

Categorical Models of the Differential λ -Calculus Revisited

J.R.B. Cockett^{1,2} J.D. Gallagher³

*Department of Computer Science
University of Calgary
Calgary, Canada*

Abstract

The paper shows that the Scott-Koymans theorem for the untyped λ -calculus extends to the differential λ -calculus. The main result is that every model of the untyped differential λ -calculus may be viewed as a differential reflexive object in a Cartesian closed differential category. This extension of the Scott-Koymans theorem depends critically on unravelling the somewhat subtle issue of which idempotents can be split so that differential structure lifts to the idempotent splitting.

The paper uses (total) Turing categories with “canonical codes” as the basic categorical semantics for the λ -calculus. It shows how the main result may be developed in a modular fashion by first adding left-additive structure to a Turing category, and then – on top of that – differential structure. For both levels of structure it is necessary to identify how “canonical codes” behave with respect to the added structure and, furthermore, how “universal objects” behave. The latter is closely tied to the question – which is the crux of the paper – of which idempotents can be split in these more structured settings.

Keywords: Scott-Koymans, Differential Lambda Calculus, Categorical Models

1 Introduction

In [12], Ehrhard and Regnier introduced the differential λ -calculus to give a syntactic counterpart for the models of linear logic which Ehrhard had introduced in [10,11]. In these models proofs were interpreted as differentiable maps with the linear maps, in the sense of linear logic, becoming rather elegantly the maps which were linear in the differential sense. The differential λ -calculus, introduced a new aspect because, as described in [12], it was an untyped system with a confluent rewriting system and, thus provided a model of computability. Furthermore, it was immediately apparent that this calculus was very closely related to the resource λ -calculus [6,5,7]. This

¹ partially supported by NSERC

² Email: robin@cpsc.ucalgary.ca

³ Email: jdgallag@ucalgary.ca

confluence of developments, thus, initiated the study of models of computability in which the computable functions admit a differentiation.

The categorical semantics for these settings was developed in series of papers. Initially the tensorial (or linear logic) side of the story was developed in [2]: this closely followed the path of Ehrhard’s more model driven development. Tensor differential categories (also written \otimes -differential categories) came equipped with a comonad (an exponential modality)⁴ and it was understood that the differential λ -calculus would then be interpreted in the coKleisli category for this comonad.

Of course, this rather indirect approach did not facilitate the exploration of these coKleisli categories which rapidly become the main focus of attention. The next step – motivated not least by desire to understand the axiomatic behaviour of differentiation in classical calculus – was, therefore, to develop a direct axiomatization for these categories. Toward this end Cartesian differential categories (also written \times -differential categories) were introduced [3]. Importantly, these were more general than simply being an axiomatization of the coKleisli category for a \otimes -differential category. While it is certainly the case that coKleisli categories of \otimes -differential categories are \times -differential categories⁵, the converse is certainly not true.

The next step in this development involved \times -differential categories which were – in the appropriate sense – Cartesian closed. In [8] a sound and complete interpretation of the simply typed differential λ -calculus into Cartesian closed differential categories was provided. Furthermore, the connection between the resource λ -calculus and the differential λ -calculus was then implicitly determined by tying both to the same categorical semantics. To complete the story it thus only remained to provide a precise account of the semantics of the untyped differential λ -calculus. At this stage, it was abundantly clear that the interpretation of the untyped differential calculus should be into some sort of reflexive object in a Cartesian closed differential category and, thus, there should be an analogue of the Scott-Koymans⁶ theorem [14,18] which says that all model arise from such a situation.

In [17], Manzonetto initiated the investigation of models of the untyped differential λ -calculus. He showed that linear reflexive objects in a Cartesian differential category soundly interpret the untyped differential λ -calculus. Furthermore, he gave a completeness theorem, for the differential λ -calculus, however, with two additional equations:

$$\lambda x.(Ds \cdot t)x = Ds \cdot t \quad \text{and} \quad a + a = a.$$

Thus, he did not manage to provide a general completeness theorem which paralleled the Scott-Koymans result. In this paper we revisit this problem and we show that

⁴ In fact, \otimes -differential categories come with different strengths of axiomatizations. In [2] an effort was made to obtain the weakest possible axiomatization. Of note, however, is the stronger notion which was introduced by Marcello Fiore in [13].

⁵ A precise characterization of when a Cartesian differential category is the coKleisli category of a \otimes -differential category is described in [4].

⁶ At the time of the development of this theorem, Koymans was a PhD. student, Dana Scott was widely publicizing his results, and also Jim Lambek with Phil Scott were in the process of writing their book, [15], and, indeed, were circulating chapters for comment. Their book, in particular, has a section on C-monoids which provide the semantics for the untyped λ -calculus with both η - and β -equality. In addition, they introduced the notion of a “weak” C-monoid in the exercises and noted that these provide the semantics of the λ -calculus with just β -equality. The notion of “weak” there is exactly our notion of having canonical codes. Note that, in this paper, we use “weak” to mean that the choice *does not* necessarily admit substitution.

by subtly changing the requirements on the idempotents which one splits one can indeed obtain a complete analogue of the Scott-Koymans result. In Manzonetto's work it was assumed – very reasonably – that the idempotents one should split had to be linear. Unfortunately, this meant that, as Curry's pairing combinator, $(a, b) \mapsto \lambda p.p\,a\,b$, is not linear one is forced to look for an alternative idempotent which can encode pairing. Manzonetto actually found such a combinator: it was the more complicated $(a, b) \mapsto \lambda y.(a + Dy \cdot b)$. However, in order to make this combinator behave correctly, he had to restrict to the case where addition was idempotent.

To access the more general situation, which is the subject of this paper, it is necessary to understand more fully the behaviour of idempotents in a Cartesian differential category. To achieve this one not only has to understand their behaviour with respect to differentiation but also with respect to the left-additive structure. We shall, therefore, redevelop this program by describing the categorical semantics not only of models of the differential λ -calculus but also of models of the “additive” λ -calculus. This is because moving directly from a Cartesian closed differential setting to a model of the differential λ -calculus, without this intermediate step, tends to obscure the subtle behaviour required of the idempotents which must be split. In fact, in this paper we shall start the development even further back: we actually redevelop the Scott-Koymans theorem itself from the more modern and unifying perspective of Turing categories [9]. One justification for this is that, in the theory of Turing categories, idempotents and idempotent completions play a very central role. This predisposes one to take the behaviour of idempotents very seriously as one begins to add structure.

Because our aim is to model the λ -calculus, we shall exclusively focus on Turing categories all of whose maps are total: these were introduced by Longo and Moggi [16] – albeit under a different name – and their realizability theory was developed by Birkedal [1]. Such a Turing category is a weakly Cartesian closed category with an object, T , called a Turing object which is universal. An object in a category is *universal* in case all objects are a retracts of it. In order to model the untyped λ -calculus, with β -equality, we shall employ the notion of a Turing category with canonical codes. This allows us to interpret the term calculus of the untyped λ -calculus very directly into these categorial models. In particular, this step does not involve consideration of idempotents. This approach, in fact, closely parallels the approach Koymans followed in [14], and allows a convenient separation of concerns which is useful later. To recapture the Scott-Koymans theorem one observes that splitting idempotents yields a Cartesian closed category in which the Turing object is still universal: this immediately makes that the Turing object a reflexive object.

