YONEDA LEMMA, AN INTRODUCTION TO CATEGORY THEORY

EMRE CANIKLIOĞLU

Abstract. In this paper, we provide an introduction to *category theory* and a brief look at *Yoneda Lemma*. We begin by establishing the axioms and preliminaries. Then we move on to defining key concepts in *category theory*, including *categories*, *functors*, and *natural transformations*. Afterwards some additional structures that build on top previous sections will be introduced and a proof of the *Yoneda Lemma* with a section dedicated to its applications will be provided.

Introduction

Category theory is a branch of mathematics that deals with the study of structures and the relationships between them. It was developed in the 1940s and 1950s by mathematicians including *Samuel Eilenberg* and *Saunders Mac Lane* as a way to provide a more formal framework for understanding *algebraic topology*. Since its inception, *category theory* had a major impact on mathematics and has found applications in many areas. One of its important result is named *Yoneda Lemma*, which was developed by a Japanese mathematician *Nobuo Yoneda* in the 1950s. It is a powerful tool for understanding the structure of *functors* and *natural transformations*.

In this paper, we will introduce the reader to the introductory concepts in *category theory* and navigate through a path aimed at *Yoneda Lemma*. We will start by introducing the core structures and gradually build on top of each other to reach our goal. To give a summary of the topic, *category theory* deals with the structure of mathematical objects and the relationships between them. It is a powerful tool for understanding and organizing mathematical concepts, and has numerous applications in a wide range of fields, including *algebraic topology* and *algebraic geometry*. At its core, *category theory* is concerned with the study of *categories*, which are abstract collections of *objects* and *morphisms*. *Categories* can be thought of as models of different mathematical structures such as *groups*, *rings*, or *topological spaces*, and the *morphisms* between them can be thought of as the structure preserving maps between these structures.

As we come to the end of the introduction I would like to take a moment and show my appreciation to *Saunders Mac Lane* for his book *Category Theory for the Working Mathematician* and *NLab Authors* for the wonderful resources they have published on their website. These resources have been a great help to me and I appreciate it. Second of all, I would like to thank *Prof. Gheondea* mainly for his course on *Analysis* which achived to open my eyes and shared a wonderfull glimps in to the world of mathematics. Last, of all I would like to express my heartfelt gratitude to my mother, who has always believed in me and supported me in all of my endeavors. She has been a constant source of love, guidance, and encouragement throughout my life, and I am so grateful to have such a caring and supportive mother. I am so grateful for all that she has done for me, and I dedicate this paper to her as a small token of appreciation for everything she has given me.

1. Preliminaries

In this section we provide the reader with the preliminaries that are required to start our journey through category theory and provide a logical foundation which we will use to build on top of. We assume that the standard *Zermelo-Fraenkel* axioms for the set theory and the existence of the universe. Denoted by U. Hence we start by assuming that the followings are true.

1. *Extensionality*: Sets with the same elements are equal.
2. *Null Set*: There exist a set with no element and it is denoted by $\emptyset$
3. *Existence*: For all sets u, v, x the sets {u, v}, (u, v), $\mathcal{P}$(u) and $\bigcup$ x exists where we write { } as a set, () as an ordered pair, $\mathcal{P}$ as a power set of a set, and $\bigcup$ as a *union*.
4. *Infinity*: *Axiom of infinity holds*
5. *Choice*: *Axiom of choice holds*
6. *Regularity*: Every non empty set A contains an element B which is disjoint from A
7. *Replacement*: the image of a *set* under a *function* is a *set*

and we define the universe U to be subjected to the followings:

1. $[\ \mathrm{x} \in \mathrm{y} \wedge \mathrm{y} \in U\ ] \implies \mathrm{x} \in U$
2. $[\ I \in U \wedge \forall \mathrm{i} \in I\ x_i \in U\ ] \implies \bigcup_{i \in I} x_i \in U$
3. $[\ \mathrm{x} \in U\ ] \implies \mathcal{P}(\mathrm{x}) \in U$
4. $[\ \mathrm{x} \in U \wedge f\colon \mathrm{x} \to \mathrm{y}$ is a surjective function $] \implies \mathrm{y} \in U$
5. $\mathcal{N} \in U$

and last of all we define small sets to be elements of the universe U. [YL]

Now that we have declared the foundation which we are working with. To create an intuition, we will introduce the reader to some *meta structures*. We will add increasing amounts of constraints until we define what it means to be a category. With that purpose in our mind we start our journey by defining *metagraph*.

