Part II

Representable Functors

Arbitrary categories are hard to deal with. The goal of Yoneda lemma and representable functors is to utilize the nice properties of the category of sets to study locally small categories. So, instead of studying the categories themselves we study the functor category to sets which have nicer properties. For details see [1].

1 | Preliminary Definitions

A set is a collection of 'elements'. A category $\mathcal{C}$ is more sophisticated, it possesses 'objects' similar to how sets posses elements, but for each pair of objects, X and Y in $\mathcal{C}$, there is a set of relations between X and Y, called morphisms, denoted by $\mathrm{Hom}_{\mathcal{C}}(X, Y)$. The Yoneda Lemma allows us to define an object by its relations to other objects.

A functor F between two categories $\mathcal{C}$ and $\mathcal{D}$ consists of a mapping of objects of $\mathcal{C}$ to objects of $\mathcal{D}$, $X \mapsto FX$ together with a map of the set of homomorphisms,

$$\mathrm{Hom}_{\mathcal{C}}(X, Y) \to \mathrm{Hom}_{\mathcal{D}}(FX, FY).$$

the image of $f \in \mathrm{Hom}_{\mathcal{C}}(X, Y)$ denoted by $F(f)$. That takes identity to identity and respects composition i.e.,

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

They are called covariant functors. A contravariant functor is a functor from the opposite category, and hence should satisfy,

$$F(f \circ g) = F(g) \circ F(f).$$

Whenever we say functor, we assume it to be covariant functor. A contravariant functor from $\mathcal{C}$ to $\mathcal{D}$ can be thought of as a covariant functor from $\mathcal{C}^{\mathrm{op}}$ to $\mathcal{D}$. A functor F is faithful if the map $\mathrm{Hom}_{\mathcal{C}}(X, Y) \to \mathrm{Hom}_{\mathcal{D}}(FX, FY)$ is injective for all X, Y. It's full if the map is surjective. If it's a bijection the functor is called fully faithful.

A natural transformation κ between two functors F and G from category $\mathcal{C}$ to $\mathcal{D}$ is a collection of mappings κ_X for every $X \in \mathcal{C}$, such that for all $f : X \to Y$, the diagram,

$$\begin{tikzcd} X \arrow[d, "f"] & FX \arrow[r, "\kappa_X"] \arrow[d, "F(f)"'] & GX \arrow[d, "G(f)"] \\ Y & FY \arrow[r, "\kappa_Y"'] & GY \end{tikzcd} \qquad \text{(natural transformation)}$$

commutes, i.e., it respects the new objects and morphisms and satisfies the composition law,

$$(\kappa \circ \varphi)_X = \kappa_X \circ \varphi_X$$

The collection of all natural transformation between two functors F and G is denoted by,

$$\mathrm{Nat}(F, G).$$

We say two functors F and G are isomorphic or naturally equivalent if the natural transformation between them is a natural isomorphism, denoted as, $F \cong G$. The collections of all functors from $\mathcal{C}$ to $\mathcal{D}$ together with the natural transformations as the morphisms between functors is a category, denoted by $\mathcal{D}^{\mathcal{C}}$. The nice thing about functor category $\mathcal{D}^{\mathcal{C}}$ is that if $\mathcal{D}$ has some nice properties then $\mathcal{D}^{\mathcal{C}}$ inherits these useful properties.

1.1 | Equivalence of Categories

Two objects X and Y in a category $\mathcal{C}$ are isomorphic if there exist morphisms $f : X \to Y$ and $g : Y \to X$ such that $f \circ g = \mathbb{1}_Y$ and $g \circ f = \mathbb{1}_X$. In such a case, both the objects carry the same information, so these objects are equivalent. We want a similar equivalence between categories. This allows us to study a new category using some already well understood category. Equivalence of two categories can be thought of as giving two complementary description of same matheamtical object. We can compare two categories $\mathcal{C}$ and $\mathcal{D}$ via the functors between them. The starting point is the functor category $\mathcal{D}^{\mathcal{C}}$.

