INTRODUCTION TO CATEGORY THEORY AND THE YONEDA LEMMA

ALEX STERN

Abstract. This expository paper serves as an introduction to the foundational aspects of Category Theory, with an emphasis on simple examples. After building intuition for fundamental categorical concepts, we will aim to elucidate the Yoneda Lemma, one of Category Theory's most important results. We finally cover universal properties to give an additional perspective into how Category Theory can be used to derive results in many contexts.

Contents

1. Basic Categorical Definitions

In simple terms, Category Theory exists as a means of comparing various mathematical topics and drawing conclusions that can be interpreted appropriately in each area. To create these analogies, we build off of two concepts: "things," and the transformations between them. For example, sets can be considered as things, and we can use functions to get from one set to another. On the other hand, we have vector spaces which can be modified into other vector spaces through linear transformations. Category Theory seeks to formalize this concept into something that applies to multiple topics. With that we can say formally:

Definition 1.1. A **category** consists of two forms of data:

- **Objects**: X, Y, Z, $\ldots$
- **Morphisms**: f, g, h, $\ldots$

These objects and morphisms have the following properties:

- Each morphism has a **domain** and a **codomain**, which are both objects. The notation $f\colon X \to Y$ indicates that the objects X and Y are the domain and codomain of the morphism f, respectively. We can also write $\mathrm{dom}(f) = X$ and $\mathrm{cod}(f) = Y$ to convey the same information.
- Each object has an **identity morphism**, denoted $1_X : X \to X$

Date: October 9, 2022.

- For any arrows f and g such that $\mathrm{cod}(f) = \mathrm{dom}(g)$, there exists a composite morphism, denoted $g \circ f$. This morphism's domain is that of f and its codomain is that of g:

$$f : X \to Y, \quad g : Y \to Z \rightsquigarrow g \circ f : X \to Z$$

Finally, these data must follow two laws:

- First, for all $f : X \to Y$:

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

- Secondly, there is associativity. For all $f : X \to Y, g : Y \to Z, h : Z \to W$:

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

With this definition established, we can begin to look at some common examples of categories.

Example 1.2. The following are common examples of categories:

- The collection of all sets forms a category, called **Set**. Sets are the objects and functions between sets are the morphisms.
- Similarly, posets form a category, with the distinction from **Set** being that all the morphisms are monotonic functions, as they preserve the ordering of each set.
- We could consider the collection of all groups, with morphisms being group homomorphisms. This category is called **Group**.
- A group can also be considered a category with only one object. This object has various morphisms going to itself, each one corresponding to an element of the group.
- We can even consider less abstract structures as categories, such as a programming language. The various types of the programming language (int, string, etc.) form the objects of the category, and then the morphisms between objects are functions with the appropriate input and output types.

With this, our understanding of what a category is becomes much clearer, but there are still many other features of categories as well as examples to consider. This paper will focus on only the most critical features to understanding Yoneda's Lemma, so we will introduce two more definitions before moving beyond the basics.

Definition 1.3. An **opposite category** or **dual** of a category $\mathcal{C}$, denoted $\mathcal{C}^{\mathbf{op}}$ is one in which the arrows are reversed. Formally, $\mathcal{C}^{\mathbf{op}}$ has the same objects as $\mathcal{C}$, and every morphism $f\colon X \to Y$ in $\mathcal{C}$ is $f\colon Y \to X$ in $\mathcal{C}^{\mathbf{op}}$.

Although the concept of "dual" is not particularly important to us now, it is simple enough to introduce at this point. Dual categories are actually quite central to category theory, as any statement made about a given category can be interpreted to create a "dual statement" by reversing the morphisms in the category. We will not go over the duals of the lemmas and theorems in this paper, as it can create additional layers of confusion and obfuscate the core premise of each statement. That said, duality can be an interesting topic to explore when one is more comfortable with categorical thinking. Now, we have one last definition before moving past the basics:

Definition 1.4. An **isomorphism** in a category $\mathcal{C}$ is a morphism $f : X \to Y$ such that there exists an inverse morphism $g : Y \to X$ with the property $f \circ g = 1_Y$ and $g \circ f = 1_X$.

