Non-Linearity as the Metric Completion of Linearity

Damiano Mazza

CNRS, UMR 7030, LIPN, Université Paris 13, Sorbonne Paris Cité
Damiano.Mazza@lipn.univ-paris13.fr

Abstract. We summarize some recent results showing how the lambda-calculus may be obtained by considering the metric completion (with respect to a suitable notion of distance) of a space of affine lambda-terms, i.e., lambda-terms in which abstractions bind variables appearing at most once. This formalizes the intuitive idea that multiplicative additive linear logic is "dense" in full linear logic (in fact, a proof-theoretic version of the above-mentioned construction is also possible). We argue that thinking of non-linearity as the "limit" of linearity gives an interesting point of view on well-known properties of the lambda-calculus and its relationship to computational complexity (through lambda-calculi whose normalization is time-bounded).

1 Linearity and approximations

The concept of linearity in logic and computer science, introduced over two decades ago [12], has now entered firmly into the "toolbox" of proof theorists and functional programming language theorists. It is present, in one way or another, in a broad range of contexts, such as: denotational semantics [11], games semantics [22] and categorical semantics [8]; computational interpretations of classical logic [18, 9]; optimal implementation of functional programming languages [3, 19]; the theory of explicit substitutions [2]; higher-order languages for probabilistic [10] and quantum computation [24]; typing systems for polynomial-time [4], non-size-increasing [14] and resource-aware computation [17]; and even concurrency theory [6, 15].

Technically, linearity imposes a severe restriction on the behavior of programs: data must be accessed exactly once. Its cousin *affinity*, which is more relevant for the purposes of this text, slightly relaxes the constraint: although data may be discarded, it may nevertheless be accessed at most once. In any case, linearity and affinity forbid re-use, forcing the programmer to explicitly keep track of how many copies of a given piece of information are needed in order to perform a computation.

How can general, non-linear computation be performed in an affine setting? In other words, how can a persistent memory be simulated by a volatile memory? The intuitive answer is clear: one persistent memory cell, accessible arbitrarily many times, may be perfectly simulated by infinitely many volatile memory cells, each accessible only once. Of course, if only a finite memory is available, then only an imperfect simulation will be possible in general. However, the important point is that affine computation may approximate non-linear computation to an arbitrary degree of precision.

2 A polyadic affine lambda-calculus

Let us see how the above intuition may be formalized. Consider the fragment

$$A, B ::= X \mid (A_1 \,\&\, 1) \otimes \cdots \otimes (A_n \,\&\, 1) \multimap B$$

of multiplicative additive linear logic (if $n = 0$, then the premise of the implication is the logical constant 1). The proofs of this simple logical system correspond to (simply-typed) terms of the following language:

$$t, u ::= x \mid \lambda x_1 \ldots x_n.t \mid t\langle u_1, \ldots, u_n\rangle,$$

with the requirement that variables appear at most once in terms. In other words, we have a "multilinear", or polyadic affine λ-calculus.

The reduction of a simply-typed non-linear λ-term such as $M = (\lambda x.Nxx)I$ may be "linearized" as

$$[\![M]\!] = (\lambda x_0 x_1.[\![N]\!]\langle x_0\rangle\langle x_1\rangle)\langle [\![I]\!], [\![I]\!]\rangle,$$

in which we see how the duplication of the subterm I by the head redex of M forces us to explicitly introduce two copies of $[\![I]\!]$ (the linearization of I). This is of course very naive: if M duplicates I again (for instance, if $N = \lambda y.zyy$), we will be forced to include additional copies of $[\![I]\!]$ in $[\![M]\!]$ and it would be hard in general to statically determine exactly how many are necessary (we would essentially need to normalize M).

We are thus naturally led to consider an *infinitary* calculus. The rigorous manipulation of infinity requires some form of topology, which will actually be the key to a satisfactory formalization of the above intuition: we will be able to say that affine terms approximate non-linear terms to an arbitrary degree of precision in a clear technical sense, that of metric spaces.

Our first step is to switch to an untyped framework, so that our analysis will be valid in the most general terms. To this extent, we introduce a term $\bot$ in the language, which is used to solve possible mismatches between the arity of abstractions and applications: when reducing $(\lambda x_0 x_1.t)\langle u\rangle$, the sequence in the outer application in not "long enough", so the term $\bot$ will be substituted to x_1.

We also switch from variables to explicit *occurrences*, which is to realize that the affine (or linear) λ-calculus is, in a way, a calculus of occurrences. This, although not technically necessary (and not done in [20]) will simplify the exposition.

So, our definition of (untyped) polyadic affine λ-calculus is the following:

$$t, u ::= \bot \mid x_i \mid \lambda x.t \mid t\mathbf{u},$$

where:

– in x_i, $i \in \mathbb{N}$ is a unique identifier of the occurrence of x, *i.e.*, we require that if x_i, x_j appear in the same term, then $i \neq j$;
– abstractions bind *variables*, *i.e.*, $\lambda x.t$ binds every free occurrence of the form x_i in t (free and bound occurrences are defined as usual);
– $\mathbf{u}$ is a finite sequence of terms. Actually, since we have $\bot$, it is technically simpler to say that $\mathbf{u}$ is a function from $\mathbb{N}$ to terms which is almost everywhere equal to $\bot$.

