Category Theory

Davi Sales Barreira

March 26, 2022

Contents

List of Definitions

List of Theorems

List of Examples

Notes mostly based on Ribeiro [7], Bradley et al. [2], Borceux [1]. Whenever $\circledast$ is present, this indicates an original idea from the author, thus, it should be taken with a grain of salt.

Notation

Here is a small guideline on the notation used.

We usually refer to generic categories as $\mathcal{C}$ and similar uppercase letters with curly font. For named categories (e.g. category of graphs, category of small categories,...) we usually use a bold font with no italic, e.g. **Gr**, **SmCat**, **Top**...

1. $\circ$ - Used to symbolize composition of morphisms similar to how we compose functions.
2. $\fatsemi$ - Represents composition, but with orders reversed, i.e. $g \circ f = f \fatsemi g$.
3. $\diamond$ - *Whiskering, prewhiskering, postwhiskering.*
4. $\cong$ - Isomorphism in the context applied (e.g. categorical, set theoretical, etc).
5. $\simeq$ - Equivalence of categories via natural transformations (weaker than isomorphism).

1 Preface

The study of Category Theory enables us to view Mathematics from a vantage point, and better understand how the different areas are connected. By looking at the subject from the distance (via Category Theory), we get a glimpse at the connections (and disconnections) between different fields.

Another interesting observation about Category Theory is that it's breaking the theoretical barrier and it's starting to be applied in real world applications. One prominent example is in programming, as highlighted by the book Milewski [5]. The book by Fong and Spivak [3] also focus on the application side of Category Theory.

For those interested in applying Category Theory, Julia has a very prominent package called Catlab.jl, which will be shown in these notes.

These notes are intended to serve as a quick introduction to people interested in applying Category Theory. Thus, instead of focusing on proving theorems or presenting a plethora of examples in mathematics, the goal is to rigorously (and quickly) introduce the field together with some intuition and useful examples.

The notes start by introducing the **big 5** of Category Theory: categories, functors, natural transformations, limits/colimits and adjoints. After that, we focus on some applications of Category Theory, mainly it's application to data structures/databases.

2 Categories

In this section, we formally define what a category is, and we provide some examples.

2.1 Universes, Sets and Classes

When working on Category Theory, it's common to find universal statements such as "for all topological spaces...". The issue with such statements is that, in a purely set-theoretical sense, we have to know whether such large collection ("all topological spaces") is indeed a set. We might be tempted to say that's true, but it's not so simple. The most known example of a possible failure of such which the answer is "no".

statements is Russel's Paradox of whether there is a set of all sets, for Therefore, in order to deal with such issue, we need a way to differentiate between a valid set and an arbitrary collection. Here is where the notion of a Universe comes in.

Definition 2.1 (Universe). We say that a set $\mathfrak{U}$ is a universe if[1]

(i) $x \in y$ and $y \in \mathfrak{U}$, then $x \in \mathfrak{U}$;

(ii) $I \in \mathfrak{U}$, and $\forall i \in I, x_i \in \mathfrak{U}$, then $\cup_{i \in I} x_i \in \mathfrak{U}$;

(iii) $x \in \mathfrak{U}$ then $\mathcal{P}(x) \in \mathfrak{U}$, where $\mathcal{P}(x)$ is the power set;

(iv) $x \in \mathfrak{U}$ and $f : x \to y$ is a surjective function, then $y \in \mathfrak{U}$;

(v) $\mathbb{N} \in \mathfrak{U}$.

From this definition, one can prove the following proposition.

Proposition 2.2. (i) $x \in \mathfrak{U}$ and $y \subset x$, then $y \in \mathfrak{U}$;

(ii) $x \in \mathfrak{U}$ and $y \subset x$, then $\{x, y\} \in \mathfrak{U}$;

(iii) $x \in \mathfrak{U}$ and $y \subset x$, then $x \times y \in \mathfrak{U}$;

(iv) $x \in \mathfrak{U}$ and $y \subset x$, then $y^x \in \mathfrak{U}$, where y^x is the set of functions $f : x \to y$.

With this definition, we state the axiom of existence universes.

Axiom 2.1. Every set S belongs to some universe $\mathfrak{U}$.

Definition 2.3. For a fixed universe $\mathfrak{U}$, if a set S is an element of $\mathfrak{U}$, then S is called a *small set*.

[1] Definition from Borceux [1]

Talking about "small sets" and "big set" might become daunting, so instead, we use a different convention which is based on Gödel-Bernays theory of sets and classes. This theory states that:

Axiom 2.2 (Gödel-Bernays). Every set is a class, and a class is a set if and only if it belongs to some (other) class.

Note that using the notion of Universes, we can recover Gödel-Bernays theory. For that, use the following definition:

Definition 2.4. For a fixed universe $\mathfrak{U}$, we call S a *set* if it's an element of $\mathfrak{U}$, and call S a *class* if it's a subset of $\mathfrak{U}$. A class that is not a set is called a *proper class*.

Since every set is a class, if $S \in \mathfrak{U}$, then S is a class, since U is a set and therefore a class, implying that S belongs to a class. On the other direction, if S is a class and $S \in V \subset \mathfrak{U}$, then since $V \subset \mathfrak{U}$, this means that $S \in \mathfrak{U}$, thus it's a set.

From now on, whenever we say *set* we are implying *small set* and whenever we say *class* we are implying either small or big sets, following Borceux [1] convention.

2.2 What is a Category?

Let's formally define a category and provide some examples.

Definition 2.5 (Category). A category $\mathcal{C} = \langle Ob_{\mathcal{C}}, Mor_{\mathcal{C}} \rangle$ consists of a class of objects $Ob_{\mathcal{C}}$ and a class of morphisms $Mor_{\mathcal{C}}$ satisfying the following conditions:

(i) Every morphism $f \in Mor_{\mathcal{C}}$ is associated to two objects $X, Y \in Ob_{\mathcal{C}}$ which is represented by $f : X \to Y$ or $X \xrightarrow{f} Y$, where $dom(f) = X$ is called the domain of f and $cod(f) = Y$ is the codomain. Moreover, we define $Mor_{\mathcal{C}}(X, Y)$ as

$$Mor_{\mathcal{C}}(X, Y) := \{f \in Mor_{\mathcal{C}} \ : \ X \in dom(f), \ Y \in cod(f)\};$$

(ii) For any three objects $X, Y, Z \in Ob_{\mathcal{C}}$, there exists a composition operator

$$\circ : Mor_{\mathcal{C}}(X, Y) \times Mor_{\mathcal{C}}(Y, Z) \to Mor_{\mathcal{C}}(X, Z),$$

(iii) For each object $X \in Ob_{\mathcal{C}}$ there exists a morfism $id_X \in Mor_{\mathcal{C}}(A, A)$ called the identity.

The composition operator must have the following properties:

(p.1) *Associative*: for every $f \in Mor_{\mathcal{C}}(A, B), g \in Mor_{\mathcal{C}}(B, C), h \in Mor_{\mathcal{C}}(C, D)$ then

$$h \circ (g \circ f) = (h \circ g) \circ f.$$

(p.2) For any $f \in Mor_{\mathcal{C}}(X,Y)$, $g \in Mor_{\mathcal{C}}(Y,X)$,

$$f \circ id_X = f, \quad id_Y \circ g = g.$$

There are many ways to refers to the class of morphisms $Mor_{\mathcal{C}}(X,Y)$, such as $\mathcal{C}(X,Y)$ or $\hom_{\mathcal{C}}(X,Y)$. The reason for this is that this set is sometimes called hom-set. In this notes, we'll use either $Mor_{\mathcal{C}}(X,Y)$ or $\mathcal{C}(X,Y)$ when there is no ambiguity. Also, we'll use dom_f to mean $dom(f)$, and similarly for the codomain.

Another point about conventions. When talking about composition, it's convenient to use the operator $\fatsemi$, which is equivalent to the composition $\circ$, but with the inverted order, i.e. $f \fatsemi g = g \circ f$. The convinience will become clearer once we introduce Hasse diagrams as a way to represent categories.

When the class of morphism $Mor_{\mathcal{C}}(A,B)$ for every pair or objects is a set, the category $\mathcal{C}$ is called a *locally small category*[2]. If both $Ob_{\mathcal{C}}$ and $Mor_{\mathcal{C}}$ are sets, we then have a *small category*.

Finally, whenever it's not ambiguous, we might drop the subscript and use Ob to refer to the objects of $\mathcal{C}$ and Mor to refer to the morphisms of $\mathcal{C}$.

2.3 Examples of Categories

It's very common to represent categories via Hasse Diagrams. In these diagrams, the objects are represented as dots, and the morphisms as arrows. Let's show some examples.

Example 2.1 (Category 1). The category $\mathbf{1}$ consists of $Ob_{\mathbf{1}} := \{A\}$ and $Mor_{\mathbf{1}} = \{id_A\}$. The diagram for such category is shown below.

Figure 1: Hasse diagram of category $\mathbf{1}$.

Example 2.2 (Category 2 and 3). The category $\mathbf{2}$ consists of $Ob_{\mathbf{2}} := \{A, B\}$ and $Mor_{\mathbf{1}} = \{id_A, id_B, f\}$, where $f : A \to B$. The diagram for such category is shown below.

[2]Note that $Mor_{\mathcal{C}}$ can be a class while each $Mor_{\mathcal{C}}(A,B)$ is a set

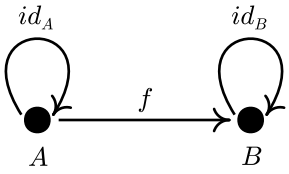


Figure 2: Hasse diagram of category **2**.

Since we know that identities are always present in categories, we'll omit them from future diagrams when there is no ambiguity. Thus, the figure below represents the same diagram as Figure 2.

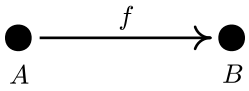


Figure 3: Hasse diagram of category **2** omitting identity morphisms.

The category **3** has three morphisms besides the identities, given by f, g and their composition $f \fatsemi g$. The figure below illustrates the category with all it's morphisms.

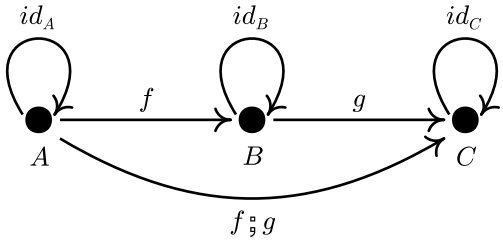


Figure 4: Hasse diagram of category **3** showing all morphisms.

Drawing all the morphisms can make the diagram become too crowded, specially as the number of objects and morphisms grows. Hence, we simplify the diagram representation by ommiting not only the identity morphisms, but also the compositions. These can always be assumed to exist, since they are a necessery condition for every category. Thus, the figure below represents the same diagram as Figure 4.

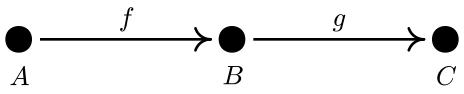


Figure 5: Hasse diagram of category **3** omitting identities and compositions.

Example 2.3 (Discrete Categories). The discrete category $\underline{\mathbf{N}}$ is the category with N objects and $Mor_{\underline{\mathbf{N}}} := \{id_1, ..., id_N\}$. An example of this category is illustrated below.

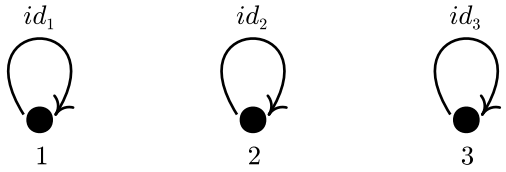


Figure 6: Example of category $\underline{\mathbf{3}}$.

Example 2.4 (Preorders). A preorder is defined by a tuple $(P, \leq)$, where P is a set of values, such that

(i) For $a, b \in P$, if $a \leq b$ and $b \leq c$, then $a \leq c$;

(ii) For every $a \in P$, $a \leq a$.

We can show that actually, this is a category, which we'll call $\mathcal{P}$, where $Ob_{\mathcal{P}} = P$ and each morphism f represents $a \leq b$, where $cod_f = a$ and $dom_f = b$. One example of preorder is the set of $\mathbb{N}$ equiped with the binary relation $\leq$ which is shown in the diagram below.

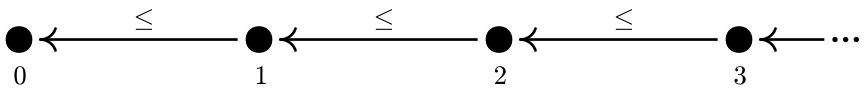


Figure 7: Hasse diagram of preorder category of natural numbers.

Note that in preorders, there is at most one morphism between each pair of objects. Thus, categories with such property are often referred as *thin categories* or *preorder category* (in Fong and Spivak [3], the authors call this a *preorder reflection*).

Example 2.5 (Monoids). A monoid is a triple $(M, \oplus, e_M)$ where $\oplus : M \times M \to M$ is a binary operation and e_M the neutral element, such that:

1. $a \oplus (b \oplus c) = (a \oplus b) \oplus c$

2. $a \oplus e_M = e_M \oplus a = a$.

Note that $(\mathbb{N} \cup \{0\}, +, 0)$ is a monoid.