We will not go into too many examples of isomorphisms, but one simple example is in the category **Set**. The isomorphisms in Set are bijective functions, which can be observed through composition, which results in the identity function, which is the identity morphism in **Set**.
With these definitions and a few examples under our belt, one might begin to ponder various other mathematical objects and how they might be considered as a category. Further, one might ask if one can consider a category where the objects themselves are also categories. In fact, this category, denoted **Cat**, is crucial in many categorical constructions. One natural question that arises however is how to interpret the morphisms between categories in **Cat**. This brings us to a discussion of functors.

Definition 1.5. A **functor** $F : \mathcal{C} \to \mathcal{D}$ between categories $\mathcal{C}$ and $\mathcal{D}$ is a mapping which associates objects and morphisms in $\mathcal{C}$ with objects and morphisms in $\mathcal{D}$. A functor must satisfy the following properties:

- For any objects X and Y in $\mathcal{C}$, $F(f : X \to Y) = F(f)$: $F(X) \to F(Y)$.
- For any morphisms f and g in $\mathcal{C}$, $F(g \circ f) = F(g) \circ F(f)$
- $F(1_X) = 1_{F(X)}$

$$\begin{tikzcd}
X \arrow[r, mapsto] \arrow[d, "f"'] & F(X) \arrow[d, "F(f)"] \\
\arrow[r, phantom, "\longmapsto"] & {} \\
Y \arrow[r, mapsto] & F(Y)
\end{tikzcd}$$

One of the most common examples of functors is the **forgetful functor**. This type of functor essentially "forgets" some of the structure of the input category. For example, a forgetful functor U: **Group** $\to$ **Set** would map each group to its underlying set. In doing so, the functor loses the group structure, leaving the category of sets. As we begin introducing the concepts of the Yoneda Lemma, the structure of functors will become more clear.

2. Hom Functors and Representations

A reoccurring theme thus far is that sets provide a useful analogy for many other mathematical objects. However, it is not always the case that the objects and morphisms of a given category can be easily turned into sets and functions.

Definition 2.1. A category is **locally small** if the morphisms between any objects in the category form a set. For two objects A in B in a locally small category, the set of morphisms between them, or the hom set, is denoted **Hom**(A,B).

While this definition is rather simple, Hom sets allow us to demonstrate how an object in a given category can be represented by the morphisms from that object. We can also use hom sets to classify functors further.

Definition 2.2. A functor $F : \mathcal{C} \to \mathcal{D}$ is

- **full** when $F_{X,Y} : \mathbf{Hom}_{\mathcal{C}}(X, Y) \to \mathbf{Hom}_{\mathcal{D}}(F(X), F(Y))$ is surjective for any objects X and $Y \in \mathcal{C}$.

- **faithful** when $F_{X,Y}$ is injective for any objects X and $Y \in \mathcal{C}$.

We can think of a faithful functor as sending each distinct morphism in the domain category to exactly one morphism in the target category. If the functor is full then the objects in the target category have no additional morphisms that do not correspond to morphisms in the domain category.

To understand these properties in context, we can consider the forgetful functor from earlier, acting on the category **Group**. This functor is faithful, but not full. It is faithful because each morphism in **Group** is a group homomorphism, each of which corresponds to a unique underlying function in **Set**, so $U_{X,Y} : \mathbf{Hom}_{\mathbf{Group}}(X,Y) \to \mathbf{Hom}_{\mathbf{Set}}(U(X),U(Y))$ is injective. However, not every function in **Set** corresponds to a group homomorphism, as certain group structures must be preserved.

We can now introduce one of the central ideas of the Yoneda lemma: using hom sets to construct a functor. Consider an arbitrary category $\mathcal{C}$ and choose an object A in that category. We can construct a Hom functor, $H_A : \mathcal{C} \to \mathbf{Set}$ such that H_A(X) = **Hom**(A,X). In other words, the functor sends each object in $\mathcal{C}$ to the morphisms from A to that object.

This establishes where objects are sent, but a functor must also have a defined mapping for morphisms. Since our functor has a codomain of **Set**, the morphisms of $\mathcal{C}$ will be mapped to functions. Suppose we have two objects X and Y, with a morphism between them, *f*. We also still have the set of morphisms between A and X, **Hom**(A,X). Recall that according to our definition of a category, if we have a morphism from A to X, and another from X to Y, there must exist a morphism from A to Y that is the composition of those two morphisms. This means that by composing each element of **Hom**(A,X) with *f*, we get an element of **Hom**(A,Y) of the form $f \circ -$. Thus, our functor sends each morphism to a function that corresponds to composing morphisms in our original category $\mathcal{C}$. What we have done is essentially construct a representation of $\mathcal{C}$ in **Set** by defining a hom functor.

