Introduction to Category Theory

Grant Talbert

September 27, 2025

College of Arts and Sciences
Department of Mathematics and Statistics

Abstract

We introduce the basic notions of category theory at an undergraduate level. We cover categories, functors, natural transformations, universals, and Yoneda.

Contents

1 Introduction

A group is (roughly) a set G, and an operation (usually addition or multiplication) on the elements of G. For example, the sets of real numbers, complex numbers, integers, and rational numbers are all groups under addition. Technically they are not groups under multiplication, but this is not very important for us.

There's another extremely important construction in the study of groups, called a *group homomorphism*. Given groups G, H where we denote both of the operations by multiplication, a *group homomorphism* is a function $\varphi : G \to H$ such that for any $g, h \in G$,

$$\varphi(g \cdot h) = \varphi(g) \cdot \varphi(h).$$

We say that φ *preserves* the group operation. The reason group homomorphisms are interesting is because the existence of a group homomorphism implies two groups have similar *structure*. We usually think of the group operation as having some additional *structure*, and the study of groups is the study of this structure. The group homomorphism φ takes the structure of G, and morphs it into something compatible with the structure of H, which is why the group operation of H is the same both inside and outside of H. This is why group homomorphisms are interesting.

The upshot of this short discussion is that the field of group theory is the study of *groups*, and THE most important tool in this is the study of *group homomorphisms*. We want to study a certain type of object (groups), and interesting types of maps between them (group homomorphisms).

This idea of having a type of *object* and a type of *map* between them is not just isolated to group theory. Many different fields of math follow this exact same recipe, such as the study of vector spaces, R-modules, rings, topological spaces, and way more.

Category theory abstracts this idea. Many fields of math are *abstractions* of more mundane constructions, which means instead of focusing on any specific construction (eg. Euclidean space $\mathbb{R}^n$), we focus on *any* construction which behaves similarly (eg. any topological space X). In category theory, instead of looking at a specific field of math, we define *categories*, which behave like these fields of math. This allows us to look at math from a higher level, and find fascinating connections between far-flung fields of math. Throughout these notes, we will see how we do this.

Category theory arose in the 1940s due to Mac Lane and Eilenberg, who founded the subject to give a precise meaning to the notion of naturality. In the 80 years it has existed, category theory has found its way into nearly every branch of math, and is foundational to understanding modern research in numerous fields. In these notes, we will introduce the basic notions of the theory.

2 Categories

The definition of a category is fairly lengthy, so the reader is encouraged to reference remarks 2.2, 2.3, and 2.4 while reading the definition for further commentary.

Definition 2.1. A *category* $\mathcal{C}$ is a mathematical object consisting of

- A collection $\mathrm{Obj}(\mathcal{C})$ of things which we call *objects*;
- A collection $\mathrm{Mor}(\mathcal{C})$ of things which we call *morphisms*;
- A function $\mathrm{dom} : \mathrm{Mor}(\mathcal{C}) \to \mathrm{Obj}(\mathcal{C})$ assigning each morphism $f \in \mathrm{Mor}(\mathcal{C})$ an object called its *domain* $\mathrm{dom}\, f \in \mathrm{Obj}(\mathcal{C})$;
- A function $\mathrm{cod} : \mathrm{Mor}(\mathcal{C}) \to \mathrm{Obj}(\mathcal{C})$ assigning each morphism $f \in \mathrm{Mor}(\mathcal{C})$ an object called its *codomain* $\mathrm{cod}\, f \in \mathrm{Obj}(\mathcal{C})$;

- A function $1_\bullet : \mathrm{Obj}(\mathcal{C}) \to \mathrm{Mor}(\mathcal{C})$ assigning each object $x \in \mathrm{Obj}(\mathcal{C})$ to a morphism $1_x \in \mathrm{Mor}(\mathcal{C})$ called the *identity on* x;
- If $f : x \to y$ and $g : y \to z$ are morphisms such that $\mathrm{dom}\, g = \mathrm{cod}\, f$, then we call (g, f) a *composable pair*. There is a function

$$\circ : \{(g, f) \mid (g, f) \text{ is a composable pair}\} \to \mathrm{Mor}(\mathcal{C})$$

  mapping each composable pair (g, f) to its *composite* $g \circ f : x \to z$.

This data has to satisfy the following axioms:

1. For each object x, the identity has $1_x : x \to x$;
2. Composition is associative, meaning for any composable triple (h, g, f), there is an equality $h \circ (g \circ f) = (h \circ g) \circ f$;
3. The identity is an identity for composition, meaning for any $f : x \to y$, there are equalities $f \circ 1_x = f$ and $1_y \circ f = f$.

Remark 2.2. The idea is that objects should represent mathematical objects such as groups, sets, rings, or basically whatever we would conventionally think of as an object. Morphisms are then the interesting ways to map between them, such as group homomorphisms. Note that we *do not require morphisms to be functions*, because many interesting categories are constructed where morphisms are not functions. Morphisms simply are things with a domain and a codomain which can be composed. In other words, they are simply arrows, and functions are simply a special case of this idea.