As usual, terms are always considered up to α-equivalence.

The most important point is how we define reduction:

$$(\lambda x.t)\mathbf{u} \quad \to \quad t[\mathbf{u}/x],$$

where the notation $t[\mathbf{u}/x]$ means that we substitute $\mathbf{u}(i)$ to the at most unique free occurrence x_i in t. We call the set of terms defined above $\Lambda_{\mathrm{p}}^{\mathrm{aff}}$. The superscript reminds us that the calculus is affine, whereas the subscript stands for "polyadic".

The calculus $\Lambda_{\mathrm{p}}^{\mathrm{aff}}$ is strongly confluent (*i.e.*, reduction in at most one step, denoted by $\to^{=}$, enjoys the diamond property) and strongly normalizing. Both properties are immediate consequences of affinity: redexes cannot be duplicated, the theory of residues is trivial and local confluence is achieved in at most one step; moreover, the size of terms strictly decreases during reduction.

3 A metric space of terms and its infinitary completion

Let us now define a function $\Lambda_{\mathrm{p}}^{\mathrm{aff}} \times \Lambda_{\mathrm{p}}^{\mathrm{aff}} \to [0,1]$, by induction on the first argument:

$$d(\bot, t') = \begin{cases} 0 & \text{if } t' = \bot \\ 1 & \text{otherwise} \end{cases}$$

$$d(x_i, t') = \begin{cases} 0 & \text{if } t' = x_i \\ 1 & \text{otherwise} \end{cases}$$

$$d(\lambda x.t_1, t') = \begin{cases} d(t_1, t'_1) & \text{if } t' = \lambda x.t'_1 \\ 1 & \text{otherwise} \end{cases}$$

$$d(t_1\mathbf{u}, t') = \begin{cases} \max\left(d(t_1, t'_1), \sup_{i\in\mathbb{N}} 2^{-i-1} d(\mathbf{u}(i), \mathbf{u}'(i))\right) & \text{if } t' = t'_1\mathbf{u}' \\ 1 & \text{otherwise} \end{cases}$$

Note that, in the abstraction case, we implicitly used α-equivalence to force the variables abstracted in t_1 and t'_1 to coincide. This small nuisance could be avoided by resorting to de Bruijn's notation [5] but, except for the following two paragraphs, we prefer to stick to the usual notation, for better readability.

One may check that d is a bounded ultrametric on $\Lambda_{\mathrm{p}}^{\mathrm{aff}}$, *i.e.*, it is a bounded (by 1) metric which further satisfies $d(t, t'') \leq \max(d(t, t'), d(t', t''))$ for all $t, t', t'' \in \Lambda_{\mathrm{p}}^{\mathrm{aff}}$ (a stronger version of the triangle inequality). A more in-depth analysis of d reveals the following. Consider the poset $\mathbb{N}^*$ of finite sequences of integers, ordered by the prefix relation. A *tree* is, as usual, a downward-closed subset of $\mathbb{N}^*$ (note that non-well-founded and infinitely branching trees are both allowed). Let $\Sigma = \{\bot, \lambda, @\} \cup \mathbb{N}^2$, and let $f : \mathbb{N}^* \to \Sigma$. We define $\operatorname{supp} f = \{a \in \mathbb{N}^* \mid f(a) \neq \bot\}$. We may see the of terms of $\Lambda^{\mathrm{aff}}_{\mathrm{p}}$ (in de Bruijn notation) as finite labeled trees, *i.e.*, as functions t from $\mathbb{N}^*$ (arbitrary integers are needed because applications have arbitrarily large width) to Σ (de Bruijn indices must be *pairs* of integers: one for identifying the abstraction, one for identifying the occurrence), such that $\operatorname{supp} t$ is a finite tree.

Now, if we endow Σ with the discrete uniformity, the ultrametric d may be seen to yield the uniformity of uniform convergence on finitely branching (but possibly infinite) trees. In this uniformity, a sequence of terms $(t_n)_{n\in\mathbb{N}}$ (which are particular functions) converges to t if, for every finitely branching tree $\tau \subseteq \mathbb{N}^*$, there exists $k \in \mathbb{N}$ such that, whenever $n \geq k$, we have $t_n(a) = t(a)$ for all $a \in \tau$. In other words, t_n eventually coincides with t on every finitely branching tree.

Let us look at an example, using the metric d. Let

$$\Delta_n = \lambda x.x_0\langle x_1, \dots, x_n\rangle$$

(when we write a sequence $\mathbf{u}$ as $\langle u_0, \dots, u_{n-1}\rangle$ we mean that $\mathbf{u}(i) = u_i$ for $0 \leq i < n$ and $\mathbf{u}(i) = \bot$ for $i \geq n$). We invite the reader to check that, for all $n \in \mathbb{N}$ and $p > 0$, $d(\Delta_n, \Delta_{n+p}) = 2^{-n-1}$, so the sequence is Cauchy.[1] And yet, no term of $\Lambda^{\mathrm{aff}}_{\mathrm{p}}$ may be the limit of $(\Delta_n)_{n\in\mathbb{N}}$, because the sequence is obviously tending to the infinitary term

$$\Delta = \lambda x.x_0\langle x_1, x_2, \dots\rangle$$

(eventually, Δ_n coincides with Δ on every finitely branching tree).

