Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 265 (2010) 213–230

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Categorical Models for Simply Typed Resource Calculi

Antonio Bucciarelli[3] Thomas Ehrhard[4,1]

Laboratoire PPS, Université Paris-Diderot, CNRS UMR 7126
2, place Jussieu (case 7014), 75251 Paris Cedex 05, France

Giulio Manzonetto[5,2]

Laboratoire LIPN, Université Paris Nord, CNRS UMR 7030
99, avenue Jean-Baptiste Clément 93430 Villetaneuse, France

Abstract

We introduce the notion of *differential* λ*-category* as an extension of Blute-Cockett-Seely's differential Cartesian categories. We prove that differential λ-categories can be used to model the simply typed versions of: (i) the *differential* λ*-calculus*, a λ-calculus extended with a syntactic derivative operator; (ii) the *resource calculus*, a non-lazy axiomatisation of Boudol's λ-calculus with multiplicities. Finally, we provide two concrete examples of differential λ-categories, namely, the category **MRel** of sets and relations, and the category **MFin** of finiteness spaces and finitary relations.

Keywords: differential λ-calculus, resource calculus, differential categories, relational models.

1 Introduction

The development of formal systems for proving computational properties of programs constitutes a crucial research area of contemporary computer science. Among the vast spectrum of aspects needing to be checked, one of the most important is the amount of resources a program will need during its execution. Resources to be bounded can be of very different kinds, for instance *memory space* (especially in

[1] This work is partly funded by the ANR project BLAN07-1 189926 *Curry-Howard for Concurrency* (CHOCO).

[2] This work is partly funded by Digiteo/Île-de-France project COLLODI (2009-28HD), by MIUR project CONCERTO and by NWO project CALMOC (612.000.936).

[3] Email: Antonio.Bucciarelli@pps.jussieu.fr

[4] Email: Thomas.Ehrhard@pps.jussieu.fr

[5] Email: Giulio.Manzonetto@lipn.univ-paris13.fr

1571-0661 © 2010 Elsevier B.V. Open access under CC BY-NC-ND license.

doi:10.1016/j.entcs.2010.08.013

presence of very small computing devices) or *non-replicable data* (naturally arising in the context of quantum computing).

In [4] Boudol designed the λ-*calculus with multiplicities*, a paradigmatic programming language developed for handling explicitly the problem of resource consumption within the λ-calculus. In this calculus two kinds of arguments are available: intuitionistic arguments that can be erased and copied as usual, and depletable arguments that must be used exactly once. Depletable arguments impose the presence of non-deterministic choices in the language. Suppose indeed that we have $(\lambda x.xx)L$ where L is a depletable argument: what occurrence of x should receive the only available copy of L? Another novelty with respect to λ-calculus is the fact that the arguments come in multisets called ‘bags’.

The original version of λ-calculus with multiplicities was only endowed with a (weak) head-reduction rule. This is crucial if we want to actually perform the non-deterministic choices during the reduction. In [13] Tranquilli defined the *resource calculus* which is a revisitation of Boudol’s calculus having more general forms of reductions that have been studied in [12]. In this calculus non-determinism is accounted for by means of formal sums of terms.

A formal system modeling this idea of ‘resource consumption’ was already present in Girard’s quantitative semantics [11]. This semantics establishes an analogy between linearity in the sense of computer science (programs using arguments exactly once) and algebraic linearity (commutation of sums and products with scalars), giving a new mathematically very appealing interpretation of resource consumption. Drawing on these insights, Ehrhard and Regnier designed a formal programming language, called the *differential* λ-*calculus* [6], that has a unique kind of arguments but two kinds of applications: the usual one, and a *linear application.* The result of applying linearly $\lambda x.M$ to L is the term $\lambda x.M$ where L is substituted for x in M exactly once (we keep the λx because one day we may want to substitute the other occurrences of x in M). The breakthrough in [6] is the fact that this ‘linear’ substitution operation can be seen as a formal derivative. Such a syntactic derivative operator can be fruitfully used to increase control over programs executed in environments with bounded resources (see, *e.g.*, [8,10]).

Although the differential λ-calculus is born from semantical considerations (*i.e.*, the deep analysis of denotational semantics of linear logic performed by Ehrhard and Regnier) the investigations on its denotational semantics are at the very beginning. On the one hand, it is known *in the folklore* that the Cartesian closed category (ccc, for short) of finiteness spaces and finitary relations [7] and the ccc of sets and (multi-)relations [5] are models of the simply typed differential λ-calculus, but no abstract definition of model has been provided. On the other hand, Blute, Cockett and Seely - inspired by the works on differential λ-calculus - defined the *differential categories* [2] and the *differential Cartesian categories* [3]. In these categories a derivative operator $D(-)$ on morphisms is equationally axiomatized. The authors have then proved that these categories are sound and complete to model suitable term calculi. However, it turns out that the properties of differential categories are too weak for modeling the full differential λ-calculus.

The aim of the present paper is to provide an abstract model theory (based on differential Cartesian categories) for the simply typed differential λ-calculus and resource calculus: a unifying categorical approach having all known semantics of these calculi as instances. We are confident that our work in this domain will open the way to find mathematical tools for studying quantitative properties of programs. The main object of our studies will be the differential λ-calculus, but we will draw conclusions also for the resource calculus.

In Section 3 we provide the formal definition of the differential λ-calculus, we define the type system characterizing its simply typed version and we recall some basic properties of the language.

In Section 4, starting from the work of Blute *et Al.* [3], we introduce the notion of *differential λ-category*. Basically, differential λ-categories are Cartesian differential categories which are Cartesian closed and satisfy a natural condition guaranteeing that the differential operator $D(-)$ behaves well with the Cartesian closed structure. We then prove that every differential λ-category constitutes a model of the simply typed differential λ-calculus: we first define the interpretation of (typing judgements of) differential λ-terms in the category, and secondly we prove that such an interpretation is *sound*.

In Section 5 we show that - as expected - the category **MFin** of finiteness spaces and finitary relations and the category **MRel** of sets and relations are instances of differential λ-categories.

Finally, Section 6 is devoted to recall the syntax of the simply typed resource calculus and to prove that differential λ-categories can be used to model this calculus. This is done by defining a translation map from the resource calculus to the differential λ-calculus and proving that this translation is 'faithful'. We then define the interpretation of a (typing judgement of a) resource term in a differential λ-category as the interpretation of its translation.

The investigations in this paper fit in a more ambitious program whose research lines are discussed in Section 7.

2 Preliminaries

To keep this article self-contained we summarize some definitions and results that will be used in the sequel. Our main reference for category theory is [1].

2.1 Sets and Multisets

Let S be a set. We denote by $\mathcal{P}(S)$ the powerset of S. A *multiset* m over S can be defined as an unordered list $m = [a_1, a_2, \ldots]$ with repetitions such that $a_i \in S$ for all i. A multiset m is called *finite* if it is a finite list, we denote by $[]$ the empty multiset. If m is a multiset over S, then its *support* $\mathrm{supp}(S)$ is the set of elements of S occurring in m. Given two multisets $m_1 = [a_1, a_2, \ldots]$ and $m_2 = [b_1, b_2, \ldots]$ the *multi-union* of m_1, m_2 is defined by $m_1 \uplus m_2 = [a_1, b_1, a_2, b_2, \ldots]$. We will write $\mathcal{M}_f(S)$ for the set of all finite multisets over S. Moreover, given a set $U \subseteq \mathcal{M}_f(S)$ we set $\mathrm{supp}(U) = \cup_{m \in U} \mathrm{supp}(m)$.

2.2 Cartesian (Closed) Categories

Let $\mathbf{C}$ be a small *Cartesian category* and A, B, C be arbitrary objects of $\mathbf{C}$. We write $\mathbf{C}(A, B)$ for the homset of morphisms from A to B; when there is no chance of confusion we will write $f : A \to B$ instead of $f \in \mathbf{C}(A, B)$. We usually denote by $A \times B$ the *product* of A and B, by $\pi_1 : A \times B \to A$, $\pi_2 : A \times B \to B$ the associated *projections* and, given a pair of arrows $f : C \to A$ and $g : C \to B$, by $\langle f, g\rangle : C \to A \times B$ the unique arrow such that $\pi_1 \circ \langle f, g\rangle = f$ and $\pi_2 \circ \langle f, g\rangle = g$.