Definition 1.1. A *metagraph* consists of *objects* a, b, c... arrows f, g, h, ... , and two operations as follows:

1. *Domain*, which assigns to each arrow f an *object* a = *dom*(f)
2. *Codomain*, which assigns to each arow f an *object* b = *cod*(f)

One can think of the *metagraph* as just a collection of objects together with the collection of relations in between those object.

Example 1.2. As an example *metagraph* consider the objects a, b and c and arrows f, g and h where $f\colon\ a \to b$. $g\colon\ b \to c$ and $h\colon\ c \to a$ then following diagram would represent this metagraph

$$\begin{tikzcd}
 & b \arrow[dr, "g"] & \\
a \arrow[ur, "f"] & & c \arrow[ll, "h"']
\end{tikzcd}$$

By its own a *metagraph* gives us a way to put objects and their relations in a structured way. But in order to create a more coherent structure we define the *meta category*.

Definition 1.3. A *meta category* is a *metagraph* with two additional operations defined on top of it:

1. *Identity* which assigns to each object a an arrow $id_a : a \to a$.
2. *Composition* which assigns to each pair of (g, f) of arrows with $dom(g) = cod(f)$ an arrow $g \circ f$ called their composite with $g \circ f\colon\ dom(f) \to cod(g)$i

Where both the *identity* and *composition* operations are subjected to following constraints:

1. *Associativity*: Let a, b, c and d be *objects* and let $f\colon\ a \to b$, $g\colon\ b \to c$, $h\colon\ c \to d$ following equality always holds $h \circ (g \circ\ f) = (h\ \circ g) \circ\ f$
2. *Unit Law*: Let a and b be arbitrary *objects* than for all arrows $f\colon\ a \to b$ and $g\colon$ $b \to c$ *composition* with the *identity* arrow 1_b yields $1_b \circ f = f$ and $g \circ 1_b = g$

Example 1.4. Let a, b, c be objects and f and g be arrows such that $f\colon\ a \to b$ and $g\colon\ b \to c$ then following diagram represents the *metacategory* defined by these objects and arrows

$$\begin{tikzcd}
 & b \arrow[dr, "g"] & \\
a \arrow[ur, "f"] & & c \arrow[ll, "g \circ f"']
\end{tikzcd}$$

Adding *identity* and *composition* on top of the *meta graph* results in a better navigable structure. Where we can combine relations between objects and investigate complex relations between them. Up until now, we defined the structures without specifying the underlying collections. This was due to the complications brought by *Russel's Paradox*. We avoid this famous paradox by using the universe U defined in the beginning together with *small sets*. But from now on to create a more rigorous foundation we will continue by defining these structures interpreted in *set theory*. From now on when we talk about a *set*, we will be talking about a *small set*. Now that this is out of the way we continue with the definition of a *directed graph* and a *product over O*.

Definition 1.5. A directed graph is a set O of *objects* and a set A of *arrows* with two functions:

1 *Dom*: $A \to O$ which gives the domain of the mappings
2 *Cod*: $A \to O$ which gives the codomain of the mapping

Definition 1.6. Using the above notation we define the *product over O* to be the set

$$\mathrm{A} \times A_O = \{\ (g, f) — g, f \in A \text{ and } dom(g) = cod(f)\ \}$$

With the definition of *directed graph* and *product over O* we can now introduce the concept of a *category* together with the definitions of *identity* and *composition* within a more rigorous frame.

Definition 1.7. A *category* is a *graph* with two additional functions

1 *Identity* $O \to A$
2 *Composition* $A \times A_O \to A$

such that $dom(id_a) = \mathrm{a} = cod(id_a)$, $dom(\mathrm{g} \circ \mathrm{f}) = dom(\mathrm{f})$ and $cod(\mathrm{g} \circ \mathrm{f}) = cod(\mathrm{g})$ and for all objects and pairs of arrows associativity and unit laws holds.

Categories are used to formalize the concept of maps between objects in a way that is independent of the specific nature of the objects and maps involved. This abstraction enables us to reason about and compares different kinds of structures in a very general way. These maps or arrows between objects are often called morphisms between objects and from now on we will use the term morphisms rather than the arrow. Objects and morphisms of a category can be anything as long as they satisfy the above conditions. Because of the generalized constraints, it is easily observable that almost all if not most of the structures in mathematics can be used to construct categories. This enables us to investigate the relations between mathematical structures and the properties of these structures in a generalized manner. We will now provide the reader with some examples of categories, and some necessary small definitions and continue with the second section where the core structures defined on top of categories will be introduced.

