How Nice is this Functor?
Two Squares and Some Homology go a Long Way

Benjamin Merlin Bumpus
University of Florida

James Fairbanks
University of Florida

Fabrizio Genovese
20squares

Caterina Puca
Quantinuum
17 Beaumont Street
Oxford OX1 2NA, United Kingdom

Daniel Rosiak
NIST

We draw some squares and apply some homology in the hope of classifying different flavors of functors in a neat way. Our wishes are granted and we obtain homological measures of how much a given map can fail to be: (1) a pseudofunctor (2) a discrete Conduché functor and (3) a discrete fibration. By leveraging a known result of Bénabou, we furthermore obtain measures of how far certain lax functors are from being pseudofunctors. Finally, we apply the tools developed to (1) decorated Petri nets to classify how the execution of decorated nets differ, qualitatively, from the execution of undecorated ones, and (2) to delta lenses to study how far they differ from c-lenses.

1 Introduction

1.1. This paper is built on a bunch of simple observations. Given a(ny flavor of) category C, we can always consider the following diagram:

$$\cdots \;\; \mathsf{C}_3 \;\; \mathsf{C}_2 \;\; \mathsf{C}_1 \;\; \mathsf{C}_0$$

Here C_0 represents the objects of C, C_1 represents its morphisms, and C_n represents n-tuples of morphisms with matching domain and codomain (for example, $(A \xrightarrow{f} B, B \xrightarrow{g} C)$ could be an element of C_2). The arrows s,t represent the usual source and target assignments, whereas $\fatsemi$ represents composition. Notice that there are n different composition mappings from C_{n+1} to C_n, as we can choose to compose any two adjacent morphisms in a $n+1$-tuple to obtain a n-tuple – for instance, we can map C_3 to C_2 by sending the generic (f,g,h) to either $(f \fatsemi g, h)$ or to $(f, g \fatsemi h)$.

1.2. In the most standard setting (to which we will stick in this paper), the one above is a diagram of sets and functions[1], but we may definitely opt for something more complicated. In the case of double categories, for instance, it would be a diagram of categories and functors.

1.3. In any case, just looking at this diagram and asking some simple questions about its commutativity, one finds interesting facts:

- The diagram $\mathsf{C}_1 \underset{t}{\overset{s}{\rightrightarrows}} \mathsf{C}_0$ does not commute generally, and its equalizer is the set of *loops* of C, that is, morphisms $f : A \to A$ with the same source and target.

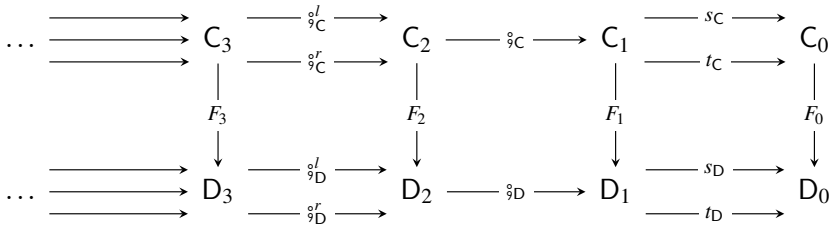
[1] Note that we are assuming all categories considered henceforward to be small. We will denote categories as C and double categories as $\mathbb{C}$. If there is no ambiguity, we will freely exchange C with $\mathbb{C}$, obtained from C by adding the needed trivial cells.

Submitted to:
ACT 2024

- Similarly $\mathsf{C}_2 \xrightarrow{\fatsemi} \mathsf{C}_1 \underset{t}{\overset{s}{\rightrightarrows}} \mathsf{C}_0$ does not generally commute, and its equalizer is the set of *loop decompositions* of C, that is, all the pairs of morphisms that give a loop when they are composed (e.g. $f : A \to B$ and $f' : B \to A$).

- The diagram $\mathsf{C}_3 \underset{\fatsemi^r}{\overset{\fatsemi^l}{\rightrightarrows}} \mathsf{C}_2 \xrightarrow{\fatsemi} \mathsf{C}_1$ commutes when C is a category, and amounts to state the familiar associativity condition $(f \fatsemi g) \fatsemi h = f \fatsemi (g \fatsemi h)$. When C is not a category – so for instance a non-trivial 2-category or bicategory – the equalizer of this diagram is the subset of morphism triples that are strictly associative.

1.4. Up to now, this is not incredibly interesting. But amazing things happen when we put two of these diagrams together. Given a functor $F : \mathsf{C} \to \mathsf{D}$, we can consider:



Here F_0 is F defined on objects, F_1 is F defined on morphisms, whereas any other F_n acts on n-tuples by applying F_1 component-wise.

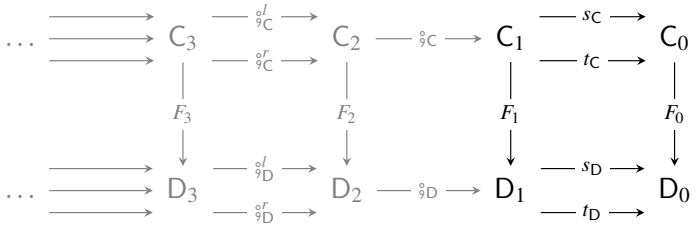
1.5. The purpose of this paper is focusing on these squares, in particular on the two rightmost ones, to say things about F. Using tools borrowed from homology, we will also be able to measure how much a functor fails to be 'something' – a discrete (op-)fibration, a pseudofunctor, ... – depending on context.

1.6. We shall note here that techniques to qualify obstructions to lax functors being strong have already been proposed in [27], where some special kind of directed posets, heavily inspired by homotopy theory, are introduced to detect failure of compositionality in the context of open systems.

1.7. This paper focuses on goal of qualifying obstructions to a range of properties a functor may satisfy, while applying homology directly to the categorical setting. This refines and extends the intuition (also motivating former work [27]) that topological holes and obstructions to category-theoretic compositionality shall be regarded as two sides of the same coin.

1.8. More generally, this work falls within the line of thought that compositionality, far from being a universal notion, is to be understood as a spectrum of distinct, context-dependent nuances [6, 11].

2 The rightmost square



2.1. We start by focusing at the rightmost square. Decoupling things a bit, we see how this actually consists of two different squares, merged:

$$\begin{array}{ccc} \mathsf{C}_1 & \xrightarrow{s_\mathsf{C}} & \mathsf{C}_0 \\ \downarrow{\scriptstyle F_1} & & \downarrow{\scriptstyle F_0} \\ \mathsf{D}_1 & \xrightarrow{s_\mathsf{D}} & \mathsf{D}_0 \end{array} \qquad \begin{array}{ccc} \mathsf{C}_1 & \xrightarrow{t_\mathsf{C}} & \mathsf{C}_0 \\ \downarrow{\scriptstyle F_1} & & \downarrow{\scriptstyle F_0} \\ \mathsf{D}_1 & \xrightarrow{t_\mathsf{D}} & \mathsf{D}_0 \end{array} \tag{1}$$

2.2. Since functors preserve source and target of morphisms, it follows that if $F : \mathsf{C} \to \mathsf{D}$ is a functor, then the two squares in Equation (1) commute. But there's more:

Definition 2.3. A functor $F : \mathsf{C} \to \mathsf{D}$ is a **discrete fibration** if for each $y \in \mathsf{C}_0$ and $f_\mathsf{D} : x' \to F_0(y)$, there exists a unique $f_\mathsf{C} : x \to y$ such that $F_1(f_\mathsf{C}) = f_\mathsf{D}$.

