A Non-Uniform Finitary Relational Semantics of System T

Lionel Vaux*
Laboratoire de Mathématiques de l'Université de Savoie
UFR SFA, Campus Scientifique, 73376 Le Bourget-du-Lac Cedex, FRANCE
lionel.vaux@univ-savoie.fr
August 11, 2009

Abstract

We study iteration and recursion operators in the denotational semantics of typed λ-calculi derived from the multiset relational model of linear logic. Although these operators are defined as fixpoints of typed functionals, we prove them finitary in the sense of Ehrhard's finiteness spaces.

1 Introduction

Finiteness spaces were introduced by Ehrhard [1], refining the purely relational model of linear logic. A finiteness space is a set equipped with a finiteness structure, i.e. a particular set of subsets which are said to be finitary; and the model is such that the relational denotation of a proof in linear logic is always a finitary subset of its conclusion. By the usual co-Kleisli construction, this also provides a model of the simply typed λ-calculus: the cartesian closed category $\underline{\mathrm{Fin}}$. The main property of finiteness spaces is that the intersection of two finitary subsets of dual types is always finite. This feature allows to reformulate Girard's quantitative semantics [2] in a standard algebraic setting, where morphisms interpreting typed λ-terms are analytic functions between the topological vector spaces generated by vectors with finitary supports. This provided the semantical foundations of Ehrhard-Regnier's differential λ-calculus [3] and motivated the general study of a differential extension of linear logic (e.g., [4, 5, 6, 7, 8, 9, 10]).

It is worth noticing that finiteness spaces can accomodate typed λ-calculi only. In particular, the relational semantics of fixpoint combinators is never finitary. The whole point of the finiteness construction is actually to reject infinite computations, ensuring the intermediate sets involved in the relational interpretation of a cut are all finite. Despite this restrictive design, Ehrhard proved that a limited form of recursion was available, by defining a finitary tail-recursive iteration operator.

The main result of the present paper is that finiteness spaces can actually accomodate the standard notion of primitive recursion in λ-calculus, Gödel's system T: we prove $\underline{\mathrm{Fin}}$ admits a weak natural number object in the sense of [11, 12], and we more generally exhibit a finitary recursion operator for this interpretation of the type of natural numbers. This achievement is twofold:

- Before considering finiteness, we must define a recursion operator in the cartesian closed category deduced from the relational model of linear logic. For that purpose, we cannot follow Ehrhard and use the flat interpretation of the type Nat of natural numbers. Indeed, if t, u and v are terms of types respectively Nat, $\mathsf{Nat} \Rightarrow X \Rightarrow X$ and X, the recursion step $\mathsf{R}(\mathsf{S}t)uv \rightsquigarrow ut(\mathsf{R}tuv)$ puts t in argument position. In case u is a constant function, t is not used in the reduced form. The recursor R must however discriminate between $\mathsf{S}t$ and O, hence the successor S cannot be linear: it must produce information independently from its input. Though it might be obscure for the reader not familiar with the relational or coherence semantics, this argument will be made formal in the paper. This was already noted by Girard in coherence spaces [13]: we adopt the solution he proposed, and interpret terms of type Nat by so-called *lazy* natural numbers. An notable outcome is that our

*This work has been partially funded by the French ANR projet blanc "Curry Howard pour la Concurrence" CHOCO ANR-07-BLAN-0324.

interpretation provides a semantic evidence of the well-known gap in expressive power between the iterator and recursor variants of system T.

- The second aspect of our work is to establish that this relational semantics is finitary. This is far from immediate because the recursion operator is defined as the fixpoint of finitary approximants: since fixpoints themselves are not finitary relations, it is necessary to obtain stronger properties of these approximants to conclude.

Structure of the paper. In section 2, we briefly describe two cartesian closed categories: the category Rel of sets and relations from multisets to points, and the category Fin of finiteness spaces and finitary relations from multisets to points. In section 3, we give an explicit presentation of the relational semantics of typed λ-calculi in Rel and Fin, which we extend to system T in section 4. In section 5, we establish a uniformity property of iteration-definable morphisms, which does not hold for recursion in general.

2 Sets, Relations and Finiteness Spaces

If A is a set, denote by $\mathfrak{P}(A)$ the powerset of A, by $\mathfrak{P}_{\mathrm{f}}(A)$ the set of all finite subsets of A and by $A^!$ the set of all finite multisets of A. If $(\alpha_1,\dots,\alpha_n)\in A^n$, we write $\overline{\alpha}=[\alpha_1,\dots,\alpha_n]$ for the corresponding multiset, and denote multiset union additively. Let $f\subseteq A\times B$ be a relation from A to B, we write $f^\perp=\{(\beta,\alpha);\ (\alpha,\beta)\in f\}$. For all $a\subseteq A$, we set $f\cdot a=\{\beta\in B;\ \exists\alpha\in a,\ (\alpha,\beta)\in f\}$. We write Rel for the coKleisli category of the comonad $(-)^!$ in the relational model of linear logic (see e.g. [14]): objects are sets and $\underline{\mathrm{Rel}}(A,B)=\mathfrak{P}\left(A^!\times B\right)$; the identity on A is $id_A=\{([\alpha],\alpha);\ \alpha\in A\}$; if $f\in\underline{\mathrm{Rel}}(A,B)$ and $g\in\underline{\mathrm{Rel}}(B,C)$ then $g\circ f=\left\{(\sum_{i=1}^n\overline{\alpha}_i,\gamma);\ \exists\overline{\beta}=[\beta_1,\dots,\beta_n]\in B^!,\ (\overline{\beta},\gamma)\in g\wedge\forall i\ (\overline{\alpha}_i,\beta_i)\in f\right\}$.

The category Rel is cartesian closed. The cartesian product is given by the disjoint union of sets $A\uplus B=(\{1\}\times A)\cup(\{2\}\times B)$, with terminal object the empty set $\emptyset$. Projections are $\{([(1,\alpha)],\alpha);\ \alpha\in A\}\in\underline{\mathrm{Rel}}(A\uplus B,A)$ and $\{([(2,\beta)],\beta);\ \beta\in B\}\in\underline{\mathrm{Rel}}(A\uplus B,B)$. If $f\in\underline{\mathrm{Rel}}(C,A)$ and $g\in\underline{\mathrm{Rel}}(C,B)$, pairing is given by: $\langle f,g\rangle=\{(\overline{\gamma},(1,\alpha));\ (\overline{\gamma},\alpha)\in f\}\cup\{(\overline{\gamma},(2,\beta));\ (\overline{\gamma},\beta)\in g\}\in\underline{\mathrm{Rel}}(C,A\uplus B)$. The unique morphism from A to $\emptyset$ is $\emptyset$. The adjunction for closedness is $\underline{\mathrm{Rel}}(A\uplus B,C)\cong\underline{\mathrm{Rel}}(A,B^!\times C)$ which boils down to the bijection $(A\uplus B)^!\cong A^!\times B^!$.