Example 1.8. **O** is the empty category with no objects and no *morphisms* and **1** is the category with one object and only the *identity morphisms*.

Example 1.9. **Set** is the category whose objects are all *small sets* and *morphisms* are *functions* in between.

Example 1.10. **Grp** is the category whose objects are all *small groups* and its *morphisms* are all the *group morphisms*.

Example 1.11. **Rng** is the category whose objects are all *small rings* and *morphisms* are all the *ring morphisms*.

Example 1.12. **Top** is the category whose objects are *small topological spaces* and *morphisms* are all the *continuous maps*.

Another important definition which is necessary before we move on is the *hom-set* of two objects.

Definition 1.13. Let C be a category and let a, b $\in C$ we define the *hom-set* denoted by Hom(a, b) to be

$$Hom(\mathrm{a}, \mathrm{b}) = \{\, f \text{ s.t } f \in C,\ dom(f) = a,\ cod(f) = \mathrm{b}\}$$

Now that we have defined what it means to be a category and the hom set we can continue and introduce the concepts to classify different categories. There are three different classifications. A category can be *locally small*, *small* or *large*.

Definition 1.14. A category is *locally Small* if all of its *hom-sets* are *small sets*

Definition 1.15. A category is *small* if all of its *hom-sets* are *small sets* and the set that holds its objects is a *small set*.

Definition 1.16. A category is *large* if it is neither *small* or *locally small*.

2. Core Structures

Now that we provided the reader with enough intuition to know to understand what it means to be a *category*. We will continue this section by defining important structures and tools on top *categories*. We start this by introducing the concept of a *functor*.

Definition 2.1. Let A and B be two categories. A *functor* T: $A \to B$ with domain A and codomain B is a relations that consists of two functions:

1 *Object Function* T, which assigns to each object a $\in A$ an object $Ta \in B$
2 *Arrow Function* which is again denoted with T, which assigns to each arrow f: a $\to$ a' of A to an arrow Tf: $Ta \to Ta'$ of B

such that

$$T(1_a) = 1_{Ta}$$

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

A functor can be thought of as a structure-preserving relation between two categories. It respects both composition and identity operation. Given a *functor* $F\colon C \to D$ where both C and D are categories. Its object function creates a relation between the objects of C and D and $\forall$ c, d $\in C$, c', d' $\in D$. Its arrow function creates a link between, Hom(c, d) and Hom(c', d'). Hence it enables us to move in between categories and analyze their properties of them.

Example 2.2. Let C and D be two category for d $\in D$ define a *functor* $F\colon C \to D$ such that $\forall$ c $\in C$ Fc = d. F is a constant functor which takes each object of C to a constant object d in D

Example 2.3. Let $F\colon \mathbf{Set} \to \mathbf{Set}$ such that $\forall$ s $\in \mathbf{Set}$ Fs = $\mathcal{P}$(s) and for each morphism f: x $\to$ y to $F(f)$: F(x) $\to$ F(y). F is called the power set functor it maps sets to their power sets.

With the establishment of *functor*. We continue by quickly giving the definitions of *faithful functor* and *full functor* which are special functors satisfying certain constraints.

Definition 2.4. Let C and D be two categories. A *functor* $F\colon C \to D$ is called *faithful*. If for each pair of object c_1, $c_2 \in C$ its function $F_{x,y}$: $Hom(c_1, c_2) \to Hom(F(c_1), F(c_2))$ and $(x \to y) \to (F(x) \to F(y))$ is *injective*.

Example 2.5. The inclusion *functor* $F\colon \mathbf{Ab} \to \mathbf{Grp}$ is an example of a *faithful functor*.

Definition 2.6. Let C and D be two categories and consider the *functor* $F\colon C \to D$. F is called full if $\forall$ c_1, $c_2 \in C$ the function f: $Hom(c_1, c_2) \to Hom(F(c_1), F(c_2))$ between the *hom-sets* is *surjective*

Up until now we constructed *categories* and defined *functors* as relations on top of them. We now move a step higher and introduce the concept of a *natural transformation*. Just like *functors natural transformation* are relations between two objects. But the difference is that rather than creating relations in between *categories* they allow us to create relations in between given two *functors*.

