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, need 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, and *to build theory from the ground up*.[1.1] This endeavor make us think about things in the framework of category theory. 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. For each commutative diagram, the link to quiver is provided.

1.4 How to Read

While reading this book, it is strongly suggested to draw commutative diagrams whenever it is needed. You will find it quite easy if you do keep drawing commutative diagrams and quite difficult if not.

1.5 Notations

Since category theory is the math of arrows, we use arrow notation $\to$ thoroughly. This means the general notations like $\Rightarrow$ and the general expression like "natural in X" are avoided. Instead, for $A \to B$, by recognizing the "type" of A and B, we know this arrow indicates a functor if both A and B are categories, a natural transformation if both are functors, and so on.

1.1. Richard Feynman: "What I cannot create, I do not understand."

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 topological 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 algebraic 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 satisfied:

- ***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 horizontal 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*
- ***transitivity** 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 equipped with a preorder P on S.*

With these preliminaries, 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 $a \to b$.*

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 **supremum** of A if it satisfies:*

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

This is, again, a weird 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 $\forall a \in A$, $x \to a$ and,*
- *for $\forall y \in \mathbb{R}$, if $y \to a$ for $\forall a \in A$, then $y \to x$.*

Weird again, but now you may have been familiar with the weird. Hint: the word weird 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 $y \to x$ by $x \to y$). 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 morphisms 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 mathematics

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 in the set theory.

You have been familiar with the Cartesian product of two sets. Given two sets A and B, recall that the Cartesian 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 weird at the first sight.

Definition 2.14. *[Product of Two Objects] Given a category C. For any $A, B \in \mathsf{C}$, the **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 a unique morphism $\gamma: C' \to C$ so that $\alpha' = \alpha \circ \gamma$ and $\beta' = \beta \circ \gamma$.*

So, a 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 multiplication 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 product in the sense of isomorphism.

Theorem 2.16. *[Uniqueness of Product] Given a category C. For any $A, B \in \mathsf{C}$ and any two products (C, α, β) and (C', α', β'). Then, there exists a 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 product. This furnishes a new concept we called coproduct.

Definition 2.17. *[Coproduct of Two Objects] Given a category C. For any $A, B \in \mathsf{C}$, the **coproduct** 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 a unique morphism $\gamma: C \to C'$ so that $\alpha' = \gamma \circ \alpha$ and $\beta' = \gamma \circ \beta$.*

Again, a coproduct of objects A and B and a triplet (C, α, β). Comparing with product, coproduct is nothing but exchanging domain and codomain for all the arrows in the statement of 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 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: coproduct 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 category 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*** $F \colon \mathsf{C} \to \mathsf{D}$ *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* $f \colon A \to B, g \colon B \to C$*,* $F(f \circ g) = F(f) \circ F(g)$*,*
- ***identity****: for* $\forall A \in \mathsf{C}$*,* $F(1_A) = 1_{F(A)}$*.*

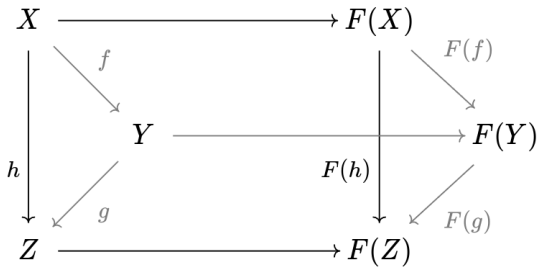

Figure 2.1. Indicates the functor $F \colon \mathsf{C} \to \mathsf{D}$. This diagram commutes.

Functor as morphism does build a category of categories. Indeed, functors can be composed by extending the commutative 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.

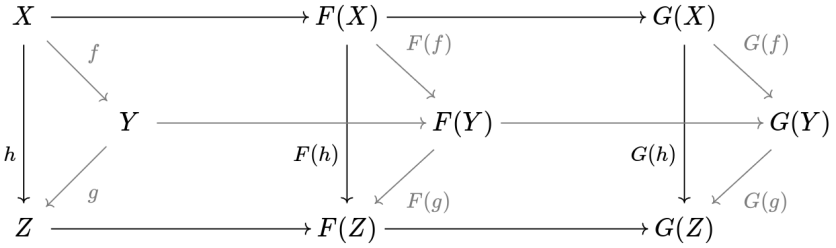

Figure 2.2. Extending the commutative diagram 2.1 to the right. This diagram commutes.

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 that 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 $F : \mathsf{C}(A, B) \to \mathsf{D}(F(A), F(B))$ is always 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 Fully faithful functor preserves isomorphisms

Fully faithful functor is found to be important because it preserves isomorphisms. To be clear, recall that notation $X \cong Y$ means there is an isomorphism between objects X and Y, we have the following lemma.

Lemma 2.21. *[Fully Faithful Functor] Given a fully faithful functor $F\colon \mathsf{C} \to \mathsf{D}$ and some $X, Y \in \mathsf{C}$, we have*

$$X \cong Y \Leftrightarrow F(X) \cong F(Y).$$

Proof. First, we are to prove that $X \cong Y \Rightarrow F(X) \cong F(Y)$. The $X \cong Y$ means there exists $\alpha\colon X \to Y$ and $\beta\colon Y \to Z$ such that $\beta \circ \alpha = 1_X$ and $\alpha \circ \beta = 1_Y$. By the composition axiom, we have $F(\beta \circ \alpha) = F(\beta) \circ F(\alpha)$. And by the identity axiom, we have $F(1_X) = 1_{F(X)}$. So, we have $F(\beta) \circ F(\alpha) = 1_{F(X)}$. The same, we get $F(\alpha) \circ F(\beta) = 1_{F(Y)}$. This simply means $F(X) \cong F(Y)$.

Then, we are to prove that $F(X) \cong F(Y) \Rightarrow X \cong Y$. The $F(X) \cong F(Y)$ means there exists $\omega\colon F(X) \to F(Y)$ and $\zeta\colon F(Y) \to F(X)$ such that $\zeta \circ \omega = 1_{F(X)}$ and $\omega \circ \zeta = 1_{F(Y)}$. Since F is full, we must have $\alpha\colon X \to Y$ and $\beta\colon Y \to X$ such that $\omega = F(\alpha)$ and $\zeta = F(\beta)$. By the composition axiom, we have $\zeta \circ \omega = F(\beta) \circ F(\alpha) = F(\beta \circ \alpha)$. By the identity axiom, $1_{F(X)} = F(1_X)$. So, we have $F(\beta \circ \alpha) = F(1_X)$. Since F is faithful, this implies $\beta \circ \alpha = 1_X$. The same, we get $\alpha \circ \beta = 1_Y$. This simply means $X \cong Y$. □

2.3.7 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 examined will be $F(A) \overset{F(f)}{\to} F(B) \overset{F(g)}{\to} F(C)$ 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 $A \overset{f}{\to} B$ and $B' \overset{g}{\to} C$ 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 destroyed 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 a unique B that is mapped to $F(B)$. In this case, there is no choice but $A \overset{f}{\to} B \overset{g}{\to} C$. 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 2.22. *[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 $F, G: \mathsf{C}\to\mathsf{D}$, the morphism from F to G is a natural transformation.