This is the approach that we then follow, in a modular fashion, as we successively add structure. First we add left-additive structure and develop the categorical semantics of the untyped additive λ -calculus, and then, following the same pattern, arrive at the categorical semantics of the untyped differential λ -calculus. At each stage we must appropriately strengthen both the notion of being weakly cartesian closed (with canonical codes) and of being a universal object – where the notion of a universal object, as one adds structure, crucially depends on understanding which

idempotents can be split.

The payoff of this reconstruction is that it allows a general analogue for the Scott-Koymans theorem for untyped differential λ -models.

2 The λ -Calculus

The classical notion of a categorical model of the untyped λ -calculus is as a reflexive object in a cartesian closed category. This is an object U which has as a retract its object of endomorphisms, $[U, U] \triangleleft_r^s U$.⁷ One way to approach the Scott-Koymans completeness result is to interpret the λ -calculus across this retraction: this is rather messy as one must repeatedly use the section and retraction to interpret terms. An alternative approach, which parallels the approach of Koymans, is to interpret the λ -calculus into a Turing category with canonical codes and then show that splitting the idempotents results in a Cartesian closed category in which the Turing object is still universal. This approach gives an elegant separation of the two concerns (interpretation and idempotent splitting) and, furthermore, allows the introduction of a more direct categorical counterpart to the untyped λ -calculus: namely Turing categories with canonical codes.

2.1 Syntax of the λ -calculus

The syntax of the λ -calculus may be described as “untyped” terms in context. The term formation rules consist of term formation rules for a Cartesian theory, see table 1, and the special terms for the “untyped” λ -calculus see Table 2.

In the term formation rules for a Cartesian theory we insist that the variables of a context are distinct. We shall only allow variables for atomic types and allow the formation of variable patterns corresponding to composite Cartesian products of objects. This allows a smooth transition to the categorical semantics as we can turn any sequent into a map, a sequent with one premise:

$$p : X \vdash t : T$$

by simply collecting the context, using the pattern rules, into one type. When substituting for a pattern one then has to match the term being substituted to the pattern: thus, for example, the substitution $[(t, s)/(x, y)]$ becomes, equivalently, $[t/x, s/y]$. The type system ensures that term and pattern in a substitution have exactly the same form.

As usual we allow α -conversion of bound variables and β -equality

$$\lambda x.m \equiv_\alpha \lambda y.(m[y/x]) \quad (\lambda x.m)n \equiv_\beta m[n/x]$$

and generate the theory Λ_β by forming the smallest congruence on terms that contains the above equations. There is an associated category of this theory, $\mathcal{C}(\Lambda_\beta)$,

⁷ Here $[U, U]$ is the internal hom sometimes written U^U or $U \Rightarrow U$. The notation $[U, U] \triangleleft_r^s U$ means $[U, U]$ is a retract of U with $s : [U, U] \rightarrow U$ the section and $r : U \rightarrow [U, U]$ the retraction so $sr = 1_{[U, U]}$.

$\frac{A \text{ an atomic type}}{\Gamma, x : A, \Gamma' \vdash x : A} \text{ Projection}$
$\frac{\Gamma, \Gamma' \vdash t : X}{\Gamma, () : 1, \Gamma' \vdash t : X} \text{ Unit Pattern}$
$\frac{\Gamma, x : X, y : Y, \Gamma' \vdash t : Z}{\Gamma, (x, y) : X \times Y, \Gamma' \vdash t : Z} \text{ Pair Pattern}$
$\frac{\Gamma \vdash t : X \Gamma \vdash s : Y}{\Gamma \vdash (t, s) : X \times Y} \text{ Pairing}$
$\frac{\Gamma, \Gamma' \vdash vs : X \Gamma, p : X, \Gamma' \vdash t : Z}{\Gamma, \Gamma' \vdash t[s/p] : Z} \text{ Cut/Substitution}$

Table 1
Term formation rules for a Cartesian theory

$\frac{\Gamma \vdash m : U \quad \Gamma \vdash n : U}{\Gamma \vdash mn : U} \text{ Application}$	$\frac{\Gamma, x : U, \Gamma' \vdash m : U}{\Gamma, \Gamma' \vdash \lambda x. m : U} \text{ Abstraction}$
--	---

Table 2
Term formation rules for the λ -calculus

called the classifying category of Λ_β :

Objects: Words in $U, \times, 1$.

Arrows: A map $T \rightarrow S$ corresponds to a sequent with one type on the left hand side

$$p : T \vdash m : S$$

under the term equivalence of Λ_β .

Composition: Composition is substitution:

$$\frac{p : X \vdash m : Y \quad p' : Y \vdash n : Z}{p : X \vdash n[m/p'] : Z}$$

We note, rather modestly, that this is a Cartesian category:

Proposition 2.1 $\mathcal{C}(\Lambda_\beta)$ is a Cartesian category.

The rest of this section develops the properties of this category.

2.2 Total Turing categories with canonical codes

A **(total) Turing category** \mathbb{X} is a Cartesian category with:

[T.1] An object T and for each pair of objects B, C a map $T \times B \xrightarrow{\bullet_{BC}} C$;

[T.2] For each map $A \times B \xrightarrow{f} C$ a map $A \xrightarrow{c_f} T$ such that

$$\begin{array}{ccc} T \times B & \xrightarrow{\bullet_{BC}} & C \\ c_f \times 1 \uparrow & \nearrow f & \\ A \times B & & \end{array}$$

A Turing category has **canonical codes** when there is a function

$$\mathbb{X}(A \times B, C) \xrightarrow{\lambda(_)} \mathbb{X}(A, T)$$

such that

$$\begin{array}{ccc} & T & \\ \lambda[(g \times 1)f] \nearrow & \uparrow \lambda[f] & \\ D \xrightarrow{g} & A & \end{array} \quad \begin{array}{ccc} T \times B & \xrightarrow{\bullet_{BC}} & C \\ \lambda[f] \times 1 \uparrow & \nearrow f & \\ A \times B & & \end{array}$$

commute. This means that this function, $\lambda[_]$, for forming codes has a substitutional property, namely that $\lambda[(g \times 1)f] = g\lambda[f]$ and that $\lambda[f]$ is indeed a code for f in the sense that $(\lambda(f) \times 1) \bullet_{BC} = f$.

