Table of contents

Chapter 1
Preface

1.1 Motivation

This note is about the basic aspects of category theory. There have been many books on category theory, almost all of them contains many examples from multiple areas of mathematics. These examples, however, needs mathematical background on many areas, making it hard to read.

In this note we focus on the core concepts of category theory, trying to understand them in an intuitive way. This endeavor make us think about things from the view of category theory, whenever we meet a mathematical concept, no matter it is known or new to us, Examples are shown only when it is essential.

1.2 Writing Style

The writing style follows the suggestions given by *How to Think Like a Mathematician*. Some conventions make it better:

- Title of each subsection is a sentence that summarizes that section. With this, you can review by simply reading content.
- Definitions are bold.

1.3 Technique Limitation

It is hard to draw diagrams in TeXmacs. So, all the commutative diagrams are drawn by quiver online, and pasted to the text by screenshot.

Chapter 2

Category, Functor, and Natural Transformation

2.1 Category

2.1.1 Category is about arrows

Category is the fundamental element of category theory. A category consists of arrows and objects, which are employed to declare arrows: where an arrow emits, and where it ends. Moreover, several properties relating to these components should be satisfied.

Strangely, in category, arrow is called morphism, a word derived from isomorphism. And isomorphism is constructed from iso-morphe-ism, where morphe, a Greek word, means shape or form. So, isomorphism means equal shape or form. This can be easily illustrated in topology, where the two isomorphic tologogical space share the same form (but may not the same shape). But semantically, this is far from what arrow should mean. So, the question is why mathematician use the word morphism for arrow. A guess is that it may come from homomorphism; and in algebra, a homomorphism is an arrow between algebric structures.

Definition 2.1. *[Category] A* ***category*** C *consists of*

- *a collection of* ***objects****,* ob_{C}*,*
- *for each* $A, B \in \mathrm{ob}_{\mathsf{C}}$*, a collection of* ***morphisms*** *from* A *to* B*,* $\mathrm{mor}_{\mathsf{C}}(A, B)$*, where* A *is the* ***domain*** *and* B *the* ***codomain****,*
- *for each* $f \in \mathrm{mor}_{\mathsf{C}}(A, B)$ *and* $g \in \mathrm{mor}_{\mathsf{C}}(B, C)$*, a* ***composition*** *of* f *and* g *that furnishes a morphism* $g \circ f \in \mathrm{mor}_{\mathsf{C}}(A, C)$*, and*
- *for each* $A \in \mathrm{ob}_{\mathsf{C}}$*, an* ***identity*** $1_A \in \mathrm{mor}_{\mathsf{C}}(A, A)$*,*

such that the following axioms are satisified:

- ***associativity****: for each* $f \in \mathrm{mor}_{\mathsf{C}}(A, B)$*,* $g \in \mathrm{mor}_{\mathsf{C}}(B, C)$*, and* $h \in \mathrm{mor}_{\mathsf{C}}(C, D)$*, we have* $(h \circ g) \circ f = h \circ (g \circ f)$*, and*
- ***identity****: for each* $f \in \mathrm{mor}_{\mathsf{C}}(A, B)$*, we have* $f \circ 1_A = 1_B \circ f$*.*

But, this notation system is a little complicated. Usually, we simplify it by employing the notations from set theory.

Notation 2.2. *Given category* C*, we simplify the notation* $A \in \mathrm{ob}_{\mathsf{C}}$ *by* $A \in \mathsf{C}$*, and for each* A*,* $B \in \mathsf{C}$*, denote* $f \in \mathrm{mor}_{\mathsf{C}}(A, B)$ *by* $f\colon A \to B$*,* $A \xrightarrow{f} B$*, or* $f \in \mathsf{C}(A, B)$*.*

Now, category becomes much more familiar to us. We can think the objects of C as sets, and morphism as function, which is the map between sets. Indeed, the collection of sets and functions does form a category: the category of sets.

Definition 2.3. *[Category of Sets] The* ***category of sets****, denoted as* Set*, has the collection of all sets as its objects, and for each* $A, B \in \mathsf{Set}$*, the collection of all functions from* A *to* B *as its morphisms from* A *to* B*.*

It is easy to check that the axioms are satisfied. But, be caution! Set is just a specific category, it helps us understanding what a category might look like. But, Set has much more axioms, or restrictions, than the category itself, thus may blind us to the potential power of arrows.

Indeed, there exist categories whose objects are not sets, or whose morphisms are not maps. So, a better way of thinking category is keeping objects and morphisms abstract. You can think objects as dots and morphisms as arrows between the dots.

2.1.2 Objects may not be sets

We know that the symmetry group of rectangle is dihedral group D_2. The group elements are operations: identity, rotation of 180 degrees, and reflections along vertical and horizental directions. The operand is unique: the rectangle. These operations can be viewed as arrows from the rectangle to itself. So, this symmetry group describes a category, called $\mathsf{B}D_2$. The axioms of category are satisfied because of the group properties. In this category, object is not a set, but an rectangle.

In fact, all groups are examples illustrating that objects may not be sets. Here, we need the definition of isomorphism in category.

Definition 2.4. *[Isomorphism in Category] In a category* C*, for any objects* $A, B \in \mathsf{C}$*, a morphism* $f: A \to B$ *is an* ***isomorphism*** *if there exists* $g: B \to A$ *such that* $g \circ f = 1_A$ *and* $f \circ g = 1_B$*. Denote* $A \cong B$ *if* A *and* B *are isomorphic.*

With the aid of isomorphism, we definition the groupid.

Definition 2.5. *[Groupid] A* ***groupid*** *is a category in which all morphisms are isomorphisms.*

Now, we come to the big step: define the group! But, wait a minute. We have learned abstract algebra and known what a group is. The point here is that category theory provide a new, but equivalent, way of defining group, using arrows!

