Coend Calculus via String Diagrams

The Coend Approach to Open Diagrams

Mario Román

April 5, 2020

Abstract

Morphisms in a monoidal category are usually as *processes* or *black boxes* that can be composed sequentially and in parallel. In practice, we are faced with the problem of interpreting what non-square boxes ought to represent and, more importantly, how should they be composed. Examples of this situation include *lenses* or *learners*. We propose a description of these non-square boxes, which we call open diagrams, in terms of coends and the cartesian bicategory of profunctors, with features of what could be considered a graphical calculus for these coends. It allows us to describe possible compositions of these open diagrams but also to reason about their concrete descriptions. This is work in progress.

1 Introduction

1.1 Open Diagrams

Morphisms in monoidal categories are interpreted as processes with inputs and outputs and generally represented by square boxes. This interpretation, however, raises the question of what to do if the process does not consume all the inputs at the same time or if it does not produce 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 producing an output. Graphically, we have a clear idea of how this process should be represented, even if it is not a morphism in the category.

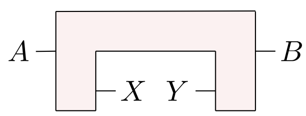


Figure 1: A process with a non-standard shape. The input A is taken at the beginning, then the output X is produced, strictly after that, the input Y is taken; finally, the output B is produced.

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.

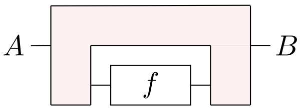


Figure 2: It is possible to plug a morphism $f\colon X \to Y$ inside the previous process (Figure 1), and, importantly, get back a morphism $A \to B$.

The particular shape depicted above has been extensively studied by [Ril18] under the name of (monoidal) *optic*; it can be also called a *monoidal lens*; and it has applications in bidirectional data accessing [PGW17, Kme18] or compositional game theory [GHWZ18]. A multi-legged generalization has appeared also in the work on causality of Kissinger and Uijlen as a notational convention [KU17, Rom20]. It can be shown that *boxes* of that shape should correspond to elements of a suitable *coend* (Figure 5, see also §1.2 and [Mil17]). The intuition for this 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.

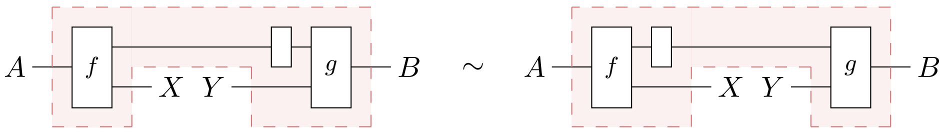


Figure 3: A box of this shape is meant to represent a pair of morphisms in a monoidal category quotiented out by "sliding a morphism" over the upper wire.

It has 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 these *open* or *incomplete* diagrams? What happens with all the other possible shapes that one would want to consider in a monoidal category? They are different from the usual squares; for instance, the second one of the shapes in Figure 4 has three inputs and two outputs, but the first input cannot affect the last output; and the last input cannot affect the first one.[1]

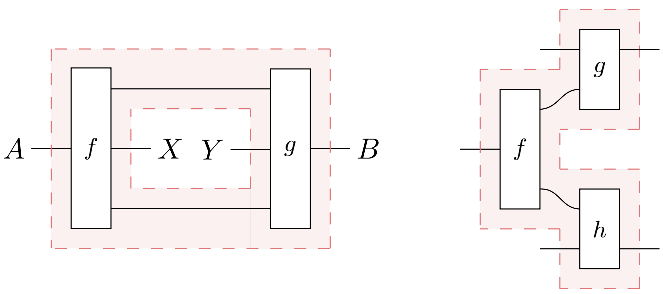


Figure 4: Some other shapes for boxes in a monoidal category.

This note presents the idea that incomplete diagrams should be interpreted in the cartesian bicategory of profunctors; compositions of incomplete diagrams correspond to reductions that employ the cartesian bicategory structure. At the same time, this gives a graphical presentation of *coend calculus*.

[1] This particular shape comes from a question by Nathaniel Virgo on `categorytheory.zulipchat.com`.

1.2 Coend Calculus

Coends are particular cases of colimits. *Coend calculus* is practical formalism that uses the Yoneda lemma to compute with them. Their dual counterparts are *ends*, and formalisms for both interact nicely in a *(Co)End calculus* [Lor15].

Definition 1.1. The **coend** $\int^{X \in \mathbf{C}} P(X,X)$ of a profunctor $P \colon \mathbf{C}^{op} \times \mathbf{C} \to \mathbf{Set}$ is the universal object endowed with morphisms

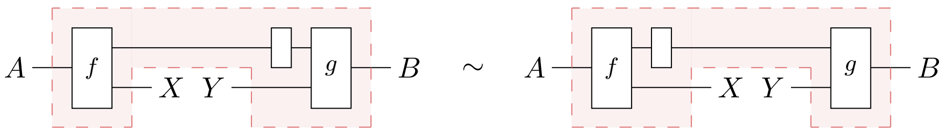
$$i_A \colon P(A,A) \to \left(\int^{X \in \mathbf{C}} P(X,X) \right)$$

for every $A \in \mathbf{C}$ such that, for any morphism $f \colon B \to A$ in $\mathbf{C}$, they satisfy $i_B \circ P(f, \mathrm{id}) = i_A \circ P(\mathrm{id}, f)$. It is universal in the sense that any other object D endowed with morphisms $j_A \colon P(A,A) \to D$ satisfying the same condition factors uniquely through it.

