Coend Calculus via String Diagrams

The Coend approach to Incomplete Diagrams

Mario Román

April 4, 2020

Abstract

If we interpret the morphisms in a monoidal category as black boxes that can be composed sequentially and in parallel, we are faced with the problem of interpreting what non-square boxes ought to represent. We propose a description in terms of coends and the cartesian bicategory of profunctors, with features of what could be considered a graphical calculus for these coends.

1 Introduction

1.1 Coend Calculus

1.2 Open Diagrams

Morphisms in monoidal categories are interpreted as processes with inputs and outputs and generally represented by boxes. However, that raises the question of what to do if the process does not consume all the inputs at the same time or produces all the outputs at the same time. Consider for instance a process that consumes an input, produces an output, then consumes a second input and ends up producing an final output. Graphically, we have a clear idea of how this process should be represented, even if it is not a morphism in the category.

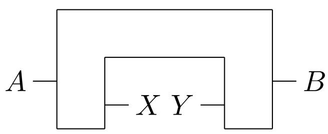


Figure 1: A process with a non-standard shape. The input A is taken at the beginning, then the output X is produced, strictly after that, the input Y is taken; finally, the output B is produced.

Reasoning graphically, it is obvious, for instance, that we should be able to *plug* a morphism connecting the first output to the second input inside this process and get back an actual morphism of the category.

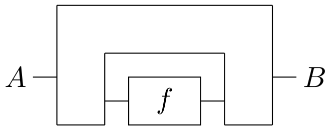
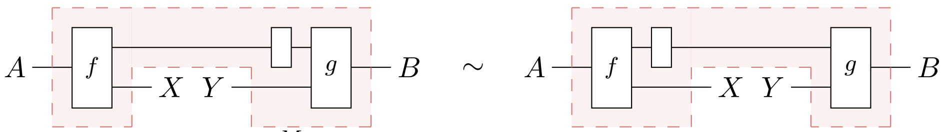


Figure 2: It is possible to plug a morphism $f\colon X \to Y$ inside the previous process (Figure 1).

The particular shape depicted above has been extensively studied by [Ril18] under the name of (monoidal) *optic*; and it has many applications in bidirectional data accessing [PGW17, Kme18]. It has appeared also in the work of Kissinger and Uijlen as a notational convention [KU17]. It can be shown that *boxes* of that shape should correspond to elements of a suitable *coend* (Figure 4, for an introduction to coend calculus see [Lor15]). The intuition under a coend representation is that one should consider a tuple of morphisms and then quotient out by an equivalence relation generated by all the wires that are connected between these morphisms.



$$\int^{M} \mathbf{C}(A, M \otimes X) \times \mathbf{C}(M \otimes Y, B).$$

Figure 3: A box of this shape is meant to represent a pair of morphisms in a monoidal category quotiented out by "sliding a morphism" over the upper wire. It coincides with the quotienting arising from the dinaturality of a coend.

It had remained unclear, however, how this process should be carried in full generality and if it was in solid ground. Are we being formal when we use *incomplete* diagrams? What happens with all the other possible shapes that one would want to consider in a monoidal category?[1] They are interesting on their own; for instance, the last one of these has three inputs and two outputs, but the first input cannot affect the last output; and the last input cannot affect the first one.

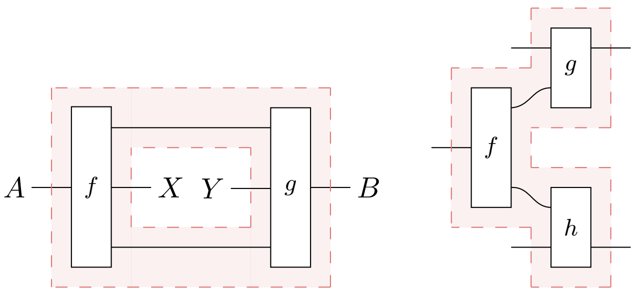


Figure 4: Some other shapes for boxes in a monoidal category.

This note starts from the idea that incomplete diagrams should be interpreted in the cartesian bicategory of profunctors. Compositions of incomplete diagrams corresponds to

[1]For the last of these shapes, this author thanks Nathaniel Virgo, who posed it as a question on `categorytheory.zulipchat.com`.

reductions that employ the cartesian bicategory structure.