Observe first that the property of having canonical codes can be simplified:

Lemma 2.2 *A Turing category \mathbb{X} has canonical codes if and only if there is a code $\lambda[\bullet_{BC}]$ for each $\bullet_{BC} : T \times B \rightarrow C$ such that whenever $(c_1 \times 1) \bullet_{BC} = f = (c_2 \times 1) \bullet_{BC}$ that is c_1 and c_2 are codes for f , then $c_1 \lambda[\bullet_{BC}] = c_2 \lambda[\bullet_{BC}]$.*

Proof. The proof follows from considering the following diagram

$$\begin{array}{ccc} & T \times B & \xrightarrow{\bullet} C \\ & \uparrow \lambda(\bullet) \times 1 & \nearrow \bullet \\ & T \times B & \\ \lambda(f) \times 1 \nearrow & \uparrow c_i \times 1 & \\ A \times B & & \end{array}$$

□

The following is an economical “recognition” theorem for Turing categories with canonical codes:

Lemma 2.3 *To have a Turing category with canonical codes is to have a Cartesian category with a universal object T and a “Turing” map $T \times T \xrightarrow{\bullet} T$ which has canonical codes for all maps $T \times T \rightarrow T$.*

Proof. For the universality of the Turing object, in a Turing category, obtain the section as a code for $A \times 1 \xrightarrow{\pi_0} A$. To extract a Turing structure from a universality of T and a Turing map, derive a universal application for arbitrary objects by

$$T \times B \xrightarrow{1 \times s_B} T \times T \xrightarrow{\bullet} T \xrightarrow{r_C} C.$$

The canonical code for f is $\lambda[f] := s_A \lambda[(r_A \times r_B) f s_C]$. □

We use this to show:

Theorem 2.4 *$\mathcal{C}(\Lambda_\beta)$ is a Turing category with canonical codes.*

Proof. The universal map is $U \times U \xrightarrow{\bullet} U; (m, n) \vdash mn$. By lemma 2.3, it suffices to show that \bullet has canonical codes for terms $U \times U \rightarrow U; (x, y) \vdash t$. Define $\lambda[(x, y) \vdash t] := x \vdash \lambda y.t$. It is easy to see that this is a canonical code.

1 is a retract of U . To see that $U \times U$ is a retract of U use Curry’s pairing and projection combinators: the section is $(a, b) \vdash \lambda p.p a b$, and for the retraction take $a \vdash (a(\lambda xy.x), a(\lambda xy.y))$. □

From a logical perspective this theorem proves the *completeness* of the untyped λ -calculus with respect to models which are Turing categories with canonical codes. With some more work (some of which is developed below) one can exhibit this as part of an adjunction between λ -theories and Turing categories with canonical codes. As this is off the path of our development we shall leave it for a fuller exposition.

2.3 Interpreting the λ -calculus

We now show that Turing categories with canonical codes are sound models for the untyped λ -calculus by showing there is a canonical functor $\llbracket _ \rrbracket : \mathcal{C}(\Lambda_\beta) \rightarrow \mathbb{X}$ which carries the universal object onto the Turing object and application onto the Turing morphism $\bullet : T \times T \rightarrow T$. At this stage we should make an important remark: a Turing category may have more than one possible Turing object and more than one Turing morphism as one only needs the existence of such structure to be a Turing category. Thus, in this development, we need to strengthen the notion of a Turing category to specify, as part of being a Turing category, the intended Turing structure. In particular, for a Turing category with canonical codes the function which supplies the canonical codes will be part of this structure.

With this understanding, let \mathbb{X} be a Turing category with canonical codes, whose Turing object is T and universal map is $T \times T \xrightarrow{\bullet} T$. We define the interpretation of the untyped λ -calculus, $\llbracket _ \rrbracket : \mathcal{C}(\Lambda_\beta) \rightarrow \mathbb{X}$, as follows:

Objects: On objects define $\llbracket X \rrbracket$ to be $X[T/U]$.

Maps: On arrows:

$$\begin{aligned}
\llbracket p : X \vdash () : 1 \rrbracket &= !\llbracket X \rrbracket \\
\llbracket p : X \vdash (m, n) : Y \times Z \rrbracket &= \langle \llbracket p : X \vdash m : Y \rrbracket, \llbracket p : X \vdash n : Z \rrbracket \rangle \\
\llbracket x : U \vdash x : U \rrbracket &= 1_T \\
\llbracket (p_1, p_2) : X_1 \times X_2 \vdash x : U \rrbracket &= \pi_i \llbracket p_i : X_i \vdash x : U \rrbracket \quad \text{where } x \in p_i \\
\llbracket p : X \vdash mn : U \rrbracket &= \langle \llbracket p : X \vdash m : U \rrbracket, \llbracket p : X \vdash n : U \rrbracket \rangle \bullet \\
\llbracket p : X \vdash \lambda z. m : U \rrbracket &= \lambda [\llbracket (p, z) : X \times U \vdash m : U \rrbracket]
\end{aligned}$$

We then have:

Theorem 2.5 *If \mathbb{X} is a Turing category with canonical codes,*

$$\llbracket - \rrbracket : \mathcal{C}(\Lambda) \rightarrow \mathbb{X}$$

is a Cartesian functor which preserves the Turing object, the Turing map, and the canonical codes.

The proof is standard.

2.4 Reflexive objects

An important observation for Turing categories is that the idempotent splitting of a Turing category \mathbb{X} , $\text{Split}(\mathbb{X})$, is again a Turing category (see [9]), furthermore, the Turing object and the Turing morphism are preserved in this splitting. Our next objective is to show that splitting the idempotents of a Turing category which has canonical codes results not just in a Turing category with canonical codes but in a Cartesian closed Turing category. As the Turing object is still universal in the idempotent splitting, this, in particular, means that it is a reflexive object. This allows us to conclude that all models of the untyped λ -calculus arise from a reflexive object $[T, T] \triangleleft_r^s T$ in a Cartesian closed Turing category, and this is how we interpret the Scott-Koymans theorem.

Theorem 2.6 *When \mathbb{X} is a Turing category with canonical codes then $\text{Split}(\mathbb{X})$ is a cartesian closed Turing category.*

The proof is that of Koymans [14].

Note that $[1_T, 1_T] = \lambda[\bullet]$ is an idempotent which in $\text{Split}(\mathbb{X})$ witnesses that the Turing object in the splitting is a reflexive object.

3 The additive λ -calculus

The previous section sets the broad outline for this and the next section. In these sections we will follow our program of adding structure step by step to the untyped λ -calculus. The first step in this program is to consider left-additive structure: it is required in order to discuss differential structure.

A **left-additive category** \mathbb{X} is a category in which each homset $\mathbb{X}(A, B)$ is a

$$\frac{\Gamma \vdash m : U \quad \Gamma \vdash n : U}{\Gamma \vdash (m + n) : U} \text{ Addition} \qquad \frac{}{\Gamma \vdash 0 : U} \text{ Zero}$$

Table 3
Term formation rules addition in the additive λ -calculus

commutative monoid, and where

$$f(g + h) = fg + fh \qquad f0 = 0.$$

A map f in a left additive category is **additive** when $(g + h)f = gf + hf$ and $0f = 0$.

A **Cartesian left additive category** is a left additive category with finite products such that

$$(f + g) \times (h + k) = (f \times h) + (g \times k) \qquad 0 \times 0 = 0$$

and where $\Delta : A \rightarrow A \times A$ and all projections π_i are additive. An important characterization of Cartesian left-additive categories is:

Proposition 3.1 (Proposition 1.2.2 [3]) *To have a Cartesian left additive category is to have a Cartesian category in which each object, X , has a canonical commutative monoid structure, $X \times X \xrightarrow{+X} X \xleftarrow{0_X} 1$, that satisfies the following coherence:*

$$+_{A \times B} := (A \times B) \times (A \times B) \xrightarrow{\text{ex}} (A \times A) \times (B \times B) \xrightarrow{+A \times +B} A \times B.$$

3.1 Syntax of the additive λ -calculus

The syntax of the untyped additive λ -calculus is again defined by untyped terms in context. The term formation rules use the usual Cartesian rules Table 1, the formation rules for the λ -calculus Table 2, and the additive rules of Table 3.

