ if x does not occur free in η.

Proposition 8. *If e is typable in contexts $\mathcal{P}$, $\mathcal{N}$ and $e \to e'$ then e' is typable in contexts $\mathcal{P}$, $\mathcal{N}$, belongs to the same syntactic category as e and has the same type as e (when it applies).*

$$[\zeta_i]_{\boldsymbol{\zeta}}(\boldsymbol{P}) = P_i \qquad [!\sigma]_{\boldsymbol{\zeta}}(\boldsymbol{P}) = !(\underline{[\![\sigma]\!]_{\boldsymbol{\zeta}}(\boldsymbol{P})}^{\perp})$$
$$[\varphi \otimes \psi]_{\boldsymbol{\zeta}}(\boldsymbol{P}) = [\varphi]_{\boldsymbol{\zeta}}(\boldsymbol{P}) \otimes [\psi]_{\boldsymbol{\zeta}}(\boldsymbol{P})$$
$$[\varphi \oplus \psi]_{\boldsymbol{\zeta}}(\boldsymbol{P}) = [\varphi]_{\boldsymbol{\zeta}}(\boldsymbol{P}) \oplus [\psi]_{\boldsymbol{\zeta}}(\boldsymbol{P})$$
$$[\mathsf{Fix}\,\zeta \cdot \varphi]_{\boldsymbol{\zeta}} = \mathrm{Fix}([\varphi]_{\boldsymbol{\zeta},\zeta}) \qquad [\![\top]\!]_{\boldsymbol{\zeta}}(\boldsymbol{P}) = 0$$
$$[\![?\varphi]\!]_{\boldsymbol{\zeta}}(\boldsymbol{P}) = !(\underline{[\varphi]_{\boldsymbol{\zeta}}(\boldsymbol{P})}^{\perp}) \qquad [\![\varphi \multimap \sigma]\!]_{\boldsymbol{\zeta}}(\boldsymbol{P}) = [\varphi]_{\boldsymbol{\zeta}}(\boldsymbol{P}) \otimes [\![\sigma]\!]_{\boldsymbol{\zeta}}(\boldsymbol{P})$$

Fig. 8. Semantics of types in Λ_{LLP}

The proof is a straightforward verification. As usual one has first to state and prove a Substitution Lemma.

The reduction $\to$ on Λ_{LLP} enjoys Church-Rosser as we shall see in a longer paper. The denotational semantics that we outline now gives us another proof that this calculus is sound.

4.2 Denotational semantics

Assume to be given a model of LL $\mathcal{L}$ as specified in Section 2. With any positive type φ, negative type σ and sequence of pairwise distinct type variables $\boldsymbol{\zeta} = (\zeta_1, \ldots, \zeta_n)$ containing all free variables of φ and σ, we associate the continuous functors $[\varphi]_{\boldsymbol{\zeta}}, [\![\sigma]\!]_{\boldsymbol{\zeta}} : (\mathcal{L}^!_{\subseteq})^n \to \mathcal{L}^!_{\subseteq}$ defined in Figure 8 on objects, the definition on morphisms being similar[8]. With any positive terms and contexts P and π with $\mathcal{P} \vdash P : \varphi \,|\, \mathcal{N}$ and $\mathcal{P} \,|\, \pi : \sigma \vdash \mathcal{N}$ we associate coalgebra morphisms $[P]_{\mathcal{P},\mathcal{N}} \in \mathcal{L}^!([\mathcal{P}] \otimes [\![\mathcal{N}]\!], [\varphi])$ and $[\pi]_{\mathcal{P},\mathcal{N}} \in \mathcal{L}^!([\mathcal{P}] \otimes [\![\mathcal{N}]\!], [\![\sigma]\!])$ and with any negative terms and contexts M and η with $\mathcal{P} \vdash M : \sigma \,|\, \mathcal{N}$ and $\mathcal{P} \,|\, \eta : \varphi \vdash \mathcal{N}$ we associate morphisms $[M]_{\mathcal{P},\mathcal{N}} \in \mathcal{L}(\underline{[\mathcal{P}]} \otimes \underline{[\![\mathcal{N}]\!]}, \underline{[\![\sigma]\!]}^{\perp})$ and $[\eta]_{\mathcal{P},\mathcal{N}} \in \mathcal{L}(\underline{[\mathcal{P}]} \otimes \underline{[\![\mathcal{N}]\!]}, \underline{[\varphi]}^{\perp})$. Last, with any command c such that $\mathcal{P} \vdash c \,|\, \mathcal{N}$ we associate a morphism $[c]_{\mathcal{P},\mathcal{N}} \in \mathcal{L}(\underline{[\mathcal{P}]} \otimes \underline{[\![\mathcal{N}]\!]}, \perp)$.

