# On the discriminating power of tests in resource lambda-calculus

Flavien Breuvart

**HAL Id: hal-00698609**

**https://hal.archives-ouvertes.fr/hal-00698609v2**

Submitted on 22 May 2012

# On the discriminating power of tests in resource $\lambda$-calculus

May 22, 2012

**Abstract**

Since its discovery, differential linear logic (DLL) inspired numerous domains. In denotational semantics, categorical models of DLL are now commune, and the simplest one is Rel, the category of sets and relations. In proof theory this naturally gave birth to differential proof nets that are full and complete for DLL. In turn, these tools can naturally be translated to their intuitionistic counterpart. By taking the co-Kleisly category associated to the ! comonad, Rel becomes MRel, a model of the $\lambda$-calculus that contains a notion of differentiation. Proof nets can be used naturally to extend the $\lambda$-calculus into the lambda calculus with resources, a calculus that contains notions of linearity and differentiations. Of course MRel is a model of the $\lambda$-calculus with resources, and it has been proved adequate, but is it fully abstract?

That was a strong conjecture of Bucciarelli, Carraro, Ehrhard and Manzonetto in [4]. However, in this paper we exhibit a counter-example. Moreover, to give more intuition on the essence of the counter-example and to look for more generality, we will use an extension of the resource $\lambda$-calculus also introduced by Bucciarelli *et al* in [4] for which $\mathcal{M}_\infty$ is fully abstract, the tests.

## 1 Introduction

The first extension of the $\lambda$-calculus with resources, by Boudol in 1993 [1], was introducing a special resource sensitive application that may involve multisets of affine arguments (each one has to be used at most one time). This was a natural way to export resource sensitiveness to the functional setting. However, gathering no known and interesting properties (confluence, linearity...), it was not fully explored.

Later on, Ehrhard and Regnier, working on functional interpretation of differential proof nets, discovered a calculus similar to Boudol's one, named differential $\lambda$-calculus [10]. By adding to the $\lambda$-calculus a derivative operation $\frac{\partial M}{\partial x}(N)$, which syntactically corresponds to a linear substitution of $x$ by $N$ in $M$, it recovers the resource-sensitiveness. This is done through the translation

1

$(\lambda x.M)\ [N_1, ..., N_n; N^!] \simeq (\lambda x.\frac{\partial^{(n)}M}{x}(N_1,..,N_n))\ N$ where $N_i$ are the linear arguments and $N$ is the non linear one. This more semantical view even allow for the generalisation of the operation and recover excellent semantical properties (confluence, Taylor expansion...). We will adopt the syntax of [13] that re-implements improvements from differential $\lambda$-calculus into Boudol's calculus, and we will call it $\partial\lambda$-calculus.

The category Rel of set and relations is known to model the linear logic, and, despite its high degree of degeneration ($Rel^{op} = Rel$), it is a very natural construction. Indeed, what appeared to be a degeneration is in reality a natural choice that preserves all proofs, *i.e.* the interpretation function from proof to MRel is injective up to isomorphism ([8]). But our principal interest for this category is that it models the differential linear logic, and of known such category it is the simplest and more natural.

As for every categorical model of linear logic, the interpretation of the ! induced a comonad. From that comonad we can construct the co-kleisly category. In the case of Rel, this new category, MRel, corresponds to the category of sets with, as morphisms from $A$ to $B$, the relations from $\mathcal{M}_f(A)$ (the finite multisets over $A$) to $B$. It is then a model of the $\lambda$-calculus and of the $\partial\lambda$-calculus. This construction being the most natural we can do, MRel is, *a priori*, one of the most natural models of the $\partial\lambda$-calculus (even if non well pointed [6]).

It is only natural, then, to question on the depth of the link among the reflexive elements of MRel and the $\partial\lambda$-calculus. And more precisely among MRel's canonical reflexive element $\mathcal{M}_\infty$ and the $\partial\lambda$-calculus. Until now we knew that $\mathcal{M}_\infty$ ([6]) was adequate for the $\lambda$-calculus, *i.e.* that two terms carrying the same interpretations in MRel behave the same way in all contexts. But we did not know anything about the counterpart, named full abstraction.