Definition 2.23. *[Natural Transformation] Given two functors $F, G: \mathsf{C}\to\mathsf{D}$, a **natural transformation** $\eta: F \to G$ is a family of morphisms in D, $\eta := \{\eta_A: F(A) \to G(A) | \forall A \in \mathsf{C}\}$, such that for each $A, B \in \mathsf{C}$ and each $f: A \to B$, we have $G(f) \circ \eta_A = \eta_B \circ F(f)$. The η_A is called a **component** of η.*

Remark 2.24. [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, morphism 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.

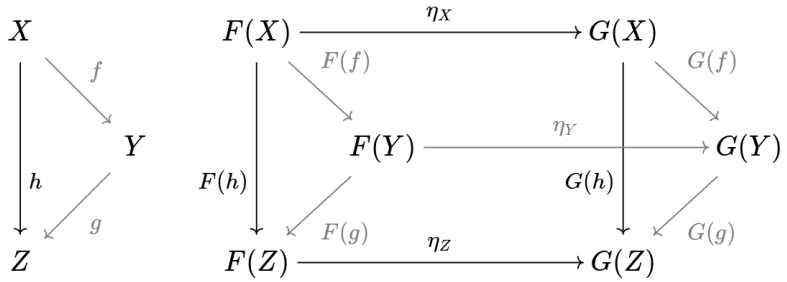

Figure 2.3. Indicates the natural transformation $\eta: F \to G$. The diagram commutes.

Natural transformation as morphism does build a category of functors. Indeed, natural transformation can be composed by extending the commutative 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.

2.4.2 Natural transformation preserves the structure of functor

TODO

2.4.3 Natural isomorphism is equivalent to isomorphisms in 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 morphism 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 in category D, we can first consider a much simpler case, that is, $\alpha_A: F(A) \to G(A)$ is isomorphic. This isomorphism is in 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 2.25. *[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. □

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

It also implies that natural isomorphism has two parts: "natural" and "ismorphism". Isomorphism means $F(X)\cong G(X)$ for each $X\in\mathsf{C}$ (recall that the $\cong$ between objects indicates an isomorphism); and naturality means figure 2.3 commutes. So, there is another way of saying that there is a natural isomorphism between F and G, that is, "$F(X)\cong G(X)$ natural in X".

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 shall 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 will 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 shall 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.

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 $F:\mathsf{C}\to\mathsf{D}$ and $G:\mathsf{D}\to\mathsf{C}$, 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 shall 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 2.25, $G\circ F\cong 1_\mathsf{C}$ means, for each $A\in\mathsf{C}$, there exists an isomorphism $\alpha_A:(G\circ F)(A)\to A$, that is $(G\circ F)(A)\cong A$. This implies that, instead of $G\circ F=1_\mathsf{C}$, the correct condition for equivalence shall be $G\circ F\cong 1_\mathsf{C}$. The same, $F\circ G\cong 1_\mathsf{D}$.

Definition 2.26. *[Equivalent Categories] Categories C and D are **equivalent** if there exist functors $F:\mathsf{C}\to\mathsf{D}$ and $G:\mathsf{D}\to\mathsf{C}$ 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 can be anything. So, it can 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 can 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.

2.5.2 Proof in category theory is easy

As you might noticed, the proof in category theory is almost nothing but expanding definitions. Once you have clearly realized what the concepts mean, you get the proof.

Why is proof in category theory so easy? An educated guess is that category theory is quite fundamental. In analysis, almost for every critical concept, there are a plenty of lemmas, theorems, and corollaries related to this concept. This is because analysis is not fundamental, and is supported by other mathematical areas, such as set theory, topology, and linear algebra (for higher dimension). So, to prove a theorem, there will be a large amount of combinations of the more fundamental lemmas, theorems, and corollaries. The proof, thus, cannot be generally easy. But, for category theory, there is no other mathematical area that supports, and the combination is quite limited. Even though most proofs in category theory are trivial, the theorems to be proven are generally quite far-reaching.

Chapter 3
Representation

3.1 Preliminary: Russell's Paradox (TODO)

3.1.1 The set of all sets implies Russell's paradox

3.1.2 Class extends set to avoid Russell's paradox

3.1.3 Category can be small or locally small

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.*

Definition 3.2. *[Locally Small Category] A category* C *is* ***locally small*** *if*

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

3.2 Hom-Functor and Yoneda Functor [3.1]

3.2.1 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 locally 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.

3.2.2 Morphisms with fixed codomain can be represented by hom-functor

Given the object, there will be many morphisms with this object as codomain (or domain). But, for the convenience of discussion, it will be better to use one morphism to represent them all. Precisely, consider a locally small category C. For each $X \in \mathsf{C}$, we are to represent the collection $\{\mathsf{C}(Y, X) | \forall Y \in \mathsf{C}\}$ 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 morphism $\varphi : Y \to X$ to a morphism $\psi : Z \to X$. How can it be? By chaining the objects by morphisms, 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 the arrow of f is flipped to $f : Z \to Y$. Now, we find an arrow emits to Y! And, by chaining objects by morphisms, 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^{op}, $\mathsf{C}(-, X)(f) := f^*$.

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

Remark 3.3. 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 $f\colon A \to B$ 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!

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 hom-functor of X. (Recall that $\mathsf{C}(Y,X)$ is a set for each $Y \in \mathsf{C}$ when C is locally small.)

Definition 3.4. *[Hom-Functor] Let C a locally small category. 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 ***hom-functor*** *of X in C.*

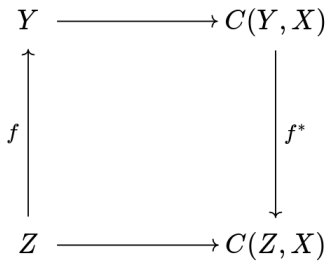

Figure 3.1. Indicates $\mathsf{C}(-,X)\colon \mathsf{C}^{\mathrm{op}} \to \mathsf{Set}$.

Remark 3.5. [Hom-] We would better to use "mor", which means "morphic", instead of, "hom", which means "homomorphic" in the word "hom-functor". But, historically, mathematicians employed homomorphism for indicating morphism. So now, the morphic functor, like $\mathsf{C}(-,X)$, is called homomorphic functor.