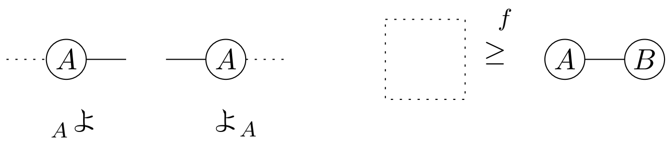
2 The Cartesian Bicategory of Profunctors

Definition 2.1. There exists a cartesian bicategory **Prof** of profunctors having as objects the (small) categories; as 1-cells from **A** to **B** the profunctors $\mathbf{A}^{op} \times \mathbf{B} \to \mathbf{Set}$; as 2-cells, the natural transformations; and as tensor product, the cartesian product of categories. Composition of two profunctors $P \colon \mathbf{A}^{op} \times \mathbf{B} \to \mathbf{Set}$ and $Q \colon \mathbf{B}^{op} \times \mathbf{C} \to \mathbf{Set}$ returns a profunctor $(P \diamond Q) \colon \mathbf{A}^{op} \times \mathbf{C} \to \mathbf{Set}$ defined by

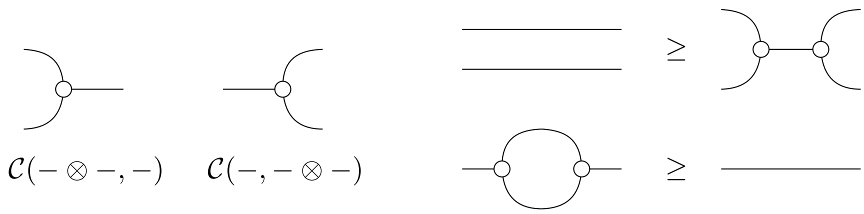
$$(P \diamond Q)(A, C) \coloneqq \int^{B \in \mathbf{B}} P(A, B) \times Q(B, C).$$

We will employ the graphical calculus of monoidal categories for the cartesian structure of **Prof**; and, at the same time, we will use inequalities between diagrams to denote the 2-cells of the bicategorical structure, which are natural transformations. This is in analogy with the graphical calculus for categories of relations.

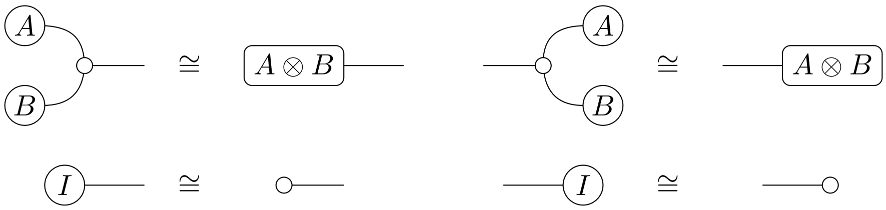
Definition 2.2 (Yoneda embeddings)**.** Every object $A \in \mathbf{C}$ determines two profunctors ${}_A\text{よ} \coloneqq \mathbf{C}(A, -) \colon \mathbf{1}^{op} \times \mathbf{C} \to \mathbf{Set}$ and $\text{よ}_A \coloneqq \mathbf{C}(-, A) \colon \mathbf{C}^{op} \times \mathbf{1} \to \mathbf{Set}$ called their contravariant and covariant Yoneda embeddings. Every morphism $f \in \mathbf{C}(A, B)$ can be seen as a natural transformation $f \colon \mathbf{1} \to {}_A\text{よ} \diamond \text{よ}_B$.



Definition 2.3. A **promonoidal category** is a pseudomonoid in the category **Prof**. A **procomonoidal category** is a pseudocomonoid in the category **Prof**. Every monoidal category is both promonoidal and procomonoidal. Moreover, the promonoidal structure is right adjoint to the procomonoidal structure.



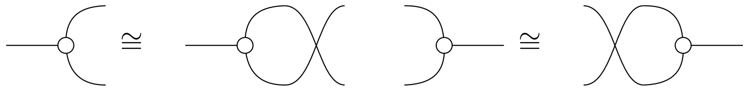
This monoidal structure interacts nicely with Yoneda embeddings in the following sense.



Remark 2.4. Scalars on this monoidal category are sets. We will be defining sets by using this graphical calculus.

Proposition 2.5. *Note that every object is a monoid and a comonoid in a canonical way in* **Prof***. A monoidal category is* ***cartesian*** *if and only if its promonoidal structure coincides with its canonical comonoid structure; a monoidal category is* ***cocartesian*** *if and only if its procomonoidal structure coincides with its canonical monoid structure.*

Proposition 2.6. *A category* **C** *is symmetric exactly when its probimonoidal structure is commutative.*

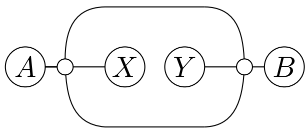


3 Examples of Open Diagrams

We have all the ingredients to start considering open diagrams. Let us consider the following examples. On the left we have a shape, on the right we have its description as an open shape. Lastly, as they are scalars, their interpretation is a set described as a coend.

