Monoidal Reverse Differential Categories
arXiv:2203.12478

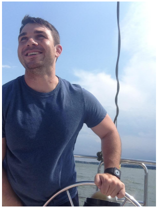
JS Pacaud Lemay (he/him)

Email: jsplemay@gmail.com

Website: https://sites.google.com/view/jspl-personal-webpage

Joint Work With These Awesome Mathematicians

Geoff Cruttwell



Jonathan Gallagher

Dorette Pronk

JS PL

Cruttwell, G., Gallagher, J., Lemay, J. S. P., & Pronk, D. *Monoidal Reverse Differential Categories.*
https://arxiv.org/abs/2203.12478

Forward and Reverse

In programming there are two types of derivative operations:

FOWARD	**REVERSE**

Forward and Reverse

In programming there are two types of derivative operations:

FOWARD **REVERSE**

The forward derivative is used in:

- Classical Math: Differential Calculus and Differential Geometry
- Differential Linear Logic
- Differential λ-Calculus

The categorical story of forward differentiation is captured by **differential categories**.

Forward and Reverse

In programming there are two types of derivative operations:

FOWARD **REVERSE**

The forward derivative is used in:

- Classical Math: Differential Calculus and Differential Geometry
- Differential Linear Logic
- Differential λ-Calculus

The categorical story of forward differentiation is captured by **differential categories**.

The reverse derivative is more practical and used in:

- Automatic Differentiation
- Machine Learning
- Differentiable Programming

The categorical story of reverse differentiation is captured by **reverse differential categories**:

Cockett, R., Cruttwell, G., Gallagher, J., Lemay, J. S. P., MacAdam, B., Plotkin, G., & Pronk, D. (2020). *Reverse derivative categories.* In the proceedings of CSL2020.

The story of **reverse differential categories** is just starting!

Rise of Reverse Differential Categories

Wilson, P., & Zanasi, F. Reverse (2020) *Categories of Differentiable Polynomial Circuits for Machine Learning.* **ACT2022**

Cruttwell, G., Gavranović, B., Ghani, N., Wilson, P., & Zanasi, F.: *Categorical foundations of gradient-based learning.* ESOP2022

Wilson, P., & Zanasi, F. Reverse (2020) *Derivative Ascent: A Categorical Approach to Learning Boolean Circuits.* ACT2020

Cruttwell, G., Gallagher, J., & Pronk, D. (2020) *Categorical semantics of a simple differential programming language.* ACT2020

What is Today's Story?

- This paper:

  Cockett, R., Cruttwell, G., Gallagher, J., Lemay, J. S. P., MacAdam, B., Plotkin, G., & Pronk, D. (2020). *Reverse derivative categories.* In the proceedings of CSL2020.

  introduced Cartesian reverse differential categories.

What is Today's Story?

- This paper:

  Cockett, R., Cruttwell, G., Gallagher, J., Lemay, J. S. P., MacAdam, B., Plotkin, G., & Pronk, D. (2020). *Reverse derivative categories.* In the proceedings of CSL2020.

  introduced Cartesian reverse differential categories.

- The forward counterpart are Cartesian differential categories – which capture multivariable calculus over Euclidean spaces and the differential λ-calculus.

What is Today's Story?

- This paper:

  Cockett, R., Cruttwell, G., Gallagher, J., Lemay, J. S. P., MacAdam, B., Plotkin, G., & Pronk, D. (2020). *Reverse derivative categories.* In the proceedings of CSL2020.

  introduced Cartesian reverse differential categories.

- The forward counterpart are Cartesian differential categories – which capture multivariable calculus over Euclidean spaces and the differential λ-calculus.

- Monoidal differential categories capture the algebraic foundations of differentiation and differential linear logic. An important result in **forward differential categories** is the following:

Theorem

The coKleisli category of a **MONOIDAL differential category** *is a* **CARTESIAN differential category**.

What is Today's Story?

- This paper:

  Cockett, R., Cruttwell, G., Gallagher, J., Lemay, J. S. P., MacAdam, B., Plotkin, G., & Pronk, D. (2020). *Reverse derivative categories*. In the proceedings of CSL2020.

  introduced Cartesian reverse differential categories.

- The forward counterpart are Cartesian differential categories – which capture multivariable calculus over Euclidean spaces and the differential λ-calculus.

- Monoidal differential categories capture the algebraic foundations of differentiation and differential linear logic. An important result in **forward differential categories** is the following:

Theorem

The coKleisli category of a **MONOIDAL differential category** *is a* **CARTESIAN differential category**.