Definition 2.6. *[Group as Category] A* ***group*** *is a groupid in which there is only one object.*

Notation 2.7. *Because there is only one object, we can simplify the notation of morphism like* $f: A \to B$ *to* f*, and denote* $f \in \mathsf{G}$ *for that* f *is a morphism of a group* G*.*

If we start at defining group by arrows, we have to declare that the properties (axioms) of group studied in algebra are satisfied.

Theorem 2.8. *Let* G *a group, then we have*

- ***associativity****: for* $\forall f, g, h \in \mathsf{G}$*,* $(f \circ g) \circ h = f \circ (g \circ h)$*,*
- ***identity element****: there exists* $1 \in \mathsf{G}$ *such that for* $\forall f \in \mathsf{G}$*,* $f \circ 1 = 1 \circ f = f$*.*
- ***inverse element****: for* $\forall f \in \mathsf{G}$*, there exists* $g \in \mathsf{G}$ *such that* $f \circ g = g \circ f = 1$*.*

Proof. The associativity of group is identified as the associativity of category. The same for identity element. The inverse element comes from the fact that all morphisms are isomorphisms. □

So, a group is a category. And as compared with the group defined in abstract algebra, we find that the unique object in this category is not set, and that discussing what the object should be is meaningless.

2.1.3 Morphisms may not be maps

To illustrate that morphisms may not be maps, we need to define preorder and poset.

Definition 2.9. *[Preorder] Given a set S, a* ***preorder*** *P on S is a subset of $S\times S$ such that the following axioms are satisfied:*

- ***reflexivity**: for $\forall a\in S$, $(a,a)\in P$, and*
- ***tansitivity** for $\forall a,b,c\in S$, if $(a,b)\in P$ and $(b,c)\in P$, then $(a,c)\in P$.*

For example, "no greater than" is a preorder, where S is the set of real numbers and $(a,b)\in P$ means $a\leqslant b$. "Is subset of" is another example of preorder, where S is the set of sets and $(a,b)\in P$ means $a\subseteq b$.

Definition 2.10. *[Poset] A preordered set, or* ***poset**, (S,P) is a set S equiped with a preorder P on S.*

With these prelimiaries, we claim that a poset is a category.

Definition 2.11. *[Poset as Category] Given poset (S,P), a category* Poset *can be constructed by regarding the elements in S as objects and regarding $(a,b)\in P$ as $a\rightarrow b$.*

Because of the axioms of preorder, the axioms of category are satisfied. The category Poset illustrates that morphisms are not maps. In Poset, morphisms are "no greater than"s or "is subset of"s.

2.1.4 Supremum and infimum are dual

Arrows can represent many mathematical objects. For example, in Poset with set $\mathbb{R}$ and preorder $\leqslant$, we can describe supremum as follow.

Definition 2.12. *[Supremum in Category] Given a subset $A\subset\mathbb{R}$. An $x\in\mathbb{R}$ is the* ***supermum*** *of A if it satisfies:*

- *for $\forall a\in A$, $a\rightarrow x$ and,*
- *for $\forall y\in\mathbb{R}$ and $\forall a\in A$, if $y\rightarrow a$, then $y\rightarrow x$.*

This is, again, a weired definition on supremum. But, if you check carefully, you can see that this definition is equivalent to that studied in analysis. Also, we can define the infimum in the same fashion.

Definition 2.13. *[Infimum in Category] Given a subset $A\subset\mathbb{R}$. An $x\in\mathbb{R}$ is the* ***infimum*** *of A if it satisfies:*

- *for $\forall a\in A$, $x\rightarrow a$ and,*
- *for $\forall y\in\mathbb{R}$ and $\forall a\in A$, if $a\rightarrow y$, then $x\rightarrow y$.*

Weired again, but now you may have been familiar with the weired. Hint: the word weired also has the meaning of fate. Indeed, you are on the load to wonderland. By comparing the definition of infimum to that of supremum, we find all statements are the same except that we replaced supremum by infimum and domain by codomain (for instance, replaced $y\rightarrow x$ by $x\rightarrow y$). Two statements are **dual** if you can get one from the other by simply flipping all the arrows, or, equivalently, by exchanging the domain and codomain for each morphism in the statement. We say that supremum and infimum are dual.

2.1.5 Morphisms in the dual category of Set are not maps

There are also dual categories. Given a category C, its **dual category**, denoted by C^{op}, is obtained from C by exchanging the domain and codomain for each morphism in C.

So, in the dual category of Set, i.e. $\mathsf{Set}^{\mathrm{op}}$, we find that arrows are not functions, not even maps! Yet another example whose morphisms are not maps.

2.2 Why Category?

2.2.1 Arrows generalize concepts and theorems from one area to every area in mathemtatics

Why category theory? Or say, why arrows? One benefit of re-claim everything in arrows is the ability of generalizing a concept in one area to area domain in mathematics. An example comes from generalizing the Cartesian product, also called direct product, in the set theory.

You have been familiar with the direct product of two sets. Given two sets A and B, recall that the direct product $A \times B := \{(a, b) | a \in A, b \in B\}$. Again, for generalizing the concepts using category theory, we have to re-write the concepts using arrows. And again, this re-writing looks weired at the first sight.

Definition 2.14. *[Direct Product of Two Objects] Given a category C. For any $A, B \in \mathsf{C}$, the **direct product** of A and B is another object $C \in \mathsf{C}$ together with two morphisms $\alpha: C \to A$ and $\beta: C \to B$ such that, for any $C' \in \mathsf{C}$, any $\alpha': C' \to A$ and $\beta': C' \to B$, there exists an unique morphism $\gamma: C' \to C$ so that $\alpha' = \alpha \circ \gamma$ and $\beta' = \beta \circ \gamma$.*

So, a direct product of objects A and B and a triplet (C, α, β). Applying to Set, as you can check directly, it goes back to the Cartesian product of two sets. We can also apply it to Grp, which furnishes the group direct product:

Definition 2.15. *[Group Direct Product] Given two groups G and H, the **group direct product** of G and H is defined as $\{(g, h) | g \in G, h \in H\}$ equipped with group multiplication $(g, h) \times (g', h') := (g \circ g', h \cdot h')$ where $\circ$ is the muliplication of G and $\cdot$ of H.*

It is like the Cartesian product, but extra structure are *implied*.