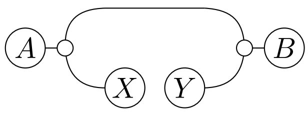
3.1 Motivating examples

Consider the examples that motivated the introduction.

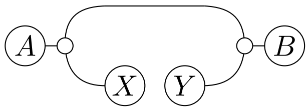


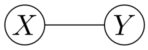
3.2 Lenses and optics

Definition 3.1. A monoidal lens [Ril18, "Optic" in Definition 2.0.1], from $A, B \in \mathbf{C}$ to $X, Y \in \mathbf{C}$ is an element of the following set.

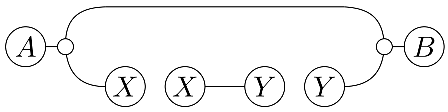


Example 3.2. As detailed in the introduction, a lens $(A, B) \to (X, Y)$ can be composed with a continuation $X \to Y$ to obtain a morphism $A \to B$. Let us illustrate this composition in the graphical calculus of **Prof**. It is also interpreted into the following chain of coend calculus, that describes that same composition.

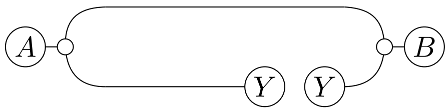




$\cong$

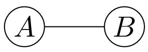


$\geq$



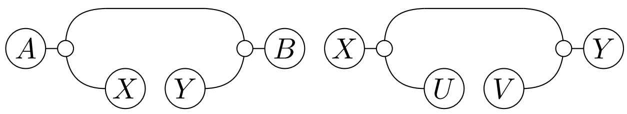
$\geq$

$\geq$

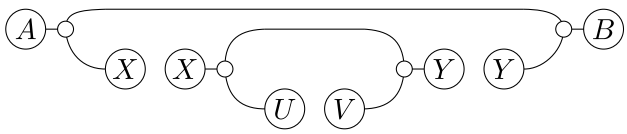


$$
\begin{aligned}
&\left(\int^{M} \mathbf{C}(A, M \otimes X) \times \mathbf{C}(M \otimes Y, B)\right) \times \mathbf{C}(X, Y) \\
\to\ & \\
&\int^{M} \mathbf{C}(A, M \otimes X) \times \mathbf{C}(M \otimes X, M \otimes Y) \times \mathbf{C}(M \otimes Y, B) \\
\to\ & \\
&\int^{M,N} \mathbf{C}(A, M \otimes N) \times \mathbf{C}(M \otimes N, B) \\
\to\ & \\
&\mathbf{C}(A, B).
\end{aligned}
$$

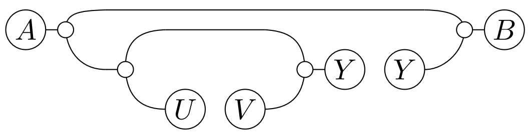
Example 3.3. Two lenses of types $(A, B) \to (X, Y)$ and $(X, Y) \to (U, V)$ can be also composed with each other.



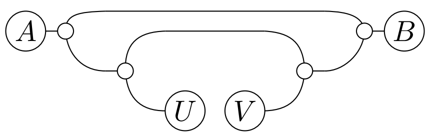
$\cong$



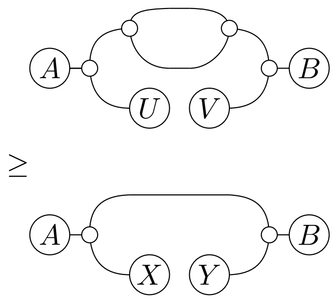
$\geq$



$\geq$



$\geq$

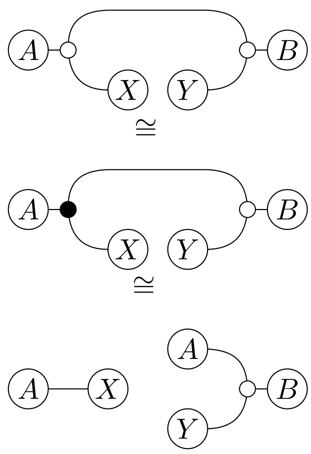


Example 3.4. There is, however, another way of composing two lenses. Let us write a lens $(A, B) \to (X, Y)$ as $(A, Y) \twoheadrightarrow (B, X)$. It can be composed with a lens $(B, X) \twoheadrightarrow (C, Z)$ into a lens $(A, B) \twoheadrightarrow (C, Z)$.