3.2.3 Yoneda functor connects an object to its hom-functor

Our aim is to study the relation between an object and its hom-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 shall 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, shall be functorial. On objects, as we expected, the functor shall 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 collection of all natural transformations from $\mathsf{C}(-,X)$ to $\mathsf{C}(-,Y)$. That is, for each $f\colon X \to Y$, what is the corresponding natural transformation $\eta(f)\colon \mathsf{C}(-,X) \to \mathsf{C}(-,Y)$?

We shall consider the component of $\eta(f)$, say $\eta(f)_A$ for any object $A \in \mathsf{C}$. We have $\eta(f)_A\colon \mathsf{C}(A,X) \to \mathsf{C}(A,Y)$, which sends a morphism $\varphi\colon A \to X$ to one of $A \to Y$. How can it be? The only possibility that $\eta(f)_A$ can be constructed is using the $f\colon X \to Y$; and for keeping the "types" correct, it can only be $f \circ \varphi\colon A \to Y$. 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 $g\colon B \to A$, figure 3.2 commutes.

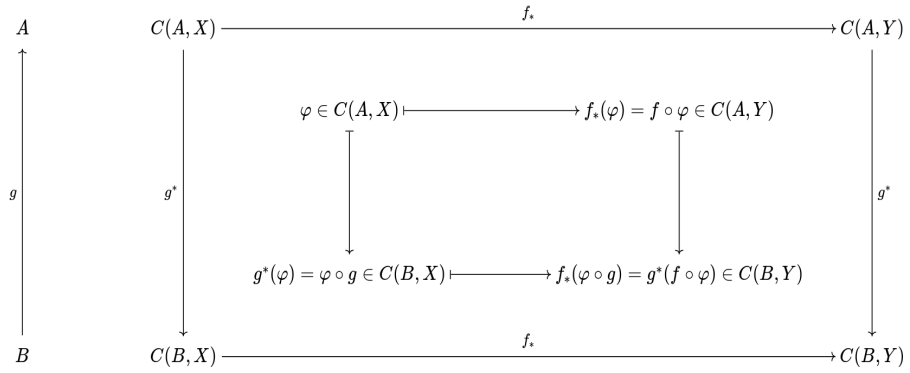

Figure 3.2. Indicates that the f_* is a natural transformation from $\mathsf{C}(-,X)$ to $\mathsf{C}(-,Y)$. The inner circle indicates the element-wise map.

So, we have constructed a functor from C to $[\mathsf{C}^{\mathrm{op}}, \mathsf{Set}]$. As in the construction of hom-functor, 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 3.6. *[Yoneda functor] Let C a locally small category. A functor $\mathcal{Y}\colon \mathsf{C} \to [\mathsf{C}^{\mathrm{op}}, \mathsf{Set}]$ defined by*

- *for each $X \in \mathsf{C}$, $\mathcal{Y}(X) := \mathsf{C}(-,X)$, and*
- *for each $X, Y \in C$ and each $f\colon X \to Y$, $\mathcal{Y}(f)\colon \mathsf{C}(-,X) \to \mathsf{C}(-,Y)$ 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** (or **Yoneda embedding**) of C.*

As discussed in sections 2.3.4 and 2.3.5, a functor can be injective and/or surjective on objects, and be full and/or faithful. Next, we check these properties for Yoneda functor.

3.2.4 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 can be represented as $\mathsf{C}(-,X)$ for some $X\in\mathsf{C}$. For instance, the functor that maps every object in C^{op} to empty set and every morphism to identity is not representable. Indeed, for any $X\in\mathsf{C}$, there is at least one element in $\mathsf{C}(X,X)$, that is the identity 1_X, thus cannot be empty. We will discuss the non-surjection further in section 3.3.

3.2.5 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 will 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 $\eta=f_*=\mathcal{Y}(f)$.

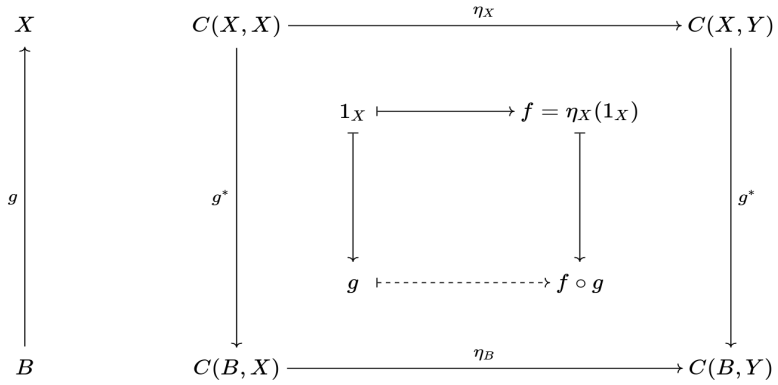

Figure 3.3. The dash arrow indicates what is implied.

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 will 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 $\mathsf{Nat}(\mathsf{C}(-,X),\mathsf{C}(-,Y))$ is injective.

We conclude the analysis in this section as follow.[3.2]

Lemma 3.9. *Let C a locally small category. We have, for each $X,Y\in\mathsf{C}$,*

$$\mathsf{C}(X,Y)\cong\mathsf{Nat}(\mathsf{C}(-,X),\mathsf{C}(-,Y)).$$

3.2. For a locally small category C, functor from C^{op} to Set is almost everywhere so that it deserves a name, presheaf.

Definition 3.7. *[Presheaf] Let C a locally small category. Functor from C^{op} to Set is called a **presheaf** on C.*

So, the category $[\mathsf{C}^{\mathrm{op}},\mathsf{Set}]$ is also called the **category of presheaves** on C. With presheaf defined, we generalize our lemma as follow.

Lemma 3.8. *[Yoneda Lemma] Let C a locally small category and F a presheaf on C. We have, for each $X\in\mathsf{C}$,*

$$F(X)\cong\mathsf{Nat}(\mathsf{C}(-,X),F).$$

This lemma is called **Yoneda lemma**. In the case $F=\mathsf{C}(-,Y)$, we go back to our lemma.

The trick for proving Yoneda lemma is the same as that for proving our lemma. For this reason, we left the proof of Yoneda lemma to reader.

Theorem 3.10. *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 hom-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!

3.2.6 All is arrow

Now, we come to the end of section 3.2. In category theory, all is arrow. This tenet indicates that an object might be precisely represented by arrows only. This should be true, otherwise the tenet would be wrong. With this vaguely destination, we stepped on the journey, constructing the essential concepts including hom-functor and Yoneda functor. In the end, we arrived at the destination by proving that Yoneda functor is injective (on objects) and fully faithful. So, indeed, all is arrow, including object.

