Reverse Derivative Categories*

Robin Cockett
University of Calgary,
Department of Computer Science, Canada
robin@ucalgary.ca

Geoffrey Cruttwell
Mount Allison University, Department of
Mathematics and Computer Science, Canada
gcruttwell@mta.ca

Jonathan Gallagher
Dalhousie University, Department of
Mathematics and Statistics, Canada
jonathan.gallagher@dal.ca

Jean-Simon Pacaud Lemay
University of Oxford,
Department of Computer Science, UK
jean-simon.lemay@kellogg.ox.ac.uk

Benjamin MacAdam
University of Calgary,
Department of Computer Science, Canada
benjamin.macadam@ucalgary.ca

Gordon Plotkin
Google Research
gdp@inf.ed.ac.uk

Dorette Pronk
Dalhousie University, Department of
Mathematics and Statistics, Canada
dorette.pronk@dal.ca

Abstract

The reverse derivative is a fundamental operation in machine learning and automatic differentiation [1, 12]. This paper gives a direct axiomatization of a category with a reverse derivative operation, in a similar style to that given by [2] for a forward derivative. Intriguingly, a category with a reverse derivative also has a forward derivative, but the converse is not true. In fact, we show explicitly what a forward derivative is missing: a reverse derivative is equivalent to a forward derivative with a dagger structure on its subcategory of linear maps. Furthermore, we show that these linear maps form an additively enriched category with dagger biproducts.

2012 ACM Subject Classification Theory of computation → Semantics and reasoning; Theory of computation → Program semantics

Keywords and phrases Reverse Derivatives, Cartesian Reverse Differential Categories, Categorical Semantics, Cartesian Differential Categories, Dagger Categories, Automatic Differentiation

Digital Object Identifier 10.4230/LIPIcs.CSL.2020.18

Related Version A full version of the paper is available at `https://arxiv.org/abs/1910.07065`.

Funding *Robin Cockett*: Partially supported by NSERC (Canada).
Geoffrey Cruttwell: Partially supported by NSERC (Canada).
Jonathan Gallagher: Supported in part by NSERC and AARMS (Canada).
Jean-Simon Pacaud Lemay: Supported by Kellogg College, the Clarendon Fund, and the Oxford Google-DeepMind Graduate Scholarship (UK).
Benjamin MacAdam: Partially supported by NSERC (Canada).
Gordon Plotkin: Supported by ESPRC (UK).
Dorette Pronk: Partially supported by NSERC (Canada).

Acknowledgements We thank Robert Seely for participating in the discussion on reverse differentiation with us.

[*] This paper is the result of a joint working session that the authors participated in at the Foundational Methods in Computer Science workshop in June, 2019.

28th EACSL Annual Conference on Computer Science Logic (CSL 2020).
Editors: Maribel Fernández and Anca Muscholl; Article No. 18; pp. 18:1–18:16
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

The use of derivatives and differentiation in programming and machine learning is becoming ubiquitous. As a result, there has been an increased interest in axiomatic setups for differentiation; in particular, categorical models for differentiation have become more central. There are two types of derivative operations used in programming: the *forward* derivative and the *reverse* derivative. From the programmer's perspective, it is much more common for the reverse derivative to play the central role due to its increased efficiency and improved accuracy when computing with functions from $\mathbb{R}^n$ to $\mathbb{R}$ (due to the so called cheap gradient principle). The importance of this principle was already recognized by Linnainmaa in 1976 [16] and was specifically used for back-propagation in multi-layer networks and deep learning. This was further spelled out in detail in [18]. Also, Tensorflow, Google's new interface for expressing machine learning algorithms, uses the reverse mode of automatic differentiation as the basic building block minimizing cost functions [1].

The categorical approaches to differentiation to date have all exclusively focused on the abstract properties of the forward derivative [2]. This thus leaves a significant gap which needs to be filled: an axiomatic categorical setting for reverse differentiation. The main goal of this paper is to introduce such a structure and explore some of its properties and consequences.

A "Cartesian reverse differential category" (a category equipped with a reverse derivative operation as introduced in this paper) is already a Cartesian differential category (the standard axiomatics for a category with a forward derivative). We show that a category equipped with a reverse derivative also has a forward derivative (i.e., it has a Cartesian differential structure). Moreover, a reverse differential category has a fibered dagger structure on its subcategory of linear maps, a structure which does not automatically exist in a Cartesian differential category. Suitably axiomatized, we show that having such a dagger structure is enough to ensure that a Cartesian differential category structure gives a reverse differential category. These results provide a starting point to build categorical semantics of differential programming languages [17], as they provide axiomatically enough structure to handle both forward and reverse derivatives.

The paper is structured as follows. In section 2, we recall the basic notation and definitions of a Cartesian differential category ("a category equipped with a forward derivative"). We do this first to acclimatize the reader to the general style of this categorical definition, and to recall the structure of Cartesian left additive categories, which are necessary to define both forward and reverse differential categories. In section 3, we introduce our definition of a reverse differential category. We explore some of the important consequences of the definition noted above: (a) Cartesian differential structure, (b) how to define and work with linear maps in this setting, and (c) a dagger structure on the linear maps. In section 4, we show how to go back: given a Cartesian differential category with a "contextual dagger", we build a Cartesian reverse differential category. There is much more work to be done with this structure and these ideas: in section 5, we describe some of the ways in which this work can be extended, including allowing partial functions.

As far as we are aware, this paper represents the first categorical axiomatization of the reverse derivative. However, [11] does have some related ideas. There, the relationship between the reverse derivative and coproducts was noticed, and the author specified an internal category which satisfies some of the axioms of a Cartesian differential category in a functional programming language. This work expands that observation by developing the dagger biproduct structure using the reverse derivative and relating this to the dual of the simple slice fibration.

Full proofs of the theorems in this paper can be found in an expanded version [10], and we have indicated in this version, the corresponding theorem number in the longer version.

2 Forward derivatives

The standard setting for a "category with a forward derivative" is a Cartesian differential category, first introduced in [2]. Following that paper, we write composition in diagrammatic order, so that f, followed by g, is written as fg.

2.1 Cartesian left additive categories

A Cartesian differential category first consists of a Cartesian left additive category, and so we begin by recalling this notion. Recall that a category $\mathbb{X}$ is said to be **Cartesian** when there are chosen binary products $\times$, with projection maps π_i and pairing operation $\langle -, - \rangle$, and a chosen terminal object $\mathbf{1}$, with unique maps $!$ to the terminal object.

▶ **Definition 1.** *A* ***left additive category*** *[2, Definition 1.1.1] is a category $\mathbb{X}$ such that each hom-set is a commutative monoid, with addition operation $+$ and zero maps 0, such that composition on the left preserves the additives structure in the sense that $x(f+g) = xf + xg$ and $x0 = 0$. Maps h which preserve the additive structure by composition on the right ($(x+y)h = xh+yh$ and $0h = 0$) are called* ***additive****. A* ***Cartesian left additive category*** *[2, Definition 1.2.1] is a left additive category $\mathbb{X}$ which is Cartesian and such that all projection maps π_i are additive*[1]*.*

Cartesian left additive categories can alternatively be defined as Cartesian categories in which each object A canonically bears the structure of a commutative monoid with addition $+_A : A \times A \longrightarrow A$ and zero $0_A : \mathbf{1} \longrightarrow A$.