A functor $F : \mathcal{C} \to \mathcal{D}$ is an equivalence of categories if there is a functor $G : \mathcal{D} \to \mathcal{C}$ such that

$$GF \cong \mathbb{1}_{\mathcal{C}}, \quad \text{and} \quad FG \cong \mathbb{1}_{\mathcal{D}},$$

where the identity functor $\mathbb{1}_{\mathcal{C}}$ sends objects of $\mathcal{C}$ to the same objects, and morphisms to the same morphisms. G is called quasi-inverse functor. In such a case, $\mathcal{C}$ and $\mathcal{D}$ are said to be equivalent.

Lemma 1.1. *$F : \mathcal{C} \to \mathcal{D}$ is an equivalence of categories iff F is fully faithful and for every object $Y \in \mathcal{D}$ there exists an object X such that FX is isomorphic to Y.*

Proof

$\Rightarrow$ Suppose F is an equivalence of categories then there exists a functor $G : \mathcal{D} \to \mathcal{C}$ such that $FG \cong \mathbb{1}_{\mathcal{D}}$ and $GF \cong \mathbb{1}_{\mathcal{C}}$. So, by this, there exists for each $X \in \mathcal{C}$ and $Y \in \mathcal{D}$, isomorphisms,

$$\varphi_X : GFX \to X, \quad \varkappa_Y : FGY \to Y.$$

So, each object $Y \in \mathcal{D}$ is isomorphic to the object FX where $X = GY$. To show that this is fully faithful, we have to show it gives homset isomorphism. Let $f \in \mathrm{Hom}_{\mathcal{C}}(X, X')$ then we have the following diagram which commutes,

$$\begin{tikzcd} GFX \arrow[r, "\varphi_X"] \arrow[d, "GF(f)"'] & X \arrow[d, "f"] \\ GFX' \arrow[r, "\varphi_{X'}"] & X' \end{tikzcd}$$

note here that φ_X is invertible. Hence we can construct f as,

$$f = \varphi_{X'} \circ GF(f) \circ \varphi_X^{-1}$$

So, each f can be constructed from $F(f)$. Given any map $g \in \mathrm{Hom}_{\mathcal{D}}(FX, FX')$, set,

$$f = \varphi_{X'} \circ G(g) \circ \varphi_X^{-1} \in \mathrm{Hom}_{\mathcal{C}}(X, X').$$

So we have $G(g) = GF(f)$ and this gives us a hom set isomorphism or that F is fully faithful.

$\Leftarrow$ Assuming to each $Y \in \mathcal{D}$ there corresponds $X_Y \in \mathcal{C}$ such that there exists an isomorphism $\varkappa_Y : FX_Y \to Y$. We have to construct a quasi-inverse functor using these isomorphisms. Set $GY = X_Y$, and for each morphism $g \in \mathrm{Hom}_{\mathcal{D}}(Y, Y')$, set,

$$G(g) = \varkappa_{Y'}^{-1} \circ g \circ \varkappa_Y$$

then we have $G(g) \in \mathrm{Hom}_{\mathcal{D}}(FGY, FGY')$ which is same as $\mathrm{Hom}_{\mathcal{C}}(GY, GY')$ because we had assumed F is fully faithful, i.e., the hom sets are isomorphic. It's easyto check that G is a functor and is a quasi-inverse to F. $\square$

Let $\mathcal{C}$ be a category, and let for any pair of objects X, Y of $\mathcal{C}$ an equivalence relation $\sim_{X,Y}$ in $\mathrm{Hom}_{\mathcal{C}}(X, Y)$ be given then, we can define a new category, called the quotient category $\mathcal{D} = \mathcal{C}/\sim$, and the quotient functor $Q : \mathcal{C} \to \mathcal{D}$ such that,

$$f \sim f' \implies Qf = Qf'$$

and every functor $F : \mathcal{C} \to \mathcal{D}'$, with $Ff = Ff'$ whenever $f \sim f'$, factors through $\mathcal{D}$, i.e., there exists a unique functor $G : \mathcal{D} \to \mathcal{D}'$ such that $F = G \circ Q$. Note that in this category $\mathcal{D}$, the objects are the same, but the hom sets are reduced, by the equivalence relation.

$$\mathrm{Hom}_{\mathcal{D}}(X, Y) = \mathrm{Hom}_{\mathcal{C}}(X, Y)/\sim_{X,Y}.$$

2 | Representable Functors

Many of the definitions and properties of algebraic objects can be expressed in categorical language. Representable functor define new properties using functors we understand well. Definitions are simpler to study and they inherit many interesting properties from nicely behaved categories such as the category of sets.

