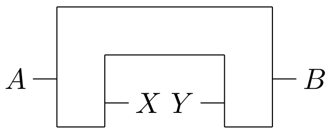
Monoidal Shapes Via Coends

Mario Román

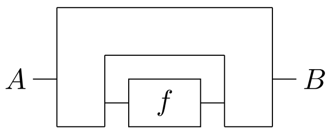
April 3, 2020

1 Introduction

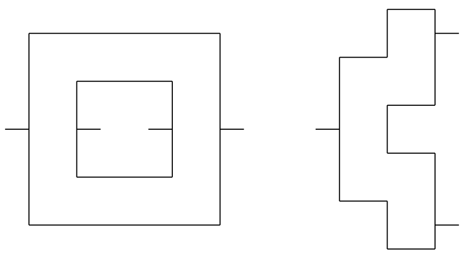
Morphisms in monoidal categories are interpreted as processes with inputs and outputs and generally represented by boxes. However, that raises the question of what to do if the process does not consume all the inputs at the same time or produces all the outputs at the same time. Consider for instance 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.



Reasoning graphically, it is obvious that we should be able to *plug* a morphism $X \to Y$ inside this process and get back an actual morphism of the category.



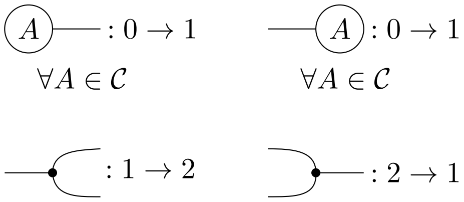
The particular shape depicted above has been extensively studied by [Ril18] under the name of *monoidal optic*; and it has many applications in bidirectional data accessing [PGW17, Kme18]. It can be shown that *boxes* of that shape should correspond to elements of a suitable *coend*. It remains unclear, however, how this process should be carried in full generality. What happens with all the other possible shapes that one would want to consider in a monoidal category?



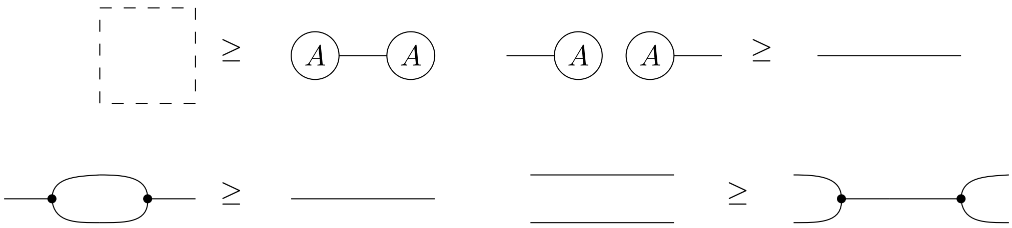
The main idea of this note is to consider a poset-enriched PRO of *open shapes,* with semantics in the cartesian bicategory of profunctors.

2 The PRO of Open Shapes

Definition 2.1. Let $(\mathcal{C}, \otimes, I)$ be a monoidal category. We define the PRO of open monoidal shapes over $\mathcal{C}$ as the one freely generated from the following set of generators.



We impose the following 2-cells



We say that a **monoidal shape** is a scalar in this PRO.

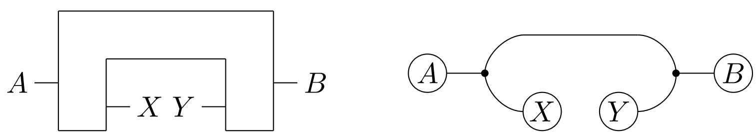
2.1 Semantics in profunctors

Definition 2.2. There exists a pseudofunctor from that 2-PRO to *Prof* as a cartesian bicategory. We construct a functor interpreting

- $よ_A\colon \mathcal{C} \to Set$, the covariant hom-functor;
- $よ^A\colon \mathcal{C}^{op} \to Set$, the contravariant hom-functor;
- $\mu\colon \mathcal{C}^{op} \times \mathcal{C}^{op} \times \mathcal{C} \to Set$, the promonoidal structure $\mathcal{C}(- \otimes -, -)$;
- $\delta\colon \mathcal{C}^{op} \times \mathcal{C} \times \mathcal{C} \to Set$, the promonoidal structure $\mathcal{C}(-, - \otimes -)$.

The inequalities are sent to the only natural morphisms with that type. The image of a shape under this functor is the **interpretation** of that shape.

Example 2.3. 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.



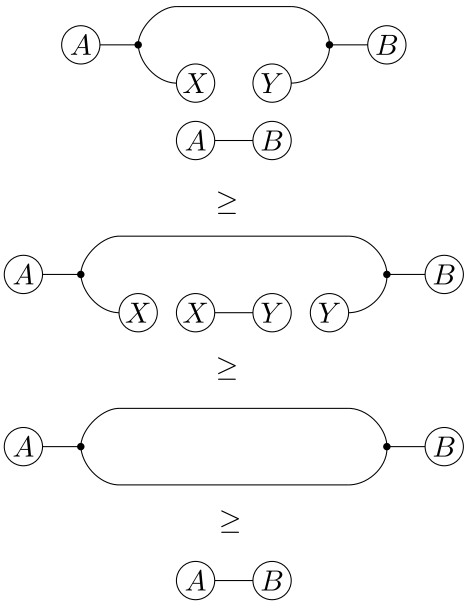
$$\int^{M} \mathcal{C}(S, M \otimes A) \times \mathcal{C}(M \otimes B, T)$$

Figure 1: Monoidal lenses as an open shape, they get interpreted to their formula.

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

Figure 2: The simple box is interpreted (as expected!) as a morphism.

The advantage of this interpretation is that we can describe compositions of shapes as reductions. Every time that we can combine two shapes into a new one, we get a way of composing their monoidal descriptions.



This gets interpreted into the following chain of functions, that describes that same composition.

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

3 Shapes with crossing

When the category is braided (or symmetric), we can add a braiding to our PRO (or start considering a PROP). The semantics of the crossing are given by the braiding (or symmetry) of the category.

4 Shapes with loops

We have much more expressivity in the category of profunctors. We can consider loops and get back *learners* or the *Circ* construction.

References

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

[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.