We recall the few notions we shall use on finiteness spaces. For a detailled presentation, the obvious reference is [1]. Let $\mathfrak{F}\subseteq\mathfrak{P}(A)$ be any set of subsets of A. We define the pre-dual of $\mathfrak{F}$ in A as $\mathfrak{F}^\perp=\{a'\subseteq A;\ \forall a\in\mathfrak{F},\ a\cap a'\in\mathfrak{P}_{\mathrm{f}}(A)\}$. By a standard argument, we have the following immediate properties: $\mathfrak{P}_{\mathrm{f}}(A)\subseteq\mathfrak{F}^\perp$; $\mathfrak{F}\subseteq\mathfrak{F}^{\perp\perp}$; if $\mathfrak{G}\subseteq\mathfrak{F}$, $\mathfrak{F}^\perp\subseteq\mathfrak{G}^\perp$. By the last two, we get $\mathfrak{F}^\perp=\mathfrak{F}^{\perp\perp\perp}$. A *finiteness structure* on A is a set $\mathfrak{F}$ of subsets of A such that $\mathfrak{F}^{\perp\perp}=\mathfrak{F}$. Then a *finiteness space* is a dependant pair $\mathscr{A}=(|\mathscr{A}|,\mathfrak{F}(\mathscr{A}))$ where $|\mathscr{A}|$ is the underlying set, called the *web* of $\mathscr{A}$, and $\mathfrak{F}(\mathscr{A})$ is a finiteness structure on $|\mathscr{A}|$. We write $\mathscr{A}^\perp$ for the dual finiteness space: $\left|\mathscr{A}^\perp\right|=|\mathscr{A}|$ and $\mathfrak{F}\left(\mathscr{A}^\perp\right)=\mathfrak{F}(\mathscr{A})^\perp$. The elements of $\mathfrak{F}(\mathscr{A})$ are called the *finitary subsets* of $\mathscr{A}$.

For all set A, $(A,\mathfrak{P}_{\mathrm{f}}(A))$ is a finiteness space and $(A,\mathfrak{P}_{\mathrm{f}}(A))^\perp=(A,\mathfrak{P}(A))$. In particular, each finite set A is the web of exactly one finiteness space: $(A,\mathfrak{P}_{\mathrm{f}}(A))=(A,\mathfrak{P}(A))$. We introduce the empty finiteness space $\mathscr{T}=(\emptyset,\{\emptyset\})$ and the finiteness space of *flat natural numbers* $\mathscr{N}=(\mathbf{N},\mathfrak{P}_{\mathrm{f}}(\mathbf{N}))$. If $\mathscr{A}$ and $\mathscr{B}$ are finiteness spaces, we define $\mathscr{A}\,\&\,\mathscr{B}$ and $\mathscr{A}\Rightarrow\mathscr{B}$ as follows. Let $|\mathscr{A}\,\&\,\mathscr{B}|=|\mathscr{A}|\uplus|\mathscr{B}|$ and $\mathfrak{F}(\mathscr{A}\,\&\,\mathscr{B})=\{a\uplus b;\ a\in\mathfrak{F}(\mathscr{A})\wedge b\in\mathfrak{F}(\mathscr{B})\}$. Let $|\mathscr{A}\Rightarrow\mathscr{B}|=|\mathscr{A}|^!\times|\mathscr{B}|$ and set $f\in\mathfrak{F}(\mathscr{A}\Rightarrow\mathscr{B})$ iff: $\forall a\in\mathfrak{F}(\mathscr{A})$, $f\cdot a^!\in\mathfrak{F}(\mathscr{B})$, and $\forall\beta\in|\mathscr{B}|$, $(f^\perp\cdot\{\beta\})\cap a^!$ is finite. It is easily seen that $\mathscr{A}\,\&\,\mathscr{B}$ is a finiteness space, but the same result for $\mathscr{A}\Rightarrow\mathscr{B}$ is quite technical and the only known proof uses the axiom of choice [1]. We call *finitary relations* the elements of $\mathfrak{F}(\mathscr{A}\Rightarrow\mathscr{B})$.

Notice that $\mathfrak{F}(\mathscr{A}\Rightarrow\mathscr{B})\subseteq\underline{\mathrm{Rel}}(|\mathscr{A}|,|\mathscr{B}|)$. We write Fin for the category of finiteness spaces with $\underline{\mathrm{Fin}}(\mathscr{A},\mathscr{B})=\mathfrak{F}(\mathscr{A}\Rightarrow\mathscr{B})$ and composition defined as in Rel. It is cartesian closed with terminal object

$$\frac{}{\Gamma,x:A,\Delta\vdash x:A}\,(\mathrm{Var})\qquad \frac{}{\Gamma\vdash\langle\rangle:\top}\,(\mathrm{Unit})\qquad \frac{a\in\mathfrak{C}_A}{\Gamma\vdash a:A}\,(\mathrm{Const})$$

$$\frac{\Gamma,x:A\vdash s:B}{\Gamma\vdash\lambda x\,s:A\to B}\,(\mathrm{Abs})\qquad \frac{\Gamma\vdash s:A\to B\qquad \Gamma\vdash t:A}{\Gamma\vdash s\,t:B}\,(\mathrm{App})$$

$$\frac{\Gamma\vdash s:A\qquad \Gamma\vdash t:B}{\Gamma\vdash\langle s,t\rangle:A\times B}\,(\mathrm{Pair})\qquad \frac{\Gamma\vdash s:A\times B}{\Gamma\vdash\pi_l\,s:A}\,(\mathrm{Left})\qquad \frac{\Gamma\vdash s:A\times B}{\Gamma\vdash\pi_r\,s:B}\,(\mathrm{Right})$$