In other words, the coend is the coequalizer of the action on morphisms on both arguments of the profunctor. An element of the coend is an equivalence class of pairs $[X, x \in P(X,X)]$ quotiented by the equivalence relation generated by $[X, P(f,-)(u)] \sim [Y, P(-,f)(u)]$.

$$\int^{X \in \mathbf{C}} P(X,X) \cong \mathrm{coeq} \left(\textstyle\bigsqcup_{f \colon B \to A} P(A,B) \rightrightarrows \bigsqcup_{X \in \mathbf{C}} P(X,X) \right).$$

Our plan is to use these quotient relations to deal with the naturality arising in non-square monoidal shapes.



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

Figure 5: We go back to the previous example to check how it coincides with the quotienting arising from the dinaturality of a coend.

1.3 Contributions

- A description of how to use the graphical calculus of the monoidal bicategory of profunctors to reason about *open diagrams* and *coend calculus*, paying special attention to how the monoidal structure of a category is represented with profunctors (§2 and §3.1 3.1).
- A study of the multiple ways of composing monoidal lenses, and a recast of some coend calculus constructions on the literature on optics in terms of monoidal categories (§3.2).
- A description of how some two constructions in the literature are instances of *open diagrams* and can be described in terms of coends. These are the Circ construction [KSW02] (§3.3) and *learners* [FJ19] (§3.4).

2 The Monoidal Bicategory of Profunctors

Definition 2.1. There exists a monoidal bicategory **Prof** having as objects the (small) categories; as 1-cells from $\mathbf{A}$ to $\mathbf{B}$ 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 [Lor15]. 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

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

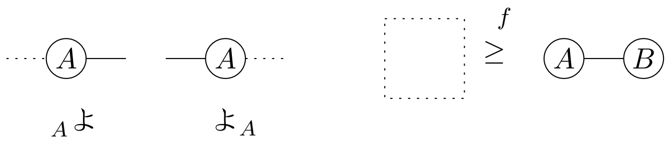
The monoidal product of two profunctors $P_1\colon \mathbf{A}_1^{op} \times \mathbf{B}_1 \to \mathbf{Set}$ and $P_2\colon \mathbf{A}_2^{op} \times \mathbf{B}_2 \to \mathbf{Set}$ is the profunctor $(P_1 \otimes P_2)\colon (\mathbf{A}_1 \times \mathbf{A}_2)^{op} \times (\mathbf{B}_1 \times \mathbf{B}_2) \to \mathbf{Set}$ defined by

$$(P_1 \otimes P_2)(A_1, A_2, B_1, B_2) \coloneqq P_1(A_1, B_1) \times P_2(A_2, B_2).$$

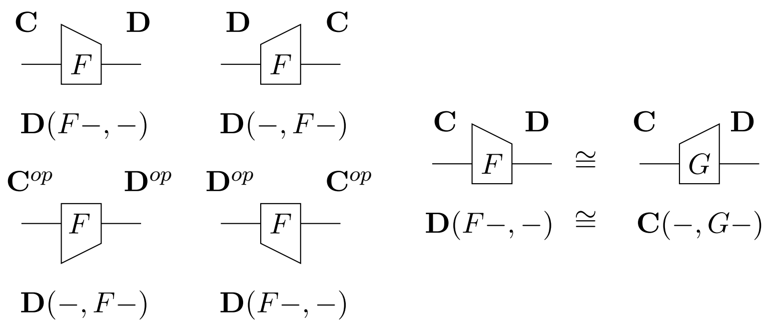
We will employ the graphical calculus of monoidal categories for the cartesian structure of **Prof**; and, at the same time, we will use inequalities between diagrams to denote the 2-cells of the bicategorical structure, which are natural transformations. This is in analogy with the graphical calculus for categories of relations, but note that the category is not posetal.

Remark 2.2. Scalars in this monoidal category are sets. We will be defining sets by using this graphical calculus during the rest of the text.

Definition 2.3 (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 $f\colon \mathbf{1} \to {}_A\text{よ} \diamond \text{よ}_B$.

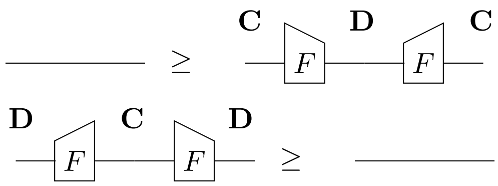


Definition 2.4 (Yoneda embedding of functors). Let $F\colon \mathbf{C} \to \mathbf{D}$ be a functor. It can be embedded into profunctors in two ways, as a profunctor $\mathbf{D}(F-, -)\colon \mathbf{C}^{op} \times \mathbf{D} \to \mathbf{Set}$ or as a profunctor $\mathbf{D}(-, F-)\colon \mathbf{D}^{op} \times \mathbf{C} \to \mathbf{Set}$. We denote them with the following graphical notation and we note the particular case of what it means to be an adjoint functor.



