Category Theory

Steven Hwang

May 2020

Introduction

The purpose of this paper is to acquaint the reader with the fundamental concepts of category theory. This paper is aimed at those who have are already accustomed with various concepts in abstract algebra, and groups in particular. However, definitions of these abstract structures are also provided for completeness.

The paper will be light on theorems, as the approach of this paper will be to define concepts in category theory and then demonstrate their applications to various other fields. Through this paper, we aim to give the reader a taste of how category theory really is a good generalization of many other fields.

Category Theory can be thought of as the study of the most general abstract structures. Almost every structure in every branch of mathematics may be considered to be a category, including (but not limited to): Groups, Monoids, Vector Spaces, Sets, Ordered Sets, Graphs, Topological Spaces, and even Categories themselves. As a result, many results in category theory have applications to other branches of mathematics.

But aside from their applications, categories themselves are interesting objects to work with. We hope that by reading this paper, the reader will be inspired by both the beauty and the complexity of these abstract objects.

Categories

Categories can be thought of as the biggest generalization of abstract structures. There is some collection of things, called *objects*; some way to move from a thing to another thing, called *morphisms*; a way to put together multiple movements, called *composition*, and some way to get from each thing to itself, called *identity*.

We categories more formally as follows.

Definition. *category.* A category C consists of

- A collection of objects $ob(C)$.
- A collection of morphisms $hom(C)$. Each morphism $f \in hom(C)$ maps from a source object $a \in ob(C)$ to a target object $b \in ob(C)$, notated $f : a \to b$.
- A composition operation on morphisms $\circ$. For elements $a, b, c \in ob(C)$, given a morphism $f : a \to b$ and another morphism $g : b \to c$, then the composition operation satisfies $f \circ g : a \to c$.
- An identity morphism id_a for each object $a \in ob(C)$. The identity map sends the object to itself, $id_a : a \to a$. If the object is implied then the identity is simply denoted id.

For morphisms, it will be useful to define the following notations:

Definition. *src, tgt.* Let $f : A \to B$ be a morphism in a category C. Define:

$$src(f) = A$$
$$tgt(f) = B$$

We have now defined categories as a collection of things, but the thing doesn't have any structure yet. So we define some axioms on the categories to give it structure.

Definition (cont'd). *category axioms.* All categories C must also satisfy:

- Uniqueness: $f : a \to b, f : a' \to b' \Rightarrow a = a', b = b'$.
- Associativity: $f \circ (g \circ h) = (f \circ g) \circ h$
- Identity Composition: $f \circ id = id \circ f = f$

Example. *Sets.* A classical example of a category is the category of sets, denoted *Set*. We define each part of the category as follows:

The category is the category of sets, $C = Set$

- The objects are sets, $a \in ob(Set) \Rightarrow a = S$
- The morphisms are total functions on the sets (that is, functions which are defined on each element in the set), $f \in hom(Set) \Rightarrow f : S \to T$
- The composition operation is the standard composition operation for total functions on sets, $f : S \to T, g : T \to U \Rightarrow g \circ f : S \to U$ where $g \circ f(x) = g(f(x))$ for $x \in S$

- The identity morphism for each object is the identity operation for sets, $id_S = id : S \to S$, where $id(x) = x$ for $x \in S$

Additionally, we should verify that the axioms of categories hold for the category *Set*.

- Uniqueness: total functions on sets are unique in the sense that they are only defined one source set, and they always take that source set to the same target set
- Associativity: total functions on sets are associative under the composition operation
- Identity Composition: composing a total function between sets with the identity function on either set (source or target) will preserve the original function

Example. *Groups.* Groups also form a category, denoted *Grp*. We define each part of this category like so:

The category is the category of groups, $C = Grp$

- The objects are groups, $a \in ob(Grp) \Rightarrow a = G$
- The morphisms are homomorphisms between the groups (essentially, maps which preserve the operational structure of the groups), $f \in hom(Grp) \Rightarrow f : G \to H$
- The composition operation is the standard composition operation for homomorphisms on groups, $f : G \to H, g : H \to I \Rightarrow g \circ f : G \to I$ where $g \circ f(x) = g(f(x))$ for $x \in G$
- The identity morphism for each object is the identity homomorphism for groups, $id_G = id : G \to G$, where $id(x) = x$ for $x \in G$

The verification that the above definition of the category *Grp* also obeys the axioms of categories is similar to the verification performed for the category *Set*, and is left as an exercise for the reader.