▶ **Example 2.** Here are examples of Cartesian left additive categories that we will consider throughout this paper:

1. Any category with finite biproducts is a Cartesian left additive category where every map is additive. And conversely, in a Cartesian left additive category where every map is additive, the finite product is a finite biproduct [2, Proposition 1.2.2].
2. Let R be a commutive rig (also known as a commutative semiring). Let POLY_R be the category of polynomials with coefficients in R; that is, the category whose objects are the natural numbers $n \in \mathbb{N}$ and where a map $n \xrightarrow{P} m$ is an m-tuple of polynomials $P := \langle p_1(\vec{x}), \dots, p_m(\vec{x}) \rangle$, where $p_i(\vec{x}) \in R[x_1, \dots, x_n]$ (the polynomial ring in n-variables over R). POLY_R is a Cartesian left additive category where composition is given by the standard composition of polynomials, the product on objects is given by the sum of natural numbers, and the additive structure is given by the sum of polynomials.
3. Let $\mathbb{R}$ be the set of real numbers and let Smooth be the category of smooth real functions, that is, the category whose objects are again the natural numbers $n \in \mathbb{N}$ and where a map $n \xrightarrow{F} m$ is a smooth function $\mathbb{R}^n \xrightarrow{F} \mathbb{R}^m$. Smooth is a Cartesian left additive category where composition is given by the standard composition of smooth functions, the product on objects is given by the sum of natural numbers, and the additive structure is given by the sum of smooth functions. Note that a smooth map $\mathbb{R}^n \xrightarrow{F} \mathbb{R}^m$ is actually an m-tuple of smooth functions $F = \langle f_1, \dots, f_m \rangle$, where $\mathbb{R}^n \xrightarrow{f_i} \mathbb{R}$ and therefore $\mathsf{POLY}_{\mathbb{R}}$ is a sub-Cartesian left additive category of Smooth.

[1] Note that this a slight variation on the definition of a Cartesian left additive category found in [2], but it is indeed equivalent.

As not every map in a Cartesian left additive category is additive, the product × is not a coproduct, thus is *not* a biproduct. However, it is still possible to define injection maps. So in a Cartesian left additive category, define $\iota_0 := \langle 1, 0 \rangle : A \longrightarrow A \times B$ and $\iota_1 := \langle 0, 1 \rangle : B \longrightarrow A \times B$. For maps $A \xrightarrow{f} C$ and $B \xrightarrow{g} C$, we define $\langle f | g \rangle := \pi_0 f + \pi_1 g : A \times B \longrightarrow C$, and finally for maps $A \xrightarrow{h} B$ and $C \xrightarrow{k} D$ we write $h \oplus k := \langle h\iota_0 | k\iota_1 \rangle : A \times B \longrightarrow C \times D$. Although this notation is suggestive, we again stress this is not part of a coproduct or biproduct structure. However, in what follows we will define the category of linear maps where the above will witness a biproduct structure on that category. We leave the following lemma as an easy exercise to the reader:

▶ **Lemma 3.** *In a Cartesian left additive category, $f\iota_0 + g\iota_1 = \langle f, g \rangle$ and $h \oplus k = h \times k$.*

2.2 Cartesian differential categories

This section reviews Cartesian differential categories which provide the semantics for forward differentiation [2].

▶ **Definition 4.** *A* ***Cartesian differential category*** *[2] is a Cartesian left additive category with a combinator* D*, called the* ***differential combinator****, which written as an inference rule is given by:*

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

where $\mathsf{D}[f]$ is called the derivative of f, and such that the following equalities hold[2]*:*

[CDC.1] $\mathsf{D}[f+g] = \mathsf{D}[f] + \mathsf{D}[g]$ *and* $\mathsf{D}[0] = 0$*;*
[CDC.2] $\langle a, b+c \rangle \mathsf{D}[f] = \langle a, b \rangle \mathsf{D}[f] + \langle a, c \rangle \mathsf{D}[f]$ *and* $\langle a, 0 \rangle \mathsf{D}[f] = 0$*;*
[CDC.3] $\mathsf{D}[1] = \pi_1$*,* $\mathsf{D}[\pi_0] = \pi_1\pi_0$*, and* $\mathsf{D}[\pi_1] = \pi_1\pi_1$*;*
[CDC.4] $\mathsf{D}[\langle f, g \rangle] = \langle \mathsf{D}[f], \mathsf{D}[g] \rangle$*;*
[CDC.5] $\mathsf{D}[fg] = \langle \pi_0 f, \mathsf{D}[f] \rangle \mathsf{D}[g]$*;*
[CDC.6] $\langle \langle a, b \rangle, \langle 0, c \rangle \rangle \mathsf{D}[\mathsf{D}[f]] = \langle a, c \rangle \mathsf{D}[f]$*;*
[CDC.7] $\langle \langle a, b \rangle, \langle c, d \rangle \rangle \mathsf{D}[\mathsf{D}[f]] = \langle \langle a, c \rangle, \langle b, d \rangle \rangle \mathsf{D}[\mathsf{D}[f]]$.

For an in-depth commentary on these axioms, we invite the reader to see the original Cartesian differential category paper [2]. Briefly, **[CDC.1]** is that the derivative of a sum is the sum of the derivatives, **[CDC.2]** states that derivatives are additive in their second argument, **[CDC.3]** says that the identity and projection maps are linear (more on what this means soon), **[CDC.4]** is that the derivative of a pairing is the pairing of the derivatives, **[CDC.5]** is the famous chain rule, **[CDC.6]** says that the derivative is linear in its second argument, and finally **[CDC.7]** is the symmetry of the mixed partial derivatives.

▶ **Example 5.** Here are some well-known examples of Cartesian differential categories.

1. Every category with finite biproducts is a Cartesian differential category where for a map $A \xrightarrow{f} B$, its derivative $A \oplus A \xrightarrow{\mathsf{D}[f]} B$ is defined as $\mathsf{D}[f] := A \oplus A \xrightarrow{\pi_1} A \xrightarrow{f} B$.

[2] Note that the order of variables is different here than in [2]; here, we write the vector variable in the second component, as this more closely aligns with standard differential calculus notation.

2. Let R be a commutative rig. POLY_R is a Cartesian differential category whose differential combinator is given by the standard differentiation of polynomials. By **[CDC.4]**, since every map in POLY_R is a tuple, it is sufficient to define the derivative of maps $n \xrightarrow{p} 1$, which are polynomials $p(\vec{x}) \in R[x_1, \dots, x_n]$. Then its derivative $n \times n \xrightarrow{\mathsf{D}[p]} 1$, viewed as polynomials $\mathsf{D}[p](\vec{x}, \vec{y}) \in R[x_1, \dots, x_n, y_1, \dots, y_n]$, is defined by the sum of partial derivatives of $p(\vec{x})$:
$$\mathsf{D}[p](\vec{x}, \vec{y}) := \sum_{i=1}^{n} \frac{\partial p}{\partial x_i}(\vec{x}) y_i$$
For example, consider the polynomial $p(x_1, x_2) = x_1^2 + 3x_1x_2 + 5x_2$, so $2 \xrightarrow{p} 1$, then $4 \xrightarrow{\mathsf{D}[p]} 1$ is $\mathsf{D}[p](x_1, x_2, y_1, y_2) = (2x_1 + 3x_2)y_1 + (3x_1 + 5)y_2$. On the other hand, for a map $n \xrightarrow{P} m$, which is a tuple $P := \langle p_1(\vec{x}), \dots, p_m(\vec{x}) \rangle$, its derivative is the tuple $\mathsf{D}[P] := \langle \mathsf{D}[p_1](\vec{x}, \vec{y}), \dots, \mathsf{D}[p_m](\vec{x}, \vec{y}) \rangle$.
3. The category Smooth is a Cartesian differential category where for a map $n \xrightarrow{F} m$, which is a smooth function $\mathbb{R}^n \xrightarrow{F} \mathbb{R}^m$, its derivative $\mathbb{R}^n \times \mathbb{R}^n \xrightarrow{\mathsf{D}[F]} \mathbb{R}^m$ is defined as
$$\mathsf{D}[F](\vec{x}, \vec{v}) := J_F(\vec{x}) \cdot v$$
where $J_F(\vec{x})$ is the Jacobian of F at $\vec{x}$ and where $\cdot$ is matrix multiplication. Of course, similar to the previous example, as every F can be viewed as a tuple, by **[CDC.4]**, it would have also been sufficient to define the differential combinator for smooth maps $\mathbb{R}^n \xrightarrow{f} \mathbb{R}$. In this case, $J_f(x)$ is better known as the gradient of f, $\nabla(f)(x) := \langle \frac{\partial f}{\partial x_1}(\vec{x}), \dots, \frac{\partial f}{\partial x_n}(\vec{x}) \rangle$, and so $\mathbb{R}^n \times \mathbb{R}^n \xrightarrow{\mathsf{D}[F]} \mathbb{R}$ is:
$$\mathsf{D}[f](\vec{x}, \vec{y}) := \nabla(f)(\vec{x}) \cdot \vec{y} = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i}(\vec{x}) y_i$$
This clearly shows that $\mathsf{POLY}_{\mathbb{R}}$ is a sub-Cartesian differential category of Smooth.