TODAY'S STORY: Introduce the notion of a **MONOIDAL reverse differential category**:

- Show that the coKleisli category is a CARTESIAN reverse differential categories.

How did you know it was possible?

QUESTION: Why did we think monoidal reverse differential categories possible?

ANSWER: Because forward differentiation and reverse differentiation are linked!

If you only remember one slide from my talk: IT'S THIS ONE!

REVERSE DIFFERENTIATION

=

FORWARD DIFFERENTIATION + **TRANSPOSE/DAGGER**

Cartesian Forward/Reverse Differential Categories

Briefly, a **Cartesian differential category** is in particular a category which has finite products and comes equipped with a **differential combinator** D:

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

R. Blute, R. Cockett, R.A.G. Seely, **Cartesian Differential Categories**

Cartesian Forward/Reverse Differential Categories

Briefly, a **Cartesian differential category** is in particular a category which has finite products and comes equipped with a **differential combinator** D:

$$\frac{f : A \to B}{\mathrm{D}[f] : A \times A \to B}$$

R. Blute, R. Cockett, R.A.G. Seely, **Cartesian Differential Categories**

Briefly, a **Cartesian reverse differential category** is in particular a category which has finite products and comes equipped with a **reverse differential combinator** R:

$$\frac{f : A \to B}{\mathrm{R}[f] : A \times B \to A}$$

Cockett, R., Cruttwell, G., Gallagher, J., Lemay, J. S. P., MacAdam, B., Plotkin, G., & Pronk, D. (2020). *Reverse derivative categories.* In the proceedings of CSL2020.

Example

Let $f : \mathbb{R}^2 \to \mathbb{R}$ with $f(x, y) = x^2 y$. Then:

$$\mathrm{D}[f] : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$$

$$\mathrm{D}[f](x, y, a, b) = 2xya + x^2 b$$

$$\mathrm{R}[f] : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^2$$

$$\mathrm{R}[f](x, y, a) = \langle 2xya, x^2 a \rangle$$

Linear Dagger

In a CDC there is a natural notion of **linear** which is likes the classical notion from linear algebra and that if you differentiate a map in that argument you get back your function.

A map $f : C \times A \to B$ is linear in its second argument $Aorlinearincontext$C if $\mathrm{D}[f] \circ \langle c, 0, 0, a \rangle = f \circ \langle c, a \rangle$. Linear maps form a fibration over the CDC.

Linear Dagger

In a CDC there is a natural notion of **linear** which is likes the classical notion from linear algebra and that if you differentiate a map in that argument you get back your function.

A map $f : C \times A \to B$ is linear in its second argument $A or linear in context$C if $\mathrm{D}[f] \circ \langle c, 0, 0, a \rangle = f \circ \langle c, a \rangle$. Linear maps form a fibration over the CDC.

A CDC has a **contextual linear dagger** if each fibre of the fibration is a dagger category. Explicitly:

$$\frac{f : C \times A \to B}{f^{\dagger[C]} : C \times B \to A}$$

which satisfies the expected contravariant involutive coherences and preserves linearity in the second argument.

Theorem

The following are equivalent:

- *Cartesian reverse differential category*
- *Cartesian differential category with a contextual linear dagger.*

Here's the plan

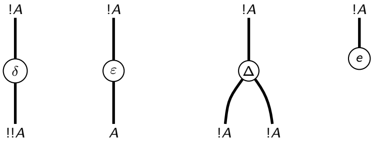
To define a monoidal reverse differential category we need to understand the following for a monoidal differential category:

- What are the linear maps (in context) in the coKleisli category
- How to define the contextual linear dagger.

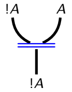
Then a monoidal reverse differential category will be a monoidal differential category with a contextual linear dagger.

Monoidal Differential Category

A **monoidal differential category** is a symmetric monoidal category which is enriched over commutative monoids (so we can add maps $+$ and have zero maps 0) and equipped with a comonad $!$ such that each $!A$ is a cocommutative comonoid [1]:



and also equipped with a **deriving transformation** $\mathsf{d}_A : !A \otimes A \to !A$:

$\mathsf{d} :=$ 

satisfying basic axioms of differentiation like the Leibniz rule and the chain rule.

