Category Theory Notes

Bryan Lu

September 2021 - January 2022

0 Foreword

These are my own cleaned up personal notes on category theory that I've been studying over the past semester (FA21) and over winter break. I've been keeping notes on paper so far, but I now have a large sheaf of paper to carry around and they're not very well-written, so I think this will help me solidify the concepts and remember things a little better when I have to put them in my own words. These are intended as a summary of the important concepts of category theory as presented mainly in Emily Riehl's *Category Theory in Context*, with maybe a little sprinkling of guidance from Paolo Aluffi's *Algebra: Chapter 0*, with an emphasis on examples to help comprehension along the way.

I've memed a lot about category theory to my suitemates and friends over the past semester or so, and they're probably sick of it by now, so thank you for putting up with my abstract nonsense :)) Seriously though, to my knowledge category theory is a really helpful framework through which we can gain a deeper understanding of structure of important set-objects we care about without necessarily having to muck through the details of what's actually in the set. Instead, we study the relationships between objects and other objects of that type, so to speak, and there are a lot of cool things we can say about this!

Some prerequisites – you really should have some abstract algebra and topology background to start. I think a lot of the examples I've found in Riehl are good, but they take me a while to unpack and I feel not having any of this background would be confusing or possibly fatal. There might be an exception here early on if you understand a little bit of functional programming, but after a certain point you need to have a certain level of mathematical sophistication to continue.

Diagrams that are not computer generated are drawn in Inkscape and hopefully look pretty clean. Also, practicing using the commutative diagrams package, which is really helpful :))

1 Chapter 1: Basics

1.1 What is a Category?

Definition 1 (Category, Object, Morphism)

A **category** consists of:

- a collection of **objects** (X, Y, Z, etc.)
- a collection of **morphisms** (f, g, h, etc.) where the morphisms have *specified* **domain** and **codomain** objects, i.e. $f : X \to Y$

such that

- each object has an **identity** morphism $\mathrm{id}_X : X \to X$
- **composition** of morphisms gives a new morphism, so that if $f : X \to Y$, and $g : Y \to Z$, then $g \circ f = gf : X \to Z$.
- the identity behaves as expected with respect to composition, i.e. for all $f : X \to Y$, that $f\mathrm{id}_X = \mathrm{id}_Y f = f$.
- the composition of morphisms is **associative**, so if we have $f : X \to Y$, $g : Y \to Z$, and $h : Z \to W$, then $(hg)f = h(gf) := hgf : X \to W$.

Example 2 (Concrete Categories)

List of examples of some common concrete categories, whose objects are backed by sets:

- **Set**, the category with sets as objects and regular functions as morphisms
- **Grp**, the category with groups as objects and group homomorphisms as morphisms
- **Ring**, the category with rings as objects and ring homomorphisms as morphisms
- **Top**, the category with topological spaces as objects and continuous maps as morphisms
- $\mathbf{Vect}_F$, the category with vector spaces over a field F as objects and linear transformations as morphisms
- $\mathbf{Set}_*$, the category with pointed sets (pairs of sets, and elements in that set) with functions that preserve the base point, i.e. a morphism $f : (A, x) \to (B, y)$ with $x \in A, y \in B$ must have $f(x) = y$.

Note that categories are often named after their objects.

Example 3 (Abstract Categories)

Some more abstract categories that will come up:

- For a ring (with unit) R, $\mathbf{Mat}_R$ is the category whose objects are the positive integers $\mathbb{Z}^+$ and the set of morphisms from m to n is the set of $n \times m$ matrices with entries in R. We use matrix multiplication for morphism composition here.
- A group G (or monoid, really) defines a category $\mathbf{B}G$ with one object, and one morphism for each element in the group G.
- A poset $(P, \leq)$ defines a category $\mathbf{P}$, where the objects of $\mathbf{P}$ are the elements in P and a unique

morphism $f : x \to y$ if $x \leq y$.

Almost immediately, you might see that we run into size issues in category theory, as there is no such thing as a set of all sets! (See **Bertrand's Paradox** for why.) We need something somewhat "bigger" than a set to describe a category, but we aren't going to be too concerned with this. It won't be lucrative to restrict ourselves to categories who have sets of objects, but we can do the following:

Definition 4 (Smallness)
We treat several different sizes of categories. A category $\mathbf{C}$ is:

- **small** if $\mathbf{C}$ only has a set's worth of morphisms.
- **locally small** if for any two objects $X, Y \in \mathbf{C}$, that the collection of morphisms $\mathbf{C}(X, Y)$ is a set.

Just to reiterate the definition for clarity:

Definition 5
In a category $\mathbf{C}$, the collection of morphisms between objects $X, Y \in \mathbf{C}$ is denoted $\mathbf{C}(X, Y)$ (other places might denote it $\mathrm{Hom}(X, Y)$). This is called a **hom-set** if $\mathbf{C}$ is locally small (otherwise, it might not be a set!).