Remark 2.3. If $\mathcal{C}$ is a category and f is a morphism, we write $f : x \to y$ to denote that f is a morphism with $\mathrm{dom}\, f = x$ and $\mathrm{cod}\, f = y$. We think of morphisms as *arrows* from the objects x to y, and we think of objects simply as points with arrows going between them.

Remark 2.4. The reasoning for requiring an identity morphism to exist will become clear later, but the point is that identities are required to define *isomorphisms*, which are a fundamental notion in the study of categories.

We will write $x \in \mathcal{C}$ to denote that x is an *object* of $\mathcal{C}$. We will typically never write $f \in \mathrm{Mor}(\mathcal{C})$ when avoidable, and instead prefer to say either "Let f be a morphism in $\mathcal{C}$", or "Let $f : x \to y$". We will write $\mathcal{C}(x, y)$ for the set of all morphisms $x \to y$ in $\mathcal{C}$, although other authors often use $\mathrm{Hom}_{\mathcal{C}}(x, y)$.

We now present some examples.

Example 2.5. The canonical example of a category is the category of *sets*, denoted by $\mathcal{S}\mathrm{et}$. Objects are *sets*, and morphisms are *functions*. The category axioms are easily checked: any function $f : X \to Y$ obviously has a domain and a codomain, and function composition is well-known to be associative. Given a set X, the identity function 1_X is simply the function $x \mapsto x$ which "does nothing". The identity is easily seen to be an identity for composition, since

$$f(1_X(x)) = f(x), \qquad 1_Y(f(x)) = f(x).$$

Example 2.6. Another example, as stated earlier, is the category $\mathcal{G}\mathrm{rp}$ of *groups* and *group homomorphisms*. Some other high level examples are

- The category $\mathcal{T}\mathrm{op}$ of topological spaces and continuous maps;

- The category $\mathcal{CG}$ of compactly generated topological spaces and continuous maps;
- The category $\mathcal{R}\mathrm{ing}$ of (unital) rings and (unital) ring homomorphisms;
- The category k-$\mathcal{V}\mathrm{ect}$ of k-vector spaces and k-linear maps (if you haven't taken abstract algebra, ignore the k);
- The category $\mathcal{A}\mathrm{b}$ of abelian groups and group homomorphisms.

We shall now work through a more interesting example. In all the previous examples, our morphisms were *functions*, but we will now present a category where morphisms are not functions. Consider the set $\mathbb{Z}$ of integers. Define a category $\mathcal{P}$ such that objects are integers

$$\mathrm{Obj}(\mathcal{P}) = \mathbb{Z},$$

and morphisms are defined as follows: for any $n, m \in \mathbb{Z}$, if $n \leq m$, then there is *exactly one morphism* $n \to m$. If $n > m$, then there is *no morphism* from n to m. Since morphisms between integers are unique, we don't need to name them, since there is only one.

Since $n \leq n$, there is exactly one morphism $n \to n$, so we define the identity as this morphism. Moreover, there is only way to define composition: if $m \leq n \leq k$, then $m \leq k$, meaning that there is precisely one arrow $m \to k$. The composite of the morphisms $m \to n$ and $n \to k$ is then defined as this arrow $m \to k$.

Categories defined this way are called *poset categories*. Any set X endowed with a relation $\leq$ which is reflexive, transitive, and antisymmetric is called a *poset*, and can be turned into a category in the same way. Our example had the added benefit of being *well-ordered*, meaning either $n \leq m$ or $m \leq n$ for all n, m, but we do not need this fact in order to define a poset category. A nice example is give any set X, the power set $\mathcal{P}(X)$ together with the $\subseteq$ relation forms a poset, and hence a poset category.

There are many interesting constructions we can make inside of categories. We will explore some of them later. For now, we would like to answer a different question. In definition 2.1, we had only one bullet point discussing objects, and every other bullet point was discussing morphisms. This means that although we are usually most interested in studying the objects in a category, the *morphisms* are the data with the most information, so they are the real objects of interest. In other words, in order to study the objects of a category, it is most convenient to study the morphisms between them. The onus then is on us to define morphisms in such a way that they carry meaningful information about a category.

With this in mind, we turn towards category theory itself. The objects of study in category theory are *categories*, but the way to study objects in category theory is to study morphisms between them. Thus, it would be nice to have a notion of morphisms of *categories*.

3 Functors and representable functors

Definition 3.1. Let $\mathcal{C}, \mathcal{D}$ be categories. A *functor* F from $\mathcal{C}$ to $\mathcal{D}$, written $F : \mathcal{C} \to \mathcal{D}$, consists of two functions, both of which are also denoted F:

- An *object function* $F : \mathrm{Obj}(\mathcal{C}) \to \mathrm{Obj}(\mathcal{D})$;
- A *morphism function* $F : \mathrm{Mor}(\mathcal{C}) \to \mathrm{Mor}(\mathcal{D})$;

and these functions have to satisfy the following axioms.