The interpretation of positive terms is defined as for Λ_{HP} (see 2.3). Negative terms: the interpretation of $\mathsf{der}\,P$ uses the dereliction morphism in $\mathcal{L}(\underline{[\varphi]}, ?\underline{[\varphi]})$, the interpretation of $\lambda x^{\varphi} M$ and of $\mathsf{fix}\, x^{!\sigma} M$ is defined as in Λ_{HP} (replacing $[\sigma]$ with $\underline{[\![\sigma]\!]}^{\perp}$) and we provide the interpretation of $\mu\alpha^{\sigma} c$: assume that $\mathcal{P} \vdash c \,|\, \mathcal{N}, \alpha : \sigma$ so that $[c]_{\mathcal{P},\mathcal{N},\alpha:\sigma} \in \mathcal{L}(\underline{[\mathcal{P}]} \otimes \underline{[\![\mathcal{N}]\!]} \otimes \underline{[\![\sigma]\!]}, \perp)$ and we set $[\mu\alpha^{\sigma}\, c]_{\mathcal{P},\mathcal{N}} = \mathrm{cur}\,[c]_{\mathcal{P},\mathcal{N},\alpha:\sigma}$. Positive contexts: we deal only with one case. Assume that $\mathcal{P} \vdash P : \varphi \,|\, \mathcal{N}$ and $\mathcal{P} \,|\, \pi : \sigma \vdash \mathcal{N}$. We have $[P]_{\mathcal{P},\mathcal{N}} \in \mathcal{L}^!([\mathcal{P}] \otimes [\![\mathcal{N}]\!], [\varphi])$ and $[\pi]_{\mathcal{P},\mathcal{N}} \in \mathcal{L}^!([\mathcal{P}] \otimes [\![\mathcal{N}]\!], [\![\sigma]\!])$ so that we can set $[P \cdot \pi]_{\mathcal{P},\mathcal{N}} = ([P]_{\mathcal{P},\mathcal{N}} \otimes [\pi]_{\mathcal{P},\mathcal{N}})\ \mathrm{c}_{[\mathcal{P}]\otimes[\![\mathcal{N}]\!]}$ whose codomain is $[\varphi] \otimes [\![\sigma]\!] = [\![\varphi \multimap \sigma]\!]$ as required. The interpretation of α uses a projection and the interpretation of $\eta^!$ uses a promotion. Negative contexts: assume that $\mathcal{P} \,|\, \pi : \sigma \vdash \mathcal{N}$, then $[\pi]_{\mathcal{P},\mathcal{N}} \in \mathcal{L}^!([\mathcal{P}] \otimes [\![\mathcal{N}]\!], [\![\sigma]\!])$ so we set

[8] Notice that the interpretation of a negative type is actually the semantics of its linear negation; we adopt this convention in order to avoid the explicit introduction of negative objects in the model. This is possible because we have assumed that $\mathcal{L}$ is $*$-autonomous.

$$\zeta^+ = \zeta \qquad (!\sigma)^+ = !(\sigma^-)$$
$$(\varphi_1 \otimes \varphi_2)^+ = \varphi_1^+ \otimes \varphi_2^+ \qquad (\varphi_1 \oplus \varphi_2)^+ = \varphi_1^+ \oplus \varphi_2^+$$
$$(\mathsf{Fix}\,\zeta \cdot \varphi)^+ = \mathsf{Fix}\,\zeta \cdot \varphi^+$$
$$\varphi^- = ?(\varphi^+) \qquad (\varphi \multimap \sigma)^- = \varphi^+ \multimap \sigma^-$$