3.3 Universal Element

3.3.1 Functor is representable if there exists universal element

By section 3.2, we know that there is a fully faithful functor (Yoneda functor) from C to $[\mathsf{C}^{\mathrm{op}}, \mathsf{Set}]$ which sends an object X to its hom-functor $\mathsf{C}(-, X)$, and that Yoneda functor is not surjective on objects, that is, not all functor are representable. It is natural to wonder the inverse problem: what is the condition for a functor $F\colon \mathsf{C}^{\mathrm{op}} \to \mathsf{Set}$ to be representable? Explicitly, is there an object $\hat{F} \in \mathsf{C}$ such that F is the hom-functor of $\hat{F}$?

Again, we shall claim the problem in the framework of category theory. Notice F is a functor, so is the $\mathsf{C}(-, \hat{F})$, then the connection between them shall be a natural transformation. So, the problem shall be claimed as what is the condition for the existence of $\hat{F} \in \mathsf{C}$ such that there is a natural isomorphism $\psi\colon \mathsf{C}(-, \hat{F}) \to F$. In this situation, we say that the presheaf (functor of "type" $\mathsf{C}^{\mathrm{op}} \to \mathsf{Set}$) F is **representable** and is **represented** by the object $\hat{F}$, which is also called the **representation** of F.

So, suppose that there exists $\hat{F} \in \mathsf{C}$ and natural isomorphism $\psi\colon \mathsf{C}(-, \hat{F}) \to F$, then figure 3.4 commutes. Interestingly, figure 3.4 also indicates that, instead of defining the e by ψ_F as $e := \psi_F(1_{\hat{F}})$,[3.3] you can conversely define the natural transformation ψ by e! Indeed, its component $\psi_X\colon \mathsf{C}(X, \hat{F}) \to F(X)$ can be defined as $\psi_X := F(-)(e)$. The naturality of ψ is an immediate result of the functoriality of F, as figure 3.5 indicates. And that ψ_X is isomorphic demands that $F(-)(e)$ should be isomorphic.

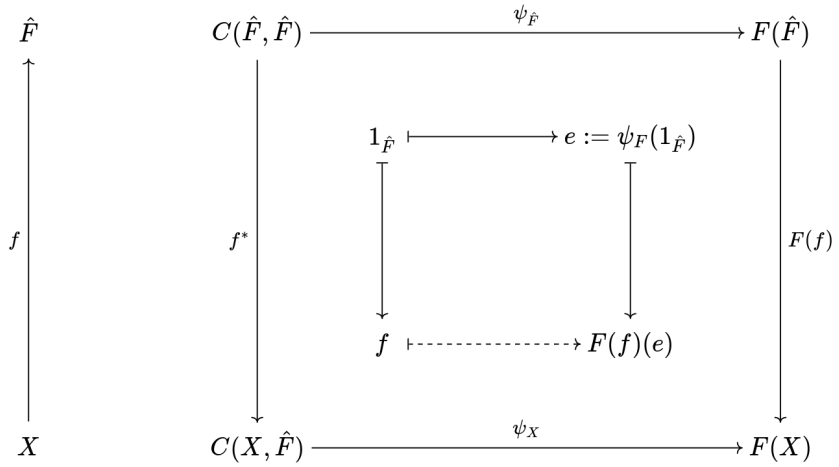

Figure 3.4. Indicates how the e is defined by ψ_F, or conversely how the ψ_X is defined by e.

3.3. We employed the notation e since it relates to identity $1_{\hat{F}}$. Recall that e is employed for identity in group theory.

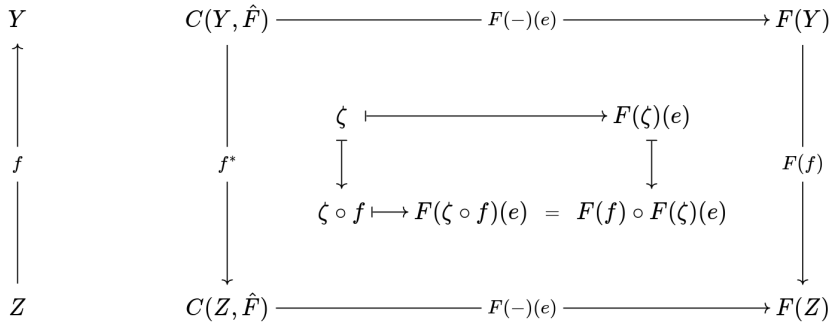

Figure 3.5. This figure proves that the naturality of ψ is an immediate result of the functoriality of F. It should be noticed that the domain of F is the dual category of C, so it should be $F(f\circ g)=F(g)\circ F(f)$.

The e in the expression $\psi_X = F(-)(e)$ holds for all $X\in\mathsf{C}$. For this reason, it is called a *universal* element, reminding that e is an element of $F(\hat{F})$ which is a set.

Definition 3.11. *[Universal Element] Given a functor $F\colon \mathsf{C}^{\mathrm{op}}\to\mathsf{Set}$, an element $e\in F(\hat{F})$, for some $\hat{F}\in\mathsf{C}$, is a **universal element** of F if $F(-)(e)\colon \mathsf{C}(X,\hat{F})\to F(X)$ is isomorphic for all $X\in\mathsf{C}$.*

The previous discussion can be summarized as follow.

Theorem 3.12. *[Universal Element]*

- *A functor $F\colon \mathsf{C}^{\mathrm{op}}\to\mathsf{Set}$ is representable if and only if there exists a universal element of F.*
- *If e is a universal element of F, then $F(-)(e)\colon \mathsf{C}(-,\hat{F})\to F$ defines a natural isomorphism.*

3.3.2 Representation is unique up to isomorphism

Yoneda functor is fully faithful. By lemma 2.21, fully faithful functor preserves isomorphisms. Explicitly, if two hom-functors $\mathsf{C}(-,X)\cong\mathsf{C}(-,Y)$, then we have $X\cong Y$, and vice versa. So, if an object $\hat{F}\in\mathsf{C}$ represents a functor $F\colon \mathsf{C}^{\mathrm{op}}\to\mathsf{Set}$, then this object is unique up to isomorphism. Otherwise, if both $\hat{F}$ and $\hat{F}'$ represent F, we should have $\mathsf{C}(-,\hat{F})\cong F\cong\mathsf{C}(-,\hat{F}')$, then $\hat{F}\cong\hat{F}'$. Thus, we have the following theorem.

Theorem 3.13. *If a presheaf F is represented by both object A and object B, then $A\cong B$.*

It is in this sense that we say the object $\hat{F}$ is *the* representation of the presheaf F.

3.3.3 Dual representation is free of charge

We have discussed the hom-functor and the possibility that a general presheaf can be isomorphic to a hom-functor, that is, representation. It is worthy of considering the dual theory of these. Since duality is free of charge, we will not tell the whole story in its dual version, but simply claim the conclusions by directly flipping arrows in the corresponding commutative diagrams. This can be seen as an exercise of flipping arrow too.

We start at the duality of hom-functor.