This question has been thoroughly studied, however, since $\mathcal{M}_\infty$ has been proved (resp. in [12], [5] and [4]) fully abstract not only for both of the principal sub-calculi of $\partial\lambda$-calculus, namely the usual $\lambda$-calculus and Kfoury's linear calculus of [11], but also for the extension with tests of [4] (denoted $\tau\partial\lambda$-calculus). Therefore Bucciarelli *et al* emit in [4] a strong conjecture of full abstraction for the $\partial\lambda$-calculus.

However, and it is our purpose here, a counter example can be found. In order to exhibit this counter example, we will take an unusual shortcut using full abstraction result for $\tau\partial\lambda$-calculus. Indeed, we will prove a slightly more general theorem: the failure of full abstraction for $\partial\lambda$-calculus of any model that is fully abstract for $\tau\partial\lambda$-calculus. Due to this generalization we will not have to introduce the full description of $\mathcal{M}_\infty$ in the core of the article (it is available in annexes).

Additionally to be considerably easier and more intuitive than the direct and usual method, this way of proceeding is part of a larger study of full abstraction. Indeed, we are looking for a mechanical way to tackle full abstraction problems in two steps. First we extend the calculus with well chosen semantical objects in order to reach the definability of compact elements. Then we study the full abstraction question indirectly via the link between the operational equivalence

of the original calculus and of its artificial extension. This reduces our mix of semantic and syntactic question to a purely syntactic one, allowing us to use powerful syntactic constructions.

Tests where introduced in [2] to have a full abstraction theorem for Boudol $\lambda$-calculus with resources. Later on, the principle was improved in [4], implementing semantic objects in the syntax in order to get full abstraction of $\mathcal{M}_\infty$, following an idea of [9]. This extension can, in our context, be compared to a basic exception mechanism. The term $\bar{\tau}(Q)$ is raising the exception (or test) $Q$, absorbing all its non resources-limited applications and the exception $\tau(M)$ is catching any exception in $M$ by annihilating all the head-$\lambda$-abstraction. The most important here being the scope of the $\tau(M)$ that act as an infinite application over $M$.

**Notations:** $\lambda\bar{x}^n$ will be used for $\lambda x_1, ..., x_n$ ($n$ is not specified when it can be any integer) and $I$ will denotes the identity $\lambda x.x$.

# 2 Background

## 2.1 $\partial\lambda$-calculus

As explained, this article is directly following [4]. For this reason we need to introduce the $\partial\lambda$-calculus and then the tests. In the $\partial\lambda$-calculus , the notion of linearity is capital. Any term in linear position will never suffer any duplication or erasing regardless the reduction strategy. Linear subterms are subterms that are either the first subterm of a lambda abstraction in linear position, the left side of an application that is in linear position, or in the linear part of its right side. The last case is the real improvement and asks for arguments to be separated in linear and non linear arguments. Therefore, the right side of the applications will be replaced by a new kind of expression different from terms, the "bags". Bags are multisets containing some linear (non banged) arguments and exactly one non linear (banged) argument:

$$\text{(terms)} \quad M, N: \quad \lambda x.M \mid M\ B \mid M{+}N \mid 0$$
$$\text{(bags)} \quad B, C: \quad [M_1, ..., M_n; M^!] \mid B + C \mid 0$$

This is the syntax of [4] modulo the macro $M{+}N = (\lambda x.x)\ [\{M{+}N\}^!]$.