We now provide a few lemmas that give alternative views on the axioms of a Cartesian differential category; these will be helpful when comparing this structure to a reverse differential category. Note that while the first lemma shows that **[CDC.4]** is actually redundant, to keep the numbering of the equations consistent with past literature on Cartesian differential categories, we chose to include it in the definition.

▶ **Lemma 6.** *[15, Lemma 2.8] In a Cartesian differential category, **[CDC.4]** is redundant.*

▶ **Lemma 7.** *[9, Proposition 4.2] In a Cartesian left additive category:*

1. *If a combinator D satisfies **[CDC.1-5,7]**, the axiom **[CDC.6]** is equivalent to:*
$$\langle 1 \times \pi_0, 0 \times \pi_1 \rangle \, \mathsf{D}[\mathsf{D}[f]] = (1 \times \pi_1)\mathsf{D}[f]$$
2. *If a combinator D satisfies **[CDC.1-6]**, the axiom **[CDC.7]** is equivalent to*
$$\mathsf{ex}\mathsf{D}[\mathsf{D}[f]] = \mathsf{D}[\mathsf{D}[f]]$$
where $\mathsf{ex} : (A \times B) \times (C \times D) \longrightarrow (A \times C) \times (B \times D)$ *is the **exchange** natural isomorphism defined as* $\mathsf{ex} := \langle \pi_0 \times \pi_0, \pi_1 \times \pi_1 \rangle$.

In a Cartesian differential category, there are two important notions: that of partial derivatives and that of linear maps. Beginning with partial derivatives, if $A \times B \xrightarrow{f} C$ then the partial derivative of f with respect to B is defined as follows:

$$D_B[f] := A \times (B \times B) \xrightarrow{\langle 1 \times \pi_0, 0 \times \pi_1 \rangle} (A \times B) \times (A \times B) \xrightarrow{\mathsf{D}[f]} C$$

This partial derivative definition induces a Cartesian differential category on the simple slice categories. Recall that the simple slice category of $\mathbb{X}$ with respect to A, denoted $\mathbb{X}[A]$, is the category with the same objects as $\mathbb{X}$ and where a map from $B \longrightarrow C$ in $\mathbb{X}[A]$ is a map $f : A \times B \longrightarrow C$ in $\mathbb{X}$; that is, in terms of homsets, $\mathbb{X}[A](B, C) = \mathbb{X}(A \times B, C)$, and composition of is given by $\langle f, \pi_1 \rangle g$.

▶ **Proposition 8.** *[2, Corollary 4.5.2] Let $\mathbb{X}$ be a Cartesian differential category and A any object. Then $\mathbb{X}[A]$ is a Cartesian differential category and the derivative of $f : A \times B \longrightarrow C$ is $D_B[f]$.*

Linear maps play a central role in the theory of Cartesian differential categories.

▶ **Definition 9.** *A map f in a Cartesian differential category is **linear** when $\mathsf{D}[f] = \pi_1 f$. Similarly, a map $A \times B \xrightarrow{f} C$ is **linear in** B if the following diagram commutes:*

$$\begin{tikzcd}
A \times (B \times B) \arrow[rr, "D_B[f]"] \arrow[dr, "1\times\pi_1"'] & & C \\
& A \times B \arrow[ur, "f"'] &
\end{tikzcd}$$

Note that a map $A \times B \xrightarrow{f} C$ is linear in B if and only if when regarded as a map $B \xrightarrow{f} C$ in $\mathbb{X}[A]$, it is linear with respect to the derivative in $\mathbb{X}[A]$.

▶ **Example 10.** Let us consider the linear maps in our examples of Cartesian differential categories from Example 5:

1. In a category with finite biproducts, every map is linear by definition of the differential combinator.
2. Let R be a commutative rig. In POLY_R, a map $n \xrightarrow{p} 1$ is linear if and only if $p(\vec{x}) = \sum\limits_{i=1}^{n} r_i x_i$ for some $r_i \in R$. And it follows that $n \xrightarrow{P} m$, with $P = \langle p_1(\vec{x}), \ldots, p_n(\vec{x}) \rangle$, is linear if and only if each $p_i(\vec{x})$ is. In other words, $n \xrightarrow{P} m$ is linear in the Cartesian differential category sense if and only if it induces an R-linear map $R^n \longrightarrow R^m$.
3. Similar to the previous example, in Smooth the linear maps in the Cartesian differential category sense are precisely the linear maps in the ordinary sense. Explicitly, $n \xrightarrow{F} m$ is linear if and only if $\mathbb{R}^n \xrightarrow{F} \mathbb{R}^m$ is a linear transformation.

For a Cartesian differential category $\mathbb{X}$, we can also form its subcategory of linear maps $\mathsf{Lin}(\mathbb{X})$, and since every linear map is additive [2], it follows that:

▶ **Proposition 11** ([2], Corollary 2.2.3)**.** *For a Cartesian differential category $\mathbb{X}$, its subcategory of linear maps $\mathsf{Lin}(\mathbb{X})$ has finite biproducts.*

Finally, we conclude this section with the observation that linearity can also be expressed in terms of injection maps:

▶ **Lemma 12.** *In a Cartesian differential category,*

- *A map $A \xrightarrow{f} B$ is linear if and only if $\iota_1 \mathsf{D}[f] = f$.*
- *A map $A \times B \xrightarrow{f} C$ is linear in B if and only if $(\iota_0 \times \iota_1)\mathsf{D}[f] = f$.*

3 Reverse derivatives

In this section we introduce our definition of a Cartesian reverse differential category. The types of axioms are similar to those for Cartesian differential categories; however, after the first two, the forms the axioms take are quite different.