Definition 3.14. *[Co-Hom-Functor] Let C a locally small category. For any object $X\in\mathsf{C}$, functor $\mathsf{C}(X,\star)\colon \mathsf{C}\to\mathsf{Set}$ is defined by*

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

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

Then, we have the dual Yoneda functor.

Definition 3.15. *[Co-Yoneda functor] Let* C *a locally small category. A functor* $\mathcal{Y}^{\mathrm{op}}\colon \mathsf{C}^{\mathrm{op}} \to [\mathsf{C}, \mathsf{Set}]$ *defined by*

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

is called the ***Co-Yoneda functor*** *(or* ***Co-Yoneda embedding****) of* C.

Lemma 3.16. *Let* C *a locally small category. We have, for each* $X, Y \in \mathsf{C}$,

$$\mathsf{C}(X, Y) \cong \mathsf{Nat}(\mathsf{C}(X, -), \mathsf{C}(Y, -)).$$

Theorem 3.17. *Co-Yoneda functor is fully faithful.*

Finally, the duality of universal element relates to the representability of co-presheaf.

Definition 3.18. *[Co-Universal Element] Given a functor* $F\colon \mathsf{C} \to \mathsf{Set}$, *an element* $e \in F(\hat{F})$, *for some* $\hat{F} \in \mathsf{C}$, *is a* ***co-universal element*** *of* F *if* $F(-)(e)\colon \mathsf{C}(\hat{F}, X) \to F(X)$ *is isomorphic for all* $X \in \mathsf{C}$.

Theorem 3.19. *[Co-Universal Element] A functor* $F\colon \mathsf{C} \to \mathsf{Set}$ *is representable if and only if there exists a co-universal element of* F. *And if* e *is a co-universal element of* F, *then* $F(-)(e)\colon \mathsf{C}(\hat{F}, -) \to F$ *defines a natural isomorphism.*

3.4 Summary

3.4.1 Embedding in the framework of category theory is the right way to extend category theory

The Yoneda functor was given out by embedding the object and its hom-functor in the framework of category theory. The same goes for diagram and cone. So, in category theory, considering a concept in the framework of category is the right way to extend category theory.

3.4.2 "Types" help to restrict the possibility of construction

In programming languages, especially for the strong type languages, types are important. For instance, a function $f\colon A \to B$ has types A and B. The same goes for category theory.

While constructing the hom-functor and Yoneda functor, we find that "types" are extremely helpful for restricting the possibility of construction, almost make it unique. So, when construct something in category theory, types should be considered at the first place.

3.4.3 "Types" help check the correctness of derivation

As in the case of static programming language, "types" provides a direct way of checking the correctness of the final result after a long long derivation. It is simple but very efficient for finding errors. And it is free of charge in category theory.

Chapter 4
Limit

4.1 Cone Functor and Limit

In this section, we discuss an important application of universal element or representation: limit. We are to show that limit is a representation of a functor that generates objects called cones.

Now, the problem turns to be why cones are important. Because many important mathematical concepts are turn out to be cones. Remember in chapter 2, we have embedded some concepts in the framework of category theory, such as supremum and infimum, Cartesian product and adjoint union. These concepts, once re-written by arrows, looks weird. You may wonder where these weird stuffs come from. In fact, they come from limit.

To define limit, we have to first introduce diagram, and then cone.

4.1.1 Diagram as a part of category is a functor

When we were discussing category, functor, and natural transformation, we used the metaphor that a category is a collection of diagrams. Well, what is a diagram precisely? Naturally, a diagram is a part of a category, say C. By part, we indicates two aspects: the set of dots (objects) in a diagram is a subset of the ob_{C}, and the set of arrows (morphisms) between dots X and Y is a subset of the $\mathsf{C}(X,Y)$.

Remind that, in category theory, everything shall be described by arrows. So, how to describe a diagram by arrows?

Let us consider a simpler case: set theory, where a part simply means a finite subset. Such a subset of set S is written as $A := \{x_1,\ldots,x_n\}$, where $x_i \in S$. A set corresponds to a discrete category, and a function between sets to a functor between discrete categories. (A discrete category C is that with $\mathsf{C}(X,X) = \{1_X\}$ and $\mathsf{C}(X,Y) = \varnothing$ for each $X, Y \in \mathsf{C}$.) So, we are to describe A by function. A natural one is $f: I \to S$, where $I := \{1,\ldots,n\}$ as an "indexing set", and $f(i) := x_i$.

Back to category theory, the indexing set analogies to a small category I called indexing category, and the function f analogies to functor $F: \mathsf{I} \to \mathsf{C}$. Indexing category is used for filtering the objects and morphisms in C, so that the image is a part of C, that is, a diagram. We summarize the previous discussion as follow.

Definition 4.1. *[Diagram] Let I a small category and C a category. An I-shaped **diagram** in C is a functor $F: \mathsf{I} \to \mathsf{C}$. It is small or locally small if C is small or locally small respectively.*

4.1.2 Cone irradiates diagram

With diagram declared, we come to cone. A light cone generated by a table lamp irradiates the table. The table lamp is an object, and the table is a diagram. So, a cone irradiates a diagram.

But, how can we describe the relation between an object and a diagram, which is a functor? To make this possible, we have to make the most boring definition so far: the constant functor. Constant functor converts an object to a functor.

Definition 4.2. *[Constant Functor] Let* C *and* D *categories. For each* $X \in \mathsf{D}$*, the* ***constant functor*** *of* X*,* $\mathrm{Const}_X \colon \mathsf{C} \to \mathsf{D}$*, is defined by*

- *for each* $A \in \mathsf{C}$*,* $\mathrm{Const}_X(A) = X$*, and*
- *for each* $f \colon A \to B$*,* $\mathrm{Const}_X(f) = 1_X$*.*

With this trivial definition, we can describe the relation between an object, or its constant functor, and a diagram.

Definition 4.3. *[Cone] For each diagram* $D \colon \mathsf{I} \to \mathsf{C}$ *and each object* $X \in \mathsf{C}$*, a cone from* X *to* D *is a natural transformation* $\lambda \colon \mathrm{Const}_X \to D$*.*

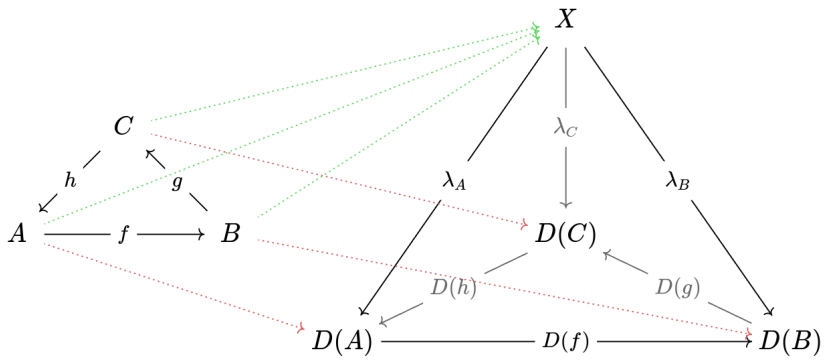

Figure 4.1. The left hand side indicates the indexing category I. And the right hand side indicates the cone from X to D. The green arrows are for functor Const_X, and the red ones for functor D, wherein maps on morphisms are not shown. As usual, identities are hidden. Since the λ is a natural transformation, the right hand side commutes.