Each $\mathrm{Hom}_{\mathcal{C}}(X, Y)$ tells us about all the relations the object X has with other object Y. The thing we should be studying is the functor $h_X = \mathrm{Hom}_{\mathcal{C}}(X, -)$ and $h^X = \mathrm{Hom}_{\mathcal{C}}(-, X)$. These are called hom functors.

$$\begin{aligned} h^X :\ & \mathcal{C}^{\mathrm{op}} \to \mathbf{Sets} \\ & Y \mapsto \mathrm{Hom}_{\mathcal{C}}(Y, X). \end{aligned}$$

which maps each morphism $f : Y \to Z$ to a morphism of hom sets given by the composition,

$$Y \xrightarrow{f} Z \xrightarrow{g} X$$

We will denote this by,

$$\begin{aligned} h^X(f) :\ & h^X(Y) \to h^X(Z) \\ & g \mapsto g \circ f. \end{aligned}$$

Similarly, we can define the contravariant hom functor. Note that we are assuming here that $\mathrm{Hom}_{\mathcal{C}}(Y, X)$s are all sets. Such categories are called locally small categories.

A contravariant functor $F : \mathcal{C}^{\mathrm{op}} \to \mathbf{Sets}$ is called representable if for some $X \in \mathcal{C}$,

$$F \cong h^X$$

in such a case, F is said to be represented by the object X. We are especially interested in contravariant functors because they correspond to pre-sheaves. For covariant functors, $G : \mathcal{C} \to \mathbf{Sets}$, this will be $G \cong h_X$. Where $\cong$ stands for natural isomorphism.

2.1 | Yoneda Embedding

Yoneda embedding and representable functors allow us to use the nice properties (ability to take limits) of the category of sets to study more complex categories that are not so nice. We want to study the objects in terms of the maps to or from the object. This information is contained in the functors $\mathrm{Hom}_{\mathcal{C}}(-, X)$ and $\mathrm{Hom}_{\mathcal{C}}(X, -)$. Yoneda lemma establishes a connection between objects $X \in \mathcal{C}$ and the functor $h^X \in \mathbf{Sets}^{\mathcal{C}^{\mathrm{op}}}$.

Theorem 2.1. (Yoneda Lemma) *For a functor $F : \mathcal{C}^{\mathrm{op}} \to \mathbf{Sets}$ and any $X \in \mathcal{C}$, there is a natural bijection,*

$$\mathrm{Nat}(h^X, F) \cong FX \tag{Strong Yoneda}$$

such that $\kappa \in \mathrm{Nat}(h^X, F) \leftrightarrow \kappa_X(\mathbb{1}_X) \in FX$.

Proof

In the natural transformation diagram, replace F by h^X, and G by F. $\kappa_X : h^X X \to FX$. Now, $h^X X = \mathrm{Hom}_{\mathcal{C}}(X, X)$, which contains $\mathbb{1}_X$. Using this we construct a map,

$$\begin{aligned} \mu : \mathrm{Nat}(h^X, F) &\to FX \\ \kappa &\mapsto \kappa_X(\mathbb{1}_X). \end{aligned}$$

We have to now check that this is a bijection. We show this by showing κ is determined by $\mu(\kappa)$ for all $Y \in \mathcal{C}$. For any $f : Y \to X$, we have,

$$\begin{tikzcd} X & h^X X \arrow[r, "\kappa_X"] \arrow[d, "h^X(f)"'] & FX \arrow[d, "F(f)"] & \mathbb{1}_X \arrow[r, mapsto, "\kappa_X"] \arrow[d, mapsto] & \mu(\kappa) \arrow[d, mapsto] \\ Y \arrow[u, "f"] & h^X Y \arrow[r, "\kappa_Y"'] & FY & f \arrow[r, mapsto, "\kappa_Y"'] & \kappa_Y(f) \end{tikzcd}$$