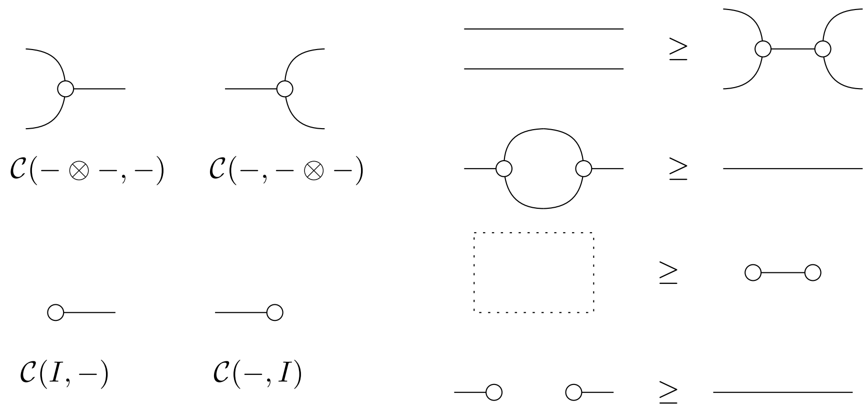
The suggestive shape of the boxes is actually matched by their semantics. Every category has a dual (namely, its opposite category) and functors circulate as expected through the cups and the caps that represent dualities.

Proposition 2.5. *In the category of profunctors, functors are themselves adjoints, in the following sense.*

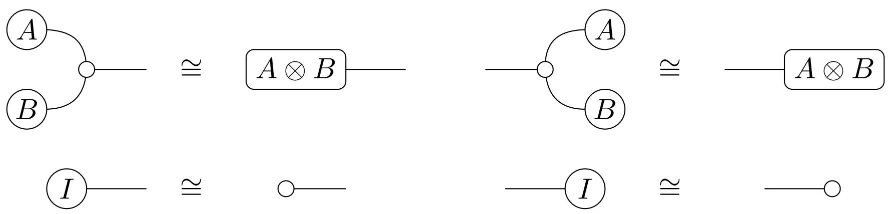


2.1 Monoidal structure

Definition 2.6. 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, because the monoidal product is a functor, the promonoidal structure is adjoint to the procomonoidal structure.

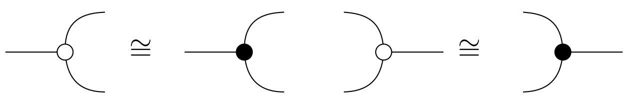


This monoidal structure interacts nicely with Yoneda embeddings precisely because it is a functor.



2.2 Cartesian and cocartesian categories

Proposition 2.7. *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.*

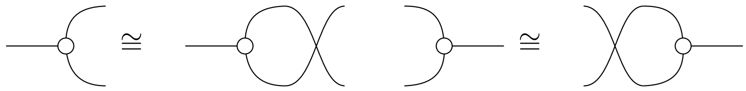


(Cartesian) *(Cocartesian)*

During this text, we use the black dots to denote the copying and comparing structure of **Prof***. We keep the white dots to represent monoidal structures of a category.*

2.3 Symmetric monoidal categories

Proposition 2.8. *A monoidal category* **C** *is* symmetric *exactly when the pseudomonoid giving its monoidal structure is commutative.*



3 Examples of Open Diagrams

We have all the ingredients to start considering open diagrams. Let us consider the following examples. 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 (in Figure 4). By writing them again with the graphical calculus of the cartesian bicategory **Prof**, we can obtain formulaic descriptions of the shapes, but also compose them in different ways.

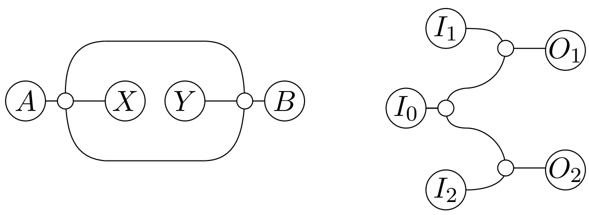


Figure 6: The shapes of Figure 4, interpreted as coends.

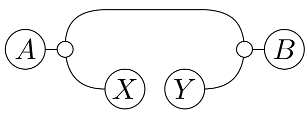
The corresponding coend descriptions are as follows.

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

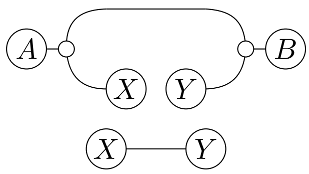
$$\int^{M,N} \mathbf{C}(I_0, M \otimes N) \times \mathbf{C}(I_1 \otimes M, O_1) \times \mathbf{C}(N \otimes I_2, O_2).$$

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.

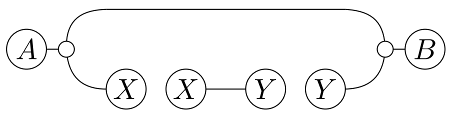


Example 3.2. As detailed in the introduction, a lens $(A, B) \to (X, Y)$ can be composed with a continuation $X \to Y$ to obtain a morphism $A \to B$. Let us illustrate this composition in the graphical calculus of **Prof**. It is also interpreted into the following chain of coend calculus, that describes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



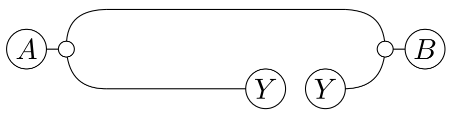
$$\left(\int^{M} \mathbf{C}(A, M \otimes X) \times \mathbf{C}(M \otimes Y, B)\right) \times \mathbf{C}(X, Y)$$