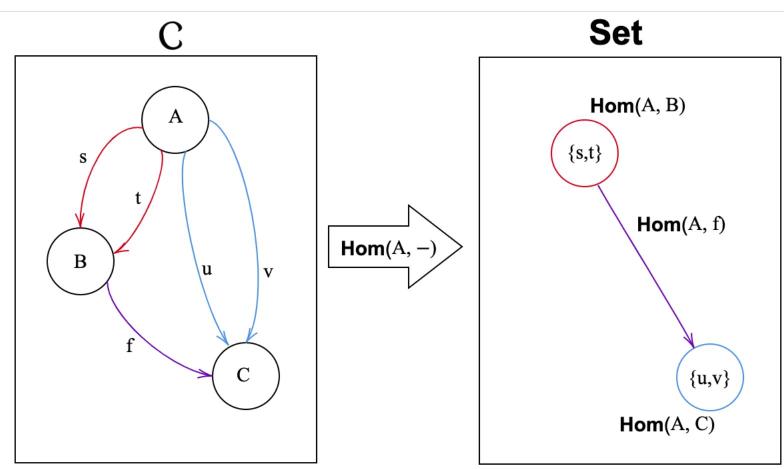

FIGURE 1. This diagram gives a visual representation of how the hom functor behaves. Each set of morphisms originating at A and going to a different object in $\mathcal{C}$ is sent to a different object in **Set**. Any morphism f that does not originate at A is sent to a function which composes f with the input morphism.

One of the central ideas of this discussion thus far has been finding easier ways to interpret categories and their structure. So far we have shown a simpler way to understand a category by using the hom functor to turn it into the category of sets. We can go much further with this idea. Now let us try to imagine how we can understand an abstract functor by analyzing its relationship with a functor we understand, the hom functor. To do this, we will need to introduce more terminology.

Definition 2.3. A **natural transformation** is a morphism between functors. More formally for two functors F and G from a category $\mathcal{C}$ to $\mathcal{D}$, a natural transformation η: F $\to$ G satisfies the following properties:

- It assigns each object X in $\mathcal{C}$ to an arrow in $\mathcal{D}$, η_X : F(X) $\to$ G(X).
- The assignments above must be such that for any f : X $\to$ Y in $\mathcal{C}$ we have $\eta_Y \circ F(f) = G(f) \circ \eta_X$ This condition is often depicted through the following diagram:

$$\begin{array}{ccc} X & & \\ \downarrow f & \rightsquigarrow & \\ Y & & \end{array} \qquad \begin{tikzcd} F(X) \arrow[r, "\eta_X"] \arrow[d, "F(f)"] & G(X) \arrow[d, "G(f)"] \\ F(Y) \arrow[r, "\eta_Y"] & G(Y) \end{tikzcd}$$

We can further say that if for each object $X \in \mathcal{C}, \eta_X$ is an isomorphism in the category $\mathcal{D}$, then F and G are **naturally isomorphic**.

Natural transformations form a sort of new dimension, and we could imagine going even further to define some sort of morphism between natural transformations, but that level of abstraction will be unnecessary for our purposes.
While we have toyed with the idea of "representing" things through sets and other means, there is a rigorous definition we will introduce that formalizes what a representation is.

Definition 2.4. A **representable functor** is a functor $F : \mathcal{C} \to$ **Set** that is naturally isomorphic to **Hom**(A,-) for some object A $\in \mathcal{C}$. To construct a representation, one chooses an object A and a natural transformation η: **Hom**(A,-) $\to F$.

As with most things in category theory, many examples exist throughout different mathematical disciplines, so let us use a group theoretical lens:

Example 2.5. The forgetful functor U : **Group** $\to$ **Set** is representable by the group $\mathbb{Z}$.