Also, all specific categories would be benefited from a theorem claimed in category theory. Such as the uniqueness of direct product in the sense of isomorphism.

Theorem 2.16. *[Uniqueness of Direct Product] Given a category C. For any $A, B \in \mathsf{C}$ and any two direct products (C, α, β) and (C', α', β'). Then, there exists an unique isomorphism $\gamma: C' \to C$ such that $\alpha' = \alpha \circ \gamma$ and $\beta' = \beta \circ \gamma$.*

That is, C and C', α and α', β and β' are equivalent in the sense of isomorphism.

Proof. Left to reader. □

This theorem holds not only for Cartesian product of sets, but also, for instance, for the group direct product.

2.2.2 Duality Helps Create New Concepts and Theorems, Freely!

Another benefit of viewing everything in arrows is duality. In category theory, it is natural to think what would happen if we exchange domain and codomain for all the arrows. Just like the relation between supremum and infimum, it is natural to ask what if we exchange domain and codomain for all the arrows in the definition of direct product. This furnishes a new concept we called direct sum.

Definition 2.17. *[Direct Sum of Two Objects] Given a category C. For any $A, B \in \mathsf{C}$, the **direct product** of A and B is another object $C \in \mathsf{C}$ together with two morphisms $\alpha: A \to C$ and $\beta: B \to C$ such that, for any $C' \in \mathsf{C}$, any $\alpha': A \to C'$ and $\beta': B \to C'$, there exists an unique morphism $\gamma: C \to C'$ so that $\alpha' = \gamma \circ \alpha$ and $\beta' = \gamma \circ \beta$.*

Again, a direct sum of objects A and B and a triplet (C, α, β). Comparing with direct product, direct sum is nothing but exchanging domain and codomain for all the arrows in the statement of direct product.

Applying to Set, as it can be directly checked, we get the disjoint union of two sets. Given two sets A and B, recall that the disjoint union $A \cup_d B: = \{(a, 1) | a \in A\} \cup \{(b, 2) | b \in B\}$. This is a surprise, since, unlike the duality between supremum and infimum, Cartesian product and disjoint union do not look like a pair at the first sight!

Recall the theorem that direct product is unique in the sense of isomorphism. If we also exchange domain and codomain for all the arrows in the statement of the theorem, as well as in the statement of its proof, then we get another theorem: direct sum is unique in the sense of isomorphism, without re-do the proof!

As a summary, the duality in category theory furnishes free lunch, which include not only the dual concepts that are very generic, but also the dual theorems that need no proof. All about is flipping arrows.

2.3 Functor

2.3.1 Structure preserving map builds category out of objects

We need some examples of category to introduce the next core concept of category theory: functor. The first example is the category of topological spaces.

Definition 2.18. *[Category of Topological Spaces] The **category of topological spaces**, denoted as* Top*, has the collection of all topological spaces as its objects, and for each* $A, B \in \mathsf{Top}$*, the collection of all continuous maps from* A *to* B *as its morphisms from* A *to* B*.*

The next example is the categroy of groups.

Definition 2.19. *[Category of Groups] The **category of groups**, denoted as* Grp*, has the collection of all groups as its objects, and for each* $A, B \in \mathsf{Grp}$*, the collection of all homomorphisms from* A *to* B *as its morphisms from* A *to* B*.*

From these two examples, we find an almost free method to construct a category out of objects. That is, a method to assign the morphisms. This method employs the maps that preserve the structure of object as the morphisms. For example, in Top, the preserved structure is continuity, and in Grp, it is the group structure.

2.3.2 Functor is the morphism of the category of categories

Notice that the objects of a category can be anything. So, it can also be categories! To construct a category out of categories, the morphisms between two categories can be the maps that preserve the structure of category. These structure preserving maps in the category of categories are functors.

Definition 2.20. *[Functor] Given two categories* C *and* D*, a **functor*** $F: \mathsf{C} \to \mathsf{D}$ *is a map that takes*

- *for each* $A \in \mathsf{C}$*,* $F(A) \in \mathsf{D}$*, and*
- *for each* $A, B \in \mathsf{C}$ *and each* $f \in \mathsf{C}(A, B)$*,* $F(f) \in \mathsf{D}(F(A), F(B))$*,*

such that the structure of category is preserved, that is

- ***composition**: for* $\forall A, B, C \in \mathsf{C}$ *and* $f: A \to B, g: B \to C$*,* $F(f \circ g) = F(f) \circ F(g)$*,*
- ***identity**: for* $\forall A \in \mathsf{C}$*,* $F(1_A) = 1_{F(A)}$*.*

$$
\begin{tikzcd}
X \arrow[rrr] \arrow[dr, "f"] \arrow[dd, "h"'] & & & F(X) \arrow[dr, "F(f)"] \arrow[dd, "F(h)"'] & \\
& Y \arrow[rrr] \arrow[dl, "g"] & & & F(Y) \arrow[dl, "F(g)"] \\
Z \arrow[rrr] & & & F(Z) &
\end{tikzcd}
$$

Figure 2.1. Indicates the functor $F: \mathsf{C} \to \mathsf{D}$. This diagram commutes.

Functor as morphism does build a category of categories. Indeed, functors can be composed by extending the commuative diagram 2.1 to the right (figure 2.2). Identity functor is the one that maps everything in a category to itself. Finally, associativity and identity axioms can be checked directly.

$$
\begin{tikzcd}
X \arrow[rr] \arrow[dd, "h"'] \arrow[dr, "f"] & & F(X) \arrow[rr] \arrow[dd, "F(h)"'] \arrow[dr, "F(f)"] & & G(X) \arrow[dd, "G(h)"'] \arrow[dr, "G(f)"] & \\
& Y \arrow[rr] \arrow[dl, "g"] & & F(Y) \arrow[rr] \arrow[dl, "F(g)"] & & G(Y) \arrow[dl, "G(g)"] \\
Z \arrow[rr] & & F(Z) \arrow[rr] & & G(Z) &
\end{tikzcd}
$$

Figure 2.2. Extending the commuative diagram 2.1 to the right. This diagram commutes.