If $\mathbf{C}$ is a Cartesian closed category (ccc, for short) we write $A \Rightarrow B$ for the *exponential object* and $\mathrm{ev}_{AB} : (A \Rightarrow B) \times A \to B$ for the *evaluation morphism*. Moreover, for any object C and arrow $f : C \times A \to B$, $\Lambda(f) : C \to A \Rightarrow B$ stands for the (unique) morphism such that $\mathrm{ev}_{AB} \circ (\Lambda(f) \times \mathrm{Id}_A) = f$. Finally, $\mathbb{1}$ denotes the terminal object and $!_A$ the only morphism in $\mathbf{C}(A, \mathbb{1})$.
We recall that in every ccc the following equalities hold:

$$\text{(pair)} \qquad \langle f, g\rangle \circ h = \langle f \circ h, g \circ h\rangle \qquad \Lambda(f) \circ g = \Lambda(f \circ (g \times \mathrm{Id})) \qquad \text{(Curry)}$$

$$\text{(beta-cat)} \qquad \mathrm{ev} \circ \langle \Lambda(f), g\rangle = f \circ \langle \mathrm{Id}, g\rangle \qquad \Lambda(\mathrm{ev}) = \mathrm{Id} \qquad \text{(Id-Curry)}$$

Moreover, we can define $\Lambda^- = \mathrm{ev}\circ(-\times\mathrm{Id})$. From (beta-cat), (Curry) and (Id-Curry) it follows that $\Lambda(\Lambda^-(f)) = f$ and $\Lambda^-(\Lambda(g)) = g$.

3 The Simply Typed Differential λ-calculus

In this section we recall the definition of the simply typed version of differential λ-calculus [6], together with some standard properties of the language. The set Λ^d of *differential λ-terms* and the set Λ^s of *simple terms* are defined by mutual induction as follows:

$$\Lambda^d : \quad S, T, U, V ::= \quad 0 \mid s \mid s + T \qquad\qquad \Lambda^s : \quad s, t, u, v ::= \quad x \mid \lambda x.s \mid sT \mid \mathsf{D}s \cdot t$$

The term $\mathsf{D}s \cdot t$ is the *linear application* of s to t. Intuitively, this means that s is provided with exactly one copy of t.

We consider differential λ-terms up to α-conversion, and up to associativity and commutativity of the sum. The term 0 is the neutral element of the sum, thus we add $S + 0 = S$. We write $S \equiv T$ if S and T are syntactically equal up to the above mentioned equivalences. The set $\mathrm{FV}(S)$ of *free variables* of S is defined as usual.

Notation 1 *We will often use the following abbreviations:* $\lambda x.(\sum_{i=1}^k s_i)$ *for* $\sum_{i=1}^k \lambda x.s_i$, $\mathsf{D}(\sum_{i=1}^k s_i) \cdot (\sum_{j=1}^n t_j)$ *for* $\sum_{i,j} \mathsf{D}s_i \cdot t_j$ *and* $(\sum_{i=1}^k s_i)t$ *for* $\sum_{i=1}^k s_i t$. *Notice that these are just syntactic sugar, not real terms.*

We introduce two kinds of substitutions on differential λ-terms: (i) the *capture-free substitution*, denoted by $S[T/x]$ and defined as usual; (ii) the *differential sub-*

stitution, denoted by $\frac{\partial S}{\partial x}\cdot T$ and defined by induction on S as follows[6]:

$$\frac{\partial y}{\partial x}\cdot T = \begin{cases} T & \text{if } x = y \\ 0 & \text{otherwise} \end{cases} \qquad \frac{\partial}{\partial x}(sU)\cdot T = (\frac{\partial s}{\partial x}\cdot T)U + (\mathsf{D}s\cdot(\frac{\partial U}{\partial x}\cdot T))U$$

$$\frac{\partial}{\partial x}(\lambda y.s)\cdot T = \lambda y.\frac{\partial s}{\partial x}\cdot T \qquad \frac{\partial}{\partial x}(\mathsf{D}s\cdot u)\cdot T = \mathsf{D}(\frac{\partial s}{\partial x}\cdot T)\cdot u + \mathsf{D}s\cdot(\frac{\partial u}{\partial x}\cdot T)$$

$$\frac{\partial 0}{\partial x}\cdot T = 0 \qquad \frac{\partial}{\partial x}(s+U)\cdot T = \frac{\partial s}{\partial x}\cdot T + \frac{\partial U}{\partial x}\cdot T$$

The *differential* λ*-calculus* is generated by the β-reduction $(\lambda x.s)T \to_\beta s[T/x]$ and the linear reduction $\mathsf{D}(\lambda x.s)\cdot t \to_{\beta_D} \lambda x.\frac{\partial s}{\partial x}\cdot t$. We write $\to_D$ for the contextual closure of $\to_\beta \cup \to_{\beta_D}$, and $\twoheadrightarrow_D$ (resp. $=_D$) for the transitive and reflexive (resp. transitive, reflexive and symmetric) closure of $\to_D$.

We now introduce the type system D that characterizes the *simply typed differential* λ*-calculus.*

Type System D.

$$\frac{\Gamma(x)=\sigma}{\Gamma\vdash_D x:\sigma}\ (\mathrm{Dx}) \quad \frac{\Gamma; x:\sigma\vdash_D s:\tau}{\Gamma\vdash_D \lambda x.s:\sigma\to\tau}\ (\mathrm{D}\lambda) \quad \frac{\Gamma\vdash_D s:\sigma\to\tau \quad \Gamma\vdash_D U:\sigma}{\Gamma\vdash_D sU:\tau}\ (\mathrm{D@})$$

$$\frac{\Gamma\vdash_D s:\sigma\to\tau \quad \Gamma\vdash_D t:\sigma}{\Gamma\vdash_D \mathsf{D}s\cdot t:\sigma\to\tau}\ (\mathrm{Dd}) \quad \frac{\Gamma\vdash_D s_i:\sigma \quad \text{for all } 1\le i\le k\neq 1}{\Gamma\vdash_D \sum_{i=1}^k s_i:\sigma}\ (\mathrm{D+})$$

Note that, if s has a function type $\sigma\to\tau$ and t has type σ, then $\mathsf{D}s\cdot t$ has the same function type of s, thus the derivative does not decrease the type.

Lemma 3.1 *Let $\Gamma; x:\sigma\vdash_D S:\tau$ and $\Gamma\vdash_D T:\sigma$. We have:*

(i) $\Gamma\vdash_D S[T/x]:\tau$,

(ii) $\Gamma; x:\sigma\vdash_D \frac{\partial S}{\partial x}\cdot T:\tau$,

(iii) *(Subject Reduction for D) if $T\to_D T'$ then $\Gamma\vdash_D T':\sigma$.*

Proof. *(i)* and *(ii)* follow by straightforward induction on the length of the proofs of $\Gamma\vdash_D S[T/x]:\tau$ and $\Gamma; x:\sigma\vdash_D \frac{\partial S}{\partial x}\cdot T:\tau$, respectively.

(iii) Suppose $T\equiv(\lambda x.s)U$ and $T'\equiv S[U/x]$, then it follows from (i). Suppose $T\equiv\mathsf{D}(\lambda x.s)\cdot U$ and $T'\equiv\lambda x.\frac{\partial S}{\partial x}\cdot U$, then it follows from (ii). We conclude the proof since type derivations are contextual. □

Definition 3.2 Let $\mathcal{T}$ be a collection of judgments of the shape $\Gamma\vdash S=T:\sigma$ such that $\Gamma\vdash S:\sigma$ and $\Gamma\vdash T:\sigma$. $\mathcal{T}$ is called a (typed) *differential* λ*-theory* if it is

[6] The definition of differential substitution strongly uses the abbreviations introduced above.

closed under the following rules:

$$\frac{\Gamma \vdash (\lambda x.s)T : \tau}{\Gamma \vdash (\lambda x.s)T = s[T/x] : \tau}\ (\beta) \qquad \frac{\Gamma \vdash \mathsf{D}(\lambda x.s) \cdot t : \tau}{\Gamma \vdash \mathsf{D}(\lambda x.s) \cdot t = \lambda x.\frac{\partial s}{\partial x} \cdot t : \tau}\ (\beta_D)$$

$$\frac{\Gamma; x : \sigma \vdash s = t : \tau}{\Gamma \vdash \lambda x.s = \lambda x.t : \sigma \to \tau}\ (\xi) \qquad \frac{\Gamma \vdash s = v : \sigma \to \tau \quad \Gamma \vdash T = T' : \sigma}{\Gamma \vdash sT = vT' : \tau}\ (Ap)$$

$$\frac{\Gamma \vdash s = t : \sigma \quad x : \tau \notin \Gamma}{\Gamma; x : \tau \vdash s = t : \sigma}\ (W) \qquad \frac{\Gamma \vdash s = v : \sigma \to \tau \quad \Gamma \vdash t = t' : \sigma}{\Gamma \vdash \mathsf{D}s \cdot t = \mathsf{D}v \cdot t' : \sigma \to \tau}\ (DAp)$$

plus the obvious rules for symmetry, reflexivity, transitivity and the sums.

4 A Differential Model Theory

In this section we will provide the categorical framework characterizing the models of the simply typed differential λ-calculus. The material presented in Subsection 4.1 is borrowed from [3].

4.1 Cartesian Differential Categories

A category $\mathbf{C}$ is *left-additive* whenever each homset has a structure of commutative monoid $(\mathbf{C}(A, B), +_{AB}, 0_{AB})$ and $(g + h) \circ f = (g \circ f) + (h \circ f)$ and $0 \circ f = 0$.
A morphism f in $\mathbf{C}$ is said to be *additive* if, moreover, it satisfies $f \circ (g + h) = (f \circ g) + (f \circ h)$ and $f \circ 0 = 0$.