Moreover, each single monoid can be defined as a category itself. For monoid $(M, \oplus, e_M)$, define a category $\mathcal{C}$ such that $Ob_{\mathcal{C}} := \{M\}$, and the set of morphisms are the elements of M, i.e. $Mor_{\mathcal{C}} := \{a \in M\}$. Finally, we define the composition operartion as $a \circ b := a \oplus b$ Thus, $(\mathbb{N} \cup \{0\}, +, 0)$ is the category illustrated in Figure 8.

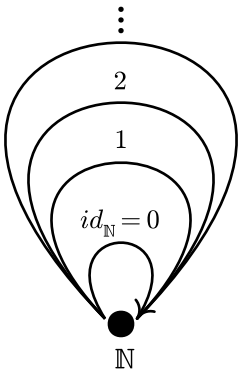


Figure 8: Hasse diagram of monoid category of natural numbers.

There are many other examples:

1. **Set** which is the category of sets, where the objects are sets and the morphisms are functions between sets.

2. **Top** is the category where topological spaces are the objects and continuous functions are the morphisms.

3. $\mathbf{Vec}_{\mathbb{F}}$ is the category where vector spaces over field $\mathbb{F}$ are the objects, and linear transformations are the morphisms.

4. **Gr** is the category of directed graphs, where $Ob_{\mathbf{Gr}} := \{\text{Vertex}, \text{Arrow}\}$, and the morphisms are
$$Mor_{\mathbf{Gr}} := \{src, tgt, id_{\text{Vertex}}, id_{\text{Arrow}}\}$$
where $src : \text{Arrow} \to \text{Vertex}$ returns the source vertex for each arrow and $tgt : \text{Arrow} \to \text{Vertex}$ returns the target vertex.

2.4 Programming with Category Theory

One might be surprised to find out that Category Theory, although very abstract in nature, has actual applications in the "real world". A very interesting example of this is in programming.

In programming languages such as Julia, we can think of 'Types' as objects and functions as morphisms.

2.5 Isomorphism, monomorphism and epimorphism

A very important definition in Category Theory is the notion of isomorphism. In Set Theory, we say that two sets are isomorphic if there is an invertible function between them. Yet, this concept is not restricted to Set Theory and can be generalized in Category Theory as follows:

Definition 2.6 (Categorical Isomorphism). Let $\mathcal{C}$ be a category with $X, Y \in Ob_{\mathcal{C}}$ and $f \in Mor_{\mathcal{C}}(X, Y)$.

(i) We say that f is *left invertible* if there exists $f_l \in Mor_{\mathcal{C}}(Y, X)$ such that $f_l \circ f = id_X$;

(ii) We say that f is *right invertible* if there exists $f_r \in Mor_{\mathcal{C}}(Y, X)$ such that $f \circ f_r = id_Y$;

(iii) We say that f is invertible if it's both left and right invertible.

When an invertible morphism exists between X and Y, we say that they are isomorphic.

Proposition 2.7. The following properties on inverses are true:

1. If f is an invertible morphism, then the left and right inverses are the same.

2. If f and g are invertible and composable, then $f \circ g$ is also invertible.

Proof. 1. Let f be invertible with left inverse f_l and right inverse f_r. Therefore,

$$f_l \circ id_Y = f_l \circ f \circ f_r = id_X \circ f_r = f_r.$$

2. Let $f : A \to B$ and $g : B \to C$ be invertible and composable, with $f \circ g : A \to C$. There exists the inverses $g^{-1} : C \to B$ and $f^{-1} : B \to A$. Note that $f^{-1} \circ g^{-1} : C \to A$, thus

$$(f^{-1} \circ g^{-1}) \circ (g \circ f) = f^{-1} \circ (g^{-1} \circ g) \circ f = (f^{-1} \circ id_B) \circ f = f^{-1} \circ f = id_A.$$

$$(g \circ f) \circ (f^{-1} \circ g^{-1}) = g \circ (f \circ f^{-1}) \circ g^{-1} = g \circ (id_B \circ g^{-1}) = g \circ g^{-1} = id_B.$$

We conclude that $(g \circ f)^{-1} = f^{-1} \circ g^{-1}$.

□

Although similar to set isomorphism, categorical isomorphism is in a sense more general, and captures our intuition of isomorphism between categories better than the set theoretic case, even when we have finite objects.

Consider the following example. Let $\mathcal{P}$ be the category of Posets, where posets $(P, \leq_p)$ are the objects. Take two objects $P_1, P_2 \in Ob_{\mathcal{P}}$, where $P_1 := \{a, b\}$ with a and b **not** comparable, and $P_2 := \{0, 1\}$ where indeed $0 \leq 1$. The question is whether P_1 is "actually" isomorphic to P_2, and our intuition say that they should not be, since P_1 has two incomparable elements while P_2 has two comparable elements.

If we use the set theoretic definition, we would conclude that they **are** isomorphic, since there is a bijective function between P_1 and P_2. Take for example $f : P_1 \to P_2$ where $f(a) = 0$ and $f(b) = 1$. So the set theoretic isomorphism does not capture what we want. What about the categorical isomorphism? We can prove that this will not be an isomorphism using the categorical definition. Yet, in order to prove this, we need to specify what are the morphisms between the posets, and to do this, we need to define what are functors.

In the same way that set isomorphism is not the same as categorical isomorphism, the notions of injectivity and surjectivity are not equivalent to their categorical counterparts, which are called monomorphism and epimorphism.

Definition 2.8 (Monomorphism). Let $\mathcal{C}$ be a category and $f \in Mor_{\mathcal{C}}(A, B)$. We say that f is a monomorphism (or monic), if

$$f \circ g = f \circ h \implies g = h.$$

Definition 2.9 (Epimorphism). Let $\mathcal{C}$ be a category and $f \in Mor_{\mathcal{C}}(A, B)$. We say that f is an epimorphism (or epic), if

$$g \circ f = h \circ f \implies g = h.$$

Important! A morphism f can be both epic and monic, without being an isomorphism, which again highlights the difference between this concepts and their set-theoretic counterparts.

Proposition 2.10. The following properties on monomorphism and epimorphism are true:

1. f left-invertible $\implies$ f is monic. The converse are is not true.
2. f right-invertible $\implies$ f is epic. The converse are is not true.
3. f invertible $\implies$ f is monic and epic. The converse are is not true.

4. f monic and right-invertible $\implies$ f is isomorphism.

5. f epic and left-invertible $\implies$ f is isomorphism.

Proof. 1. Note $f : A \to B$ left-invertible implies that there exists a $f_l : B \to A$ such that $f_l \circ f = id$. Hence, for a $g : B \to C$ and $h : B \to C$, if

$$f \circ g = f \circ h,$$

then we have that

$$f_l \circ f \circ g = f_l \circ f \circ h \implies g = h.$$

To show that the converse is false, consider the category **2**(Figure 2). Note that $f : A \to B$ is monic, since the only morphism that composes with f is id_A and id_B. Yet, note that f is not left invertible, since there isn't even a morphism from B to A.

2. Use the same argument, but reversing the order of the compositions. For the converse, again consider the same category **2**. Note that $f : A \to B$ is epic, but it's not right invertible.

3. True since invertible means left and right invertible.

4. Since $f : A \to B$ right invertible, then there exists $f_r : B \to A$ such that $f \circ f_r = id_B$. Thus,

$$id_B \circ f = (f \circ f_r) \circ f = f \circ (f_r \circ f) = f \circ id_A \implies f_r \circ f = id_A.$$

5. Same argument. □

2.6 Zero, Initial and Terminal Objects

Definition 2.11 (Zero, Initial and Terminal). Let $\mathcal{C}$ be a category.

1. An object $I \in Ob_{\mathcal{C}}$ is *initial* if for every $A \in Ob_{\mathcal{C}}$, there is exactly one morphism from I to A. Thus, from I to I there is only the identity.

2. An object $T \in Ob_{\mathcal{C}}$ is *terminal* if for every $A \in Ob_{\mathcal{C}}$, there is exactly one morphism from A to T. Thus, from I to I there is only the identity.

3. An object is *zero* if it's both terminal and initial.

Note that in the definitions above, we are defining these objects in terms of existence and uniqueness of morphisms, which is known in category theory as **universal constructions** (more on this later).

Theorem 2.12. Every *initial* object is unique up to an isomorphism, i.e. if in a category there are two *initial* objects, then they are isomorphic. Similarly, *terminal* objects are unique up to an isomorphism. Moreover, the isomorphism is unique between initial object, and between terminal objects.

Proof. Let I_1, I_2 be initial. Then, there exists only $f : I_1 \to I_2$ and $g : I_2 \to I_1$. But since $g \circ f : I_1 \to I_1$ is a morphism from the initial object I_1, it must be equal to id_{I_1}. The same for I_2, which implies that f and g are inverses, and thus the objects are isomorphic. Since both f and g are the only morphisms from I_1 and I_2, this also implies that their are the only isomorphism. The same proof works for terminal objects. □

Example 2.6 (Terminal and Initial Objects in Set). Without thinking too much, one might assumet that in the category **Set** the empty set would be a zero object; but that's not true. In reality, the empty set is the initial object, since $f : \varnothing \to B$ is the only function from the empty set to any other set. Why is this valid?

Remember that in set theory, a function from two sets is defined as a binary relation such that for every $x \in dom_f$, there is a unique $y \in cod_f$, i.e. f is a triple (A, B, G), where $A = dom_f$, $B = cod_f$ and $G \subset A \times B$ such that $\forall x \in dom_f$, there exists a unique $y \in B$, such that $(x, y) \in G$.

Since $dom_f = \varnothing$, we have that $G \subset \varnothing \times B$, but this is actually empty. Why? If $\varnothing \times B$ is not empty, then there exists $(a, b) \in \varnothing \times B$, which is false, since this would imply that $a \in \varnothing$, which contradicts the definition of the empty set that says that it can have no elements (note that $\varnothing \in \varnothing$ is actually false).

With this, we have that $G = \varnothing$, thus, the only possible function from $\varnothing$ to B is $f = (\varnothing, \varnothing, B)$. Which proves that the empty set is initial.

But what about terminal? The empty set actually does not have any morphisms that arrives on it, since there is no function $f : A \to \varnothing$. The terminal sets in **Set** are actually all the singletons (sets with only one element), since for any $\{a\}$, there will be only one function $g : A \to \{a\}$, which is $g(x) = a$.

Another definition we have is that of a *zero* morphism. The idea here is that this morphism must take the elements of an object A to the zero element in B, for example, a for two vector spaces $\mathbb{R}^n$ and $\mathbb{R}^m$, the zero linear transformation $z : \mathbb{R}^n \to \mathbb{R}^m$ should take every vector n-dimensional vector to the $\mathbf{0}$ m-dimensional vector. In Category Theory we do not talk about morphisms according to how they act on the elements, but only in the objects. So we cannot define z by saying to which element it maps. Yet, there is a way to do this in Category Theory, which gives rise to the *zero* morphism definition.

Definition 2.13 (Zero Morphism). Let $\mathcal{C}$ be a category, and 0 be a zero object. A morphism $z : A \to B$ is a zero morphism if there exists two morphisms $f : A \to 0$ and $g : 0 \to B$, such that

$$z = g \circ f.$$

See that this makes intuitive sense. In our example, since we wish to take $v \in \mathbb{R}^n$ to $0 \in \mathbb{R}^m$, we first take all v to the zero object, which in the category of vector spaces will be the zero vector space, i.e. $\{0\}$ the space where $0 \in \mathbb{R}$ is the only element. So now all v are 0. Note

that every linear transformation from $\{0\}$ to $\mathbb{R}^m$ must take 0 to $\mathbf{0} \in \mathbb{R}^m$, otherwise, suppose that $g(0) = \mathbf{v} \neq \mathbf{0}$, hence for a scalar α,

$$g(0) = g(\alpha 0) = \mathbf{v} \neq \alpha \mathbf{v} = \alpha g(0).$$

This is a contradiction, since g is a linear transformation.

Theorem 2.14. Let $\mathcal{C}$ be a catgory with zero object 0. Then there exists a unique zero morphism between any two objects.

Proof. Let $A, B \in Ob_{\mathcal{C}}$. By the definition of the zero object, there exists a unique $f : A \to 0$ and $g : 0 \to B$, thus, $g \circ f$ is a zero morphism by definition and is unique, since there is no other f and g with these respective domain and codamain.

Moreover, note that if $z : A \to B$ is our zero morphism and $h : B \to C$, then

$$h \circ z = h \circ (g \circ f) = (h \circ g) \circ f.$$

But, $l = (h \circ g) : 0 \to C$, which means that $l \circ f$ is a zero morphism. The same argument works with a composition from the other direction. This means that compositions with zero morphisms return zero morphisms. □

2.7 Understanding Duality

In several fields of Mathematics, one is faced with the informal notion of a dual. Mathematicians define a concept, and call them the dual in some sense, for example, the dual vector space, the dual of an optimization problem, and many more. I always found puzzling what exactly held these things together, i.e. what was the underlying principle that made something a dual of another.

Fortunately, Category Theory has a very elegant answer. For a given category $\mathcal{C}$, the dual category is denoted by $\mathcal{C}^{op}$ where are the objects are the same, but the morphisms are inverted. This means that $Ob_{\mathcal{C}} = Ob_{\mathcal{C}^{op}}$, and for every $f \in Mor_(A, B)$, we have $f^{op} \in Mor_{\mathcal{C}^{op}}(B, A)$.