$\cong \qquad \cong$



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

$\geq \qquad \to$

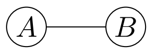


$$\int^{M} \mathbf{C}(A, M \otimes Y) \times \mathbf{C}(M \otimes Y, B)$$

$\geq \qquad \to$

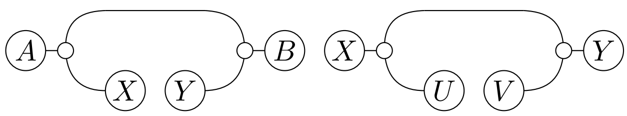
$$\int^{M,N} \mathbf{C}(A, M \otimes N) \times \mathbf{C}(M \otimes N, B)$$

$\geq \qquad \to$

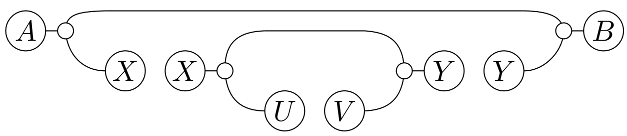


$$\mathbf{C}(A, B)$$

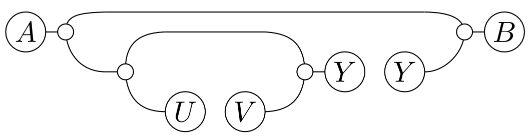
Example 3.3. Two lenses of types $(A, B) \to (X, Y)$ and $(X, Y) \to (U, V)$ can be also composed with each other.



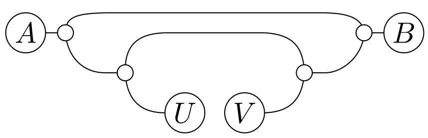
$\cong$



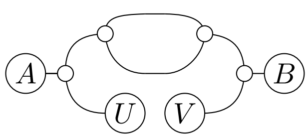
$\geq$



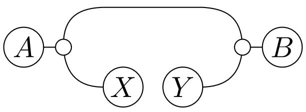
$\geq$



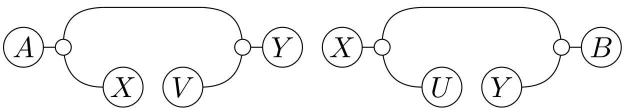
$\geq$



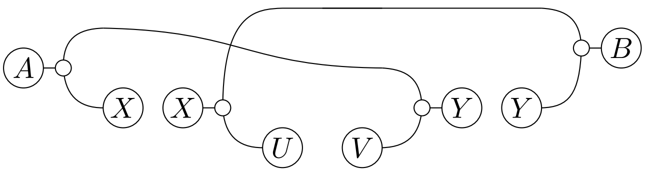
$\geq$



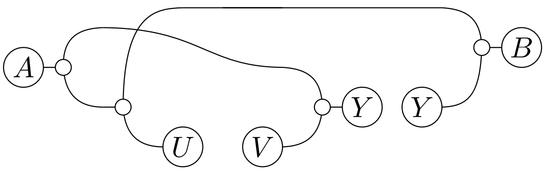
Example 3.4. There is, however, another way of composing two lenses when the category is symmetric. A lens $(A, Y) \to (X, V)$ can be composed with a lens $(X, B) \to (U, Y)$ into a lens $(A, B) \to (U, Y)$. It can be composed with a lens $(B, X) \to (C, Z)$ into a lens $(A, B) \to (C, Z)$. As we will see during its construction, this is an option only when the category is symmetric.



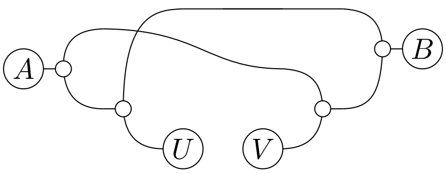
$\cong$



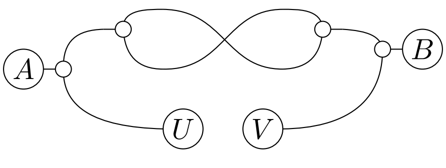
$\to$



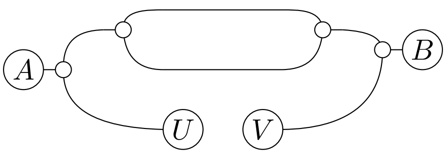
$\to$



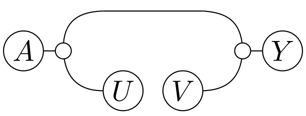
$\to$



$\cong$



$\to$

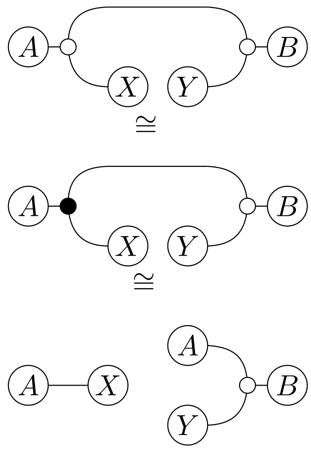


That is, even when the category **Prof** is symmetric; in order to produce a valid composition into a new lens, one explicitly needs symmetry on the base category $\mathbf{C}$.

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 $\mathbf{C}(A \times Y, B)$. 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].