R. Blute, R. Cockett, R.A.G. Seely, **Differential Categories** (2006)

R. Blute, R. Cockett, R.A.G. Seely, JS Lemay **Differential categories revisited**. (2019)

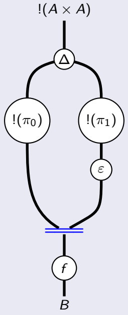
[1] String diagrams read from top to bottom

CoKleisli is a CDC

Recall that the coKleisli category of a comonad ! has the same objects as the base category where a map from $A \to B$ is a map $!A \to B$.

Theorem

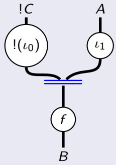
For a monoidal differential category with finite product, its coKleisli category is a Cartesian differential category where for a map $f : !A \to B$ its derivative $\mathsf{D}[f] : !(A \times A) \to B$ is defined as follows:

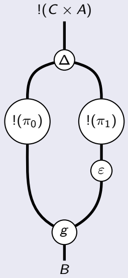


Linear Maps in CoKleisli

Theorem (*NEW*)

For a monoidal differential category with finite product, linear maps in context $f : !(C \times A) \to B$ in the coKleisli category correspond precisely to maps $g : !C \otimes A \to B$ in the base category.





This gives an isomorphism of fibrations.

Contextual Linear Dagger

To give a contextual linear dagger in a monoidal differential category is equivalent to giving a combinator:

$$\frac{f : !C \otimes A \to B}{f^{\dagger[C]} : !C \otimes B \to A}$$

Contextual Linear Dagger

To give a contextual linear dagger in a monoidal differential category is equivalent to giving a combinator:

$$\frac{f : !C \otimes A \to B}{f^{\dagger[C]} : !C \otimes B \to A}$$

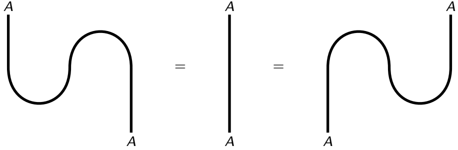
It turns out that under mild assumptions (that any model of differential linear logic satisfies), to give a contextual linear dagger is equivalent to requiring that the base category is a **self-dual compact closed category**.

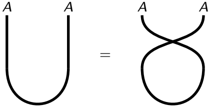
Self-Dual Compact Closed

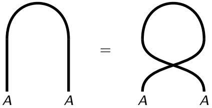
By a **self-dual compact closed category**, we mean a compact closed category where $A^* = A$ and also the twist equations [2] hold:

$\cup_A$:= 

$\cap_A$:= 







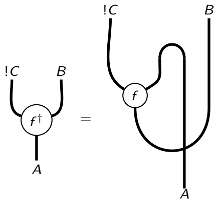
[2] The twist equations are not strictly necessary for the story. But they do greatly simply the string diagrams and all examples we have so far satisfy them. Furthermore, in most of the literature when considering self-dual compact closed categories, the twist equations are often taken as axioms.

Contextual Linear Dagger

To give a contextual linear dagger in a monoidal differential category is equivalent to giving a combinator:

$$\frac{f : !C \otimes A \to B}{f^{\dagger[C]} : !C \otimes B \to A}$$

The contextual linear dagger is defined using the cups and caps:

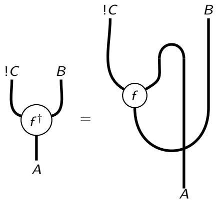


Contextual Linear Dagger

To give a contextual linear dagger in a monoidal differential category is equivalent to giving a combinator:

$$\frac{f : !C \otimes A \to B}{f^{\dagger[C]} : !C \otimes B \to A}$$

The contextual linear dagger is defined using the cups and caps:



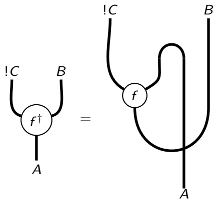
So a **monoidal reverse differential category** will be a monoidal differential category which is also a self-dual compact closed category.

Contextual Linear Dagger

To give a contextual linear dagger in a monoidal differential category is equivalent to giving a combinator:

$$\frac{f : !C \otimes A \to B}{f^{\dagger[C]} : !C \otimes B \to A}$$

The contextual linear dagger is defined using the cups and caps:



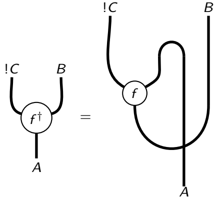
So a **monoidal reverse differential category** will be a monoidal differential category which is also a self-dual compact closed category. But can we define the **reverse differential** separately?