The equations for β -equality and α -conversion hold together with equations to make $(U, +, 0)$ into a commutative monoid, and

$$\begin{aligned} \lambda x.(m + n) &= \lambda x.m + \lambda x.n & \text{and} & & \lambda x.0 &= 0 \\ (m + n)a &= ma + na & \text{and} & & 0a &= 0. \end{aligned}$$

These equations generate the theory $\Lambda_{\beta+}$.

As before there is an associated category, $\mathcal{C}(\Lambda_{\beta+})$, called the classifying of the additive λ -calculus. The objects are words in $U, \times, 1$ and the arrows are sequents with one premise on the left of the turnstile. Composition is, again, substitution. We state modestly:

Proposition 3.2 $\mathcal{C}(\Lambda_{\beta+})$ is a Cartesian category in which U is a commutative monoid.

Our next task is to develop the categorical semantics of the additive λ -calculus, and thus, the properties of this category. To do this we must address two related issues: namely what it means to be an *additive* universal object and what it means to have *additive* canonical codes.

3.2 Additive universal objects

The idea behind a universal object in a category is that the entire category is determined by the monoid of endomorphisms of that object. For plain categories this just means that every object is a retract of U . However, in a Cartesian left additive category a universal object must have a stronger property as it must also induce the additive structure on each object. An object U in a left-additive category \mathbb{X} is an **additive universal** object in case every object, A , is a retract of U in such a manner that the retraction r_A , in $A \triangleleft_{r_A}^{s_A} U$, is an additive map. This requirement ensures that the additive structure on A is determined by that of U :

Lemma 3.3 In any left-additive Cartesian category with a retract $A \triangleleft_{r_A}^{s_A} U$ which has r_A additive:

- (i) $A \times A \xrightarrow{+_A} A = A \times A \xrightarrow{s_A \times s_A} U \times U \xrightarrow{+} U \xrightarrow{r_A} A$ and $0_A = 0r_A$;
- (ii) $e_A = r_A s_A$ satisfies $+_U e_A = (e_A \times e_A) +_U e_A$.

Note that (i) implies that $(fe + ge)r = (f + g)r$. When an idempotent $e : U \rightarrow U$ satisfies part (ii), that is $+_U e = (e \times e) +_U e$, we will say e is a **retractively additive** idempotent.

The following – completely general – lemma lets us define additive structure on objects using additive retractions:

Lemma 3.4 If $(U, +_U, 0_U)$ is a commutative monoid in a Cartesian category, \mathbb{X} , and $A \triangleleft_{r_A}^{s_A} U$ with $e_A = r_A s_A$ retractively additive – that is $(e_A \times e_A) +_U e_A = +_U e_A$ – then there is a unique commutative monoid structure on A which makes r_A a homomorphism.

This means, when U is a universal object, which is a commutative monoid, we may, using Lemma 3.4, induce additive structure on any retract A of U , whose induced idempotent is retractively additive: this will automatically make the retract additive. In order, to create left-additive structure on the whole category from the additive structure on U , we must select a particular way in which each object is an additive retract so that we may induce a unique additive structure on each object. Furthermore, to ensure the result is a *Cartesian* left-additive category, it is also necessary for these induced additions to be compatible with the product: this means we must demand that the additive structure on the product $A \times B$ be defined componentwise.

A **universal structure**, \mathcal{U} , for $U \in \mathbb{X}$ consists of, for each $A \in \mathbb{X}$, a way in which A is a retract of U , $\mathcal{U}(A) = A \triangleleft_{r_A}^{s_A} U$. Clearly if U has a universal structure

it must be a universal object. A Cartesian category, \mathbb{X} , has **chosen products** in case there are chosen functors $\times : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{X}$ and $1 : \mathbb{K} \rightarrow \mathbb{X}$ right adjoint to the diagonal and final functors respectively. In a Cartesian category with chosen products, if $(U, +_U, 0_U)$ is a commutative monoid, then a universal structure \mathcal{U} for U is said to be **additively coherent** in case:

[UAC.1] $\mathcal{U}(U) = U \triangleleft_{1_U}^{1_U} U$;

[UAC.2] The retraction, $r_{U \times U}$, of $\mathcal{U}(U \times U)$ satisfies:

$$\begin{array}{ccc} U \times U & \xrightarrow{+U} & U \\ \downarrow r_{U \times U} \times r_{U \times U} & & \downarrow r_{U \times U} \\ (U \times U) \times (U \times U) & \xrightarrow{\text{ex}} (U \times U) \times (U \times U) \xrightarrow{+U \times +U} & U \times U \end{array}$$

[UAC.3] Each idempotent $e_A = r_A s_A$, of $\mathcal{U}(A)$ is retractively U -additive;

[UAC.4] $\mathcal{U}(A \times B)$ has $s_{A \times B} = (s_C \times s_B) s_{U \times U}$ and $r_{A \times B} = r_{U \times U} (r_A \times r_B)$.

In a left-additive Cartesian category with an additive universal object, there is no guarantee that one has an additively coherent universal structure. However, given an additive universal structure one can generate a left-additive category:

Proposition 3.5 *If \mathbb{X} is a category with chosen products and a commutative monoid $(U, +_u, 0_U)$ which has an additively coherent universal structure, \mathcal{U} , then there is a unique left additive structure on \mathbb{X} making each retraction in \mathcal{U} additive.*

The proof uses the fact that the retractions are homomorphisms of the commutative monoid structure on objects.

3.3 Left-additive Turing categories

A **left-additive Turing category** is a Turing category which is Cartesian left-additive and has each universal application $\bullet_{BC} : T \times B \rightarrow C$ additive in its first argument:

$$\langle h + k, g \rangle \bullet = \langle h, g \rangle \bullet + \langle k, g \rangle \bullet \quad \text{and} \quad \langle 0, g \rangle \bullet = 0$$

A left-additive Turing category \mathbb{X} has **additive canonical codes** when, as a Turing category, \mathbb{X} has canonical codes such that, in addition, $\lambda[f + g] = \lambda[f] + \lambda[g]$ and $\lambda[0] = 0$.

We have the following recognition theorem for left-additive Turing categories:

Proposition 3.6 *\mathbb{X} is a left-additive Turing category if and only if \mathbb{X} is Cartesian left-additive, with an additively universal object T which has a Turing map $\bullet : T \times T \rightarrow T$ which is additive in its first argument.*

Furthermore, the Turing category has additively canonical codes if and only if the Turing map has additively canonical codes.

That objects have additive retractions from T uses the retraction from Proposition 2.3.