There's also another cool way to create a category using groups. Rather than letting our category represent all groups, we can actually create a one-element category from a particular group. Here's how.

The category is the one-element category for a group, $C = Grp^*$.

- There is only one object. This object represents the group G, but it doesn't really matter what we call it. So let's denote it $\triangle \in ob(Grp^*)$

- The morphisms are all morphisms from $\triangle$ to itself. We let each element of the group $g \in G$ represent a morphism from $\triangle$ to itself, $g \in hom(Grp^*)$
- Out of all the morphisms from $\triangle$ to itself, one of them should be the identity morphism. We let the identity element e of G represent this identity morphism, $id_\triangle = e$
- We define composition between morphisms as multiplication between the corresponding elements of the group. In other words, if $g_1 : \triangle \to \triangle$ and $g_2 : \triangle \to \triangle$ are morphisms $g_1, g_2 \in hom(Grp^*)$, and they are also elements of the group $g_1, g_2 \in G$, then we define $g_2 \circ g_1 = g_2 * g_1 : \triangle \to \triangle$, where $*$ is the multiplication operation of the group.

The verification that this definition of the category of a group indeed satisfies the category axioms should be informative in understanding the nature of categories, and is left as an exercise for the reader.

Example. *Graphs.* A more novel example of a category is the category of directed multigraphs, denoted *Grph*. This category is defined as such:

The category is the category of a directed multigraphs (graphs where each edge has a direction, and there may be multiple edges connecting the same two vertices), $C = Grph$. It will also be helpful to define the graph $G = (V, E)$, where G is a directed multigraph, V is the set of vertices in the graph, and E is the set of edges in the graph

- The objects are vertices in G, $a \in ob(Grph) \Rightarrow a = v$ for $v \in V$
- The morphisms are directed edges between the vertices, $f \in hom(Grph) \Rightarrow f = e$ for $e \in E$. Also note that the source and the target of morphism f correspond to the source and target vertices of the direct edge
- The composition operation is the path adjoining operator for directed edges, $e_1 : v_1 \to v_2, e_2 : v_2 \to v_3 \Rightarrow e_2 \circ e_1 : v_1 \to v_3$
- The identity morphism for each object is the self-loop at each vertex, $id_v = id : v \to v$ where id is the directed edge from v to itself. For the types of graphs in this category, each vertex must have a self-loop

We will verify that the category *Grph* obeys the axioms of categories.

- Uniqueness: edges are only defined between one source vertex and one target vertex
- Associativity: when composing directed edges to create a larger directed path, the order in which the edges are combined does not matter

- Identity Composition: when moving along a directed edge, going around a self-loop at either the source or the target vertex does not change the original path of the edge

In addition to the examples mentioned above, there are many other types of categories, such as the category of monoids (groups without inverses) *Mon*, the category of vector spaces over a common field K $Vect_K$, the category of topological spaces *Top*, and even the category of small categories *Cat* (more on this later). It just goes to show how general of a structure categories are!

Categories are a cool abstract generalization of various structures in mathematics, but by themselves they aren't too useful. We want to be able to say something about them, and some statements that we make about general categories might also translate to theorems about more specific fields of math. Like with most abstract algebra courses (such as group theory), we begin by exploring the notion of nested categories (categories within categories).

Definition. *subcategory.* A subcategory D of category C is a category such that

- Subset of Objects: $ob(D) \subset ob(C)$
- Subset of Morphisms: $hom(D) \subset hom(C)$
- Identity Morphisms Exist: $\forall d \in ob(D)$, $id_d \in hom(D)$
- Closedness of Morphisms: $\forall f \in hom(D), src(f), tgt(f) \in ob(D)$

Notice that the above definition of subcategory doesn't actually explicitly state that D is a category. However, these are in fact equivalent.

Theorem. Using the above definition of subcategory, D is a subcategory of C if and only if D is a category with objects and morphisms inherited from C.

Proof. The proof involves verifying that all of the properties and axioms of categories hold for the subcategory D, and is left to the reader as an exercise.

Example. *subcategories of Grph.* When the category is *Grph*, and the entire category represents one big graph G, the subcategories are closed subgraphs G' of G. In other words, if $G = (V, E)$, then a subgraph $G' = (V', E')$ has some subset of vertices of the original graph $V' \subset V$, and the edges are a subset of the edges in the original restricted to the vertices of the subgraph, $e' \in E' \Rightarrow e' = (v_1', v_2')$ where $v_1', v_2' \in V'$. The notation $e = (v_1, v_2)$ is used to denote an directed edge from vertex v_1 to vertex v_2.

