

# TITLE

Daniel Sinderson

May 2024

## Introduction

### Category Theory Basics

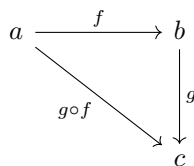
Category theory is a tall tower. From its vantage point it's possible to see the patterns that repeat across the panorama of mathematical fields: from group theory to linear algebra to topology. This view is spectacular. It's also useful. Category theory helps us structure our thinking by finding and giving a name to those patterns and abstractions common to fields within mathematics and beyond, such as science, engineering, and linguistics.

### Categories

What gives Category theory this bird's eye view is the way that it abstracts the concept of composition. Of how, given a path from  $a$  to  $b$  and a path from  $b$  to  $c$ , you can create a new path from  $a$  to  $c$ . This probably seems unassumingly simple, and it really is. As such categories themselves are fairly simple to define.

**Definition 1** (Category). A category  $\mathcal{C}$  is defined by the following:

1.  $\mathcal{C}$  contains a collection of objects  $ob(\mathcal{C})$
2. For any two objects  $a, b \in ob(\mathcal{C})$  there is a collection of morphisms between those objects  $\mathcal{C}(a, b)$  called the homset. We'll denote an element  $f \in \mathcal{C}(a, b)$  using function notation  $f : a \rightarrow b$ .
3. Every object  $a \in ob(\mathcal{C})$  has a morphism to itself  $1_a : a \rightarrow a$ .
4. For every two morphisms  $f : a \rightarrow b$  and  $g : b \rightarrow c$  there's a third morphism  $g \circ f : a \rightarrow c$  that's their composition.



These objects and morphisms are then under the following two constraints:

1. (Unitality) Any morphism  $f : a \rightarrow b$  can be composed with the identity morphisms of  $a$  and  $b$  such that  $f \circ 1_a = 1_b \circ f = f$ . Thus making the following diagram commute:

$$\begin{array}{ccc} a & \xrightarrow{f} & b \\ 1_a \downarrow & & \downarrow 1_b \\ a & \xrightarrow{f} & b \end{array}$$

2. (Associativity) For any morphisms  $f : a \rightarrow b$ ,  $g : b \rightarrow c$ , and  $h : c \rightarrow d$ ,  $h \circ (g \circ f) = (h \circ g) \circ f$ .

$$\begin{array}{ccccc} a & & & & c & \xrightarrow{h} & d \\ & \searrow^{g \circ f} & & \nearrow_{h \circ g} & & & \\ a & \xrightarrow{f} & b & \xrightarrow{g} & c & & \end{array}$$

It turns out that categories are a fantastically powerful abstraction. Since composition lies at the heart of most of mathematics and many other disciplines such as physics, engineering, and linguistics, an abstraction over composition has applications in a stunningly wide range of fields. Categories can even model categories.

## Functors and Categories of Categories

Since categories are a mathematical structure, like groups or vector spaces, it makes sense to ask if there exists a kind of structure-preserving map between them, much like a homomorphism between groups or a linear transformation between vector spaces. For categories, such a map would have to preserve identities, composition, unitality, and associativity. We call such a map a functor.

**Definition 2 (Functor).** A functor  $F : \mathcal{C} \rightarrow \mathcal{D}$  is a map between categories  $\mathcal{C}$  and  $\mathcal{D}$  such that the following hold:

1. For any object  $a \in ob(\mathcal{C})$  there is an object  $Fa \in ob(\mathcal{D})$ .
2. For any morphism  $f : a \rightarrow b$  between objects  $a$  and  $b$  in  $\mathcal{C}$  there is a morphism  $Ff : Fa \rightarrow Fb$  between objects  $Fa$  and  $Fb$  in  $\mathcal{D}$ .
3. For all objects  $a \in ob(\mathcal{C})$  and  $Fa \in ob(\mathcal{D})$ ,  $F1_a = 1_{Fa}$ .
4. For any composition of morphisms  $g \circ f$  in  $\mathcal{C}$ ,  $Fg \circ Ff = F(g \circ f)$  in  $\mathcal{D}$ .