Figure 1: Rules of typed λ-calculi with products

$$\frac{}{\Gamma^{\square},x^{[\alpha]}:A,\Delta^{\square}\vdash x^{\alpha}:A}\,[\![\mathrm{Var}]\!]\qquad \frac{a\in\mathfrak{C}_A\quad \alpha\in[\![a]\!]}{\Gamma^{\square}\vdash a^{\alpha}:A}\,[\![\mathrm{Const}]\!]$$

$$\frac{\Gamma,x^{\overline{\alpha}}:A\vdash s^{\beta}:B}{\Gamma\vdash\lambda x\,s^{(\overline{\alpha},\beta)}:A\to B}\,[\![\mathrm{Abs}]\!]\qquad \frac{\Gamma_0\vdash s^{([\alpha_1,\dots,\alpha_k],\beta)}:A\to B\qquad \Gamma_1\vdash t^{\alpha_1}:A\quad\cdots\quad \Gamma_k\vdash t^{\alpha_k}:A}{\sum_{j=0}^{k}\Gamma_j\vdash s\,t^{\beta}:B}\,[\![\mathrm{App}]\!]$$

$$\frac{\Gamma\vdash s_i^{\alpha}:A_i}{\Gamma\vdash\langle s_1,s_2\rangle^{(i,\alpha)}:A_1\times A_2}\,[\![\mathrm{Pair}_i]\!]\qquad \frac{\Gamma\vdash s^{(1,\alpha)}:A\times B}{\Gamma\vdash\pi_l\,s^{\alpha}:A}\,[\![\mathrm{Left}]\!]\qquad \frac{\Gamma\vdash s^{(2,\beta)}:A\times B}{\Gamma\vdash\pi_r\,s^{\beta}:B}\,[\![\mathrm{Right}]\!]$$

Figure 2: Computing points in the relational semantics

$\mathscr{T}$, product $-\,\&\,-$ and exponential $-\Rightarrow-$: the definitions of those functors on morphisms, the natural transformations, and the adjunction required for cartesian closedness are exactly the same as for Rel.

3 The Multiset Relational Semantics of Typed λ-Calculi

Typed λ-calculi. In this section, we give an explicit description of the interpretation in Rel and Fin of the basic constructions of typed λ-calculi with products. Type and term expressions are given by:

$$A,B ::= X \mid A\to B \mid A\times B \mid \top \qquad\text{and}\qquad s,t ::= x \mid a \mid \lambda x\,s \mid s\,t \mid \langle s,t\rangle \mid \pi_l\,s \mid \pi_r\,s \mid \langle\rangle$$

where X ranges over a fixed set $\mathfrak{A}$ of atomic types, x ranges over term variables and a ranges over term constants. To each variable or constant, we associate a type, and we write $\mathfrak{C}_A$ for the collection of constants of type A. A typing judgement is an expression $\Gamma\vdash s:A$ derived from the rules in Figure 1 where contexts Γ and Δ range over lists $(x_1:A_1,\dots,x_n:A_n)$ of typed variables. The operational semantics of a typed λ-calculus is given by a contextual equivalence relation $\simeq$ on typed terms: if $s\simeq t$, then s and t have the same type, say A; we then write $\Gamma\vdash s\simeq t:A$ for any suitable Γ. In general, we will give $\simeq$ as the reflexive, symmetric and transitive closure of a contextual relation $>$ on typed terms. We define $>_0$ as the least one such that: $\pi_l\,\langle s,t\rangle >_0 s$, $\pi_r\,\langle s,t\rangle >_0 t$ and $(\lambda x\,s)\,t >_0 s\,[x:=t]$ (with the obvious assumptions ensuring typability), and we write $\simeq_0$ for the corresponding equivalence.

Relational interpretation and finiteness property. Assume a set $[\![X]\!]$ is given for each base type X; then we interpret type constructions by $[\![A\to B]\!]=[\![A]\!]^{!}\times[\![B]\!]$, $[\![A\times B]\!]=[\![A]\!]\uplus[\![B]\!]$ and $[\![\top]\!]=\emptyset$. Further assume that with every constant $a\in\mathfrak{C}_A$ is associated a subset $[\![a]\!]\subseteq[\![A]\!]$. The relational semantics of a derivable typing judgement $x_1:A_1,\dots,x_n:A_n\vdash s:A$ will be a relation $[\![s]\!]_{x_1:A_1,\dots,x_n:A_n}\subseteq[\![A_1]\!]^{!}\times\cdots\times[\![A_n]\!]^{!}\times[\![A]\!]$. We first introduce the deductive system of Figure 2. In this system, derivable judgements are semantic annotations of typing judgements: $x_1^{\overline{\alpha}_1}:A_1,\dots,x_n^{\overline{\alpha}_n}:A_n\vdash s^{\alpha}:A$ stands for $(\overline{\alpha}_1,\dots,\overline{\alpha}_n,\alpha)\in[\![s]\!]_{x_1:A_1,\dots,x_n:A_n}$ where each $\overline{\alpha}_i\in[\![A_i]\!]^{!}$ and $\alpha\in[\![A]\!]$. In rules $[\![\mathrm{Var}]\!]$ and $[\![\mathrm{Const}]\!]$, $\Gamma^{\square}$ denotes an annotated context of the form $x_1^{[]}:A_1,\dots,x_n^{[]}:A_n$. In rule $[\![\mathrm{App}]\!]$, the sum of annotated contexts is defined pointwise: $\left(x_1^{\overline{\alpha}_1}:A_1,\dots,x_n^{\overline{\alpha}_n}:A_n\right)+\left(x_1^{\overline{\alpha}'_1}:A_1,\dots,x_n^{\overline{\alpha}'_n}:A_n\right)=\left(x_1^{\overline{\alpha}_1+\overline{\alpha}'_1}:A_1,\dots,x_n^{\overline{\alpha}_n+\overline{\alpha}'_n}:A_n\right)$. The semantics of a term is the set of its annotations: $[\![s]\!]_{x_1:A_1,\dots,x_n:A_n}=\left\{(\overline{\alpha}_1,\dots,\overline{\alpha}_n,\alpha);\ x_1^{\overline{\alpha}_1}:A_1,\dots,x_n^{\overline{\alpha}_n}:A_n\vdash s^{\alpha}:A\right\}$. Notice there is no rule for $\langle\rangle$ in Figure 2, hence $[\![\langle\rangle]\!]_\Gamma=\emptyset$ for all Γ.

Theorem 3.1 (Invariance)**.** *If* $\Gamma\vdash s\simeq_0 t:A$ *then* $[\![s]\!]_\Gamma=[\![t]\!]_\Gamma$.

Proof. We followed the standard interpretation of typed λ-calculi in cartesian closed categories, in the particular case of Rel. A direct proof is also easy, first proving a substitution lemma: if $\Gamma_0,x:A^{[\alpha_1,\dots,\alpha_k]},\Delta_0\vdash s^\beta:B$, and, for all $j\in\{1,\dots,k\}$, $\Gamma_j,\Delta_j\vdash t^{\alpha_j}:A$, then $\sum_{j=0}^k\Gamma_j,\sum_{j=0}^k\Delta_j\vdash s[x:=t]^\beta:B$. □

The relational interpretation also defines a semantics in Fin: assume a finiteness structure $\mathfrak{F}(X)$ is given for all atomic type X, so that $X^*=([\![X]\!],\mathfrak{F}(X))$ is a finiteness space, and set $(A\to B)^*=A^*\Rightarrow B^*$, $(A\times B)^*=A^*\,\&\,B^*$ and $\top^*=\mathscr{T}$. Then, further assuming that, for all $a\in\mathfrak{C}_A$, $[\![a]\!]\in\mathfrak{F}(A^*)$, we obtain:

Theorem 3.2 (Finiteness)**.** *If* $x_1:A_1,\dots,x_n:A_n\vdash s:A$ *then* $[\![s]\!]_{x_1:A_1,\dots,x_n:A_n}\in\mathfrak{F}(A_1^*\Rightarrow\cdots\Rightarrow A_n^*\Rightarrow A^*)$.