Contextual Linear Dagger

To give a contextual linear dagger in a monoidal differential category is equivalent to giving a combinator:

$$\frac{f : !C \otimes A \to B}{f^{\dagger[C]} : !C \otimes B \to A}$$

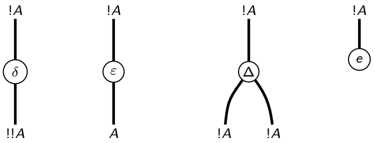
The contextual linear dagger is defined using the cups and caps:



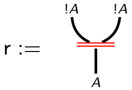
So a **monoidal reverse differential category** will be a monoidal differential category which is also a self-dual compact closed category. But can we define the **reverse differential** separately? YES!

Monoidal Reverse Differential Category

A **monoidal reverse differential category** is a self-dual compact closed [3] which is enriched over commutative monoids (so we can add maps $+$ and have zero maps 0) and equipped with a comonad $!$ such that each $!A$ is a cocommutative comonoid:



and also equipped with a **reverse deriving transformation** $\mathsf{r}_A : !A \otimes !A \to A$:



satisfying reverse analogues of the **deriving transformation** but with the adjoints of the comonad structure.

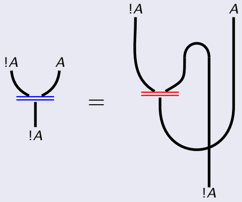
[3]Unlike in the Cartesian case, the transpose does not seem to come for free and we need to assume it in the definition...

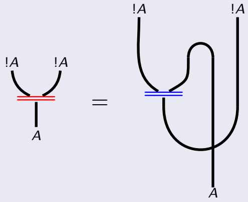
Monoidal Reverse Differential Category

Theorem

The following are equivalent:

- *Monoidal reverse differential category*
- *Monoidal differential category which is self-dual compact closed.*

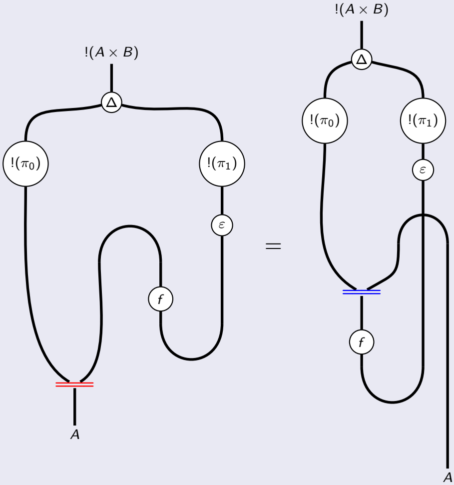




CoKleisli is a CRDC

Theorem

For a monoidal reverse differential category with finite product, its coKleisli category is a Cartesian reverse differential category where for a map $f : !A \to B$ its reverse derivative $\mathsf{R}[f] : !(A \times B) \to A$ is defined as follows:



Relation to Categorical Quantum

Surprisingly, monoidal reverse differential categories have some ties too categorical quantum!

- Selinger and Valiron's categorical model of a quantum lambda calculus (a programming language for quantum computation with classical control) is a monoidal reverse differential category.

  Selinger, P., & Valiron, B. *On a Fully Abstract Model for a Quantum Linear Functional Language.*

  In future work, it would be interesting to study in more detail the consequence of reverse differential structure in this model of quantum lambda calculus.

- Every monoidal reverse differential category where ! is the cofree coalgebra modality is a model of Vicary's categorical quantum harmonic oscillator.

  Vicary, J. *A categorical framework for the quantum harmonic oscillator.*

  In future work it would be interesting to revisit Vicary's categorical quantum harmonic oscillators from the point of view of (reverse) differential categories.

We can therefore apply automatic differentiation and machine learning algorithms like supervised learning to the coKleisli categories of these quantum models of programming.

Refinement

Other examples include:

- Sets and relations
- Finite dimensional vector spaces with chosen basis over $\mathbb{Z}_2$

But examples of monoidal reverse differential categories are not easy to find... Indeed self-dual compact closed models of (differential) linear logic in general are difficult to find because often ! has a infinite dimensional flavour while compact closed has a finite dimensional flavour...

Lemay, J-S. P. *Why FHilb is NOT an interesting (co)differential category*

Refinement

Other examples include:

- Sets and relations
- Finite dimensional vector spaces with chosen basis over $\mathbb{Z}_2$

But examples of monoidal reverse differential categories are not easy to find... Indeed self-dual compact closed models of (differential) linear logic in general are difficult to find because often ! has a infinite dimensional flavour while compact closed has a finite dimensional flavour...