It should be noted that, in figure 4.1, some morphism are redundant. For instance, $\lambda_A = D(h) \circ \lambda_C$ and $\lambda_B = D(f) \circ \lambda_A = D(f) \circ D(h) \circ \lambda_C$. So, when λ_C has been drawn, λ_A and λ_B are omitable, usually absent in the diagram.

4.1.3 Cone functor generates cones

Notice a cone consists two parts: the diagram and the summit object. Given a diagram $D \colon \mathsf{I} \to \mathsf{C}$ and a summit object $X \in \mathsf{C}$, the natural transformation $\lambda \colon \mathrm{Const}_X \to D$ is not unique. There can be a plenty of such natural transformations, depending on how many morphisms there are between objects in C. We can collect these natural transformations together as a set, $\mathrm{Cone}(X, D)$.

For the set of morphisms from X to Y, $\mathsf{C}(X, Y)$, we constructed a functor $\mathsf{C}(-, Y) \colon \mathsf{C}^{\mathrm{op}} \to \mathsf{Set}$, hom-functor. The same goes for $\mathrm{Cone}(X, D)$. We can construct a functor $\mathrm{Cone}(-, D) \colon \mathsf{C}^{\mathrm{op}} \to \mathsf{Set}$. Naturally, it maps $X \in \mathsf{C}$ to $\mathrm{Cone}(X, D)$, but what about the morphism, say $f \colon Y \to X$? Consider a cone $\lambda \in \mathrm{Cone}(X, D)$, which is a natural transformation from $\mathrm{Const}_X \colon \mathsf{I} \to \mathsf{C}$ to $D \colon \mathsf{I} \to \mathsf{C}$. A component $\lambda_A \colon X \to D(A)$. To construct a morphism $Y \to D(A)$ out of λ_A and f, the only possibility is $Y \xrightarrow{f} X \xrightarrow{\lambda_A} D(A)$, that is $\lambda_A \circ f$.

Definition 4.4. *[Cone Functor] For each diagram* $D \colon \mathsf{I} \to \mathsf{C}$*, the* ***cone functor*** $\mathrm{Cone}(-, D) \colon \mathsf{C}^{\mathrm{op}} \to \mathsf{Set}$ *is defined by*

- *for each* $X \in \mathsf{C}$*, mapping* X *to* $\mathrm{Cone}(X, D)$*, the set of all cones from* X *to* D*, and*
- *for each* $f \colon Y \to X$*, mapping* f *to function* $f^* \colon \mathrm{Cone}(X, D) \to \mathrm{Cone}(Y, D)$ *defined by* $f^*(\lambda_A) = \lambda_A \circ f$ *for each component* λ_A*. (See also figure 4.2.)*

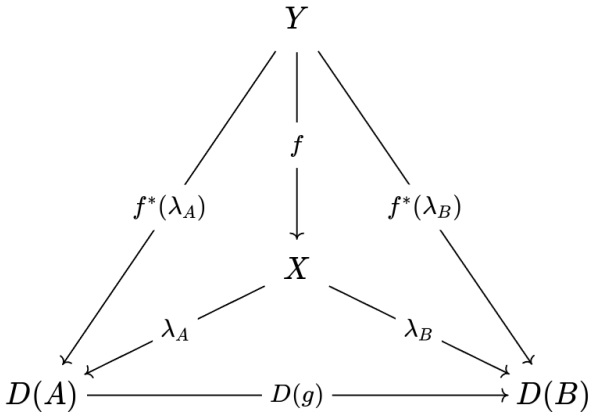

Figure 4.2. Indicates the map on f. As an instance, the indexing category I is simply $A \xrightarrow{g} B$.

4.1.4 Limit is the representation of cone functor

Given diagram $D\colon \mathsf{I} \to \mathsf{C}$ where C is locally small (or say, a locally small diagram), a cone functor $\mathrm{Cone}(-, D)\colon \mathsf{C}^{\mathrm{op}} \to \mathsf{Set}$ is a presheaf, thus may be representable. Theorem 3.12 states that a presheaf is representable if and only if there is a universal element of it. Let us restate the definition of universal element for cone functor, which is called universal cone.

A universal element of a cone functor $\mathrm{Cone}(-, D)$ is a morphism $e \in \mathrm{Cone}(\lim D, D)$ for some $\lim D \in \mathsf{C}$ such that, for each $X \in \mathsf{C}$, $\mathrm{Cone}(-, D)(e)\colon \mathsf{C}(X, \lim D) \to \mathrm{Cone}(X, D)$ is an isomorphism. First, we have to evaluate $\mathrm{Cone}(-, D)(e_A)$ for each component e_A. For each $f\colon X \to \lim D$, we have

$$\begin{aligned} & \mathrm{Cone}(f, D)(e_A) \\ \{\text{definition of } \mathrm{Cone}(-, D)\} &= f^*(e_A) \\ \{\text{definition of } (-)^*\} &= e_A \circ f. \end{aligned}$$

Next, the isomorphism $\mathrm{Cone}(-, D)(e)$ is in fact a bijection, since both sides of the arrow are sets. As a bijection, it means that for each cone $\lambda \in \mathrm{Cone}(X, D)$, there is a unique morphism $f\colon X \to \lim D$ such that $\lambda_A = e_A \circ f$ for each $A \in \mathsf{I}$. Finally, we put these together as follow.

Definition 4.5. *[Universal Cone] For a locally small diagram $D\colon \mathsf{I} \to \mathsf{C}$, a **universal cone** is a cone $e \in \mathrm{Cone}(\lim D, D)$ for some $\lim D \in \mathsf{C}$ such that, for each $X \in \mathsf{C}$ and each cone $\lambda \in \mathrm{Cone}(X, D)$, there is a unique morphism $f\colon X \to \lim D$ such that $\lambda_A = e_A \circ f$ for each $A \in \mathsf{I}$. The $\lim D$ is called the **limit** of D.*

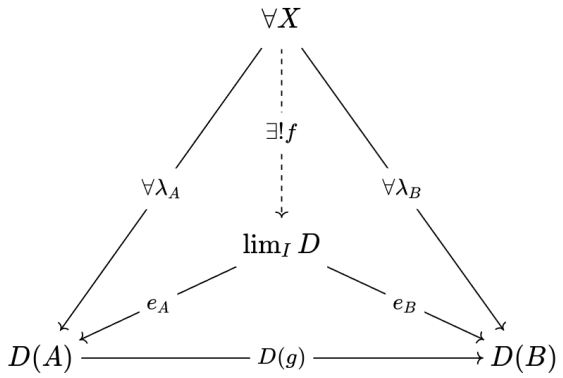

Figure 4.3. Indicates the limit. The dash arrow is for implication. As an instance, the indexing category I is simply $A \xrightarrow{g} B$.