A category is *Cartesian left-additive* if it is a left-additive category with products such that all projections and pairings of additive maps are additive. A ccc is *Cartesian closed left-additive* if it is a Cartesian left-additive category satisfying:

$$(+\text{-curry}) \quad \Lambda(f + g) = \Lambda(f) + \Lambda(g) \qquad\qquad \Lambda(0) = 0 \quad (0\text{-curry})$$

Remark 4.1 From (+-curry) it also follows that $\Lambda^-(f + g) = \Lambda^-(f) + \Lambda^-(g)$. Precomposing $\langle \mathrm{Id}, h\rangle$ on both sides we get $\mathrm{ev} \circ \langle f + g, h\rangle = \mathrm{ev} \circ \langle f, h\rangle + \mathrm{ev} \circ \langle g, h\rangle$.

Definition 4.2 A *Cartesian (closed) differential category* is a Cartesian (closed) left-additive category having an operator $D(-)$ that maps a morphism $f : A \to B$ into a morphism $D(f) : A \times A \to B$ and satisfies the following axioms:

D1. $D(f + g) = D(f) + D(g)$ and $D(0) = 0$

D2. $D(f) \circ \langle h + k, v\rangle = D(f) \circ \langle h, v\rangle + D(f) \circ \langle k, v\rangle$ and $D(f) \circ \langle 0, v\rangle = 0$

D3. $D(\mathrm{Id}) = \pi_1$, $D(\pi_1) = \pi_1 \circ \pi_1$ and $D(\pi_2) = \pi_2 \circ \pi_1$

D4. $D(\langle f, g\rangle) = \langle D(f), D(g)\rangle$

D5. $D(f \circ g) = D(f) \circ \langle D(g), g \circ \pi_2\rangle$

D6. $D(D(f)) \circ \langle\langle g, 0\rangle, \langle h, k\rangle\rangle = D(f) \circ \langle g, k\rangle$

D7. $D(D(f)) \circ \langle\langle 0, h\rangle, \langle g, k\rangle\rangle = D(D(f)) \circ \langle\langle 0, g\rangle, \langle h, k\rangle\rangle$

We try to provide some intuitions on these axioms. (D1) says that the operator $D(-)$ is linear; (D2) says that $D(-)$ is additive in its first coordinate; (D3) and

(D4) ask that $D(-)$ behaves coherently with the product structure; (D5) is the usual chain rule; (D6) requires that $D(f)$ is linear in its first component. (D7) states the independence of order of "partial differentiation".

Remark 4.3 In a Cartesian differential category we obtain partial derivatives from the full ones by "zeroing out" the components on which the differentiation is not required. *E.g.*, suppose that we want to define the partial derivative $D_1(f)$ of $f : C \times A \to B$ on its 1st component; then, it is sufficient to set $D_1(f) = D(f) \circ (\langle \mathrm{Id}_C, 0_A\rangle \times \mathrm{Id}_{C\times A}) \colon C \times (C\times A) \to B$. Similarly, we define $D_2(f) \colon A \times (C\times A) \to B$, the partial derivative of f on its 2nd component.

This remark follows since every differential $D(f)$ can be reconstructed from its partial derivatives as follows:

$$\begin{aligned} D(f) &= D(f) \circ \langle\langle \pi_1 \circ \pi_1, \pi_2 \circ \pi_1\rangle, \pi_2\rangle \\ &= D(f) \circ \langle\langle \pi_1 \circ \pi_1, 0\rangle, \pi_2\rangle + D(f) \circ \langle\langle 0, \pi_2 \circ \pi_1\rangle, \pi_2\rangle \\ &= D(f) \circ (\langle \mathrm{Id}, 0\rangle \times \mathrm{Id}) \circ (\pi_1 \times \mathrm{Id}) + D(f) \circ (\langle 0, \mathrm{Id}\rangle \times \mathrm{Id}) \circ (\pi_2 \times \mathrm{Id}) \\ &= D_1(f) \circ (\pi_1 \times \mathrm{Id}) + D_2(f) \circ (\pi_2 \times \mathrm{Id}). \end{aligned}$$

4.2 Differential λ-Categories

Cartesian closed differential categories are not enough to interpret the differential λ-calculus, since the differential operator does not behave automatically well with respect to the Cartesian close structure. For this reason, we now introduce the notion of *differential λ-category.*

Definition 4.4 A *differential λ-category* is a Cartesian closed differential category such that, for all $f : C \times A \to B$:

$$\textit{(D-curry)}\ D(\Lambda(f)) = \Lambda(D(f) \circ \langle \pi_1 \times 0_A, \pi_2 \times \mathrm{Id}_A\rangle)$$

Intuitively, (D-curry) requires that in a λ-category we have two equivalent ways to derivate $f : C\times A \to B$ in its 1st component: we can use the trick of Remark 4.3, or we can 'hide' the component A by currying f and then derive $\Lambda(f)$.

Lemma 4.5 *In every differential λ-category the following axiom holds (for all $h : C \to A \Rightarrow B$ and $g : C \to A$):*

$$\text{(D-eval)}\ D(\mathrm{ev} \circ \langle h, g\rangle) = \mathrm{ev} \circ \langle D(h), g \circ \pi_2\rangle + D(\Lambda^-(h)) \circ \langle\langle 0_C, D(g)\rangle, \langle \pi_2, g \circ \pi_2\rangle\rangle.$$

Proof. Let $h' = \Lambda^{-}(h) : C \times A \to B$.

$$\begin{array}{ll}
D(\mathrm{ev} \circ \langle h, g\rangle) = & \text{by def. of } h' \\
D(\mathrm{ev} \circ \langle \Lambda(h'), g\rangle) = & \text{by (beta-cat)} \\
D(h' \circ \langle \mathrm{Id}_C, g\rangle) = & \text{by (D5)} \\
D(h') \circ \langle D(\langle \mathrm{Id}_C, g\rangle), \langle \mathrm{Id}_C, g\rangle \circ \pi_2\rangle = & \text{by (D4) and (D3)} \\
D(h') \circ \langle\langle \pi_1, D(g)\rangle, \langle \pi_2, g \circ \pi_2\rangle\rangle = & \text{since pairing is additive} \\
D(h') \circ \langle\langle \pi_1, 0_A\rangle + \langle 0_C, D(g)\rangle, \langle \pi_2, g \circ \pi_2\rangle\rangle = & \text{by (D2)} \\
D(h') \circ \langle\langle \pi_1, 0_A\rangle, \langle \pi_2, g \circ \pi_2\rangle\rangle & \\
\quad + D(h') \circ \langle\langle 0_C, D(g)\rangle, \langle \pi_2, g \circ \pi_2\rangle\rangle = & \text{by (proj)} \\
D(h') \circ \langle \pi_1 \times 0_A, \pi_2 \times \mathrm{Id}_A\rangle \circ \langle \mathrm{Id}_{C \times C}, g \circ \pi_2\rangle & \\
\quad + D(h') \circ \langle\langle 0_C, D(g)\rangle, \langle \pi_2, g \circ \pi_2\rangle\rangle = & \text{by (beta-cat)} \\
\mathrm{ev} \circ \langle \Lambda(D(h') \circ \langle \pi_1 \times 0_A, \pi_2 \times \mathrm{Id}_A\rangle), g \circ \pi_2\rangle & \\
\quad + D(h') \circ \langle\langle 0_C, D(g)\rangle, \langle \pi_2, g \circ \pi_2\rangle\rangle = & \text{by (D-curry)} \\
\mathrm{ev} \circ \langle D(\Lambda(h')), g \circ \pi_2\rangle & \\
\quad + D(\Lambda^{-}(\Lambda(h'))) \circ \langle\langle 0_C, D(g)\rangle, \langle \pi_2, g \circ \pi_2\rangle\rangle = & \text{by def. of } h' \\
\mathrm{ev} \circ \langle D(h), g \circ \pi_2\rangle + D(\Lambda^{-}(h)) \circ \langle\langle 0_C, D(g)\rangle, \langle \pi_2, g \circ \pi_2\rangle\rangle &
\end{array}$$

□

(D-eval) can be seen as a chain rule for denotations of terms (cf. Lemma 4.10(i)).

Definition 4.6 In differential λ-categories we are able to define a binary operator $\star$ on morphisms, that can be seen as the semantic counterpart of differential substitution:

$$\frac{f : C \times A \to B \quad g : C \to A}{f \star g : C \times A \to B}\ (\star)$$

This operator is defined by $f \star g = D(f) \circ \langle\langle 0_C^{C \times A}, g \circ \pi_1\rangle, \mathrm{Id}_{C \times A}\rangle$.

Intuitively, the morphism $f \star g$ is obtained by force-feeding the 2nd argument A of f with *one copy* of the result of g. The type is not modified because $f \star g$ may still depend on A.

Definition 4.7 Let $\mathrm{sw}_{ABC} = \langle\langle \pi_1 \circ \pi_1, \pi_2\rangle, \pi_2 \circ \pi_1\rangle : (A \times B) \times C \to (A \times C) \times B$.