Definition 2.7. Let A and B be two categories and let T and S be two functors such that T, S: $A \to B$. A *natural transformation* τ: $T \to S$ is a function which assigns to each object a $\in A$ an arrow $\tau_a = \tau$a: Ta $\to S$a of B in such way that every arrow f: a $\to$ a' in$A$ yields the following commutative diagram

$$\begin{array}{ccc} Sa & \xrightarrow{\tau_c} & Ta \\ {\scriptstyle Sf}\downarrow & & \downarrow{\scriptstyle Tf} \\ Sa' & \xrightarrow{\tau_{c'}} & Ta' \end{array}$$

A *natural transformation* is a way to create a mapping between two *functors*. It provides a way to connect two *functors* by mapping each object in one *functor* to an object in the other *functor* in a way that is consistent with the *morphisms* in the *category*. This gives us a way to compare and contrast the behavior of two functors, and to understand how they relate to each other. A *natural transformations* is often called a *morphisms* of *functors* and we denote the set of all *natural transformations* between two *functors* as $Nat(F,\ T)$ where both F and T are *functors*.

Example 2.8. Consider the he *category Grp* which is the *category* of all *small groups* and consider the *identity functor* id_{Grp}: *Grp* $\to$ *Grp* and the opposite functor *Op*: *Grp* $\to$ *Grp* such that for an object $(G,\ *)$ $Op(G,\ *) = (G^{op},\ *^{op})$ and consider the *natural transformation* τ: $id_{Grp} \to Op$ between these two such that $\tau(\mathrm{a}) = a^{-1}$ *functors*. Then we observe that τ is a *natural isomorphism* between these two functors. Which is a concept which will be introduced near the end of the third section. But briefly it shows that each group is isomorphic to its opposite group.

3. Advanced Structures

With the introduction of core concepts in *category theory*, we put a hold on our introduction through *category theory* and move forwards with an aim to build towards*Yoneda Lemma*. Hence in this section we will continue by introducing the necessary structures to understand *Yoneda Lemma*. These new structures are going to be special cases of *functors* and *natural transformations*. Afterwards we will continue the paper with the last section before the final remarks and provide the reader with the definition and proof of the *Yoneda Lemma* itself. With this in mind we first start by giving the definition of an *opposite category*.

Definition 3.1. Let C be a category we construct the *opposite categories* of C by reversing all of its *morphisms*.

One can think of the opposite category as a mirror image of the original category, where all the arrows go in the opposite direction. For example, if C is a *category* of *sets* and *functions*, then the *opposite category* C^{op} will also be a *category* of *sets* and *functions*, but the only difference in between them would be that the direction of all the *morphisms* in the C^{op} would be reversed. This is often used in *category theory* as a way to flip the direction of *morphisms* and study the behavior of a *category* from a different perspective. It can also be useful for constructing new *categories* from existing ones, or for studying *duality* and *symmetry.*

Example 3.2. Let C be a *category* of groups, then the *opposite category* C^{op} will also be a *category* of groups, but the direction of all the groups homomorphisms will be reversed. This can be used for studying the behavior of *group homomorphisms* in the opposite direction.

In the previous section we have provided the definitions of *functors* and *natural transformations.* One could have observed the resemblance of *natural transformations* as morphisms in between *functors.* Following this statement we define *functor categories* to be the following.

Definition 3.3. Let C and D be two categories, the *functor category* formed by all the possible *functors* between these two *categories* as its objects and the *natural transformations* between these *functors* as its *morphisms* is denoted by $[C, D]$ or D^C

Intuitively, one can think of a *functor category* as a way to package up *functors* and *natural transformations* between two *categories* in a single, coherent structure. This enables us to study and compare different *functors* and *natural transformations* systematically and to understand how they relate to each other. This newly constructed special type of *category* provides a way to understand how to structure and properties can be transferred between different *categories.*

Example 3.4. Consider *Set* and D be the category of *abelian groups* where the objects *abelian groups* and the *morphims* are *group homomorphisms*, then an object of the *functor category* $[\mathbf{Set}, D]$ is a functor that assigns to each set in *Set* an *abelian group* in D

A *presheaf* is a generalization of the concept of a function defined on an open cover of a topological space. It allows us to assign an object in a *category* to each open set of a topological space in a way that is compatible with the inclusions of the open sets. The motivation for using *presheaves* comes from the fact that they provide a way to capture the local behavior of a system in a global way.