Proof. This is a straightforward consequence of the fact that the cartesian closed structure of Fin is given by the same morphisms as in Rel. A direct proof is also possible, by induction on typing derivations. □

Examples. Pure typed λ-calculi are those with no additional constant or conversion rule: fix a set $\mathfrak{A}$ of atomic types, and write $\Lambda_0^{\mathfrak{A}}$ for the calculus where $\mathfrak{C}_A=\emptyset$ for all A, and $s\simeq t$ iff $s\simeq_0 t$. This is the most basic case and we have just shown that Rel and Fin model $\simeq_0$. Be aware that if we introduce no atomic type, then the semantics is actually trivial: in $\Lambda_0^{\emptyset}$, all types and terms are interpreted by $\emptyset$.

By contrast, we can consider the internal language $\Lambda_{\underline{\mathrm{Rel}}}$ of Rel in which all relations can be described: fix $\mathfrak{A}$ as the collection of all sets (or a fixed set of sets) and $\mathfrak{C}_A=\mathfrak{P}([\![A]\!])$. Then set $s\simeq_{\underline{\mathrm{Rel}}}t$ iff $[\![s]\!]_\Gamma=[\![t]\!]_\Gamma$, for any suitable Γ. The point in defining such a monstrous language is to enable very natural notations for relations: in general, we will identify closed terms in $\Lambda_{\underline{\mathrm{Rel}}}$ with the relations they denote in the empty context. For instance, we write $id_A=\lambda x x$ with x of type A; and if $f\in\underline{\mathrm{Rel}}(A,B)$ and $g\in\underline{\mathrm{Rel}}(B,C)$, we have $g\circ f=\lambda x(g(f x))$. Similarly, the internal language $\Lambda_{\underline{\mathrm{Fin}}}$ of Fin, where $\mathfrak{A}$ is the collection of all finiteness spaces and $\mathfrak{C}_A=\mathfrak{F}(A^*)$, allows to denote conveniently all finitary relations.

The main contribution of the present paper is to establish that Fin models Gödel's system T, which can be presented in various ways. The iterator version of system T is the typed λ-calculus with an atomic type Nat of natural numbers, and constants O of type Nat, S of type $\mathsf{Nat}\to\mathsf{Nat}$ and for all type A, I_A of type $\mathsf{Nat}\to(A\to A)\to A\to A$ and subject to the following additional conversions: $\mathsf{I}\,\mathsf{O}\,u\,v>v$ and $\mathsf{I}\,(\mathsf{S}t)\,u\,v>u\,(\mathsf{I}\,t\,u\,v)$ (we will in general omit the type subscript of such parametered constants). The recursor variant is similar, but the iterator is replaced with R_A of type $\mathsf{Nat}\to(\mathsf{Nat}\to A\to A)\to A\to A$ subject to conversions $\mathsf{R}\,\mathsf{O}\,u\,v>v$ and $\mathsf{R}\,(\mathsf{S}t)\,u\,v>u\,t\,(\mathsf{R}\,t\,u\,v)$. Those systems allow to represent exactly the same functions on the set of natural numbers, where the number n is denoted by $\mathsf{S}^n\,\mathsf{O}$: this is the consequence of a normalization theorem (see [13]). In fact, we can define a recursor using iteration and products with the standard encoding $\mathsf{rec}=\lambda x\lambda y\lambda z\,\pi_l\,(\mathsf{I}\,x\,(\lambda w\,\langle y\,(\pi_r w)\,(\pi_l w),\mathsf{S}\,(\pi_r w)\rangle)\,\langle z,\mathsf{O}\rangle)$, and we get $\mathsf{rec}\,(\mathsf{S}^n\,\mathsf{O})uv\simeq\mathsf{R}\,(\mathsf{S}^n\,\mathsf{O})uv$: the idea is to reconstruct the integer argument on the fly. But this encoding is valid only for ground terms of type Nat: $\mathsf{rec}\,(\mathsf{S}t)\,u\,v\simeq u\,t\,(\mathsf{rec}\,t\,u\,v)$ holds only if we suppose t is of the form $\mathsf{S}^n\,\mathsf{O}$, or reduces to such a term. By contrast, the encoding of the iterator by $\mathsf{iter}=\lambda x\lambda y\lambda z\,(\mathsf{R}\,x\,(\lambda x'\,y)\,z)$ is extensionally valid: $\mathsf{iter}\,\mathsf{O}\,u\,v\simeq v$ and $\mathsf{iter}\,(\mathsf{S}t)\,u\,v\simeq u\,(\mathsf{iter}\,t\,u\,v)$ for all t,u,v.

The fact that one direction of the encoding holds only on ground terms indicate that the algorithmic properties of both systems may differ. And these differences will appear in the semantics (see the final section). Also, recall the discussion in our introduction: the tail recursive variant of iterator, J subject to

$\mathsf{J}(\mathsf{S}t)\,u\,v > \mathsf{J}\,t\,u\,(u\,v)$, uses its integer argument linearly. This enabled Ehrhard to define a semantics of iteration, with $\mathsf{Nat}^* = \mathscr{N} = (\mathbf{N}, \mathfrak{P}_{\mathrm{f}}(\mathbf{N}))$, $[\![\mathsf{O}]\!] = \mathscr{O} = \{0\}$ and $[\![\mathsf{S}]\!] = \mathscr{S} = \{([n], n+1);\ n \in \mathbf{N}\}$. Such an interpretation of natural numbers, however, fails to provide a semantics of I or R, in $\underline{\mathrm{Rel}}$ or $\underline{\mathrm{Fin}}$.