This definition gives rise to a very interesting result (observation), called the *Duality Principle*.

Definition 2.15 (Dual Property and Dual Statement). We say p^{op} is the dual property for p if for all categories

$$\mathcal{C} \text{ has } p^{op} \iff \mathcal{C}^{op} \text{ has } p.$$

For a statement s about a category $\mathcal{C}$, the dual statement is the same statement, but with regards to $\mathcal{C}^{op}$.

For example, "a category has an initial object if and only if the dual category has a terminal object". In this example, the property of having an initial object is the dual property of having a terminal object, since the above statement is true for any category. What about the dual statement? The dual for the statement "the category $\mathcal{C}$ has an initial object" is "the category $\mathcal{C}^{op}$ has an initial object". Note that the dual statement is not always true. And here is where we get the duality principle.

Definition 2.16 (Duality Principle). If a statement s is true for every category, then the dual statement is also true for every category.

Let's digest a bit what this principle states. If we can prove that a certain statement is true for any arbitrary category, then it's dual will also be true without any effort what so ever. Roman et al. [8] gives a nice example of this. We already prove that for any category, if an initial object exists, this initial object is unique up to an isomorphism. Note that this is a statement that is true for any category, so the duality principle applies, i.e. the dual statement is true. And what is the dual statement? That for every $\mathcal{C}^{op}$ the initial object is unique up to an isomorphism. But an initial object in $\mathcal{C}^{op}$ is a terminal object in $\mathcal{C}$. So we have, without any effort, that every terminal object is unique up to an isomorphism.

2.8 Categorical Product and Coproduct

In set theory, we are used to the notion of a Cartesian product. Similarly to how we did for isomorphism, the idea of a product can be generalized via Category Theory. Here is how it's done.

Definition 2.17 (Span). Let A, B be objects in a category $\mathcal{C}$. A span on A and B is a triple (Z, f, g) where $f : Z \to A$ and $g : Z \to B$ are morphisms in $\mathcal{C}$. This is shown in figure 9.

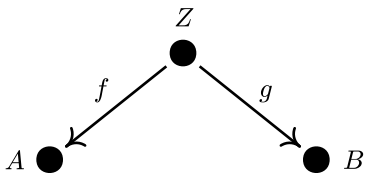


Figure 9: Diagrams showcasing a span between A and B.

Definition 2.18 (Categorical Product). Let A, B be objects in a category $\mathcal{C}$. A span $(A \times B, \pi_1, \pi_2)$ is called a product between A and B if for every span (Z, f, g) of A and B, there exists a unique morphism $h_{f,g} : Z \to A \times B$ such that $h_{f,g} \fatsemi \pi_1 = f$ and $h_{f,g} \fatsemi \pi_2 = g$. That is the same as saying that the diagram 10 commutes.

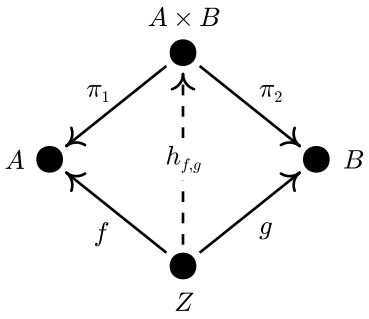


Figure 10: Diagrams showcasing the categorical product. Note that the dashed line is intended to highlight that the morphism $h_{f,g}$ is uniquely induced by f and g.

Note that this definition of a product is a **universal construction**, since it's done via existence and uniqueness. Another important aspect to note is that not every pair of objects in a category might have a product associated.

Theorem 2.19. For a category $\mathcal{C}$, a pair of objects A and B can have more than one product construction, but if this is the case, then both these constructions will be isomorphic.

Proposition 2.20 (Categorical Product vs Set Product). The categorical product generalizes the Cartesian product in set theory.

Proof. Consider the span $(X \times Y, \pi_1, \pi_2)$ where $\pi_1(x, y) = x$ and $\pi_2(x, y) = y$. Thus, for any span (Z, f, g) of A and B, make $h_{f,g}(z) = (f(z), g(z)) \in X \times Y$. This is how the Cartesian product works.

Let's drop the subscript in $h_{f,g}$. Now we have to show that h is a unique morphism, and that $h \fatsemi \pi_1 = f$ and $h \fatsemi \pi_2 = g$.

The second condition is trivially true, just note that

$$\forall z \in Z,\ \pi_1(h(z)) = \pi_1((f(z), g(z))) = f(z) \implies h \fatsemi \pi_1 = f,$$

and the same argument works for π_2 and g.

For uniqueness, consider $h' : Z \to X \times Y$ such that $h' \fatsemi \pi_1 = f$, and $h' \fatsemi \pi_2 = g$. If $h' \neq h$, then there is a $z \in Z$, such that $h(z) = (f(z), g(z)) \neq h'(z)$. Bu then, $\pi_1(h'(z)) \neq f(z)$ or $\pi_2(h'(z)) \neq g(z)$, which is a contradiction.

□

From the definition of a product, it's easy to think of possible dual constructions by just inverting the arrows (morphisms).

Definition 2.21 (Cospan). Let A, B be objects in a category $\mathcal{C}$. A cospan on A and B is a triple (Z, f, g) where $f : A \to Z$ and $g : B \to Z$ are morphisms in $\mathcal{C}$. This is shown in Figure 11.

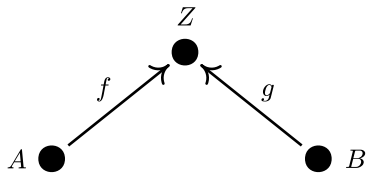


Figure 11: Diagrams showcasing a cospan between A and B.

Definition 2.22 (Categorical Coproduct). Let A, B be objects in a category $\mathcal{C}$. A cospan $(A + B, i_1, i_2)$ is called a product between A and B if for every span (Z, f, g) of A and B, there exists a unique morphism $h_{f,g} : Z \to A \times B$ such that $i_1 \fatsemi h_{f,g} = f$ and $i_2 \fatsemi h_{f,g} = g$. That is the same as saying that the diagram 12 commutes.

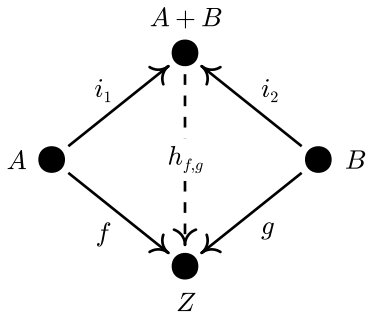


Figure 12: Diagrams showcasing the categorical coproduct. Note that the dashed line is intended to highlight that the morphism $h_{f,g}$ is uniquely induced by f and g.

While the categorical product was constructed to generalize the Cartesian set product, the coproduct was constructed in Category Theory, so the question is "to what does the coproduct corresponds in set theory?". The answer is the disjoint union! It's not a coincidence that we used "+" to symbolize it.

The idea of a product construction induced the notion of a product object $A \times B$. Yet, this construction also induces another definition, of the so called (co)product morphism.

Definition 2.23 (Product and Coproduct Morphism). Let $A, B, C, D \in Ob_{\mathcal{C}}$, $(A \times B, \pi_1^{A,B}, \pi_2^{A,B})$ and $(C \times D, \pi_1^{C,D}, \pi_2^{C,D})$ two products in the category $\mathcal{C}$, $f : A \to C$ and $g : B \to D$ two morphisms in $\mathcal{C}$. Hence, the morphism induced by $\pi_1^{A,B} \fatsemi f$ and $\pi_2^{A,B} \fatsemi g$ is called product morphism and denoted by $f \times g$, where

$$f \times g := (\pi_1^{A,B} \fatsemi f, \pi_2^{A,B} \fatsemi g) : A \times B \to C \times D.$$

Theorem 2.24. The product morphism $f \times g$ is the only morphism in $Mor_{\mathcal{C}}(A \times B, C \times D)$ such that

$$\pi_1^{C,D} \circ (f \times g) = f \circ \pi_1^{A,B}, \quad \pi_2^{C,D} \circ (f \times g) = g \circ \pi_2^{A,B}.$$

The coproduct morphism follows the same definition, but with coproducts.

Finally, one might be wondering what is an actual example of a product morphisms. For sets, it's the intuitive object, e.g. for two functions $f : A \to B$ and $g : C \to D$, the product morphism $f \times g : A \times B \to C \times D$ is just $(f \times g)(x, y) = (f(x), g(y))$.

2.9 Cartesian Categories

2.10 Pullback, Pushout and Equalizers

3 Functors

FUNCTORS PRESERVE ISOMORPHISMS? CHECK,WRITE ABOUT Another central definition in Category Theory is that of Functors. While morphisms relate objects inside a category, a Functor establishes a relation between categories, thus, it's one level higher in terms of abstraction.

3.1 What is a Functor?

Let's formally define a Functor.

Definition 3.1 (Functor). Let $\mathcal{C}$ and $\mathcal{D}$ be two categories. A functor $F : \mathcal{C} \Rightarrow \mathcal{D}$ is a pair of mappings with the following properties:

(i) a mapping between objects

$$F : Ob_{\mathcal{C}} \to Ob_{\mathcal{D}},$$

where for each $A \in Ob_{\mathcal{C}}$, $F(A) \in Ob_{\mathcal{D}}$.

(ii) a mapping between morphisms

$$F : Mor_{\mathcal{C}} \to Mor_{\mathcal{D}},$$

where there are two possibilities:

(a) **Covariant Functor**, in which

$$F : Mor_{\mathcal{C}}(A, B) \to Mor_{\mathcal{D}}(F(A), F(B)),$$

hence for a morphism $f : A \to B$, then $F(f) : F(A) \to F(B)$.

(b) **Contravariant Functor**, in which

$$F : Mor_{\mathcal{C}}(A, B) \to Mor_{\mathcal{D}}(F(B), F(A)),$$

hence for a morphism $f : A \to B$, then $F(f) : F(B) \to F(A)$.

(iii) Identities morphisms are preserved, i.e. for $A \in Ob_{\mathcal{C}}$

$$F(id_A) = id_{F(A)}.$$

(iv) Compositions are preserved, i.e. for $f \in Mor_{\mathcal{C}}(A, B)$ and $g \in Mor_{\mathcal{C}}(B, C)$,

(a) For a **Covariant Functor**,

$$F(f \circ g) = F(f) \circ F(g).$$

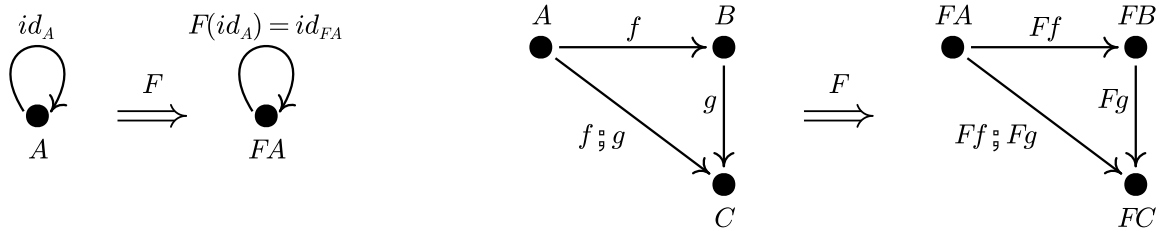
(b) For a **Contravariant Functor**,

$$F(f \circ g) = F(g) \circ F(f).$$

It's common for authors to refer to covariant functors only as functors, i.e. whenever someone say that F is a functor, it might be implied that it's a covariant functor. We'll also use this convention whenever it's not ambiguous, and we'll always. Also, we'll sometimes use FA to mean $F(A)$.

Again, the use of diagrams may help understand what is going on. The figure below illustrates the identity and composition preservation of Functors.

Covariant Functor



Contravariant Functor

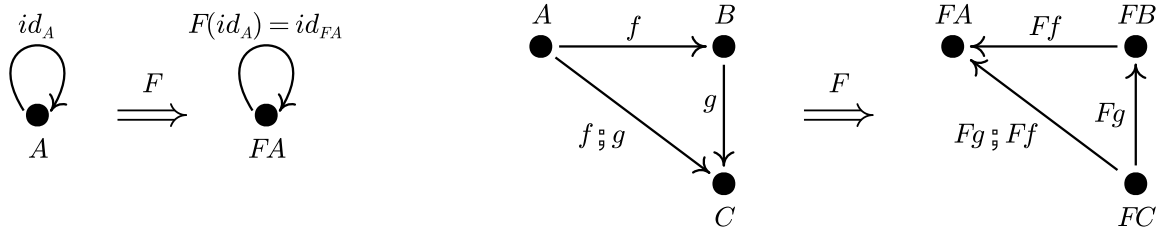


Figure 13: Diagrams showcasing the properties of Functors.

3.2 Category of Small Categories

One might realize that functors are acting on categories in a very similar way as morphisms do to objects. Indeed, we can define a functor composition to be such that for two functors $F : \mathcal{C} \Rightarrow \mathcal{D}$ and $G : \mathcal{D} \Rightarrow \mathcal{E}$, then $G \circ F$ is a functor from $\mathcal{C}$ to $\mathcal{E}$ where

1. For any $A \in Ob_{\mathcal{C}}$, $G \circ F(A) = G(F(A))$,
2. For any $f \in Mor_{\mathcal{C}}$, $G \circ F(f) = G(F(f))$.