Fig. 9. Translation of types

$[\mathsf{der}\,\pi]_{\mathcal{P},\mathcal{N}} = \mathrm{der}'_{\underline{[\![\sigma]\!]}}\,[\pi]_{\mathcal{P},\mathcal{N}} \in \mathcal{L}(\underline{[\mathcal{P}]} \otimes \underline{[\![\mathcal{N}]\!]}, ?\underline{[\![\sigma]\!]})$ which makes sense since $?\underline{[\![\sigma]\!]} = \underline{[!\sigma]}^\perp$ (see Figure 8). The interpretation of $\mathsf{pr}_i\,\eta$ uses $\mathrm{pr}_i^\perp \in \mathcal{L}(\underline{[\varphi_i]}^\perp, (\underline{[\varphi_1]} \otimes \underline{[\varphi_2]})^\perp)$. To define $[[\eta_1, \eta_2]]_{\mathcal{P},\mathcal{N}}$ we simply use the pairing operation associated with the cartesian product & of $\mathcal{L}$ (warning: *not* of $\mathcal{L}^!$!) which is the linear "De Morgan" dual of the coproduct $\oplus$ of $\mathcal{L}$. The interpretation of $\tilde{\mu}x^\varphi\,c$ uses a linear curryfication. Commands: assume that $\mathcal{P} \vdash P : \varphi \,|\, \mathcal{N}$ and $\mathcal{P} \,|\, \eta : \varphi \vdash \mathcal{N}$ so that $[P]_{\mathcal{P},\mathcal{N}} \in \mathcal{L}^!([\mathcal{P}] \otimes [\![\mathcal{N}]\!], [\varphi]) \subseteq \mathcal{L}(\underline{[\mathcal{P}]} \otimes \underline{[\![\mathcal{N}]\!]}, \underline{[\varphi]})$ and $[\eta]_{\mathcal{P},\mathcal{N}} \in \mathcal{L}(\underline{[\mathcal{P}]} \otimes \underline{[\![\mathcal{N}]\!]}, \underline{[\varphi]}^\perp)$ and we set $[P * \eta]_{\mathcal{P},\mathcal{N}} = \mathrm{ev}\,([\eta]_{\mathcal{P},\mathcal{N}} \otimes \overline{[P]}_{\mathcal{P},\mathcal{N}})\ \mathrm{c}_{[\mathcal{P}]\otimes[\![\mathcal{N}]\!]}$. The interpretation of $M * \pi$ is similar.

Proposition 9. *Assume that $\mathcal{P} \vdash P : \varphi \,|\, \mathcal{N}$ and that e is a well-typed expression in typing contexts $x : \varphi, \mathcal{P}, \mathcal{N}$. Then we have $[e\,[P/x]]_{\mathcal{P},\mathcal{N}} = [e]_{x:\varphi,\mathcal{P},\mathcal{N}}\,([P]_{\mathcal{P},\mathcal{N}} \otimes \mathrm{Id})\ \mathrm{c}_{[\mathcal{P}]\otimes[\![\mathcal{N}]\!]}$. Assume that $\mathcal{P} \,|\, \pi : \sigma \vdash \mathcal{N}$ and that e is a well-typed expression in typing contexts $\mathcal{P}$, $\mathcal{N}, \alpha : \sigma$. Then we have $[e\,[\pi/\alpha]]_{\mathcal{P},\mathcal{N}} = [e]_{\mathcal{P},\mathcal{N},\alpha:\sigma}\,(\mathrm{Id} \otimes [\pi]_{\mathcal{P},\mathcal{N}})\ \mathrm{c}_{[\mathcal{P}]\otimes[\![\mathcal{N}]\!]}$.*

Theorem 6. *If e is a well-typed expression in typing contexts $\mathcal{P}$, $\mathcal{N}$ and if $e \to e'$ then $[e]_{\mathcal{P},\mathcal{N}} = [e']_{\mathcal{P},\mathcal{N}}$.*

The proof is routine, using Proposition 9.