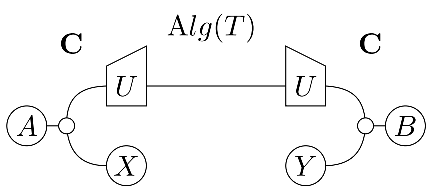


□

3.2.2 Algebraic lenses

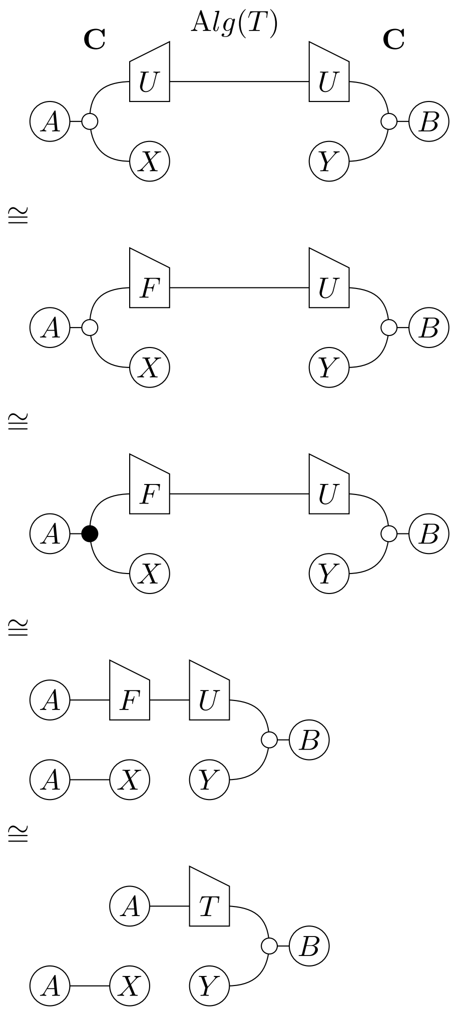
We have discussed the topic of monoidal lenses in the introduction, but there is a much more general notion of lenses, called optics. All of them can be nicely represented in the graphical calculus of **Prof** but, for simplicity, we will focus on a particular family of lenses that restrict the category 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; let $F \colon \mathbf{C} \to \mathrm{Alg}(T)$ and $U \colon \mathrm{Alg}(T) \to \mathbf{C}$ be the free and forgetful functors to the Eilenberg-Moore category, respectively. An algebraic lens $(A, B) \to (X, Y)$ is an element of the following set.



Proposition 3.7. *[CEG$^+$20, Proposition 3.10] In a cartesian category, an algebraic lens is equivalent to a pair of functions $\mathbf{C}(A, X)$ and $\mathbf{C}(TA \times Y, B)$.*

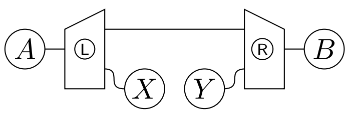
Proof.



□

3.2.3 Mixed optics

For the specialized reader, it may be of interest to finally consider the definition of a **mixed optic**, the generalized version of an optic. Let $\mathbf{M}$ a monoidal category acting with two monoidal actions Ⓛ: $\mathbf{M} \times \mathbf{C} \to \mathbf{C}$ and Ⓡ: $\mathbf{M} \times \mathbf{D} \to \mathbf{D}$. A mixed optic $(A, B) \to (X, Y)$ is an element of the following set.

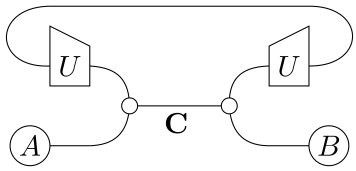


3.3 Circ construction

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 3.8. [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.

$\mathrm{Core}(\mathbf{C})^{op}$



This is equivalently, an element of the following coend.

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

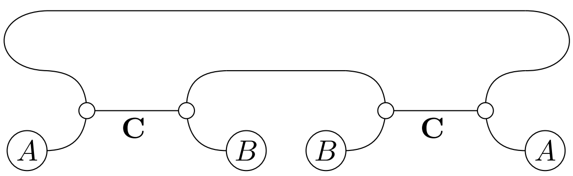
As we can see this is a much compressed description of the free category with feedback. It is also arguable that it makes intuitive sense: we are literally adding the feedback as a wire, and the fact that only isomorphisms travel the feedback reflects in the choice of wire. From this description, is easy to see that the Circ construction can be slightly modified into multiple variants by just changing the category over which we take the coend, in the same style we did for algebraic lenses.

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 3.9. 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.

$\mathbf{C}^{op}$



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.

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

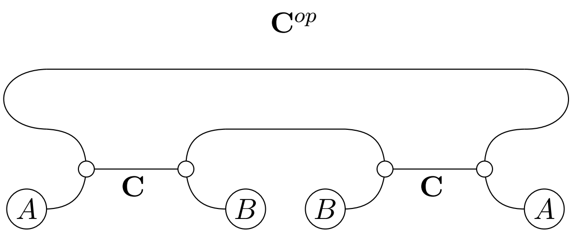
The original definition is conceptually clearer as it splits the learner into the functions is should implement. Let us show both are equivalent in the case of **Set**.