To see that we can construct canonical codes for $A \times B \rightarrow C$ from canonical codes for $T \times T \rightarrow T$ does require a bit of care concerning the definition of the addition on the maps: as only T has additive structure, $f + g : A \rightarrow B$ should be thought of as $(fs + gs)r$, where $B \triangleleft_s^r T$. The canonical code is $s_A \lambda[(r_A \times r_B)(fs_C + gs_C)]$.

We are now ready to state:

Theorem 3.7 $\mathcal{C}(\Lambda_{\beta+})$ is a left-additive Turing category with additively canonical codes.

The proof of this theorem relies heavily on the fact that Curry's retraction $U \rightarrow U \times U$ is additive, and this is enough to show that U has an additive universal structure. The rest of the proof follows easily.

3.4 Interpreting the additive λ -calculus

Our next objective is to show, that for any left-additive Turing category, \mathbb{X} , with additively canonical codes, there is a left-additive functor $\mathcal{C}(\Lambda_{\beta+}) \rightarrow \mathbb{X}$. From a logical perspective this says that left-additive Turing categories with additively canonical codes are sound models of the additive λ -calculus.

Let \mathbb{X} be a left-additive Turing category with additive canonical codes with Turing object T and Turing map is $\bullet : T \times T \rightarrow T$. The functor $\llbracket - \rrbracket : \mathcal{C}(\Lambda_{\beta+}) \rightarrow \mathbb{X}$ is defined in the same way as before on variables, application, abstraction, and tuples: all we have to describe is the interpretation of the additive structure:

$$\begin{aligned} \llbracket p : X \vdash 0 : U \rrbracket &= 0 : \llbracket X \rrbracket \rightarrow T \\ \llbracket p : X \vdash m + n : U \rrbracket &= \llbracket p : X \vdash m : U \rrbracket + \llbracket p : X \vdash n : U \rrbracket : \llbracket X \rrbracket \rightarrow T \end{aligned}$$

Proposition 3.8 When \mathbb{X} is a left-additive Turing category with additive canonical codes

$$\llbracket - \rrbracket : \mathcal{C}(\Lambda_{\beta+}) \rightarrow \mathbb{X}$$

is a Cartesian left-additive functor which preserve the Turing object, the Turing map, and its canonical codes.

The proof is by calculation and is relatively standard.

Again with more work this interpretation can be turned into an adjunction between additive λ -theories and left-additive Turing categories with additive canonical codes.

3.5 Additive reflexive objects

We now wish to split the idempotents of a left-additive Turing category with additive canonical codes. However, it is clear that we cannot split any old idempotent if we want to be able to induce an additive structure on the splitting. So clearly we should split the retractively additive idempotents. However, when we restrict the idempotents we split we have to ensure that all the structure we require, namely the Cartesian left additive structure, is still present.

Clearly all identity maps are additively retractive idempotents. Note that the product of retractively additive idempotents is retractively additive, and the requirements of being retractively additive on an idempotent is exactly what is required to obtain additive structure on each idempotent in the splitting.

Finally, note that the internal homsets of 2.6, $[u, v] = \lambda[(1 \times u) \bullet_{AB} v]$, are additively retractive.

A **Cartesian closed left-additive category** is a Cartesian left additive category which is closed and has every evaluation map additive in its first argument. As a Cartesian closed category always has canonical codes, this may equivalently be stated as the requirement that it has additive canonical codes. This gives:

Theorem 3.9 *If \mathbb{X} is a left-additive Turing category with additive canonical codes then splitting the additively retractive idempotents, $\text{Split}_+(\mathbb{X})$ yields a Cartesian closed left-additive Turing category.*

By construction all the additively retractive idempotents split with additive retractions; hence, the retraction $T \rightarrow [T, T]$ is additive. This immediately means that every model of the additive λ -calculus can be seen to arise as an additive reflexive object in a Cartesian closed left-additive category.

4 The Differential λ -calculus

A **Cartesian differential category** \mathbb{X} is a Cartesian left additive category with a combinator

$$\frac{A \xrightarrow{f} B}{A \times A \xrightarrow{D[f]} B}$$

that satisfies seven axioms:

$$[\text{CD.1}] \quad D[0] = 0 \text{ and } D[f + g] = D[f] + D[g]$$

$$[\text{CD.2}] \quad \langle 0, g \rangle D[f] = 0 \text{ and } \langle h + k, g \rangle D[f] = \langle h, g \rangle D[f] + \langle k, g \rangle D[f]$$

$$[\text{CD.3}] \quad D[1] = \pi_0, \quad D[\pi_0] = \pi_0 \pi_0, \text{ and } D[\pi_1] = \pi_0 \pi_1$$

$$[\text{CD.4}] \quad D[\langle f, g \rangle] = \langle D[f], D[g] \rangle$$

$$[\text{CD.5}] \quad D[fg] = \langle D[f], \pi_1 f \rangle D[g]$$

$$[\text{CD.6}] \quad \langle \langle g, 0 \rangle, \langle h, k \rangle \rangle D[D[f]] = \langle g, k \rangle D[f]$$

$$[\text{CD.7}] \quad \langle \langle p, h \rangle, \langle g, k \rangle \rangle D[D[f]] = \langle \langle p, g \rangle, \langle h, k \rangle \rangle D[D[f]]$$

In a Cartesian differential category a map f is **linear** when $D[f] = \pi_0 f$. Intuitively this means

$$\frac{df(x)}{dx}(x) \cdot v = f(v)$$

The following is corollary 2.2.3 in [3]

Proposition 4.1 *If a map is linear, then it is additive. Moreover, the class of linear maps form a commutative monoid enriched category with biproducts.*

$$\frac{\Gamma, p : S, \Gamma' \vdash m : U \quad \Gamma, \Gamma' \vdash a : S \quad \Gamma, \Gamma' \vdash v : S}{\Gamma, \Gamma' \vdash \frac{dm}{dp}(a) \cdot v : U}$$

Table 4
Term formation rules for the differential λ -calculus

The following lemma is quite useful in many calculations

Lemma 4.2 *Let h and k be linear and f any map, then*

$$D[hfk] = (h \times h)D[f]k.$$

In a Cartesian differential category, if $f : A \times B \rightarrow C$, we may define the **partial derivative** of f with respect to A as

$$D_{\times,0} := A \times (A \times B) \xrightarrow{\langle 1, 0 \rangle \times 1} (A \times B) \times (A \times B) \xrightarrow{D[f]} C$$

One may obtain the partial derivative of f with respect to B by

$$D_{\times,1} := B \times (A \times B) \xrightarrow{\langle 0, 1 \rangle \times 1} (A \times B) \times (A \times B) \xrightarrow{D[f]} C$$

We will also make use of maps that are **linear in the first argument**. This means

$$(v, (x, y)) \mapsto \frac{df(x, y)}{dx}(x) \cdot v = f(v, y)$$

i.e. that the partial derivative in the first argument is linear.

The following is useful in many calculations:

Lemma 4.3 *Suppose $h : A \times B \rightarrow C$ is linear in its first argument. Then $(1 \times g)h$ is linear in its first argument for any g .*

Intuitively this is so because g does not “touch” the first argument of h .