Proof. Since we have already chosen the object $\mathbb{Z} \in$ **Group** as part of our representation, all we must do is construct a natural transformation η and show that it is an isomorphism. Consider a group G with element g. For each element g, there is a unique group homomorphism from $\mathbb{Z}$ to G that maps 1 to the element g. In other words, each group homomorphism's behavior is determined by where it sends the generator of the group of integers, 1. This means that there are exactly as many elements of G as there are group homomorphisms between $\mathbb{Z}$ and G.
Thus, our natural transformation η sends each group G to the bijective function from the underlying set of G, $U(G)$, to the set **Hom**($\mathbb{Z}$, G) which we described above. Now we must show this satisfies the naturality condition by considering another group H with a morphism $f : G \to H$ such that $\eta_H \circ U(f) = \mathbf{Hom}(\mathbb{Z}, f) \circ \eta_G$. The left side sends an element g to some element h in a group H, and then h is

sent to a bijection as described above. The right side sends g to its bijection, and then composes it with the f, which gives the same result of a function that sends 1 to h. $\square$

This example illuminates how representations can help us understand a functor. In the previous example, we were able to represent the abstract idea of the forgetful functor and understand its relationship to the set of integers, a far more familiar setting. In more complex categories, representable functors allow us to make sense of mathematical structures we otherwise could not make sense of just using sets.
Representable functors also provide a nice transition into the central lemma of this paper, and category theory in general. The question we now ask is: If we have a set-valued functor $F : \mathcal{C} \to \mathbf{Set}$, what information is required to determine if it is representable by a given object $c \in \mathcal{C}$? Further, how can we construct that representation?

3. The Yoneda Lemma

The Yoneda lemma is a crucial result of category theory and it elucidates, among other things, a broader way of thinking about mathematics. Although the lemma's results are far-reaching, the reason we are interested in it for now is that it will answer our question of when functors are representable and how we might represent them.

Theorem 3.1. *(The Yoneda Lemma) For a locally small category $\mathcal{C}$ and a functor F: $\mathcal{C} \to$ **Set**, the set of natural transformations from **Hom**(A,-) to F, denoted **Nat(Hom**(A,-), F) is isomorphic to F(A). Formally:*

$$\boldsymbol{Nat}(\boldsymbol{Hom}(A,-),F) \cong F(A).$$

Another way to phrase this is that there are exactly as many natural transformations between $\mathbf{Hom}(A,-)$ and F as there are elements in the set $F(A)$. We will also write $\mathbf{Hom}(A,-)$ as h^A for simplicity.

Proof. Let us begin by constructing an arbitrary natural transformation $\eta : h^A \to F$. As per the definition of a natural transformation, each object X in $\mathcal{C}$ has a corresponding morphism, η_X in $\mathbf{Set}$. We then have the following commutative diagram:

$$\begin{tikzcd}
h^A(A) \arrow[r, "\eta_A"] \arrow[d, "h^A(f)"] & F(A) \arrow[d, "F(f)"] \\
h^A(X) \arrow[r, "\eta_X"] & F(X)
\end{tikzcd}$$

Note that f is general, as this diagram must commute for any of the morphisms between A and X. The most important concept in this proof is defining the action of η on the identity morphism of A, 1_A. For the diagram to commute we have:

$$\eta_X \circ h^A(f)(1_A) = F(f) \circ \eta_A(1_A)$$

Recall that h^A sends morphisms to a sort of "composition" function, $h^A(f) : \alpha \mapsto f \circ \alpha$. Thus:

$$\begin{aligned} h^A(f)(1_A) &= f, \\ \eta_X(f) &= F(f)(\eta_A 1_A). \end{aligned}$$

What this equation essentially tells us is that the behavior of η for a given object X is entirely determined by the behavior of η_A on the identity morphism of A, 1_A. We can think of this as showing that $\eta \mapsto \eta_A 1_A$ is an injection. This is not entirely sufficient, as we have yet to show that this function is a bijection.
What we must show is that for each object in the set $F(A)$, there is a natural transformation defined by sending 1_A to that object. In other words, that $\eta_X : f \mapsto F(fa)$ must be a natural transformation for any $a \in F(A)$. To show that this is true, we can consider an object $Y \in \mathcal{C}$, a morphism $g : X \to Y$, and a morphism $\alpha \in h^A(B)$. From the same equation above we have:

$$\eta_Y h^A g(\alpha) = \eta_Y(g\alpha) = F(g\alpha)a$$

Because F is a functor, we have:

$$F(g\alpha)a = F(g)F(\alpha)a$$

Since $\eta_X(g) = Fga$ for any object X in $\mathcal{C}$:

$$F(g)F(\alpha)a = F(g)\eta_X(\alpha).$$

What we have shown is that for any element $\alpha \in h^A(X), \eta_Y h^A g(\alpha) = F(g)\eta_X(\alpha)$. That is equivalent to saying that the following diagram commutes:

$$\begin{tikzcd}
h^A(X) \arrow[r, "\eta_X"] \arrow[d, "h^A(g)"] & F(X) \arrow[d, "F(g)"] \\
h^A(Y) \arrow[r, "\eta_Y"] & F(Y)
\end{tikzcd}$$