For convenience, the finite sums will be denoted $\Sigma_i Q_i$ and the different $0$'s are just the neutral elements of the different sums. This demonic sum had to be implemented since we want the calculus to be resource sensitive and confluent, thus there is no other choice than to considere the sum of all the possible outcomes. Sums distribute with any linear context:

$$\lambda x.(\Sigma_i M_i) = \Sigma_i(\lambda_x.M_i) \qquad (\Sigma_i M_i)\ (\Sigma_j N_j) = \Sigma_{i,j}(M_i\ N_j)$$
$$([(\Sigma_{i_1 \le k_1} M_{i_1}^1), ..., (\Sigma_{i_n \le k_n} M_{i_n}^n); M^!]) = \Sigma_{(i_j)_j \le (k_j)_j}[M_{i_1}^i, ..., M_{i_n}^n; M^!]$$

In the application, each linear argument will replace one and only one occurrence of the variable, thus the need of two kinds of substitutions, the usual one, denoted $\{.\}$, and the linear one, denoted $\langle.\rangle$. This last will act like a derivation $M\langle N/x\rangle \sim \frac{\partial M}{\partial x}(N)$:

$$x\langle N/x\rangle = N \qquad\qquad x\langle N/y\rangle = 0 \qquad\qquad (\lambda y.M)\langle N/x\rangle = \lambda y.(M\langle N/x\rangle)$$
$$([M_1,...,M_n;M^!])\langle N/x\rangle \;=\; (\Sigma_{i=1}^n[M_1,..,M_i\langle N/x\rangle,...M_n;M^!]) + [M_1,...,M_n,M\langle N/x\rangle;M^!]$$
$$(M\ P)\langle N/x\rangle = (M\langle N/x\rangle\ P) + (M\ P\langle N/x\rangle)$$

This enables us to describe the $\beta$-reduction:

$$(\beta)\qquad (\lambda x.M)[N_1,...,N_n;N^!]\quad\to M\langle N_1/x\rangle\cdots\langle N_n/x\rangle\{N^!/x\}$$

In other words $(\lambda x.M)[N_1,...,N_n;N^!]\to \frac{\partial^n M}{\partial^n x}(N_1,...,N_n)(N)$

## 2.2 $\tau\partial\lambda$-calculus

In differential proof nets the 0-ary tensor and the 0-ary par can be added freely in the sense that we still have a natural interpretation in MRel and $\mathcal{M}_\infty$. These operations can be translated in our calculus as an exception mechanism. With on one side a $\tau(Q)$ that "raises" the exception (or test) $Q$ by burning its applicative context (whenever these applications do not have any linear component, otherwise it diverges). And with on the other side a $\bar\tau(M)$ that "catch" the exceptions in $M$ by burning the abstraction context of $M$ (whenever this abstraction is dummy). The main difference with a usual exception system is the divergence of the catch if no exception are raised.

We introduce a new operators and a new kind of expression that will play the role of exception, the tests:

$$(\text{terms})\quad M,N:\bar\tau(Q)$$
$$(\text{test})\qquad Q,R:\qquad \epsilon\mid Q|R\mid \tau(M)\mid Q{+}R\mid 0$$

New operator immediately imply new distribution rules for the sum and the linear substitution:

$$\tau(\Sigma_i M_i) = \Sigma_i\tau(M_i)\qquad \bar\tau(\Sigma_i Q_i) = \Sigma_i\bar\tau(Q_i)\qquad \|_j\Sigma_i Q_{i(j)} = \Sigma_i\|_j Q_{i(j)}$$
$$\tau(M)\langle N/x\rangle = \tau(M\langle N/x\rangle)\qquad \bar\tau(Q)\langle N/x\rangle = \bar\tau(Q\langle N/x\rangle)$$
$$(Q{+}R)\langle N/x\rangle = Q\langle N/x\rangle{+}R\langle N/x\rangle\qquad (Q|R)\langle N/x\rangle = Q\langle N/x\rangle|R\langle N/x\rangle$$

Here is the corresponding operational semantics:

$$
\begin{array}{clcl}
(\gamma) & \tau[\bar\tau(Q)] & \to & Q\\[4pt]
(\tau) & \tau(\lambda x.M) & \to & \tau(M\{0/x\})\\[4pt]
(\bar\tau_1) & (\bar\tau(Q))\,[M^!] & \to & \bar\tau(Q)\\[4pt]
(\bar\tau_2) & (\bar\tau(Q))\,[M_1,...,M_{n\geq 1};M^!] & \to & 0\\[4pt]
(\epsilon) & \epsilon|\epsilon & \to & \epsilon
\end{array}
$$

The intuition of $\bar\tau(Q)$ is an operator that take a test (a Boolean value), compute it and returns an infinite $\lambda$-abstraction with no occurrence of the abstracted variables. The test $\tau(M)$ is taking a term and returns a successful test if the term is converging in a context that consists of an infinite empty application.