Imagine a category as a series of diagrams with dots and arrows between dots. The apply of a functor changes the style of the dots and arrows. This means it has become another category, but the structure, or form, of the diagrams are invariant.

2.3.3 Functor preserves the structure of category

It should be checked that functor preserves the structure of category. Comparing with the definition of category, functor has preserved composition and identity. So, we just have to prove that the axioms of category are also preserved.

Directly, for each $A \xrightarrow{f} B \xrightarrow{g} C \xrightarrow{h} D$ in C, we have

$$
\begin{aligned}
& (F(f) \circ F(g)) \circ F(h) \\
\{\text{composition of } F\} &= F(f \circ g) \circ F(h) \\
&= F((f \circ g) \circ h) \\
\{\text{associativity of } \mathsf{C}\} &= F(f \circ (g \circ h)) \\
\{\text{composition of } F\} &= F(f) \circ F(g \circ h) \\
&= F(f) \circ (F(g) \circ F(h)),
\end{aligned}
$$

so the associativity axiom is preserved. And, for each $f: A \to B$ in C, since

$$
\begin{aligned}
& F(f) \circ F(1_A) \\
\{\text{identity of } F\} &= F(f) \circ 1_{F(A)} \\
\{\text{identity of } \mathsf{D}\} &= 1_{F(B)} \circ F(f) \\
\{\text{identity of } F\} &= F(1_B) \circ F(f),
\end{aligned}
$$

the identity axiom is preserved.

2.3.4 Surjective functor may not be full

Since functor preserves the structure of category, the image of a functor should be a subcategory of the codomain of the functor. Explicitly, given a functor $F: \mathsf{C} \to \mathsf{D}$, the image $F(\mathsf{C})$ is a subcategory of D. Naturally, we can consider if the subcategory $F(\mathsf{C})$ is D itself. Naturally, you may think that $F(\mathsf{C}) = \mathsf{D}$ if for each object $A' \in \mathsf{D}$, there exists an object $A \in \mathsf{C}$ such that $F(A) = A'$. This, however, is not true.

A functor is not a map. In fact, it has two maps as components: the map on objects and that on morphisms. So, $F(\mathsf{C})$ can still be a proper subcategory of D even though the map $F: \mathrm{ob}_{\mathsf{C}} \to \mathrm{ob}_{\mathsf{D}}$ is surjective, as long as for some objects $A, B \in \mathsf{C}$, the map $F: \mathsf{C}(A, B) \to \mathsf{D}(F(A), F(B))$ is not. So, for functor, we have to distinguish two kinds of surjection. One is that the map $F: \mathrm{ob}_{\mathsf{C}} \to \mathrm{ob}_{\mathsf{D}}$ is surjective, for which we say F is **surjective on objects**. The other is that, for every $A, B \in \mathsf{C}$, the map $F: \mathsf{C}(A, B) \to \mathsf{D}(F(A), F(B))$ is surjective, for which we say F is **full**.

Surjection on objects and fullness are two independent properties of functor. We may find a functor is not surjective on objects, but is full. That is, there is object in D that is out of the image of F, but for every two objects in the image of F, say $F(A)$ and $F(B)$, the map on morphisms $F\colon \mathsf{C}(A,B) \to \mathsf{D}(F(A),F(B))$ is surjective.

2.3.5 Injective functor may not be faithful

The same discussion applies for the injection of functor. Again, you may think that $F(\mathsf{C})$ can still be multiple-to-one correspondent to C even though the map $F\colon \mathrm{ob}_{\mathsf{C}} \to \mathrm{ob}_{\mathsf{D}}$ is injective, as long as for some objects $A, B \in \mathsf{C}$, the map $F\colon \mathsf{C}(A,B) \to \mathsf{D}(F(A),F(B))$ is not. So, again, there are two kinds of injection. One is that map $F\colon \mathrm{ob}_{\mathsf{C}} \to \mathrm{ob}_{\mathsf{D}}$ is injective, for which we say F is **injective on objects**. The other is that, for every $A, B \in \mathsf{C}$, the map $F\colon \mathsf{C}(A,B) \to \mathsf{D}(F(A),F(B))$ is injective, for which we say F is **faithful**.

For the same reason, injection on objects and faithfulness are two independent properties of functor.

2.3.6 Image of functor may not be a category

The image of a group homomorphism is a subgroup of the codomain. But, this is not generally true for category. (Remind that group is the category with single object.)

What is the problem? Since functor preserves the structure of category, the composition, identity, and the axioms of associativity and identity of category are automatically fulfilled. All left is the closure of composition.

Consider a functor $F\colon \mathsf{C} \to \mathsf{D}$, the generic pattern to be examed would be $F(A) \xrightarrow{F(f)} F(B) \xrightarrow{F(g)} F(C)$ in the image $F(\mathsf{C})$. The problem is if $F(g) \circ F(f)$ still in the $F(\mathsf{C})$? By the axiom of composition, $F(g) \circ F(f) = F(g \circ f)$. So, that $F(g) \circ F(f)$ is in $F(\mathsf{C})$ is equivalent to that $g \circ f$ is in C. The later is only possible when the domain of g is exactly the codomain of f, but this may not be true. For instance, let $A \xrightarrow{f} B$ and $B' \xrightarrow{g} C$ with $F(B) = F(B')$, then $g \circ f$ does not exist at all, neither does $F(g \circ f)$. So, $F(g) \circ F(f)$ is not in $F(\mathsf{C})$, and $F(\mathsf{C})$ is not a category.

Why is it so? The key point in this analysis is $F(B) = F(B')$. It is this construction that destoried the $g \circ f$. In other words, F is not injective on objects.