The above example proves that the metric space $(\Lambda^{\mathrm{aff}}_{\mathrm{p}}, d)$ is not complete. We denote its completion by $\Lambda^{\mathrm{aff}}_{\infty}$. Its terms may no longer be defined inductively, because they may have infinite height. However, they are well-founded, *i.e.*, as trees, they contain no infinite branch from their root. In terms of the above description of terms, $t \in \Lambda^{\mathrm{aff}}_{\infty}$ iff, as a function $t : \mathbb{N}^* \to \Sigma$, $\operatorname{supp} t$ is a well-founded tree. This means that the strict subterm relation $t \sqsubset t'$ is well-founded, so we may still reason by induction on $\Lambda^{\mathrm{aff}}_{\infty}$, in stark contrast with usual infinitary λ-calculi [16]. This is a consequence of the notion of (uniform) convergence induced by d: since a sequence $(t_n)_{n\in\mathbb{N}}$ tending to t must eventually coincide with t on every finitely branching tree, it coincides in particular on infinite trees, which, by König's lemma, must be non-well-founded. But if $(t_n)_{n\in\mathbb{N}}$ is a sequence of $\Lambda^{\mathrm{aff}}_{\mathrm{p}}$, every t_n is finite and in particular well-founded, so it cannot coincide with t on a non-well-founded tree unless t is also well-founded.

On the other hand, finitely high but infinitely wide terms such as Δ are the typical inhabitants of $\Lambda^{\mathrm{aff}}_{\infty} \setminus \Lambda^{\mathrm{aff}}_{\mathrm{p}}$. In fact, in [20] we defined the metric so that only terms of finite height are added to the completion (it is enough to consider the ultrametric $\max(d, \rho)$, where ρ is the discrete pseudometric such that $\rho(t, t') = 0$ as soon as t, t' have the same height, and $\rho(t, t') = 1$ otherwise), on the grounds that these are the most interesting ones and are easier to manipulate (we may apply induction on the height even in the infinitary case). However, in this

[1] Since d is an ultrametric, it is actually enough to check this for $p = 1$ only.

exposition we prefer to bring forth the more natural and topologically better behaved metric d.

Reduction in $\Lambda_\infty^{\text{aff}}$ is defined just as in $\Lambda_{\text{p}}^{\text{aff}}$:

$$(\lambda x.t)\mathbf{u} \quad \to \quad t[\mathbf{u}/x],$$

except that now it may be necessary to perform infinitely many (linear) substitutions, because we may have that x_i is free in t for infinitely many $i \in \mathbb{N}$. We would like to observe that, from a topological point of view, this obvious definition is actually the only possible one. Indeed, it is possible to show, in a sense that we do not make precise here, that reduction as defined above is continuous on $\Lambda_\infty^{\text{aff}}$.[2] Since a continuous function is entirely determined by its behavior on a dense subset like $\Lambda_{\text{p}}^{\text{aff}}$, there is really no other topologically sound way of extending reduction to infinitary terms.

In spite of the presence of infinitary terms, reduction is strongly confluent, because the calculus is still affine, *i.e.*, it is a "calculus of occurrences", in which no subterm is duplicated during reduction. In spite of this, infinitary terms may not normalize. This is easily seen by considering the term

$$\Omega = \Delta\langle \Delta, \Delta, \ldots \rangle,$$

which reduces to itself. Indeed, Δ takes a possibly infinite list, extracts the head (which is $\bot$ if the list is empty) and applies it to the rest of the list. If the list we feed to Δ is made up of infinitely many copies of Δ itself, we obviously loop.

This example gives us the opportunity to see concretely, in a simple but already meaningful case, how affine terms approximate non-linear terms. Of course, technically speaking, the term Ω above is still affine. However, it behaves exactly like its namesake term in the usual λ-calculus (indeed, we will see that it corresponds to it in a precise sense), so we may consider it to be an example of non-linear term. Consider now the finite terms

$$\Omega_n = \Delta_n\langle \overbrace{\Delta_n, \ldots, \Delta_n}^{n \text{ times}} \rangle.$$

We invite again the reader to check that $d(\Omega_n, \Omega) = 2^{-n-1}$, so that $\lim \Omega_n = \Omega$. Hence, Ω_n is supposed to approximate Ω better and better, as n grows. In the case of Ω, there is not much to approximate except divergence; and in fact, $\Omega_n \to^* \bot\langle\rangle$ in $n+1$ steps, *i.e.*, the reduction of Ω_n is longer and longer, approximating the diverging behavior of Ω.

---

[2] We are alluding to Proposition 8 of [20]. Unfortunately, we made a mistake in that paper and Proposition 8 is actually false for the metric used therein. The result does hold for the metric d considered here, which is why we said above that it is "topologically better behaved". The mistake luckily does not affect the main results of [20], in which Proposition 8 plays no role.

4 Uniformity and the isomorphism with the usual lambda-calculus

There are far too many terms (a continuum of them) in $\Lambda_\infty^{\mathrm{aff}}$ for it be directly in correspondence with the usual λ-calculus. We might say that $\Lambda_\infty^{\mathrm{aff}}$ is a *non-uniform* λ-calculus, in the same sense as non-uniform families of circuits: if we accept $\Lambda_\infty^{\mathrm{aff}}$ as a computational model, every function on $\mathbb{N}$ becomes computable, with respect to any standard encoding of natural numbers. To retrieve the λ-calculus, we need to introduce some notion of uniformity.