Lemma 3.1. *Assume $[\![\mathsf{Nat}]\!] = |\mathscr{N}|$, $[\![\mathsf{O}]\!] = \mathscr{O}$ and $[\![\mathsf{S}]\!] = \mathscr{S}$, and let A be any type such that $[\![A]\!] \neq \emptyset$. Then there is no $\mathscr{I}_A \subseteq [\![\mathsf{Nat} \to (A \to A) \to A \to A]\!]$ such that, setting $[\![\mathsf{I}_A]\!] = \mathscr{I}_A$, we obtain $[\![\mathsf{I}\,\mathsf{O}\,u\,v]\!]_\Gamma = [\![v]\!]_\Gamma$ and $[\![\mathsf{I}\,(\mathsf{S}t)\,u\,v]\!]_\Gamma = [\![u\,(\mathsf{I}\,t\,u\,v)]\!]_\Gamma$ as soon as $\Gamma \vdash t : \mathsf{Nat}$, $\Gamma \vdash u : A \to A$ and $\Gamma \vdash v : A$.*

Proof. By contradiction, assume the above equations hold. By the second equation and Theorem 3.1, $[\![\mathsf{I}\,(\mathsf{S}x)\ (\lambda z'\,y)\ z]\!] = [\![y]\!]$, and thus $x^{[]} : \mathsf{Nat}, y^{[\alpha]} : A, z^{[]} : A \vdash \mathsf{I}\,(\mathsf{S}x)\ (\lambda z'\,y)\ z^{\alpha} : A$. Inversing the rules of Figure 2, we obtain that $([], [([], \alpha)], [], \alpha) \in [\![\mathsf{I}]\!]$ and then $([([], \alpha)], [], \alpha) \in [\![\mathsf{I}\,\mathsf{O}]\!]$. Since $[\![A]\!] \neq \emptyset$, this contradicts the fact that, by the first equation: $[\![\mathsf{I}\,\mathsf{O}]\!] = [\![\lambda y\,\lambda z\,(\mathsf{I}\,\mathsf{O}\,y\,z)]\!] = [\![\lambda y\,\lambda z\,z]\!] = \{([], [\alpha], \alpha);\ \alpha \in [\![A]\!]\}$. □

4 A finitary relational interpretation of primitive recursion

Lazy natural numbers. That $x^{[]} : \mathsf{Nat}, y^{[\alpha]} : A, z^{[]} : A \vdash \mathsf{I}\,(\mathsf{S}x)\ (\lambda z'\,y)\ z^{\alpha} : A$ implies $([], [([], \alpha)], [], \alpha) \in [\![\mathsf{I}]\!]$ holds because $[\![\mathsf{S}]\!] = \mathscr{S}$ is linear, hence strict: this reflects the general fact that, if $s \in \underline{\mathrm{Rel}}(A,B)$ contains no $([], \beta)$ then, for all $t \in \underline{\mathrm{Rel}}(B,C)$, $([], \gamma)$ in $t \circ s$ iff $([], \gamma) \in t$. Such a phenomenon was also noted by Girard in his interpretation of system T in coherence spaces [13]. His evidence that there was no interpretation of the iteration operator using the linear successor relied on a coherence argument. The previous lemma is stronger: it holds in any web based model as soon as the interpretation of successor is strict.

In short, strict morphisms cannot produce anything *ex nihilo*; but the successor of any natural number should be marked as non-zero, for the iterator to distiguish between both cases. Hence the successor should be affine: similarly to Girard's solution, we will interpret Nat by so-called *lazy* natural numbers. Let $\mathscr{N}_l = (|\mathscr{N}_l|, \mathfrak{P}_{\mathrm{f}}(|\mathscr{N}_l|))$ be such that $|\mathscr{N}_l| = \mathbf{N} \cup \mathbf{N}^>$, where $\mathbf{N}^>$ is just a disjoint copy of $\mathbf{N}$. The elements of $\mathbf{N}^>$ are denoted by $k^>$, for $k \in \mathbf{N}$: $k^>$ represents a partial number, not fully determined but *strictly greater than* k. If $\nu \in |\mathscr{N}_l|$, we define ν^+ as $k+1$ if $\nu = k$ and $(k+1)^>$ if $\nu = k^>$. Then we set $\mathscr{S}_l = \{([], 0^>)\} \cup \{([\nu], \nu^+)\}$, which is affine. Notice that $\mathscr{O} \in \mathfrak{F}(\mathscr{N}_l)$ and $\mathscr{S}_l \in \mathfrak{F}(\mathscr{N}_l \Rightarrow \mathscr{N}_l)$.

Fixpoints. For all finiteness space $\mathscr{A}$, write $Rec[\mathscr{A}] = \mathscr{N}_l \Rightarrow (\mathscr{N}_l \Rightarrow \mathscr{A} \Rightarrow \mathscr{A}) \Rightarrow \mathscr{A} \Rightarrow \mathscr{A}$. We want to introduce a recursion operator $\mathscr{R}_{\mathscr{A}} \in \mathfrak{F}(Rec[\mathscr{A}])$ intuitively subject to the following definition: $\mathscr{R}\,t\,u\,v =$ match t with $\left\{ \begin{array}{lcl} \mathsf{O} & \mapsto & v \\ \mathsf{S}t' & \mapsto & u t'\,(\mathscr{R}\,t'\,u\,v) \end{array} \right.$. This definition is recursive, and a natural means to obtain such an operator is as the fixpoint of $\mathscr{S}tep = \lambda \mathscr{X}\,\lambda x \lambda y \lambda z \left(\text{match } x \text{ with } \left\{ \begin{array}{lcl} \mathsf{O} & \mapsto & z \\ \mathsf{S}x' & \mapsto & y x'\,(\mathscr{X}\,x'\,y\,z) \end{array} \right. \right)$. The cartesian closed category $\underline{\mathrm{Rel}}$ is cpo-enriched, the order on morphisms being inclusion. Hence it has fixpoints at all types: for all set A and $f \in \underline{\mathrm{Rel}}(A,A)$, the least fixpoint of f is $\bigcup_{k \geq 0} f^k \emptyset$, which is an increasing union. The least fixpoint operator is itself definable as the supremum of its approximants, $\mathscr{F}ix_A = \bigcup_{k\geq 0} \mathscr{F}ix_A^{(k)}$, where $\mathscr{F}ix_A^{(0)} = \emptyset$ and $\mathscr{F}ix_A^{(k+1)} = \lambda f \left(f \left(\mathscr{F}ix_A^{(k)}\, f \right) \right)$, more explicitly $\mathscr{F}ix_A^{(k+1)} = \left\{ ([([\alpha_1, \dots, \alpha_n], \alpha)] + \sum_{i=1}^n \overline{\varphi}_i, \alpha);\ \forall i,\ (\overline{\varphi}_i, \alpha_i) \in \mathscr{F}ix_A^{(k)} \right\}$. Notice that these approximants are finitary: if $\mathscr{A}$ is a finiteness space then, for all k, $\mathscr{F}ix_{\mathscr{A}}^{(k)} = \mathscr{F}ix_{|\mathscr{A}|}^{(k)} \in \mathfrak{F}((\mathscr{A} \Rightarrow \mathscr{A}) \Rightarrow \mathscr{A})$. The fixpoint, however, is not finitary in general: for instance $\mathscr{F}ix\,\mathscr{S}_l = \mathbf{N}^> \notin \mathfrak{F}(\mathscr{N}_l)$ hence $\mathscr{F}ix_{\mathscr{N}_l} \notin \mathfrak{F}((\mathscr{N}_l \Rightarrow \mathscr{N}_l) \Rightarrow \mathscr{N}_l)$. So we proceed in two steps: we first introduce the finitary approximants $\mathscr{R}_{\mathscr{A}}^{(k)} \in \mathfrak{F}(Rec[\mathscr{A}])$ by $\mathscr{R}_{\mathscr{A}}^{(k)} = \mathscr{S}tep_{\mathscr{A}}^k\,\emptyset$, then we prove $\mathscr{R}_{\mathscr{A}} = \bigcup_{k \geq 0} \mathscr{R}_{\mathscr{A}}^{(k)} \in \mathfrak{F}(Rec[\mathscr{A}])$.