1. F commutes with domain and codomain. More precisely, if $f : x \to y$ in $\mathcal{C}$, then $F(f) : F(x) \to F(y)$ in $\mathcal{D}$.

2. F preserves identity. More precisely, if 1_x is the identity on some $x \in \mathcal{C}$, then $F(1_x) = 1_{F(x)}$ is the identity on $F(x) \in \mathcal{D}$.

3. F preserves composition. More precisely, if $f : x \to y$ and $g : y \to z$ in $\mathcal{C}$, then $F(g \circ f) = F(g) \circ F(f)$ in $\mathcal{D}$.

Functors are a good notion of morphisms between categories because they preserve basically all the information in a category. Recall that in section 1, we said that because group homomorphisms preserved group structure, they were interesting to look at, because they made the structure of one group compatible with another. Functors are the same. They make the structure of one category compatible with another.

Some examples of functors are the *forgetful functors*. Let $U : \mathcal{G}\mathrm{rp} \to \mathcal{S}\mathrm{et}$ be the functor which maps a group $(G, \cdot)$ to the underlying set G, and which maps group homomorphisms to themselves. The functor U basically just "forgets" the group structure of a group, and views it as a set. It then views group homomorphisms simply as regular functions between regular sets. We call functors like this *forgetful*, and most categories where objects have underlying sets admit a forgetful functor to $\mathcal{S}\mathrm{et}$.

Example 3.2. A more important example of a functor is a representable functor. Let $\mathcal{C}$ be a category, and $c \in \mathcal{C}$. Recall that $\mathcal{C}(x, y)$ is the set of morphisms $x \to y$ in $\mathcal{C}$, so we can consider an object function $\mathcal{C}(c, -)$ which maps objects x to the hom-set $\mathcal{C}(c, x)$. This function takes objects x of $\mathcal{C}$, and maps them to *sets* $\mathcal{C}(c, x)$, so if we can define a compatible action on morphisms, then this will give us a functor $\mathcal{C}(c, -) : \mathcal{C} \to \mathcal{S}\mathrm{et}$.

To define action on morphisms, let $f : x \to y$. To prevent cumbersome notation, we will write f_* instead of $\mathcal{C}(c, f)$. We want f_* to be a morphism $\mathcal{C}(c, x) \to \mathcal{C}(c, y)$ in $\mathcal{S}\mathrm{et}$. Morphisms in $\mathcal{S}\mathrm{et}$ are just functions, so we can define f_* like a normal function. Let $\varphi : c \to x$ be an element of $\mathcal{C}(c, x)$. Since $f : x \to y$, we note that $f \circ \varphi$ is a morphism $c \to x \to y$, and thus an element $f \circ \varphi \in \mathcal{C}(c, y)$. We can thus define $f_*(\varphi) = f \circ \varphi$. We call this *postcomposition*. Indeed, this definition is functorial, although we leave it as an exercise to prove that this definition satisfies the functor axioms.

A functor $\mathcal{C}(c, -)$ is called a *representable functor* represented by c, and c is the representation for the functor. As we will see in section 5, representable functors have interesting properties.

The reader is at this point likely wondering if $\mathcal{C}(-, c)$ is a functor $\mathcal{C} \to \mathcal{S}\mathrm{et}$ as well. Indeed, we can define the object function $x \mapsto \mathcal{C}(x, c)$, but we run into an issue with the morphism function. Let $f : x \to y$ in $\mathcal{C}$, and let $\varphi \in \mathcal{C}(x, c)$. Note that it is *impossible* to compose f and φ, so we cannot define a functor the same way as in example 3.2. However, if we let $\varphi \in \mathcal{C}(y, c)$, then we note that we can compose $\varphi \circ f : x \to c$, giving us a function $\mathcal{C}(y, c) \to \mathcal{C}(x, c)$. This is called *precomposition*, and we write the precomposition function as $f^* : \mathcal{C}(y, c) \to \mathcal{C}(x, c)$. We would like for this to also be a functor, but here is the issue. If $f : x \to y$ in $\mathcal{C}$, then $f^* : \mathcal{C}(y, c) \to \mathcal{C}(x, c)$ in $\mathcal{S}\mathrm{et}$, meaning $\mathcal{C}(-, c)$ has *flipped the direction* of the morphism f. Definition 3.1 says that functors have to preserve the direction of f. Luckily, we can fix this with the notion of an *opposite category*.