3.2.1 Cartesian lenses

Proposition 3.5. *In a cartesian category* $\mathbf{C}$*, a lens* $(A, B) \to (X, Y)$ *is given by a pair of morphisms* $\mathbf{C}(A, X)$ *and* $\mathbf{C}(A \times X, Y)$*. In a cocartesian category, these are called* prisms *[Kme18] and they are given by a pair of morphisms* $\mathbf{C}(S, A + T)$ *and* $\mathbf{C}(B, T)$*.*

Proof. We write the proof for lenses, the proof for prisms is dual. The original proof can be found in [Mil17].



$\square$

3.2.2 Algebraic lenses

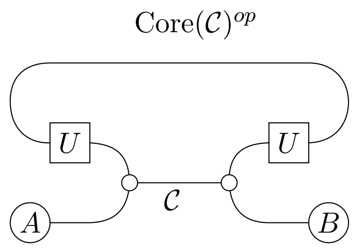
We have discussed the topic of monoidal lenses in the introduction, but there is a much more general notion of lenses, called optics. All of them can be nicely represented in the graphical calculus of **Prof** but, for simplicity, we will focus on a particular family of lenses that restrict the category over which the coend is taken. This showcases a technique that we consider important on its own: restricting the coend to slightly adjust a definition.

Definition 3.6. [CEG+20, Definition 3.9] Let $\mathbf{C}$ be a cartesian category and let $T\colon \mathbf{C} \to \mathbf{C}$ be a monad.

3.3 Circ construction

As one may expect, depicting diagrams in the category of profunctors gives us much more freedom than one would have drawing diagrams in the monoidal category. This extra freedom can be used to represent monoidal processes of different shapes, but in some sense it also allows us to identify any two wires. If we do not preserve the flow of time (understood as the order of sequential compositions) during this identification, we can introduce notions of feedback.

Definition 3.7. [KSW02, Definition 2.4] Let $\mathbf{C}$ be a monoidal category and let $U\colon \mathrm{Core}(\mathbf{C}) \to \mathbf{C}$ be the obvious inclusion functor from its core. For any two objects $X, Y \in \mathbf{C}$, we define a morphism in the $\mathsf{Circ}_{\mathbf{C}}$ construction to be an element of the following set.



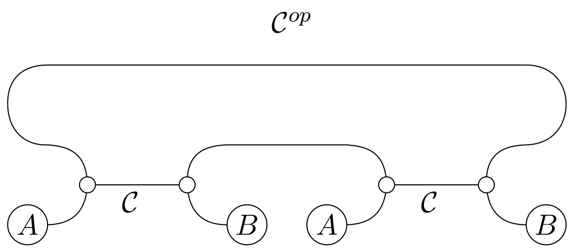
This is equivalently, an element of the following coend.

$$\mathsf{Circ}_{\mathbf{C}}(X, Y) := \int^{M \in \mathrm{Core}(\mathbf{C})} \mathbf{C}(M \otimes X, M \otimes Y).$$

3.4 Learners

On a recent article, Fong and Johnson [FJ19] have proposed an approach to machine learning based on category theory by constructing a monoidal category where morphisms represent supervised learning algorithms. In this category **Learn**, a morphism is given by the following data.

Definition 3.8. Let $(\mathbf{C}, \otimes, I)$ be a monoidal category. A (monoidal) **learner** taking inputs on an object A and producing outputs on an object B is an element of the following set.



This is a generalization to the monoidal case of an alternative definition proposed by Riley [Ril18]. It lets us view learners under a different light, even if it can be extremely different from the usual presentation. The original definition is conceptually clear as it splits the learner into the functions is should implement. Let us show both are equivalent in the case of **Set**.

Proposition 3.9. *A* ***learner*** *taking inputs on a set A and producing outputs on a set B is given by*

- *a set of* parameters P,
- *an* implementation *function* $i \colon P \times A \to B$,
- *an* update *function* $u \colon P \times A \times B \to P$, *and*
- *a* request *function* $r \colon P \times A \times B \to A$.