Pattern matching on lazy natural numbers. We introduce a finitary operator $\mathscr{C}ase$, intuitively defined as: $\mathscr{C}ase\, t\, u\, v = \text{match } t \text{ with } \left\{ \begin{array}{lcl} \mathsf{O} & \mapsto & v \\ \mathsf{S}t' & \mapsto & u t' \end{array} \right.$. More formally:

Definition 4.1. *If $\overline{\nu} = [\nu_1, \ldots, \nu_k] \in |\mathscr{N}_l|^!$, we write $\overline{\nu}^+ = \left[\nu_1^+, \ldots, \nu_n^+\right]$. Then for all set A, let $\mathscr{C}ase_A = \{([0], [], [\alpha], \alpha);\ \alpha \in A\} \cup \left\{([0^>] + \overline{\nu}^+, [(\overline{\nu}, \alpha)], [], \alpha);\ \overline{\nu} \in |\mathscr{N}_l|^! \wedge \alpha \in A\right\}$.*

Lemma 4.1. *Pattern matching is finitary: $\mathscr{C}ase_{\mathscr{A}} = \mathscr{C}ase_{|\mathscr{A}|} \in \mathfrak{F}(\mathscr{N}_l \Rightarrow (\mathscr{N}_l \Rightarrow \mathscr{A}) \Rightarrow \mathscr{A} \Rightarrow \mathscr{A})$. Moreover, $y : \mathscr{N}_l \Rightarrow \mathscr{A}, z : \mathscr{A} \vdash \mathscr{C}ase\, \mathscr{O}\, y\, z \simeq z : \mathscr{A}$ and $x : \mathscr{N}_l, y : \mathscr{N}_l \Rightarrow \mathscr{A}, z : \mathscr{A} \vdash \mathscr{C}ase\, (\mathscr{S}_l x)\, y\, z \simeq y\, x : \mathscr{A}$.*

Proof. That the equations hold is a routine exercise. To prove $\mathscr{C}ase$ is finitary, we check the definition of $\mathfrak{F}(- \Rightarrow -)$. For the first direction: for all $n \in \mathfrak{F}(\mathscr{N}_l)$, $\mathscr{C}ase\, n \subseteq \{([], [\alpha], \alpha);\ \alpha \in |\mathscr{A}|\} \cup \left\{([(\overline{\nu}, \alpha)], [], \alpha);\ \overline{\nu}^+ \in n^! \wedge \alpha \in |\mathscr{A}|\right\}$; hence, setting $n' = \{\nu;\ \nu^+ \in n\} \in \mathfrak{F}(\mathscr{N}_l)$, we obtain $\mathscr{C}ase\, n \subseteq (\lambda y \lambda z z) \cup (\lambda y \lambda z (y n'))$, and we conclude since the union of two finitary subsets is finitary. In the reverse direction, we prove that, for all $\gamma \in |(\mathscr{N}_l \Rightarrow \mathscr{A}) \Rightarrow \mathscr{A} \Rightarrow \mathscr{A}|$, setting $N' = \mathscr{C}ase^{\perp} \cdot \{\gamma\}$, $n^! \cap N'$ is finite; this is immediate because N' has at most one element. □

A recursor in Rel. We introduce the relation $\mathscr{R}$ as the fixpoint of $\mathscr{S}tep$.

Definition 4.2. *Fix a set A. Let $\mathscr{S}tep_A = \lambda \mathscr{X}\, \lambda x \lambda y \lambda z (\mathscr{C}ase_A\, x\, (\lambda x' (y x' (\mathscr{X} x' y z)))\, z)$. and, for all $k \in \mathbf{N}$, let $\mathscr{R}_A^{(k)} = \mathscr{S}tep_A^k\, \emptyset$. Then we define $\mathscr{R}_A = \bigcup_{k \geq 0} \mathscr{R}_A^{(k)}$, and fix $[\![\mathsf{R}]\!] = \mathscr{R}$.*

Lemma 4.2. *For all finiteness space $\mathscr{A}$, $\mathscr{S}tep_{\mathscr{A}} = \mathscr{S}tep_{|\mathscr{A}|} \in \mathfrak{F}(Rec[\mathscr{A}] \Rightarrow Rec[\mathscr{A}])$ and, for all k, $\mathscr{R}_{\mathscr{A}}^{(k)} = \mathscr{R}_{|\mathscr{A}|}^{(k)} \in \mathfrak{F}(Rec[\mathscr{A}])$. Moreover, we have: $\mathscr{R}_{\mathscr{A}}^{(0)} = \emptyset$ and $\mathscr{R}_{\mathscr{A}}^{(k+1)} = \{([0], [], [\alpha], \alpha);\ \alpha \in |\mathscr{A}|\} \cup \left\{\left([0^>] + \sum_{i=0}^n \overline{\nu}_i^+, [(\overline{\nu}_0, [\alpha_1, \ldots, \alpha_n], \alpha)] + \sum_{i=1}^n \overline{\varphi}_i, \sum_{i=1}^n \overline{\alpha}_i, \alpha\right);\ \forall i,\ (\overline{\nu}_i, \overline{\varphi}_i, \overline{\alpha}_i, \alpha_i) \in \mathscr{R}_{\mathscr{A}}^{(k)}\right\}$.*

Proof. The finiteness of the approximants follows from Theorem 3.2. The explicit description of $\mathscr{R}_{\mathscr{A}}^{(k)}$ is a direct application of the definition of the relational semantics. □

Theorem 4.3 (Correctness)**.** *For all suitable Γ and Δ, $[\![\mathsf{R}\, \mathsf{O}\, y\, z]\!]_{\Gamma} = [\![z]\!]_{\Gamma}$ and $[\![\mathsf{R}\, (\mathsf{S} x)\, y\, z]\!]_{\Delta} = [\![y\, x\, (\mathsf{R} x y z)]\!]_{\Delta}$.*

Proof. This follows directly from Lemma 4.1 and the fact that $\mathscr{R} = \mathscr{S}tep\, \mathscr{R}$. □

Finiteness. It only remains to prove $\mathscr{R}$ is finitary. Following the definition of $(- \Rightarrow -)$, we proceed in two steps: the image of a finitary subset of $\mathscr{N}_l$ is finitary; conversely, the preimage of a singleton is "anti-finitary".