Definition 3.5. A *Presheaf* on a small category C is a functor defined as;

$$F : C^{op} \to \mathbf{Set}$$

[NLab]

A *presheaf* is a *functor* from the *opposite category* to the *Set*. This can be seen as a generalization of the notion of a *set-valued function*. It is a way to assign each object in a given category to a set of objects, in such a way that the assignments are compatible with the morphisms in the *category*.

Example 3.6. Consider the *functor* F: $\mathbf{Grp}^{op} \to \mathbf{Set}$ its *domain* is the *opposite category* of all small groups and its *codomain* is the *category* of all small sets.

To study the behavior of the properties of pairs of *objects* and *morphisms*. We introduce the concept of a *product category*. It allows us to take the product of two categories and create a new *category* out of it. This enables us to couple the objects and morphisms of the two *categories* and studies their behaviors as pairs. It is also used to study and *projections* in *categories*.

Definition 3.7. Let C and D be two categories. Their *product category* is denoted as $C \times D$ and is the category whose:

1. Objects are ordered pairs $(c,\ d)$ where $c \in C$ and $d \in D$
2. Morphisms are ordered pairs $((c \to c'),\ (d \to d'))$
3. Composition of morphisms is defined componentwise

We now come to a significant point in the paper where we are almost ready to introduce the reader with *Yoneda Lemma*. But before we do that we need to go ahead and introduce the concept of a *natural isomorphism* which we will use to prove *Yoneda's Lemma*.

Definition 3.8. Let F and G be two functors. A *natural isomorphism* τ: $F \implies G$ is a *natural transformation* with two-sided inverse.

A natural isomorphism is a special type of morphism that establishes a correspondence between two functors in a way that is consistent with the structure of the categories involved. The importance of natural isomorphism lies in the fact that they provide a way to compare functors and determine whether they are essentially the same.

We now give the last definition which is *representable functor*. It is a *functor* that can be represented by an object in the *category*. This means that the object representing the *functor* captures all of the relevant information about its output.

Definition 3.9. Let C be a *locally small category* and $c \in C$ than we define a representable functor F such that:

$$Hom(c,\text{-}):\ Hom \to Set$$

$$Hom(c, \text{-})(a) = Hom(c,\ a)$$

and if $f \in Hom(a,\ b)$ then,

$$Hom(c, \text{-})(f) = Hom(c, f)\colon\ Hom(c,\ a) \to Hom(a,\ b)$$

Now that we have introduced the reader to all of the necessary information that is required to discuss *Yoneda's Lemma*. We finish this chapter and continue with the next one where the reader will be provided with *Yoneda Lemma* and proof of it.

4. Yoneda Lemma

Up until now we created a foundation to work on top of and introduced the reader with the necessary structures and tools to understand *Yoneda's Lemma*. In this section we will move forward and introduce the lemma and provide a proof for it. Afterwards discuss some of its applications and finish the paper with the ending remarks.

Lemma 4.1. *Let $\mathcal{A}$ be a locally small category and consider a functor F: $\mathcal{A} \to$* **Set**. *An object $A \in \mathcal{A}$ and the corresponding representable functor Hom(A, -): $\mathcal{A} \to$* **Set**. *Then the following correspondence is a bijection. Moreover, this bijection is a natural in both way.* [CTP3]

$$\theta_{F,A} : Nat(Hom(A,-), F) \to FA$$

$$\theta_{F,A}(\alpha) = \alpha_A(1_A)$$

Proof. *Yoneda Lemma* states that the given correspondence $\theta_{F,A}$ with the *domain* $Nat(Hom(\text{A}, \text{-}),\ F)$ which is the set of all *natural transformations* in between the *representable functor* $Hom(\text{A}, \text{-})$ and the set valued *functor* F and *codomain* $F\text{A}$, which is a *small set*, This correspondence can be thought as matching each *natural transformation* with a distinct set. Hence it enables us to create a link between these sets and *natural transformations*. To prove this *natural* correspondence. We will define a *natural transformation* and show that it a *two-sided inverse* of $\theta_{F,A}$ that will show the *natural isomorphism* and *bijection* will follow.