The reader may work out the analogues of subcategories for the categories *Set* and *Grp*.

The notion of subcategories will come in handy in the future. But with just categories and subcategories, we still can't say anything particularly meaningful about categories. Once again like with many fields of abstract algebra (such as group theory), we explore structure-preserving maps between categories. In category theory, these maps are known as functors.

Functors

Functors are maps between categories which preserve their categorical structure. However, since categories consist of a lot of objects and operations, we should be more specific about what the functors are actually doing within the categories. Their formal definition is the following.

Definition. *functor.* A functor F between categories C and D, notated $F : C \to D$, is a map which does the following:

- F sends objects in C to objects in D: $F(c) = d$, where $c \in ob(C)$ and $d \in ob(D)$
- F sends morphisms in C to morphisms in D: $F(f) = g$, where $f \in hom(C)$ and $g \in hom(D)$

So now we know which things the functors are acting on, but we also said that functors should preserve categorical structure. What we mean by that is that functors should also satisfy the following axioms:

Definition (cont'd). *functor axioms.* All functors F must also satisfy:

- F sends identity morphisms to identity morphisms: if $c \in ob(C)$ and $d = F(c) \in ob(D)$, then $F(id_c) = id_d$, where $id_c \in hom(C)$ and $id_d \in hom(D)$
- F sends the composition operation to the composition operation: if $f_1, f_2 \in hom(C)$ and $g_1 = F(f_1), g_2 = F(f_2) \in hom(D)$, then $F(f_1 \circ f_2) = g_1 \circ g_2$, where the former composition is composition in C, while the second composition is composition in D

Just like in other fields of abstract algebra, when we consider functors between categories, we also like to consider functors from categories to themselves. These are known as endofunctors.

Definition. *endofunctor.* A functor F from a category C to itself, $F : C \to C$, is called an endofunctor.

Functors between categories is already a pretty abstract concept to wrap our heads around, so let's see some concrete examples of what functors between categories can look like.

Example. *functors between Grp^*.* Suppose we have two groups G and H, and we build categories Grp_G^* and Grp_H^* out of both of them (remember, these are the one-element categories we defined earlier). We would like to define a functor between them, $F : Grp_G^* \to Grp_H^*$.

Since these are one-element categories, let $ob(Grp_G^*) = \{\triangle\}$, and let $ob(Grp_H^*) = \{\square\}$. The first thing that functor F should do is that it should map objects in Grp_G^* to objects in Grp_H^*. Since there is only one object in either of these groups, functor F satisfies

$$F(\triangle) = \square$$

In addition to mapping objects to objects, functor F should also map morphisms to morphisms. Remember, the morphisms of this particular category is just the elements of their respective groups: $hom(Grp_G^*) = G$ and $hom(Grp_H^*) = H$. Therefore, in terms of its treatment of morphisms, functor F is effectively a map between groups: for each $g \in G$,

$$F(g) = h$$

where $h \in H$.

We should also make sure that F obeys the functor axioms. For $\triangle \in ob(Grp_G^*)$ and $\square \in ob(Grp_H^*)$, F must map the identity morphism for $\triangle$ to the identity morphism for $\square$. We previously said that these identities were e_G and e_H respectively, where e_G and e_H are the identity elements of their respective groups. Hence, this property ultimately tells us that

$$F(e_G) = e_H$$

The other functor axiom is that morphism composition should be preserved through application of the functor. In other words, if g_1 and g_2 are morphisms in $hom(Grp_G^*)$, then they must satisfy

$$F(g_2 \circ g_1) = F(g_2) \circ F(g_1)$$

In fact, if you take a step back and look at F's effect on the morphisms (its effect on the objects doesn't mean much, as there is just one object in both categories), we've shown:

- F takes elements $g \in G$ (morphisms in $hom(Grp_G^*)$) to elements $h \in H$ (morphisms in $hom(Grp_G^*)$)
- F takes e_G (the identity morphism for $\triangle$) to e_H (the identity morphism for $\square$)

- F applied to a composition of two elements (the composition of two morphisms in $hom(Grp^*_G)$) equals F applied to each individual element and then composed (the composition of two morphisms in $hom(Grp^*_G)$.

For groups, this is exactly the definition of a homomorphism between group G and group H! Hence, functors between categories Grp^*_G and Grp^*_H are essentially equivalent to homomorphisms between groups G and H.