4.1 Syntax of the differential λ -calculus

The syntax of the untyped differential λ -calculus is again defined by untyped terms in context. Term formation uses the rules of Cartesian theories Table 1, the rules for the λ -calculus Table 2, the rules for the additive λ -calculus, Table 3, and a single new rule for the differential λ -calculus, Table 4. Note that the syntax we use here is slightly different from (but equivalent to) the syntax used by Ehrhard and Regnier in [12]⁸.

We have the following equations on terms

⁸ In Ehrhard and Regnier’s syntax, $D(m) \cdot v := \lambda a. \frac{dmz}{dz}(a) \cdot v$.

$$[\text{Dt.1}] \quad \frac{dm_1+m_2}{dp} (a) \cdot v = \frac{dm_1}{dp} (a) \cdot v + \frac{dm_2}{dp} (a) \cdot v \quad \frac{d0}{dp} (a) \cdot v = 0$$

$$[\text{Dt.2}] \quad \frac{dm}{dx} (a) \cdot (v_1 + v_2) = \frac{dm}{dx} (a) \cdot v_1 + \frac{dm}{dx} (a) \cdot v_2 \quad \frac{dm}{dx} (a) \cdot 0 = 0$$

$$[\text{Dt.3}] \quad \bullet \quad \frac{dx}{dp} (a) \cdot v = v,$$

$$\bullet \quad \frac{dt}{d(p,p')} ((a, a')) \cdot (v, v') = \frac{dt[a'/p']}{dp} (a) \cdot v + \frac{dt[a/p]}{dp'} (a') \cdot v'$$

$$[\text{Dt.4}] \quad \frac{dm[t/q]}{dp} (a) \cdot v = \frac{dm}{dq} (t[a/p]) \cdot \frac{dt}{dp} (a) \cdot v$$

Note in the above, by the context-formation rules, $p \notin \text{fv}(m)$.

$$[\text{Dt.5}] \quad \frac{d \frac{dm}{dp} (a) \cdot q}{dq} (b) \cdot v = \frac{dm}{dp} (a) \cdot v$$

$$[\text{Dt.6}] \quad \frac{d \frac{dm}{dp_1} (a_1) \cdot v_1}{dp_2} (a_2) \cdot v_2 = \frac{d \frac{dm}{dp_2} (a_2) \cdot v_2}{dp_1} (a_1) \cdot v_1$$

$$[\text{Dt.7}] \quad \frac{d\lambda y.m}{dp} (a) \cdot v = \lambda y. \frac{dm}{dp} (a) \cdot v$$

By the context formation rules, $x \notin (a, v)$

$$[\text{Dt.8}] \quad \frac{d\lambda z.yz}{dy} (a) \cdot v = \lambda z.vz$$

Note that when we write $\frac{dt}{dp} (a) \cdot v$ in [Dt.4] the t could be a tuple of terms. However, this is only a notation for distributing the derivative down onto term of type U as only these have derivatives. This means $\frac{d()}{dp} (a) \cdot v = ()$ and $\frac{d(t_1, t_2)}{dp} (a) \cdot v = (\frac{dt_1}{dp} (a) \cdot v, \frac{dt_2}{dp} (a) \cdot v)$.

Lemma 4.4 In $\Lambda_{\beta\partial}$:

(i) If $y \notin \text{fv}(m)$ then $\frac{dm}{dy} (a) \cdot v = 0$;

(ii) $\frac{dm}{d(p,p')} ((a, a')) \cdot (v, 0) = \frac{dm[a'/p']}{dp} (a) \cdot v$;

(iii) $\frac{dm}{d(p,p')} ((s, s')) \cdot (0, v') = \frac{dm[a/p]}{dp'} (a') \cdot v'$;

(iv) If $y \notin \text{fv}(m)$ then $\frac{dy m}{dy} (a) \cdot v = vm$;

(v) $\frac{dm n}{dx} (a) \cdot v = (\frac{dm}{dx} (a) \cdot v) (n[a/x]) + \frac{d(m[a/x])}{dz_1} z_1 (n[a/x]) \cdot \frac{dn}{dx} (a) \cdot v$.

The proof is straightforward; the first three are from [3].

There is an associated category $\mathcal{C}(\Lambda_{\beta\partial})$. We have

Proposition 4.5 $\mathcal{C}(\Lambda_{\beta\partial})$ is a Cartesian category in which $(U, +_U, 0_U)$ is a commutative monoid.

4.2 Differential universal objects

In subsection 3.2, we introduced the notion of an additive universal object that induces an additive structure on each object. In a Cartesian differential category, we must again strengthen the notion of universal object, so that the derivative on maps $U \rightarrow U$ induces a derivative on all maps. An object U in a Cartesian differential category \mathbb{X} is a **differential universal object** in case every object A is a retract of U , $A \triangleleft_{r_A^A}^{s_A} U$, and the retraction is linear.

Lemma 4.6 In any Cartesian differential category with a retract $A \triangleleft_{r_A^A}^{s_A} U$ in which r_A is linear

- (i) For any $A \xrightarrow{f} B$, $D[f] = (s_A \times s_A)D[r_A f s_B]r_B$;
(ii) $e_A = r_A s_A$ satisfies $(e_A \times e_A)D[e_A] = D[e_A]$ and $D[e_A]e_A = \pi_0 e_A$

The proof makes repeated use of 4.2.

When an idempotent satisfies part (ii) of the above and is retractively additive, we will say that it is a **retractively linear idempotent**.

We now extend the extend the notion of additively coherent universal structure. In a Cartesian category with chosen products, where $(U, +_U, 0_U)$ is a commutative monoid, a universal structure \mathcal{U} for U is **differentially coherent** if it is additively coherent and additionally:

[UDC.1] There is a differential operator for U ; i.e.

$$\frac{U \xrightarrow{f} U}{U \times U \xrightarrow{D[f]} U}$$

that satisfies [CD.1,2,3,5].

[UDC.2] Each idempotent $e_A = r_A s_A$ of $\mathcal{U}(A)$ is retractively U -linear in the sense that

$$(e_A \times e_A)D[e_A] = D[e_A] \quad \text{and} \quad D[e_A]e_A = \pi_0 e_A$$

[UDC.3] For $\mathcal{U}(U \times U) = U \times U \triangleleft_{r_{U \times U}}^{s_{U \times U}}$, we have that

$$\begin{aligned} (s_{U \times U} \times s_{U \times U})D[r_{U \times U} \pi_0] &= \pi_0 \pi_0 \\ (s_{U \times U} \times s_{U \times U})D[r_{U \times U} \pi_1] &= \pi_0 \pi_1 \end{aligned}$$

And that for any $f, g : U \rightarrow U$:

$$D[\langle f, g \rangle s_{U \times U}]r_{U \times U} = \langle D[f], D[g] \rangle$$

[UDC.4] Let $U \xrightarrow{f} U$. The map

$$D^2[f] := (U \times U) \times (U \times U) \xrightarrow{s_{U \times U} \times s_{U \times U}} U \times U \xrightarrow{D[r_{U \times U} D[f]]} U$$

satisfies

$$\begin{aligned} \langle \langle g, 0 \rangle, \langle h, k \rangle \rangle D^2[f] &= \langle g, k \rangle D[f] \\ \langle \langle 0, h \rangle, \langle g, k \rangle \rangle D^2[f] &= \langle \langle 0, g \rangle, \langle h, k \rangle \rangle D^2[f] \\ ((e_A \times e_A) \times (e_A \times e_A))D^2[f] &= D^2[f] \end{aligned}$$

Given a differentially coherent universal structure on a category, one can generate a Cartesian differential category.