With functors, it's possible to construct a category where the objects themselves are categories and the morphism between them are functors. There are some technicalities for such constructions involving formal notions of local and global size that are necessary to avoid the Russel-esque paradox of the category of categories containing itself, but we'll ignore these in this paper.

## Functor Categories and Natural Transformations

Taking one more step into the rarefied air, it's also possible to create a category whose objects are functors and whose morphisms are structure-preserving maps called natural transformations.

**Definition 3** (Natural Transformation). A natural transformation  $\alpha : F \Rightarrow G$  is a map between functors  $F : \mathcal{C} \rightarrow \mathcal{D}$  and  $G : \mathcal{C} \rightarrow \mathcal{D}$  such that the following holds:

1. For each object  $a \in Ob(\mathcal{C})$ , there is a morphism  $\alpha_a : Fa \rightarrow Ga$  in  $\mathcal{D}$ . This is called the  $c$ -component of  $\alpha$ .
2. For every morphism  $f : a \rightarrow b$  in  $\mathcal{C}$ ,  $\alpha_b \circ Ff = Gf \circ \alpha_a$ . This is called the naturality condition, and it preserves functoriality.

This means that the diagram below commutes.

$$\begin{array}{ccc} Fa & \xrightarrow{\alpha_a} & Ga \\ Ff \downarrow & & \downarrow Gf \\ Fb & \xrightarrow{\alpha_b} & Gb \end{array}$$

## Structure within Categories

### Kinds of Morphisms

NOTE: Isomorphism in the category **Set** are bijective functions.

**Definition 4** (Isomorphism). We say that two objects in a category are isomorphic if there exists a morphism  $f : a \rightarrow b$  for which there is another morphism  $f^{-1} : b \rightarrow a$  such that  $g \circ f = 1_b$  and  $f \circ g = 1_a$ . Such a morphism is called an isomorphism.

NOTE: Monomorphism in the category **Set** are injective functions.

**Definition 5** (Monomorphism). A morphism  $f : b \rightarrow c$  is a monomorphism if for any pair of parallel morphisms  $g, h : a \rightarrow b$ ,  $f \circ g = f \circ h$  implies that  $g = h$ .

NOTE: Epimorphisms in the category **Set** are surjective functions.

**Definition 6** (Epimorphism). A morphism  $f : b \rightarrow c$  is an epimorphism if for any pair of parallel morphisms  $g, h : c \rightarrow d$ ,  $g \circ f = h \circ f$  implies that  $g = h$ .

## Universal Properties

**Definition 7** (Terminal Object). Let  $t \in Ob(\mathcal{C})$  be given. Then  $t$  is a terminal object in  $\mathcal{C}$  if the following hold:

1. For all  $x \in Ob(\mathcal{C})$ , there is exactly one morphism  $f : x \rightarrow t$ .
2. For any other object  $t'$  in  $\mathcal{C}$  that meets condition (1), there exists a unique isomorphism  $g : t \rightarrow t'$ .

Dual to terminal objects are initial objects.

**Definition 8** (Initial Object). Let  $i \in Ob(\mathcal{C})$  be given. Then  $i$  is a initial object in  $\mathcal{C}$  if the following hold:

1. For all  $x \in Ob(\mathcal{C})$ , there is exactly one morphism  $f : i \rightarrow x$ .
2. For any other object  $i'$  in  $\mathcal{C}$  that meets condition (1), there exists a unique isomorphism  $g : i \rightarrow i'$ .

In category theory there's a universal property that generalizes notions of multiplication called the product.

**Definition 9** (Product). Let  $a, b, v \in Ob(\mathcal{C})$  and morphisms  $f : v \rightarrow a$  and  $g : v \rightarrow b$  be given. The object  $v$  is a product of  $a$  and  $b$  if the following holds:

1. For any other object  $x$  with morphisms  $f' : x \rightarrow a$  and  $g' : x \rightarrow b$ , there is a unique isomorphism  $k : x \rightarrow v$  such that  $f \circ k = f'$  and  $g \circ k = g'$ .