▶ **Definition 13.** *A Cartesian left additive category $\mathbb{X}$ has* ***reverse derivatives*** *in case there is a combinator* R*, called the* ***reverse differential combinator****, which written as an inference rule is given by:*

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

where $\mathsf{R}[f]$ *is called the reverse derivative of f, and such that the following coherences are satisfied:*

[RD.1] $\mathsf{R}[f+g] = \mathsf{R}[f] + \mathsf{R}[g]$ *and* $\mathsf{R}[0] = 0$*;*

[RD.2] $\langle a, b+c\rangle\, \mathsf{R}[f] = \langle a, b\rangle\, \mathsf{R}[f] + \langle a, c\rangle\, \mathsf{R}[f]$ *and* $\langle a, 0\rangle\, \mathsf{R}[f] = 0$

[RD.3] $\mathsf{R}[1] = \pi_1$*, while for the projections, the following diagrams commute:*

$$\frac{A \times B \xrightarrow{\pi_0} A}{(A \times B) \times A \xrightarrow[\mathsf{R}[\pi_0]]{} A \times B}$$

$$\begin{tikzcd}
(A \times B) \times A \arrow[rr, "\mathsf{R}[\pi_0]"] \arrow[dr, "\pi_1"'] & & A \times B \\
& A \arrow[ur, "\iota_0"'] &
\end{tikzcd}$$

$$\frac{A \times B \xrightarrow{\pi_1} B}{(A \times B) \times B \xrightarrow[\mathsf{R}[\pi_1]]{} A \times B}$$

$$\begin{tikzcd}
(A \times B) \times B \arrow[rr, "\mathsf{R}[\pi_1]"] \arrow[dr, "\pi_1"'] & & A \times B \\
& B \arrow[ur, "\iota_1"'] &
\end{tikzcd}$$

[RD.4] *For a tupling of maps f and g, the following equality holds:*

$$\frac{A \xrightarrow{f} B}{A \times B \xrightarrow[\mathsf{R}[f]]{} A} \qquad \frac{A \xrightarrow{g} C}{A \times C \xrightarrow[\mathsf{R}[g]]{} A} \qquad \frac{A \xrightarrow{\langle f, g\rangle} B \times C}{A \times (B \times C) \xrightarrow[\mathsf{R}[\langle f, g\rangle]]{} A}$$

$$\mathsf{R}[\langle f, g\rangle] = (1 \times \pi_0)\mathsf{R}[f] + (1 \times \pi_1)\mathsf{R}[g]$$

While for the unique map to the terminal object: $!_A : A \longrightarrow \mathbf{1}$*, the following equality holds:*

$$\mathsf{R}[!_A] = 0$$

[RD.5] *For composable maps f and g, the following diagram commutes:*

$$\frac{A \xrightarrow{f} B}{A \times B \xrightarrow[\mathsf{R}[f]]{} A} \qquad \frac{B \xrightarrow{g} C}{B \times C \xrightarrow[\mathsf{R}[g]]{} B} \qquad \frac{A \xrightarrow{fg} C}{A \times C \xrightarrow[\mathsf{R}[fg]]{} A}$$

$$\begin{tikzcd}
A \times C \arrow[r, "\mathsf{R}[fg]"] \arrow[d, "{\langle \pi_0, \langle \pi_0 f, \pi_1\rangle\rangle}"'] & A \\
A \times (B \times C) \arrow[r, "1 \times \mathsf{R}[g]"'] & A \times B \arrow[u, "\mathsf{R}[f]"']
\end{tikzcd}$$

[RD.6] $\langle 1 \times \pi_0, 0 \times \pi_1\rangle\, (\iota_0 \times 1)\mathsf{R}[\mathsf{R}[\mathsf{R}[f]]]\pi_1 = (1 \times \pi_1)\mathsf{R}[f]$

[RD.7] $(\iota_0 \times 1)\mathsf{R}[\mathsf{R}[(\iota_0 \times 1)\mathsf{R}[\mathsf{R}[f]]\pi_1]]\pi_1 = \mathsf{ex}(\iota_0 \times 1)\mathsf{R}[\mathsf{R}[(\iota_0 \times 1)\mathsf{R}[\mathsf{R}[f]]\pi_1]]\pi_1$
A ***Cartesian reverse differential category*** *is a Cartesian left additive category with a reverse differential combinator.*

The axioms of the reverse differential combinator mirror those of a differential combinator. [**RD.1**] states that the reverse derivative of a sum is the sum of the reverse derivatives while [**RD.2**] says that the reverse derivative is additive in its second argument. [**RD.3**] and [**RD.4**] respectively explain what the reverse derivatives of the identity, projection, and tuples are. [**RD.5**] is the reverse derivative version of the chain rule. Lastly, [**RD.6**] expresses that the reverse derivative is linear in its second argument and [**RD.7**] gives the symmetry of the mixed partial reverse derivatives.

▶ **Example 14.** Here are some examples of reverse differential categories:

1. Let R be a commutative rig. POLY_R is a reverse differential category whose reverse differential combinator R is again defined using partial derivatives of polynomials. For a map $n \xrightarrow{P} m$, $P := \langle p_1(\vec{x}), \dots, p_m(\vec{x})\rangle$ with $p_i(\vec{x}) \in R[x_1, \dots, x_n]$, its reverse derivative $n \times m \xrightarrow{\mathsf{R}[P]} n$ is the tuple:

$$\mathsf{R}[P] := \langle \sum_{i=1}^{m} \frac{\partial p_i}{\partial x_1}(\vec{x})y_i, \dots, \sum_{i=1}^{m} \frac{\partial p_i}{\partial x_n}(\vec{x})y_i \rangle$$

   where each component of $\mathsf{R}[P]$ is a polynomial in $R[x_1, \dots, x_n, y_1, \dots, y_m]$. For example, consider from before the polynomial $p(x_1, x_2) = x_1^2 + 3x_1x_2 + 5x_2$, then $3 \xrightarrow{\mathsf{R}[p]} 2$ is the tuple of polynomials in 3 variables, $\mathsf{R}[p] = \langle (2x_1 + 3x_2)y, (3x_1 + 5)y \rangle$.

2. Smooth is a reverse differential category whose reverse differential combinator is defined using the transpose of the Jacobian. For a map $n \xrightarrow{F} m$, that is, a smooth function $\mathbb{R}^n \xrightarrow{F} \mathbb{R}^m$, its reverse derivative $n \times m \xrightarrow{\mathsf{R}[F]} n$ is the smooth map $\mathbb{R}^n \times \mathbb{R}^m \xrightarrow{\mathsf{R}[F]} \mathbb{R}^n$ defined as:

$$\mathsf{R}[F](\vec{x}, \vec{y}) := (J_f(x))^T \cdot \vec{y}$$

   In particular for a smooth map $\mathbb{R}^n \xrightarrow{f} \mathbb{R}$, its reverse derivative $\mathbb{R}^n \times \mathbb{R} \xrightarrow{\mathsf{R}[f]} \mathbb{R}^n$ is calculated out to be:

$$\mathsf{R}[f](\vec{x}, y) := \langle \frac{\partial f}{\partial x_1}(\vec{x})y, \dots, \frac{\partial f}{\partial x_n}(\vec{x})y \rangle$$

   And as before, $\mathsf{POLY}_{\mathbb{R}}$ is a sub-reverse differential category of Smooth.

The following lemma captures some basic properties of the reverse derivative.