Proposition 4.7 *If \mathbb{X} is a category with chosen products and a commutative monoid $(U, +_U, 0_U)$ which has a differentially coherent universal structure, \mathcal{U} , then there is a Cartesian differential structure on \mathbb{X} that makes each retraction in \mathcal{U} linear.*

We first show that \mathbb{X} has a Cartesian differential structure.

Note the following facts about \mathcal{U} idempotents.

- For all $U \xrightarrow{f} U$, $D[fe_B]r_B = D[f]r_B$
- For all $A \xrightarrow{f} B$, $(e_A \times e_A)D[r_Afs_B] = D[r_Afs_B]$

The differential operator $\mathcal{D}[_]$ is defined on maps $A \rightarrow B$:

$$\frac{A \xrightarrow{f} B}{A \times A \xrightarrow{\mathcal{D}[f]} B} = \frac{\frac{U \xrightarrow{r_A} A \xrightarrow{f} B \xrightarrow{s_B} U}{U \times U \xrightarrow{D[r_Afs_B]} U}}{A \times A \xrightarrow{s_A \times s_A} U \times U \xrightarrow{D[r_Afs_B]} U \xrightarrow{r_B} B}$$

4.3 Differential Turing categories

A **differential Turing category** is a Turing category which is also a Cartesian differential category and where additionally each universal application $T \times B \xrightarrow{\bullet_{BC}} C$ is linear in its first argument.

The above definition has the property that if c_f is a code for f , then

$$\frac{df(x, y)}{dx}(a) \cdot v = \frac{dc_f(x) y}{dx}(a) \cdot v = \left(\frac{dc_f(x)}{dx}(a) \cdot v \right) y$$

so that the derivative of f in its first argument is given by the derivative of c_f .

A differential Turing category \mathbb{X} has **differential canonical codes** when \mathbb{X} has additive canonical codes $\lambda(_)$ that satisfy in addition:

$$D[\lambda(f)] = \lambda(\langle \pi_0 \times 0, \pi_1 \times 1 \rangle D[f])$$

The following is a recognition theorem for differential Turing categories.

Proposition 4.8 *A Cartesian differential category \mathbb{X} is a differential Turing category if and only if*

- \mathbb{X} has a differentially universal object T ;
- There is a map $T \times T \xrightarrow{\bullet} T$ that is universal for maps $T \times T \rightarrow T$ and that is linear in its first argument.

Furthermore, \mathbb{X} is a differential Turing category with differential canonical codes if and only if (i) and (ii) hold and $T \times T \xrightarrow{\bullet} T$ has differential canonical codes.

Proof. The usual retraction is linear; this follows as $T \times 1 \xrightarrow{\bullet} T$ is linear in its first argument, and ignores its second.

To show that we can construct differentially canonical codes for $A \times B \rightarrow C$ from canonical codes for $T \times T \rightarrow T$, consider the diagram that codes $\langle \pi_0 \times 0, \pi_1 \times 1 \rangle D[f]$ and use that r is linear, so that one takes the canonical code for $(r \times r) \langle \pi_0 \times 0, \pi_1 \times 1 \rangle (s \times s) D[rfs]$. \square

For completeness we show that $\mathcal{C}(\Lambda_{\beta\partial})$ is a differential Turing category with differentially canonical codes. We will first show that $\mathcal{C}(\Lambda_{\beta\partial})$ is a Cartesian differential category in which U is a differentially universal object using proposition 4.7.

The differential structure on $\mathcal{C}(\Lambda_{\beta\partial})$ is defined as follows:

$$\begin{aligned} D[p : S \vdash m : 1] &:= () \\ D[p : S \vdash m : U] &:= (v, p) : S \times S \vdash \frac{dm}{dp}(p) \cdot v : U \\ D[p : S \vdash (m_1, m_2) : R \times T] &:= \langle D[p : S \vdash m_1 : R], D[p : S \vdash m_2 : T] \rangle \end{aligned}$$

Proposition 4.9 *$\mathcal{C}(\Lambda_{\beta\partial})$ is a Cartesian differential category in which U is a differentially universal object.*

This proposition uses the fact that Curry’s retraction is linear, and that this is enough to determine a differential universal structure on U .

This leads to:

Theorem 4.10 *$\mathcal{C}(\Lambda_{\beta\partial})$ is a differential Turing category with differentially canonical codes.*

The proof involves a relatively straightforward sequence of calculations which show that codes are differentially canonical.

4.4 Interpreting the differential λ -calculus

Let \mathbb{X} be a differential Turing category with differential canonical codes. We will show that there is a functor $\mathcal{C}(\Lambda_{\beta\partial}) \rightarrow \mathbb{X}$ that preserves all the differential structure and the canonical differential Turing structure. The functor $\llbracket - \rrbracket : \mathcal{C}(\Lambda_{\beta\partial}) \rightarrow \mathbb{X}$ is defined in the same way as before for variables, applications, abstractions, sums, 0, and tuples of terms. For the differentials:

$$\left\llbracket q \vdash \frac{dm}{dp}(a) \cdot v \right\rrbracket := \langle \langle \llbracket q \vdash u \rrbracket, 0 \rangle, \langle \llbracket q \vdash a \rrbracket, 1 \rangle \rangle D[\llbracket (p, q) \vdash t \rrbracket]$$

To show that $\llbracket - \rrbracket$ is a functor a crucial step is to establish:

Lemma 4.11 (Substitution lemma) *For $\llbracket - \rrbracket : \mathcal{C}(\Lambda_{\beta\partial}) \rightarrow \mathbb{X}$ where \mathbb{X} is a differential Turing category with differential canonical codes,*

$$\llbracket p \vdash m[n/q] \rrbracket = \llbracket p \vdash n \rrbracket \llbracket q \vdash m \rrbracket$$

This then allows:

Proposition 4.12 *When \mathbb{X} is a differential Turing category with differentially canonical codes, then*

$$\llbracket - \rrbracket : \mathcal{C}(\Lambda_{\beta\partial}) \rightarrow \mathbb{X}$$

is a Cartesian differential functor.

4.5 Differential reflexive objects

To obtain the analog of the Scott-Koymans theorem we split the retractively linear idempotents in a differential Turing category with differential canonical codes to obtain a closed differential Turing category. As a first step, note:

Lemma 4.13 *In any Cartesian differential category, the class \mathcal{E} of retractively linear idempotents is closed to identities and products. Furthermore, retractively linear idempotents are retractively additive.*

The above lemma ensures that $\mathbf{Split}_{\mathcal{E}}(\mathbb{X})$ is a Cartesian left additive category. The following proposition shows that if we split retractively linear idempotents, we can lift the differential structure from a Cartesian differential category to this idempotent splitting.