A functor $F : \mathsf{C} \to \mathsf{D}$ is a **discrete opfibration** if for each $x \in \mathsf{C}_0$ and $f_\mathsf{D} : F_0(x) \to y'$, there exists a unique $f_\mathsf{C} : x \to y$ such that $F_1(f_\mathsf{C}) = f_\mathsf{D}$.

Proposition 2.4. Let $F : \mathsf{C} \to \mathsf{D}$ be a functor. The right square in Equation (1) is a pullback square if and only if F is a discrete fibration. Similarly, the left square in Equation (1) is a pullback if and only if F is a discrete opfibration.

Proof. Let the right square be a pullback square. This means that

$$\mathsf{C}_1 \simeq \mathsf{D}_1 \times_{\mathsf{D}_0} \mathsf{C}_0 := \{(f_\mathsf{D}, y) \mid f_\mathsf{D} \in \mathsf{D}_1, y \in \mathsf{C}_0, t_\mathsf{D}(f_\mathsf{D}) = F_0(y)\}$$

Let's fix $y \in \mathsf{C}_0$ and $f_\mathsf{D} : x' \to F_0(y)$ in D_1 and prove that it admits a unique lift. Since $\mathsf{C}_1 \simeq \mathsf{D}_1 \times_{\mathsf{D}_0} \mathsf{C}_0$, the couple (f_D, y) uniquely corresponds to some morphism $f_\mathsf{C} \in \mathsf{C}_1$. If P_1, P_2 are the pullback projections, $P_1(f_\mathsf{D}, y) = f_\mathsf{D} = F_1(f_\mathsf{C})$ and $P_2(f_\mathsf{D}, y) = y = t_\mathsf{C}(f_\mathsf{C})$, so f_C is the unique lift we were looking for.

Viceversa, if F is a discrete fibration, by definition every $f_\mathsf{C} : x \to y$ corresponds uniquely to some (f_D, y) such that $t_\mathsf{D}(f_\mathsf{D}) = F_0(y)$. Also, since $F_1(f_\mathsf{C}) = f_\mathsf{D}$ and $t_\mathsf{C}(f_\mathsf{C}) = y$, we derive that F_1 and t_C behave exactly like the pullback projections.

Mutatis mutandis, the proof for opfibrations works morally the same and will be omitted for the sake of brevity. □

2.5. The content of Proposition 2.4 can be refined. Indeed, one can qualify 'how much' F fails to be a discrete (op-)fibration. To do so, let us first rewrite the diagrams in Equation (1) as:

$$C_1 \xrightarrow{(s_C, F_1)} C_0 \times D_1 \underset{s_D}{\overset{F_0}{\rightrightarrows}} D_0 \qquad C_1 \xrightarrow{(t_C, F_1)} C_0 \times D_1 \underset{t_D}{\overset{F_0}{\rightrightarrows}} D_0$$

2.6. If F is a functor, these diagrams commute. Now we apply the left adjoint $\mathsf{Set} \to \mathsf{AbGrp}$ to them. This means replacing all sets with the abelian groups freely generated by them, and all functions with the corresponding homorphisms. Now homorphisms can be summed and inverted pointwise, and we write:

$$0 \longrightarrow C_1 \xrightarrow{(s_C, F_1)} C_0 \times D_1 \xrightarrow{F_0 - s_D} D_0 \qquad 0 \longrightarrow C_1 \xrightarrow{(t_C, F_1)} C_0 \times D_1 \xrightarrow{F_0 - t_D} D_0$$

Definition 2.7. Since the original diagram commutes, (s_C, F_1) equalizes F_0 and s_D, and so the composition above evaluates to 0, implying $\mathsf{Im}\,(s_\mathsf{C}, F_1) \subseteq \ker(F_0 - s_\mathsf{D})$. This places us into homology land, and we define:

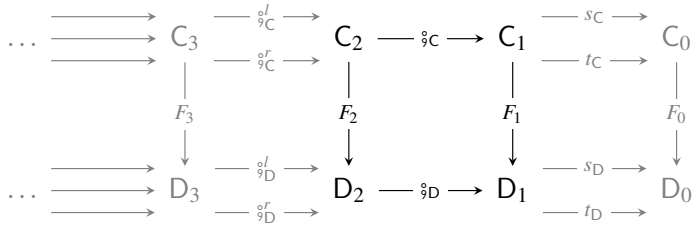
$$H^{-1}_{\text{fib}} := \ker(s_{\mathsf{C}}, F_1) \qquad H^{0}_{\text{fib}} := \ker(F_0 - s_{\mathsf{D}}) / \operatorname{Im}(s_{\mathsf{C}}, F_1)$$
$$H^{-1}_{\text{opfib}} := \ker(t_{\mathsf{C}}, F_1) \qquad H^{0}_{\text{opfib}} := \ker(F_0 - t_{\mathsf{D}}) / \operatorname{Im}(t_{\mathsf{C}}, F_1).$$

2.8. These groups provide a qualitative description of the *obstructions* for the diagrams in Equation (1) being a pullback. When they are trivial there are no obstructions, and putting this fact together with Proposition 2.4, we get:

Proposition 2.9. Let $F : \mathsf{C} \to \mathsf{D}$ be a functor. $H^{-1}_{\text{fib}}, H^{0}_{\text{fib}}$ are trivial if and only if F is a discrete fibration. Similarly, $H^{-1}_{\text{opfib}}, H^{0}_{\text{opfib}}$ are trivial if and only if F is a discrete opfibration.

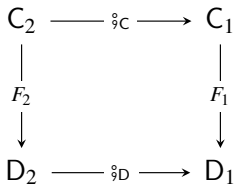
Proof. Consider $\mathsf{Eq}(F_0, s_{\mathsf{D}})$. Since C_1 equalizes F_0 and s_{D}, there is a unique map $C_1 \to \mathsf{Eq}(F_0, s_{\mathsf{D}})$. H^{0}_{fib} is trivial if and only if this map is surjective, whereas H^{-1}_{fib} is trivial if and only if it is injective. So $H^{-1}_{\text{fib}}, H^{0}_{\text{fib}}$ are trivial if and only if $C_1 \simeq \mathsf{Eq}(F_0, s_{\mathsf{D}})$, which by definition holds if and only if the corresponding square in Proposition 2.4 is a pullback. But because of the very same Proposition 2.4, this is true if and only if F is a discrete fibration. A similar argument holds for opfibrations. □