▶ **Lemma 15.** *In a Cartesian reverse differential category, the following equalities holds:*

1. $\mathsf{R}[fg] = \langle \pi_0, \langle \pi_0 f, \pi_1 \rangle \, \mathsf{R}[g] \rangle \, \mathsf{R}[f]$;
2. $\mathsf{R}[\iota_0] = \pi_1\pi_0$ *and* $\mathsf{R}[\iota_1] = \pi_1\pi_1$;
3. $\mathsf{R}[\pi_0 f] = (\pi_0 \times 1)\mathsf{R}[f]\iota_0$ *and* $\mathsf{R}[\pi_1 f] = (\pi_1 \times 1)\mathsf{R}[f]\iota_1$;
4. $\mathsf{R}[f\pi_0] = (1 \times \iota_0)\mathsf{R}[f]$ *and* $\mathsf{R}[f\pi_1] = (1 \times \iota_1)\mathsf{R}[f]$;
5. $\mathsf{R}[f \times g] = \mathsf{ex}(\mathsf{R}[f] \times \mathsf{R}[g])$;
6. $\mathsf{R}[\iota_0 f] = (\iota_0 \times 1)\mathsf{R}[f]\pi_0$ *and* $\mathsf{R}[\iota_1 f] = (\iota_1 \times 1)\mathsf{R}[f]\pi_1$;
7. $\mathsf{R}[f\iota_0] = (1 \times \pi_0)\mathsf{R}[f]$ *and* $\mathsf{R}[f\iota_1] = (1 \times \pi_1)\mathsf{R}[f]$;
8. $\mathsf{R}[\langle f|g\rangle] = \langle \mathsf{D}[f\iota_0] | \mathsf{R}[g\iota_1]\rangle$;
9. $\mathsf{R}[f \oplus g] = \mathsf{ex}(\mathsf{R}[f] \times \mathsf{R}[g])$;

Proof. See the extended version [10] Lemma 15. ◀

3.1 Forward Differential Structure

Here we explain how every reverse derivative operator induces a forward derivative operator, that is, how every Cartesian reverse differential category is a Cartesian differential category. The trick was noticed in [6]: the reverse derivative in Smooth is the transpose of the Jacobian, which is linear, hence applying the reverse derivative again allows one to reconstruct the forward derivative. We formalize this in an arbitrary Cartesian reverse differential category as follows. Consider the resulting type of applying the reverse differential combinator twice:

$$\cfrac{\cfrac{A \xrightarrow{f} B}{A \times B \xrightarrow{\mathsf{R}[f]} A}}{(A \times B) \times A \xrightarrow[\mathsf{R}[\mathsf{R}[f]]]{} (A \times B)}$$

▶ **Theorem 16.** *If $\mathbb{X}$ is a Cartesian reverse differential category, then $\mathbb{X}$ is a Cartesian differential category with differential combinator D defined as follows (for any map $A \xrightarrow{f} B$):*

$$\mathsf{D}[f] := A \times A \xrightarrow{(\langle 1, 0 \rangle \times 1)} (A \times B) \times A \xrightarrow{\mathsf{R}[\mathsf{R}[f]]} A \times B \xrightarrow{\pi_1} B$$

Proof. See [10] Theorem 16. ◀

▶ **Example 17.** For both $\mathsf{POLY}_{\mathbb{R}}$ and Smooth, applying Theorem 16 to their respective reverse differential operators defined in Example 14 results precisely in their differential combinators defined in Example 5. This follows from the fact that there is a bijective correspondence between a reverse differential combinator and a differential combinator with an involution operation, which we will discuss in Section 4.

3.2 Dagger Structure and Linear Maps

We now investigate the subcategory of linear maps of the induced Cartesian differential category structure from Theorem 16 of a Cartesian reverse differential category. In particular we will show that the subcategory of linear maps has a dagger structure.

▶ **Definition 18.** *A **$\dagger$-category** [19] is a category $\mathbb{X}$ with a stationary on objects involution $\mathbb{X}^{op} \xrightarrow{(\_)^\dagger} \mathbb{X}$. A $\dagger$-category that also has finite biproducts $\oplus$, with projection maps π_i and injection maps ι_i, is said to have **$\dagger$-biproducts** [19] when $\pi_i^\dagger = \iota_i$ (or equivalently if $\iota_i^\dagger = \pi_i$).*

Note that having $\dagger$-biproducts implies that $0^\dagger = 0$ and $(f+g)^\dagger = f^\dagger + g^\dagger$. At this point we can also point out that in the same way that every category with finite biproducts is a Cartesian differential category, we have the following basic example of a reverse differential category:

▶ **Example 19.** Every $\dagger$-category with finite $\dagger$-biproducts is a reverse differential category where for a map $A \xrightarrow{f} B$, $A \oplus B \xrightarrow{\mathsf{R}[f]} A$ is defined as $\mathsf{R}[f] := A \oplus B \xrightarrow{\pi_1} B \xrightarrow{f^\dagger} A$. As a particular example, let R be a commutative rig and let $\mathsf{MAT}(R)$ be the category of matrices over R, that is, the category whose objects are the natural numbers $n \in \mathbb{N}$ and where a map $n \xrightarrow{A} m$ is an $n \times m$-matrix A with coefficients in R. $\mathsf{MAT}(R)$ admits finite biproducts where on objects $n \oplus m := n + m$ and where the projection and injection maps are the obvious matrices. $\mathsf{MAT}(R)$ also admits a $\dagger$ defined as the transpose of matrices and this makes $\mathsf{MAT}(R)$ into a $\dagger$-category with finite $\dagger$-biproducts.

For any map $A \xrightarrow{f} B$ in a reverse differential category, we can define a map of opposite type $B \xrightarrow{f^\dagger} A$ by $f^\dagger := \iota_1 \mathsf{R}[f]$. As the following example shows, however, in general this operation is neither functorial nor involutive.

▶ **Example 20.** With our standard example $2 \xrightarrow{p} 1$ in POLY_R, $p(x_1, x_2) = x_1^2 + 3x_1x_2 + 5x_2$, one computes that $1 \xrightarrow{p^\dagger} 2$ is the tuple of 1 variable polynomials $p^\dagger = \langle 0, 5x \rangle$.

However, as we shall see, † is well behaved for linear maps.

▶ **Lemma 21.** *With the preceding definition of † in a reverse differential category, one has that* $\pi_i^\dagger = \iota_i$ *and* $\iota_i^\dagger = \pi_i$*.*

▶ **Lemma 22.** *In a Cartesian reverse differential category, for any map* $A \xrightarrow{f} B$*, the following are equivalent:*

1. f *is linear (Definition 9) with respect to the differential combinator of Theorem 16;*
2. $\iota_1(\iota_0 \times 1)\mathsf{R}[\mathsf{R}[f]]\pi_1 = f$*;*
3. $f^{\dagger\dagger} = f$*.*

Proof. That *1* ⇔ *2* follows from the fact that by definition, the left hand side of *2* can be re-expressed as $\iota_1(\iota_0 \times 1)\mathsf{R}[\mathsf{R}[f]]\pi_1 = \langle 0, 1\rangle \, \mathsf{D}[f]$, and so *2* holds precisely when $\langle 0, 1 \rangle \, \mathsf{D}[f] = f$, which by Lemma 12 is equivalent to $\mathsf{D}[f] = \pi_1 f$, that is, that f is linear. Next we show that *2* ⇔ *3*. First note that $\iota_1(\iota_0 \times 1) = \iota_1(\iota_1 \times 1)$ since:

$$\iota_1(\iota_0 \times 1) = \langle 0, 1\rangle \, (\langle 1, 0 \rangle \times 1) = \langle 0, 1 \rangle = \langle 0, 1 \rangle \, (\langle 0, 1 \rangle \times 1) = \iota_1(\iota_1 \times 1)$$

And then by Lemma 15.6, we have the following equality:

$$f^{\dagger\dagger} = \iota_1 \mathsf{R}[\iota_1 \mathsf{R}[f]] = \iota_1(\iota_1 \times 1)\mathsf{R}[\mathsf{R}[f]]\pi_1 = \iota_1(\iota_0 \times 1)\mathsf{R}[\mathsf{R}[f]]\pi_1$$

Then it immediately follows that $f^{\dagger\dagger} = f$ if and only if $f = \iota_1(\iota_0 \times 1)\mathsf{R}[\mathsf{R}[f]]\pi_1$. ◀