Given a negative type σ, observe that $\Omega^\sigma = \mathsf{fix}\,x^{!\sigma}\,(\mu\alpha^\sigma\,(x * \mathsf{der}\,\alpha))$ is a closed negative term of type σ. A consequence of Theorem 6 is that Λ_{LLP} is sound in the sense that the reflexive and transitive closure of $\to$ does not equate *e.g.* the two booleans $\mathsf{in}_1(\Omega^\top)^!$ and $\mathsf{in}_2(\Omega^\top)^!$ (closed positive terms of type $!\top \oplus !\top$). Indeed it is easy to build models where these constants have distinct interpretations (for instance the Scott model **Polr** of Section 3).

4.3 Translating Λ_{HP} into Λ_{LLP}

With any positive type φ of Λ_{HP} we associate a positive type φ^+ of Λ_{LLP} and with any general type σ we associate a negative type of Λ_{LLP}. The translation does almost nothing apart adding a few "?" to make sure that σ^- is negative, see Figure 9. Given a Λ_{HP} typing context $\mathcal{P} = (x_1 : \varphi_1, \ldots, x_n : \varphi_n)$, we set $\mathcal{P}^+ = (x_1 : \varphi_1^+, \ldots, x_n : \varphi_n^+)$. With any term M of Λ_{HP} such that $\mathcal{P} \vdash M : \sigma$ one associates a term M^- of Λ_{LLP} such that $\mathcal{P}^+ \vdash M^- : \sigma^- \,|\,$. We give two cases of this translation, the complete translation will be given in a longer paper.

If $M = \langle N\rangle R$ with $\mathcal{P} \vdash N : \varphi \multimap \sigma$ and $\mathcal{P} \vdash R : \varphi$ then by inductive hypothesis, $\mathcal{P}^+ \vdash N^- : \varphi^+ \multimap \sigma^- \,|$ and $\mathcal{P}^+ \vdash R^- : ?\varphi^+ \,|$. We have $\mathcal{P}^+, x : \varphi^+ \,|\, x\cdot\alpha : \varphi^+ \multimap \sigma^- \vdash \alpha : \sigma^-$, hence $\mathcal{P}^+, x : \varphi^+ \vdash N^- * (x\cdot\alpha) \,|\, \alpha : \sigma^-$. Therefore we have $\mathcal{P}^+ \,|\, \mu x^{\varphi^+}\, N^- * (x\cdot\alpha) : \varphi^+ \vdash \alpha : \sigma^-$ so that $\mathcal{P}^+ \,|\, (\mu x^{\varphi^+}\, N^- * (x\cdot\alpha))^{!} : ?\varphi^+ \vdash \alpha : \sigma^-$ and we set $\langle N\rangle R^- = \mu\alpha^{\sigma^-}\, (R^- * (\mu x^{\varphi^+}\, N^- * (x\cdot\alpha))^{!})$.

If $M = \langle M_1, M_2\rangle$ there are two possible translations (similar phenomena occur in CPS translation, see for instance [29]), a *left first* translation and a *right first* translation. We give the first one, the other one being obtained by swapping the roles of M_1 and M_2. We assume that $\mathcal{P} \vdash M_i : \varphi_i$ and hence $\mathcal{P}^+ \vdash M_i^- : ?\varphi_i^+ \,|$ for $i = 1, 2$. We have $\mathcal{P}^+, x_2 : \varphi_2^+ \,|\, \tilde{\mu} x_1^{\varphi_1^+}\, \mathsf{der}\,\langle x_1, x_2\rangle * \alpha : \varphi_1^+ \vdash \alpha : ?(\varphi_1^+ \otimes \varphi_2^+)$, hence $\mathcal{P}^+, x_2 : \varphi_2^+ \vdash M_1^- * (\tilde{\mu} x_1^{\varphi_1^+}\, \mathsf{der}\,\langle x_1, x_2\rangle * \alpha)^{!} \,|\, \alpha : ?(\varphi_1^+ \otimes \varphi_2^+)$ and hence we set $\langle M_1, M_2\rangle^- = \mu\alpha^{?(\varphi_1^+ \otimes \varphi_2^+)}\, M_2^- * \left(\tilde{\mu} x_2^{\varphi_2^+}\, M_1^- * (\tilde{\mu} x_1^{\varphi_1^+}\, \mathsf{der}\,\langle x_1, x_2\rangle * \alpha)^{!}\right)^{!}$.