3 The middle square

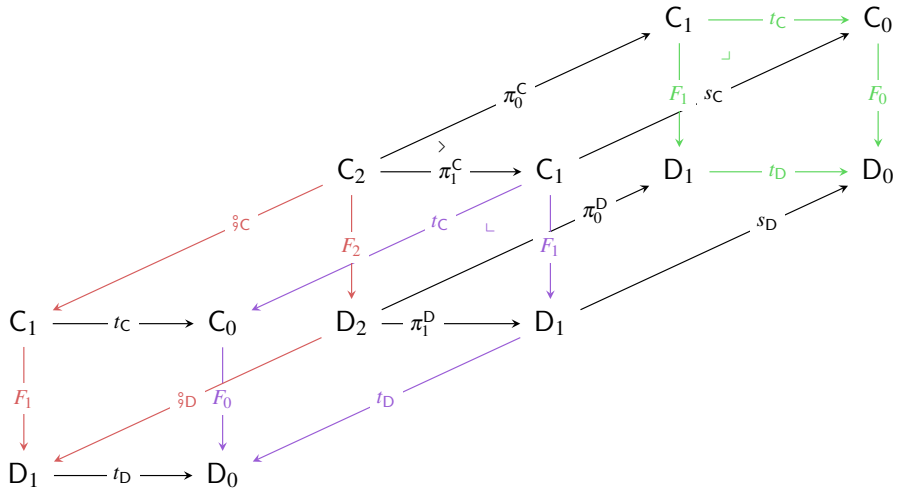


3.1. The middle square is probably the most interesting one in our endeavor. It takes a couple of morphisms $f : A \to B$ and $g : B \to C$ in C_2, and maps them to $F_1(f \fatsemi_{\mathsf{C}} g)$ (right-down) and to $F_1 f \fatsemi_{\mathsf{D}} F_1 g$ (down-right), respectively. The square commutativity states the familiar condition $F(f \fatsemi g) = Ff \fatsemi Fg$, that is, the preservation of composition by a functor $F : \mathsf{C} \to \mathsf{D}$. We now prove that the pullback conditions on the middle and right squares are related.

Proposition 3.2. If any of the squares in Equation (1) is a pullback, then the following is also a pullback:



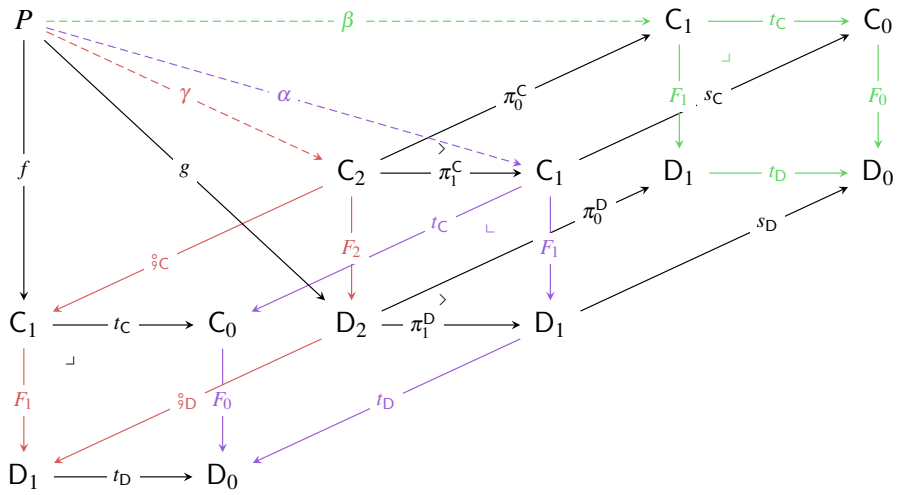
Proof. Assume that the left square in Equation (1) is a pullback, and consider the diagram below. All the squares commute. The squares in purple and green are pullbacks by hypothesis. We want to prove that the red square is a pullback as well.



To do so, suppose to have some object P and morphisms $P \xrightarrow{f} \mathsf{C}_1, P \xrightarrow{g} \mathsf{D}_2$ such that $f \fatsemi F_1 = g \fatsemi (\fatsemi_{\mathsf{D}})$. We need to prove that there is a unique morphism $P \to \mathsf{C}_2$ commuting with the red square. First, consider:

$$f \fatsemi F_1 = g \fatsemi (\fatsemi_{\mathsf{D}}) \rightsquigarrow f \fatsemi F_1 \fatsemi t_{\mathsf{D}} = g \fatsemi (\fatsemi_{\mathsf{D}}) \fatsemi t_{\mathsf{D}} \rightsquigarrow f \fatsemi t_{\mathsf{C}} \fatsemi F_0 = g \fatsemi \pi_1^{\mathsf{D}} \fatsemi t_{\mathsf{D}}$$

Hence we can apply the pullback property of the purple square to $f \fatsemi t_{\mathsf{C}} = g \fatsemi \pi_1^{\mathsf{D}}$, and get a unique morphism $P \xrightarrow{\alpha} \mathsf{C}_1$ such that: $\alpha \fatsemi t_{\mathsf{C}} = f \fatsemi tc$ and $\alpha \fatsemi F_1 = g \fatsemi \pi_1^{\mathsf{D}}$.



But now,

$$g \fatsemi \pi_0^{\mathsf{D}} \fatsemi t_{\mathsf{D}} = g \fatsemi \pi_1^{\mathsf{D}} \fatsemi s_{\mathsf{D}} = \alpha \fatsemi F_1 \fatsemi s_{\mathsf{D}} = \alpha \fatsemi s_{\mathsf{C}} \fatsemi F_0$$

And so we can apply the property of the green pullback to $g \fatsemi \pi_0^{\mathsf{D}}$ and $\alpha \fatsemi s_{\mathsf{C}}$ to obtain a unique morphism $P \xrightarrow{\beta} \mathsf{C}_1$ such that $\beta \fatsemi F_1 = g \fatsemi \pi_0^{\mathsf{D}}$ and $\beta \fatsemi t_{\mathsf{C}} = \alpha \fatsemi s_{\mathsf{C}}$.

Now, notice that the top square composed by $\pi_0^{\mathsf{C}}, t_{\mathsf{C}}, s_{\mathsf{C}}, \pi_0^{\mathsf{C}}$ is a pullback by construction of C_2: Focusing on the equation $\beta \fatsemi t_{\mathsf{C}} = \alpha \fatsemi s_{\mathsf{C}}$, we see that we can apply the property of this pullback to α and β to obtain a unique γ such that $\gamma \fatsemi \pi_1^{\mathsf{C}} = \alpha$ and $\gamma \fatsemi \pi_0^{\mathsf{C}} = \beta$.

To finish, notice that also the frontmost square $F_1, t_{\mathsf{C}}, F_0, t_{\mathsf{D}}$ is a pullback by hypothesis, which forces $\gamma \fatsemi (\fatsemi_{\mathsf{C}}) = f$. Similarly, the square $\pi_0^{\mathsf{D}}, t_{\mathsf{D}}, s_{\mathsf{D}}, \pi_1^{\mathsf{D}}$ is a pullback by construction of D_2, forcing $\gamma \fatsemi F_2 = g$.