Definition 1 (Uniformity). *We define $\approx$ to be the smallest partial equivalence relation on $\Lambda_\infty^{\mathrm{aff}}$ such that:*

- *$x_i \approx x_j$ for every variable x and $i, j \in \mathbb{N}$;*
- *if $t \approx t'$, then $\lambda x.t \approx \lambda x.t'$ for every variable x;*
- *if $t \approx t'$ and $\mathbf{u}, \mathbf{u}'$ are such that, for all $i, i' \in \mathbb{N}$, $\mathbf{u}(i) \approx \mathbf{u}'(i')$, then $t\mathbf{u} \approx t'\mathbf{u}'$.*

A term t is uniform *if $t \approx t$. We denote by $\Lambda_\infty^{\mathrm{u}}$ the set of uniform terms.*

Intuitively, $\approx$ equates terms that "look alike" under any possible permutation of the terms appearing in its application sequences. In particular, it equates all occurrences of the same variable: while it is important that we distinguish two occurrences of x by naming one of them x_i and the other x_j (with $i \neq j$), it does not matter which is assigned i and which j.

A term u is uniform if $u \neq \bot$ and if u "looks like itself" even if we permute some of its subterms in application sequences. For instance, any term containing a finite application, such as $z_0\langle x_0 \rangle$, cannot be uniform, because $\langle x_0 \rangle = \langle x_0, \bot \rangle$ and $z_0\langle x_0, \bot \rangle$ and $z_0\langle \bot, x_0 \rangle$ do not "look alike" (indeed, $x_0 \not\approx \bot$). On the other hand, terms like Δ and Ω are uniform (but not Δ_n or Ω_n: by the above remark, a finite approximation of a uniform term containing an application can never be uniform). Note that, if $t\mathbf{u}$ is uniform, then every $\mathbf{u}(i)$ has the same height, that of $\mathbf{u}(0)$. Hence, uniform terms all have finite height. This is why we said above that the terms of finite height are "the most interesting ones".

The set $\Lambda_\infty^{\mathrm{u}}$ is not closed under reduction: in $t = x_0\langle u, u, \ldots \rangle$, with u closed, uniform and such that $u \to u'$, the reduct $t \to x_0\langle u', u, \ldots \rangle$ is in general not uniform, because u' has no reason to "look like" u. The solution is obvious: we must reduce *all* of the copies of u at the same time:

Definition 2 (Infinitary reduction). *We define the relations $\Rightarrow_k$ on $\Lambda_\infty^{\mathrm{u}}$, with $k \in \mathbb{N}$, as the smallest relations satisfying:*

- *$(\lambda x.t)\mathbf{u} \Rightarrow_0 t[\mathbf{u}/x]$;*
- *if $t \Rightarrow_k t'$, then $\lambda x.t \Rightarrow_k \lambda x.t'$;*
- *if $t \Rightarrow_k t'$, then $t\mathbf{u} \Rightarrow_k t'\mathbf{u}$;*
- *if $t\mathbf{u} \in \Lambda_\infty^{\mathrm{u}}$ and $\mathbf{u}(0) \Rightarrow_k u_0'$, by uniformity the "same" reduction may be performed in all $\mathbf{u}(i)$, $i \in \mathbb{N}$, obtaining the term u_i'. If we define $\mathbf{u}'(i) = u_i'$ for all $i \in \mathbb{N}$, then $t\mathbf{u} \Rightarrow_{k+1} t\mathbf{u}'$.*

We denote by $\Rightarrow$ the union of all $\Rightarrow_k$, for $k \in \mathbb{N}$.

Note that $\Rightarrow_k$ is infinitary iff $k > 0$. Indeed, $\Rightarrow_0$ is head reduction,[3] which corresponds to a single reduction step (which may of course perform infinitely many substitutions, but this is not what we mean by "infinitary". Rather, we mean that infinitely many reductions steps are performed together).

Proposition 1. *Let $t \in \Lambda^{\mathrm{u}}_{\infty}$. Then:*

- *$t \Rightarrow t'$ implies $t' \in \Lambda^{\mathrm{u}}_{\infty}$;*
- *furthermore, for all $u \approx t$, $u \Rightarrow u' \approx t'$.*

Proposition 1 asserts that uniform terms are stable under $\Rightarrow$ and that such a rewriting relation is compatible with the equivalence classes of $\approx$. Therefore, the set $\Lambda^{\mathrm{aff}}_{\infty}/\approx$ may be endowed with the (one-step) reduction relation $\Rightarrow$. It turns out that this is exactly the usual, non-linear λ-calculus. In the following, we write Λ for the set of usual λ-terms and $\to_\beta$ for usual β-reduction.