Remark 4.8 $\mathrm{sw} \circ \mathrm{sw} = \mathrm{Id}_{(A \times B) \times C}$, $\mathrm{sw} \circ \langle\langle f, g\rangle, h\rangle = \langle\langle f, h\rangle, g\rangle$ and $D(\mathrm{sw}) = \mathrm{sw} \circ \pi_1$.

The following two technical lemmas will be used in Subsection 4.3 to show that differential λ-categories are models of the simply typed differential λ-calculus.

Lemma 4.9 *Let $f : (C \times A) \times D \to B$ and $g : C \to A$, $h : C \to B'$. Then:*

(i) $\pi_2 \star g = g \circ \pi_1$

(ii) $(h \circ \pi_1) \star g = 0$

(iii) $\Lambda(f) \star g = \Lambda(((f \circ \mathrm{sw}) \star (g \circ \pi_1)) \circ \mathrm{sw})$.

Proof. (Outline) (i) follows by applying *D3*. (ii) follows by applying *D2*, *D3* and *D5*. (iii) follows by *(Curry)*, *(D-curry)* and *D2, D3, D5.* □

Lemma 4.10 *Let* $f : C \times A \to [D \Rightarrow B]$ *and* $g : C \to A$, $h : C \times A \to D$. *Then:*

(i) $(\mathrm{ev} \circ \langle f, h \rangle) \star g = \mathrm{ev} \circ \langle f \star g + \Lambda(\Lambda^-(f) \star (h \star g)), h \rangle$

(ii) $\Lambda(\Lambda^-(f) \star h) \star g = \Lambda(\Lambda^-(f \star g) \star h) + \Lambda(\Lambda^-(f) \star (h \star g))$

(iii) $\Lambda(\Lambda^-(f) \star h) \circ \langle \mathrm{Id}_C, g \rangle = \Lambda(\Lambda^-(f \circ \langle \mathrm{Id}_C, g \rangle) \star (h \circ \langle \mathrm{Id}_C, g \rangle))$

Proof. (Outline) *(i)* follows by applying *(D-eval)* and *(beta-cat)*.

(ii) This equation can be simplified by using the axioms of Cartesian closed left-additive categories. Indeed, the right side can be written as $\Lambda((\Lambda^-(f \star g) \star h) + \Lambda^-(f) \star (h \star g))$. By taking a morphism f' such that $f = \Lambda(f')$ and by applying Lemma 4.9(iii) the item (ii) becomes equivalent to $((f' \star h) \circ \mathrm{sw}) \star (g \circ \pi_1) \circ \mathrm{sw} = (((f' \circ \mathrm{sw}) \star (g \circ \pi_1)) \circ \mathrm{sw}) \star h + f' \star (h \star g)$. This follows by *(Curry)* and *D2-7.*

(iii) follows by *(Curry)* and *D2-5.* □

4.3 Interpreting the Differential Lambda Calculus

In this section we define the interpretation of the simply typed differential λ-calculus in a fixed differential λ-category $\mathbf{C}$.

Types are interpreted as follows:

- $|\alpha| = A$, for some object A,
- $|\sigma \to \tau| = |\sigma| \Rightarrow |\tau|$.

Contexts are interpreted as usual:

- $|\emptyset| = \mathbb{1}$,
- $|\Gamma; x : \sigma| = |\Gamma| \times |\sigma|$.

The interpretation of a judgement $\Gamma \vdash S : \sigma$ will be a morphism from $|\Gamma|$ to $|\sigma|$ denoted by $|S^\sigma|_\Gamma$ and defined inductively as follows.

Interpretation of judgements

- $|x^\sigma|_{\Gamma;x:\sigma} = \pi_2 : |\Gamma| \times |\sigma| \to |\sigma|$,
- $|y^\tau|_{\Gamma;x:\sigma} = |y^\tau|_\Gamma \circ \pi_1 : |\Gamma| \times |\sigma| \to |\tau|$ for $x \neq y$,
- $|(sU)^\tau|_\Gamma = \mathrm{ev} \circ \langle |s^{\sigma \to \tau}|_\Gamma, |U^\sigma|_\Gamma \rangle : |\Gamma| \to |\tau|$,
- $|(\lambda x.s)^{\sigma \to \tau}|_\Gamma = \Lambda(|s^\tau|_{\Gamma;x:\sigma}) : |\Gamma| \to |\sigma| \Rightarrow |\tau|$,
- $|(\mathsf{D}\, s \cdot t)^{\sigma \to \tau}|_\Gamma = \Lambda(\Lambda^-(|s^{\sigma \to \tau}|_\Gamma) \star |t^\sigma|_\Gamma) : |\Gamma| \to |\sigma| \Rightarrow |\tau|$,
- $|0^\sigma|_\Gamma = 0 : |\Gamma| \to |\sigma|$,
- $|(s + S)^\sigma|_\Gamma = |s^\sigma|_\Gamma + |S^\sigma|_\Gamma : |\Gamma| \to |\sigma|$.

We will sometimes omit the superscript σ in $|S^\sigma|_\Gamma$, when there is no chance of confusion. Given a differential λ-category $\mathbf{C}$ we can define the *theory of* $\mathbf{C}$ by:

$$\mathrm{Th}(\mathbf{C}) = \{\Gamma \vdash S = T : \sigma \mid \Gamma \vdash_D S : \sigma, \quad \Gamma \vdash_D T : \sigma, \quad |S^\sigma|_\Gamma = |T^\sigma|_\Gamma\}.$$

At the end of this section we will prove that the interpretation $|-|$ is *sound* for the differential λ-calculus, *i.e.,* that $\mathrm{Th}(\mathbf{C})$ is a differential λ-theory. In order to prove this result, we need first some technical results.

The following lemma is easy to prove.

Lemma 4.11 *Let $S \in \Lambda^d$, then:*

(i) *If $\Gamma \vdash_D S : \tau$ and $x \notin \mathrm{FV}(S)$ then $|S^\tau|_{\Gamma;x:\sigma} = |S^\tau|_\Gamma \circ \pi_1$,*

(ii) $|S|_{\Gamma;x:\sigma;y:\tau} = |S|_{\Gamma;y:\tau;x:\sigma} \circ \mathrm{sw}$.

Theorem 4.12 *(Substitutions) Let $\Gamma; x : \sigma \vdash_D S : \tau$ and $\Gamma \vdash_D T : \sigma$, then:*

(i) $|(S[T/x])^\tau|_\Gamma = |S^\tau|_{\Gamma;x:\sigma} \circ \langle \mathrm{Id}_{|\Gamma|}, |T^\sigma|_\Gamma\rangle$,

(ii) $|(\frac{\partial S}{\partial x} \cdot T)^\tau|_{\Gamma;x:\sigma} = |S^\tau|_{\Gamma;x:\sigma} \star |T^\sigma|_\Gamma$.

Proof. *(i)* By induction on S. The only interesting case is $S \equiv \mathsf{D}s \cdot u$.

By def. of substitution we have $|(\mathsf{D}s \cdot u)[T/x]|_\Gamma = |\mathsf{D}s[T/x] \cdot u[T/x]|_\Gamma$. By def. of $|-|$ this is equal to $\Lambda(\Lambda^-(|s[T/x]|_\Gamma) \star |u[T/x]|_\Gamma)$. By induction hypothesis (IH) we get $\Lambda(\Lambda^-(|s|_{\Gamma;x:\sigma} \circ \langle \mathrm{Id}, |T|_\Gamma\rangle) \star (|u|_{\Gamma;x:\sigma} \circ \langle \mathrm{Id}, |T|_\Gamma\rangle))$. By applying Lemma 4.10(*iii*) this is equal to $\Lambda(\Lambda^-(|s|_{\Gamma;x:\sigma}) \star |u|_{\Gamma;x:\sigma}) \circ \langle \mathrm{Id}, |T|_\Gamma\rangle = |\mathsf{D}s \cdot u|_{\Gamma;x:\sigma} \circ \langle \mathrm{Id}, |T|_\Gamma\rangle$.

(ii) By structural induction on S.