Lemay, J-S. P. *Why FHilb is NOT an interesting (co)differential category*

Instead there's a way that every MDC gives a CRDC.

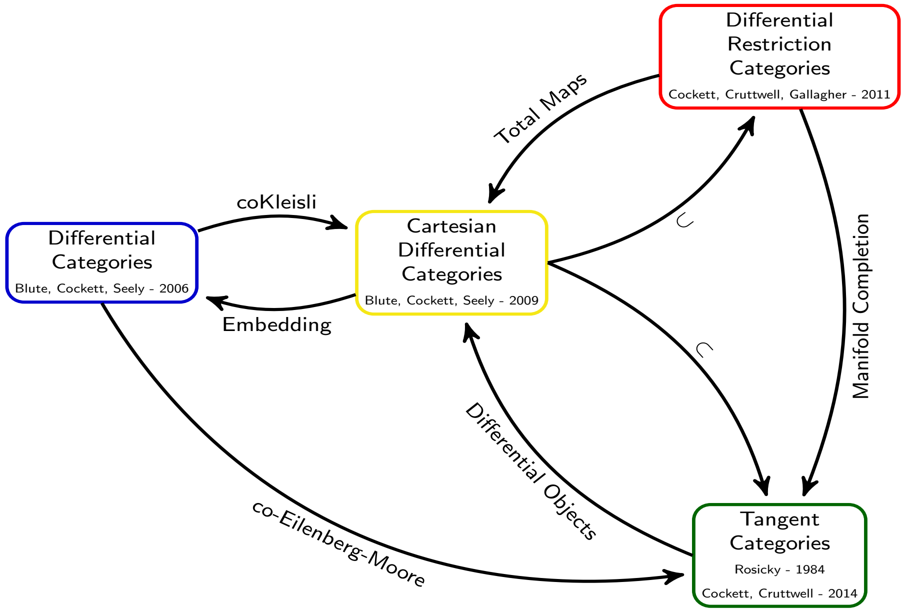
Theorem

For a monoidal differential category with finite products, the full subcategory of the coKleisli category whose objects are self-dual (in the base category) is a Cartesian reverse differential category.

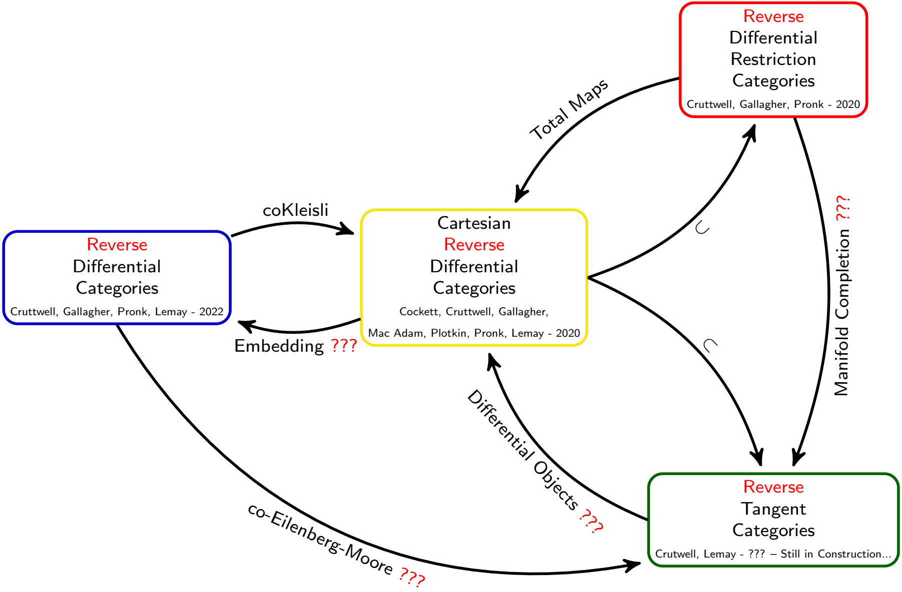
This construction recaptures some main examples of CRDC like polynomials and real smooth functions.

However the main goal of this story was to build a model where the entire coKleisli category was a CRDC. This essentially forces self-dual compact closed. Nevertheless, this gives suitable models of differential logic to apply machine learning techniques too.

The Differential Category World: It's all connected!



We want to map the world of Reverse Differential Categories



We want to map the world of Reverse Differential Categories