In other words, η as defined is indeed a natural transformation, so the bijection between $\mathbf{Nat}(\mathbf{Hom}(A, -), F)$ and $F(A)$ defined by $\eta \mapsto \eta_A 1_A$ holds. □

And with this we have proven one of the most crucial lemmas in Category Theory. One of the most crucial things we can get from the lemma is not just specific isomorphisms and relations, but a better philosophical understanding of category theory. One of the most common analogies is the saying "Tell me who your friends are, and I will tell you who you are." While this might seem trite it actually sums up the Yoneda perspective quite well. In a mathematical sense, the identity of mathematical objects is determined entirely by how they are related to other objects. We can continue to delve into this philosophy later on, but let us consider the quintessential example, the Yoneda Embedding.

Example 3.2. The Yoneda Embedding, y, is a functor $y : \mathcal{C}^{\mathbf{op}} \to \mathbf{Set}^{\mathcal{C}}$ which sends an object $X \in \mathcal{C}$ to its corresponding hom functor, $\mathbf{Hom}(X,\text{-})$. This functor also sends each morphism $f : Y \to X$ (notice the direction) to the natural transformation $\mathbf{Hom}(f,\text{-})$.

Before going into too much detail, let's break down some of the unfamiliar notation. The domain of y is the dual category of $\mathcal{C}$, which we defined earlier. The codomain is a category of functors that have have domain **Set** and codomain $\mathcal{C}$. Now that the setting is clear, we can make a powerful statement about this functor.

Theorem 3.3. *Yoneda's Embedding is full and faithful.*

Proof. Due to the way that it was defined, the embedding is injective. This is because each morphism $f \in \mathcal{C}^{\mathbf{op}}$ is sent to a natural transformation $\mathbf{Hom}(f,\text{-})$ uniquely defined by f. Thus $yf = yg \Rightarrow f = g$. Thus, the embedding is faithful (which is required to be called an embedding). Fullness can be proven using the lemma:

$$\mathbf{Nat}(\mathbf{Hom}(A,-),F) \cong F(A)$$

In this case F is also a hom functor, this time on the object B:

$$\mathbf{Nat}(\mathbf{Hom}(A,-),\mathbf{Hom}(B,-)) \cong \mathbf{Hom}(B,A)$$

Using the notation of the embedding we have:

$$\mathbf{Hom}(yA,yB) \cong \mathbf{Hom}(B,A)$$

Thus, since the two sets are isomorphic, and $y_{A,B}$ is an injection, $y_{A,B}$ must also be a surjection. Thus, y is full. □

It might not be immediately obvious why such a statement is remarkable. Yoneda's Embedding is valuable because it gives us a way of viewing a category $\mathcal{C}$ as a subcategory of functors from $\mathcal{C}$ to **Set**. This fact has many further fascinating implications for category theory, but let us move to more application based examples.
Yoneda's Lemma is sometimes referred to as a generalization of Cayley's Theorem, an important theorem in group theory. Using the lemma, we are able to prove Cayley's Theorem with very little effort and demonstrate the overarching nature of the lemma.

Theorem 3.4. *(Cayley's Theorem) Every group G is isomorphic to a subgroup of a symmetric group.*

Proof. To begin, we go back to the example of considering a group as a category with only one object. Let us consider such a category C_G with its sole object being G. We can then do some simple rearrangements using the statement in the Yoneda Lemma:

$$\mathbf{Nat}(\mathbf{Hom}(G,-),F) \cong F(G).$$

We can then choose our functor F to be $\mathbf{Hom}(G,-)$:

$$\mathbf{Nat}(\mathbf{Hom}(G,-),\mathbf{Hom}(G,-)) \cong \mathbf{Hom}(G,G).$$

But this statement shows exactly what we want, as we just need to interpret each side. On the left side, we have maps from G onto itself. On the right set, we have each element of G, because the group G is the same thing as $\mathbf{Hom}(G,G)$. Thus we have shown that the group of permutations of G is isomorphic to G itself. In other words, any group is isomorphic to a permutation group. □

This example should give a good idea of how powerful Yoneda's Lemma and category theory in general can be when applied directly to particular categories. Let us move on to one last application, this time with the category **Mat**.