Construction 3.3. Let $\mathcal{C}$ be a category, and define a new category $\mathcal{C}^{\mathrm{op}}$ as follows:

- $\mathrm{Obj}(\mathcal{C}) = \mathrm{Obj}(\mathcal{C}^{\mathrm{op}})$;

- For any $x, y \in \mathcal{C}$, we have $\mathcal{C}(x, y) = \mathcal{C}^{\mathrm{op}}(y, x)$. In other words, if $f : x \to y$ in $\mathcal{C}$, then $f : y \to x$ in $\mathcal{C}^{\mathrm{op}}$. We basically have flipped the domain and codomain of all morphisms;

- Identities are defined the same as in $\mathcal{C}$;
- If $f : x \to y$ and $g : y \to z$ in $\mathcal{C}^{\text{op}}$, then $f : y \to x$ and $g : z \to y$ in $\mathcal{C}$. Note that $f \circ g$ exists in $\mathcal{C}$, so we define $g \circ f$ in $\mathcal{C}^{\text{op}}$ as $f \circ g$ in $\mathcal{C}$.

The category $\mathcal{C}^{\text{op}}$ is called the *opposite category* of $\mathcal{C}$.

Since $\mathcal{C}^{\text{op}}$ flips the direction of every morphism, we can view $\mathcal{C}(-, c)$ as a functor $\mathcal{C}^{\text{op}} \to \mathcal{S}\text{et}$, since every $f : x \to y$ in $\mathcal{C}$ is a morphism $f : y \to x$ in $\mathcal{C}^{\text{op}}$, which points in the same direction as f^* in $\mathcal{S}\text{et}$. Functors whose domain is an opposite category are called *contravariant functors*. We usually just think of contravariant functors as functors which *flip* the direction of domain, codomain, and composition. Functors of the form $\mathcal{C}(-, c)$ are also called representable.

4 (Co)limits

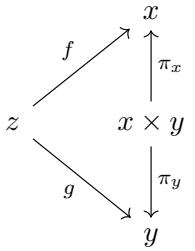
Oftentimes in category theory, asking whether two objects c, d are *equal* is not useful, and it is more useful to ask whether or not they *have the same categorical propeties*. In other words, two objects which behave the same way can be *thought of* as being exactly equal, since they have the exact same properties in their category. To rigorously define when two objects have the same properties, we introduce the notion of an *isomorphism*.

Definition 4.1. Let $\mathcal{C}$ be a category, and $f : x \to y$ in $\mathcal{C}$. We say the morphism f is an *isomorphism* if there is another morphism $g : y \to x$ such that $f \circ g = 1_y$ and $g \circ f = 1_x$. We call g the *inverse* of f, and usually write it as f^{-1}. We may also refer to isomorphisms as *invertible morphisms*. If there is an isomorphism between two objects x and y, then we call those objects *isomorphic*, and write $x \cong y$. We may also write $f : x \cong y$ to denote that the isomorphism is given by f.

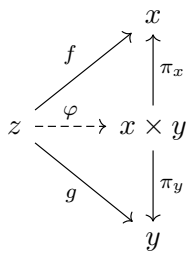
Let $f : x \to y$ be an isomorphism. Note that any morphism $\varphi : x \to z$ can be turned into a morphism $y \to z$ by precomposing by f^{-1}, and any morphism $\varphi : y \to z$ can be turned into a morphism $x \to z$ by precomposing by f. Since the data of a category is encoded in morphisms, two objects should have the same cateorical properties if their morphisms are the same. Thus, we might think that isomorphic objects have the same categorical properties, since we can switch morphisms around between them. We will see that indeed, isomorphic objects can easily be thought of as the same.

With the notion of isomorphism defined, we can now turn to likely the most powerful notion in all of category theory: a *universal property*. We begin with an informal (but sufficient) description, and later we introduce the formal definition.

It is most convenient to begin with an example. Let $x, y \in \mathcal{C}$ be objects of some category $\mathcal{C}$. A *product* of x and y, denoted $x \times y$, is an object $x \times y \in \mathcal{C}$ toghether with morphisms $\pi_x : x \times y \to x$ and $\pi_y : x \times y \to y$ called the *natural projections*, which satisfies the following propery: let $z \in \mathcal{C}$ be any object of $\mathcal{C}$, and $f : z \to x$ and $g : z \to y$ be any morphisms. We can write this information using a diagram



A product $x \times y$ has the property that for any diagram like this, there is a *unique* morphism $\varphi : z \to x \times y$ indicated by the dashed arrow



making the diagram *commute*.

A diagram *commutes* when any two paths between two objects are the same. In this diagram, there are two paths from z to x: we can take $f : z \to x$, or we can take $\pi_x \circ \varphi : z \to x \times y \to x$. For the diagram to *commute*, these must be the same, meaning

$$\pi_x \circ \varphi = f.$$

Similarly, we have

$$\pi_y \circ \varphi = g.$$

The reader is likely confused why we are calling this a *product*. To see why, let X, Y be sets, and consider the *cartesian product* $X \times Y$. Define the functions $\pi_X : X \times Y \to X$ and $\pi_Y : X \times Y \to Y$ by $(x, y) \mapsto x$ and $(x, y) \mapsto y$, respectively. I claim that $X \times Y$ is a product of X and Y, in the above sense.