Given a Λ_{HP} value V such that $\mathcal{P} \vdash V : \varphi$, one defines straightforwardly a positive term V^+ of Λ_{LLP} such that $\mathcal{P}^+ \vdash V^+ : \varphi^+ \,|$: one sets $(M^!)^+ = (M^-)^!$, $\langle V_1, V_2\rangle^+ = \langle V_1^+, V_1^+\rangle$ and similarly for the other cases.

Lemma 5. *If $\mathcal{P} \vdash V : \varphi$ in Λ_{HP} and if V is a value then $V^- \to_{\mathsf{w}}^* \mathsf{der}\,(V^+)$.*

Lemma 6. *If $\mathcal{P}, x : \varphi \vdash M : \sigma$ and $\mathcal{P} \vdash V : \varphi$ in Λ_{HP} (with V being a value) then $M\,[V/x]^- \to_{\mathsf{w}}^* M^-\,[V^+/x]$.*

Proof. Induction on M, using Lemma 5 when $M = x$. □

Theorem 7. *If $\mathcal{P} \vdash M : \sigma$ in Λ_{HP} then $\mathcal{P}^+ \vdash M^- : \sigma^- \,|$. Moreover, if $M \to_{\mathsf{w}} M'$ then there is a negative term R of Λ_{LLP} such that $M^- \to^* R$ and $M'^- \to^* R$.*

As a consequence, using confluence of $\to^*$ in Λ_{LLP}, one proves that, if $M \to^* M'$ in Λ_{HP} there is a negative term R of Λ_{LLP} such that $M^- \to^* R$ and $M'^- \to^* R$ (induction on the length of the reduction $M \to^* M'$). This shows that Λ_{HP} embeds in Λ_{LLP} by a translation $M \mapsto M^-$ which is compatible with the operational semantics. In the long version of this paper, we will also describe a simple relation between the semantics of M and of M^-.

Conclusion. The half-polarized and fully polarized calculi proposed in this paper admit LL-based models featuring non-trivial computational effects such as non-deterministic or probabilistic computations [9] (with full abstraction properties [12]). One can accommodate other effects by extending the language with monad type constructions interpreted as monads acting on $\mathcal{L}$ and featuring a strength, in the spirit of [17], as we will explain in a forthcoming paper.

I would like to thank the referees for their insightful suggestions and comments. I also would like to mention the work [6] which shares some features with ours (in particular an emphasis on the resource modality "!"); these papers have been written independently and simultaneously.

This work has been partly funded by the French-Chinese project ANR-11-IS02-0002 and NSFC 61161130530 *Locali.*