▶ **Lemma 23.** *In a Cartesian reverse differential category, for any* $A \xrightarrow{f} B$*, its reverse derivative* $A \times B \xrightarrow{\mathsf{R}[f]} A$ *is linear in* B *(Definition 9) with respect to the differential combinator of Theorem 16. Furthermore, the following diagram commutes:*

$$\begin{tikzcd}
((A \times B) \times A) \times (A \times B) \arrow[r, "\mathsf{R}^{(3)}[f]"] & (A \times B) \times A \arrow[d, "\pi_1"] \\
A \times B \arrow[u, "\langle \iota_0, 0 \rangle \times \iota_1"] \arrow[r, "\mathsf{R}[f]"'] & A
\end{tikzcd}$$

Proof. That $A \times B \xrightarrow{\mathsf{R}[f]} A$ is linear in B follows immediately from the **[RD.6]** (we leave it as an exercise to re-express **[RD.6]** in terms of partial derivatives). Commutativity of the diagram follows by applying Lemma 12 to $\mathsf{R}[f]$. ◀

▶ **Proposition 24.** *For a Cartesian reverse differential category* $\mathbb{X}$*, the category of linear maps of the induced Cartesian differential category structure from Theorem 16,* $\mathsf{Lin}(\mathbb{X})$*, is a †-category with finite †-biproducts.*

Proof. By Proposition 11, we already know that $\mathsf{Lin}(\mathbb{X})$ has finite biproducts. We need to show that $\mathsf{Lin}(\mathbb{X})$ also has a †. Lemma 22 shows that the linear maps are precisely those for which $f^{\dagger\dagger} = f$, and thus if f is linear then $f^\dagger$ is linear. Therefore $\dagger$ is well-defined and involutive. We now show that $\dagger$ is a contravariant functor. First that $\dagger$ preserves the identity:

$$1^\dagger = \iota_1 \mathsf{R}[1] = \iota_1 \pi_1 = 1$$

Next, that $\dagger$ preserves composition (recall that if f is linear, then $0f = 0$):

$$(fg)^\dagger = \iota_1 \mathsf{R}[fg] = \iota_1 \langle \pi_0, \langle \pi_0 f, \pi_1 \rangle \mathsf{R}[g] \rangle \mathsf{R}[f] = \langle \iota_1 \pi_0, \langle \iota_1 \pi_0 f, \iota_1 \pi_1 \rangle \mathsf{R}[g] \rangle \mathsf{R}[f]$$

$$= \langle 0, \langle 0f, 1 \rangle \mathsf{R}[g] \rangle \mathsf{R}[f] = \langle 0, \langle 0, 1 \rangle \mathsf{R}[g] \rangle \mathsf{R}[f] = \langle 0, 1 \rangle \mathsf{R}[g] \langle 0, 1 \rangle \mathsf{R}[f] = g^\dagger f^\dagger$$

Note in the above that functoriality only relies on f preserving 0. Thus $\mathsf{Lin}(\mathbb{X})$ is a $\dagger$-category. Lastly by Lemma 21, $\mathsf{Lin}(\mathbb{X})$ also has $\dagger$-biproducts. ◀

4 From forward derivatives to reverse derivatives

In the previous section, we showed that a Cartesian reverse differential category gives rise to a Cartesian differential category in which the subcategory of linear maps has a dagger biproduct structure. For the converse we need to develop Cartesian differential categories where every simple slice linear map category is a dagger category with dagger biproducts. The conceptual structure behind this is what we call a dagger fibration with fibered dagger biproducts. We will show that when a Cartesian differential category's linear map fibration is such a dagger fibration then the category is also a Cartesian reverse differential category.

4.1 Review of Fibrations and the Dual Fibration

We first recall the notion of fibration (for example, see [13, Section 1.1]) and the lesser-known idea of the dual of a fibration. These will be helpful concepts in which to frame our characterization of reverse differential categories (Theorem 42) and to describe how the reverse derivative is functorial (Proposition 31).

▶ **Definition 25.** *Suppose that $q : \mathbb{X} \longrightarrow \mathbb{B}$ is a functor.*

1. *Say that a morphism $f : X \longrightarrow Y$ in $\mathbb{X}$ is* ***over*** *a morphism $u : I \longrightarrow J$ in $\mathbb{B}$ if $q(f) = u$.*
2. *Say that a morphism $f : X \longrightarrow Y$ in $\mathbb{X}$ is* ***Cartesian over*** *$u : I \longrightarrow J$* ***in*** *$\mathbb{B}$ if f is over u, and for every $g : Z \longrightarrow Y$ in $\mathbb{X}$ such that $q(q) = wu$ for some $w : q(Z) \longrightarrow I$, there is a unique $h : Z \longrightarrow X$ in $\mathbb{X}$ over w such that $hf = g$:*

$$\begin{tikzcd}
Z \arrow[drr, bend left, "g"] \arrow[dr, dotted, "h"] & & \\
& X \arrow[r, "f"] & Y
\end{tikzcd}$$

$$\begin{tikzcd}
q(Z) \arrow[drr, bend left, "q(g)"] \arrow[dr, "w"] & & \\
& I \arrow[r, "u"] & J
\end{tikzcd}$$

3. *Say that q is a* ***fibration*** *if for every Y in $\mathbb{X}$ and every $u : I \longrightarrow q(Y)$ in $\mathbb{B}$, there is a Cartesian morphism $f : X \longrightarrow Y$ in $\mathbb{X}$ above u.*
4. *Say that an arrow $f : X \longrightarrow Y$ in $\mathbb{X}$ is* ***vertical*** *if f is over an identity map.*

5. *For an object I in $\mathbb{B}$, the* ***fibre of*** *q* ***over*** *I, denoted $q^{-1}(I)$, is the category whose objects are those objects of $\mathbb{X}$ for which $q(X) = I$, and whose arrows are vertical morphisms between them.*

▶ **Example 26.** If $\mathbb{X}$ is a Cartesian category, then the **simple fibration** [13, Definition 1.3.1] $\widetilde{\mathbb{X}} \xrightarrow{\pi} \mathbb{X}$ is described as follows: the total category $\widetilde{\mathbb{X}}$ has objects pairs of objects of $\mathbb{X}$ and a map $(I, A) \xrightarrow{(f,g)} (J, B)$ is given by a pair of maps of type $I \xrightarrow{f} J$ and $I \times A \xrightarrow{g} B$. The identity of (I, A) is $(1_A, \pi_1)$ while the composition of maps $(I, A) \xrightarrow{(f,g)} (J, B)$ and $(J, B) \xrightarrow{(f',g')} (K, C)$ is defined as: $(I \xrightarrow{f} J \xrightarrow{f'} K, I \times A \xrightarrow{\langle \pi_0 f, g\rangle} J \times B \xrightarrow{g'} C)$. The fibration $\widetilde{\mathbb{X}} \xrightarrow{\pi} \mathbb{X}$ is the functor which on objects is $\pi(I, A) = I$ and on maps is $\pi(f, g) := f$. The vertical arrows in $\widetilde{\mathbb{X}}$ are precisely those of the form $(I, A) \xrightarrow{(1,g)} (I, B)$ while the Cartesian arrows are those of the form $(I, A) \xrightarrow{(f,\pi_1)} (J, A)$.