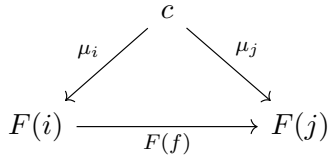
Indeed, let $f : Z \to X$ and $g : Z \to Y$ be functions. We want to define a function $\varphi : Z \to X \times Y$ such that $f(z) = (\pi_X \circ \varphi)(z)$ and $g(z) = (\pi_Y \circ \varphi)(z)$. Define $\varphi(z) = (x, y)$. We then have

$$f(z) = \pi_X(x, y) = x, \qquad g(z) = \pi_Y(x, y) = y.$$

It follows that $x = f(z)$ and $y = g(z)$, meaning we are forced to define $\varphi(z) = (f(z), g(z))$. This definition makes the diagram commute, and it was *forced*, meaning it is unique. Therefore, the cartesian product is indeed a product.

However, there's more. In every branch of math, there is a notion of a product. We can consider a *direct product* of groups, the *product topology* for two topologiacal spaces, a *direct product* of rings, a *cartesian product* of vector spaces, and so on. *All* of these products are products in the above sense. In other words, the notion of a product introduced earlier is a *universal* definition for a product in *any category*. This is the idea behind a universal property. To formalize what a universal property is, we introduce the notion of a *cone*.

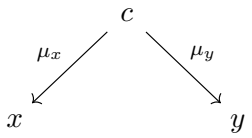
Definition 4.2. Let $\mathcal{J}$ be a (small) category, and let $F : \mathcal{J} \to \mathcal{C}$ be a functor. A *cone* for F is an object $c \in \mathcal{C}$, and a collection $\{\mu_j : c \to F(j)\}_{j \in \mathcal{J}}$ of morphisms of $\mathcal{C}$, such that for every $f : i \to j$ in $\mathcal{J}$, the diagram



commutes. The object c is called the vertex of the cone.

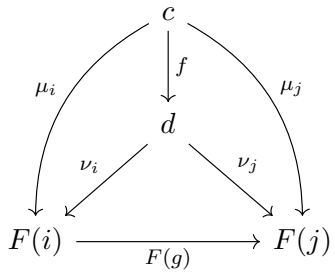
The reason we call this a cone is as follows. Instead of thinking of F as a functor, think of it as a *diagram*. Every object $j \in \mathcal{J}$ simply indexes the object $F(j)$ in $\mathcal{C}$, and every $f : i \to j$ in $\mathcal{J}$ simply indexes the morphism $F(f) : F(i) \to F(j)$ in $\mathcal{C}$. Thus, we can imagine visualizing the image of F as a massive diagram in $\mathcal{C}$ corresponding to the entire category of $\mathcal{J}$. Now imagine the object c sitting above the entire diagram, with morphisms μ_j coming down to every single object. This shape looks a little like a cone, hence we call it a cone.

For example, consider the category $\mathcal{J}$ with two objects $x, y \in \mathcal{J}$, and *no* non-identity morphisms in $\mathcal{J}$. A functor $F : \mathcal{J} \to \mathcal{C}$ is then just a choice of two objects $c_x, c_y \in \mathcal{C}$, and then a map $x \mapsto c_x$ and $y \mapsto c_y$. A *cone* for F is then an object $c \in \mathcal{C}$, and two morphisms $\mu_x : c \to c_x$ and $\mu_y : c \to c_y$:



Since there are no non-identity morphisms in $\mathcal{J}$, there is no diagram to make commute.

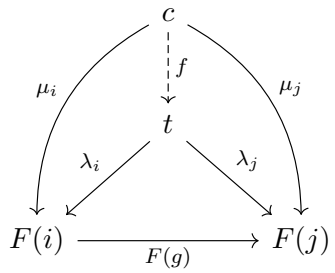
Definition 4.3. Let $F : \mathcal{J} \to \mathcal{C}$ be a functor, and let $(c, \mu), (d, \nu)$ be cones for F. A *morphism* $f : (c, \mu) \to (d, \nu)$ of cones is a morphism $f : c \to d$ in $\mathcal{C}$ such that for any $g : i \to j$ in $\mathcal{J}$, the diagram commutes:



Including the morphism g in the definition is not required, since that part of the diagram commutes for any cone, but we include it to keep the idea of a cone fresh in the reader's mind. This allows us to construct a category $\mathrm{Cone}(F)$ of cones for F. We will combine this category with the following two definitions:

Definition 4.4. An object i in a category $\mathcal{C}$ is *initial* if for all $x \in \mathcal{C}$, there is exactly one morphism $i \to x$. An object $t \in \mathcal{C}$ is *terminal* or *final* if for all $x \in \mathcal{C}$, there is exactly one morphism $x \to t$.

Terminal objects in the category $\mathrm{Cone}(F)$ are cones (t, λ) such that for any cone (c, μ), there is a *unique morphism* $f : c \to t$ making the diagram commute:



We again indicate unique morphisms by dashed arrows. Terminal objects in $\mathrm{Cone}(F)$ are called *limits*.