We can also define an identity functor $I : \mathcal{C} \Rightarrow \mathcal{C}$, where $F(A) = A$ and $F(f) = f$.

Therefore, we might wonder whether there exists a category of all categories where objects are categories and morphisms are functors. The answer is "no". Similar to the set of all sets, it can be proven that this category does not exists. Yet, the category of all *small categories* does.

Remember, a small category is one where both morphisms and objects are sets. With this, let's prove our first theorem.

Theorem 3.2 (Category of Small Categories). Let $Ob_{\mathbf{SmCat}}$ be small categories and $Mor_{\mathbf{SmCat}}$ be functors. This constitutes a category.

Proof. To prove this, we'll use the fact that in Gödel-Bernays class set theory, the axiom 2.2 implies what is called a *comprehension scheme*.

Proposition 3.3 (Comprehension Scheme from Borceux [1]). If $\phi(x_1, ..., x_n)$ is a formula that the quantification occurs on set variables, then there exists a class A such that

$$(x_1, ..., x_n) \in A \iff \phi(x_1, ..., x_n).$$

Note that

$$\mathcal{C} := \langle Ob_{\mathcal{C}}, Mor_{\mathcal{C}} \rangle \in \mathbf{SmCat} \iff \mathcal{C} \text{ "is a category".}$$

Since every $\mathcal{C}$ is small, then the formula to check whether $\mathcal{C}$ is a category iterates over set variables, i.e. $Ob_{\mathcal{C}}$ and $Mor_{\mathcal{C}}$. Thus, the Comprehension Scheme proposition guarantees that **SmCat** exists. □

3.3 Types of Functors

Before we go on to provide examples of functors, let's present a way to classify different functors.

Definition 3.4 (Faithful, Full, Fully Faithful and Embedding). Here we follow Roman et al. [8]. Let F be a functor between categories $\mathcal{C}$ and $\mathcal{D}$.

1. F is **faithful** if for every $A, B \in Ob_{\mathcal{C}}$, $F : Mor_{\mathcal{C}}(A, B) \to Mor_{\mathcal{C}}(FA, FB)$ is injective.
2. F is **full** if for every $A, B \in Ob_{\mathcal{C}}$, $F : Mor_{\mathcal{C}}(A, B) \to Mor_{\mathcal{C}}(FA, FB)$ is surjective.
3. F is **fully faithful** if for every $A, B \in Ob_{\mathcal{C}}$, $F : Mor_{\mathcal{C}}(A, B) \to Mor_{\mathcal{C}}(FA, FB)$ is bijective.
4. F is an **embedding** of $\mathcal{C}$ in $\mathcal{D}$ if F is fully faithful and $F : Ob_{\mathcal{C}} \to Ob_{\mathcal{D}}$ is injective.

3.4 Subcategories

Definition 3.5 (Subcategory). Let $\mathcal{D}$ be a category. We say that $\mathcal{C}$ is a subcategory of $\mathcal{D}$ if $Ob_{\mathcal{C}} \subset Ob_{\mathcal{D}}$ and $Mor_{\mathcal{C}} \subset Mor_{\mathcal{D}}$, such that $\mathcal{C}$ is a category.

If for every $A, B \in Ob_{\mathcal{D}}$, we have that $Mor_{\mathcal{D}}(A, B) = Mor_{\mathcal{C}}(A, B)$, then $\mathcal{C}$ is a *full subcategory*.

From the definition of a functor, one might wonder whether for any functor $F : \mathcal{C} \Rightarrow \mathcal{D}$, the image $F(\mathcal{C})$ is a subcategory of $\mathcal{D}$, i.e. if $\langle Ob_{F(\mathcal{C})}, Mor_{F(\mathcal{C})} \rangle$ is a category where

$$Ob_{F(\mathcal{C})} := \{F(A) : A \in Ob_{\mathcal{C}}\}, \quad Mor_{F(\mathcal{C})} := \{F(f) : f \in Mor_{\mathcal{C}}\}.$$

The answer is no.

3.5 Relevant Examples of Functors

Now that we know what a functor is, let's showcase some relevant examples that might be useful to someone applying Category Theory to another field.

Example 3.1 (Power Set Functor). The power set functor $\mathcal{P} : \mathbf{Set} \to \mathbf{Set}$ takes a set to it's power set and a function $f : A \to B$ to the image function $Imf : \mathcal{P}(A) \to \mathcal{P}(B)$, i.e. for a subset $S \subset A$ in the domain, returns $f(S) := \{f(x) \ : x \in S\}$.

Another even more relevant example is the *contravariante* power set functor $F : \mathbf{Set} \to \mathbf{Set}$ that takes A to $\mathcal{P}(A)$ and f to the inverse image f^{-1}.

Example 3.2 (Identity Functor). This does what one might expect from the name. The identity functor is $1_{\mathcal{C}} : \mathcal{C} \to \mathcal{C}$, such that $1_{\mathcal{C}}(A) = A \in Ob_{\mathcal{C}}$ and $1_{\mathcal{C}}(f) = f \in Mor_{\mathcal{C}}$.

Example 3.3 (Inclusion Functor). For a subcategory $\mathcal{S}$ of $\mathcal{C}$, the inclusion functor is $I_{\mathcal{C}} : \mathcal{S} \to \mathcal{C}$, such that $I_{\mathcal{C}}(A \in Ob_{\mathcal{S}}) = A \in Ob_{\mathcal{C}}$ and $I_{\mathcal{C}}(f \in Mor_{\mathcal{S}}) = f \in Mor_{\mathcal{C}}$.

3.6 Formalizing Diagrams

We've drawn many diagrams representing either categories or portions of categories. Up until now, this has been done in a informal way. Yet, it's possible to define such diagrams rigorously. One of the reasons that formal diagrams are useful is that they allow us to talk about portions of some category $\mathcal{C}$ in a rigorous manner.

Definition 3.6 (Diagram). A $\mathcal{I}$-diagram (or just diagram) D of a category $\mathcal{C}$ is a functor $D : \mathcal{I} \to \mathcal{C}$, where $\mathcal{I}$ is called the index category.

The definition above is pretty much a "placeholder", i.e. any functor can be seen as a diagram since there is no conditions placed on the index category. It's common to say that D is an I-shaped diagram. The reason for this is that the functor D maps the objects and morphisms of $\mathcal{C}$ in the I category. This can be better understood from the figure below.

(Figure here of I shaped)

We define a *path* in D to be an n-tuple of morphisms in D where the codomain of the morphism matches the domain following morphism, e.g. for $f, g, h \in Mor_{\mathcal{I}}$, we have a path (Df, Dg, Dh) if the codomain of Df mathces the domain of Dg and the codomain of Dg matches the domain of Dh. The length of a path is equal to the number of morphisms. For a path $(Df_1, Df_2, ..., Df_n)$, the composition along such path is a morphism $Df_1 \fatsemi ... \fatsemi Df_n$.

We can now formally define a commutative diagram.

Definition 3.7 (Commutative Diagram). A diagram $D : \mathcal{I} \to \mathcal{C}$ is said to be commutative if for every pair of objects $D(A), D(B) \in Ob_{\mathcal{C}}$, every composition along paths from $D(A)$ to $D(B)$ are equal.

From the definition above, we say that the diagram shown below commutes if $h = f \fatsemi g$.

Aesthetic differences when drawing diagrams. If you've picked up any book on Category Theory, you'll note that almost every drawing of a diagram uses the label of a node instead of the black circle as we've done here. This is just a matter of aesthetics, yet, authors sometimes draw the black circles when they want to emphasize the "directed graph" nature of the diagram, i.e. they want to highlight that this diagram can be seen as a directed graph.

Therefore, the choice of how to draw these diagram comes down to aesthetic preferences. In these notes, we've opted for drawing the black dots.

4 Natural Transformations

Similar to how morphisms define relations between objects in a category, natural transformations define relations between functors. The term "natural" in natural transformations was coined by Eilenberg and MacLane (the founders of Category Theory) due to the fact that these transformations were developed with the aim of explaining why some constructions in Mathematics were "natural" while others were not.

4.1 Defining Natural Transformations

Similar to functors, the formal definition of natural transformations are somewhat obscure at first sight, by as we dig a bit, we see that the is some intuition behind it that makes it easier to understand it.

Definition 4.1 (Natural Transformations). Let $\mathcal{C}$ and $\mathcal{D}$ be categories, and let $F, G : \mathcal{C} \to \mathcal{D}$ be functors. A natural transformation $\alpha : F \Rightarrow G$ is such that:

(i) For all $A \in \mathcal{C}$, there exists $\alpha_A : F(A) \to G(A)$ that is a morphism in $\mathcal{D}$, i.e $\alpha_A \in Mor_{\mathcal{D}}(F(A), G(A))$;

(ii) (Naturality) For all $f \in Mor_{\mathcal{C}}(A, B)$, then

$$F(f) \fatsemi \alpha_B = \alpha_A \fatsemi G(f).$$

Remember that for a morphism $f : A \to B$, we have that $F(f) : F(A) \to F(B)$ and $G(f) : G(A) \to G(B)$. Since $\alpha_A : F(A) \to G(A)$ and $\alpha_B : F(B) \to G(B)$, then $F(f)$ composes with α_B and $G(f)$ composes with α_A, and our definition above works.

Another way to represent property (ii) in the definition of natural transformations is to affirm that the diagram below.

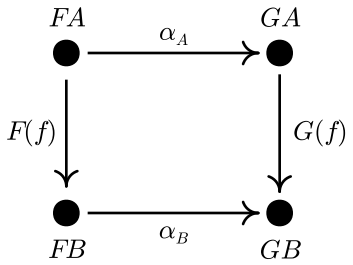


Figure 14: Commutative diagram of a natural transformation highlighting the commutative property of the definition.

From the definition, the natural transformation α is a transformation between functors that is in some sense associative, i.e. the order does not matter, we can apply α to FA and then apply Gf, or we can apply Ff to FA an then apply the natural transformation. Both return the same result.

4.2 The Category of Functors

The natural transformation is also ideal for defining a category of functors.

Definition 4.2. For categories $\mathcal{C}$ and $\mathcal{D}$, then $\mathcal{D}^{\mathcal{C}}$ is the category of functors from $\mathcal{C}$ to $\mathcal{D}$ where the morphisms are natural transformations, i.e.

$$Ob_{\mathcal{D}^{\mathcal{C}}} := \text{Functors from } \mathcal{C} \text{ to } \mathcal{D}$$

$$Mor_{\mathcal{D}^{\mathcal{C}}}(F, G) := \text{Natural Transformations from } F \text{ to } G.$$

Let's prove that the definition above is indeed a category. For that, we need to define a way to compose natural transformations that is associative, and we also need to define an identity morphism.

Proposition 4.3. In the category $\mathcal{D}^{\mathcal{C}}$, define the composition of two natural transformations $\alpha : F \to G$ and $\beta : G \to H$ as

$$\forall c \in \mathcal{C},\ \alpha_c \fatsemi \beta_c = (\alpha \fatsemi \beta)_c.$$

Then, using this definition, $\alpha \fatsemi \beta$ is indeed a normal transformation from $F \to H$, and this composition is associative.

Proof. First, since $\alpha_c \in Mor_{\mathcal{D}}(F(c), G(c))$, $\beta_c \in Mor_{\mathcal{D}}(G(c), H(c))$, $\gamma_c \in Mor_{\mathcal{D}}(H(c), J(c))$, then

$$\begin{aligned}\alpha \fatsemi (\beta \fatsemi \gamma) \iff \forall c \in \mathcal{C}, \alpha_c \fatsemi (\beta \fatsemi \gamma)_c &= \alpha_c \fatsemi (\beta_c \fatsemi \gamma_c) \\ &= (\alpha_c \fatsemi \beta_c) \fatsemi \gamma_c \\ &= (\alpha_c \fatsemi \beta_c) \fatsemi \gamma_c \\ &= (\alpha \fatsemi \beta)_c \fatsemi \gamma_c \iff (\alpha \fatsemi \beta) \fatsemi \gamma.\end{aligned}$$

We proved the associative part. Now, we need to show that such $\alpha \fatsemi \beta$ is a natural transformation from F to H. Take $f : c \to c' \in Mor_{\mathcal{C}}$. Consider $(\alpha \fatsemi \beta)_c : F(c) \to H(c)$, and $(\alpha \fatsemi \beta)_{c'} : F(c') \to H(c')$. Note that

$$F(f) \fatsemi \alpha_{c'} = \alpha_c \fatsemi G(f), \quad G(f) \fatsemi \beta_{c'} = \beta_c \fatsemi H(f).$$

Therefore, we have

$$\begin{aligned}F(f) \fatsemi (\alpha \fatsemi \beta)_{c'} = (F(f) \fatsemi \alpha_{c'}) \fatsemi \beta_{c'} &= (\alpha_c \fatsemi G(f)) \fatsemi \beta_{c'} \\ &= \alpha_c \fatsemi (G(f) \fatsemi \beta_{c'}) \\ &= \alpha_c \fatsemi \beta_c \fatsemi H(f) \\ &= (\alpha \fatsemi \beta)_c \fatsemi H(f).\end{aligned}$$

□

Since natural transformations are morphisms between functors in this category of functors, we can now ponder what would an isomorphism between natural transformations look like. Remember the definition of a categorical isomorphism 2.6. With such definition, we can prove the following result.