Theorem 1 (Isomorphism). *We have*

$$(\Lambda^{\mathrm{aff}}_{\infty}/\approx, \Rightarrow) \quad \cong \quad (\Lambda, \to_\beta),$$

in the Curry-Howard sense, i.e., there exist two maps

$$(\!|\cdot|\!) : \Lambda^{\mathrm{u}}_{\infty} \to \Lambda \qquad\qquad [\![\cdot]\!] : \Lambda \to \Lambda^{\mathrm{u}}_{\infty}$$

such that, for all $M \in \Lambda$ and $t \in \Lambda^{\mathrm{u}}_{\infty}$:

1. *$(\!|[\![M]\!]|\!) = M$;*
2. *$[\![(\!|t|\!)]\!] \approx t$;*
3. *$M \to_\beta M'$ implies $[\![M]\!] \Rightarrow t' \approx [\![M']\!]$;*
4. *$t \Rightarrow t'$ implies $(\!|t|\!) \to_\beta (\!|t'|\!)$.*

The two maps of the isomorphism are both defined by induction. For what concerns $(\!|\cdot|\!)$, we have:

$$\begin{aligned} (\!|x_i|\!) &= x \quad (\text{for all } i \in \mathbb{N}), \\ (\!|\lambda x.t|\!) &= \lambda x.(\!|t|\!), \\ (\!|t\mathbf{u}|\!) &= (\!|t|\!)(\!|\mathbf{u}(0)|\!). \end{aligned}$$

For what concerns the other direction, we first fix a bijective function $\ulcorner\cdot\urcorner : \mathbb{N}^* \to \mathbb{N}$ to encode finite sequences of integers as integers. Then, we define a family of parametric maps $[\![\cdot]\!]_a$, with $a \in \mathbb{N}^*$, as follows:

$$\begin{aligned} [\![x]\!]_a &= x_{\ulcorner a \urcorner} \\ [\![\lambda x.M]\!]_a &= \lambda x.[\![M]\!]_a \\ [\![MN]\!]_a &= [\![M]\!]_{a0}\langle [\![N]\!]_{a1}, [\![N]\!]_{a2}, [\![N]\!]_{a3}, \ldots \rangle \end{aligned}$$

One can prove that, for any $a, a' \in \mathbb{N}^*$ and any $t \in \Lambda^{\mathrm{u}}_{\infty}$, we actually have $[\![t]\!]_a \in \Lambda^{\mathrm{u}}_{\infty}$ and $[\![t]\!]_a \approx [\![t]\!]_{a'}$. Of course, Theorem 1 holds for any choice of $a \in \mathbb{N}^*$, that is why we simply write $[\![\cdot]\!]$. We let the reader check that the uniform infinitary terms Δ and Ω introduced above are (modulo $\approx$) the images through $[\![\cdot]\!]$ of their well-known namesake λ-terms.

[3] It is actually *spinal* reduction, but the distinction is inessential.

5 The proof-theoretic perspective

As already mentioned, our idea of obtaining the λ-calculus through a metric completion process has proof-theoretic roots, in particular in linear logic. In fact, the above constructions may be reformulated using proofs instead of λ-terms. In [21], we show how a fully-complete model of polarized multiplicative exponential linear logic may be built as a metric completion of a model of the sole multiplicative fragment. Roughly speaking, we take objects which are very much related to the *designs* of Girard's ludics [13], introduce a metric completely analogous to the one given here, and construct the model in the completed space. What we obtain closely resembles Abramsky, Jagadeesan and Malacaria's formulation of games semantics [1].

Recently, Melliès and Tabareau [23] used a similar idea to provide an explicit formula for constructing the free commutative comonoid in certain symmetric monoidal categories. This offers a categorical viewpoint on our work, and yields some potentially interesting remarks.

Melliès and Tabareau's construction starts with a symmetric monoidal category $(\mathcal{C}, \otimes, 1)$ with finite products, which we denote by $A \mathbin{\&} B$. We define $\dagger A = A \mathbin{\&} 1$, the free co-pointed object on A, with its canonical projection $\pi^A : \dagger A \longrightarrow 1$. We also inductively define $A^{\otimes 0} = 1$, $A^{\otimes n+1} = A^{\otimes n} \otimes A$.

Using the symmetry of $\mathcal{C}$, for every $n \in \mathbb{N}$ we may build $n!$ parallel isomorphisms $\sigma_i^{A,n} : (\dagger A)^{\otimes n} \longrightarrow (\dagger A)^{\otimes n}$. We define $A^{\leq n}$ to be the equalizer, if it exists, of $\sigma_1^{A,n}, \ldots, \sigma_{n!}^{A,n}$.

Now, by the universal property of equalizers on the morphism π^A, we know that there is a canonical projection $\pi_n^A : A^{\leq n+1} \longrightarrow A^{\leq n}$, for all $n \in \mathbb{N}$. Then, we define $!A$ to be the limit, if it exists, of the diagram

$$A^{\leq 0} \xleftarrow{\pi_0^A} A^{\leq 1} \xleftarrow{\pi_1^A} A^{\leq 2} \xleftarrow{\pi_2^A} \cdots$$

Melliès and Tabareau's result is the following:

Proposition 2 ([23]). *If the equalizers and the projective limit considered above exist in $\mathcal{C}$ and if these limits commute with the tensor product of $\mathcal{C}$, then, for every object A of $\mathcal{C}$, $!A$ is the free commutative comonoid on A.*

It is known that, in a $*$-autonomous category with finite products, the existence of the free commutative comonoid on every object yields a denotational model of full linear logic (a result due to Lafont, see Melliès's survey in [8]). Therefore, Proposition 2 provides a way of building, under certain conditions, models of full linear logic starting from models of its multiplicative additive fragment.