Definition 4.5. Let $F : \mathcal{J} \to \mathcal{C}$. A *limit* for F is a *terminal cone* (t, λ), also called an *limiting cone*. We usually abuse language and call the object t the limit, forgetting the morphisms λ. We write $\lim F$ for the object t in the limiting cone.

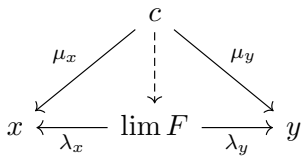
Note that a functor can have *more than one limit*, and limits for a functor may not exist at all. However, if they exist, then they are all *isomorphic* to each other, meaning they behave the exact same way. This is proven by the following proposition:

Proposition 4.6. *Let $\mathcal{C}$ be a category. All initial objects (if they exist) are isomorphic, and all final objects (if they exist) are isomorphic.*

Proof. We prove the statement for final objects. Let $s, t \in \mathcal{C}$ be final. Since t is final, there is exactly one morphism $s \to t$, and likewise there is only one morphism $t \to s$. If we compose these morphisms, we get an arrow $s \to s$ or an arrow $t \to t$. Since s, t are terminal, there is only one morphism $s \to s$ and $t \to t$, and this must be the identity by the category axioms. Thus, the arrows $s \to t$ and $t \to s$ are inverses. ∎

This shows that all objects in $\mathrm{Cone}(F)$ are isomorphic. However, since morphisms of cones are also morphisms in $\mathcal{C}$, we have that the vertices of cones are *also isomorphic in* $\mathcal{C}$. Thus, limits are all isomorphic as objects of the target category.

To see an example of a limit, consider the diagram $\mathcal{J}$ with two objects and no nontrivial morphisms again, and $F : \mathcal{J} \to \mathcal{C}$. A cone for that we said was simply some $c \in \mathcal{C}$ and morphisms $\mu_x : c \to x$ and $\mu_y : c \to y$. A *limit* for this diagram is then a cone $(\lim F, \lambda)$ such that given another cone (c, μ), there is a unique dashed arrow as indicated



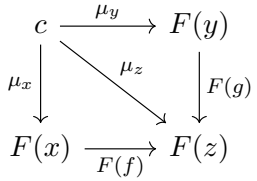
rendering the diagram commutative. This is *precisely* the same definition as a product $x \times y$ of x and y. In other words, *products are limits.*

We now see the power of limits. Recall that the definition we gave above for products was a "universal propety" in the sense that it provided a way to construct products in any category. We were able to formalize this notion by defining a category $\mathcal{J}$ which encoded the information of this universal property in such a way that products are simply limits of functors $\mathcal{J} \to \mathcal{C}$. In other words, the category $\mathcal{J}$ is a *universal* way to construct products in any category.

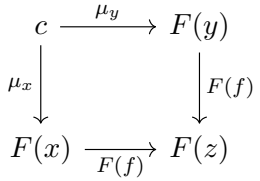
Another example of a limit is as follows. Consider the category whose only objects and nontrivial morphisms are indicated in the diagram

$$x \xrightarrow{f} z \xleftarrow{g} y.$$

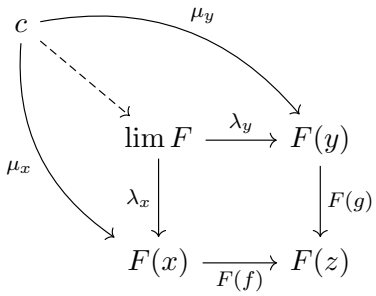
A cone for a functor $F : \mathcal{J} \to \mathcal{C}$ is an object c such that the diagram



commutes. We prefer not to write the morphism μ_z, since we can always figure out how it is defined using the fact that the diagram commutes: we can simply define it as $F(g) \circ \mu_y$ or $F(f) \circ \mu_x$. It is enough to ask that the square



commutes. A *limit* for F is then a universal cone:

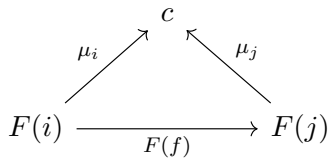


commutes. This is a special type of construction known as a *pullback*, and because we defined it as a limit, we now have a way to define it in *any category*, by simply defining the correct functor $\mathcal{J} \to \mathcal{C}$. The intersection of a family of sets can even be defined as a pullback.

Because limits can be thought of as universal constructions, we say they are *universal*, or that they satisfy *universal properties*. For example, might say that a product $x \times y$ is *universal* with respect to mapping to x and y. This is decoded as: construct a functor such that a cone contains a map to x and y and no other data, and a product is then a limit for that functor. A pullback could also be described as "universal with respect to commutative squares".

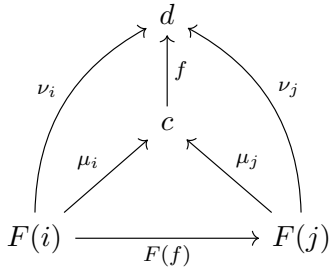
Before we close the section, there is a dual notion to limits. If we *flip the direction* of the morphisms μ in a cone (c, μ), we obtain a *cocone*.