Contrarily, when F does be injective on objects, then there must be an unique B that is mapped to $F(B)$. In this case, there is no choice but $A \xrightarrow{f} B \xrightarrow{g} C$. This means $g \circ f$ exists in C, so does the $F(g \circ f)$, or $F(g) \circ F(f)$, in $F(\mathsf{C})$. So, the closure of $F(\mathsf{C})$ is satisfied. Now, $F(\mathsf{C})$ becomes a category.

We conclude this section by a theorem.

Theorem 2.21. *[Functorial Image as Category] Given a functor $F\colon \mathsf{C} \to \mathsf{D}$, the image $F(\mathsf{C})$ is a category if and only if F is injective on objects.*

In the case of group, there is only one object, so a functor, or group homomorphism, is always injective on objects.

2.4 Natural Transformation

2.4.1 Natural transformation is morphism of the category of functors

As we have defined category, and as we have built a category out of categories by defining functor, we can also build a category out of functors by defining natural transformation. Precisely, given two categories C and D, a category of functors from C to D, denoted by $[\mathsf{C}, \mathsf{D}]$, has the collection of all functors from C to D as its objects. In addition, for any two functors $F, G\colon \mathsf{C} \to \mathsf{D}$, the morphism from F to G is a natural transformation.

Definition 2.22. *[Natural Transformation] Given two functors $F, G\colon \mathsf{C} \to \mathsf{D}$, a **natural transformation** $\eta\colon F \to G$ is a family of morphisms in D, $\{\eta_A\colon F(A) \to G(A) | \forall A \in \mathsf{C}\}$, such that for each $A, B \in \mathsf{C}$ and each $f\colon A \to B$, we have $G(f) \circ \eta_A = \eta_B \circ F(f)$. The η_A is called a **component** of η.*

Remark 2.23. [Component] With the aid of component, natural transformation, which is originally designed as a morphism between functors, now reduces to simply a collection of morphisms between objects (in the target category D), which has already been defined. Comparing with morphism between functors, morphsim between objects is much familiar to us. So, this definition of natural transformation with component is easy to understand and would be easy to use.

$$\begin{tikzcd}
X \arrow[dr, "f"] \arrow[dd, "h"] & & F(X) \arrow[rr, "\eta_X"] \arrow[dr, "F(f)"] \arrow[dd, "F(h)"'] & & G(X) \arrow[dr, "G(f)"] \arrow[dd, "G(h)"'] & \\
& Y \arrow[dl, "g"] & & F(Y) \arrow[rr, "\eta_Y"] \arrow[dl, "F(g)"] & & G(Y) \arrow[dl, "G(g)"] \\
Z & & F(Z) \arrow[rr, "\eta_Z"] & & G(Z) &
\end{tikzcd}$$

Figure 2.3. Indicates the natural transformation $\eta: F \to G$. The diagram commutes.

Natural transformation as morphism does build a category of functors. Indeed, natural transformation can be composed by extending the commuative diagram 2.3 to the right (in the same way as figure 2.2 for functor). Identity natural transformation has identity morphism as its component (recall that components of a natural transformation are simply morphisms in the target category). Finally, associativity and identity axioms can be checked directly.

Remember the metaphor for functor, wherein the action of a functor is like applying a style to the dots and arrows of the diagrams, the action of η is like changing the style of dots and arrows from style F to style G.

2.4.2 Natural transformation preserves the structure of functor

TODO

2.4.3 Natural isomorphism is equivalent to isomorphisms of category.

With the category of functors, we can discuss whether two functors are equivalent or not. This relates to the isomorphism between functors. Since a morphisim in this category is called a natural transformation, an isomorphism is called a natural isomorphism. Given the general definition of isomorphism, a natural transformation $\alpha: F \to G$ is **natural isomorphism** between functors F, $G: \mathsf{C} \to \mathsf{D}$, if there exists a natural transformation $\beta: G \to F$ such that $\beta \circ \alpha = 1_F$ and $\alpha \circ \beta = 1_G$. As usual, if natural isomorphism exists between F and G, then denote $F \cong G$.

This definition is quite complicated, since it is an isomorphism on $[\mathsf{C}, \mathsf{D}]$, which we are not familiar with. But, because α is a family of morphisms of category D, we can first consider a much simplier case, that is, $\alpha_A: F(A) \to G(A)$ is isomorphic. This isomorphism is of category D, instead of $[\mathsf{C}, \mathsf{D}]$. So, we guess, or hope, that, if for each $A \in \mathsf{C}$, there exists a morphism on D, β_A: $G(A) \to F(A)$, such that $\alpha_A \circ \beta_A = 1_{G(A)}$ and $\beta_A \circ a_A = 1_{F(A)}$, then the family of β_A might be the correct natural transformation β we need.

Lemma 2.24. *[Natural Isomorphism] A natural transformation α is a natural isomorphism if and only if each component of α is an isomorphism.*

Proof. The relation $\beta \circ \alpha = 1_F$ means that $\beta \circ \alpha$ acts as 1_F. What does the natural transformation 1_F acts? For each $A \in \mathsf{C}$, $(1_F)_A = 1_{F(A)}$; and the relation $F(f) \circ (1_F)_A = (1_F)_A \circ F(f)$ should hold. But, if $(1_F)_A = 1_{F(A)}$, then the relation becomes $F(f) \circ 1_{F(A)} = 1_{F(B)} \circ F(f)$, which is fulfilled on its own. So, the relation $\beta \circ \alpha = 1_F$ simply means, for each $A \in \mathsf{C}$, $\beta_A \circ \alpha_A = 1_{F(A)}$. The same statement holds for $\alpha \circ \beta = 1_G$, that is, $\alpha_A \circ \beta_A = 1_{G(A)}$. So, we find the conclusion that α is a natural isomorphism on $[\mathsf{C}, \mathsf{D}]$ if and only if for each $A \in \mathsf{C}$, α_A is an isomorphism on D. $\square$

With this lemma, a isomorphism on $[\mathsf{C},\mathsf{D}]$ now reduces to a family of isomorphisms on D, which is quite familiar to us.