The conditions required by Proposition 2 are however not anodyne. In fact, Tasson showed [23] how the construction fails in a well known model of linear logic, Ehrhard's finiteness spaces [11]. In this model, although all the required limits exist, the projective limit does not commute with the tensor product.

Our approach seems to offer an alternative construction to that of Melliès and Tabareau's, in which the two main steps for building the free comonoid are reversed: first one computes a projective limit, then one equalizes. This follows our procedure for recovering the λ-calculus: we first complete the space $\Lambda_{\mathrm{p}}^{\mathrm{aff}}$ to obtain $\Lambda_{\infty}^{\mathrm{aff}}$, then we introduce uniformity and obtain the λ-calculus as the quotient $\Lambda_{\infty}^{\mathrm{aff}}/\approx$.

More in detail, we start by defining $p_n^A : (\dagger A)^{\otimes n+1} \longrightarrow (\dagger A)^{\otimes n}$ as the morphism obtained by composing $id_{(\dagger A)^{\otimes n}} \otimes \pi^A$ with the iso $(\dagger A)^{\otimes n} \otimes 1 \cong (\dagger A)^{\otimes n}$. Then, we define ∇A as the limit (if it exists) of the diagram

$$(\dagger A)^{\otimes 0} \xleftarrow{p_0^A} (\dagger A)^{\otimes 1} \xleftarrow{p_1^A} (\dagger A)^{\otimes 2} \xleftarrow{p_2^A} \cdots$$

At this point, if we suppose that the above limit commutes with the tensor, *i.e.*, that $\nabla A \otimes \nabla A$ is the limit of the diagram

$$(\dagger A)^{\otimes 0} \otimes (\dagger A)^{\otimes 0} \xleftarrow{p_0^A \otimes p_0^A} (\dagger A)^{\otimes 1} \otimes (\dagger A)^{\otimes 1} \xleftarrow{p_1^A \otimes p_1^A} (\dagger A)^{\otimes 2} \otimes (\dagger A)^{\otimes 2} \xleftarrow{p_2^A \otimes p_2^A} \cdots,$$

then it is not hard to see that ∇A is also a cone for the second diagram, and that $\nabla A \otimes \nabla A$ is a cone for the first. Therefore, we have two canonical morphisms $\varphi : \nabla A \longrightarrow \nabla A \otimes \nabla A$ and $\psi : \nabla A \otimes \nabla A \longrightarrow \nabla A$. Using these and the symmetry of $\mathcal{C}$, we build infinitely many endomorphisms of ∇A, of the form $\nabla A \longrightarrow (\nabla A)^{\otimes n} \longrightarrow (\nabla A)^{\otimes n} \longrightarrow \nabla A$. We define $!A$ to be the equalizer (if it exists) of all these endomorphisms.

If we apply this construction to the category of finiteness spaces, $!A$ actually turns out to be the free commutative comonoid on A. Whether this is this just a coincidence or whether a suitable rephrasing of Proposition 2 holds is currently unknown and is doubtlessly an interesting topic of further research.

6 Complexity-bounded calculi

We add purely linear terms to our syntax, *i.e.*, we consider a denumerably infinite set of linear variables, disjoint from the set of usual variables and ranged over by $a, b, c, \ldots$, and we modify the grammar defining $\Lambda_{\mathrm{p}}^{\mathrm{aff}}$ as follows:

$$t, u ::= \bot \mid x_i \mid \lambda x.t \mid t\mathbf{u} \mid a \mid \ell a.t \mid tu.$$

Furthermore, we require that:

- occurrences of variables (x_i) and linear variables (a) both appear at most once in terms;
- in $\ell a.t$, which is a linear abstraction, the variable a must appear free in t;
- in $t\mathbf{u}$, no $\mathbf{u}(i)$ contains free linear variables, for $i \in \mathbb{N}$.

We denote by $\ell\Lambda_{\mathrm{p}}^{\mathrm{aff}}$ the set of terms thus obtained.

Proof-theoretically, this calculus corresponds to allowing simple linear implication in the fragment of multiplicative additive linear logic we consider:

$$A, B ::= X \mid A \multimap B \mid (A_1 \,\&\, 1) \otimes \cdots \otimes (A_n \,\&\, 1) \multimap B.$$

Reduction in $\ell\Lambda_{\rm p}^{\rm aff}$ is defined by adding a purely linear β-reduction rule besides the one already present in $\Lambda_{\rm p}^{\rm aff}$:

$$(\ell a.t)u \to t[u/a],$$
$$(\lambda x.t)\mathbf{u} \to t[\mathbf{u}/x].$$

Note that the absence of types produces "clashes", *i.e.*, terms of the form $(\ell a.t)\mathbf{u}$ or $(\lambda x.t)u$, which look like redexes (especially the latter...) but are not reduced. This is unproblematic for our purposes.