Definition 4.7. Let $F : \mathcal{J} \to \mathcal{C}$ be a functor. A *cocone* for F is an object $c \in \mathcal{C}$, and a collection of morphisms $\{\mu_j : F(j) \to c\}_{j \in \mathcal{J}}$, such that for any $f : i \to j$ in $\mathcal{J}$, the diagram



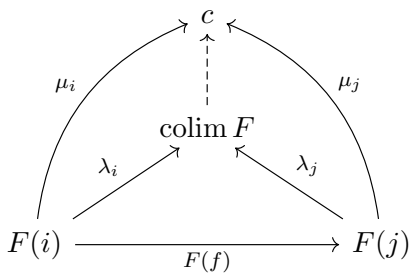
commutes.

A morphism $f : (c, \mu) \to (d, \nu)$ of cocones is then given by flipping the arrows for a morphism of cones:

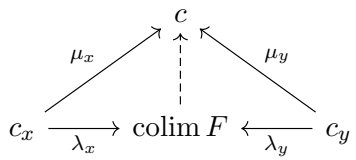


We then form a category $\mathrm{Cone}(F)$ of cocones for F. This is the same as the notation for the category of cones, so usually we specify whether we are considering cones or cocones. However, we may also simply refer to cocones as cones, which makes it slightly confusing. Usually, it will be clear from contet.

A *colimit* for the functor F, denoted $\operatorname{colim} F$, is an *initial* cocone. Diagrammatically, this looks like



For example, if $\mathcal{J}$ is the category with two objects and no nontrivial morphisms, a colimit for $F : \mathcal{J} \to \mathcal{C}$ is a diagram

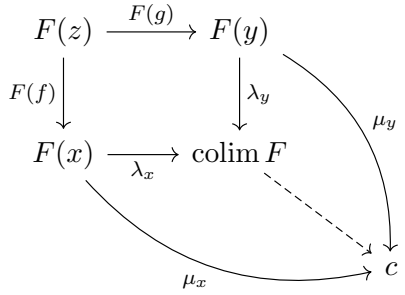


We call this a *coproduct*, and denote it by $c_x \amalg c_y$. In $\mathcal{S}\mathrm{et}$, this is the *disjoint union*.

If $\mathcal{J}$ is the category whose only objects and nontrivial morphisms are given by the diagram

$$x \xleftarrow{f} z \xrightarrow{g} y,$$

then a colimit for $F : \mathcal{J} \to \mathcal{C}$ is a diagram



We have recalled that we may drop the arrows $F(z) \to \operatorname{colim} F, c$ by the same logic as with pullbacks. We call these objects *pushouts*. A union of a family of sets is an example of a pushout. Similarly to limits, colimits are unique up to isomorphism, and are said to satisfy *universal properties*.

5 Natural transformations and Yoneda

We now come to the final section of these notes. We have defined categories, functors, and the most powerful way of constructing important objects in categories: universal properties. We now would like to define an extremely important type of category: a functor category.

Definition 5.1. Let $F, G : \mathcal{C} \to \mathcal{D}$ be functors. A *natural transformation* $\alpha : F \Rightarrow G$ is a collection of morphisms such that for each $x \in \mathcal{C}$, there is a morphism $\alpha_x : F(x) \to G(x)$ in $\mathcal{D}$, such that for any morphism $f : x \to y$ in $\mathcal{C}$, the diagram

$$\begin{array}{ccc} F(x) & \xrightarrow{\alpha_x} & G(x) \\ {\scriptstyle F(f)}\downarrow & & \downarrow{\scriptstyle G(f)} \\ F(y) & \xrightarrow[\alpha_y]{} & G(y) \end{array}$$

commutes. We call squares of this form *naturality squares*, and we call the commutativity condition the *naturality condition*.

Of course, natural transformations can be composed. If $\alpha : F \Rightarrow G$ and $\beta : G \Rightarrow H$, then for any $f : x \to y$ there is a diagram

$$\begin{array}{ccccc} F(x) & \xrightarrow{\alpha_x} & G(x) & \xrightarrow{\beta_x} & H(x) \\ {\scriptstyle F(f)}\downarrow & & {\scriptstyle G(f)}\downarrow & & \downarrow{\scriptstyle H(f)} \\ F(y) & \xrightarrow[\alpha_y]{} & G(y) & \xrightarrow[\beta_y]{} & H(y) \end{array}$$

Of course, if we ignore the arrow $G(x) \to G(y)$, we get that the morphisms $\beta_x \circ \alpha_x$ form a natural transformation $\beta \circ \alpha : F \Rightarrow H$. Thus, we obtain a *category* $\mathrm{Fun}(\mathcal{C}, \mathcal{D})$, where objects are functors $\mathcal{C} \to \mathcal{D}$, and morphisms are natural transformations. This category is also denoted $\mathcal{D}^{\mathcal{C}}$. We also obtain a notion of a *natural isomorphism*, which is an invertible natural transformation.