Finally, some common terminology about morphisms:

Definition 6 (Morphism Types)
A morphism $f : X \to Y$ is:

- an **isomorphism** if there exists a morphism $g : Y \to X$ that acts as a "right and left inverse," i.e. $fg = \mathrm{id}_Y$, $gf = \mathrm{id}_X$. We say then that X and Y are **isomorphic**, so $X \cong Y$. (This is pretty similar to one characterization of bijections as "invertible functions" in regular set theory.)
- an **endomorphism** if Y and X are the same object, so f is actually a morphism $X \to X$
- an **automorphism** if it is an isomorphism and an endomorphism.

Often, the prefixes endo- and auto- will be applied in other contexts, where they mean roughly the same thing.

1.2 Opposite Day

Definition 7 (Opposite Category)
If $\mathbf{C}$ is a category, we define the **opposite category** $\mathbf{C}^{\mathrm{op}}$ as the category with the same objects as C, but for all morphisms $f : X \to Y$ in $\mathbf{C}$, $\mathbf{C}^{\mathrm{op}}$ has morphisms $f^{\mathrm{op}} : Y \to X$ in $\mathbf{C}^{\mathrm{op}}$.

To be explicit, the opposite category $\mathbf{C}^{\mathrm{op}}$ has the following properties:

- Every object $X \in \mathbf{C}$ has an identity morphism $\mathrm{id}_X{}^{\mathrm{op}}$.
- Composition works in the reverse order. We define composition of $f^{\mathrm{op}} : X \to Y$ with $g^{\mathrm{op}} : Y \to Z$ by taking $g^{\mathrm{op}} \circ f^{\mathrm{op}} = f \circ g^{\mathrm{op}}$, since f^{op} corresponds to $f : Y \to X$ in $\mathbf{C}$, and g^{op} corresponds to $g : Z \to Y$ in $\mathbf{C}$, so $f \circ g : Z \to X$ in $\mathbf{C}$ will correspond to the result of our composition in $\mathbf{C}^{\mathrm{op}}$.

Note that the operation of composition behaves roughly the same way in the opposite category as in the regular category, so we see that any theorem that holds for a regular category can be shown for an opposite category by "flipping all the arrows around," and vice versa. This will allow us to essentially get double the theorems and propositions for any statement we make, and this idea is called **duality**.

Here's an example of a lemma where appealing to duality cleans up half of the work for us:

Lemma 8

1.2.3 in Riehl?

Proof. hiho □

Here's a pair of dual definitions for morphisms that correspond loosely to our ideas of "injective" and "surjective" as set-functions.

Definition 9 (Monic, Epic)

A morphism $f : x \to y$ is:

- a **monomorphism** if for all $h, k : w \to x$ parallel morphisms (going between the same two objects), having $fh = fk$ implies $h = k$. f is then said to be **monic**.
- an **epimorphism** if for all $h, k : y \to z$ parallel morphisms (going between the same two objects), having $hf = kf$ implies $h = k$. f is then said to be **epic**.

These are fairly opaque definitions, so why do these correspond to the ideas of "injective" and "surjective"?

<insert explicit contrapositive idea here>

However, we CANNOT say that if a morphism $f : x \to y$ is monic and epic, then it is an isomorphism. For instance, consider the ring homomorphism by inclusion $i : \mathbb{Z} \to \mathbb{R}$. <fill in the details>

1.3 What is a Functor?

We have functions between sets, but what is a relation between categories? Introducing the **functor**:

Definition 10 (Functor)

Let $\mathbf{C}, \mathbf{D}$ be categories. A (covariant) **functor** $F : \mathbf{C} \to \mathbf{D}$ consists of:

- an object Fc for every $c \in \mathbf{C}$
- a morphism $Ff : Fc \to Fc'$ for every $f : c \to c'$ in $\mathbf{C}$

such that $F(\mathrm{id}_c) = \mathrm{id}_{Fc}$ for all $c \in \mathbf{C}$ and $Fg \circ Ff = F(g \circ f)$.

Again, we shower you with a lot of examples here, so that you get an idea:

Example 11 (Covariant Functors)

examples

1.4 What is a Natural Transformation?

Natural transformations are a little weird, because we don't really have a common set equivalent out in the world – the best way to describe it is as a "function" between two functors $F, G : \mathbf{C} \to \mathbf{D}$. Here's the definition first:

Definition 12

Let $\mathbf{C}, \mathbf{D}$ be categories and $F, G : \mathbf{C} \to \mathbf{D}$ be functors. A **natural transformation** $\alpha : F \Rightarrow G$ is a collection of arrows $\alpha_c : Fc \to Gc$ in $\mathbf{D}$, where this collection ranges over all $c \in \mathbf{C}$, such that for all $f : c \to c'$ in $\mathbf{C}$, we have that the following diagram **commutes**:

$$\begin{array}{ccc} Fc & \xrightarrow{\alpha_c} & Gc \\ {\scriptstyle Ff}\downarrow & & \downarrow{\scriptstyle Gf} \\ Fc' & \xrightarrow[\alpha_{c'}]{} & Gc' \end{array}$$

When we say a diagram **commutes**, it means we get the same resulting morphism by composition tracing along any path between two objects on the diagram.

1.5 Equivalence of Categories

1.6 (there might be more important stuff here that I skipped idk)

2 Chapter 2: Representability, Yoneda, Universality

2.1 Representable Functors

2.2 The Yoneda Lemma

2.3 Universal Properties

2.4 The Category of Elements