2.4.4 Isomorphic objects should be viewed as one

Isomorphic topological spaces are the same. So it is for the isomorphic groups, isomorphic vector spaces, and so on. This means we should view isomorphic objects are one object. Regarding to morphisms, consider isomorphic objects X and Y, and another object Z of the same category. If X and Y are viewed as one, then there should be bijections between $\mathrm{mor}(X,Z)$ and $\mathrm{mor}(Y,Z)$, and between $\mathrm{mor}(Z,X)$ and $\mathrm{mor}(Z,Y)$.

If visualizing a category as diagrams of dots and arrows between dots, then we should pinch two isomorphic objects together. This leads to equivalent, but simplified, diagrams. The category obtained by pinching isomorphic objects as one in category C is called the **skeleton** of C.

Currently, this viewpoint is simply for intuition. Later in TODO, we will prove this fact seriously.

2.4.5 Natural isomorphism describes equivalence between categories

Given two categories C and D, how can we say they are equivalent? A natural possibility is using isomorphic functor. Precisely, there exist functors $F\colon \mathsf{C}\to\mathsf{D}$ and $G\colon\mathsf{D}\to\mathsf{C}$, such that $G\circ F = 1_{\mathsf{C}}$ and $F\circ G = 1_{\mathsf{D}}$. Even though this definition is quite natural, however, it is not true. The reason is that there exist isomorphic objects. For instance, if $G\circ F(A) = B$, which is not equal, but isomorphic, to A, then the categories can still be equivalent. This reflects our previous idea that isomorphic objects should be viewed as one. So, instead of $G\circ F(A) = A$, as $G\circ F = 1_{\mathsf{C}}$ indicates, we should say $G\circ F(A)\cong A$. By lemma 2.24, $G\circ F\cong 1_{\mathsf{C}}$ means, for each $A\in\mathsf{C}$, there exists an isomorphism $\alpha_A\colon (G\circ F)(A)\to A$, that is $(G\circ F)(A)\cong A$. This implies that, instead of $G\circ F = 1_{\mathsf{C}}$, the correct condition for equivalence should be $G\circ F\cong 1_{\mathsf{C}}$. The same, $F\circ G\cong 1_{\mathsf{D}}$.

Definition 2.25. *[Equivalent Categories] Categories C and D are* ***equivalent*** *if there exist functors $F\colon\mathsf{C}\to\mathsf{D}$ and $G\colon\mathsf{D}\to\mathsf{C}$ such that $G\circ F\cong 1_{\mathsf{C}}$ and $F\circ G\cong 1_{\mathsf{D}}$.*

Historically, functor is defined for describing natural transformation, and natural transformation, or natural isomorphism, is defined for describing equivalence between categories.

2.5 Summary

2.5.1 Category theory is built by recursion

In this chapter we first defined category. This was the unique starting point; and all the left were built by recursion. When the category was defined, the object was quite abstract and generic. It could be anything. So, it could be category itself! This implied a category of categories. Therein, the morphism, or functor, was defined as the structure preserving map. Again, when functor was defined, object could then be functor! A category of functors could be built, where the morphism was natural transformation.

So, the basic conceptions, which are category, functor, and natural transformation, were defined recursively.

Chapter 3

Representation

In this section, we discuss how to represent an object by the relationship the object has with any other objects. This forces us to restrict discussion to small categories.

Definition 3.1. *[Small Category] A category* C *is small if*

- *the collection* ob_{C} *is a set, and*
- *for each* $A, B \in \mathsf{C}$*, the collection* $\mathrm{mor}_{\mathsf{C}}(A, B)$ *is a set.*

3.1 Representable Functor and Yoneda Functor [3.1]

3.1.1 Object equals to its relations with others and with itself

Who are you, and what is your self? Your self is encoded in your relationships with others as well as with yourself. So is an object in a small category. In this section, we are to show that an object can be defined by the morphisms to (or from) this object in the category.

3.1.2 Morphisms with fixed codomain can be represented by a functor

Given the object, there will be many morphisms with this object as codomain (or domain). But, for the convenience of discussion, it would be better to use one morphism to represent them all. Precisely, consider a small category. For each $X \in \mathsf{C}$, we are to represent the set $\{\mathsf{C}(Y, X) | \forall Y \in \mathsf{C}\}$ by a map $Y \to \mathsf{C}(Y, X)$. Say, a map $\mathsf{C}(-, X): Y \to \mathsf{C}(Y, X)$. In addition, we hope that this map will preserve the structure of category, which is important when we are discussing category theory. That is, we are to define how the $\mathsf{C}(-, X)$ acts on morphisms of C, so that it can be a functor.

To figure this out, we have to claim the problem explicitly. We want to find a map from a morphism $f: Y \to Z$ in C to a map from the set $\mathsf{C}(Y, X)$ to the set $\mathsf{C}(Z, X)$. The later maps a function $\varphi: Y \to X$ to a function $\psi: Z \to X$. How can it be? By chaining the objects, we find it impossible. Indeed, φ emits from Y, but there is no arrow that emits to Y! So, we conclude that there is no such functor map from C. One possibility to solve this problem is consider the dual of C, the C^{op}, where in the arrow in f is flipped to $f: Z \to Y$. Now, we find an arrow emits to Y! And, by chaining objects, it is easy to find $\psi = \varphi \circ f$. By denoting $f^*(\varphi) := \varphi \circ f$, we have $\psi = f^*(\varphi)$. So, we guess that, for each morphism $f: Z \to Y$ in C, $\mathsf{C}(-, X)(f) := f^*$.

Remark 3.2. In the course of this reasoning, we find that making an educated guess in category theory is quite easy, since with the restriction of "types" (in programming language, a function f: $A \to B$ has types A and B), only a few possibilities are left to exam. So, we can quick reach the destination, no matter whether the ending is positive (constructed what we want) or not (found it impossible to construct). The types is extremely helpful in computer programming, so is it in category theory!