The last thing we would like to touch upon in this paper is how functors can be used in a more theoretical context. We've seen what functors translate to when applied to specific types of categories, but how might we use them outside of applications? One cool thing functors allow us to do is to define the category of categories!

Before we can define a category of categories, there's one other technicality we have to think about. All this time, we've been thinking about the collection of objects and the collection of morphisms as sets—but in reality, this only applies to a specific type of category. We provide the following definition to clear up this distinction.

Definition. *small category.* A category C is called small if $ob(C)$ and $hom(C)$ can be treated as sets.

After seeing this definition, a logical question would be to ask, "how can a category not be small?" Well, in cardinality theory it happens to be the case that some collections are "too large" to be sets. For instance, the set of all sets is not considered to be a set, because it is "too large." Another example of something that is "too large" to be a set is the set of all groups. Yet while some of the categories we've been working with, namely *Set* and *Grp*, are large categories, others are small categories, such as *Grph.* The actual distinction between what can be called a set and what is too large to be called a set is outside of the scope of this paper, although it is relevant for the next definition.

We propose that the category whose objects are small categories forms a category. The intuition for why large categories must be excluded is that for categories which are too large, it becomes difficult to define morphisms between those categories.

Example. *Small Categories.* The category of small categories, denoted *Cat*, is defined as such:

- The objects are small categories, $ob(Cat)$ = set of all small categories.
- The morphisms are functors between the small categories, $hom(Cat)$ = set of all functors on small categories.
- Given a small category $C \in ob(Cat)$, the identity morphism on that object is the identity endofunctor I_C on that category.

- The composition of morphisms $F_1 \circ F_2$ is the composition of functors, and is defined as one would expect: if $F_1 : A \to B$ and $F_2 : B \to C$, then $F_2 \circ F_1 : A \to C$ such that the transformations on both the objects and the morphisms are composed, $F_2 \circ F_1(a) = F_2(F_1(a))$ and $F_2 \circ F_1(f) = F_2(F_1(f))$.

Theorem. The category of small categories *Cat* defined above is actually a category.

Proof. This is a routine verification of the axioms of categories, but it is worth reviewing for completeness.

First, we verify that morphisms have unique sources and targets. Functors are only defined as a map between one source category and one target category, so if $F : A \to B$ and $F : A' \to B'$, then $A = A'$ and $B = B'$ because it wouldn't make sense otherwise.

Then, we show that morphisms are associative. Functors are indeed associative, both when it comes to objects and when it comes to morphisms in the categories. First, suppose $F_1 : A \to B$, $F_2 : B \to C$, and $F3 : C \to D$. Consider some object $a \in ob(A)$. Then, just using the definition of the composition of functors,

$$
\begin{aligned}
&(F_3 \circ (F_2 \circ F_1))(a) \\
&= F_3((F_2 \circ F_1)(a)) \\
&= F_3(F_2(F_1(a))) \\
&= (F_3 \circ F_2)(F_1(a)) \\
&= ((F_3 \circ F_2) \circ F_1)(a)
\end{aligned}
$$

The same process should apply for morphisms $f \in hom(A)$, because the definition of functor composition is no different for morphisms than it is for objects,

$$
\begin{aligned}
&(F_3 \circ (F_2 \circ F_1))(f) \\
&= F_3((F_2 \circ F_1)(f)) \\
&= F_3(F_2(F_1(f))) \\
&= (F_3 \circ F_2)(F_1(f)) \\
&= ((F_3 \circ F_2) \circ F_1)(f)
\end{aligned}
$$

This shows that functors are associative under composition, in the sense that both their object mappings and morphism mappings are also associative.

Finally, we show that composition with the identity morphism does nothing. The way identity endofunctor I_C on a category C is specifically defined as the "do nothing" operation: it sends objects to themselves $I_C(c) = c$ and it sends morphisms to themselves $I_C(f) = f$. When the identity endofunctor is composed with any other functor, the end result should just be the other functor, because the identity endofunctor leaves its category in the exact same state as before. Hence $F \circ I = I \circ F = F$.

References

https://maartenfokkinga.github.io/utwente/mmf92b.pdf
https://en.wikipedia.org/wiki/Category_(mathematics)
https://en.wikipedia.org/wiki/Functor
https://math.stackexchange.com/questions/1147444/group-as-a-category-with-one-object
https://math.stackexchange.com/questions/678287/what-is-an-example-of-a-large-category
https://proofwiki.org/wiki/Composition_of_Functors_is_Associative