As figure 4.3 indicates, the limit is the lowest object towards (the image of) the diagram. So, a limit is the closest object to the diagram, thus literally a limit.

The dual limit is colimit, by simply exchanging domain and codomain in limit. As figure 4.4 shows.

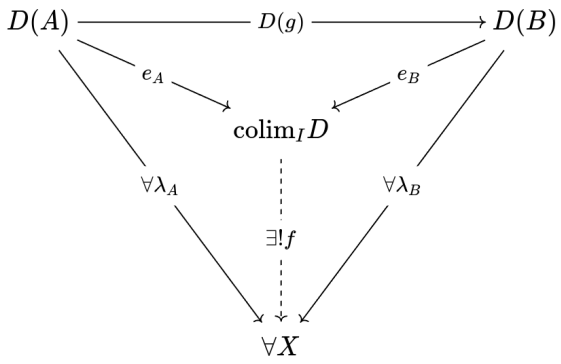

Figure 4.4. Indicates the colimit. The dash arrow is for implication. As an instance, the indexing category I is simply $A \xrightarrow{g} B$.

4.1.5 Limit is unique up to isomorphism

Recall that the representation of a functor, if exists, is unique up to isomorphism. Thus, limit and colimit, which are representation of the cone and cocone functor respectively, are unique up to isomorphism.

Theorem 4.6. *[Limit (Colimit) is Unique] The limit (colimit) of a locally small diagram is unique up to isomorphism.*

4.1.6 Infimum is a limit on poset $\mathbb{R}$

In section 2.1.4, we roughly showed what infimum would be like in the framework of category theory. Here, we make it more precise. Recalling the definition of poset (see definition 2.10), the field $\mathbb{R}$ forms a poset by preorder $\leqslant$. So, it forms a poset category Poset, with object a real number and morphism $x \to y$ if $x \leqslant y$. Consider an indexing category I as the poset $(A, \leqslant)$ where $A \subset \mathbb{R}$. Diagram $D\colon \mathsf{I} \to \mathbb{R}$ is defined by $D(x) := x$ for each $x \in \mathbb{R}$. So, $\lim D$ has the property that, for each $x \in \mathbb{R}$ and $a \in A$, if $x \leqslant D(a)$, then $x \leqslant \lim D \leqslant D(a)$. This means $\lim D$ is the infimum of the set A. So, infimum is a universal cone, or a limit, on poset category.

Since limit, or generally representation, is unique up to isomorphism, infimum and supremum of a subset of $\mathbb{R}$ is unique. (Recall that the only isomorphism in poset $\mathbb{R}$ is $x \leqslant x$ for each $x \in \mathbb{R}$.)

The same goes for its dual, supremum. If we flip the arrows in the definitions of limit, we get the dual, colimit . So, directly, we find supremum is the colimit on Poset.

4.1.7 Product is a limit with discrete indexing category

In section 2.2.1, we showed what Cartesian product would be like in the framework of category theory. Let C a category. For any $A, B \in \mathsf{C}$, the product of A and B is another object $C \in \mathsf{C}$ together with two morphisms $\alpha\colon C \to A$ and $\beta\colon C \to B$ such that, for any $C' \in \mathsf{C}$, any $\alpha'\colon C' \to A$ and $\beta'\colon C' \to B$, there exists a unique morphism $\gamma\colon C' \to C$ so that $\alpha' = \alpha \circ \gamma$ and $\beta' = \beta \circ \gamma$. If we convert this statement to commutative diagram (figure 4.5), then it is apparent that product is a limit, in which the discrete category with two elements is the indexing category.

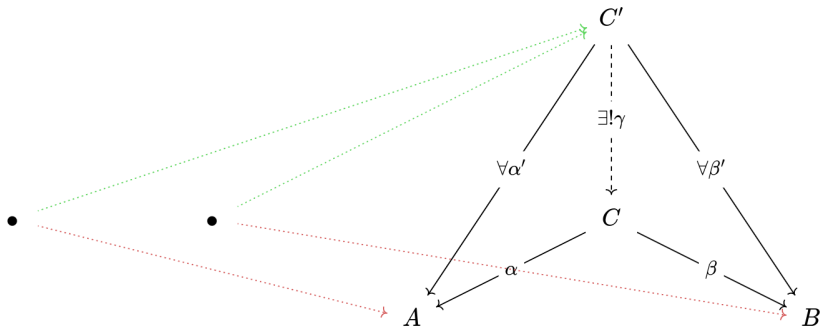

Figure 4.5. Indicates that product is a limit. The left hand side is the indexing category. There is no morphism (except the hidden identities) since it is a discrete category. The name of the object is irrelevant, so the two objects are shown as dots. The right hand side is the commutative diagram for the definition of product. The green arrows is for the constant functor, and red arrows for the diagram functor.

Again, since limit is unique up to isomorphism, product is unique up to isomorphism. Recall that we have proved the uniqueness before. Now, the proof is apparent on its own.

The same goes for coproduct (definition 2.17).

4.2 Construction of Limit

4.2.1 Limit helps generalize concepts from set theory to category theory

Mathematical concepts are built on the base of set theory, including concepts in analysis, topology, and even algebra. But, generalizing a concept in set theory by arrows, as shown by generalizing Cartesian product in section 2.2.1, is far from straight forward. At the first sight, it seems that Cartesian product of sets and its generalization, product of objects, have no relationship. So, you may wonder, how can we get some hint for generalizing Cartesian product, or even more complicated concepts in set theory, to its categorical version.