Theorem 4.4 (Natural Isomorphism [9]). Let $\mathcal{C}$ and $\mathcal{D}$ be categories, and $F, G : \mathcal{C} \to \mathcal{D}$ functors. A natural transformation $\alpha : F \to G$ is an isomorphism in $\mathcal{D}^{\mathcal{C}}$ if and only if $\alpha_c : F(c) \to G(c)$ is an isomorphism in $\mathcal{D}$ for every $c \in Ob_{\mathcal{C}}$.

Proof. Check proposition 5.3.2.12 from Spivak [9]. □

Example 4.1 (Natural Transformations in Sets). Consider the categories **1** and **Set**. We wish to construct the category $\mathbf{Set}^{\mathbf{1}}$, i.e. of functors from **1** to **Set**. Since **1** has only one object 1 and one morphisms id_1, then each functor $F : \mathbf{1} \to \mathbf{Set}$ is completely defined by $F(1) = S$, where S is a set, and, by the definition of a functor, $F(id_1) = id_S$. This means that every functor can be indexed by the set to which it takes the object of **1**, e.g. F_A is the functor that $F(1)$ is equal to set A.

Next, we want to define a natural transformation $\alpha : F_A \to F_B$. By definition, there are two criteria to satisfy. First, for every object $1 \in Ob_{\mathbf{1}}$ we have $\alpha_1 : F_A(1) \to F_B(1)$ such that $\alpha_1 \in Mor_{\mathbf{Set}}(F_A(1), F_B(1))$. Since the only object in **1** is 1, we only need to define α_1 to fully define α. Since α_1 must be a morphism of **Set**, then α_1 is a function from A to B.

Secondly, for every $f \in Mor_{\mathbf{1}}(1,1)$, it's required that

$$F_A(f) \fatsemi \alpha_1 = \alpha_1 \fatsemi F_B(f).$$

Again, since the only morphism in $\mathbf{1}$ is id_1, we only need to prove that

$$F_A(id_1) \fatsemi \alpha_1 = \alpha_1 \fatsemi F_B(id_1).$$

Note that $F_A(id_1) = id_A$ and $F_B(id_1) = id_B$, and by the definition of the identity, we have that for $g \in Mor(A, B)$, $id_A \fatsemi g = g \fatsemi id_B$. Hence, the second condition is trivially satisfied.

With this, we conclude that the natural transformations from functors F_A to F_B are, in a sense, isomorphic to functions from sets A to B.

Considering the category **SmCat** of small categories, we can actually prove that **Set** and $\mathbf{Set}^{\mathbf{1}}$ are isomorphic.

Consider the functor $I : \mathbf{Set}^{\mathbf{1}} \to \mathbf{Set}$ where $I(F_A) = A \in \mathbf{Set}$ and for any natural transformation $\alpha^{(f)}$ where $\alpha_1^{(f)} = f$, we have $I(\alpha^{(f)}) = f$. Also, define $I^{-1}(A) = F_A$ and $I^{-1}(f) = \alpha_f$. Thus, I defines an isomorphism between the two categories.

4.3 Equivalence Between Categories

We've used the existence of two functors $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$, where $F \fatsemi G = id_{\mathcal{D}}$ and $G \fatsemi F = id_{\mathcal{C}}$, to define that two categories are equivalent up to an isomorphism. Note that in the category of small categories, besides morphisms and objects we also have natural transformations. Thus, we can make use of natural transformations to define a new notion of equivalence between categories that is less strict then the existence of isomorphic functors.

Definition 4.5 (Equivalence of Categories). We say that a functor $F : \mathcal{C} \to \mathcal{D}$ defines an equivalence of categories if there exists another functor $G : \mathcal{D} \to \mathcal{C}$ and two natural transformations $\alpha : id_{\mathcal{C}} \to G \circ F$ and $\beta : id_{\mathcal{D}} \to F \circ G$, where α and β are natural isomorphisms, i.e. for any $c \in \mathcal{C}$, $\alpha_c : id_{\mathcal{C}}(c) \to (G \circ F)(c)$ is an isomorphism in $\mathcal{C}$, and for any $d \in \mathcal{D}$, $\beta_d : id_{\mathcal{D}}(d) \to (F \circ G)(d)$ is an isomorphism in $\mathcal{D}$.

Moreover, we say that F are G are mutually inverse equivalences.

The above definition is convoluted, so let's try to get some intuition. First, note that $id_{\mathcal{C}}(c) = c \in \mathcal{C}$, so the natural transformation α defines for each object $c \in \mathcal{C}$ a morphism α_c from c to $c' = G(F(c)) \in \mathcal{C}$, where c and c' are actually isomorphic.

Now it's a bit clearer what is going on. The definition is stating that two categories are equivalent if there is a pair of functors which are almost the inverse of one another, but instead of $(F \circ G)(c) = c$ we have that $(F \circ G)(c) = c'$ where c' is isomorphic to c. Thus,

our functors are not inverses, because they scramble the objects, but in the end, they just exchange objects that are isomorphic to one another, hence, they "keep everything almost the same".

4.4 Vertical vs Horizontal Compostion

We already introduced how we compose natural transformations in a category of functors $\mathcal{D}^{\mathcal{C}}$. This composition is referred by Spivak [9] as *vertical*, because it can be depicted by the Figure below.

Yet, there is another type of operation between natural transformations, called *Whiskering*, also referred as *horizontal composition* by Spivak [9]. This time, the name is inspired by the diagram below.

Definition 4.6 (Prewhiskering). Let $\mathcal{B}, \mathcal{C}, \mathcal{D}$ be categories, and define functors $G_1, G_2 : \mathcal{C} \to \mathcal{D}$, $F : \mathcal{B} \to \mathcal{C}$ and the natural transformation $\alpha : G_1 \to G_2$. A *postwhiskering* of α by $\mathcal{G}$ is denoted by $\alpha \diamond G : G_1 \circ F \to G_2 \circ F$, where for every $b \in Ob_{\mathcal{B}}$, $(\alpha \diamond F)_b : (G_1 \circ F)(b) \to (G_2 \circ F)(b)$ is defined to be $\alpha_{F(b)}$

Definition 4.7 (Postwhiskering). Let $\mathcal{B}, \mathcal{C}, \mathcal{D}$ be categories, and define functors $F_1, F_2 : \mathcal{B} \to \mathcal{C}$, $G : \mathcal{C} \to \mathcal{D}$ and the natural transformation $\beta F_1 \to F_2$. A *postwhiskering* of β by $\mathcal{G}$ is denoted by $G \diamond \beta : G \circ F_1 \to G \circ F_2$, where for every $c \in Ob_{\mathcal{C}}$, $(\beta \diamond G)_b : (G \circ F_1)(c) \to (G \circ F_2)(c)$ is defined to be $G(\beta_c)$.

From these definitions of prewhiskering and postwhiskering, we construct the Whiskering, which is just the composition of a prewhiskering with a postwhiskering.

Definition 4.8 (Horizontal Composition/Whiskering of Natural Transformations). Let $\mathcal{B}, \mathcal{C}, \mathcal{D}$ be categories, and define functors $F_1, F_2 : \mathcal{B} \to \mathcal{C}$ and $G_1, G_2 : \mathcal{C} \to \mathcal{D}$. For two natural transformations $\alpha : G_1 \to G_2$ and $\beta : F_1 \to F_2$, a *whiskering* (horizontal composition) is denoted by $\alpha \diamond \beta : G_1 \circ F_1 \to G_2 \circ F_2$ where:

$$\alpha \diamond \beta = (\alpha \diamond F_1) \circ (G_2 \diamond \beta) = (G_1 \diamond \beta) \circ (\alpha \diamond G_2).$$

Theorem 4.9 (Interchange for Whiskering). (image here)

Given the setup above, then we have

$$(\beta_2 \circ \beta_1) \diamond (\alpha_2 \circ \alpha_1) = (\beta_2 \diamond \alpha_2) \circ (\beta_1 \diamond \alpha_1).$$

5 Limits, Colimits

Limits and colimits are universal constructions that generalize some of the categorical concepts we already talked about, such as products, equalizers, pullbacks and pushouts. Limits have a terminal sort of universal properties. Colimits have an initial sort of universal properties.

They are of interest, because whenever a category is complete with respect to limits or colimits, these end up corresponding to some classical concept.

5.1 Cones and Cocones

Before talking about limits, we need to introduce the notion of a cone and it's dual, the cocone.

Definition 5.1 (Cone). Let $\mathcal{C}$ be a *small* category. The (left) cone $\mathcal{C}^{\triangleleft}$ is a category constructed by adding an object LC, called the cone point, into the set of objects of $\mathcal{C}$, and making it an initial object by guaranteeing that there is only one morphism l_A from LC to each $A \in Ob_{\mathcal{C}}$. In other words, $\mathcal{C}^{\triangleleft}$ is a category where:

$$Ob(\mathcal{C}^{\triangleleft}) := \{LC\} \sqcup Ob(\mathcal{C}),$$

$$Mor_{\mathcal{C}^{\triangleleft}}(A,B) := \begin{cases} Mor_{\mathcal{C}}(A,B) & \text{if } A,B \in Ob(\mathcal{C}), \\ \{l_A\} & \text{if } A = LC, B \in Ob(\mathcal{C}), \\ \{id_{LC}\} & \text{if } A = LC, B = LC, \\ \varnothing & \text{if } A \in Ob(\mathcal{C}), B = LC. \end{cases}$$

As we pointed out, the cocone is just the dual.

Definition 5.2 (Cocone (Right Rone)). Let $\mathcal{C}$ be a *small* category. The cocone $\mathcal{C}^{\triangleright}$ is a category constructed by adding an object RC, called the cone point, into the set of objects of $\mathcal{C}$, and making it an initial object by guaranteeing that there is only one morphism r_A from $A \in Ob_{\mathcal{C}}$ to each RC. In other words, $\mathcal{C}^{\triangleright}$ is a category where:

$$Ob(\mathcal{C}^{\triangleright}) := \{RC\} \sqcup Ob(\mathcal{C}),$$

$$Mor_{\mathcal{C}^{\triangleright}}(A,B) := \begin{cases} Mor_{\mathcal{C}}(A,B) & \text{if } A,B \in Ob(\mathcal{C}), \\ \{r_B\} & \text{if } A = RC, B \in Ob(\mathcal{C}), \\ \{id_{RC}\} & \text{if } A = RC, B = RC, \\ \varnothing & \text{if } A = RC, B \in Ob(\mathcal{C}), \end{cases}$$

Caution! When constructing, for example, the cone category, there can be only a single morphism leaving LC. One might be tempted to construct this category by drawing an

arrow from the initial object LC to every object in the category $\mathcal{C}$, but this might lead to more than one morphism from LC to an object.

Example 5.1 (Cone in Discrete Categories). Let $\underline{\mathbf{3}}$ be the discrete category with three objects.

Example 5.2 (Cone in Gr). Let $\underline{\mathbf{3}}$ be the discrete category with three objects.

5.2 Slice Categories

A slice category is a way to define categories of certain diagrams (functors) inside another category. The definition by itself is very obscure, so we instead present how to construct the slice category for *spans* (remember definition 2.17 and Figure 9). This example makes clear the motivation behind the slice category.

Example 5.3 (Category of Spans in $\mathcal{C}$). Consider a diagram $D : \underline{2} \to \mathcal{C}$. Note that the cone $\underline{2}^{\triangleleft}$ is a span as shown in the Figure below.

(Figure of span from $\underline{2}^{\triangleleft}$)

Now, consider the category $\mathcal{C}^{\underline{2}^{\triangleleft}}$, i.e. the category of functors $F : \underline{2}^{\triangleleft} \to \mathcal{C}$, where the morphisms are natural transformations. Each functor $F \in Ob_{\mathcal{C}^{\underline{2}^{\triangleleft}}}$ is a span inside category $\mathcal{F}$, as shown in the Figure below.

(Figure here)

The next step is to formally define the category of spans between two given objects, e.g. we want to formally define a category of spans in $A, B \in Ob_{\mathcal{C}}$, where each object is a functor $F : \underline{2}^{\triangleleft} \to \mathcal{C}$, such that $F(1) = A$ and $F(2) = B$.

The caveat here is that we have to do this in a categorical way, without actually applying restrictions to the object specifications. To do this, we define a functor $X : \underline{2} \to \mathcal{C}$, where $X(1) = A$ and $X(2) = B$.

Next, we define an inclusion functor $i : \underline{2} \to \underline{2}^{\triangleleft}$ which sends everything in $\underline{2}$ to the same point, but in the bigger category $\underline{2}^{\triangleleft}$ (similar to how we can define $i : \mathbb{N} \to \mathbb{R}$ where $i(1) = 1.0$).

We can now construct the following diagram:

Thus, if $F \circ i = X$, then the diagram above commutes and $F(1) = A, F(2) = B$, which means that F is a span of A and B. This is the categorical way to restrict our permissible functors to spans of A, B. Therefore, we define the category of spans between A, B as $\mathcal{C}_{\backslash X}$, where

$$Ob_{\mathcal{C}_{/X}} := \{F : \underline{2}^{\triangleleft} \to \mathcal{C} \mid F \circ i = X\}.$$
$$Mor_{\mathcal{C}_{/X}}(F, G) := \{\alpha : F \Rightarrow G \mid \alpha \diamond i = id_X\}.$$

Note that α is a natural transformation, and $\alpha \diamond i$ is the *prewhiskering* operation 4.6. The composition between morphisms is defined as in the category of functors.