The ultrametric d on $\ell\Lambda_{\rm p}^{\rm aff}$ is defined just as in Sect. 3 for the inductive cases already present in $\Lambda_{\rm p}^{\rm aff}$, and is trivially extended to the other cases:

$$d(a,t') = \begin{cases} 0 & \text{if } t' = a \\ 1 & \text{otherwise} \end{cases}$$
$$d(\ell a.t_1,t') = \begin{cases} d(t_1,t_1') & \text{if } t' = \ell a.t_1' \\ 1 & \text{otherwise} \end{cases}$$
$$d(t_1u,t') = \begin{cases} \max\left(d(t_1,t_1'),d(u,u')\right) & \text{if } t' = t_1'u' \\ 1 & \text{otherwise} \end{cases}$$

We denote by $\ell\Lambda_\infty^{\rm aff}$ the completion of $\ell\Lambda_{\rm p}^{\rm aff}$ with respect to d.

The partial equivalence relation $\approx$ is extended to $\ell\Lambda_\infty^{\rm aff}$ in the obvious way: $a \approx a$ for every linear variable a; if $t \approx t'$, then $\ell a.t \approx \ell a.t'$; if $t \approx t'$ and $u \approx u'$, then $tu \approx t'u'$. Hence, a term $t \in \ell\Lambda_\infty^{\rm aff}$ is uniform if $t \approx t$. Infinitary reduction is also extended to the uniform terms of $\ell\Lambda_\infty^{\rm aff}$ in the obvious way (the index of $\Rightarrow_k$ does not increase when reducing inside the argument of a linear application).

Of course, $\ell\Lambda_\infty^{\rm aff}$ brings nothing really new with respect to $\Lambda_\infty^{\rm aff}$. In particular, if we are only interested in the λ-calculus, purely linear terms are useless. They become interesting when we restrict the space of finite terms, *i.e.*, the approximations we are allowed to use.

Definition 3 (Depth, stratified term). *The depth of a free occurrence of variable x_i in a term $t \in \ell\Lambda_{\rm p}^{\rm aff}$, denoted by $\delta_{x_i}(t)$, is defined by induction on t:*
- $\delta_{x_i}(x_i) = 0$*;*
- $\delta_{x_i}(\lambda y.t_1) = \delta_{x_i}(\ell a.t_1) = \delta_{x_i}(t_1)$*;*
- *if $t = t_1\mathbf{u}$, then x_i is free in $\mathbf{u}(p)$ for some $p \in \mathbb{N}$, and we set $\delta_{x_i}(t) = \delta_{x_i}(\mathbf{u}(p)) + 1$;*
- *similarly, if $t = t_1t_2$, then x_i must be free in t_p for $p \in \{1,2\}$, and we set $\delta_{x_i}(t) = \delta_{x_i}(t_p)$.*

A term $t \in \ell\Lambda_{\rm p}^{\rm aff}$ is stratified *if:*
- *whenever x_i is free in t, $\delta_{x_i}(t) = 1$;*
- *for every subterm of t of the form $\lambda x.u$ and for every $i \in \mathbb{N}$ such that x_i is free in u, $\delta_{x_i}(u) = 1$.*

We denote by $\ell\Lambda_{\rm p}^{\rm s}$ the set of all stratified terms.

The definition of stratified term clarifies why we need to consider purely linear terms: in their absence, the only stratified applications would be of the form $\bot\mathbf{u}$, *i.e.*, head variables are excluded, because their depth is always 0.

As a subset of $\ell\Lambda_{\mathrm{p}}^{\mathrm{aff}}$, $\ell\Lambda_{\mathrm{p}}^{\mathrm{s}}$ is also a metric space, with the same ultrametric d. However, $\ell\Lambda_{\mathrm{p}}^{\mathrm{s}}$ is not dense in $\ell\Lambda_{\infty}^{\mathrm{aff}}$. In fact, its completion, which is equal to its topological closure as a subset of $\ell\Lambda_{\infty}^{\mathrm{aff}}$ and which we denote by $\ell\Lambda_{\infty}^{\mathrm{s}}$, is strictly smaller. We may see this by considering the term Δ introduced in Sect. 3. In order for any $t \in \ell\Lambda_{\mathrm{p}}^{\mathrm{aff}}$ to be such that $d(t, \Delta) < 1$, we must have $t = \lambda x.x_0\mathbf{u}$, which is not stratified. Hence, no sequence in $\ell\Lambda_{\mathrm{p}}^{\mathrm{s}}$ ever tends to Δ, and this term is not present in $\ell\Lambda_{\infty}^{\mathrm{s}}$. Similarly, $\Omega \notin \ell\Lambda_{\infty}^{\mathrm{s}}$.

The above example is interesting because it excludes the most obvious source of divergence in $\ell\Lambda_{\infty}^{\mathrm{aff}}$. In fact, $\ell\Lambda_{\infty}^{\mathrm{s}}/\approx$ is actually an elementary λ-calculus, in the same sense as that of [7]. When suitably typed in a system/logic containing a type $\mathbf{N}$ corresponding to natural numbers, the terms of type $\mathbf{N} \to \mathbf{N}$ represent exactly the elementary functions, which are those computable by a Turing machine in time bounded by a tower of exponentials of fixed height.