▶ **Example 27.** If $\mathbb{X}$ is a Cartesian differential category, we denote by $\widetilde{\mathsf{Lin}(\mathbb{X})}$ the **simple linear fibration**, whose objects are pairs of objects in $\mathbb{X}$ and whose maps $(I, A) \xrightarrow{f,g} (J, B)$ are pairs of maps $I \xrightarrow{f} J$ and $I \times A \xrightarrow{g} B$ where g is linear in B. Composition and identities of $\widetilde{\mathsf{Lin}(\mathbb{X})}$ are defined as for the simple fibration. The fiber over A of this fibration is denoted $\mathsf{Lin}(\mathbb{X})[A]$. Note that by [2, Proposition 1.5.4], every fiber of $\widetilde{\mathsf{Lin}(\mathbb{X})}$ has biproducts.

▶ **Definition 28.** *Suppose that $\mathbb{X} \xrightarrow{q} \mathbb{B}$ is a fibration. The* ***dual fibration*** *of q [5, 14] is a fibration $\mathbb{X}^* \xrightarrow{q^*} \mathbb{B}$ whose total category $\mathbb{X}^*$ has the same objects as $\mathbb{X}$ and where a map $X \longrightarrow Y$ in $\mathbb{X}^*$ is an equivalence class of spans*

$$\begin{tikzcd}
S \arrow[r, "c"] \arrow[d, "v"'] & Y \\
X &
\end{tikzcd}$$

where v is vertical and c is Cartesian (over $q(c)$) under the equivalence relation $(v, c) \sim (v', c')$ when there is a vertical isomorphism α that makes the following diagram commute.

$$\begin{tikzcd}
& S' \arrow[dl, "v'"'] \arrow[d, "\alpha"] \arrow[dr, "c'"] & \\
X & S \arrow[l, "v"] \arrow[r, "c"'] & Y
\end{tikzcd}$$

To compose such spans, note that given a cospan $S \xrightarrow{c} X' \xleftarrow{v'} S'$ with c cartesian and v' vertical,that there is a cartesian arrow over $q(c)$ with codomain S', and this induces uniquely a v'' making the relevant square commute, and we get a span $S \xleftarrow{v''} S'' \xrightarrow{\hat{c}} S'$ with v'' vertical and $\hat{c}$ cartesian; this span is used to from the composite of the spans $(v, c)(v', c')$. For more details, see [14]. The fibration q^* is defined on objects as $q^*(A) := q(A)$, and defined on maps as $q^*(v, c) := q(X) = q(S) \xrightarrow{q(C)} q(Y)$.

▶ **Example 29.** The dual of the simple fibration, $\widetilde{X}^*$, can be described as the category with objects pairs of objects of $\mathbb{X}$ and with maps $(I, A) \xrightarrow{(f,g)} (J, B)$ where $I \xrightarrow{f} J$ and a $I \times B \xrightarrow{g} A$. The identity on (I, A) is $(1, \pi_1)$, while composition of maps $(I, A) \xrightarrow{(f,g)} (J, B)$ and $(J, B) \xrightarrow{(f',g')} (K, C)$ is defined to be

$$(I \xrightarrow{ff'} K, I \times C \xrightarrow{\langle \pi_0, \langle \pi_0 f, \pi_1 \rangle\rangle} (I \times (J \times C)) \xrightarrow{1 \times g'} I \times B \xrightarrow{g} A).$$

▶ **Example 30.** The dual of the linear fibration, $\widetilde{\mathsf{Lin}(\mathbb{X})}^*$, has again objects (I, A) but now maps $(I, A) \xrightarrow{(f,g)} (J, B)$ consist of pairs of a map $I \xrightarrow{f} J$ and a map $I \times B \xrightarrow{g} A$ such that g is linear in B.

The dual of the linear fibration allows us to describe how the reverse derivative is functorial:

▶ **Proposition 31.** *For a Cartesian reverse differential category $\mathbb{X}$, there is a product-preserving functor $\mathbb{X} \longrightarrow \widetilde{\mathsf{Lin}(\mathbb{X})}^*$ defined on objects as $A \mapsto (A, A)$ and on maps as $f \mapsto (f, R[f])$.*

Proof. This follows from [**RD.3**] and [**RD.5**]. ◀

▶ **Lemma 32.** *A fiber of the dual fibration is isomorphic to the opposite category of the associated fiber of the starting fibration; that is, for any A in $\mathbb{B}$, $q^{*\text{-}1}(A) \simeq (q^{\text{-}1}(A))^{op}$ and moreover the isomorphism is stationary on objects.*

Proof. See [10] Lemma 32. ◀

Note that $\mathbb{X}$ and $\mathbb{X}^{**}$ are also isomorphic as fibrations over $\mathbb{B}$; see [14, Proposition 3.4].

4.2 Dagger fibrations

We now introduce the notion of a dagger fibration. First recall that a *morphism of fibrations* (over a fixed base) is a commuting triangle:

$$\begin{tikzcd}
\mathbb{X} \arrow[rr, "h"] \arrow[dr, "p"'] & & \mathbb{Y} \arrow[dl, "q"] \\
& \mathbb{B} &
\end{tikzcd}$$

where h carries Cartesian maps to Cartesian maps.

▶ **Definition 33.** *A **dagger fibration** is given by a fibration $\mathbb{X} \xrightarrow{q} \mathbb{B}$ with a morphism of fibrations $\mathbb{X} \xrightarrow{(\_)^\dagger} \mathbb{X}^*$ such that*

$$\begin{tikzcd}
\mathbb{X} \arrow[r, "(\_)^\dagger"] \arrow[rr, bend left, "1_{\mathbb{X}}"] \arrow[dr, "q"'] & \mathbb{X}^* \arrow[r, "(\_)^\dagger"] \arrow[d, "q^*"] & \mathbb{X}^{**} = \mathbb{X} \arrow[dl, "q"] \\
& \mathbb{B} &
\end{tikzcd}$$

*and such that $\dagger$ is stationary on objects. A dagger fibration has a **dagger cleavage** when $(\_)^\dagger$ sends cloven cartesian arrows to cloven cartesian arrows.*

Our main example of a dagger fibration will be the linear fibration of a Cartesian reverse differential category. We begin by defining the required dagger (this is a more general form of the dagger discussed earlier in Section 3.2):

▶ **Definition 34.** *In a Cartesian reverse differential category $\mathbb{X}$, for a map $C \times A \xrightarrow{f} B$, define the **contextual** $\dagger$ of f, $C \times B \xrightarrow{f^{\dagger[C]}} A$, as follows:*

$$f^{\dagger[C]} := C \times B \xrightarrow{\iota_0 \times 1} (C \times A) \times B \xrightarrow{\mathsf{R}[f]} C \times A \xrightarrow{\pi_1} A$$

▶ **Lemma 35.** *In a Cartesian reverse differential category, for any map $C \times A \xrightarrow{f} B$, the following are equivalent:*

1. *f is linear in A (Definition 9) with respect to the differential combinator of Theorem 16;*
2. *$(\iota_0 \times \iota_1)\mathsf{ex}\mathsf{D}[f] = f$;*
3. *$f^{\dagger[C]\dagger[C]} = f$.*

Proof. *1* ⇔ *2* follows from Lemma 12. To show that *2* ⇔ *3* requires a bit more work, but the proof is essentially the same as in Lemma 22. ◀

▶ **Corollary 36.** *Let $\mathbb{X}$ be a Cartesian reverse differential category and let $I \times A \xrightarrow{g} B$ be linear in A. Then $I \times B \xrightarrow{g^{\dagger[I]}} A$ is linear in B.*

▶ **Theorem 37.** *If $\mathbb{X}$ is a Cartesian reverse differential category, then its associated linear fibration is a dagger fibration, with dagger as in Definition 34.*

Proof. See [10] Theorem 37. ◀

▶ **Lemma 38.** *If $\mathbb{X} \xrightarrow{q} \mathbb{B}$ is a dagger fibration with a dagger cleavage, then each fiber $q^{-1}(A)$ is a †-category, and reindexing preserves the dagger.*

Proof. See [10] Lemma 38. ◀

4.3 Characterization of Cartesian reverse differential categories

We have seen in the previous sections that a Cartesian reverse differential category is a Cartesian differential category whose associated linear fibration is a dagger fibration in which each fibre has †-biproducts. In this final section, we show that this collection of structures characterizes Cartesian reverse differential categories.