- case $S \equiv x$. Then $|\frac{\partial x}{\partial x} \cdot T|_{\Gamma;x:\sigma} = |T|_{\Gamma;x:\sigma} = |T|_\Gamma \circ \pi_1 = \pi_2 \star |T|_\Gamma = |x|_{\Gamma;x:\sigma} \star |T|_\Gamma$ by Lemma 4.9(*i*).
- case $S \equiv z \neq x$. Then $|\frac{\partial z}{\partial x} \cdot T|_{\Gamma;x:\sigma} = |0|_{\Gamma;x:\sigma} = 0$. By Lemma 4.9(*ii*) we have $0 = (|z|_\Gamma \circ \pi_1) \star |T|_\Gamma = |z|_{\Gamma;x:\sigma} \star |T|_\Gamma$.
- case $S \equiv \lambda z.v$. By def. of differential substitution we have that $|\frac{\partial(\lambda z.v)}{\partial x} \cdot T|_{\Gamma;x:\sigma} = |\lambda z.\frac{\partial v}{\partial x} \cdot T|_{\Gamma;x:\sigma} = \Lambda(|\frac{\partial v}{\partial x} \cdot T|_{\Gamma;x:\sigma;z:\gamma})$. Applying Lemma 4.11(*ii*), this is equal to $\Lambda(|\frac{\partial v}{\partial x} \cdot T|_{\Gamma;z:\gamma;x:\sigma} \circ \mathrm{sw})$. By IH we obtain $\Lambda((|v|_{\Gamma;z:\gamma;x:\sigma} \star |T|_{\Gamma;z:\gamma}) \circ \mathrm{sw})$. Supposing wlog that $z \notin \mathrm{FV}(T)$ we have $|T|_{\Gamma;z:\gamma} = |T|_\Gamma \circ \pi_1$ (by Lemma 4.11(*i*)). Thus, by Lemma 4.9(*iii*), we have that $\Lambda((|v|_{\Gamma;z:\gamma;x:\sigma} \star (|T|_\Gamma \circ \pi_1)) \circ \mathrm{sw}) = \Lambda(|v|_{\Gamma;z:\gamma;x:\sigma} \circ \mathrm{sw}) \star |T|_\Gamma$, which is $\Lambda(|v|_{\Gamma;x:\sigma;z:\gamma}) \star |T|_\Gamma$ by Lemma 4.11(*ii*). We conclude since $\Lambda(|v|_{\Gamma;x:\sigma;z:\gamma}) \star |T|_\Gamma = |\lambda z.v|_{\Gamma;x:\sigma} \star |T|_\Gamma$.
- case $S \equiv sU$. By def. of differential substitution we have that $|\frac{\partial(sU)}{\partial x} \cdot T|_{\Gamma;x:\sigma} = |(\frac{\partial s}{\partial x} \cdot T)U|_{\Gamma;x:\sigma} + |(\mathsf{D}s \cdot (\frac{\partial U}{\partial x} \cdot T))U|_{\Gamma;x:\sigma}$. Let us consider the two addenda componentwise. We have $|(\frac{\partial s}{\partial x} \cdot T)U|_{\Gamma;x:\sigma} = \mathrm{ev} \circ \langle |\frac{\partial s}{\partial x} \cdot T|_{\Gamma;x:\sigma}, |U|_{\Gamma;x:\sigma}\rangle$ which is equal, by IH, to $\mathrm{ev} \circ \langle |s|_{\Gamma;x:\sigma} \star |T|_\Gamma, |U|_{\Gamma;x:\sigma}\rangle$. On the other side we have: $|(\mathsf{D}s \cdot (\frac{\partial U}{\partial x} \cdot T))U|_{\Gamma;x:\sigma} = \mathrm{ev} \circ \langle \Lambda(\Lambda^-(|s|_{\Gamma;x:\sigma}) \star |\frac{\partial U}{\partial x} \cdot T|_{\Gamma;x:\sigma}), |U|_{\Gamma;x:\sigma}\rangle$, by IH this is equal to $\mathrm{ev} \circ \langle \Lambda(\Lambda^-(|s|_{\Gamma;x:\sigma}) \star (|U|_{\Gamma;x:\sigma} \star |T|_\Gamma)), |T|_{\Gamma;x:\sigma}\rangle$. By applying Remark 4.1 we can rewrite the sum of this two addenda as follows: $\mathrm{ev} \circ \langle |s|_{\Gamma;x:\sigma} \star |T|_\Gamma + \Lambda(\Lambda^-(|s|_{\Gamma;x:\sigma}) \star (|U|_{\Gamma;x:\sigma} \star |T|_\Gamma)), |U|_{\Gamma;x:\sigma}\rangle$. By Lemma 4.10(*i*) this is $(\mathrm{ev} \circ \langle |s|_{\Gamma;x:\sigma}, |U|_{\Gamma;x:\sigma}\rangle) \star |T|_\Gamma = |sU|_{\Gamma;x:\sigma} \star |T|_\Gamma$.

- case $S \equiv \mathsf{D}v \cdot u$. By definition, we have that $|\frac{\partial}{\partial x}(\mathsf{D}v \cdot u) \cdot T|_{\Gamma;x:\sigma} = |\mathsf{D}(\frac{\partial v}{\partial x} \cdot T) \cdot u|_{\Gamma;x:\sigma} + |\mathsf{D}v \cdot (\frac{\partial u}{\partial x} \cdot T)|_{\Gamma;x:\sigma}$. Consider the two addenda separately. $|\mathsf{D}(\frac{\partial v}{\partial x} \cdot T) \cdot u|_{\Gamma;x:\sigma} = \Lambda(\Lambda^-(|\frac{\partial v}{\partial x} \cdot T|_{\Gamma;x:\sigma}) \star |u|_{\Gamma;x:\sigma})$. By IH this is equal to $\Lambda(\Lambda^-(|v|_{\Gamma;x:\sigma} \star |T|_\Gamma) \star |u|_{\Gamma;x:\sigma})$. On the other hand, we have that $|\mathsf{D}v \cdot (\frac{\partial u}{\partial x} \cdot T)|_{\Gamma;x:\sigma} = \Lambda(\Lambda^-(|v|_{\Gamma;x:\sigma}) \star |\frac{\partial u}{\partial x} \cdot T|_{\Gamma;x:\sigma})$. By IH this is $\Lambda(\Lambda^-(|v|_{\Gamma;x:\sigma}) \star (|u|_{\Gamma;x:\sigma} \star |T|_\Gamma))$. By applying Lemma 4.10(*ii*) to the sum of the two morphisms, we obtain $\Lambda(\Lambda^-(|v|_{\Gamma;x:\sigma}) \star |u|_{\Gamma;x:\sigma}) \star |T|_\Gamma$ which is equal to $|\mathsf{D}v \cdot u|_{\Gamma;x:\sigma} \star |T|_\Gamma$.
- all other cases (*i.e.*, $S \equiv 0$ and $S \equiv s + U$) are straightforward.

□

We are now able to prove the main theorem of this section.

Theorem 4.13 *Let* **C** *be a differential* λ*-category. Then* Th(**C**) *is a differential* λ*-theory.*

Proof. We have to check that Th(**C**) is closed under the rules presented in Def. 3.2.

(β) Let $|(\lambda x.s)T|_\Gamma = \mathrm{ev} \circ \langle \Lambda(|s|_{\Gamma;x:\sigma}), |T|_\Gamma \rangle$. By the Theorem 4.12(i), we have $|s[T/x]|_\Gamma = |s|_{\Gamma;x:\sigma} \circ \langle \mathrm{Id}, |T|_\Gamma \rangle$ and, by (beta-cat), $|s|_{\Gamma;x:\sigma} \circ \langle \mathrm{Id}, |T|_\Gamma \rangle = \mathrm{ev} \circ \langle \Lambda(|s|_{\Gamma;x:\sigma}), |T|_\Gamma \rangle$.

(β_D) Let $|\mathsf{D}(\lambda x.s) \cdot t|_\Gamma = \Lambda(\Lambda^-(\Lambda(|s|_{\Gamma;x:\sigma})) \star |t|_\Gamma) = \Lambda(|s|_{\Gamma;x:\sigma} \star |t|_\Gamma)$. By applying Theorem 4.12(ii), this is equal to $\Lambda(|\frac{\partial s}{\partial x} \cdot t|_{\Gamma;x:\sigma}) = |\lambda x.\frac{\partial s}{\partial x} \cdot t|_\Gamma$.

For the weakening (W) we use Lemma 4.11(i). Symmetry, reflexivity and transitivity hold since Th(**C**) is an equivalence. The rule for sums follows from the definition of the interpretation of sums. Finally, (ξ), (Ap) and (DAp) follow by definition of the interpretation of abstraction, application and linear application (respectively). □

5 Examples of Differential λ-Categories

In this section we provide two examples of differential λ-category: (i) **MRel**, which is the co-Kleisli category of the functor $\mathcal{M}_f(-)$ over the $\star$-autonomous category **Rel** of sets and relations [11,5]; (ii) the category **MFin**, which is the co-Kleisli of the functor $\mathcal{M}_f(-)$ over the $\star$-autonomous category of finiteness spaces and finitary relations [7].

5.1 Relational Semantics

We provide here a direct definition of the category **MRel**:

- The objects of **MRel** are all the sets.
- A morphism from A to B is a relation from $\mathcal{M}_f(A)$ to B, in other words, $\mathbf{MRel}(A, B) = \mathcal{P}(\mathcal{M}_f(A) \times B)$.
- The identity of A is the relation $\mathrm{Id}_A = \{([a], a) \mid a \in A\} \in \mathbf{MRel}(A, A)$.
- The composition of $s \in \mathbf{MRel}(A, B)$ and $t \in \mathbf{MRel}(B, C)$ is defined by:

$$t \circ s = \{(m, c) \mid \exists (m_1, b_1), \ldots, (m_k, b_k) \in s \text{ such that } m = m_1 \uplus \ldots \uplus m_k \text{ and } ([b_1, \ldots, b_k], c) \in t\}.$$

Theorem 5.1 *The category* **MRel** *is a differential λ-category.*

Proof. (Outline) The fact that **MRel** is a ccc is proved, for instance, in [5]. The categorical product[7] (&) is the disjoint union ($\dot\cup$) and $\emptyset$ is the terminal object. Given $s_i \in \mathbf{MRel}(A, B_i)$ (for $i = 1, 2$), the corresponding morphism $\langle s_1, s_2 \rangle \in \mathbf{MRel}(A, B_1 \& B_2)$ is given by: $\langle s_1, s_2 \rangle = \{(m, (i, a)) \mid (m, a) \in s_i, \text{ for } i = 1, 2\}$.