The proof when the right square in Equation (1) is a pullback is analogous. □

3.3. Now, we investigate what it means for the middle square to be a pullback. Indeed, as in the case of discrete (op-)fibrations, this corresponds to a well-known concept.

Definition 3.4. A functor $F : \mathsf{C} \to \mathsf{D}$ is a **Conduché functor** if for $\alpha : a \to b$ in C_1 and any factorisation $F_0(a) \xrightarrow{\beta} c \xrightarrow{\gamma} F_0(b)$ of $F_1(\alpha)$, we have:

- There exists a factorisation $a \xrightarrow{\beta'} d \xrightarrow{\gamma'} b$ of α such that $F_1(\beta') = \beta$ and $F_1(\gamma') = \gamma$.
- Any two such factorisations in C are connected by a zigzag of commuting morphisms which map to the identity.

A Conduché functor is **discrete** if each factorisation is unique.

3.5. In a way very similar to the proof of Proposition 2.4, we can prove the following:

Proposition 3.6. Let $F : \mathsf{C} \to \mathsf{D}$ be a functor. The middle square is a pullback square if and only if F is a discrete Conduché functor.

3.7. Putting together Propositions 2.4, 3.2 and 3.6 we recover the fact that discrete Conduché functors are a generalization of discrete (op-)fibrations: Every discrete (op-)fibration is also a strict Conduché functor. The relevance of Proposition 3.2 is that it does not use the definition of discrete (op-)fibration or of Conduché functor: it makes sense in any category with pullbacks, so, for instance, it would keep holding if we would consider our diagrams to be internal in Cat.

3.8. As in the previous section, when the middle square is not a pullback we may want to investigate 'how far' F is from being a Conduché functor. Again, abelianizing the diagram this amounts to consider:

$$0 \longrightarrow C_2 \xrightarrow{(\fatsemi_C, F_2)} C_1 \times D_2 \xrightarrow{F_1 - \fatsemi_D} D_1$$

Definition 3.9. Commutativity of the diagram implies that $\mathsf{Im}(\fatsemi_{\mathsf{C}}, F_2) \subseteq \ker(F_1 - \fatsemi_{\mathsf{D}})$ and so we can define

$$H^{-1}_{\mathrm{Cond}} := \ker(\fatsemi_{\mathsf{C}}, F_2) \qquad H^{0}_{\mathrm{Cond}} := \ker(F_1 - \fatsemi_{\mathsf{D}}) / \mathsf{Im}(\fatsemi_{\mathsf{C}}, F_2)$$

3.10. In a way similar to Proposition 2.9, we can prove the following:

Proposition 3.11. Let $F : \mathsf{C} \to \mathsf{D}$ be a functor. $H^{-1}_{\mathrm{Cond}}, H^{0}_{\mathrm{Cond}}$ are trivial if and only if F is a Conduché functor.

4 Applications: Petri nets

4.1. An application of the categorical tools introduced so far can be illustrated by an illuminating example in the context of categorical semantics for Petri nets.

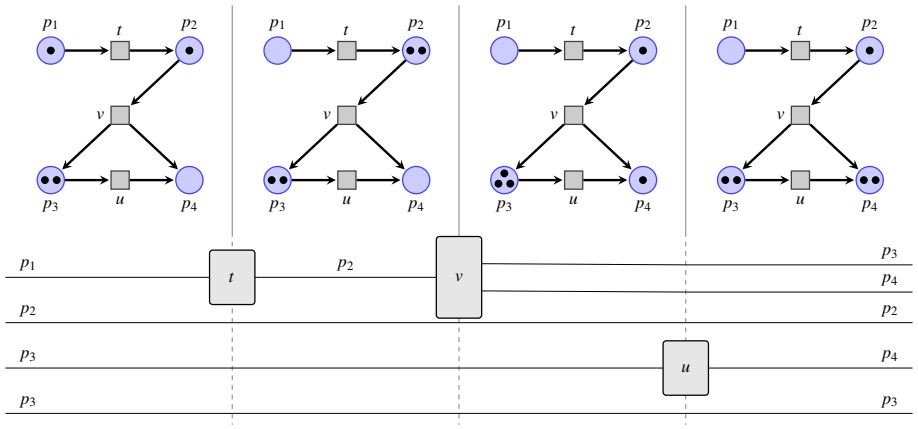
4.2. In the following, we will denote with $S^{\oplus}$ the set of multisets over a set S. Multiset sum will be denoted with $\oplus$, and difference (only partially defined) with $\ominus$. $S^{\oplus}$ with $\oplus$ and the empty multiset is isomorphic to the free commutative monoid on S.

Definition 4.3 (Petri nets and their semantics)**.** A *Petri net* N is defined by a couple of functions $T \xrightarrow{s,t} S^{\oplus}$ for some sets T and S, called the set of places and transitions of the net, respectively. A *marking* for a net $T \xrightarrow{s,t} S^{\oplus}$ is an element of $S^{\oplus}$, representing a distribution of tokens in the net places. A transition u is *enabled* in a marking M if $M \ominus s(u)$ is defined. An enabled transition can *fire*, moving tokens in the net. Firing is considered an atomic event, and the marking resulting from firing u in M is $M \ominus s(u) \oplus t(u)$.

Given N, we can generate a *free symmetric strict monoidal category*, $\mathfrak{F}(N)$, as follows:

- The monoid of objects is $S^{\otimes}$, the set of strings over S. Monoidal product of objects A, B, denoted $A \otimes B$, is given by string concatenation.
- Morphisms are generated by T: each $u \in T$ corresponds to a morphism generator $su \xrightarrow{u} tu$, where su, tu are obtained by choosing some ordering on their underlying multisets; morphisms are obtained by considering all the formal horizontal and vertical compositions of generators, identities and symmetries.

Similarly, we can generate a free commutative strict monoidal category $\mathfrak{C}(N)$ – that is, a monoidal category where symmetries are identities – by considering the set of multisets $S^{\oplus}$ as the monoid of objects, and multiset sum as the monoidal product. Details of these constructions can be found in [1].



4.4. Petri nets are very well-known models of concurrent automata, and are widely used in computer science. Given a net N, morphisms in the categories $\mathfrak{F}(N)$ and $\mathfrak{C}(N)$ correspond to *executions* of the net, as shown in the picture above. The fundamental difference between the semantics $\mathfrak{F}(N)$ and $\mathfrak{C}(N)$ is that the latter does not distinguish between tokens living in the same place, whereas the former does.

4.5. Petri nets can be made more expressive in various ways: For instance, one may consider *guarded nets* where tokens are endowed with colors, and transition inputs/outputs perform extra tasks depending on the color of the tokens. Sometimes, these extensions are strictly more expressive than the classic Petri net formalism, sometimes they are not. Categorically, this process is described by endowing a net N with some sort of functor $\mathfrak{F}(N) \to D$ or, depending on the choice of token philosophy, $\mathfrak{C}(N) \to D$. In practice, D is often taken to be Span, and the functor, which we usually denote with $N^{\sharp}$, is taken to be lax. For instance, guarded nets can be described as nets endowed with strict monoidal functors $\mathfrak{F}(N) \to \mathsf{Span}$ [20], whereas *bounded nets* – nets where a given place cannot contain more than a predetermined number of tokens – can be described as lax-monoidal-lax functors $\mathfrak{C}(N) \to \mathsf{Span}$ [22]. Other flavors of nets can also be represented in this way [17, 21].