Hence we define the following, $\forall$ B $\in \mathcal{A}$ let

$$\tau(a)_B : \mathcal{A}(A,-) \to F$$

$$\tau(a)_B : \mathcal{A}(A,B) \to FB$$

$$\tau(a)_B(f) = F(f)(a) \in FB$$

Since $\forall\ g\colon\ B \to C \in \mathcal{A}$

$$Fg \circ \tau(a)_B(f) = Fg(Ff(a)) = Fg \circ Ff(a)$$

$$Fg \circ Ff(a) = F(g \circ f)(a) = \tau(a)_C(A(A, g)(f))$$

Hence,

$$F(g) \circ \tau(a)_B = \tau(a)_C \circ A(A, g)$$

holds, and the following diagram commutes

$$\begin{array}{ccc} A(A,B) & \xrightarrow{\tau(a)_B} & FB \\ {\scriptstyle A(A,g)}\downarrow & & \downarrow{\scriptstyle Fg} \\ A(A,C) & \xrightarrow{\tau(a)_C} & FC \end{array}$$

Since the above diagram commutes, we observe that τ is a *natural transformation* and the only thing that is needed to complete the proof is to show that θ and τ are inverses of each other. Hence, we first start by showing the right hand side. Let $a \in F$A, then we have the following:

$$\theta_{F,A}(\tau(a)) = \tau(a)_A(1_A) = (F1_A)(a)$$

and

$$(F1_A)(a) = 1_{FA}(a)$$

For the left hand side, let α: A(A, -) $\implies$ F and f: A $\to$ B $\in$ A then,

$$\tau(\theta_{F,A}(\alpha))_B(f) = \tau(\alpha_A(1_A))_B(f)$$

$$\tau(\alpha_A(1_A))_B(f) = F(f)(\alpha_A(1_A))$$

Here we use the fact that α is a *natural transformation*

$$F(f)(\alpha_A(1_A)) = \alpha_B(A(A, f)(1_A)) = \alpha_B(f \circ 1_A)$$

and finally we observe that

$$\alpha_B(f \circ 1_A) = \alpha_B(f)$$

Hence we have show that τ is a two sided inverse of θ and completed the proof. [YL]

$\square$

Now that we have provided the reader with the lemma itself and a proof of it last of all we will talk about some of its applications both in different domains of mathematics and in other domains such as computer science. In *category theory* itself, the *Yoneda Lemma* is used in the study of *functors* and *natural transformations*, and it is also used in the study of *adjoint functors*, which are pairs of *functors* that behave in a certain way with respect to one another.

In *algebraic topology*, *Yoneda Lemma* is used to define *Yoneda Product* which in return plays a critical role while defining *cohomology groups*, which are a way of measuring the *symmetry* or *asymmetry* of a *topological space*. another example from another domain of mathematics is in *algebraic geometry*, where *Yoneda Lemma* is used to define the concept of a *sheaf*, which is a mathematical object that captures the local behavior of a space.

For an example outside mathematics in *theoretical computer science*, it is used in the study of *type theory* to define the concept of a *higher inductive type*, which is a way of defining complex types using recursive definitions and with the rest of the *category theory* it plays a abstract yet critical role in the design of programming language type systems, most famously *Haskell*.

5. Final Remarks

Throughout this paper, we introduced the reader with the basic concepts in category theory and introduced one of the fundamental results named *Yoneda Lemma*. We also provided introductory concetps in *category theory*, and highlighted some of the many applications that they offer. *Category theory* provides a powerful and versatile *mathematical language* for understanding and manipulating structure in a very general way, and *Yoneda Lemma* is a significant result of it. Overall, the study of *category theory* can be challenging, but it is a rewarding and exciting field that offers many insights and applications. Understanding these concepts can help us to better understand and solve problems in a wide range of areas, and it can provide us with a new way of thinking about structure and abstraction. We hope that this paper has provided a helpful introduction to these meta and fascinating ideas.

References

[Mac Lane] S. Mac Lane, *Categories for the Working Mathematician*, Springer, March 1997

[NLab] NLab Authors, *Category Theory*, December 2022 https://ncatlab.org/nlab/revision/category+theory/264

[YL] R. Thomas, T. Jean-Claude, *Practical Work In Mathematics, Yoneda Lemma*, Ecole Polytechnique Federale De Lausanne, 10 May 2013

[CTP3] Q. Kuang, *Category Theory, Part III*, University of Cambridge, Fall 2018

[TYL] P. Asima, *The Yoneda Lemma*, Ludwig Maximilians-Universitat, München, 15 September 2021

[ESPF] H. Peter., *Examples of Some Properties of Functors*, 3 January 2018

[CTM] A.J. Berrick, M.E. Keating, *Category Theory and Modules*, Cambridge University Press, 2000

Department of Mathematics, Bilkent University, 06800 Bilkent, Ankara, Turkey
Email address: emre.caniklioglu@ug.bilkent.edu.tr