Proposition 4.14 *Let \mathbb{X} be a Cartesian differential category, and \mathcal{E} the class of retractively linear idempotents. Then there is a unique differential structure on $\mathbf{Split}_{\mathcal{E}}(\mathbb{X})$ in which all $e \in \mathcal{E}$ split with linear retraction and for which the inclusion $\mathbb{X} \hookrightarrow \mathbf{Split}_{\mathcal{E}}(\mathbb{X})$ is a Cartesian differential functor.*

Proof. From 3.9, we know that as \mathcal{E} is a product closed of retractively additive idempotents that contains the identities, that $\mathbf{Split}_{\mathcal{E}}(\mathbb{X})$ is a Cartesian left additive category.

In order for $\mathbb{X} \hookrightarrow \mathbf{Split}_{\mathcal{E}}(\mathbb{X})$ to be a Cartesian differential functor, the differential on maps between identities $f : 1_A \rightarrow 1_B$ is forced to be the differential from \mathbb{X} .

If each idempotent is to split with a linear retraction, the derivative must satisfy $(s_A \times s_A)D[r_A f s_B]r_B = D[f]$.

This means the derivative for a $e \xrightarrow{f} e'$ is:

$$D[f] := (e_A \times e_A)D[e_A f e_B]e_B = (e_A \times e_A)D[f]e_B = D[f]e_B$$

The details of the proof that this does indeed give a differential structure on $\mathbf{Split}_{\mathcal{E}}(\mathbb{X})$ are relatively straightforward; the proof will be somewhat similar to the proof of proposition 4.7. \square

Next, we show that differential Turing structure lifts to the idempotent splitting.

Proposition 4.15 *When \mathbb{X} is a differential Turing category (with Turing object T), then so is $\mathbf{Split}_{\mathcal{E}}(\mathbb{X})$ where \mathcal{E} is the class of retractively linear idempotents.*

Proof. We have already seen that $\mathbf{Split}_{\mathcal{E}}(\mathbb{X})$ is both a Cartesian differential category and a Turing category with Turing object 1_T .

Each idempotent e is also a retract of the Turing object 1_T .

Hence, using Proposition 4.8, it suffices to show that the Turing morphism

$$1_T \times 1_T \xrightarrow{\bullet} 1_T$$

is linear in its first argument, which is immediate. \square

Finally, we show that if \mathcal{E} is the collection of retractively linear idempotents in a differential Turing category with canonical codes, then $\mathbf{Split}_{\mathcal{E}}(\mathbb{X})$ is a Cartesian closed differential Turing category. We do not get for free that $\mathbf{Split}_{\mathcal{E}}(\mathbb{X})$ is still a Cartesian closed category: we must show that \mathcal{E} is closed to forming internal homs. Once we have established this, we will have that $\mathbf{Split}_{\mathcal{E}}(\mathbb{X})$ is a Cartesian closed category that is also a differential Turing category. Moreover, it is then immediate that the coherence for *Cartesian closed differential categories*

$$D[\lambda(f)] = \lambda(\langle \pi_0 \times 0, \pi_1 \times 1 \rangle D[f])$$

holds because $\lambda(f)$ in $\mathbf{Split}_{\mathcal{E}}(\mathbb{X})$ is the canonical code $\lambda(f)$ from \mathbb{X} . Thus, the proof of the following theorem requires only that we show retractively linear maps are closed to forming internal homs.

We make use of lemma 4.3. Also, recall that if h is linear in its first argument, then

$$\langle \pi_0 \times 0, \pi_1 \times 1 \rangle D[h] = a_{\times}(\langle 1, 0 \rangle \times 1) D[h] = a_{\times}(1 \times \pi_1) h = (\pi_0 \times 1) h$$

A few long calculations provide the proof of:

Theorem 4.16 *When \mathcal{E} is the collection of retractively linear idempotents in a differential Turing category with canonical codes, then $\mathbf{Split}_{\mathcal{E}}(\mathbb{X})$ is a Cartesian closed differential Turing category.*

This allows us to conclude the Scott-Koymans theorem for the differential λ -calculus.

Corollary 4.17 *When \mathcal{E} is the class of retractively linear maps of a differential Turing category with canonical codes, the Turing object 1_T is a reflexive object in $\mathbf{Split}_{\mathcal{E}}(\mathbb{X})$ and the retraction is $1_T \rightarrow [1_T, 1_T]$ is a linear map.*

Thus, every model of the differential λ -calculus may be seen to arise as a differential reflexive object in a Cartesian closed differential category.

References

- [1] Birkedal, L., “Developing Theories of Types via Computability and Realizability,” Ph.D. thesis, Carnegie Mellon University (1999).
- [2] Blute, R., J. Cockett and R. Seely, *Differential categories*, in: *Mathematical Structures in Computer Science*, 2006.
- [3] Blute, R., J. Cockett and R. Seely, *Cartesian Differential Categories*, Theory and Application of Categories **22** (2009), pp. 622–672.
- [4] Blute, R., J. Cockett and R. Seely, *Cartesian differential storage categories*, Theory and Application of Categories (2015).
- [5] Boudol, G., *The Lambda-Calculus with Multiplicities*, in: *CONCUR '93 Proceedings of the 4th International Conference on Concurrency Theory*, 1993.
- [6] Boudol, G., P. Curien and C. Lavatelli, *A semantics for lambda calculi with resource*, Mathematical Structures in Computer Science (1999).

- [7] Boudol, G. and C. Laneve, *Lambda-Calculus, Multiplicities, and the π -Calculus*, Technical report, Institut National de Recherche en Informatique et en Automatique (1995).
- [8] Bucciarelli, A., T. Ehrhard and G. Manzonetto, *Categorical Models for Simply Typed Resource Calculi*, *Electronic Notes in Theoretical Computer Science* **265** (2010), pp. 213–230.
- [9] Cockett, J. and P. Hofstra, *Introduction to Turing Categories.*, *Annals of Pure and Applied Logic*. **156**. (2008.), pp. 183–209.
- [10] Ehrhard, T., *On Köthe Sequence Spaces and Linear Logic*, in: *Mathematical Structures in Computer Science*, 2002.
- [11] Ehrhard, T., *Finiteness spaces*, *Mathematical Structures in Computer Science* (2005).
- [12] Ehrhard, T. and L. Regnier, *The Differential Lambda Calculus*, *Theoretical Computer Science* **309** (2003), pp. 1–41.
- [13] Fiore, M., *Differential structure in models of multiplicative biadditive intuitionistic linear logic*, *TLCA* (2007).
- [14] Koymans, C., *Models of lambda calculus*, *Information and Control* (1982).
- [15] Lambek, J. and P. Scott, “An introduction to higher order categorical logic,” Cambridge University Press, 1986.
- [16] Longo, G. and E. Moggi, *A category theoretic characterization of functional completeness*, *Theoretical Computer Science* (1990).
- [17] Manzonetto, G., *What is a Categorical Model of the Differential and the Resource Lambda Calculi?*, *Mathematical Structures in Computer Science* **22** (2012), pp. 451–520.
- [18] Scott, D., *Relating theories of the λ -calculus*, *Essays on Combinatory Logic, Lambda Calculus, and Formalism* (1980).