Proposition 3.10. *[FJ19, Definition 4.1] A* ***learner*** *taking inputs on a set A and producing outputs on a set B is given by*

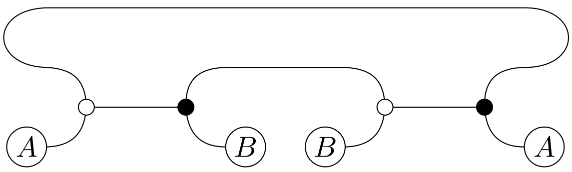
- *a set of* parameters P*,*
- *an* implementation *function* $i \colon P \times A \to B$*,*
- *an* update *function* $u \colon P \times A \times B \to P$*, and*
- *a* request *function* $r \colon P \times A \times B \to A$*.*

Let $(\mathbf{C}, \otimes, I)$ be a monoidal category and consider shapes of the following form. In the case of cartesian categories, this coincides with a learner.

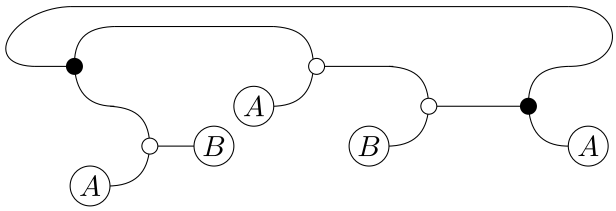
Proof. The following coend derivation appears, in the case of **Set**, in [Ril18, Definition 6.4.1]. It is included here for the sake of completeness and the slight generalization, but also to compare how it works under the graphical calculus.



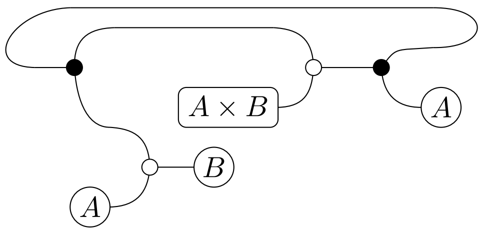
$\cong$ ($\mathbf{C}$ is cartesian)



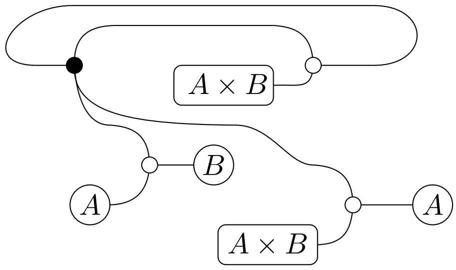
$\cong$ (Split)



$\cong$ (Monoidal product)



$\cong$ (Split)



This is the definition in terms of parameters, implementation, update and request. The looping wire represents the parameters quotiented by a coend. □

4 Related work

- The cartesian bicategory of relations, as used in [BPS17], and graphical regular logic [FS18] are arguably presenting a decategorification of this same idea.
- Finite *combs*, as described in [Rom20] (and previously by [CDP08] and [KU17]), can be combined in many different ways with this technique. We are presenting a way of describing these possible compositions. The fact that they can be nicely structured on an operad has been claimed by Jules Hedges in personal communication. Note however, that present approach is strictly more general than considering combs, as *combs* assume the symmetric structure of the underlying monoidal category, whereas the present technique can express arbitrary shapes that do not assume symmetry. Moreover, symmetry can be additionally required when needed (see, for instance, the case in Remark 3.2).

5 Conclusions

We have presented a way to study and compose *black boxes* in monoidal categories that do not necessarily have the usual shape of a square box. This allows us to consider processes with non-trivial dependencies between inputs and outputs. The calculus of the category **Prof** seems particularly practical to accomodate descriptions of open systems without losing the usual language of monoidal categories.

This technique is backed up at the same time with a practical formalism in terms of coend calculus [Lor15]. We can argue that the graphical representation of coend calculus is helpful to its understanding. Constrasting with usual presentations of coends that are usually centered around the Yoneda reductions; the graphical approach seems to put more weight in the non-reversible transformations and make most applications of Yoneda lemma transparent.

Finally, there is an important shortcoming to this approach that we leave as further work: the present graphical calculus allows us to compute using *coend calculus*, but not *(co)end calculus*. In other words, *ends* are missing from the picture; and it seems as this should be expected. As it happens with regular logic, only the existential quantifier is recovered. Diagrammatic approaches to obtaining the universal quantifier in a situation like this go back to Peirce and are described by Haydon and Sobociński [HS20].

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 two-dimensional 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].

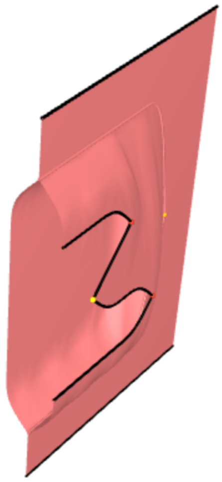


Figure 7: A monoidal lens is composed with a particular continuation. It is similar (except for the choice of continuation) to the construction in Example 3.2.

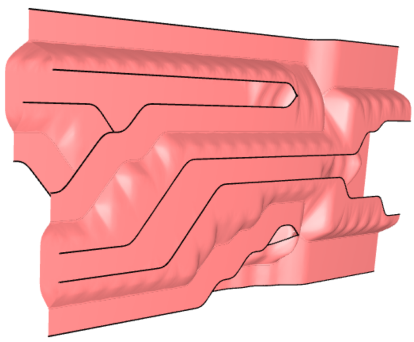


Figure 8: Two lenses composing. At the left end, we can see the figure of the two lenses. They get merged in the middle. At the right end, we can see a single lens.