▶ **Definition 39.** *Let $\mathbb{X}$ be a Cartesian differential category. We say that $\mathbb{X}$ has a* ***contextual linear dagger*** *when the linear fibration is a dagger fibration*

$$\begin{tikzcd}
\widetilde{\mathsf{Lin}(\mathbb{X})} \arrow[rr, "(\_)^{\dagger}"] \arrow[dr, "\pi"'] & & \widetilde{\mathsf{Lin}(\mathbb{X})}^{*} \arrow[dl, "\pi^{*}"] \\
& \mathbb{X} &
\end{tikzcd}$$

and each fiber category $\mathsf{Lin}(\mathbb{X})[A]$ has †-biproducts.

By Lemma 38, every fiber of such a fibration is a †-category, and reindexing functors preserve the dagger. We denote the † in the fiber $\mathsf{Lin}(\mathbb{X})[A]$ by $(\_)^{\dagger[A]}$. In particular we note that $(\_)^{\dagger[A]}$ preserves the additive structure. Before giving the main theorems of this section, we will need the following lemma:

▶ **Lemma 40.** *Let $\mathbb{X}$ be a Cartesian differential category with a contextual linear dagger. For any map $A \xrightarrow{f} B$ the following diagram commutes.*

$$\begin{tikzcd}
(A \times B) \times A \arrow[r, "\mathsf{D}[\mathsf{D}[f]^{\dagger[A]}]^{\dagger[A \times B]}"] & A \times B \arrow[d, "\pi_1"] \\
A \times A \arrow[u, "\langle 1,0\rangle \times 1"] \arrow[r, "\mathsf{D}[f]"'] & B
\end{tikzcd}$$

Proof. See [10] Lemma 40. ◀

▶ **Theorem 41.** *A Cartesian differential category $\mathbb{X}$ with a contextual linear dagger is a Cartesian reverse differential category with reverse differential combinator* R *defined as follows (for a map $A \xrightarrow{f} B$):*

$$\mathsf{R}[f] := A \times B \xrightarrow{\mathsf{D}[f]^{\dagger[A]}} B$$

Proof. See [10] Theorem 41. ◀

We conclude with the main result of this paper:

▶ **Theorem 42.** *A Cartesian reverse differential category is precisely a Cartesian differential category with a contextual linear dagger.*

Proof. See [10] Theorem 42. ◀

5 Concluding remarks

This paper begins the story of categories with a reverse derivative; however, there is much more that needs to be done in this area. Perhaps the most important next step is to add partiality into this setting. One way to add partiality to categories is via a restriction structure [8]. The paper [7] showed how to combine a Cartesian differential structure with a restriction structure to obtain "differential restriction categories." This provides an axiomatization for categories of smooth partial maps. A key next step is then to combine reverse differential categories with restriction structure, and check that many of the results that held for differential restriction categories hold for "reverse differential restriction categories". Such a structure would bring us even closer to a true categorical semantics for differential programming.

Another important aspect to develop will be the term logic for reverse differential categories. The term logic for Cartesian differential categories greatly facilitates the ability to establish and prove results in that abstract setting; a term logic for reverse differential categories is similarly important.

Tensors are another important aspect of differential programming, and form the foundations on which modern, large scale machine learning platforms are based [1]. In [3], monoidal structure was described in a way that interacts well with differentiation. In particular, $V \otimes W$ is the object for which bilinear maps $V \times W \longrightarrow U$ correspond to linear maps $V \otimes W \longrightarrow U$. Developing a similar structure for the reverse derivative will thus also be important. More generally, there should be a notion of (monoidal) reverse differential category. These should provide additional examples of Cartesian reverse differential categories: just as the coKleisli category of a (monoidal) differential category [4] is a Cartesian differential category, so should the coKleisli category of a monoidal reverse differential category be a Cartesian reverse differential category.

Finally, an important generalization of Cartesian differential categories are tangent categories [9], a categorical setting for differential geometry which axiomatizes the existence of a "tangent bundle" for each object. Every Cartesian differential category gives rise to a tangent category. A reverse derivative category should give a "category with a cotangent bundle for each object"; defining such categories will be another important extension of this work.

References

1 Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz

Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. URL: `https://www.tensorflow.org/`.

2 R. Blute, R. Cockett, and R. Seely. Cartesian Differential Categories. *Theory and Applications of Categories*, 22:622–672, 2009.

3 R. Blute, R. Cockett, and R. Seely. Cartesian Differential Storage Categories. *Theory and Applications of Categories*, 30(18):620–686, 2015.

4 R.F. Blute, J.R.B. Cockett, and R.A.G. Seely. Differential categories. *Mathematical structures in computer science*, 16(6):1049–1083, 2006.

5 F. Borceaux. *Handbook of categorical algebra II.* Cambridge University Press, 2008.

6 Bruce Christianson. A Leibniz notation for automatic differentiation. In *Recent Advances in Algorithmic Differentiation*, volume 87 of *Lecture Notes in Computational Science and Engineering*, pages 1–9. Springer, 2012.

7 J.R.B. Cockett, G.S.H. Cruttwell, and J.D. Gallagher. Differential restriction categories. *Theory and applications of categories*, 25(21):537–613, 2011.

8 J.R.B. Cockett and Stephen Lack. Restriction categories I: categories of partial maps. *Theoretical Computer Science*, 270(1):223–259, 2002. `doi:10.1016/S0304-3975(00)00382-0`.

9 R. Cockett and G. Cruttwell. Differential structure, tangent structure, and SDG. *Applied Categorical Structures*, 22:331–417, 2014.

10 Robin Cockett, Geoffrey Cruttwell, Jonathan Gallagher, Jean-Simon Lemay, Benjamin MacAdam, Gordon Plotkin, and Dorette Pronk. Reverse derivative categories. *Arxiv*, 2019.

11 Conal Elliott. The simple essence of automatic differentiation. *Proceedings of the ACM on Programming Languages*, 2(ICFP):70, 2018.

12 Andreas Griewank. Who invented the reverse mode of differentiation. *Documenta Mathematica, Extra Volume ISMP*, pages 389–400, 2012.

13 B. Jacobs. *Categorical logic and type theory.* Number 141 in Studies in logic and the foundations of mathematics. Elsevier, 1999.

14 Anders Kock. The dual fibration in elementary terms. *arXiv e-prints*, page arXiv:1501.01947, January 2015. `arXiv:1501.01947`.

15 J-S P. Lemay. A Tangent Category Alternative to the Faa Di Bruno Construction. *Theory and Applications of Categories*, 33(35):1072–1110, 2018.

16 S. Linnainmaa. Taylor expansion of the accumulated rounding error. *BIT Numerical Mathematics*, 16:146–160, 1976.

17 G. Plotkin. A Simple Differential Programming Language. MFPS 2018 Keynote Address, June 2018.

18 David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by backpropagating errors. *Cognitive modeling*, 5:3, 1988. URL: `www.cs.toronto.edu/hinton/naturebp.pdf`.

19 Peter Selinger. Dagger Compact Closed Categories and Completely Positive Maps. *Electron. Notes Theor. Comput. Sci.*, 170:139–163, March 2007. `doi:10.1016/j.entcs.2006.12.018`.