We already explained the motivation for why the definition of $Ob_{\mathcal{C}_{/X}}$ is restricted to $F \circ i = X$, but not why $\alpha \diamond i = id_X$ in the morphisms. This comes from the fact that for $\alpha : F \Rightarrow G$, then $\alpha \diamond i : F \circ i \to G \circ i$. Since F and G are objects in $\mathcal{C}_{/X}$, we have $F \circ i = X$ and $G \circ i = X$, implying that $\alpha \diamond i : X \to X$, i.e. $\alpha \diamond i = 1_X$.

This proves that indeed such restriction is what we want for morphisms. The intuition is just that we want a natural transformation between spans A, B to exists only if α sends A to A and B to B, as shown in the figure below.

Surprisingly, the product $(A \times B, \pi_1, \pi_2)$ is the terminal object of the category of spans between A and B, and, as we'll see, this means that the product is the limit of $X : \underline{2} \to \mathcal{C}$.

So, in the example above, we've constructed the category of spans between A, B, which consisted the category of functors $S : \underline{2}^{\triangleleft} \to C$, restricted to $S \circ i = X$, where $X : \underline{2} \to \mathcal{C}$ is a diagram on the discrete category of two elements with $X(1) = A, X(2) = B$. We can therefore generalize this construction by considering an arbitrary category I instead of $\underline{2}$, and any functor (I-shaped diagram) $X : I \to \mathcal{C}$.

Definition 5.3 (Slice Category). Let $\mathcal{C}$ and $\mathcal{I}$ be two categories, and $i : I \to I^{\triangleleft}$ the inclusion functor. For a given functor $X : I \to \mathcal{C}$, the slice category of $\mathcal{C}$ over X is denoted $\mathcal{C}_{/X}$, where:

$$Ob_{\mathcal{C}_{/X}} := \{F : I^{\triangleleft} \to \mathcal{C} \mid F \circ i = X\}.$$
$$Mor_{\mathcal{C}_{/X}}(F, G) := \{\alpha : F \Rightarrow G \mid \alpha \diamond i = id_X\}.$$

By taking the dual definition, we can define the *coslice*.

Definition 5.4 (Coslice Category). Let $\mathcal{C}$ and $\mathcal{I}$ be two categories, and $i : I \to I^{\triangleright}$ the inclusion functor. For a given functor $X : I \to \mathcal{C}$, the coslice category of $\mathcal{C}$ over X is denoted $\mathcal{C}_{X/}$, where:

$$Ob_{\mathcal{C}_{X/}} := \{F : I^{\triangleright} \to \mathcal{C} \mid F \circ i = X\}.$$
$$Mor_{\mathcal{C}_{X/}}(F, G) := \{\alpha : F \Rightarrow G \mid \alpha \diamond i = id_X\}.$$

5.3 Defining Limits and Colimits

With the definition of slices and coslices, the definition of a limit and colimit is immediate.

Definition 5.5 (Limit). Let $\mathcal{C}$ and $\mathcal{I}$ be two categories, and $X : I \to \mathcal{C}$ a functor. The limit of X is a terminal object in the slice category $\mathcal{C}_{/X}$, which is denoted by $\lim X$ or $\lim_I X$.

Remember that a category can have several terminal objects, but they will all be isomorphic, with a unique isomorphism between them. Thus, in a sense, a limit will always be unique up to an isomorphism.

Again, we can easily define a dual definition.

Definition 5.6 (Colimit). Let $\mathcal{C}$ and $\mathcal{I}$ be two categories, and $X : I \to \mathcal{C}$ a functor. The limit of X is a initial object in the coslice category $\mathcal{C}_{X/}$, which is denoted by $\text{colim}X$ or $\text{colim}_I X$.

6 Adjoints

Adjoints, or, adjunctions, are not easily understood from the definition alone. A good reference for a more intuitive description is the blog Math3ma, where the intuition for adjoints was taken from.

6.1 Intuitive Explanation for Adjoints

Consider two functors $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$. We already saw that there is a weaker notion of equivalence between categories(4.5), where instead of requiring the existence of an isomorphism given by $F \circ G = id_{\mathcal{D}}$ and $F \circ G = id_{\mathcal{C}}$, we only ask for $F \circ G \cong id_{\mathcal{D}}$ and $G \circ F \cong id_{\mathcal{C}}$.

We can go further and devise an even weaker notion of This is where adjunctions come in. Instead of isomorphisms, we can ask for the existence of two morphisms (which in this context will be natural transformations), where one morphism is $\varepsilon : F \circ G \to id_{\mathcal{D}}$ (called counit) and another morphism is $\eta : id_{\mathcal{C}} \to G \circ F$ (called unit), and they are somehow related. The question is then, what such relation should be?

Note that for $c \in Ob_{\mathcal{C}}$, then $\eta_c : c \to G(F(c))$ is $c \mapsto G(F(c))$. Similarly, $\varepsilon_{F(c)} : F(G(F(c))) \to F(c)$ is $F \circ G \circ F(c) \mapsto F(c)$.

Now, there is a reason that our adjoints are either left or right. The adjunction is how we can "translate" between categories where one is not as "expressive" as the other. How so? Consider for example the case of Linear Algebra, where for a given matrix $F \in \mathbb{R}^{n \times m}$ such that $n > m$. Now, suppose that there exists a matrix $G \in \mathbb{R}^{m \times n}$ such that G is the pseudoinverse of F and $GF = I_m$, i.e. the *left inverse*. Since G is the pseudoinverse, we have that $FGF = F$ and $GFG = G$.

The pseudoinverse captures exactly the relation we were looking for. If F and G are functors, we want to categorically specify that for them to be adjoints, it's required that $F \circ G \circ F = F$ and $G \circ F \circ G = G$. This can be done using the natural transformations. Note that we want to make

$$F \circ G \circ F(c) = F \circ \eta_c(c) = F(c) = \varepsilon_c(c),$$
$$G \circ F \circ G(F(c)) = G \circ \varepsilon_{F(c)}(F(c)) = G(F(c)) = \eta_c(c),$$

where we considered η_c as a function, and $\eta_c(c) = G(F(c))$. Thus, the restriction we want to impose is that

$$F \circ \eta_c(c) = \varepsilon_c(c), \quad G \circ \varepsilon_{F(c)}(F(c)) = \eta_c(c).$$

In this example, the category of vector spaces in $\mathbb{R}^m$ is "smaller" than $\mathbb{R}^n$, yet, we want to somehow translate between them. The idea of how to properly translate between vector spaces is captured by inverse matrices. When it's not possible to find inverse, the *best translation* is captured by the pseudoinverse, which is what the adjoint is trying to capture. The pseudoinverse G is the left ajoint of F.

Hopefully, this motivates the idea of adjoints. In a sense, one wishes to formalize this types of operations in which there is some sort of translation between categories with different expressiveness. We've shown how this is somehow related to weakening the requirement for the existence of isomorphisms. Now, let's indeed formalize a definition.

6.2 General Definitions

The definition we are going to present assumes that both categories are locally small. This will be enough for most applications. Yet, there are a more complicated definition that also takes into account categories that are not locally small. Remember that a category $\mathcal{C}$ is locally small if for every pair of objects A and B, then $Mor_{\mathcal{C}}(A, B)$ is a set.

Definition 6.1. Let $\mathcal{C}$ and $\mathcal{D}$ be locally small categories. The adjoints (adjunction) between $\mathcal{C}$ and $\mathcal{D}$ are the functors

$$L : \mathcal{C} \to \mathcal{D}, \quad R : \mathcal{D} \to \mathcal{C},$$

together with a natural transformation α such that for every $A \in \mathcal{C}$ and $B, \in \mathcal{D}$,

$$\alpha_{A,B} : Mor_{\mathcal{D}}(L(A), B) \xrightarrow{\cong} Mor_{\mathcal{C}}(A, R(B)),$$

i.e. for every pair of objects $A \in \mathcal{C}$ and $B \in \mathcal{D}$, there is an isomorphism between the sets $Mor_{\mathcal{D}}(L(A), B)$ and $Mor_{\mathcal{C}}(A, R(B))$.

Adjoints in Category Theory vs. Linear Algebra. Names are suggestive in Mathematics, but even more so in Category Theory. Note that the definition above of an adjoint resembles somewhat the idea of adjoint linear transformation in Linear Algebra. Remember:

$$\langle Ax, y\rangle = \langle x, A^* y\rangle, \quad \forall x, y \in X,$$

where X is our vector space and $A : X \to X$ is a linear transformation.

Proposition 6.2. Let $\mathcal{C}$ and $\mathcal{D}$ be two locally small categories, and $F : \mathcal{C} \to \mathcal{D}$ a functor. If both functors $G, G' : \mathcal{D} \to \mathcal{C}$ are right (or left) adjoints to F, then there exists a natural isomorphism between G and G'.

7 Graphs and Schemas in Category Theory

IMPROVE THIS SECTION! IT'S ALL OVER THE PLACE. A schema is a tuple (G, R) where G is a graph and $R \subset Path_G \times Path_G$ is a set of relations in the paths of G. Similar to how a monoid can be seen as generating a category, a schema also generates a category. Actually, the category is schemas is equivalent to the category of small categories.

TLDR. Every category has an underlying graph, which can be obtained via a forgetful functor $|-| : \mathbf{SmCat} \to \mathbf{Set}^{\mathbf{Gr}}$. There is also a free category functor $\langle - \rangle : \mathbf{Set}^{\mathbf{Gr}} \to \mathbf{SmCat}$ which takes a graph $G : \mathbf{Gr} \to \mathbf{Set}$ and returns a category $\langle G \rangle$.

From the free category, we generate a quotient category $\langle G, R \rangle$ via a quotient map q_R that sends $q_R(f) = q_R(g)$ if $(f, g) \in R$. Thus, to obtain schemas from categories, one first extracts the underlying graph, then generates the free category applying the free functor, and finally applies the quotient map to the set of relations R, thus getting the schema for that category, which is called the presentation of that category.

We can also construct a functor from schemas to small categories. Just send the vertices to be objects, and the morphisms to be the paths modulo the relation R. The composition of morphisms is just the path concatenation.

Now that some of the basics of Category Theory has been introduced, we can more easily talk about some actual applications. Still, when trying to apply a categorical view to a subject, we need to formalize concepts into categories, functors, natural transformations and so on, in order to be able to incorporate the results from Category Theory.

We begin this section by formalizing some of the tools that we have been using informally, which are the diagrams. This helps us clarify the connection between graphs and categories. Next, we define what are schemas, presentations, $\mathcal{C}$-sets, free categories and preorder reflections.

The rest of the section focuses on showing how these concepts are used to model data structures/databases, which can very elegantly be understood in Category Theory, and which help us deal with the very common issue of data migration.

7.1 Underlying Graphs

Our definition of diagrams was in no way "visual", i.e. when we draw diagrams, we are actually drawing graphs. Therefore, we have to formalize how we relate graphs and categories. For that, we'll use both the category of graphs **Gr**, the category of small categories **SmCat** and some others, which will be introduced.

The idea here is to show that we can define a functor G that takes categories and returns graphs in $Ob_{\mathbf{Gr}}$. Similarly, we want to know if and when we can use graphs to generate categories. As we shall see, this can indeed be done, but will be left for the following section.

Remember, a graph G is an object of **Gr** and it consists of a quadruple (V, A, src, tgt). In **Gr**, morphisms between categories are graph homomorphisms. You'll see that the definition of a graph homomorphism is actually very similar (same) to that of a functor.

Definition 7.1 (Graph Homomorphism [9]). For two graphs $G = (V, A, src, tgt)$ and $G' = (V', A', src', tgt')$, a graph homomorphism is a function $f : G \to G'$ where f consists in two functions $f_0 : V \to V'$ and $f_1 : A \to A'$, such that

$$f_1 \fatsemi src' = src \fatsemi f_0, \quad f_1 \fatsemi tgt' = tgt \fatsemi f_0.$$

Definition 7.2 (Underlying Graph of a Category). Every small category $\mathcal{C}$ has an underlying graph which consists of

$$G(\mathcal{C}) := \begin{cases} \text{Vertices} = Ob_{\mathcal{C}} \\ \text{Arrows} = Mor_{\mathcal{C}} \\ \text{Source} = \{dom(f) : f \in Mor_{\mathcal{C}}\} \\ \text{Target} = \{cod(f) : f \in Mor_{\mathcal{C}}\}. \end{cases}$$

Moreover, we can actually define a functor $G : \mathbf{SmCat} \to \mathbf{Gr}$, where $G(\mathcal{C}) \in Ob_{\mathbf{Gr}}$. Remember that functor in the category of small categories are composable, hence, for any functor $F : \mathcal{C} \to \mathcal{D}$, we can do $G(F) : G(\mathcal{C}) \to G(\mathcal{D})$, which is a graph homomorphism.

Note that we restricted our underlying graph to small categories so that the vertices and arrows in the graph are actually sets.