4.6. To apply the results heretofore developed to Petri nets, we make use of the following fact:

Lemma 4.7. [Due to Bénabou [2, 3]] Fix B. Any lax double functor $\mathbb{B} \xrightarrow{F} \mathsf{Span}(\mathsf{Set})$ is a pseudofunctor if and only if the projection $\int F \xrightarrow{\pi_F} \mathsf{B}$ of its Grothendieck construction is a discrete Conduché functor.

Corollary 4.8. For any lax double functor $\mathbb{B} \xrightarrow{F} \mathsf{Span}(\mathsf{Set})$, the groups

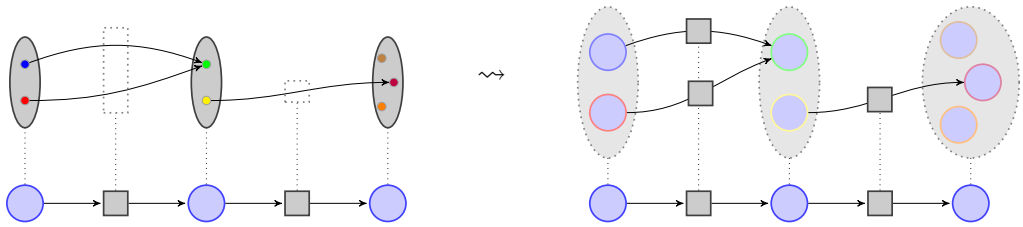
$$H^{-1}_{cond}\left(\int F \xrightarrow{\pi_F} \mathsf{B}\right) \qquad H^{0}_{cond}\left(\int F \xrightarrow{\pi_F} \mathsf{B}\right)$$

measure obstructions to pseudofunctoriality of F.

4.9. Thanks to Lemma 4.7 and Corollary 4.8, we can measure how far a categorical decoration $N^\sharp$ for a Petri net N is from being lax. But what meaning does this have from the point of view of Petri nets?

4.10. In ¶ 4.5, we remarked how an extended Petri net formalism is sometimes exactly as expressive as the classic Petri net one. This means that the extended formalism is nothing more than syntactic sugar, and the extended Petri net can be *internalized*: That is, we can associate a traditional Petri net to it which has the same executions.

4.11. Given a net N and a functor $N^\sharp$, the process of internalization is described by taking its Grothendieck construction $\int N^\sharp$. When the domain of this functor is itself of the form $\mathfrak{F}(M)$ or $\mathfrak{C}(M)$ for some net M, then the decoration semantics is internalizable: M has the same computational behavior of the decorated net N. This has been proved to hold for different flavors of Petri nets [17, 20, 21, 22].



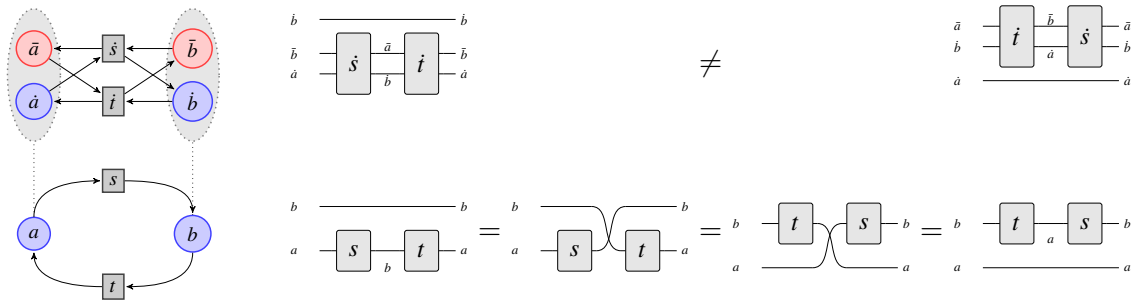
4.12. For example, take guarded nets. We decorate a 'base net' N with token colors and transition guards by defining a strict monoidal functor $N^\sharp : \mathfrak{F}(N) \to \mathsf{Span}$. Since $\mathfrak{F}(N)$ is freely generated, this amounts to map each place to a set of colors and each transition to a span, representing how colors are correlated by transitions. This is the picture on the left above. When we internalize, we obtain a net M such that $\int N^\sharp = \mathfrak{F}(M)$. This is the net sitting on top of the original net N, on the right above. In practice, M is obtained by promoting token colors and arcs in the left picture to places and transitions, respectively.

4.13. The functor $N^\sharp : \mathfrak{F}(N) \to \mathsf{Span}$ for guarded nets is always strict, and so the functor $\int N^\sharp \to \mathfrak{F}(N)$ is always Conduché, but not always a fibration. This makes sense: By looking at the example picture above, it is clear that the fibration condition isn't always satisfied; take for instance the leftmost transition in N and the yellow circled place in M: The discrete fibration condition requires a transition leading into it, but we have none (failure of existence). In the case of the green circled place, we have more than one (failure of unicity).

4.14. So, the homology groups defined in Definition 2.7 give us qualitative information about which token colors are 'problematic' with respect to the (op)fibration condition failing, which in practice means that non-equal token colors in the source of a transition get mapped into the same color by its target.

4.15. In the example of ¶ 4.13, the case made for the green-circled place would be picked up as a non-trivial element of the homology group $H^{-1}_{fib}\left(\int F \xrightarrow{\pi_F} \mathsf{B}\right)$, whereas the case made for the yellow-circled place would be picked up as a non-trivial element of the homology group $H^{0}_{fib}\left(\int F \xrightarrow{\pi_F} \mathsf{B}\right)$.

4.16. Another interesting case is the one of bounded nets. Our semantics here is a *lax* functor $N^\sharp : \mathfrak{C}(N) \to \mathsf{Span}$. This semantics is again internalizable, and so $\int N^\sharp = \mathfrak{F}(M)$, as in the picture below. Basically, M is obtained by adding a new 'antiplace' (colored red below) for each place in M. Given a place a, the antiplace corresponding to it keeps track of how many tokens can still be added to it. So, for instance, three tokens in the antiplace mean '3 more tokens can be added to a'. Transitions consuming tokens from a place add tokens to the corresponding antiplace; transition outputting tokens into a place need to consume an equivalent amount from the antiplace.



4.17. Since our functor $N^\sharp$ is only lax, Lemma 4.7 tells us that the functor $\int N^\sharp \to \mathfrak{C}(N)$ is not always Conduché. We can see this by examining the picture above on the right. In N (bottom), we have a particular execution, which can be written down in two equivalent ways because we are using the commutative semantics. This execution, which is a morphism of $\mathfrak{C}(N)$, corresponds to two different executions of M, which are in turn morphisms in $\int N^\sharp = \mathfrak{C}(M)$. The reason why this equality is not lifted to $\mathfrak{C}(M)$ is that in M all the places of N are doubled, so whereas we could exchange any place b with itself in $\mathfrak{C}(N)$ using commutativity, we cannot do the same in $\mathfrak{C}(M)$, as $\dot{b}$ and $\tilde{b}$, the place and antiplace corresponding to b, are considered as different generators and cannot be swapped one for the other.