This means the following diagram commutes:

$$\begin{array}{ccc}
 & x & \\
 f' \swarrow & \downarrow k & \searrow g' \\
 & v & \\
 f \swarrow & & \searrow g \\
 a & & b
 \end{array}$$

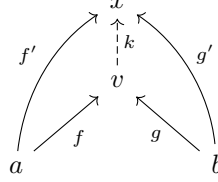
This conditions ensures that the object  $v$  is the universal or canonical object satisfying the pattern, with the morphisms of any other such object being able to be written as a sort of embellishment of the morphisms from  $v$ .

Dual to products are coproducts, which generalize the notion of addition.

**Definition 10** (Coproduct). Let  $a, b, v \in Ob(\mathcal{C})$  and morphisms  $f : a \rightarrow v$  and  $g : b \rightarrow v$  be given. The object  $v$  is a coproduct of  $a$  and  $b$  if the following holds:

1. For any other object  $x$  with morphisms  $f' : a \rightarrow x$  and  $g' : b \rightarrow x$ , there is a unique isomorphism  $k : v \rightarrow x$  such that  $k \circ f = f'$  and  $k \circ g = g'$ .

This means the following diagram commutes:

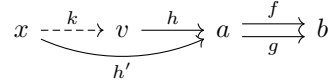


NOTE: In **Set**, equalizers pick out the subset of a set  $a$  that is the intersection of the preimages  $f^{-1} : b \rightarrow a$  and  $g^{-1} : b \rightarrow a$ .

**Definition 11** (Equalizer). Let  $a, b, v \in Ob(\mathcal{C})$  and let morphisms  $f : a \rightarrow b$ ,  $g : a \rightarrow b$ , and  $h : v \rightarrow a$  be given. The object  $v$  is an equalizer if the following hold:

1.  $f \circ h = g \circ h$ .
2. For any other object  $x \in Ob(\mathcal{C})$  with morphism  $h' : x \rightarrow a$  such that  $f \circ h' = g \circ h'$ , there exists a unique morphism  $k : x \rightarrow v$  such that  $h' = h \circ k$ .

This means the following diagram commutes:

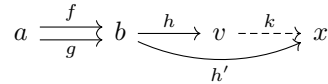


NOTE: In **Set**, coequalizers pick out the subset of a set  $b$  that is the intersection of the images  $f : a \rightarrow b$  and  $g : a \rightarrow b$ .

**Definition 12** (Coequalizer). Let  $a, b, v \in Ob(\mathcal{C})$  and let morphisms  $f : a \rightarrow b$ ,  $g : a \rightarrow b$ , and  $h : b \rightarrow v$  be given. The object  $v$  is a coequalizer if the following hold:

1.  $h \circ f = h \circ g$ .
2. For any other object  $x \in Ob(\mathcal{C})$  with morphism  $h' : b \rightarrow x$  such that  $h' \circ f = h' \circ g$ , there exists a unique morphism  $k : v \rightarrow x$  such that  $h' = k \circ h$ .

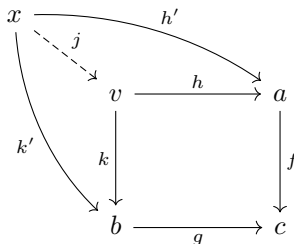
This means that the following diagram commutes:



NOTE: In **Set**, pullbacks are the fiber product of two sets.

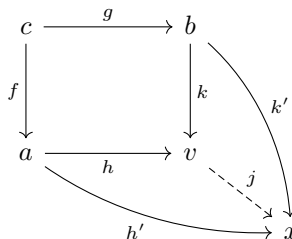
**Definition 13** (Pullback). Let  $a, b, c, v \in Ob(\mathcal{C})$  and let morphisms  $f : a \rightarrow c$ ,  $g : b \rightarrow c$ ,  $h : v \rightarrow a$ , and  $k : v \rightarrow b$  be given. The object  $v$  is a pullback if the following hold:

- This means that the following diagram commutes:



**Definition 14** (Pushout). Let  $a, b, c, v \in \text{Ob}(\mathcal{C})$  and let morphisms  $f : c \rightarrow a$ ,  $g : c \rightarrow b$ ,  $h : a \rightarrow v$ , and  $k : b \rightarrow v$  be given. The object  $v$  is a pushout if the following hold:

- This means that the following diagram commutes:



6

A cocone consists of any finite collection of objects and morphisms between them, combined with an additional object  $v$  called the vertex of the cocone and morphisms from all other objects in the collection to  $v$ . Such a cocone is universal if for all other cocones of that type in the category there is a unique morphisms from  $v$  to their vertex: i.e,  $k : \rightarrow v'$  exists and is unique.

## Limits and Colimits

Given any cone, if there is an instance of that cone for which all other instances have a unique morphism to it, that cone is the limit. Dually for cocones and colimits. In fact, it's possible to construct a category whose objects are the cones and whose morphisms are these unique factorizations. In this category, the limit is a terminal object. Dually for cocones and colimits again, except the colimit is an initial object in the constructed category of cocones instead of a terminal object.

## Structure Between Categories

### Kinds of Functors

**Definition 15** (Full Functor). A functor  $F : \mathcal{C} \rightarrow \mathcal{D}$  is full if for each  $x, y \in Ob(\mathcal{C})$ , the map  $\mathcal{C}(x, y) \rightarrow \mathcal{D}(Fx, Fy)$  is surjective.

**Definition 16** (Faithful Functor). A functor  $F : \mathcal{C} \rightarrow \mathcal{D}$  is faithful if for each  $x, y \in Ob(\mathcal{C})$ , the map  $\mathcal{C}(x, y) \rightarrow \mathcal{D}(Fx, Fy)$  is injective.

**Definition 17** (Essentially Surjective On Objects). A functor  $F : \mathcal{C} \rightarrow \mathcal{D}$  is essentially surjective on objects if for each  $y \in Ob(\mathcal{D})$ , there is some  $x \in Ob(\mathcal{C})$  such that  $y$  is isomorphic to  $Fx$ .

**Definition 18** (Injective on Objects). A functor  $F : \mathcal{C} \rightarrow \mathcal{D}$  is injective on objects if for each  $x \in Ob(\mathcal{C})$ , there is a unique  $y \in Ob(\mathcal{D})$  such that  $y = Fx$ .

NOTE: Play a huge role in category theory: i.e in the Yoneda lemma

NOTE: Representable functors require that the source category  $\mathcal{C}$  be locally small so that the collections of morphisms between objects in the category are sets.

**Definition 19** (Covariant Representable). A functor  $F : \mathcal{C} \rightarrow \mathbf{Set}$  is representable if there is an object  $c \in Ob(\mathcal{C})$  such that the functor  $\mathcal{C}(c, -) : \mathcal{C} \rightarrow \mathbf{Set}$  is naturally isomorphic to  $F$ .

**Definition 20** (Contravariant Representable). A functor  $F : \mathcal{C}^{op} \rightarrow \mathbf{Set}$  is representable if there is an object  $c \in Ob(\mathcal{C})$  such that the functor  $\mathcal{C}(-, c) : \mathcal{C}^{op} \rightarrow \mathbf{Set}$  is naturally isomorphic to  $F$ .

**Definition 21** (Forgetful). A forgetful functor  $F : \mathcal{C} \rightarrow \mathbf{Set}$  is a functor that sends objects in a category of structures-on-sets, like groups or preorders, to the underlying sets themselves. They "forget" the additional structure in the source category.

**Definition 22** (Free). A free functor  $F : \mathbf{Set} \rightarrow \mathcal{C}$  is a functor that sends sets to the free construction on that set, like the free group on the set. They construct the desired structure on the set in a way that requires no choices.

### Comparing Categories

**Definition 23** (Subcategory). A category  $\mathcal{C}$  with a functor  $F : \mathcal{C} \rightarrow \mathcal{D}$  that is both faithful and injective on objects is a subcategory of the domain  $\mathcal{D}$ . A functor  $F$  of this type is called an **embedding** of  $\mathcal{C}$  into  $\mathcal{D}$ .