Definition 4.4. *If $\overline{\alpha} = [\alpha_1, \ldots, \alpha_k] \in a^!$, we denote the support of $\overline{\alpha}$ by $\mathrm{Supp}(\overline{\alpha}) = \{\alpha_1, \ldots, \alpha_k\} \subseteq a$, and the size of $\overline{\alpha}$ by $\#(\overline{\alpha}) = k$. If $n \in \mathfrak{F}(\mathscr{N}_l)$, we set $\max(n) = \max\{k;\ k \in n \vee k^> \in n\}$, with the convention $\max(\emptyset) = 0$. Then if $\overline{\nu} \in |\mathscr{N}_l|^!$ we set $\max(\overline{\nu}) = \max(\mathrm{Supp}(\nu))$, and if $\overline{n} \subseteq n^!$ for some $n \in \mathfrak{F}(\mathscr{N}_l)$, $\max(\overline{n}) = \max(\bigcup_{\overline{\nu} \in \overline{n}} \mathrm{Supp}(\overline{\nu}))$.*

Lemma 4.3. *For all $\gamma = (\overline{\nu}, \overline{\varphi}, \overline{\alpha}, \alpha) \in \mathscr{R}_{\mathscr{A}}$, $\gamma \in \mathscr{R}_{\mathscr{A}}^{(\max(\overline{\nu})+1)}$.*

Proof. By induction on $\max(\overline{\nu})$, using Lemma 4.2. □

Lemma 4.4. *If $n \in \mathfrak{F}(\mathscr{N}_l)$, then $\mathscr{R}_{\mathscr{A}}\, n \in \mathfrak{F}((\mathscr{N}_l \Rightarrow \mathscr{A} \Rightarrow \mathscr{A}) \Rightarrow \mathscr{A} \Rightarrow \mathscr{A})$.*

Proof. The previous Lemma entails $\mathscr{R}_{\mathscr{A}}\, n = \mathscr{R}_{\mathscr{A}}^{(\max(n)+1)}\, n$. We conclude recalling that $\mathscr{R}_{\mathscr{A}}^{(\max(n)+1)}\, n \in \mathfrak{F}((\mathscr{N}_l \Rightarrow \mathscr{A} \Rightarrow \mathscr{A}) \Rightarrow \mathscr{A} \Rightarrow \mathscr{A})$, because $\mathscr{R}_{\mathscr{A}}^{(\max(n)+1)} \in Rec[\mathscr{A}]$. □

Definition 4.5. *For all* $\overline{\varphi} = [(\overline{v}_1, \overline{\alpha}_1, \alpha_1), \ldots, (\overline{v}_k, \overline{\alpha}_k, \alpha_k)] \in |\mathscr{N}_l \Rightarrow \mathscr{A} \Rightarrow \mathscr{A}|^!$*, let* $\#\#(\overline{\varphi}) = \sum_{j=1}^{k} \#(\overline{v}_j)$.

Lemma 4.5. *If* $(\overline{v}, \overline{\varphi}, \overline{\alpha}, \alpha) \in \mathscr{R}_{\mathscr{A}}$*, then* $\#(\overline{v}) = \#(\overline{\alpha}) + \#(\overline{\varphi}) + \#\#(\overline{\varphi})$.

Proof. Using Lemma 4.2, the result is proved for all $(\overline{v}, \overline{\varphi}, \overline{\alpha}, \alpha) \in \mathscr{R}_{\mathscr{A}}^{(k)}$, by induction on k. □

Theorem 4.6 (The recursion operator is finitary)**.** $\mathscr{R}_{\mathscr{A}} \in \mathfrak{F}(Rec[\mathscr{A}])$.

Proof. By Lemma 4.4, we are left to prove that, for all $n \in \mathfrak{F}(\mathscr{N}_l)$ and $\gamma \in |(\mathscr{N}_l \Rightarrow \mathscr{A} \Rightarrow \mathscr{A}) \Rightarrow \mathscr{A} \Rightarrow \mathscr{A}|$, $N = n^! \cap (\mathscr{R}^{\perp} \cdot \{\gamma\})$ is finite. But by Lemma 4.5,

$$N \subseteq \left\{ \overline{v} \in |\mathscr{N}_l|^!;\ \#(\overline{v}) = \#(\overline{\alpha}) + \#(\overline{\varphi}) + \#\#(\overline{\varphi}) \wedge \max(\overline{v}) \leq \max(n) \right\}$$

which is finite. □

Remark 4.7. *We keep calling* $\mathscr{R}$ *"the" recursion operator, but notice such an operator is not unique in* <u>Rel</u> *or* <u>Fin</u>*: let* $\mathscr{C}ase'_A = \{([0,0], [], [\alpha], \alpha);\ \alpha \in A\} \cup \left\{ ([0^>] + \overline{v}^+, [(\overline{v}, \alpha)], [], \alpha);\ \overline{v} \in |\mathscr{N}_l|^! \wedge \alpha \in A \right\}$*, for instance; this variant of matching operator behaves exactly like* $\mathscr{C}ase$*, and one can reproduce our construction of the recursor based on that.*

5 About iteration

We have just provided a semantics of system T with recursor. Now let $\mathscr{I}_A = \lambda x \lambda y \lambda z (\mathscr{R}_A x (\lambda x' y) z)$ for all set A. By Theorem 4.6, $\mathscr{I}_{\mathscr{A}} = \mathscr{I}_{|\mathscr{A}|} \in \mathfrak{F}(Iter[\mathscr{A}])$. Moreover, by Theorem 4.3 this defines an iteration operator and we obtain that the triple $(|\mathscr{N}_l|, \mathscr{O}, \mathscr{S}_l)$, resp. $(\mathscr{N}_l, \mathscr{O}, \mathscr{S}_l)$, is a weak natural number object [11, 12] in the cartesian closed category <u>Rel</u>, resp. <u>Fin</u>.