If F, G are functors such that there are isomorphisms $F(x) \cong G(x)$ for each object x, we say this isomorphism is *natural in* x if the isomorphisms assemble into a natural isomorphism $F \cong G$.

Recall that isomorphic objects are basically the same. Thus, isomorphic functors are basically the same. Amongst other things, this allows us to extend our definition of *representable* functors.

Definition 5.2. A functor $F : \mathcal{C} \to \mathcal{S}\mathrm{et}$ is *representable* if there is an object $c \in \mathcal{C}$ and a natural isomorphism $F \cong \mathcal{C}(c, -)$. Similarly, a contravariant functor $F : \mathcal{C}^{\mathrm{op}} \to \mathcal{S}\mathrm{et}$ is representable if there is an isomorphism $F \cong \mathcal{C}(-, c)$.

An interesting fact is that any functor $F : \mathcal{C} \to \mathcal{S}\mathrm{et}$ is a colimit of representable functors. This fact is more useful than it may sound. We break from our usual notation for hom-sets and write $\mathrm{Nat}(F, G)$ to denote the set of natural transformations from F to G.

Now that we have defined natural transformations, we can discuss a result of fundamental importance in category theory.

Lemma 5.3 (Yoneda)**.** *Let $F : \mathcal{C} \to \mathcal{S}\mathrm{et}$ be a functor. Then for any $c \in \mathcal{C}$, there is an isomorphism*

$$\mathrm{Nat}(\mathcal{C}(c, -), F) \cong F(c)$$

natural in c. Similarly, if $G : \mathcal{C}^{\mathrm{op}} \to \mathcal{S}\mathrm{et}$ is a contravariant functor, then there is an isomorphism

$$\mathrm{Nat}(\mathcal{C}(-, c), G) \cong G(c)$$

natural in c.

Proof. We prove only the isomorphism for the first case, and leave the second case and naturality as an exercise. We claim the isomorphism is given by $\alpha \mapsto \alpha_c(1_c)$. This is given by letting $f : c \to d$ in $\mathcal{C}$, and chasing the identity around the naturality square

$$\begin{array}{ccc} \mathcal{C}(c,c) & \xrightarrow{\alpha_c} & F(c) \\ {\scriptstyle f_*}\downarrow & & \downarrow{\scriptstyle F(f)} \\ \mathcal{C}(c,d) & \xrightarrow[\alpha_d]{} & F(d) \end{array}$$

■

One result of importance here is the Yoneda embedding. Consider the case where $G = \mathcal{C}(-,d) : \mathcal{C}^{\text{op}} \to \mathcal{S}\text{et}$. Plugging this in gives isomorphisms

$$\text{Nat}(\mathcal{C}(-,c), \mathcal{C}(-,d)) \cong \mathcal{C}(c,d).$$

In particular, each natural transformation $\mathcal{C}(-,c) \Rightarrow \mathcal{C}(-,d)$ is simply given by having each component be *postcomposition* by some $f : c \to d$ in $\mathcal{C}$. This gives us a *functor* $y : \mathcal{C}^{\text{op}} \to \text{Fun}(\mathcal{C}, \mathcal{S}\text{et})$ by mapping $c \mapsto \mathcal{C}(c,-)$, and mapping morphisms to postcomposition $f \mapsto f_*$. This functor is called the *Yoneda embedding*, and it is of importance because it is *isomorphic* on hom-sets. We call such a functor full and faithful.

The Yoneda embedding is very important in category theory, and shows up everywhere. For example, if I end up giving a second talk, we will use it to show that n-simplices of simplicial sets K are equivalent to morphisms $\Delta^n \to K$, to find a cosimplicial object $\Delta^\bullet$ in s$\mathcal{S}$et, and to define faces, horns, and boundaries of the standard n-simplices.

6 Further reading

These notes are admittedly terse due to the time constraint on the talk they acompany, and although the reader has now been exposed to many important concepts in category theory and has seen how categorical arguments, they may struggle to apply these concepts if all they have seen are these notes. If the reader is interested, we encourage them to turn to a more complete resource for continuing their study of the theory.

Unfortunately, category theory is generally only taught at the graduate level. Although it is definitely understandable with only basic knowledge of functions and sets, it is difficult to motivate, so nearly all authors assume knowledge with algebra, topology, or another higher level undergraduate subject. Of course, it is not necessary to understand all examples an author gives, so categorical texts are definitely approachable with only some knowledge of the above subjects, but if the reader is fully unfamiliar with these fields, we caution the reader not to explore further down the rabbit hole for now.

The main text we recommend is Emily Riehl's *Category Theory in Context*, a modern presentation of the basic theory. We also recommend Tom Leinster's *Basic Category Theory*, a more approachable presentation, albeit covering less material. We also note that the canonical reference on category theory is Saunders Mac Lane's *Categories for the Working Mathematician*, but we discourage learning from here because Mac Lane has a very dense writing style. This author is speaking from experience here. This book is also not available freely online, unlike the previous two, but it is quite a solid referenceso we acknowledge it here.