**Definition 24** (Full Subcategory). A category  $\mathcal{C}$  with a functor  $F : \mathcal{C} \rightarrow \mathcal{D}$  that is full, faithful, and injective on objects is a full subcategory of the domain  $\mathcal{D}$ . A functor  $F$  of this type is called a **full embedding** of  $\mathcal{C}$  into  $\mathcal{D}$ .

**Definition 25** (Natural Isomorphism). A natural isomorphism is a natural transformation  $\alpha : F \rightarrow G$  where all of the c-components of the transformation  $\alpha_c : Fc \rightarrow Gc$  are isomorphisms. This means there exists some other natural transformation  $\eta : G \rightarrow F$  such that  $\eta_c \circ \alpha_c = 1_{Fc}$  and  $\alpha_c \circ \eta_c = 1_{Gc}$ .

**Definition 26** (Adjunctions). An adjunction is a pair of functors  $F : \mathcal{C} \rightarrow \mathcal{D}$  and  $G : \mathcal{D} \rightarrow \mathcal{C}$  such that for any object  $c \in Ob(\mathcal{C})$  and any object  $d \in Ob(\mathcal{D})$  there is an isomorphism between the sets of functions between them and their images under the functors that is natural. In other words, that for every  $c \in Ob(\mathcal{C})$  and  $d \in Ob(\mathcal{D})$  there is a natural isomorphism  $\mathcal{C}(c, Gd) \cong \mathcal{D}(Fc, d)$ .

In this case, the functor  $F$  is the left adjoint functor and the functor  $G$  is the right adjoint functor.

**Definition 27** (Monad). Given a category  $\mathcal{C}$ , a monad on  $\mathcal{C}$  is a functor  $M : \mathcal{C} \rightarrow \mathcal{C}$  with the following two natural transformations:

1.  $\eta : 1_{\mathcal{C}} \rightarrow M$  called the unit.
2.  $\mu : M \circ M \rightarrow M$  called the multiplication.

These natural transformation are required to make the following diagrams commute.

$$\begin{array}{ccc}
 M \circ M \circ M & \xrightarrow{M\mu} & M \circ M \\
 \downarrow \mu M & & \downarrow \mu \\
 M \circ M & \xrightarrow{\mu} & M
 \end{array}$$



$$\begin{array}{ccccc}
M & \xrightarrow{\eta M} & M \circ M & \xleftarrow{M\eta} & M \\
& \searrow 1_M & \downarrow \mu & \swarrow 1_M & \\
& & M & & 
\end{array}$$

These commutative diagrams with the natural transformations  $\eta$  and  $\mu$  create a kind of monoidal structure on the collection of endofunctors  $M, M \circ M, M \circ M \circ M, \text{etc.}$

### The Yoneda Lemma

**Definition 28.** The Yoneda lemma states that for any functor  $F : \mathcal{C} \rightarrow \mathbf{Set}$  where  $\mathcal{C}$  is locally small and any object  $c \in \text{Ob}(\mathcal{C})$ , there is a bijection between the collection of natural transformations going from the functor  $\mathcal{C}(c, -)$  to  $F$  and the object  $Fc \in \text{Ob}(\mathbf{Set})$ .

Stated another way, this means that  $\text{Hom}(\mathcal{C}(c, -), F) \cong Fc$  for any functor  $F : \mathcal{C} \rightarrow \mathbf{Set}$  and all objects  $c \in \text{Ob}(\mathcal{C})$ .

### Adding Structure to Categories

**Definition 29** (Monoidal Categories). A category  $\mathcal{C}$  is monoidal if the following exist.

1. A functor  $\otimes : \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$  called the monoidal product.
2. An object  $\mathbb{1} \in \text{ob}(\mathcal{C})$  called the unit.
3. A natural isomorphism  $\alpha : (a \otimes b) \otimes c \rightarrow a \otimes (b \otimes c)$  called the associator, with components  $\alpha_{x,y,z} : (x \otimes y) \otimes z \rightarrow x \otimes (y \otimes z)$ .
4. A natural isomorphism  $\lambda : \mathbb{1} \otimes a \rightarrow a$  called the left unitor with components  $\lambda_x : \mathbb{1} \otimes x \rightarrow x$ .
5. A natural isomorphism  $\rho : a \otimes \mathbb{1} \rightarrow a$  called the right unitor with components  $\rho_x : x \otimes \mathbb{1} \rightarrow x$ .