7.2 Generating Categories from Graphs via Free Functors

We showed in the previous section how we can take a small category and turn it into a graph. But what about the other way around? Can we take a graph and turn it into a category? More so, can any small category be generated from a graph? As we'll see, the answer to the last question is a no. Yet, there is a something very similar to a graph that indeed can generate every small category, and these are what we call a schema. We leave the discussion of schemas to the next section.

Definition 7.3 (Paths [9]). Let $G := (V, A, src, tgt)$ be a graph. A *path* of graph G is an ordered list of arrows indexed by the starting vertex, e.g. $(a_1, ..., a_n)_v$, such that $src(a_i) = tgt(a_{i+1})$ for $i \in \{1, ..., n-1\}$. The length of the path is equal to the number of arrows in the list. The empty path $()_v$ is a path of length 0 starting at vertex v. We use $Path_G^{(n)}$ to represent the set of paths with length n and $Path_G$ to be the set of all paths in G. Thus, $Path_G^{(0)}$ is equal to the set of vertices in the graph.

We can define functions $\overline{src}, \overline{tgt} : Path_G \to V$, which do the same thing that the source and target functions do in arrows, but applied to paths.

Lastly, one can concatenate paths via $*$, such that for $p = (a_1, ..., a_n)$ and $q = (b_1 = a_n, ..., b_m)$, then

$$p * q = (a_1, ..., a_n = b_1, b_2, ..., b_m).$$

Definition 7.4 (Path Functor). Given a graph $G := (V, A, src, tgt)$, we can construct a new graph

$$Path(G) := (V, Path_G, \overline{src}, \overline{tgt}),$$

which we'll called the path graph version of G. Moreover, we can actually define a functor $Path : \mathbf{Gr} \to \mathbf{Gr}$, which takes graphs to their path graphs, i.e. $Path(G) \in \mathbf{Gr}$ and graph homomorphisms $f : G \to G'$ to $Path(f) : Path(G) \to Path(G')$, where $P(f_0) = f_0$ and

$$P(f_1)(a_1 * a_2 * ... * a_m) = f_1(a_1) * f_1(a_2) * ... * f_1(a_m), \forall a_1, ..., a_m \in A.$$

An example of a graph an it's path graph is the graph in figure 3 and 2, respectively. As we can see, the path graph contains an arrow for each morphism in the **2** category, which is the one represented in these diagrams. Thus, for any path graph $Path(G) \in Ob_{\mathbf{Gr}}$, we can define a category $\mathcal{C}(Path(G))$ by assigning vertices to objects, making $Mor_{\mathcal{C}(Path(G))}(v_1, v_2)$ equal to the set of paths such that $\overline{src}(Path(G)) = v_1$ and $\overline{tgt}(Path(G)) = v_2$, and sending the empty paths $()_v$ each to the identity morphism.

Since we can define a path graph for each graph G, we can then create a functor that sends a graph G to the category generated by the path graph of G. This functor is what we call the *free category functor*.

Definition 7.5 (Free Category). Let $G := (V, A, src, tgt)$ be a graph and $\mathcal{C}$ a small category. The free category functor $Free : \mathbf{Gr} \to \mathcal{C}$ is such that for $G \in Ob_{\mathbf{Gr}}$, $Free(G) \in Ob_{\mathcal{C}}$ sends G to the category $\mathcal{G}$ constructed from $Path(G)$. The category $\mathcal{G}$ is constructed by making $V = Ob_{\mathcal{G}}$, $Mor_{\mathcal{C}(Path(G))}(v_1, v_2)$ equal to the set of paths such that $\overline{src}(Path(G)) = v_1$ and $\overline{tgt}(Path(G)) = v_2$, and sending the empty paths $()_v$ each to the identity morphism. For graph homomorphisms $f : G \to G'$, send $F(f)$ to $Path(f)$.

We can now generate categories from graphs. But can this method generate every possible small category? The answer is no. Here is an example. Consider the graph (diagram) in the figure 15. If we apply the free functor to it, we come up with a category $\mathcal{G}$ where $Ob_{\mathcal{G}} = \{A, B, C, D\}$ and $Mor_{\mathcal{G}} = \{id_A, id_B, id_C, id_D, f, g, h, i, f \fatsemi i, g \fatsemi h\}$. Thus, the free functor is not able to generate the category where $f \fatsemi i = g \fatsemi h$, i.e. where there is only one path of length 2 from A to D.

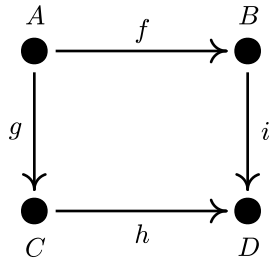


Figure 15: Example of ambiguous generated category from graph.

7.3 Categories from Preorders

In a sense, the free functor generates categories from graphs with the highest number of different morphisms as possible. What about the other end of the spectrum? Can we generate categories from graphs with the least number of possible morphisms? Yes, and the key for this are preorders.

First, let's present a somewhat related result, but that is not necessary for our goals.

Proposition 7.6 (Preorders to Graphs). Let **Pr** be the category of preorders and **Gr** the category of graphs. There exists a functor $P : \mathbf{Pr} \to \mathbf{Gr}$.

Proof. Define P as the following. For any preorder $\mathcal{X} = (X, \leq_X) \in Ob_{\mathbf{Pr}}$, $P(\mathcal{X})$ is a graph with vertices equal to set X and an arrow going from $a \in X$ to $b \in X$ if $a \leq_X b$. Remember that $\leq_X$ is a binary relation $R_{\mathcal{X}}$, i.e. it's nothing more than a subset of $X \times X$. Thus, $P(\mathcal{X}) = (X, R_{\mathcal{X}}, src_{\mathcal{X}}, tgt_{\mathcal{X}})$ where the source and the target functions are just the projection functions $\pi_1(x, y) = x$ and $\pi_2(x, y) = y$, respectively.

A morphism between preorders $(X, \leq_X)$ and $(Y, \leq_Y)$ is any monotone function that $x \leq_X x' \implies f(x) \leq_Y f(x')$. We need to take f to a graph homomorphism between graphs $P(\mathcal{X})$ and $P(\mathcal{Y})$. For that, make $P(f)$ equal to a tuple (f_0, f_1) where $f_0 : X \to Y$ is equal to the original f, and $f_1 : R_{\mathcal{X}} \to R_{\mathcal{Y}}$ is just $(x_1, x_2) \mapsto (f(x_1), f(x_2))$. Now, we need to check if this satisfies the graph homomorphism property.

$$(src_{\mathcal{Y}} \circ f_1)(x_1, x_2) = src_{\mathcal{Y}}(f(x_1), f(x_2)) = f(x_1) = f_0(x_1) = (f_0 \circ src_{\mathcal{Y}})(x_1, x_2).$$

The same argument works for the tgt, hence, $P(f)$ is indeed a graph homomorphism. □

Although the result above might be useful, our goal is to generate categories from graphs with the least amount of morphisms. This can be done by turning graphs into preorders and then preorders into categories.

Let's turn a graph $G = (V, A, src, tg)$ into a preorder $\mathcal{P} = (P, \leq_P)$[3]. First, make $P = V$. Secondly, collapse every parallel arrow into a single tuple, i.e. if $a_1, a_2 \in A$ such that $src(a_1) = src(a_2)$ and $tgt(a_1) = tgt(a_2)$, collapse them into $(src(a_1), tgt(a_1)) \in \leq_P$. Hence, this generates a preorder.

To get a category from a preorder, just use the procedure we already explained, where $x \leq_P y$ defines a morphism from x to y, and $P = Ob_{\mathcal{P}}$.

Similarly to how we came up with an underlying graph from a category, we can come up with a preorder from a category. This is deemed a preorder reflection.

Definition 7.7 (Preorder Reflection). Let $\mathcal{C}$ be a small category. We can obtain a preorder $(C, \leq_C)$ from $\mathcal{C}$ by making $Ob_{\mathcal{C}} = C$ and making $c_1 \leq_C c_2$ if $Mor_{\mathcal{C}}(c_1, c_2) \neq \emptyset$. Such preorder is called the preorder reflection of $\mathcal{C}$.

7.4 Generating Categories from Schemas

As we saw in the previous section, the free functor is a way to construct a category from a graph, but it is not able to produce every possible category. This implies that a graph is not expressive enough to determine a unique category, i.e. from a simple drawing of a graph we can imply different categories. Thus, we need more information added to the graph in order to know exactly what category it generates. We saw that another way to generate a category from a graph is by turning a graph into a preorder and then a preorder into a category. But again, this method is limited. Actually, both methods live in a sense in opposite sides of a spectrum [3]. So how can we generate the categories living in the center of the spectrum?

Well, we already did this throughout the text by writing equations such as $f \fatsemi i = g \fatsemi h$ at the bottom of the diagrams. This is the key to solve our problem. We can generate a free category from a graph, and then reduce the number of morphisms by equating them. If we put enough equations, we end up with the category generated via the preorder. These equations are formalized by using what Spivak [9] calls a *path equivalent declaration* (PED). A graph together with such equations will determine what we call a *schema*.

Definition 7.8 (Path Equivalence Declaration (PED) [9]). For two paths $p, q \in Path_G$, where $Path_G$ is the set of paths in G, a PED is an expression of the form $p \simeq q$, where $\overline{src}(p) = \overline{src}(q)$ and $\overline{tgt}(p) = \overline{tgt}(q)$. Moreover, a set of PED's form an equivalence relation in $Path_G$, such that if $p \simeq q$, $k \simeq l$ and the target of p and q is the same as the source of k and l, then $p * k \simeq q * l$. The set of PED's on G is also called a congruence on G.

Definition 7.9 (Schema [9]). A schema is a tuple $(G, \simeq)$ consisting of a graph G and a set of PED's denoted by $\simeq$.

The next section tries to define the morphisms between schemas in a way that is compatible with categories. The short version is that we indeed can define morphisms between categories

[3]Remember that $\leq_P \subset P \times P$

and thus define a category **Sch** of schemas. It can then be proven that **Sch** is equivalent (see definition 4.5) to **SmCat**[4], meaning that we can indeed use schemas to represent every small category. This gives rise to the definition of a presentation.

Definition 7.10 (Presentation of a Category). The presentation of a small category $\mathcal{C}$ is the schema that generates it. If the schema has a finite graph with finite path equations (PEDs), then we call it the finite presentation of $\mathcal{C}$.

Note that the term "presentation" is very suggestive. It's used to symbolize that a category can be represented by a (sometimes) finite generator, even if the category has, for example, infinite morphisms. One example of this is the category of natural numbers as a monoid $(\mathbb{N} \cup \{0\}, +, 0)$, which we already talked about. How can we present this category as a schema? Simple, just draw a vertex with one loop and no PED, i.e. the free functor applied to the graph of one vertex and one path.

7.5 More on Generating Categories from Schemas *

This is more technical and tries to formalize how we define morphisms between schemas and the category **Sch**.

Since we are now dealing with schemas instead of graphs, we need to define what are morphisms between schemas. Unfortunately, this is not as straightforward as we might think. One of the reasons for this is that our PED is an equivalence between paths on a graph, and not the arrows. This means that a graph homomorphism that preserve the PEDs is not actually the "correct" morphism between schemas.

How do we go about defining this morphism? What is the guiding principle? To understand why we'll construct the morphism in the way that we will, one has to understand what is the goal in mind. Our goal is to define a category of schemas, and prove that it is equivalent to **SmCat**. Thus, our morphism between schemas has to be "functorial", since functors are how we define morphisms in **SmCat**. We know that graph homomorphisms are the equivalent definition for the category **Gr**, so we want to find a similar way to relate schemas. Since schemas as graphs with congruence relations, a good start is to use graph homomorphisms. Yet, as we already stated, it's not so simple.

To define a morphism between schemas, we actually first turn the graphs into the path graphs, and then we send vertices to vertices (these do no change), and paths to paths while maintaining the PED. This process can be "simplified" a bit by skipping the step of sending a graph to it's own path, and sending it directly to the paths in the target graph. These steps are delineated below.

First, remember that $Path : \mathbf{Gr} \to \mathbf{Gr}$ is a functor that takes graphs to it's path graph, and a graph homomorphism $f : G \to H$ to $Path(f) : Path(G) \to Path(H)$ as described

[4]See Spivak [9].

in 7.4. For two schemas $\mathcal{S} = (G, \simeq_G)$ and $\mathcal{S}' = (G', \simeq_{G'})$, our goal is to define a graph homomorphism between $Path(G)$ and $Path(G')$, and then enforce the PEDs.

For the two graphs G, G', suppose that we have a graph homomorphism $f : G \to Path(G')$ (yes, between G and the path graph of G', not between G and G'), which is valid, since $Path(G')$ is also a graph. We can then apply the $Path$ functor to f, which will give $Path(f) : Path(G) \to Path(Path(G'))$. Note that $Path(Path(G'))$ is a graph where the arrows are paths of paths, e.g. if G' has two arrows f from a to b and g from b to c, then $Path(G')$ has six arrows, which are the paths then $Path(G')$ has six arrows, which are the paths $()_a, ()_b, ()_c, (f)_a, (g)_b, (f, g)_a$ and $Path(Path(G'))$ has $(()_a)_a, ((), (f)_a)_a, (()_b)_b, ((), (g)_b)_b$...