6 Acknowledgements

Thanks go to whole Compositional Systems and Methods group at TalTech for discussions on cartesian bicategories that arrived just at the right moment (including a seminar on these by Jens Seeber and the lucid account of coend calculus of Fosco Loregian [Lor15]). A question by Nathaniel Virgo motivates the last of the examples in Figure 4.

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

[BPS17] Filippo Bonchi, Dusko Pavlovic, and Paweł Sobociński. Functorial semantics for relational theories. *CoRR*, abs/1711.08699, 2017.

[CDP08] G. Chiribella, G. M. D'Ariano, and P. Perinotti. Quantum Circuits Architecture. *Physical Review Letters*, 101(6), Aug 2008.

[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.

[FS18] Brendan Fong and David I. Spivak. Graphical regular logic. *CoRR*, abs/1812.05765, 2018.

[GHWZ18] Neil Ghani, Jules Hedges, Viktor Winschel, and Philipp Zahn. Compositional Game Theory. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 472–481, 2018.

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

[HS20] Nathan Haydon and Paweł Sobociński. Compositional diagrammatic first-order logic. *In Peer Review*, 2020.

[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.

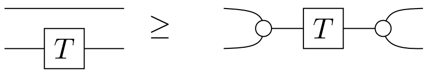
[Rom20] Mario Román. Comb diagrams for discrete-time feedback. *arXiv preprint arXiv:2003.06214*, 2020.

7 Appendix

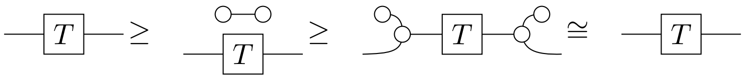
7.1 The Profunctor Representation Theorem

During the following section we fix a pair of actions Ⓛ: $\mathbf{M} \times \mathbf{C} \to \mathbf{C}$ and Ⓡ: $\mathbf{M} \times \mathbf{D} \to \mathbf{D}$. They are both, together with the monoidal structure of $\mathbf{M}$, to be represented by a white dot.

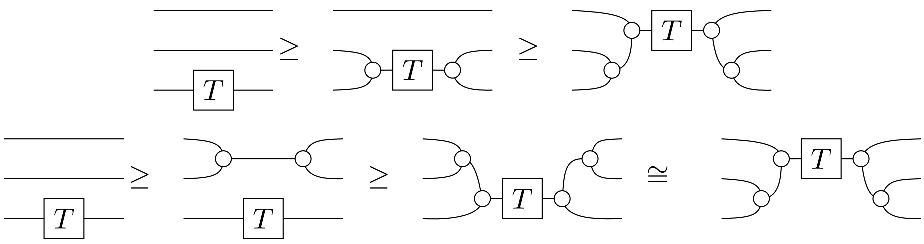
Definition 7.1 (Tambara module)**.** A **Tambara module** is a profunctor $T \colon \mathbf{C}^{op} \times \mathbf{D} \to$ **Set** equipped with a natural transformation as follows.



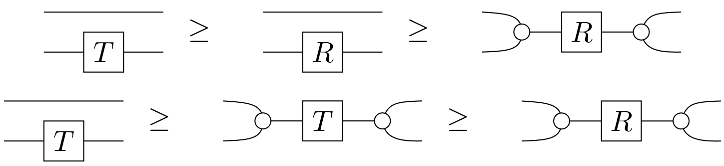
Such that the following morphism is the identity.



And such that the following two morphisms coincide

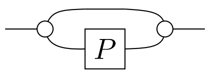


Definition 7.2 (Morphism of Tambara modules)**.** A morphism of Tambara modules $T \to R$ is a natural transformation between the profunctors such that the following two morphisms coincide.

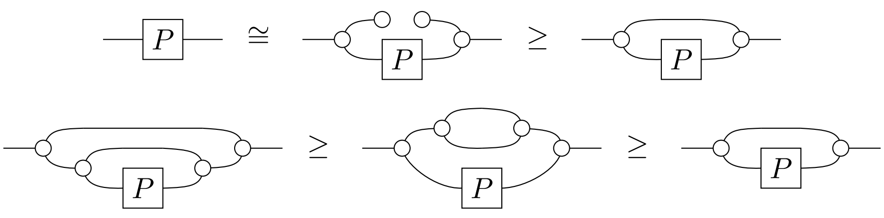


Tambara modules form a category. It can be checked that the composition of two morphisms of Tambara modules is a morphism of Tambara modules.

Definition 7.3 (Pastro-Street monad)**.** Consider the functorial assigment that maps a profunctor P to the following profunctor.



It is a monad with the unit and multiplication given by the following morphisms.

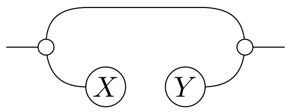


Proof. □

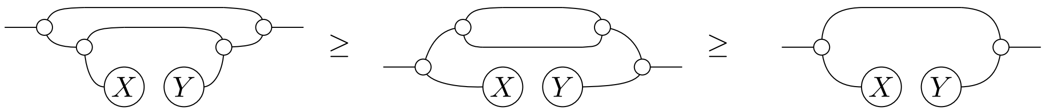
Theorem 7.4. *Tambara modules are the algebras of the Pastro-Street monad.*

Proof. □

Lemma 7.5. *The following profunctor is a Tambara module.*

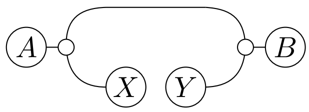


Proof. Its algebra morphism is given as follows. It remains to check the algebra axioms.