## 2.3 Observational order and full abstraction

In order to ask for full abstraction, one has to specify a reduction strategy. A natural choice would be the head reduction, but this would make $(\lambda x.x\,[0,0])$ a normal form while no applicative instantiation of $x$ allow the convergence of this term. Therefore, the reduction strategy we are considering will not be head-reduction, but the outer-head-reduction. This reduction will reduce subterms in linear position after the subterms in head positions ([14]). The corresponding normal forms are terms and tests of the form:

$$M + \lambda\bar{x}.y\,[N_{1,1}, ..., N_{1,k_1}; L_1^!] \cdots [N_{n,1}, ..., N_{n,k_n}; L_n^!]$$
$$M + \lambda\bar{x}.\bar{\tau}(Q)$$
$$Q + (\|\tau(N_i))$$

Where every $N_-$ and $Q$ must be in outer-head normal forms and can't be a sum (but the $L_i$ are of any kind).

**Definition 1** *M is observationally below N, if for all context $C(\!|.|\!)$, we have $C(\!|N|\!)$ which is outer-head-converging whenever $C(\!|M|\!)$ is outer-head-converging. They are observationally equivalent if moreover N is observationally below M*

In the particular case of the $\tau\partial\lambda$-calculus , we can easily restrict contexts to test-contexts, which is contexts whose output is a tests. This will be applied systematically for simplification.
We will denote $\leq_{\tau\partial}$ and $\equiv_{\tau\partial}$ the observational order and equivalence of the $\tau\partial\lambda$-calculus and $\leq_\partial$ and $\equiv_\partial$ those of the $\partial\lambda$-calculus .

Bucciarelli, Carraro, Ehrhard and Manzonetto were then able to prove a strong theorem relating the model to the calculus:

**Theorem 1** *$\mathcal{M}_\infty$ is fully abstract for the $\Lambda$-calculus with resources and tests: for all closed terms $M, N$ with resources ans tests,*

$$[\![M]\!] = [\![N]\!] \iff M \equiv_{\tau\partial} N$$

# 3 The counter-example

In order to exhibit our counter-example we will use the following property:

**Fact 1** *Let $\mathcal{B}$ a calculus and $\mathcal{A}$ a super-calculus. Let $\mathcal{M}$ a model that is fully abstract for $\mathcal{A}$. $\mathcal{M}$ is fully abstract for $\mathcal{B}$ iff the operational equivalences of $\mathcal{B}$ and $\mathcal{A}$ are equal on their domain intersection.*

In our context it means that, in order to prove the non full abstraction for the $\partial\lambda$-calculus , it is sufficient to find two terms of the $\partial\lambda$-calculus that cannot be separated by any context of the $\partial\lambda$-calculus but that are separated by a context of the $\tau\partial\lambda$-calculus . This makes the research and the proof quite easier when the terms of the $\partial\lambda$-calculus involved are complex but not the context of the $\tau\partial\lambda$-calculus .

We are firstly exhibiting a term $A$ of the $\partial\lambda$-calculus that is observationally above the identity in the $\partial\lambda$-calculus ', but not in the $\tau\partial\lambda$-calculus 's observational order:

$$A = \Theta \ [\lambda uvw.w \ [I \ [v^!]] \ , \ (\lambda uvw.u \ [(v \ [w^!])^!])^!] \tag{1}$$

where $\Theta$ is the Turing fix point combinator:

$$(\lambda gu.u[(g \ [(g \ [u^!])^!])^!]) \ [(\lambda gu.u[(g \ [(g \ [u^!])^!])^!])^!]$$

This term seems quite complex, but, modulo $\eta$-equivalence, $A$ reduces exactly to any of the elements of the following sum, and thus can be think as an equivalent:

$$\Sigma_{n\geq 1}B_n = \Sigma_{n\geq 1}\lambda\bar{u}^n w.w \ [I \ [u_1^!] \ [u_2^!] \cdots [u_n^!]] \tag{2}$$