We consider the canonical bijection between $\mathcal{M}_f(A_1) \times \mathcal{M}_f(A_2)$ and $\mathcal{M}_f(A_1 \& A_2)$ as an equality, hence we will still denote by (m_1, m_2) the corresponding element of $\mathcal{M}_f(A_1 \& A_2)$. Given two sets A, B we have $A \Rightarrow B = \mathcal{M}_f(A) \times B$ and:

$$\mathrm{ev}_{AB} = \{(([(m, b)], m), b) \mid m \in \mathcal{M}_f(A) \text{ and } b \in B\} \in \mathbf{MRel}((A \Rightarrow B) \& A, B)\,.$$

Given any set A and any morphism $s \in \mathbf{MRel}(A \& B, C)$, there is exactly one morphism $\Lambda(s) = \{(p, (m, b)) \mid ((p, m), b) \in s\} \in \mathbf{MRel}(A, B \Rightarrow C)$ such that $\mathrm{ev}_{BC} \circ \langle \Lambda(s), \mathrm{Id}_B \rangle = s$.

MRel is a Cartesian closed left-additive category since the homsets can be endowed with the following additive structure $(\mathbf{MRel}(A, B), \cup, \emptyset)$.

Finally, given $f \in \mathbf{MRel}(A, B)$ we can define its derivative as follows:

$$D(f) = \{(([a], m), b) \mid (m \uplus [a], b) \in f\} \in \mathbf{MRel}(A \& A, B).$$

It is not difficult to check that $D(-)$ satisfies (D1-7) and (D-curry). □

Thus, the operation $\star$ can be directly defined in **MRel** as follows:

$$f \star g = \{((m_1 \uplus m_2, m), b) \mid (m_1, a) \in g,\ ((m_2, m \uplus [a]), b) \in f\} : C \& A \to B.$$

5.2 Finiteness Spaces Semantics

We provide here a brief account of [7], in order to give a direct presentation of the Cartesian closed category of finiteness spaces and finitary relations. All the categorical constructions are tightly related to the corresponding ones in **MRel**.

Let X be a set and $a, b \subseteq X$. We say that a, b are *orthogonal*, written $a \perp b$, if $a \cap b$ is a finite set. Given $\mathcal{F} \subseteq \mathcal{P}(X)$, we set $\mathcal{F}^\perp = \{\, b \in \mathcal{P}(X) \mid \forall a \in \mathcal{F}\ a \perp b \,\}$.

Definition 5.2 A *finiteness space* is a pair $\mathcal{X} = (X, \mathsf{F}(\mathcal{X}))$ where X is a countable set and $\mathsf{F}(\mathcal{X})$ is a subset of $\mathcal{P}(X)$ satisfying $\mathsf{F}(\mathcal{X})^{\perp\perp} = \mathsf{F}(\mathcal{X})$. The elements of $\mathsf{F}(\mathcal{X})$ are called the *finitary sets* of $\mathcal{X}$.

Given a finiteness space $\mathcal{X} = (X, \mathsf{F}(\mathcal{X}))$, define $!\mathcal{X} = (\mathcal{M}_f(X), \mathsf{F}(!\mathcal{X}))$ where $\mathsf{F}(!\mathcal{X}) = \{U \subseteq \mathcal{M}_f(X) \mid \mathrm{supp}(U) \in \mathsf{F}(X)\}$.

[7] In this section the symbol × is kept to denote the usual set-theoretical Cartesian product.

Definition 5.3 A *finitary relation* from $\mathcal{X}$ to $\mathcal{Y}$ is a relation $R \subseteq X \times Y$ such that:

- for all $a \in \mathsf{F}(\mathcal{X})$, $R(a) = \{\beta \in Y \mid \exists \alpha \in a\ (\alpha, \beta) \in R\} \in \mathsf{F}(\mathcal{Y})$, and
- for all $\beta \in Y$, $R^{\perp}(\beta) = \{\alpha \in X \mid (\alpha, \beta) \in R\} \in \mathsf{F}(\mathcal{X})^{\perp}$.

The category **MFin** can be directly defined as follows:

- The objects are the finiteness spaces.
- A morphism from $\mathcal{X}$ to $\mathcal{Y}$ is a finitary relation from $!\mathcal{X}$ to $\mathcal{Y}$.
- Identities and composition are defined as in **MRel**.

Theorem 5.4 *The category* **MFin** *is a differential* λ*-category.*

Proof. (Outline) The categorical product $\mathcal{X} \& \mathcal{Y}$ of the finiteness spaces $\mathcal{X}$ and $\mathcal{Y}$ is $(X \& Y, \mathsf{F}(\mathcal{X} \& \mathcal{Y}))$, where $X \& Y = X \dot{\cup} Y$ and $\mathsf{F}(X \& Y) = \{a \dot{\cup} a' \mid a \in \mathsf{F}(\mathcal{X}),\ a' \in \mathsf{F}(\mathcal{Y})\}$. Projections and pairing are precisely like in **MRel**.

The exponential object $\mathcal{X} \Rightarrow \mathcal{Y} = (\mathcal{M}_f(X) \times Y, \mathsf{F}(\mathcal{X} \Rightarrow \mathcal{Y}))$; we refer to [7] for the precise definition of $\mathsf{F}(\mathcal{X} \Rightarrow \mathcal{Y})$. The additive structure on homsets, $\Lambda(-)$, ev and $D(-)$ are defined as in **MRel**. To conclude the proof it is sufficient to check that ev is a finitary morphism and that for every finitary f of the correct type we have that $\Lambda(f)$ and $D(f)$ are finitary.

We explicit here the case of $D(f)$. We then want to show that for every $f \in \mathbf{MFin}(\mathcal{X}, \mathcal{Y})$ we have $D(f) = \{(([\alpha], m), \beta) \mid (m \uplus [\alpha], \beta) \in f\} \in \mathbf{MFin}(\mathcal{X} \& \mathcal{X}, \mathcal{Y})$.

As a preliminary remark we note that, for every $\mathcal{X}$, $\mathsf{F}(\mathcal{X})$ and $\mathsf{F}(\mathcal{X})^{\perp}$ contain all finite subsets of X and are closed with respect to arbitrary intersections, finite unions and subsets. As in the preceding subsection we will use implicitly the isomorphism between $\mathcal{M}_f(X \dot{\cup} Y)$ and $\mathcal{M}_f(X) \times \mathcal{M}_f(Y)$.

We start by proving that, given a finitary set $U \in \mathsf{F}(!(\mathcal{X} \ \& \ \mathcal{X}))$, we have $(D(f))(U) \in \mathsf{F}(\mathcal{Y})$. By definition of $\mathsf{F}(!(\mathcal{X} \ \& \ \mathcal{X}))$, we have that $\pi_1(\mathrm{supp}(U))$, $\pi_2(\mathrm{supp}(U)) \in \mathsf{F}(\mathcal{X})$, and hence $\pi_1(\mathrm{supp}(U)) \cup \pi_2(\mathrm{supp}(U)) \in \mathsf{F}(\mathcal{X})$. Now, defining $U' = \{m_1 \uplus m_2 \mid (m_1, m_2) \in U\}$, we have that $\pi_1(\mathrm{supp}(U)) \cup \pi_2(\mathrm{supp}(U)) = \mathrm{supp}(U')$, hence $U' \in F(!\mathcal{X})$, and $f(U') \in \mathsf{F}(\mathcal{Y})$, f being finitary. Since $(D(f))(U) \subseteq f(U')$, we have finished.

It remains to show that, for all $\beta \in Y$, we have $(D(f))^{\perp}(\beta) \in \mathsf{F}(!(\mathcal{X} \& \mathcal{X}))^{\perp}$. Given $U \in \mathsf{F}(!(\mathcal{X} \& \mathcal{X}))$, we define $U' = \{m_1 \uplus m_2 \mid (m_1, m_2) \in U\} \in \mathsf{F}(\mathcal{X})$ as above. We know that $f^{\perp}(\beta) \cap U'$ is finite, and we have to show that $(D(f))^{\perp}(\beta) \cap U$ is finite. It is easy to see that $(D(f))^{\perp}(\beta) \cap U = \{([\alpha], m) \mid m \uplus [\alpha] \in f^{\perp}(\beta) \cap U'\}$, m being a finite multiset. Hence $D(f)^{\perp}(\beta) \cap U$ is finite, $f^{\perp}(\beta) \cap U'$ being itself finite.□

Remark 5.5 We may have presented another differential λ-category of finiteness spaces, which is obtained from **MFin** by considering as objects the R-vector spaces (for some field R) associated to the finiteness spaces, and as morphisms the continuous and linear maps between R-vector spaces [7, pag. 20]. However, this category is unnecessarily complicated for our purposes.