We can then define a function $\mu_{G'} : Path(Path(G')) \to Path(G')$ by concatenating the paths of paths, e.g. $\mu(((f)_a, (g)_b)_a) = \mu((f, g)_a)$.

Putting everything together, if we have the graph homomorphism $f : G \to G'$, it induces $Path_f : Path(G) \to Path(G')$, where $Path_f = Path(f) \fatsemi \mu_{G'}$.

Definition 7.11 (Schema Morphism). Let $G = (V, A, src, tgt)$ and $G' = (V', A', src', tgt')$ be two graphs, and $\mathcal{S} = (G, \simeq_G)$ and $\mathcal{S}' = (G', \simeq_{G'})$ be two schemas. A schema morphism $f : \mathcal{S} \to \mathcal{S}'$ is a map that sends the paths in G to paths in G' preserving the path endpoints, path concatenations and path equivalences (PED). More specifically, f is a graph homomorphism $f : G \to Path(G')$ such that for two paths $p, q \in Path_G$,

$$\text{if } p \simeq_G q, \text{ then } Path_f(p) \simeq'_G Path_f(q).$$

Where $Path_f : Path(G) \to Path(G')$ is the induced function $Path_f = Path(f) \fatsemi \mu_{G'}$.

Check Spivak [9] section 5.4.1 for a more thorough exposition.

Definition 7.12 (Category Sch). The category **Sch** has schemas as objects, and schema morphisms as morphisms. Again, check Spivak [9] for a more thorough explanation on how composition and identity work for this category.

Lastly, it can be shown that **Sch** $\simeq$ **SmCat**, thus, every category generates a schema, and every schema generates a category.

7.6 $\mathcal{C}$-Sets and Instances

A very important tool for applications of Category Theory is the concept of $\mathcal{C}$-Sets, which is a functor $I : \mathcal{C} \to$ **Set**, also known as a set-valued functor on $\mathcal{C}$ [3]. These functors are important because, as we'll see further on this section, they model the actual data inside a database.

Definition 7.13 ($\mathcal{C}$-Set). Given a *small* category[5] $\mathcal{C}$, a $\mathcal{C}$-set is a functor $I : \mathcal{C} \to$ **Set**. Also, a finite $\mathcal{C}$-set is a functor $I : \mathcal{C} \to$ **FinSet**. This functor I is also called an instance when it's referencing a schema.

[5]Remember, a small category is one where both $Ob_{\mathcal{C}}$ and $Mor_{\mathcal{C}}$ are sets.

The category $\mathbf{Set}^{\mathcal{C}}$ is the category of $\mathcal{C}$-Sets. This category has useful mathematical properties for data structures [6].

One specially interesting feature of such $\mathcal{C}$-Set functors is that they represent *instances* of schemas. For example, consider the category of graphs **Gr** as shown below. This diagram models graphs in general. How can we use Category Theory to obtain an actual example of graph, i.e. an *instance* of this category **Gr**? The answer is that the functor $G : \mathbf{Gr} \to \mathbf{Set}$ is exactly an example of a graph. In other words, for any category $\mathcal{C}$, each functor $I_1 : \mathcal{C} \to \mathbf{Set}$ is

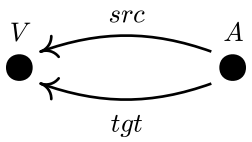


Figure 16: Diagram of category **Gr**.

8 What are Sets?

This section is based on Leinster [4].

When defining *small* and *locally small* categories, we need to differentiate between a *class* and a *set*. Anyone familiar with Russel's paradox on the set of all sets can appreciate why such distinction might be relevant.

One way to solve Russel's paradox was via Zermelo-Frankael and Choice (ZFC) axioms. Instead of strictly defining a set, the ZFC define what properties a set should have. Although this approach is the one assumed by most mathematicians, what ZFC calls a "set" does not actually match with how mathematicians use it. An example of the oddity in the definition of set's by ZFC is that elements of sets are also sets, so one could ask questions like "what are the elements of π?" [4].

Hence, instead of ZFC, we'll introduce here William Lawvere axioms as presented in Leinster [4]. Although less common, such system is more in sync with Category Theory, which is the subject at hand, and at the same time, it seems to more accurately describe what we mean by "sets".

8.0.1 Lawvere's Elementary Theory of the Category of Sets (ETCS)

As we said, to define a set we'll actually determine the properties that such object possesses. Thus, anything with such properties we'll be called a set. Of course, when stating such definition, we'll use terms that are again not tightly defined. But this is just part of life, since without such artifice, we would end up with circular definitions.

Let's now introduce the 10 axioms that make ETCS. This system of axioms is actually weaker (more general) than ZFC, and it can be shown to correspond to "Zermelo with bounded comprehension and choice" [4].

Although this axiomatization does not require Category Theory, we'll see that in some sense it has a categorical "flavor" to it.

Before stating the axioms, let's present some definitions that we'll be used in the axioms themselves. Note that these definitions only make sense once the axioms are established. But we present them now in order to make the exposition of ETCS cleaner.

Definition 8.1 (Terminal Set). A set T is called **terminal** in ETCS if for every set X there is only one function $f : X \to T$.

The terminal set is a way to define a single element set without relying on the definition of an element. In order to prove that this is indeed the case, we would need to clarify when two functions are the same, which will only be done after we present our axioms. It can

be shown that every terminal set is unique up to an isomorphism, so one could use T to represent every terminal set.

Interestingly, if we are working in a context with a restricted collection of functions, then, a set T may behave as a single element set, while it may have multiple elements in another context. Consider for example, that $T = [0.5, 1]$, and we are in the context of functions that return natural numbers. Thus, for any set X, there exists only one function $f : X \to T$, which always returns 1.

As we've seen, the category of sets ($\boldsymbol{Set}$) will consist of $\langle Ob_{\boldsymbol{Set}}, Mor_{\boldsymbol{Set}} \rangle$, where $Ob_{\boldsymbol{Set}}$ is the collection of every set, and $Mor_{\boldsymbol{Set}}$ is the collection of every function. In the ETCS, the collection of every set will not be a set itself.

Definition 8.2 (Element of a Set). Given a set X, we write $x \in X$ to mean $x : T \to X$ where T is a terminal set.

Note that in this definition of an element, what we call an element of X is actually a function. Also, for $f : X \to Y$, then $f \circ x$ is a function from T to Y, i.e. it is an element of Y, which we write as $f(x) \in Y$.

Definition 8.3 (Cartesian Product). Given sets X and Y. The Cartesian product of X and Y is a set P, with functions $p_1 : P \to X$ and $p_2 : P \to Y$, such that for any set Z and functions $f_1 : Z \to P$ and $f_2 : Z \to P$, there exists a unique function $F = (f_1, f_2) : Z \to P$ where

$$p_1 \circ (f_1, f_2) = f_1, \quad p_2 \circ (f_1, f_2) = f_2.$$

Note that the Cartesian Product determines not only a product set, but also the projection functions. Similar to terminal sets, for any sets X and Y, the triple (P, p_1, p_2) are unique up to an isomorphism. Thus, we could fix (P, p_1, p_2) to be represented by $(X \times Y, \pi_1^{X \times Y}, \pi_2^{X \times Y})$.

Definition 8.4 (Function set). Let X and Y be two sets. A **function set** from X to Y is a tuple (F, ε), where F is a set and ε is a function $\varepsilon : F \times X \to Y$ such that for all sets Z and functions $q : Z \times X \to Y$, there exists a unique function $\overline{q} : Z \to F$ with $q(t, x) = \varepsilon(\overline{q}(t), x)$ for all $t \in Z$ and $x \in X$.

Definition 8.5 (Inverse Image). Let $f : X \to Y$ be a function and $y \in Y$. The **inverse image** of y under f is a tuple (A, j) where A is a set and $j : A \to X$ is a function such that $f \circ j(a) = y$ for every $a \in A$. Also, for every set Z and function $q : Z \to X$ such that $f(q(t)) = y$ for every $t \in Z$, there is a unique function $\overline{q} : Z \to A$ such that $q = j \circ \overline{q}$.

Again it can be shown that inverse images are unique up to an isomorphism.

Definition 8.6 (Injection). An injection $j : A \to X$ is a function with the property that $j(a) = j(a') \implies a = a'$ for every $a, a' \in A$.

Definition 8.7 (Surjection). A surjection $s : X \to Y$ is a function such that for every $y \in Y$ there exists an $x \in X$ such that $s(x) = y$.

Definition 8.8 (Right inverse). The right inverse of a function $s : X \to Y$ is a function $i : Y \to X$ such that $s \circ i = 1_Y$.

Definition 8.9 (Subset Classifier). The tuple $(\mathbf{2}, t)$ where $\mathbf{2}$ is a set and $t \in \mathbf{2}$ is called a subset classifier if for all sets A, X and injections $j : A \to X$, there is a unique function $\chi : X \to \mathbf{2}$, such that (A, j) is an inverse image of t under χ.

Note that in the definition above, the function χ can be seen as a characteristic function. Suppose that we wish to define χ_A. Hence, it's required that there exists a set $\mathbf{2}$ with $t \in \mathbf{2}$ such that $\chi_A(j(a)) = t$ for every $a \in A$.

Definition 8.10 (Natural Number System). A natural number system is a triple $(N, 0, s)$ where N is a set, $0 \in N$ and $s : N \to N$, such that for any set X, $a \in X$ and $r : X \to X$, there is a unique function $x : N \to X$ where $x(0) = a$ and $x(s(n)) = r(x(n))$ for every $n \in N$.

This comes from the idea that $s(n) \cong n + 1$, that $x(0) \cong x_0$ and $x_n \cong x(s(n-1)) \cong r(x_n) \cong r(x(n))$. Once more, natural number systems are unique up to an isomorphism.

After all this definitions, we can finally state the axioms for Set Theory.

Definition 8.11 (ETCS). Lawvere's Elementary Theory of the Category of Sets consists on the following axioms:

(i) For all sets W, X, Y, Z, and functions $f : W \to X$, $g : X \to Y$, $h : Y \to Z$, we have

$$h \circ (g \circ f) = (h \circ g) \circ f.$$

For every set X and Y and function $f : X \to Y$, there exist the identity functions 1_X and 1_Y, such that

$$f \circ 1_X = f = 1_Y \circ f.$$

(ii) There exists a terminal set T.

(iii) There exists a set with no elements, i.e. an empty set denoted by $\varnothing$.

(iv) For sets X, Y and functions $f : X \to Y$ and $g : X \to Y$, if $f(x) = g(x) \forall x \in X$, then $f = g$.

(v) Every pair of sets has a Cartesian product.

(vi) For all sets X and Y, there is a function set from X to Y.

(vii) For every $f : X \to Y$ and $y \in Y$, there is an inverse image of y with respect to f.

(viii) There exists a subset classifier. This can be thought as saying that for every set we can construct a characteristic function.

(ix) There exists a natural number system.

(x) Every surjection has a right inverse.

As we pointed out, these axioms are actually weaker than ZFC, but with one extra axiom, it can be shown to be as strong as ZFC. The last axiom is the one related to the Axiom of Choice. The first axiom states that sets form a category, and the following axioms distinguish this category from others.

With these axioms stated, we can now define the notion of a subset, and clearly differentiate objects that are and that aren't actually sets. One might think that "anything we can reasonably conceive" must be a set. But this is not the case.

Definition 8.12 (Subset). Given a set X, a subset of X is a function $f : X \to \mathbf{2}$. The subset $\chi_A : X \to \mathbf{2}$ is written as $A \subset X$, where χ_A is the characteristic function with $\chi_A^{-1}(t) = A$.

Corollary 8.13. A set T is terminal if and only if it has only a single element.

Proof. $\implies$) If T is terminal, then for any set X, we have a unique $f : X \to T$. For $t_1, t_2 \in T$, then $t_1 : T' \to T$ and $t_2 : T' \to T$ where T' is a terminal set. Note that $f : T' \to T$ is unique, hence, $t_1 = t_2$, meaning that T has only a single element.

$\impliedby$) If T has a single element $t \in T$, then for a set X, take $f_1 : X \to T$ and $f_2 : X \to T$. Since T has only one element, then $f_1(x) = t = f_2(x)$, which, by Axiom 3, implies that $f_1 = f_2$. □

References

[1] Francis Borceux. *Handbook of categorical algebra: volume 1, Basic category theory*, volume 1. Cambridge University Press, 1994.

[2] Tai-Danae Bradley, Tyler Bryson, and John Terilla. *Topology: A Categorical Approach*. MIT Press, 2020.

[3] Brendan Fong and David I Spivak. *An invitation to applied category theory: seven sketches in compositionality*. Cambridge University Press, 2019.

[4] Tom Leinster. Rethinking set theory. *The American Mathematical Monthly*, 121(5): 403–415, 2014.

[5] Bartosz Milewski. *Category theory for programmers*. Blurb, 2018.

[6] Evan Patterson, Owen Lynch, and James Fairbanks. Categorical data structures for technical computing. *arXiv preprint arXiv:2106.04703*, 2021.

[7] Maico Ribeiro. *Teoria das Categorias para Matemáticos. Uma breve introdução*. 05 2020. ISBN 9786599039515.

[8] Steven Roman et al. *An Introduction to the Language of Category Theory*, volume 6. Springer, 2017.

[9] David I Spivak. *Category theory for the sciences*. MIT Press, 2014.