This is due to the following property:

**Lemma 1** *If* $A_i = \lambda\bar{x}^{i+1}.A \ (x_1 \ [x_2^!] \cdots [x_{i+1}^!])$ *then* $A_0 \rightarrow_\eta A$ *and for all* $i$,
$$A_i \rightarrow A_{i+1} + B_{i+1}$$

**Proof.** Simple reduction unfolding the $\Theta$ once. $\square$

In absence of tests, this term has a comportment similar to $\epsilon_0$ in the sense that it will converges in any applicative context provided that these applications do not carry linear components. In particular it converges more often than the identity:

**Lemma 2** *For all context* $C(\!|.|\!)$ *of the* $\partial\lambda$-calculus *, if* $C(\!|I|\!)$ *converges then* $C(\!|A|\!)$ *converges, i.e.* $I \leq_\partial A$

**Proof.** Let $C(\!|.|\!)$ a context that converge on $I$
With the context lemma ([14]), and since neither $I$ nor $A$ has free variables, we can assume that $C(\!|.|\!) = (\!|.|\!) \ P_1 \cdots P_k$, thus by lemma 1, we have $A \rightarrow^* U + B_k$
and
$$C(\!|A|\!) \rightarrow^* U' + \lambda w.w \ [I \ P_1 \cdots P_k] = U' + \lambda w.w \ [C(\!|M|\!)]$$
converges. $\square$

But in presence of real tests, its comportment appeared to be different that $\epsilon_0$ in the sense that it diverges under a $\bar{\tau}$. In particular it is not observationally above the identity in $\tau\partial\lambda$-calculus :

**Lemma 3** *In the* $\tau\partial\lambda$-calculus *,* $\tau(A \ [\epsilon_0^!])$ *diverges, while* $\tau(I \ [\epsilon_0^!])$ *converges, i.e.* $I \not\leq_{\tau\partial} A$

**Proof.** For all $i$, $\tau(A_i \ [\epsilon_0^!])$ diverges since, by co-induction:

$$
\begin{aligned}
\tau(A_i \ [\epsilon_0]) \quad &\rightarrow \quad \tau((A_{i+1} + (\lambda\bar{x}^{i+1}y.y \ [I \ [x_1^!] \cdots [x_{i+1}^!]])) \ [\epsilon_0]) \\
&= \quad \tau(A_{i+1} \ [\epsilon_0]) + \tau((\lambda\bar{x}^{i+1}y.y \ [I \ [x_1^!] \cdots [x_{i+1}^!]]) \ [\epsilon_0]) \\
&\rightarrow \quad \tau(A_{i+1} \ [\epsilon_0]) + \tau(\lambda\bar{x}^i y.y \ [\epsilon_0 \ [x_1^!] \cdots [x_{i+1}^!]]) \\
&\rightarrow^* \quad \tau(A_{i+1} \ [\epsilon_0]) + \tau(0 \ [\epsilon_0 \ [0^!] \cdots [0^!]])
\end{aligned}
$$

The non outer-head convergence comes with the co-induction hypothesis for the first term, and is trivial for the second. $\square$

Hence, we have broken the conjecture concerning the equality between the observational and denotational orders. Let's break the whole conjecture:

**Theorem 2** $\mathcal{M}_\infty$ *is not fully abstract for the* $\lambda$-calculus *with resources*

# 4 Further works

First a diligent reader will remark that we have a critical use of the demonic sum which is very powerful in this calculus. And an even more diligent one will remark that an arbitrary choice have been made concerning this sum: we could differentiate terms and reduced of terms and remove sums from the original syntax (they just have to appear in reductions of terms). The choice we made here corresponds to the one of [4] and carries an understandable counter-example. But we claim that another equivalent counter-example arises for the case with limited sum. This counter-example is, however, a little more complicated and make it necessary to rework the material of [4] (even if everything works exactly the same way).