We are to deal with this very problem by considering some instances. A proper instance is the fibered product of sets [4.1]. A **fiber product** of two sets A and B is another set denoted by $A \times_{(\zeta,\eta)} B$, where A, and B are sets, $\zeta: A \to C$ and $\eta: B \to C$ are functions for some set C. Precisely,

$$A \times_{(\zeta,\eta)} B := \{(a,b) | a \in A, b \in B, \zeta(a) = \eta(b)\}.$$

Now, we are to represent this by arrows. As we have been familiar with, for a locally small category C, the collection of morphisms, say $\mathsf{C}(X,Y)$ for some $X, Y \in \mathsf{C}$, is a set. So, we can consider the fibered product of the sets of morphisms. Materials for constructing the categorical version of fibered product should be restricted, as far as posssible, to those appeared in the definition of fibered product, so that the categorical version can go back to fibered product when C is Set. So, we start constructing the sets of morphisms by using A and B, which is nothing but the $\mathsf{C}(A,B)$. This, however, will not work, since nothing can be done with only one set. The smallest extension is considering $\mathsf{C}(-,A)$ and $\mathsf{C}(-,B)$, which means $\mathsf{C}(X,A)$ and $\mathsf{C}(X,B)$ for arbitrary object $X \in \mathsf{C}$. Exploration of the other possibility, that is $\mathsf{C}(A,-)$ and $\mathsf{C}(B,-)$, is left to reader.[4.2]

So, we replace the sets in the definition of fibered product by $A \to \mathsf{C}(-,A)$, $B \to \mathsf{C}(-,B)$, and in addition, $C \to \mathsf{C}(-,C)$. What about the functions ζ and η? We have to construct a natural transformation $\hat{\zeta}: \mathsf{C}(-,A) \to \mathsf{C}(-,C)$ for replacing $\zeta: A \to C$. Because of Yoneda functor, we have $\hat{\zeta} := \zeta_*$. The same, we replace η by η_*. Thus, the $A \times_{(\zeta,\eta)} B$ is replaced to be, for any $X \in \mathsf{C}$,

$$\mathsf{C}(X,A) \times_{(\zeta_*,\eta_*)} \mathsf{C}(X,B) = \{(f,g) | f \in \mathsf{C}(X,A), g \in \mathsf{C}(X,B), \zeta_*(f) = \eta_*(g)\}.$$

4.1. We can consider Cartesian product of sets. But, it turns out to be quite trivial. The fiber product, however, is non-trivial but still easy to compute.

4.2. It turns out to be the dual.

The right hand side is an expression of arrows, which is what we want. It simply means the commutative diagram as follow.

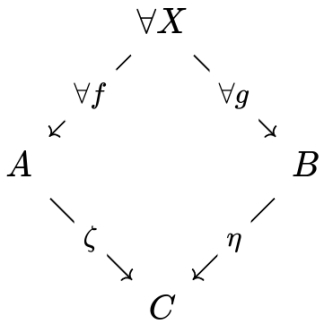

Figure 4.6. This figure indicates the fibered product $\mathsf{C}(X,A)\times_{(\zeta_*,\eta_*)}\mathsf{C}(X,B)$.

This hints for a cone with the "base" $A\xrightarrow{\zeta}C\xleftarrow{\eta}B$, or with the indexing category like $\bullet\to\bullet\leftarrow\bullet$. From this diagram, we build cone, and from the cone, we get the limit, called pullback.

Definition 4.7. *[Pullback] Let C a category. An object in C, denoted by $A\times_{(\zeta,\eta)}B$, together with $A,B\in\mathsf{C}$ and $\zeta\colon A\to C,\eta\colon B\to C$ is called a **pullback** if for each $X\in\mathsf{C}$, there exists $\gamma\colon X\to A\times_{(\zeta,\eta)}B$ such that figure 4.7 commutes.*

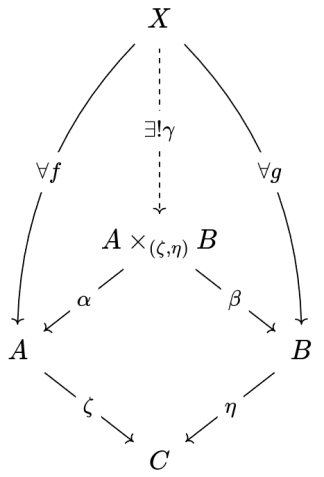

Figure 4.7. Indicates the pullback $A\times_{(\zeta,\eta)}B$. Principaly, there should be a morphism from $A\times_{(\zeta,\eta)}B$ to C (and also from X to C). But, since the figure commutes, such a morphism can be represented by $\zeta\circ\alpha$ or by $\eta\circ\beta$, thus is redundant and omitted.

We are left to check that pullback is indeed the fibered product in the case of Set. This is left to reader. The only trick for proving this is using the uniqueness of representation.

We summarize the previous steps as follow [4.3].

1. Write down the expression of the concept in terms of sets and functions.

4.3. There is a theorem that relates to these steps.

Theorem 4.8. *Let C a locally small category. For each diagram $D\colon\mathsf{I}\to\mathsf{C}$ and for each $X\in\mathsf{C}$, we have*

$$\lim\mathsf{C}(X,D(-))\cong\mathsf{C}(X,\lim D),$$

and (the dual)

$$\lim\mathsf{C}(D(-),X)\cong\mathsf{C}(\operatorname{colim}D,X).$$

But, I cannot figure out what is the explicit relationship to these steps.

2. In the expression of the concept:
   - replace the sets by the pattern $A \to \mathsf{C}(-, A)$, where the left hand side is some set A while, in the right hand side, A is an generic object in the locally small category C; and
   - replace the functions by Yoneda functor.
3. Draw the commutative diagram for the replaced expression, from which the "base" of cone can be read out.
4. Write down the limit implied by the cone; and then check that this limit will reduce to the concept in set theory when C is Set.

This series of steps is generic. By following these steps, we can "categorify" equalizer in set theory [4.4].

4.4. In set theory, given two sets A, B and two functions $u, v: A \to B$, the equalizer of u and v is defined as the subset of A

$$\mathrm{eq}(u, v) := \{a \,|\, a \in A, u(a) = v(a)\}.$$

Following the same strategy, we arrive at

$$\mathrm{eq}(u_*, v_*) = \{f \,|\, f \in \mathsf{C}(X, A), u \circ f = v \circ f\}.$$

Again, the right hand side is an expression of arrows, which hints for a cone with the "base" $A \overset{u}{\underset{v}{\rightrightarrows}} B$, or with the indexing category like $\bullet \rightrightarrows \bullet$, as figure 4.8 shows.

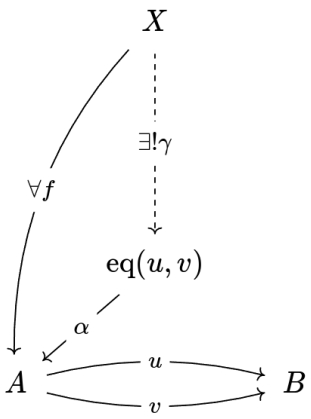

Figure 4.8. Indicates the equalizer $\mathrm{eq}(u, v)$. Principaly, there should be a morphism from $\mathrm{eq}(u, v)$ to B (and also from X to B). They are omitted for the same reason in figure 4.7.

The final step is to prove that this is indeed the equalizer in the case of Set. Now, by following the same steps, we can generalize equalizer from set theory to its categorical version. Details are left to reader.