4.18. In this case, the obstructions to being Conduché witness histories that should 'morally be the identified' in the category $\mathfrak{C}(M)$ of executions of the bounded net M, but are not. In a way, these can be considered imperfections of the bounding technique. Thanks to Lemma 4.7, the groups in Corollary 4.8 give us qualitative information about these obstructions – again to existence and uniqueness.

4.19. In the particular example of ¶ 4.17, the execution described in the figure would be picked up as a non-trivial element of the homology group $H^{-1}_{cond}\left(\int F \xrightarrow{\pi_F} \mathsf{B}\right)$.

5 Applications: Delta lenses

5.1. In the context of applied category theory, *optics* [28, 10, 4] constitute one of the main topics of research. Broadly speaking, the study of optics pertains the categorical characterization of 'bi-directional processes', that is, all techniques dealing at the same time with the transformation of data from the whole to the part, and from the part to the whole. Optics have received a lot of interest from the applied category theory community since they are the underlying concept of a lot of different, apparently unrelated topics, including categorical probability theory [31], open games [5], functional programming [28], dynamical systems [26], automatic differentiation [14], machine learning and cybernetics [7, 12, 32].

5.2. Lenses [24] are a particular type of optics, and by far the most used and well-known ones. Simply put, a lens is made of two parts: One is responsible for accessing a given part from an object constituting the whole. We call this part `get`. The other, is responsible of pushing an update of the accessed part to the

whole, and we call it `put`. This concept has been formalized in a broad variety of ways; the formalization we are the most interested in in this section is called *delta lenses*, and is mainly due to [13, 25].

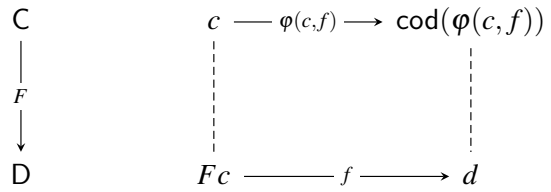
Definition 5.3 (From [9])**.** A **delta lens** $(F,\varphi) : \mathsf{C} \to \mathsf{D}$ is given by a functor $F : \mathsf{C} \to \mathsf{D}$ together with a *lifting operation*

$$(c \in \mathsf{C}, Fc \xrightarrow{f} d \in \mathsf{D}) \longmapsto \left(c \xrightarrow{\varphi(c,f)} \mathsf{cod}(\varphi(c,f))\right)$$

Such that:

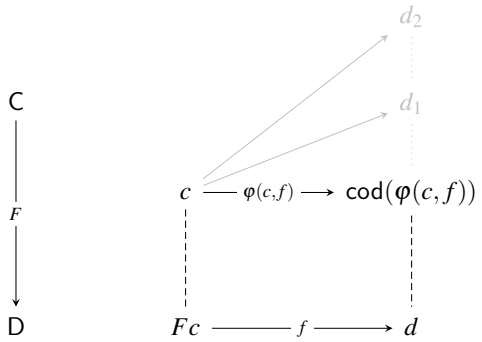
- $F\varphi(c,f) = f$;
- $\varphi(c, \mathsf{Id}_{Fc}) = \mathsf{Id}_c$;
- $\varphi(c, f \fatsemi g) = \varphi(c,f) \fatsemi \varphi(\mathsf{cod}(\varphi(c,f)), g)$.

5.4. The idea of a delta lens is the following: We interpret the category C as 'the category of the wholes', and the category D as 'the category of the parts': For each object $c \in \mathsf{C}$, $Fc \in \mathsf{D}$ represents a part of c we are focusing on. In this respect, F models the `get` side of the lens. Now, suppose to have a morphism $Fc \xrightarrow{f} d$ in D. We interpret this as a process that changes the 'zoomed in part', Fc, in some way. The lifting operation says that, for each change we apply to Fc, there should a way to 'push this change back to c', hence obtaining a corresponding process $c \xrightarrow{\varphi(c,f)} \mathsf{cod}(\varphi(c,f))$ turning c into a 'new whole' $\mathsf{cod}(\varphi(c,f))$. This is the `put` part of the lens.



5.5. The three axioms listed in Definition 5.3 are required to formally back up our intuition. The first implies that, when zooming into the new $\mathsf{cod}(\varphi(c,f))$ by applying F, we get exactly the part d resulting by changing Fc with f; The second says that if we do not change Fc, then we shouldn't change c. The third says that the 'push' side of the lens can be composed: If we edit Fc twice in sequence, then there is no difference between lifting $f \fatsemi g$ all in one go or computing the liftings sequentially.

5.6. In [9], it is pointed out how delta lenses are a generalization of the concept of opfibration: φ guarantees that, for each object $c \in \mathsf{C}$ and morphism $Fc \xrightarrow{f} d \in \mathsf{D}$, we get a corresponding lift in C; yet, this lift may not be guaranteed to be unique. Indeed, we can precisely think of a delta lens as a functor $F : \mathsf{C} \to \mathsf{D}$ with a *chosen lift* for each morphism of D.

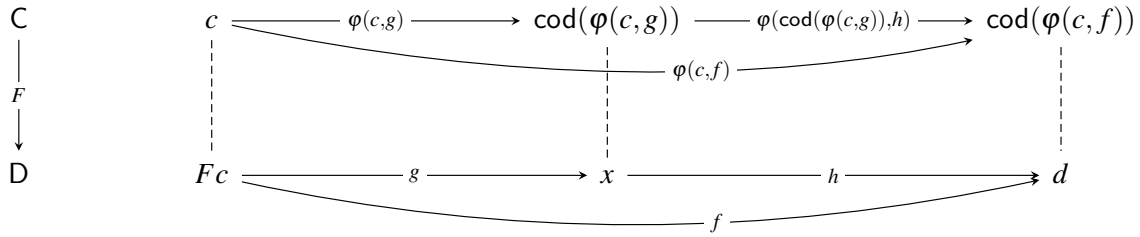


5.7. Delta lenses are part of the toolkit of categorical machine learning [12], and have been heavily studied, with all sorts of connections between delta lenses, cofunctors, discrete and split opfibrations having been explored from a formal standpoint [9, 23]. In particular in [9], a refinement of Lemma 4.7, identifying delta lenses over D with lax double functors $\mathbb{D} \to \mathsf{Span}$ that factorise in a particular way, has been provided.