Our counter-example can be translated to some related cases. In particular, to prove non full abstraction of Scott's $\mathcal{D}_\infty$ for the $\lambda$-calculus with angelic and demonic sums (conjectured in [7]). For this calculus the extension with tests exists and is fully abstract for $\mathcal{D}_\infty$, this is a trivial modification of the tests of [3] (using general demonic and angelic sums). In this framework the term $\theta(\lambda xy.x + y)$ plays exactly the role of $A$ in our example with the same output.

In the end, from a unique object that is DLL, we exhibit two natural constructions, one in the semantical world, the other in the syntactical one, but they appeared do not respect full abstraction. One would say that they are not that natural and that more natural one may be found. But this would be to easy, from the state of art we don't known more natural construction. The misunderstanding comes with the concept of naturality, it seems that the syntactic idea of "convergence" does not really correspond to the equivalent in semantical word. One being a lowest fix point and the second a largest one. This difference appears when working with the demonic sum that allow to check the convergence in unbounded applicative context.

Finally we presented tests as a general tool whose importance is above the role we gave them here. This result is interesting and important as it presents tests as useful tools to verify that full abstraction fails. But it remains a negative result that does not justify alone any real interest for them. Further works will then focus on presenting positive proofs of full abstractions that are using tests. Following this way we already submitted a revisited proof of full abstraction of the Scott's $\mathcal{D}_\infty$ for the usual $\lambda$-calculus [3].

# References

[1] Gerard Boudol (1993): *The lambda-calculus with multiplicities. INRIA Research Report 2025.*

[2] P. Boudol, P.-L. Curien & C. Lavatelli (1999): *A semantics for lambda calculi with resources. Mathematical Structures in Comput. Sci. (MSCS)* Vol. 9, pp. 437–482.

[3] Flavien Breuvart (2012): *A new proof of the Hyland/Wadsworth full abstraction theorem*. Submited.

[4] Antonio Bucciarelli, Alberto Carraro, Thomas Ehrhard & Giulio Manzonetto (2011): *Full Abstraction for Resource Calculus with Tests*. In Marc Bezem, editor: *Computer Science Logic (CSL'11) - 25th International Workshop/20th Annual Conference of the EACSL*, *Leibniz International Proceedings in Informatics (LIPIcs)* 12, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, pp. 97–111.

[5] Antonio Bucciarelli, Alberto Carraro, Thomas Ehrhard & Giulio Manzonetto (2012): *Full Abstraction for Resource Lambda Calculus with Tests, throught Taylor expansion*. Accepted.

[6] Antonio Bucciarelli, Thomas Ehrhard & Giulio Manzonetto (2007): *Not Enough Points Is Enough*. In Jacques Duparc & Thomas A. Henzinger, editors: *CSL'07: Proceedings of $16^{th}$ Computer Science Logic*, *Lecture Notes in Computer Science* 4646, Springer, pp. 268–282.

[7] Antonio Bucciarelli, Thomas Ehrhard & Giulio Manzonetto (2009): *A relational model of a parallel and non-deterministic lambda-calculus*. In Sergei N. Artëmov & Anil Nerode, editors: *Logical Foundations of Computer Science, International Symposium, LFCS 2009*, *Lecture Notes in Computer Science* 5407, pp. 107–121.

[8] Daniel de Carvalho & Lorenzo Tortora de Falco (2010): *The relational model is injective for Multiplicative Exponential Linear Logic (without weakenings)*. *CoRR* abs/1002.3131.

[9] T. Ehrhard & O. Laurent (2010): *Interpreting a finitary pi-calculus in differential interaction nets*. *Inf. Comput.* 208(6), pp. 606–633.

[10] Thomas Ehrhard & Laurent Regnier (2004): *The differential lambda-calculus*. *Theoretical Computer Science*, Elsevier.

[11] A. J. Kfoury (2000): *A linearization of the Lambda-calculus and consequences*. *J. Log. Comput.* 10(3), pp. 411–436. Available at `http://dx.doi.org/10.1093/logcom/10.3.411`.

[12] Giulio Manzonetto (2009): *A general class of models of $\mathcal{H}^\star$*. In: *Mathematical Foundations of Computer Science (MFCS'09)*, *Lecture Notes in Computer Science* 5734, Springer, pp. 574–586.