6 The Resource Calculus

6.1 Its Syntax

In this section we present the resource calculus [4] (using the formalization given in [12]) and we show that every differential λ-category is also a model of its simply typed version. In this calculus there are three syntactical sorts: *resource λ-terms* (Λ^r) that are in functional position; *bags* (Λ^b) that are in argument position and represent multisets of resources, and *sums* that represent the possible results of a computation. A *resource* ($\Lambda^{(!)}$) can be linear or intuitionistic, in the latter case it is written with a ! apex. An *expression* ($\Lambda^{(b)}$) is either a term or a bag.
Formally, we have:

$$\begin{array}{llll} \Lambda^r : & M, N, L & ::= x \mid \lambda x.M \mid MP & \text{resource } \lambda\text{-terms} \\ \Lambda^{(!)} : & M^{(!)}, N^{(!)} & ::= M \mid M^! & \text{resources} \\ \Lambda^b : & P, Q, R & ::= [M_1^{(!)}, \dots, M_n^{(!)}] & \text{bags} \\ \Lambda^{(b)} : & A, B & ::= M \mid P & \text{expressions} \end{array}$$

Concerning sums, $\mathcal{N}\langle\Lambda^r\rangle$ (resp. $\mathcal{N}\langle\Lambda^b\rangle$) denotes the set of finite formal sums of terms (resp. bags), with 0 referring to the neutral element.

$$\mathbb{M}, \mathbb{N} \in \mathcal{N}\langle\Lambda^r\rangle \qquad \mathbb{P}, \mathbb{Q} \in \mathcal{N}\langle\Lambda^b\rangle \qquad \mathbb{A}, \mathbb{B} \in \mathcal{N}\langle\Lambda^{(b)}\rangle = \mathcal{N}\langle\Lambda^r\rangle \cup \mathcal{N}\langle\Lambda^b\rangle \qquad \text{sums}$$

Note that $\mathcal{N}\langle\Lambda^{(b)}\rangle$ does not denote the $\mathcal{N}$-module generated over $\Lambda^{(b)} = \Lambda^r \cup \Lambda^b$ but rather the union of the two $\mathcal{N}$-modules. In other words, sums must be taken only in the same sort.

Notation 2 *We introduce the following abbreviations:*

- $\lambda x.\Sigma_i M_i = \lambda x.(\Sigma_i M_i)$,
- $\Sigma_i [M_i] \uplus P = [(\Sigma_i M_i)] \uplus P$,
- $\Sigma_{i,j} M_i P_j = (\Sigma_i M_i)(\Sigma_j P_j)$,
- $[M_1^!, \dots, M_k^!] \uplus P = [(\Sigma_i M_i)^!] \uplus P$.

We will write $\vec{L}$ for $L_1, \dots, L_k$ and $\vec{N}^!$ for $N_1^!, \dots, N_n^!$. We will also abbreviate $M\langle L_1/x\rangle \cdots \langle L_k/x\rangle$ in $M\langle\vec{L}/x\rangle$. Moreover, given a sequence $\vec{L}$ and an integer $1 \le i \le k$ we set $\vec{L}_{-i} = L_1, \dots, L_{i-1}, L_{i+1}, \dots, L_k$.

Every applicative term MP can be written in a unique way as $M[\vec{L}, \vec{N}^!]$.

The reduction rule generating the *resource calculus* is the following:

$$(\lambda x.M)[\vec{L}, \vec{N}^!] \to_R M\langle\vec{L}/x\rangle[\Sigma_{i=1}^n N_i/x]$$

where:

- $A[N/x]$ is the usual substitution of N for x in A. It is extended to sums as in $\mathbb{A}[\mathbb{N}/x]$ by linearity in $\mathbb{A}$.

- $A\langle N/x\rangle$ is the *linear substitution* defined inductively as follows:

$$y\langle N/x\rangle = \begin{cases} N \text{ if } x = y \\ 0 \quad \text{otherwise} \end{cases} \qquad \begin{array}{l} (\lambda y.M)\langle N/x\rangle = \lambda y.M\langle N/x\rangle \\ (MP)\langle N/x\rangle = M\langle N/x\rangle P + M(P\langle N/x\rangle) \end{array}$$

$$[M]\langle N/x\rangle = [M\langle N/x\rangle] \qquad [\,]\langle N/x\rangle = 0$$

$$[M^!]\langle N/x\rangle = [M\langle N/x\rangle,\ M^!] \quad (P \uplus R)\langle N/x\rangle = P\langle N/x\rangle \uplus R + P \uplus R\langle N/x\rangle$$

It is extended to $\mathbb{A}\langle \mathbb{N}/x\rangle$ by bilinearity[8] in both $\mathbb{A}$ and $\mathbb{N}$.

The operation $M\langle N/x\rangle$ on resource λ-terms is roughly equivalent to the operation $\frac{\partial S}{\partial x} \cdot T$ on differential λ-terms (cf. Lemma 6.2 below).

We introduce the type system R characterizing the *simply typed resource calculus*.

Type System R.

$$\frac{\Gamma(x) = \sigma}{\Gamma \vdash_R x : \sigma}\ (\mathrm{Rx}) \qquad \frac{\Gamma, x : \sigma \vdash_R M : \tau}{\Gamma \vdash_R \lambda x.M : \sigma \to \tau}\ (\mathrm{R}\lambda)$$

$$\frac{\Gamma \vdash_R M : \sigma \to \tau \quad \Gamma \vdash_R P : \sigma}{\Gamma \vdash_R MP : \tau}\ (\mathrm{R@}) \qquad \frac{\Gamma \vdash_R N : \sigma \quad \Gamma \vdash_R P : \sigma}{\Gamma \vdash_R [N^{(!)}] \uplus P : \sigma}\ (\mathrm{Rb})$$

$$\frac{}{\Gamma \vdash_R [\,] : \sigma}\ (\mathrm{R}[\,]) \qquad \frac{\Gamma \vdash_R \mathbb{A} : \sigma \quad \Gamma \vdash_R \mathbb{B} : \sigma \quad \mathbb{B} \neq 0}{\Gamma \vdash_R \mathbb{A} + \mathbb{B} : \sigma}\ (\mathrm{R+})$$

6.2 Its Semantics

In this subsection we show that differential λ-categories are models also of the simply typed resource calculus. This result is achieved by first translating the resource calculus in the differential λ-calculus, and then applying the machinery of Section 4.3.

We add the *permutative equality* to differential λ-calculus, i.e., we consider differential λ-terms up to the following equivalence $\mathsf{D}(\mathsf{D}s \cdot u) \cdot v = \mathsf{D}(\mathsf{D}s \cdot v) \cdot u$. This is useful for the translation since, in the resource calculus, bags are considered as multisets (thus they are equal up to permutation of resources).

Remark 6.1 It is not difficult to check that in every differential λ-category the interpretations of $\mathsf{D}(\mathsf{D}s \cdot u) \cdot v$ and $\mathsf{D}(\mathsf{D}s \cdot v) \cdot u$ coincide.

We can now easily translate the resource calculus into the differential λ-calculus as follows:

- $x^o = x$,
- $(\lambda x.M)^o = \lambda x.M^o$,
- $(M[\vec{L}, \vec{N}^!])^o = (\mathsf{D}^k M^o \cdot \vec{L}^o)(\Sigma_{i=1}^n N_i^o)$,

[8] $F(A, B)$ is extended by bilinearity by setting $F(\Sigma_i A_i, \Sigma_j B_j) = \Sigma_{i,j} F(A_i, B_j)$.

where $\mathsf{D}^k\, M^o \cdot \vec{L}^o$ is an abbreviation for $\mathsf{D}(\cdots(\mathsf{D}\, M^o \cdot L^o_1) \cdots\cdot L^o_k)$. The translation is then extended to elements in $\mathcal{N}\langle\Lambda^r\rangle$ by setting $(\Sigma_{i=1}^n M_i)^o = \Sigma_{i=1}^n M_i^o$.

Lemma 6.2 *Let $M, N \in \Lambda^r$ and x be a variable. Then:*

(i) $(M\langle N/x\rangle)^o = \frac{\partial M^o}{\partial x} \cdot N^o$,

(ii) $(M[N/x])^o = M^o[N^o/x]$.

Proof. (i) By structural induction on M. The only difficult case is $M \equiv M'[\vec{L}, \vec{N}^!]$. By definition of $(-)^o$ and of linear substitution we have:

$$((M'[\vec{L}, \vec{N}^!])\langle N/x\rangle)^o = (M'\langle N/x\rangle[\vec{L}, \vec{N}^!])^o + (M'([\vec{L}, \vec{N}^!]\langle N/x\rangle))^o =$$

$$\underbrace{(M'\langle N/x\rangle[\vec{L}, \vec{N}^!])^o}_{(1)} + \underbrace{(\Sigma_{j=1}^k M'[L_j\langle N/x\rangle, \vec{L}_{-j}, \vec{N}^!])^o}_{(2)} + \underbrace{(\Sigma_{i=1}^n M'[N_i\langle N/x\rangle, \vec{L}, \vec{N}^!])^o}_{(3)}.$$

Let us consider the three addenda separately.