5.8. From our point of view, by applying the techniques heretofore developed we can make a timid contribution to the field, measuring how much a given delta lens fails to be an opfibration – and, as a consequence, of how much a delta lens fails to be a c-lens, a comparison already introduced in [23]. As observed in ¶ 5.6, a delta lens $(F, \varphi) : \mathsf{C} \to \mathsf{D}$ always provides lifts for all morphisms in D and objects of C. As such, by Proposition 2.9, $H^0_{\text{opfib}}(F, \varphi)$ is always trivial. On the other hand, $H^{-1}_{\text{opfib}}(F, \varphi)$ is generally not-trivial, and measures how far we are from having a unique lift operation $\varphi(-,-)$.

5.9. From an applicative point of view, a delta lens that is also an opfibration is a very rigid structure: In a nutshell, it tells us that there is only one way to push an update of the part to the whole: The structure describing the way a part is transformed canonically induces a transformation structure for the whole.

5.10. As for a delta lens $(F, \varphi) : \mathsf{C} \to \mathsf{D}$ being a strict Conduché functor, we observe that because of the third axiom in Definition 5.3, factorizations in D always lift to factorizations in C:



5.11. Hence, as a consequence of Proposition 3.11, we have that $H^0_{\text{Cond}}(F, \varphi)$ is also trivial. As in ¶ 5.8, the same does not hold for $H^{-1}_{\text{Cond}}(F, \varphi)$: The factorization lift of a morphism is not required to be unique.

5.12. We close this section whith a conjecture, which unfortunately we did not have enough time to investigate thoroughly.

Conjecture 5.13. Let $F : \mathsf{C} \to \mathsf{D}$ be a functor. If $H^0_{\text{Opfib}}(F), H^0_{\text{Cond}}(F)$ are trivial, then assuming the axiom of choice F can be made into a **pre-delta lens**, that is, a delta lens that does not satisfy axiom 2 in Definition 5.3.

Proof attempt. Given any morphism $Fc \xrightarrow{f} d \in \mathsf{D}$, denote with $S_{c,f}$ the set of lifts of f. This set is non empty because $H^0_{\text{Opfib}}(F)$ is trivial. Furthermore, every time f factorizes as $g \fatsemi h$, the triviality of $H^0_{\text{Cond}}(F)$ guarantees that every lift in $S_{c,f}$ can be split – potentially in multiple ways – thus defining a partial, surjective function:

$$S_{c,g} \times \bigsqcup_{\mathsf{cod}\, x \mid x \in S_{c,g}} (S_{\mathsf{cod}\, x, h}) \longrightarrow S_{c,f}$$

This information can be used to define a graph highlighting the coherence requirements between all of the sets S, which are the vertexes of the graph.

We then define $\varphi(c, \mathsf{Id}_{Fc}) := \mathsf{Id}_c$ on identites. The hard part of the proof is using the graph, together with the axiom of choice, to pick an element out of every $S_{c,f}$ for f not an identity morphism. If the base category is free this is easier, as we can decompose every f as a unique composition of generators. In this case, the graph above has *leaves* – that is, vertexes with no arrows out of them – and we can define

a lens by picking any element in the S laying on its leaves arbitrarily, and then defining the other nodes inductively. The general case is much harder since the graph will have all sorts of cycles, and the choice out of every S cannot be made so liberally. □

6 Discussion and future work

Investigating failures of compositionality in category theory transcends mere theoretical relevance and has the potential of impacting several real-world applications. With many modern ML systems being inherently compositional and a growing scientific community focusing on their categorical formalization [30], one is predictably interested in looking at where such compositionality fails. Formal techniques to qualify such obstructions may hold promise in improving understanding of ML inner workings, a process often hindered by their lack of interpretability [33]. For example, functoriality of clustering algorithms has been found to be a very desirable property in computational topology. Under the view that clustering is the statistical analogue of constructing connected components of a topological space, it has been investigated how certain algorithms lose their functoriality under a different choice of morphisms in the source category [8]. Such functorial perspective potentially helps to extend algorithms and qualify modifications that break functoriality [29].

For these reasons, applying the techniques heretofore presented to the fields of categorical machine learning [15, 16] and cybernetics [7] constitutes one of the main directions for future work, to which Section 5 is a modest prelude.

Another direction of investigation is applying the techniques hereby developed to the field of fibrational linguistics [18, 19].

Finally, another important – and perhaps more obvious – direction of future work consists in asking the obvious question: 'what about the other squares to the left?' We conjecture that, by looking at the diagram in ¶ 1.4, the next square proceeding to the left can be used to measure the laxity of a given functor without necessarily going through Corollary 4.8. This is due the fact, observed in ¶ 1.3, that the diagram $\mathsf{C}_3 \underset{\fatsemi^r}{\overset{\fatsemi^l}{\rightrightarrows}} \mathsf{C}_2 \xrightarrow{\fatsemi} \mathsf{C}_1$ Commutes precisely when C is a category. Paired with the observations made in ¶ 3.1, we conjecture that there should be a way to produce some homology group measuring laxness of F directly, but this has proven to be an elusive task so far.

Acknowledgements: this collaboration was born from attending the '*Workshop on Categorical Approaches to Emergence and Non-Compositionality in Complex Systems*' hosted in Oxford, UK in 2023; for this we wish to thank the organizers for the invitations and for running the event. We also wish to thank Owen Lynch for pointing out the connection between lax functors and Conduché fibrations.

References

[1] J. C. Baez, F. Genovese, J. Master, and M. Shulman. "Categories of Nets". In: *2021 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. Los Alamitos, CA, USA: IEEE Computer Society, July 2021, pp. 1–13. DOI: 10.1109/LICS52264.2021.9470566 (cit. on p. 7).

[2] Jean Bénabou. "2-Dimensional limits and colimits of distributors". In: *Mathematisches Forschungsinstitut Oberwolfach, Tagungsbericht* 30 (1972), pp. 6–7 (cit. on p. 8).

[3] Jean Bénabou. "Distributors at work". In: *Lecture notes written by Thomas Streicher* 11 (2000) (cit. on p. 8).

[4] Guillaume Boisseau and Jeremy Gibbons. "What you needa know about Yoneda: Profunctor optics and the Yoneda Lemma (functional pearl)". In: *Proceedings of the ACM on Programming Languages* 2.ICFP (2018), pp. 1–27 (cit. on p. 9).

[5] Joe Bolt, Jules Hedges, and Philipp Zahn. "Bayesian open games". In: *Compositionality* 5 (Oct. 2023), p. 9. DOI: 10.32408/compositionality-5-9. URL: http://dx.doi.org/10.32408/compositionality-5-9 (cit. on p. 9).

[6] Benjamin Merlin Bumpus, Zoltan A. Kocsis, and Jade Edenstar Master. *Structured Decompositions: Structural and Algorithmic Compositionality*. 2023. arXiv: 2207.06091 [math.CT] (cit. on p. 2).

[7] Matteo Capucci, Bruno Gavranović, Jules Hedges, and Eigil Fjeldgren Rischel. "Towards Foundations of Categorical Cybernetics". In: *Electronic Proceedings in Theoretical Computer Science* 372 (Nov. 2022), pp. 235–248. DOI: 10.4204/eptcs.372.17. URL: http://dx.doi.org/10.4204/EPTCS.372.17 (cit. on pp. 9, 12).