All of the above must exist such that the following two diagrams, called the triangle identity and the pentagon identity, commute.

$$\begin{array}{ccc}
(x \otimes \mathbb{1}) \otimes y & \xrightarrow{\alpha_{x,y,z}} & x \otimes (\mathbb{1} \otimes y) \\
& \searrow \rho_x \otimes 1_y & \downarrow 1_x \otimes \lambda_y \\
& & x \otimes y
\end{array}$$

$$\begin{array}{ccc}
((w \otimes x) \otimes y) \otimes z & \xrightarrow{\alpha_{(w \otimes x), y, z}} & (w \otimes x) \otimes (y \otimes z) \\
\downarrow \alpha_{w, x, y} \otimes 1_z & & \downarrow \alpha_{w, x, (y \otimes z)} \\
(w \otimes (x \otimes y)) \otimes z & & \\
\downarrow \alpha_{w, (x \otimes y), z} & & \\
w \otimes ((x \otimes y) \otimes z) & \xrightarrow{1_w \otimes \alpha_{x, y, z}} & w \otimes (x \otimes (y \otimes z))
\end{array}$$

**Definition 30** (Symmetric Monoidal Category). A monoidal category equipped with a natural isomorphism  $\beta : a \otimes b \rightarrow b \otimes a$ , called its braiding, is symmetric if  $\beta_{y,x} \circ \beta_{x,y} = 1_{x \otimes y}$  for all  $x, y \in \text{Ob}(\mathcal{C})$  and if the following diagram, called the hexagon identity, commutes.

$$\begin{array}{ccccc}
(x \otimes y) \otimes z & \xrightarrow{\alpha_{x, y, z}} & x \otimes (y \otimes z) & \xrightarrow{\beta_{x, y \otimes z}} & (y \otimes z) \otimes x \\
\downarrow \beta_{x, y} \otimes 1_z & & & & \downarrow \alpha_{y, z, x} \\
(y \otimes x) \otimes z & \xrightarrow{\alpha_{y, x, z}} & y \otimes (x \otimes z) & \xrightarrow{1_y \otimes \beta_{x, z}} & y \otimes (z \otimes x)
\end{array}$$

**Definition 31** (Cartesian Monoidal Category). A monoidal category  $\mathcal{C}$  is cartesian when its monoidal product is its categorical product and its monoidal unit is its terminal object. This obviously means that  $\mathcal{C}$  must contain all products and have a terminal object.

An example of this would be **Set**, where the Cartesian product is the monoidal product and the singleton set is the monoidal unit.

## Categorical System Theory

Systems are everywhere in science and engineering. Whether discrete or continuous, deterministic or stochastic, all such systems have two things in common: states that they can be in, and rules for how the system's current state changes. These two features alone describe what are called closed systems. Closed systems are systems that don't interact with others or with an outside environment that they're a part of. This is a hobbling limitation. In order to open these systems up we need to give them an interface. We need them to be able to accept inputs that shape the way they evolve, and we need them to be able to expose some part of their current state to their surrounding environment. This is the gift that category theory will give us. By opening our systems up, we open them to the power of composition. We can connect them.

## The Category $\mathbf{Lens}_{\mathcal{C}}$

Given a cartesian category, it's possible to construct a new category of systems whose states are drawn from the objects of your base category and whose rules for updating their state are drawn from the morphism of your base category. We call this category  $\mathbf{Lens}_{\mathcal{C}}$ , where  $\mathcal{C}$  is the base category. The objects in this category are called arenas and the morphisms between them are called lenses.