Example 3.5. For a ring R, the category $\mathbf{Mat}_R$ has natural numbers as objects, and matrices with entries in R as morphisms. For two objects n and m, $\mathbf{Hom}(n,m)$ is the set of all $m \times n$ matrices. Note the order, as a morphism $A : n \to m$ is an $m \times n$ matrix. This is to preserve composition, as for two matrices A and B that are $m \times n$ and $k \times m$ respectively, we can compose morphisms A and B by matrix multiplication, $B \circ A = B \cdot A$.

We can now analyze the hom functor in this setting. For a chosen object k, $\mathbf{Hom}(k, -)$ sends an object n to the set of matrices with n rows and columns.

Now that we better understand the context of our category, we can make a broad statement about matrices.

Theorem 3.6. *Every row operation on a $n \times k$ matrix is determined by a single $n \times n$ matrix.*

Proof. Again we begin by writing out the isomorphism directly.

$$\mathbf{Nat}(\mathbf{Hom}(n, -), F) \cong F(n).$$

This time our functor F is also a hom functor that sends numbers to $k \times -$ matrices.

$$\mathbf{Nat}(\mathbf{Hom}(n, -), \mathbf{Hom}(k, -)) \cong \mathbf{Hom}(n, n).$$

Now we just interpret each side. On the left we have the set of natural transformations between the functors of $n \times -$ matrices to the functor of $k \times -$ matrices. These are the row operations we are looking for, as these correspond to ways to transform $n \times -$ matrices into $k \times -$ matrices. On the right we just have the set of $n \times n$ matrices. Thus we have shown the set of row operations on $n \times k$ matrices is isomorphic to the set of $n \times n$ matrices.

□

These two seemingly unrelated examples demonstrate how with proper categorical foundations, we derive a myriad of results from different contexts. This is the beauty of Yoneda's Lemma, and by extension category theory.

4. Universal Properties

Universal properties are another powerful tool in generalizing our results from categories into other contexts. While they are quite useful, they can be hard to comprehend, even with examples. We will provide a definition, and then go straight into the most common example.

Definition 4.1. In a locally small category $\mathcal{C}$, an object A has a **universal property** if there is a functor F represented by the object A. In addition, there is a **universal element** $X \in F(A)$ that defines a natural isomorphism between F and $\mathbf{Hom}(X, -)$ or $\mathbf{Hom}(-, X)$ depending on context.

Example 4.2. The concept of a product can be understood using universal properties. Products are defined in a variety of categories, but the simplest is the Cartesian product in the category **Set**. Recall that for two sets X and Y, the Cartesian product $X \times Y$ is the set of ordered pairs (x, y) such that $x \in X$ and $y \in Y$. There are also two projection functions $\pi_X : X \times Y \to X$ and $\pi_Y : X \times Y \to Y$:

$$\pi_X : (x, y) \mapsto x \qquad \pi_Y : (x, y) \mapsto y.$$

Now lets introduce one more set, Z, with functions $f : Z \to X$ and $g : Z \to Y$. We can easily construct a new function $h : Z \to X \times Y$, such that $h(z) = (f(z), g(z))$. We can also easily show that $\pi_X \circ h = f$ and $\pi_Y \circ h = g$:

$$\pi_X \circ h(z) = \pi_X \circ (f(z), g(z)) = f(z)$$

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

What this is actually telling us is rather simple. In the context of the Cartesian Product, if we have two functions which each go to separate sets, we could combine them to construct a function that goes to the product of the sets, and then use a projection function to get the same result. Knowing this, we can begin to attempt a categorical approach.
We want to demonstrate that $X \times Y$ has a universal property given by some functor F. It turns out that the functor behaves in the following way:

$$F : Z \mapsto \{c : Z \to X, d : Z \to Y\}$$

Equivalently, F maps Z to the set of ordered pairs of functions from Z to X and from Z to Y. What we essentially want to show is that there is a bijection between the set of ordered pairs of this kind and functions from Z to $X \times Y$. To show that F is represented by $X \times Y$ (and therefore that $X \times Y$ has a universal property), we need to show that there exists a natural isomorphism $\phi : \mathbf{Hom}(-, X \times Y)$. The Hom functor we need to represent F with in this case is $\mathbf{Hom}(-, X \times Y)$, as we are interested in functions going to the product. So, using Yoneda, we have:

$$\mathbf{Nat}(\mathbf{Hom}(Z, X \times Y), F) \cong F(Z)$$

This means that our natural isomorphism ϕ is in correspondence with a pair of functions from Z to X and Z to Y. In a more intuitive sense, functions from one set to a product of two sets can be broken down into tuples of functions.