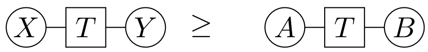


□

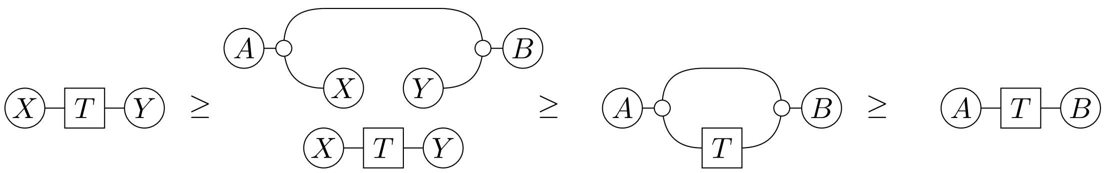
Theorem 7.6 (Profunctor Representation Theorem)**.** *There exists an isomorphism between elements of the following shape.*



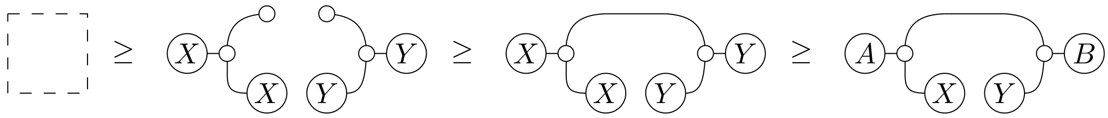
And families of morphisms of the following form natural on T a Tambara module.



Proof. Given an optic, we can construct the family of morphisms as follows.

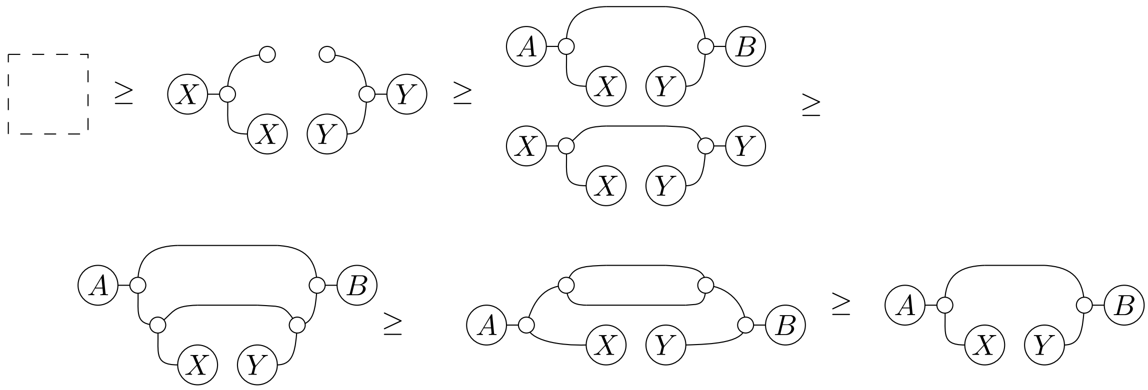


Given a family of morphisms, we can construct an optic as follows, using Lemma 7.5.

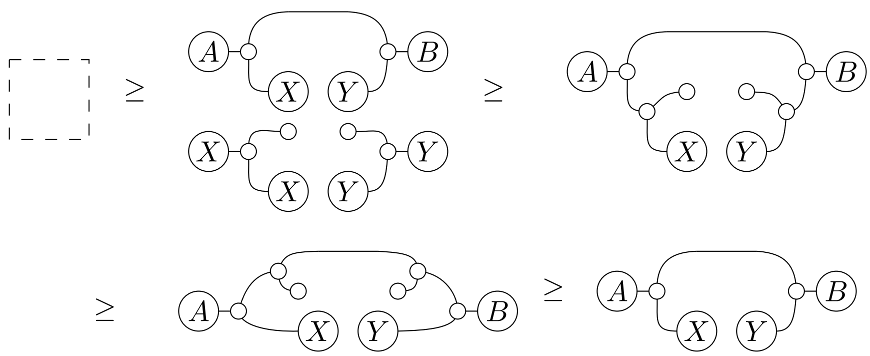


Given an optic, we can construct its family of morphisms and reconstruct the optic back.

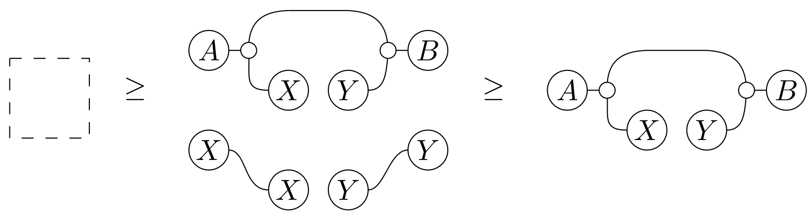
The following is the first slice of a three-dimensional proof.



Consider now the homotopic second slice.



Consider now the homotopic third slice, which is in turn homotopic to the original optic.



Now, given a family of morphisms, we can construct an optic and get back the family of morphisms again. $\square$