[13] M. Pagani & Tranquilli P. (2009): *Parallel reduction in resource lambda-calculus*. *APLAS'09* 5904 LNCS, pp. 226–242.

[14] Michele Pagani & Simona Ronchi Della Rocca (2010): *Linearity, Non-determinism and Solvability*. *Fundamenta Informaticae* 103(1-4), pp. 173–202. Available at `http://dx.doi.org/10.3233/FI-2010-324`.

# A  The model $\mathcal{M}_\infty$

## A.1  Categorical model

The category Rel of sets and relations is known to be a model of linear logic as it is a Seely category (we are giving the interpretation, but we will let the comutative diagrams to the reader since their comutations are trivial or like):

It is monoidal with tensor functor given by $A \otimes B = A \times B$, $f \otimes g = \{((u,x),(v,y))|(u,v) \in f, (x,y) \in g\}$ and with the arbitrary unit $1 = \{*\}$. It is symetric monoidal close with $(A \multimap B) = A \times B$ if we take the evaluation $ev = \{(((a,b),a),b)|a \in A, b \in B\} \in Rel((A \multimap B) \otimes A, B)$. So it is star autonomus with $1$ as dualising object (for a trvial duality).

This give us the interpretation of multiplicatives: $A \otimes B = A \,\Im\, B = A \multimap B = A \times B$

The category is cartesian, with catesian product $\bigwedge_I A_i = \{(i,x)|i \in I, x \in s_i\}$, projections $\pi_i = \{((i,a),a)|a \in A_i\}$ and product of morphisms $\bigwedge_i f_i = \{(b,(i,a))|(b,a) \in f_i\}$. The terminal object is $\top = \varnothing$.

This give us the interpretation of additives: $\bigoplus_{i \in I} A_i = \bigwedge_{i \in I} A_i = \{(i,a)|i \in I, a \in A_i\}$

We can add a comonade $(!,d,p)$ where the functor is define by $!A = \mathcal{M}_f(A)$, $!f = \{([a_1,...,a_k],[b_1,...,b_k])|\forall i \le k, (a_i,b_i) \in f\}$, the deriliction by $d_A = \{([a],a)|a \in A\}$ and the digging by $p_A = \{(m_1+\cdots+m_k,[m_1,...,m_k])|m_1,...,m_k \in \mathcal{M}_F(a)\}$.

This give us the interpretation of exponentials: $!P = ?P = \mathbb{M}_f(P)$

This is a Seely category and a model of linear logic since the isomorphismes $1 \simeq []$ is trivial and $!A \otimes !B \simeq !(A \,\&\, B)$ is defined by $([a_1,...,a_l],[b_1,...,b_r]) \simeq [(1,a_1),...,(1,a_l),(2,b_1),...,(2,b_r)]$.

But it can even be seen as a categorical model of differential linear logic. By defining the co-dereliction natural transforamtion $\bar{d}_A = \{(a,[a])|a \in A\} \in A \to !A$, we are fixing the contraction $c_A = \{(l+r,(l,r))|l,r \in !A\}$ the co-contraction $\bar{c}_A = \{((l,r),l+r)|l,r \in !A\}$, the weakening $w_A = \{([],*)\}$ and the co-weakening $\bar{w}_A = \{(*,[])\}$. So that we can define the derivative $\partial_X = (id_{!X} \otimes d_X) \circ c_X : !X \to !X \otimes X$ and the co-derivative $\bar{\partial}_X = \bar{c}_X \circ (id_{!X} \otimes \bar{d}_D) : !X \otimes X \to !X$. This derivatives are Taylor, i.e. if two morphisms $f_1, f_2 : !A \to B$ are such that $f_1 \circ \bar{\partial}_A = f_2 \circ \bar{\partial}_A$ then $f_1 + (f_2 \circ \bar{w}_X \circ w_X) = (f_1 \circ \bar{w}_X \circ w_X) + f_2$. Finally the exponential acept anti-derivatives, since it is bi-comutative and $J_A = Id_A + \bar{\partial}_A \partial_A = Id_A$ is an isomorphism. For more detail about models of DLL see **??**.