(1) By definition of the translation $(-)^o$ we have that $(M'\langle N/x\rangle[\vec{L}, \vec{N}^!])^o = (\mathsf{D}^k\,(M'\langle N/x\rangle)^o \cdot \vec{L}^o)(\Sigma_{i=1}^n N_i^o)$. By applying the induction hypothesis, this is equal to $(\mathsf{D}^k\,(\frac{\partial M'^o}{\partial x} \cdot N^o) \cdot \vec{L}^o)(\Sigma_{i=1}^n N_i^o)$.

(2) By definition of the map $(-)^o$ we have $(\Sigma_{j=1}^k M'[L_j\langle N/x\rangle, \vec{L}_{-j}, \vec{N}^!])^o = \Sigma_{j=1}^k(\mathsf{D}^{k-1}\,(\mathsf{D}\, M'^o \cdot (L_j\langle N/x\rangle)^o) \cdot \vec{L}^o_{-j})(\Sigma_{i=1}^n N_i^o)$. By the induction hypothesis, this is equal to $\Sigma_{j=1}^k(\mathsf{D}^{k-1}\,(\mathsf{D}\, M'^o \cdot (\frac{\partial L_j^o}{\partial x} \cdot N^o)) \cdot \vec{L}^o_{-j})(\Sigma_{i=1}^n N_i^o)$.

(3) By definition of the map $(-)^o$ we have $(\Sigma_{j=1}^n M'[N_j\langle N/x\rangle, \vec{L}, \vec{N}^!])^o = \Sigma_{j=1}^n(M'[N_j\langle N/x\rangle, \vec{L}, \vec{N}^!])^o = \Sigma_{j=1}^n(\mathsf{D}^k\,(\mathsf{D}\, M'^o \cdot (N_j\langle N/x\rangle)^o) \cdot \vec{L}^o)(\Sigma_{i=1}^n N_i^o)$. By induction hypothesis, this is equal to $\Sigma_{j=1}^n(\mathsf{D}^k\,(\mathsf{D}\, M'^o \cdot (\frac{\partial N_j^o}{\partial x} \cdot N^o)) \cdot \vec{L}^o)(\Sigma_{i=1}^n N_i^o)$. By permutative equality this is equal to $\Sigma_{j=1}^n(\mathsf{D}\,(\mathsf{D}^k\, M'^o \cdot \vec{L}^o) \cdot (\frac{\partial N_j^o}{\partial x} \cdot N^o))(\Sigma_{i=1}^n N_i^o)$.

To conclude the proof it is sufficient to check that $\frac{\partial}{\partial x}((\mathsf{D}^k\, M'^o \cdot \vec{L}^o)(\Sigma_{i=1}^n N_i^o)) \cdot N^o$ is equal to the sum of (1), (2) and (3).

(ii) By straightforward induction on M. □

Proposition 6.3 *For all $M \in \Lambda^r$ we have:*

(i) $M \to_R N$ *implies* $M^o \twoheadrightarrow_D N^o$*.*

(ii) $\Gamma \vdash_R M : \sigma \iff \Gamma \vdash_D M^o : \sigma$

(iii) (Subject Reduction for R) $\Gamma \vdash_R M : \sigma$ *and* $M \to_R N$ *implies* $\Gamma \vdash_R N : \sigma$*.*

Proof. (i) Let $M \equiv (\lambda x.M')[\vec{L}, \vec{N}^!]$ and $N \equiv M'\langle\vec{L}/x\rangle[\Sigma_{i=1}^n N_i/x]$. By definition of $(-)^o$ we have $((\lambda x.M')[\vec{L}, \vec{N}^!])^o = (\mathsf{D}^k\,(\lambda x.M'^o) \cdot \vec{L}^o)(\Sigma_{i=1}^n N_i^o) \twoheadrightarrow_D M'^o\langle\vec{L}^o/x\rangle[\Sigma_{i=1}^n N_i^o/x]$ which is equal to N^o by Lemma 6.2.

(ii) by induction on the length of the proofs of $\Gamma \vdash_R M : \sigma$ and $\Gamma \vdash_D M^o : \sigma$.

(iii) Suppose $\Gamma \vdash_R M : \sigma$ and $M \to_R N$. Then by (ii) we have that $\Gamma \vdash_D M^o : \sigma$. By (i) we know that $M^o \twoheadrightarrow_D N^o$ and since System D enjoys the subject reduction we get $\Gamma \vdash_D N^o : \sigma$. We conclude by (ii). □

Remark 6.4 The two results above generalize easily to sums of terms (*i.e.*, to elements $\mathbb{M} \in \mathcal{N}\langle\Lambda^r\rangle$).

Resource λ-terms can be interpreted in any differential λ-category trough their translation $(-)^o$. Indeed, it is sufficient to set $|\Gamma \vdash_R M : \sigma| = |(M^o)^\sigma|_\Gamma$. From this fact, Proposition 6.3 and Remarks 6.1,6.4 it follows that differential λ-categories are models of the simply typed resource calculus.

7 Conclusions and Further Works

In this paper we have provided sufficient conditions on Cartesian closed differential categories for being models of the simply typed differential λ-calculus. We have also shown that they can also be used for interpreting the simply typed resource calculus, by providing a faithful translation between the two calculi.

However, differential λ-categories may also provide general mathematical frameworks in which many models of the *untyped* differential λ-calculus may live. This is the case of the category **MRel**, while it is not the case of **MFin** since it does not contain any reflexive object. In a forthcoming paper we will provide suitable conditions on the reflexive objects of a differential λ-category for being models of the untyped differential λ-calculus (intuitively the retraction should be 'linear', in a sense to be specified). We will then show that the reflexive object $\mathcal{D}$ we have built in [5] satisfies these conditions.

Another interesting line of research is to characterize categorical models of the differential λ-calculus at the level of SMCC's (symmetric monoidal closed categories). In [3] Blute *et al.* show that (monoidal) differential categories [2] give rise to Cartesian differential categories via the co-Kleisli construction. In the same spirit, we would like to provide sufficient and necessary conditions on SMCC's for giving rise to differential λ-categories (indeed, the examples of differential λ-categories we gave in Section 5 may be generated in such a way).

Notice that, in monoidal frameworks, categorical proofs become often awkward due to the symmetric properties of the tensor product $\otimes$. It would be then interesting to define a graphical formalism allowing to represent in a pleasant and intuitive way the morphisms of these categories. This formalism could be inspired by differential proofnets or interaction nets [9], but should satisfy (at least) the following properties: there should be a 1-to-1 correspondence between a morphism and its graphical representation (maybe up to some well chosen equivalence on morphisms); the formalism should not ask for extra properties of the category, like the presence of the operator $\invamp$ or the dualizing object $\perp$.

Acknowledgement

Many thanks to Guy McCusker, Michele Pagani, Paolo Tranquilli and the anonymous reviewers for helpful comments and suggestions.

References

[1] A. Asperti and G. Longo. Categories, types and structures. Category theory for the working computer scientist. M.I.T. Press, 1991.

[2] R.F. Blute, J.R.B. Cockett and R.A.G. Seely. *Differential categories.* Math. Struct. in Comp. Sci.,16(6):1049–1083, Cambridge University Press, 2006.

[3] R.F. Blute, J.R.B. Cockett and R.A.G. Seely. *Cartesian differential categories.* Theory and Applications of Categories, Vol. 22, No. 23, pp 622-672, 2009.

[4] G. Boudol. *The lambda-calculus with multiplicities.* CONCUR'93, pp. 1-6, 1993.

[5] A. Bucciarelli, T. Ehrhard and G. Manzonetto. *Not enough points is enough.* CSL'07, pp. 298-312, 2007.

[6] T. Ehrhard and L. Regnier. *The differential lambda-calculus.* Theor. Comput. Sci. 309(1-3): 1-41, 2003.

[7] T. Ehrhard. *Finiteness spaces.* Math. Struct. in Comp. Sci. 15(4): 615-646, 2005.

[8] T. Ehrhard and L. Regnier. *Böhm trees, Krivine's machine and the Taylor expansion of λ-terms.* CiE, pp. 186-197, 2006.

[9] T. Ehrhard and L. Regnier. *Differential interaction nets.* Theor. Comput. Sci. 364(2): 166-195, 2006.

[10] T. Ehrhard and L. Regnier. *Uniformity and the Taylor expansion of ordinary λ-terms.* Theor. Comput. Sci. 403(2-3): 347-372, 2008.

[11] J.-Y. Girard. *Normal functors, power series and λ-calculus.* APAL. 37(2):129-177, 1988.

[12] M. Pagani and P. Tranquilli. *Parallel reduction in resource λ-calculus.* APLAv09, pp. 226-242, 2009.

[13] P. Tranquilli. *Nets Between Determinism and Nondeterminism.* PhD thesis, Univ. of Paris 7 and Univ. of Roma 3, 2009.