**Definition 32** (Lenses). Given a cartesian category  $\mathcal{C}$  and objects  $A^-, A^+, B^-, B^+ \in ob(\mathcal{C})$ , a lens consists of a passforward map  $f : A^+ \rightarrow B^+$  and a passback map  $f^\# : A^+ \times B^- \rightarrow A^-$  between two arenas as follows:

$$\begin{pmatrix} f^\# \\ f \end{pmatrix} : \begin{pmatrix} A^- \\ A^+ \end{pmatrix} \rightleftarrows \begin{pmatrix} B^- \\ B^+ \end{pmatrix}$$

**Definition 33** (The Category  $\mathbf{Lens}_{\mathcal{C}}$ ). Given a cartesian category  $\mathcal{C}$ , the category  $\mathbf{Lens}_{\mathcal{C}}$  has the following properties.

1. A collection of objects called arenas. An arena  $\begin{pmatrix} A^- \\ A^+ \end{pmatrix}$  is a pair of objects  $A^-, A^+ \in ob(\mathcal{C})$ .
2. For each pair of arenas a collection of morphisms  $\begin{pmatrix} f^\# \\ f \end{pmatrix} : \begin{pmatrix} A^- \\ A^+ \end{pmatrix} \rightleftarrows \begin{pmatrix} B^- \\ B^+ \end{pmatrix}$  called lenses.
3. For each arena an identity lens  $\begin{pmatrix} \pi_2 \\ 1_{A^+} \end{pmatrix} : \begin{pmatrix} A^- \\ A^+ \end{pmatrix} \rightleftarrows \begin{pmatrix} A^- \\ A^+ \end{pmatrix}$  where the passback map  $\pi_2$  is the projection  $\pi_2 : A^+ \times A^- \rightarrow A^-$ .
4. For any two compatible lenses a composite lens as follows:

$$\begin{aligned} \begin{pmatrix} f^\# \\ f \end{pmatrix} : \begin{pmatrix} A^- \\ A^+ \end{pmatrix} &\rightleftarrows \begin{pmatrix} B^- \\ B^+ \end{pmatrix} \\ \begin{pmatrix} g^\# \\ g \end{pmatrix} : \begin{pmatrix} B^- \\ B^+ \end{pmatrix} &\rightleftarrows \begin{pmatrix} C^- \\ C^+ \end{pmatrix} \\ \begin{pmatrix} g^\# \\ g \end{pmatrix} \circ \begin{pmatrix} f^\# \\ f \end{pmatrix} : \begin{pmatrix} A^- \\ A^+ \end{pmatrix} &\rightleftarrows \begin{pmatrix} C^- \\ C^+ \end{pmatrix} \end{aligned}$$

such that the passforward map is and the passback map sends

## References

- [Che22] Eugenia Cheng. *The joy of abstraction: an exploration of math, category theory, and life*. Cambridge University Press, 2022.
- [FS19] Brendan Fong and David I Spivak. *An invitation to applied category theory: seven sketches in compositionality*. Cambridge University Press, 2019.
- [Gol14] Robert Goldblatt. *Topoi: the categorial analysis of logic*. Elsevier, 2014.
- [Lei14] Tom Leinster. *Basic category theory*, volume 143. Cambridge University Press, 2014.
- [LR03] F William Lawvere and Robert Rosebrugh. *Sets for mathematics*. Cambridge University Press, 2003.
- [Mil18] Bartosz Milewski. *Category theory for programmers*. Blurb, 2018.
- [ML13] Saunders Mac Lane. *Categories for the working mathematician*, volume 5. Springer Science & Business Media, 2013.
- [Mye22] David Jaz Myers. Categorical systems theory. *Manuscript in preparation*, 2022.
- [nLa24] nLab authors. symmetric monoidal category. <https://ncatlab.org/nlab/show/symmetric+monoidal+category>, February 2024. Revision 54.
- [Rie17] Emily Riehl. *Category theory in context*. Courier Dover Publications, 2017.
- [Ros22] Daniel Rosiak. *Sheaf theory through examples*. MIT Press, 2022.