Let $(\mathbf{C}, \otimes, I)$ *be a monoidal category and consider shapes of the following form.*

$$\int^{P,Q \in \mathbf{C}} \mathbf{C}(P \otimes A, Q \otimes B) \times \mathbf{C}(Q \otimes A, P \otimes B)$$

In the case of cartesian categories, this coincides with a learner.

Proof. The following derivation appears, in the case of **Set**, in [Ril18, Definition 6.4.1]. It is included here for the sake of completeness. □

4 Related work

- Relational theories and graphical regular logic. We follow the same idea one dimension up.
- Combs as in [Rom20], and the fact that they can be structured on an operad as claimed by Jules Hedges[2]. The present approach is strictly more general than considering combs, as *combs* assume the symmetric structure of the underlying monoidal category.

5 Conclusions

The calculus of the category **Prof** seems particularly practical to accomodate open systems; being backed up at the same time with a nice formalism in terms of coend calculus [Lor15].

There is an important shortcoming to this approach: this allows us to compute using coend calculus, but not (co)end calculus. In other words, *ends* are missing from the picture; and this is expected. As it happens with regular logic, only the existential quantifier is recovered.

5.1 Three dimensional calculus

The graphical calculus we are using is three-dimensional in nature. It is only because of simplicity and technical constraints that we choose to work in some form of sliced bidimensional graphical calculus. It does not need to be this way, `homotopy.io` is a proof assistant supporting n-dimensional diagrammatic calculus that can render three-dimensional diagrams [HHV19].

[2] Personal communication.

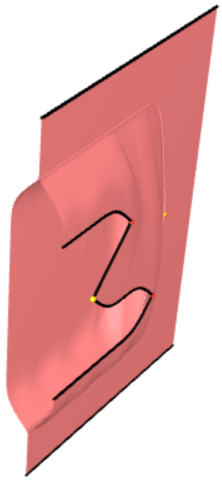


Figure 5: A monoidal lens is composed with a particular continuation. It is similar (except for the choice of continuation) to the construction in Example 3.2.

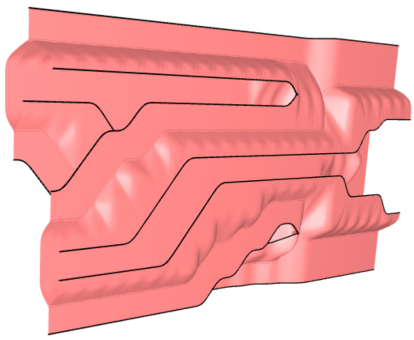


Figure 6: Two lenses composing, admittedly less clearly than the previous one.

6 Acknowledgements

Special thanks go to Fosco Loregian for sharing the art of (co)end-fu in [Lor15].

Mario Román was supported by the European Union through the ESF funded Estonian IT Academy research measure (project 2014-2020.4.05.19-0001).

References

[CEG+20] Bryce Clarke, Derek Elkins, Jeremy Gibbons, Fosco Loregian, Bartosz Milewski, Emily Pillmore, and Mario Román. Profunctor optics, a categorical update. *arXiv preprint arXiv:1501.02503*, 2020.

[FJ19] Brendan Fong and Michael Johnson. Lenses and Learners. *CoRR*, abs/1903.03671, 2019.

[HHV19] Lukas Heidemann, Nick Hu, and Jamie Vicary. *homotopy.io*. 2019.

[Kme18] Edward Kmett. lens library, version 4.16. Hackage https://hackage.haskell.org/package/lens-4.16, 2012–2018.

[KSW02] Piergiulio Katis, Nicoletta Sabadini, and Robert F. C. Walters. Feedback, trace and fixed-point semantics. *ITA*, 36(2):181–194, 2002.

[KU17] Aleks Kissinger and Sander Uijlen. A categorical semantics for causal structure. In *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12. IEEE, 2017.

[Lor15] Fosco Loregian. This is the (co)end, my only (co)friend. *arXiv preprint arXiv:1501.02503*, 2015.

[Mil17] Bartosz Milewski. Profunctor optics: the categorical view. https://bartoszmilewski.com/2017/07/07/profunctor-optics-the-categorical-view/, 2017.

[PGW17] Matthew Pickering, Jeremy Gibbons, and Nicolas Wu. Profunctor optics: Modular data accessors. *Programming Journal*, 1(2):7, 2017.

[Ril18] Mitchell Riley. Categories of Optics. *arXiv preprint arXiv:1809.00738*, 2018.

[Rom20] Mario Román. Comb diagrams for discrete-time feedback, 2020.