[8] Gunnar Carlsson and Facundo Mémoli. "Classifying clustering schemes". In: *Foundations of Computational Mathematics* 13 (2013), pp. 221–252 (cit. on p. 12).

[9] Bryce Clarke. "The double category of lenses". PhD thesis. Macquarie University, 2023 (cit. on pp. 10, 11).

[10] Bryce Clarke, Derek Elkins, Jeremy Gibbons, Fosco Loregian, Bartosz Milewski, Emily Pillmore, and Mario Román. "Profunctor Optics, a Categorical Update". In: *Compositionality* 6 (Feb. 2024), p. 1. DOI: 10.32408/compositionality-6-1. URL: http://dx.doi.org/10.32408/compositionality-6-1 (cit. on p. 9).

[11] Bob Coecke. "Compositionality as we see it, everywhere around us". In: *The Quantum-Like Revolution: A Festschrift for Andrei Khrennikov*. Springer, 2023, pp. 247–267 (cit. on p. 2).

[12] Zinovy Diskin. "General Supervised Learning as Change Propagation with Delta Lenses". In: *Foundations of Software Science and Computation Structures*. Springer International Publishing, 2020, pp. 177–197. DOI: 10.1007/978-3-030-45231-5_10. URL: http://dx.doi.org/10.1007/978-3-030-45231-5_10 (cit. on pp. 9, 11).

[13] Zinovy Diskin, Yingfei Xiong, Krzysztof Czarnecki, Hartmut Ehrig, Frank Hermann, and Fernando Orejas. "From state-to delta-based bidirectional model transformations: The symmetric case". In: *Model Driven Engineering Languages and Systems: 14th International Conference, MODELS 2011, Wellington, New Zealand, October 16-21, 2011. Proceedings 14*. Springer. 2011, pp. 304–318 (cit. on p. 10).

[14] Conal Elliott. "The simple essence of automatic differentiation". In: *Proceedings of the ACM on Programming Languages* 2 (July 2018), pp. 1–29. DOI: 10.1145/3236765 (cit. on p. 9).

[15] Bruno Gavranović. *Fundamental Components of Deep Learning: A category-theoretic approach*. 2024. arXiv: 2403.13001 [cs.LG] (cit. on p. 12).

[16] Bruno Gavranović, Paul Lessard, Andrew Dudzik, Tamara von Glehn, João G. M. Araújo, and Petar Veličković. *Categorical Deep Learning: An Algebraic Theory of Architectures*. 2024. arXiv: 2402.15332 [cs.LG] (cit. on p. 12).

[17] Fabrizio Genovese, Fosco Loregian, and Daniele Palombi. "Nets with Mana: A Framework for Chemical Reaction Modelling". In: *Lecture Notes in Computer Science*. Springer International Publishing, 2021, pp. 185–202. DOI: 10.1007/978-3-030-78946-6_10 (cit. on pp. 7, 8).

[18] Fabrizio Genovese, Fosco Loregian, and Caterina Puca. "Fibrational Linguistics (FibLang): Language Acquisition". In: *Electronic Proceedings in Theoretical Computer Science* 380 (Aug. 2023), pp. 224–236. DOI: 10.4204/eptcs.380.13. URL: http://dx.doi.org/10.4204/EPTCS.380.13 (cit. on p. 12).

[19] Fabrizio Genovese, Fosco Loregian, and Caterina Puca. *Fibrational linguistics: First concepts*. 2022. arXiv: 2201.01136 [math.CT] (cit. on p. 12).

[20] Fabrizio Genovese and David I. Spivak. "A Categorical Semantics for Guarded Petri Nets". In: *Lecture Notes in Computer Science*. Springer International Publishing, 2020, pp. 57–74. DOI: 10.1007/978-3-030-51372-6_4 (cit. on pp. 7, 8).

[21] Fabrizio Romano Genovese, Jelle Herold, Fosco Loregian, and Daniele Palombi. "A Categorical Semantics for Hierarchical Petri Nets". In: *Electronic Proceedings in Theoretical Computer Science* 350 (Dec. 2021), pp. 51–68. DOI: 10.4204/eptcs.350.4 (cit. on pp. 7, 8).

[22] Fabrizio Romano Genovese, Fosco Loregian, and Daniele Palombi. "A Categorical Semantics for Bounded Petri Nets". In: *Electronic Proceedings in Theoretical Computer Science* 372 (Nov. 2022), pp. 59–71. DOI: 10.4204/eptcs.372.5 (cit. on pp. 7, 8).

[23] Michael Johnson and Robert Rosebrugh. "Delta lenses and opfibrations". In: *Electronic Communications of the EASST* 57 (2013) (cit. on p. 11).

[24] Michael Johnson, Robert Rosebrugh, and Richard Wood. "Algebras and Update Strategies". In: *JUCS - Journal of Universal Computer Science* 16.5 (2010), pp. 729–748. DOI: 10.3217/jucs-016-05-0729. eprint: https://doi.org/10.3217/jucs-016-05-0729. URL: https://doi.org/10.3217/jucs-016-05-0729 (cit. on p. 9).

[25] Michael Johnson, Robert D Rosebrugh, et al. "Unifying Set-Based, Delta-Based and Edit-Based Lenses." In: *Bx@ ETAPS* 1571 (2016), pp. 1–13 (cit. on p. 10).

[26] David Jaz Myers. *Categorical systems theory*. 2022. URL: http://davidjaz.com/Papers/DynamicalBook.pdf (cit. on p. 9).

[27] Caterina Puca, Amar Hadzihasanovic, Fabrizio Genovese, and Bob Coecke. "Obstructions to Compositionality". In: *Electronic Proceedings in Theoretical Computer Science* 397 (Dec. 2023), pp. 226–245. DOI: 10.4204/eptcs.397.14 (cit. on p. 2).

[28] Mitchell Riley. *Categories of Optics*. 2018. arXiv: 1809.00738 [math.CT] (cit. on p. 9).

[29] Dan Shiebler. "Functorial Manifold Learning". In: *Electronic Proceedings in Theoretical Computer Science* 372 (Nov. 2022), pp. 1–13. DOI: 10.4204/eptcs.372.1. URL: http://dx.doi.org/10.4204/EPTCS.372.1 (cit. on p. 12).

[30] Dan Shiebler, Bruno Gavranović, and Paul Wilson. *Category Theory in Machine Learning*. 2021. arXiv: 2106.07032 [cs.LG] (cit. on p. 12).

[31] Toby St. Clere Smithe. *Bayesian Updates Compose Optically*. 2020. arXiv: 2006.01631 [math.CT] (cit. on p. 9).

[32] Toby St Clere Smithe. "Cyber Kittens, or Some First Steps Towards Categorical Cybernetics". In: *Electronic Proceedings in Theoretical Computer Science* 333 (Feb. 2021), pp. 108–124. DOI: 10.4204/eptcs.333.8. URL: http://dx.doi.org/10.4204/EPTCS.333.8 (cit. on p. 9).

[33] Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. "A survey on neural network interpretability". In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 5.5 (2021), pp. 726–742 (cit. on p. 12).