We believe that a polytime λ-calculus may be obtained by *considering another metric* on $\ell\Lambda_{\mathrm{p}}^{\mathrm{s}}$. That is, the approximations are the same, but they do not have the same meaning. To give an analogy (which is purely suggestive, not technical), we may consider the standard sequence spaces used in analysis. The set c_{00} of infinite sequences of real numbers which are almost everywhere null (hence virtually finite) may be endowed with many different metrics, according to which the completion only contains sequences which tend to 0. However, the rate at which they are allowed to vanish is different: any rate (c_0), strictly more than the linear inverse (ℓ^1), strictly more than the inverse square ($\ell^{\frac{1}{2}}$)...

At the moment, we have a metric such that, when we complete $\ell\Lambda_{\mathrm{p}}^{\mathrm{s}}$ with respect to it and consider uniform terms, we seem to obtain a space of terms roughly corresponding to a poly-time λ-calculus such as the one of [25]. Although we have no precise results yet, this research direction looks promising and is definitely worth further investigation. In particular, thanks to non-uniform terms, this might lead to a λ-calculus characterization of the class $\mathbf{P}/_{\mathrm{poly}}$.

Acknowledgments. This summary is mostly based on [20] and the journal version [21] (under review), which benefited from the partial support of ANR projects Complice (08-BLAN-0211-01), Panda (09-BLAN-0169-02) and Logoi (10-BLAN-0213-02).

References

1. Abramsky, S., Jagadeesan, R., Malacaria, P.: Full abstraction for PCF. Inform. Comput 163(2), 409–470 (2000)
2. Accattoli, B., Kesner, D.: Preservation of strong normalisation modulo permutations for the structural lambda-calculus. Logical Methods in Computer Science 8(1) (2012)
3. Asperti, A., Guerrini, S.: The Optimal Implementation of Functional Programming Languages. Cambridge University Press (1998)
4. Baillot, P., Terui, K.: Light types for polynomial time computation in lambda calculus. Inf. Comput. 207(1), 41–62 (2009)
5. de Bruijn, N.G.: Lambda calculus notation with nameless dummies. Indagat. Math. 34, 381–392 (1972)

6. Caires, L., Pfenning, F.: Session types as intuitionistic linear propositions. In: Gastin, P., Laroussinie, F. (eds.) Proceedings of CONCUR. Lecture Notes in Computer Science, vol. 6269, pp. 222–236. Springer (2010)
7. Coppola, P., Martini, S.: Typing lambda terms in elementary logic with linear constraints. In: Proceedings of TLCA. pp. 76–90 (2001)
8. Curien, P.L., Herbelin, H., Krivine, J.L., Melliès, P.A.: Interactive Models of Computation and Program Behavior. AMS (2010)
9. Curien, P.L., Munch-Maccagnoni, G.: The duality of computation under focus. In: Calude, C.S., Sassone, V. (eds.) Proceedings of IFIP TCS. IFIP AICT, vol. 323, pp. 165–181. Springer (2010)
10. Danos, V., Ehrhard, T.: Probabilistic coherence spaces as a model of higher-order probabilistic computation. Inf. Comput. 209(6), 966–991 (2011)
11. Ehrhard, T.: Finiteness spaces. Mathematical Structures in Computer Science 15(4), 615–646 (2005)
12. Girard, J.Y.: Linear logic. Theor. Comput. Sci. 50(1), 1–102 (1987)
13. Girard, J.Y.: Locus solum. Math. Struct. Comput. Sci. 11(3), 301–506 (2001)
14. Hofmann, M.: Linear Types and Non-Size-Increasing Polynomial Time Computation. Inform. Comput. 183(1), 57–85 (2003)
15. Honda, K., Laurent, O.: An exact correspondence between a typed pi-calculus and polarised proof-nets. Theor. Comput. Sci. 411(22-24), 2223–2238 (2010)
16. Kennaway, R., Klop, J.W., Sleep, R., de Vries, F.J.: Infinitary lambda calculus. Theor. Comput. Sci. 175(1), 93–125 (1997)
17. Lago, U.D., Gaboardi, M.: Linear dependent types and relative completeness. Logical Methods in Computer Science 8(4) (2011)
18. Laurent, O., Regnier, L.: About translations of classical logic into polarized linear logic. In: Proceedings of LICS. pp. 11–20. IEEE Computer Society (2003)
19. Mackie, I.: Efficient lambda-evaluation with interaction nets. In: van Oostrom, V. (ed.) Proceedings of RTA. LNCS, vol. 3091, pp. 155–169. Springer (2004)
20. Mazza, D.: An infinitary affine lambda-calculus isomorphic to the full lambda-calculus. In: Dershowitz, N. (ed.) Proceedings of LICS. pp. 471–480. IEEE Computer Society (2012)
21. Mazza, D.: Non-linearity as the metric completion of linearity. Submitted (2013), available online at `http://lipn.univ-paris13.fr/~mazza/?page=pub`
22. Melliès, P.A., Tabareau, N.: Resource modalities in tensor logic. Ann. Pure Appl. Logic 161(5), 632–653 (2010)
23. Melliès, P.A., Tabareau, N., Tasson, C.: An explicit formula for the free exponential modality of linear logic. In: Proc. ICALP 2009. pp. 247–260 (2009)
24. Selinger, P., Valiron, B.: A lambda calculus for quantum computation with classical control. Mathematical Structures in Computer Science 16(3), 527–552 (2006)
25. Terui, K.: Light affine calculus and polytime strong normalization. In: Proceedings of LICS. pp. 209–220. IEEE Computer Society (2001)