3.1. This section is based on the wonderful blogs (part I, II, and III) posted by Tai-Danae Bradley .

Next is to check if $\mathsf{C}(-,X)$ constructed as such is a functor. We need to check the composition and identity axioms of functor. Indeed, for each $C\xrightarrow{g}B\xrightarrow{f}A$ in C and each $\varphi\in\mathsf{C}(A,X)$,

$$\begin{aligned}
&\mathsf{C}(-,X)(f\circ g)(\varphi)\\
\{\text{definition of }\mathsf{C}(-,X)\} &= (f\circ g)^*(\varphi)\\
\{\text{definition of }-^*\} &= \varphi\circ(f\circ g)\\
\{\text{associativity}\} &= (\varphi\circ f)\circ g\\
\{\text{definition of }-^*\} &= f^*(\varphi)\circ g\\
&= g^*(f^*(\varphi))\\
\{\text{rewrite}\} &= (g^*\circ f^*)(\varphi)\\
\{\text{definition of }\mathsf{C}(-,X)\} &= [\mathsf{C}(-,X)(g)\circ\mathsf{C}(-,X)(f)](\varphi)
\end{aligned}$$

so the composition axiom, $\mathsf{C}(-,X)(f\circ g)=\mathsf{C}(-,X)(g)\circ\mathsf{C}(-,X)(f)$, is satisfied. (Recall that the domain of $\mathsf{C}(-,X)$ is the dual category of C. So, as figure 3.1 indicates, applying $\mathsf{C}(-,X)$ flips the direction of morphism, thus the direction of morphic composition.) And since

$$\begin{aligned}
&\mathsf{C}(-,X)(1_A)(\varphi)\\
\{\text{definition of }\mathsf{C}(-,X)\} &= (1_A)^*(\varphi)\\
\{\text{definition of }-^*\} &= \varphi\circ 1_A\\
\{\text{identity}\} &= \varphi\\
\{\text{definition of identity}\} &= 1_{\mathsf{C}(A,X)}(\varphi),
\end{aligned}$$

the identity axiom, $\mathsf{C}(-,X)(1_A)=1_{\mathsf{C}(A,X)}$ is satisfied. So, the $\mathsf{C}(-,X)\colon\mathsf{C}^{\mathrm{op}}\to\mathsf{Set}$ does be a functor, which is called the representable functor of X. (Recall that $\mathsf{C}(Y,X)$ is a set for each $Y\in\mathsf{C}$ when C is small.)

Definition 3.3. *[Representable Functor] Given a small category C. For any object $X\in\mathsf{C}$, functor $\mathsf{C}(-,X)\colon\mathsf{C}^{\mathrm{op}}\to\mathsf{Set}$ is defined by*

- *for each $Y\in\mathsf{C}$, $\mathsf{C}(-,X)(Y):=\mathsf{C}(Y,X)$, and*
- *for each $Y,Z\in\mathsf{C}$ and each $f\colon Z\to Y$, $\mathsf{C}(-,X)(f):=f^*$, where $f^*(\varphi):=\varphi\circ f$.*

This $\mathsf{C}(-,X)$ is called the ***representable functor*** *of X in C.*

$$\begin{tikzcd}
Y \arrow[r, "C(-{,}X)"] & C(Y,X) \arrow[d, "f^*"] \\
Z \arrow[u, "f"] \arrow[r, "C(-{,}X)"] & C(Z,X)
\end{tikzcd}$$

Figure 3.1. Indicates $\mathsf{C}(-,X)\colon\mathsf{C}^{\mathrm{op}}\to\mathsf{Set}$.

3.1.3 Yoneda functor connects an object to its representable functor

Our aim is to study the relation between an object and its representable functor, say between X and $\mathsf{C}(-,X)$. In category theory, we shall put the situation in the framework of category. So, to discuss this problem, we should first consider the categories they belong to. Obviously, $X\in\mathsf{C}$. And since $\mathsf{C}(-,X)\colon\mathsf{C}^{\mathrm{op}}\to\mathsf{Set}$, we have $\mathsf{C}(-,X)\in[\mathsf{C}^{\mathrm{op}},\mathsf{Set}]$, the category of functors from C^{op} to Set.

With this preparation, we consider the map from C to $[\mathsf{C}^{\mathrm{op}}, \mathsf{Set}]$, which preserves the structure of category. This map, thus, should be functorial. On objects, as we expected, the functor should send X to $\mathsf{C}(-, X)$. The question is how the functor maps on morphism. So, the problem reduces to how to construct the map from $\mathsf{C}(X, Y)$ to $\mathsf{Nat}(\mathsf{C}(-, X), \mathsf{C}(-, Y))$, the set of all natural transformations from $\mathsf{C}(-, X)$ to $\mathsf{C}(-, Y)$. That is, for each $f: X \to Y$, what is the corresponding natural transformation $\eta(f): \mathsf{C}(-, X) \to \mathsf{C}(-, Y)$?

We should consider the component of $\eta(f)$, say $\eta(f)_A$ for any object $A \in \mathsf{C}$. We have $\eta(f)_A$: $\mathsf{C}(A, X) \to \mathsf{C}(A, Y)$, which sends a morphism $\varphi: A \to X$ to one of $A \to Y$. How can it be? The only possibility that $\eta(f)_A$ can be constructed is using the $f: X \to Y$; and for keeping the "types" correct, it can only be $f \circ \varphi: A \to Y$. So, an educated guess is $\eta(f)_A = f_*$ for each $A \in \mathsf{C}$, where $f_*(\varphi) := f \circ \varphi$.