We now develop a semantic argument demonstrating how recursion is stricly stronger than iteration. One distinctive feature of both models is non-uniformity: if $a, a' \in \mathfrak{F}(\mathscr{A})$ then $a \cup a' \in \mathfrak{F}(\mathscr{A})$; and in the construction of $a^!$, there is no restriction on the elements of the multisets we consider. It is very different from the setting of coherence spaces for instance. But we can show the iterator only considers uniform sets of lazy numbers, in the following sense: if $k \in \mathbf{N}$, we define $\underline{k} = \mathscr{S}_l^k \mathscr{O} = \{l^>;\ l < k\} \cup \{k\} \in \mathfrak{F}(\mathscr{N}_l)$; we say $n \subseteq |\mathscr{N}_l|$ is *uniform* if $n \subseteq \underline{k}$ for some k. Notice that, in the coherence space of lazy natural numbers used by Girard in [13] to interpret system T, the sets $\underline{k}$ are the finite maximal cliques: coherence is given by $k \frown l$ iff $k = l$, $k \frown l^>$ iff $k > l$ and $k^> \frown l^>$ for all k, l. The only infinite maximal clique is $\mathbf{N}^>$ (recall this is the fixpoint of $\mathscr{S}_l$). We prove $\mathscr{I}$ considers only uniform sets of lazy numbers.

For all k, let $\mathscr{I}_{\mathscr{A}}^{(k)} = \lambda x \lambda y \lambda z \left(\mathscr{R}_{\mathscr{A}}^{(k)} x (\lambda x' y) z \right)$. Then let $\mathscr{S}tage_{\mathscr{A}}^{(0)} = \{([0], [], [\alpha], \alpha);\ \alpha \in |\mathscr{A}|\}$; $\mathscr{S}tage_{\mathscr{A}}^{(1)} = \{([0^>], [([], \alpha)], [], \alpha);\ \alpha \in |\mathscr{A}|\}$; and, for all $k > 0$, $\mathscr{S}tage_{\mathscr{A}}^{(k+1)} = \mathscr{I}_{\mathscr{A}}^{(k+1)} \setminus \mathscr{I}_{\mathscr{A}}^{(k)}$. One can check that $\mathscr{I}_{\mathscr{A}} = \bigcup_{k \geq 0} \mathscr{S}tage_{\mathscr{A}}^{(k)}$.

Lemma 5.1. *If* $\mathscr{A} \neq \mathscr{T}$ *then, for all* $k \in \mathbf{N}$, $\bigcup \left\{ \mathrm{Supp}(\overline{v});\ \exists (\overline{\varphi}, \overline{\alpha}, \alpha),\ (\overline{v}, \overline{\varphi}, \overline{\alpha}, \alpha) \in \mathscr{S}tage_{\mathscr{A}}^{(k)} \right\} = \underline{k}$.

Proof. The inclusion $\subseteq$ is easy by induction on $\underline{k}$. For $\supseteq$, consider $\lambda x \lambda z \left(\mathscr{I}^{(k)} x (\lambda z' z') z \right)$. □

As a consequence, for all $(\overline{v}, \overline{\varphi}, \overline{\alpha}, \alpha) \in \mathscr{I}$, $\mathrm{Supp}(\overline{v})$ is uniform. Of course, no such property holds for $\mathscr{R}$, because $\mathscr{R}_{\mathscr{A}}^{(1)} \supseteq \left\{ ([0^>] + \overline{v}^+, [(\overline{v}, [], \alpha)], [], \alpha);\ \alpha \in |\mathscr{A}| \wedge \overline{v} \in |\mathscr{N}_l|^! \right\}$. An immediate generalization is that no recursor can be derived from $\mathscr{I}$: the interpretation of any recursor on the natural number object $(\mathscr{N}_l, \mathscr{O}, \mathscr{S}_l)$ necessarily contains elements of the above form.

Aknowledgements

The present work stems from a discussion with Thomas Ehrhard. It also greatly benefited from many working sessions in the company of Christine Tasson, to whom I am most grateful.

References

[1] Thomas Ehrhard. Finiteness spaces. *Mathematical Structures in Computer Science*, 15(4):615–646, 2005.

[2] Jean-Yves Girard. Normal functors, power series and lambda-calculus. *Annals of Pure and Applied Logic*, 37(2):129–177, 1988.

[3] Thomas Ehrhard and Laurent Regnier. The differential lambda-calculus. *Theoretical Computer Science*, 309:1–41, 2003.

[4] Thomas Ehrhard and Laurent Regnier. Differential interaction nets. *Electr. Notes Theor. Comput. Sci.*, 123:35–74, 2005.

[5] Thomas Ehrhard and Laurent Regnier. Böhm trees, Krivine's machine and the Taylor expansion of λ-terms. In Arnold Beckmann, Ulrich Berger, Benedikt Löwe, and John V. Tucker, editors, *CiE*, volume 3988 of *Lecture Notes in Computer Science*, pages 186–197. Springer, 2006.

[6] Thomas Ehrhard and Olivier Laurent. Interpreting a finitary pi-calculus in differential interaction nets. In Luis Caires and Vasco T. Vasconcelos, editors, *Concurrency Theory (CONCUR '07)*, volume 4703 of *Lecture Notes in Computer Science*, pages 333–348. Springer, September 2007.

[7] Paolo Tranquilli. Intuitionistic differential nets and lambda-calculus. To appear in Theor. Comput. Sci, 2008.

[8] Lionel Vaux. Differential linear logic and polarization. In Curien [15], pages 371–385.

[9] Christine Tasson. Algebraic totality, towards completeness. In Curien [15], pages 325–340.

[10] Michele Pagani and Christine Tasson. The inverse Taylor expansion problem in linear logic. In *LICS*, 2009.

[11] Marie-France Thibault. Pre-recursive categories. *Journal of Pure and Applied Algebra*, 24:79–93, 1982.

[12] J. Lambek and P. J. Scott. *Introduction to higher order categorical logic*. Cambridge University Press, New York, NY, USA, 1988.

[13] Jean-Yves Girard, Paul Taylor, and Yves Lafont. *Proofs and types*. CUP, Cambridge, 1989.

[14] Antonio Bucciarelli, Thomas Ehrhard, and Giulio Manzonetto. Not enough points is enough. In *Computer Science Logic*, volume 4646 of *Lecture Notes in Computer Science*, pages 298–312. Springer Berlin, 2007.

[15] Pierre-Louis Curien, editor. *Typed Lambda Calculi and Applications, 9th International Conference, TLCA 2009, Brasilia, Brazil, July 1-3, 2009. Proceedings*, volume 5608 of *Lecture Notes in Computer Science*. Springer, 2009.