As for every categorical model of linear logic, the exponential is a comonade and induced a coKleisly $MRel = Rel_!$ whose objects are the set and whose morphisms from $P$ to $Q$ are the relations between $!P$ and $Q$. The identities are the relations $dig_P = \{(\{x\},x)|x \in P\}$ and the composition $f \circ g = \{(X,z)|\exists(Y,z) \in f, \forall y \in Y, (X,y) \in g\}$.

## A.2  Algebraic model

In order to have an algebraic model of $\partial\lambda$-calculus we only need a reflexive object, i.e. a triplet $(M, app, abs)$ where $M$ is an object of MRel, $app : (M \to$

$(!M^\perp \multimap M) = (\mathcal{M}_f(M) \times M))$ and $aps : ((\mathcal{M}_f(M) \times M) \to M)$ such that $app \circ abs = Id$. Such an object can a priory found by taking the lower fix point of $M \mapsto M \Rightarrow M = ?M \multimap M = \mathcal{M}_f(M) \times M$. But this will just leads to the trivial empty model. We will then resolve the more complicated fix point $M \mapsto (?M)^{\&\mathbb{N}} = \mathbb{N} \times \mathcal{M}_f(M)$ where the exponent represent an infinit tensor product. The lower fix point will be called $\mathcal{M}_\infty$.

An other way to see the fixpoint is to say that $M$ have to be equal to the set of quazi everywhere empty lists of finite substets of itself. Its element are the recursively defined as being either $*$, the list of empty elements, or $a{::}\alpha$ with $a \in \mathcal{M}_f(\mathcal{M}_\infty)$ and $\alpha \in \mathcal{M}_\infty$. The coresponding $app$ and $abs$ arise imediatly from the functoriality: $app = \{(a{::}\alpha, (a, \alpha)) | (a, \alpha) \in \mathcal{M}_\infty\}$ and $abs = \{((a, \alpha), a{::}\alpha) | a, \alpha \in \mathcal{M}_\infty\}$

In order to be understandable, we are presenting the interpretation of terms via a type system with types living in $\mathcal{M}_\infty$. The usual presentation of the interpretation can be recoverd from the type system:

$\llbracket M \rrbracket^{\bar{x}} = \{(\bar{a}, \alpha) | \bar{x} : \bar{a} \vdash M : \alpha\}$

$\llbracket Q \rrbracket^{\bar{x}} = \{\bar{a} | \bar{x} : \bar{a} \vdash Q\}$

The type system is the following:

$$\frac{\Gamma \vdash \Delta}{x : [\,], \Gamma \vdash \Delta} \qquad \frac{}{x : [\alpha] \vdash x : \alpha} \qquad \frac{\Gamma \vdash M : \alpha}{\Gamma \vdash M{+}N : \alpha} \qquad \frac{\Gamma \vdash N : \alpha}{\Gamma \vdash M{+}N : \alpha} \qquad \frac{\Gamma, x : v \vdash M : \alpha}{\Gamma \vdash \lambda x.M : v{::}\alpha}$$

$$\frac{\Gamma \vdash M : w{::}\alpha \qquad \Gamma' \vdash B : w}{\Gamma + \Gamma' \vdash M\, B : \alpha} \qquad \frac{\bigwedge_{j \le n} \Gamma_j \vdash L_j : \beta_j \qquad \bigwedge_{i \ge n} \Gamma_i \vdash L : \beta_i}{(\Sigma_{r=1}^{n+m} \Gamma_r) \vdash [L_1, ..., L_n; L^!] : [\beta_1, ..., \beta_{n+m}]}$$

$$\frac{\Gamma \vdash Q}{\Gamma \vdash \bar{\tau}(Q) : *} \qquad \frac{\Gamma \vdash M : *}{\Gamma \vdash \tau[M]} \qquad \frac{\Gamma \vdash Q \qquad \Gamma' \vdash R}{\Gamma + \Gamma' \vdash Q|R} \qquad \frac{}{\vdash \epsilon}$$