If our guess is correct, then $\eta(f)$ should be a natural transformation. Indeed, for each $g: B \to A$, figure 3.2 commutes.

$$
\begin{tikzcd}
A & C(A,X) \arrow[rrr, "f_*"] \arrow[ddd, "g^*"'] & & & C(A,Y) \arrow[ddd, "g^*"] \\
& & \varphi \in C(A,X) \arrow[r, mapsto] \arrow[d, mapsto] & f_*(\varphi) = f \circ \varphi \in C(A,Y) \arrow[d, mapsto] & \\
& & g^*(\varphi) = \varphi \circ g \in C(B,X) \arrow[r, mapsto] & f_*(\varphi \circ g) = g^*(f \circ \varphi) \in C(B,Y) & \\
B \arrow[uuu, "g"] & C(B,X) \arrow[rrr, "f_*"] & & & C(B,Y)
\end{tikzcd}
$$

Figure 3.2. Indicates that the f_* is a natural transformation from $\mathsf{C}(-, X)$ to $\mathsf{C}(-, Y)$. The inner circle indicates the element-wise map.

So, we have constructed a functor from C to $[\mathsf{C}^{\mathrm{op}}, \mathsf{Set}]$. As in the case of representable, this functor is easy to guess, since, to make "types" correct, the possibilities to exam are restricted to few!

This functor was first constructed by the Japanese mathematician Yoneda Nobuno. (Interestingly, Yoneda is also a computer scientist, supported the computer language ALGOL.) We summarize the definition as follow.

Definition 3.4. *[Yoneda functor] Given a small category* C*, functor* $\mathcal{Y}: \mathsf{C} \to [\mathsf{C}^{\mathrm{op}}, \mathsf{Set}]$ *defined by*

- *for each* $X \in \mathsf{C}$*,* $\mathcal{Y}(X) := \mathsf{C}(-, X)$*, and*
- *for each* $X, Y \in C$ *and each* $f: X \to Y$*,* $\mathcal{Y}(f): \mathsf{C}(-, X) \to \mathsf{C}(-, Y)$ *is a natural transformation with component* $\mathcal{Y}(f)_A := f_*$ *for any* $A \in \mathsf{C}$*, where* $f_*(\varphi) := f \circ \varphi$

*is called the **Yoneda functor** of* C*.*

As discussed in sections TODO, a functor can be injective and/or surjective on objects, and be full and/or faithful. Next, we check these properties for Yoneda functor.

3.1.4 On objects, Yoneda functor is injective but not surjective

Yoneda functor is injective on objects. That is, if $\mathsf{C}(-, X) = \mathsf{C}(-, Y)$, then $X = Y$. Also, Yoneda functor is not surjective on objects. That is, not all functors from C^{op} to Set are representable. For instance, the functor that maps every object in C^{op} to empty set and every morphism to identity is not representable. The codomain is very specific, while there is no specific object in C^{op} for which the functor is representable.

3.1.5 Yoneda functor is fully faithful

Now, we come to the interesting part. Is Yoneda functor full or faithful? That is to ask, for each $X, Y \in \mathsf{C}$, is the map from $\mathsf{C}(X, Y)$ to $\mathsf{Nat}(\mathsf{C}(-, X), \mathsf{C}(-, Y))$ surjective or injective?

For surjection, we mean, for each natural transformation $\eta\colon \mathsf{C}(-,X)\to\mathsf{C}(-,Y)$, there should be a morphism $f\colon X\to Y$ such that $\eta=\mathcal{Y}(f)$. Construct such f out of η is to find a morphism in $\mathsf{C}(X,Y)$. Notice $\eta_X\colon \mathsf{C}(X,X)\to\mathsf{C}(X,Y)$, so it is natural to get a morphism in $\mathsf{C}(X,Y)$ by take $\eta_X(1_X)$. So, let $f:=\eta_X(1_X)$. Next is to check if $\eta=\mathcal{Y}(f)$.

The condition we have is η is a natural transformation, so it has a family of commutative diagrams. Take the A to X in figure 3.2 and φ to 1_X, we have figure 3.3 commutes. This forces $\eta_B(g)=f\circ g=f_*(g)$ for each $B\in\mathsf{C}$ and each $g\colon B\to X$. This implies $\eta=\mathcal{Y}(f)$.

$$
\begin{tikzcd}
X & C(X,X) \arrow[rrr, "\eta_X"] \arrow[ddd, "g^*"'] & & & C(X,Y) \arrow[ddd, "g^*"] \\
 & & 1_X \arrow[r, mapsto] \arrow[d, mapsto] & f=\eta_X(1_X) \arrow[d, mapsto] & \\
 & & g \arrow[r, mapsto, dashed] & f\circ g & \\
B \arrow[uuu, "g"] & C(B,X) \arrow[rrr, "\eta_B"] & & & C(B,Y)
\end{tikzcd}
$$

Figure 3.3. The dash arrow indicates what is implied.

So, we get the conclusion that the map from $\mathsf{C}(X,Y)$ to $\mathsf{Nat}(\mathsf{C}(-,X),\mathsf{C}(-,Y))$ is surjective.

Again, for injection, we mean, for each $f,f'\colon X\to Y$ with $f\neq f'$, we should have $\mathcal{Y}(f)\neq\mathcal{Y}(f')$. Indeed, consider 1_X, $\mathcal{Y}(f)(1_X)=f\circ 1_X=f$ and $\mathcal{Y}(f')(1_X)=f'\circ 1_X=f'$. This implies $\mathcal{Y}(f)\neq\mathcal{Y}(f')$. So, we get the conclusion that the map from $\mathsf{C}(X,Y)$ to $\mathsf{Nat}(\mathsf{C}(-,X),\mathsf{C}(-,Y))$ surjective is injective.

We conclude the analysis in this section as follow.

Lemma 3.5. *Given a small category* C*, for each* $X,Y\in\mathsf{C}$*,* $\mathsf{C}(X,Y)\cong\mathsf{Nat}(\mathsf{C}(-,X),\mathsf{C}(-,Y))$.

Theorem 3.6. *Yoneda functor is fully faithful.*

This is an extraordinary conclusion. Notice that in the course of searching for a functor that connects an object to its representable functor, to make "types" correct, such a functor is one or none. We should not expect that the unique functor that can be constructed has any wonderful property on its own. But it has: being both full and faithful!