While the result may seem rather simple, we can use the logic here to define products in a variety of categories using universal properties.

Definition 4.3. In a category $\mathcal{C}$, a product of two objects X and Y is an object P (or $X \times Y$) with morphisms $\pi_X : P \to X$ and $\pi_Y : P \to Y$ which satisfies the following universal property: for every object $Z \in \mathcal{C}$ every pair of morphisms $f : Z \to X$ and $g : Z \to Y$, there exists a unique morphism $h : Z \to P$ such that $\pi_X \circ h = f$ and $\pi_Y \circ h = g$. This is equivalent to showing that the following diagram commutes:

$$\begin{tikzcd}
 & & X \\
Z \arrow[r, dashed, "\exists! h"] \arrow[urr, bend left, "f"] \arrow[drr, bend right, "g"'] & P \arrow[ur, "\pi_X"] \arrow[dr, "\pi_Y"] & \\
 & & Y
\end{tikzcd}$$

While this definition may seem similar to the example of the Cartesian product, it is actually far more powerful. We will conclude by considering a new type of category and applying this definition to get a notion of the product in that category.

Example 4.4. We can consider a partially ordered set to be a category. The elements of the set are the objects of the category and there is at most one morphism between two objects. In particular, if $x \leq y$ then there is one morphism $(x, y) : x \to y$. In other words there is a morphism from x to y if and only if x is less than or equal to y in the ordering. Note that this still preserves identity morphisms necessary for a category due to the reflexive property of posets.
We now have everything we need to determine what a product is in a category of this form. Suppose we have elements a and b in our poset and we want to form the product $a \times b$. From our definition we have two morphisms $\pi_a : a \times b \to a$ and

$\pi_b : a \times b \to b$. Let us then assume that there is some object c such that $c \leq a$ and $c \leq b$. If there is no such element in the set, then the product is not defined.
If there is an object c satisfying the above properties, it would have two more morphisms $(c, a) : c \to a$ and $(c, b) : c \to b$. The categorical definition of the product tells us that there exists a unique morphism $(c, a \times b) : c \to a \times b$.

$$\begin{tikzcd}
 & & a \\
c \arrow[urr, bend left, "{(c,a)}"] \arrow[drr, bend right, "{(c,b)}"'] \arrow[r, dashed, "{(c,a\times b)}"] & a \times b \arrow[ur, "\pi_a"] \arrow[dr, "\pi_b"] & \\
 & & b
\end{tikzcd}$$

We can now begin to unravel what this all means. For an object c to have a morphism to both a and b, it must be less than or equal both of them. In addition, the product definition tells us that there is a morphism $(c, a \times b)$, meaning that $c \leq a \times b$. Further, the existence of π_a and π_b show that $a \times b \leq a$ and $a \times b \leq b$. Summarizing everything we know, the product $a \times b$ is an element of the poset which is less or equal both a and b, but is greater than or equal to any other element c in the poset which is less than or equal to both a and b.
Perceptive readers might recognize this to be the greatest lower bound. As it turns out, the product in a partially ordered set category is precisely the greatest lower bound of the two elements. Colloquially, the product of two elements a and b in a poset is the greatest element in the set which is less than or equal to both a and b.

This example demonstrates how a universal property can be used to generalize results to a variety of different categories. What at first may have seemed to be a rather simple result involving sets turned out to be a property applicable to many situations, with posets being an interesting example.

Acknowledgments

It is my pleasure to thank my mentor, Livia Xu for providing guidance in writing my paper and for introducing me to Category Theory. Without Livia's help, I would have been unable to understand many of the more complex examples included in the material I studied. I would also like to thank Daniil Rudenko for organizing a wonderful series of lectures for the apprentice program. Finally, I would like to thank Peter May for organizing the REU and providing this opportunity.

References

[1] Awodey, Steve. *Category Theory.* Oxford University Press, 2011.
[2] Milewski, Bartosz. "Understanding Yoneda." *Bartosz Milewski's Programming Cafe*, 16 Apr. 2018, https://bartoszmilewski.com/2013/05/15/understanding-yoneda/.
[3] Riehl, Emily. *Category Theory in Context.* Dover Publications Inc., 2016.
[4] Spivak, David I. "Notes on Applied Category Theory: Information Structures and Modular Systems."