References

1. Jean-Marc Andreoli. Logic programming with focusing proofs in linear logic. *Journal of Logic and Computation*, 2(3):297–347, 1992.
2. P. N. Benton and Philip Wadler. Linear logic, monads and the lambda calculus. In *Proceedings, 11th Annual IEEE Symposium on Logic in Computer Science, New Brunswick, New Jersey, USA, July 27-30, 1996*, pages 420–431. IEEE Computer Society, 1996.
3. Gavin Bierman. What is a categorical model of intuitionistic linear logic? In Mariangiola Dezani-Ciancaglini and Gordon D. Plotkin, editors, *Proceedings of the second Typed Lambda-Calculi and Applications conference*, volume 902 of *Lecture Notes in Computer Science*, pages 73–93. Springer-Verlag, 1995.
4. Alberto Carraro and Giulio Guerrieri. A Semantical and Operational Account of Call-by-Value Solvability. In Anca Muscholl, editor, *Foundations of Software Science and Computation Structures - 17th International Conference, FOSSACS 2014, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2014, Grenoble, France, April 5-13, 2014, Proceedings*, volume 8412 of *Lecture Notes in Computer Science*, pages 103–118. Springer, 2014.
5. Pierre-Louis Curien. Call-By-Push-Value in system L style. Unpublished note, 2015.
6. Pierre-Louis Curien, Marcelo Fiore, and Guillaume Munch-Maccagnoni. A Theory of Effects and Resources: Adjunction Models and Polarised Calculi. In *Proceedings of POPL 2016*, 2015. To appear.
7. Pierre-Louis Curien and Hugo Herbelin. The duality of computation. In Martin Odersky and Philip Wadler, editors, *Proceedings of the Fifth ACM SIGPLAN International Conference on Functional Programming (ICFP '00), Montreal, Canada, September 18-21, 2000*, pages 233–243. ACM, 2000.
8. Pierre-Louis Curien and Guillaume Munch-Maccagnoni. The Duality of Computation under Focus. In Cristian S. Calude and Vladimiro Sassone, editors, *Theoretical Computer Science - 6th IFIP TC 1/WG 2.2 International Conference, TCS 2010, Held as Part of WCC 2010, Brisbane, Australia, September 20-23, 2010. Proceedings*, volume 323 of *IFIP Advances in Information and Communication Technology*, pages 165–181. Springer, 2010.
9. Vincent Danos and Thomas Ehrhard. Probabilistic coherence spaces as a model of higher-order probabilistic computation. *Information and Computation*, 152(1):111–137, 2011.
10. Jeff Egger, Rasmus Ejlers Møgelberg, and Alex Simpson. The enriched effect calculus: syntax and semantics. *Journal of Logic and Computation*, 24(3):615–654, 2014.
11. Thomas Ehrhard. The Scott model of Linear Logic is the extensional collapse of its relational model. *Theoretical Computer Science*, 424:20–45, 2012.
12. Thomas Ehrhard, Christine Tasson, and Michele Pagani. Probabilistic coherence spaces are fully abstract for probabilistic PCF. In Suresh Jagannathan and Peter Sewell, editors, *POPL*, pages 309–320. ACM, 2014.
13. Marcelo P. Fiore and Gordon D. Plotkin. An Axiomatization of Computationally Adequate Domain Theoretic Models of FPC. In *Proceedings of the Ninth Annual Symposium on Logic in Computer Science (LICS '94), Paris, France, July 4-7, 1994*, pages 92–102. IEEE Computer Society, 1994.
14. Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

15. Jean-Yves Girard. A new constructive logic: classical logic. *Mathematical Structures in Computer Science*, 1(3):225–296, 1991.
16. Jean-Yves Girard. Locus Solum. *Mathematical Structures in Computer Science*, 11(3):301–506, 2001.
17. Masahito Hasegawa. Linearly Used Effects: Monadic and CPS Transformations into the Linear Lambda Calculus. In Zhenjiang Hu and Mario Rodríguez-Artalejo, editors, *Functional and Logic Programming, 6th International Symposium, FLOPS 2002, Aizu, Japan, September 15-17, 2002, Proceedings*, volume 2441 of *Lecture Notes in Computer Science*, pages 167–182. Springer-Verlag, 2002.
18. 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.
19. 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).
20. Jean-Louis Krivine. A general storage theorem for integers in call-by-name λ-calculus. *Theoretical Computer Science*, 129:79–94, 1994.
21. 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.
22. Paul Blain Levy. Adjunction Models For Call-By-Push-Value With Stacks. *Electronic Notes in Theoretical Computer Science*, 69:248–271, 2002.
23. Paul Blain Levy. *Call-By-Push-Value: A Functional/Imperative Synthesis*, volume 2 of *Semantics Structures in Computation*. Springer-Verlag, 2004.
24. John Maraist, Martin Odersky, David N. Turner, and Philip Wadler. Call-by-name, call-by-value, call-by-need and the linear lambda calculus. *Theoretical Computer Science*, 228(1-2):175–210, 1999.
25. 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.
26. Paul-André Melliès. Categorical semantics of linear logic. *Panoramas et Synthèses*, 27, 2009.
27. Eugenio Moggi. Computational lambda-calculus and monads. In *Proceedings of the 4th Annual IEEE Symposium on Logic in Computer Science*. IEEE Computer Society, 1989.
28. Robert Seely. Linear logic, star-autonomous categories and cofree coalgebras. *Applications of categories in logic and computer science*, 92, 1989.
29. 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.
30. 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.
31. 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.