Hence $\kappa_Y(f) = F(f)(\mu(\kappa))$, or the action of κ_Y is determined by $\mu(\kappa)$. So, if $\mu(\kappa) = \mu(\varphi)$ then $\kappa_Y(f) = \varphi_Y(f)$ for all $Y \in \mathcal{C}$, so it's injective.

For surjectivity we have to show that for all sets $x \in FX$, there exists a natural transformation φ such that $\varphi_X(\mathbb{1}_X) = x$. For $x \in FX$, and $f : Y \to X$, construct the map,

$$\begin{aligned} \varphi : h^X &\to F \\ f &\mapsto F(f)(x). \end{aligned}$$

this satisfies the requirement that $\varphi_X(\mathbb{1}_X) = x$, because clearly, $\mathbb{1}_X \mapsto F(\mathbb{1}_X)(x) = \mathbb{1}_x(x) = x$. We must make sure it's indeed a natural transformation, i.e., check if the naturality diagram,

$$\begin{tikzcd} Y & h^X Y \arrow[r, "\varphi_Y"] \arrow[d, "h^X(g)"'] & FY \arrow[d, "F(g)"] \\ Z \arrow[u, "g"] & h^X Z \arrow[r, "\varphi_Z"'] & FZ \end{tikzcd}$$

commutes for all $Y, Z \in \mathcal{C}$, $g \in \mathrm{Hom}_{\mathcal{C}}(Z, Y)$. For $f : Y \to X$, by definition of φ,

$$F(g) \circ (\varphi_Y(f)) = F(g) \circ F(f)(x)$$

which by functoriality of F is $= F(f \circ g)(x)$. On the other hand, by definition of the hom functor, we have,

$$\varphi_Z \circ (h_X(g)(f)) = \varphi_Z(h_X(f \circ g))$$

which again by definition of φ is $= F(f \circ g)(x)$. Hence the diagram commutes, and φ is a natural transformation. The map $\mu : \mathrm{Nat}(h^X, F) \to FX$ is a bijection. $\square$

So, the information about objects is contained in their associated hom functors, for locally small categories. The proof covariant version is exactly the same, just have to reverse the arrows on the category $\mathcal{C}$. The Yoneda lemma gives us an embedding of the category $\mathcal{C}$ inside the functor category $\mathbf{Sets}^{\mathcal{C}^{\mathrm{op}}}$, given by,

$$X \mapsto h^X.$$

This embedding is called the Yoneda embedding $h^{(-)} : \mathcal{C} \to \mathbf{Sets}^{\mathcal{C}^{\mathrm{op}}}$, which sends an object $X \in \mathcal{C}$ to the sets of morphisms $\mathrm{Hom}_{\mathcal{C}}(-, X)$. These functors are fully faithful by Yoneda lemma, because by replacing the functor F by h^Y we have,

$$\mathrm{Nat}(h^X, h^Y) \cong h^Y(X) = \mathrm{Hom}_{\mathcal{C}}(X, Y). \tag{Weak Yoneda}$$

Similarly for the covariant embedding, in which case this will be $\mathrm{Nat}(h_X, h_Y) \cong \mathrm{Hom}_{\mathcal{C}}(Y, X)$.

Given a contravariant functor, $F : \mathcal{C}^{\mathrm{op}} \to \mathbf{Sets}$, the Strong Yoneda tells us that we can think of the action of F on the element X as natural transformations to the hom functor h^X in the functor category. So, every functor F can extended and be thought of as a representable functor,

$$\begin{aligned} h^F : (\mathbf{Sets}^{\mathcal{C}^{\mathrm{op}}})^{\mathrm{op}} &\to \mathbf{Sets} \\ G &\mapsto \mathrm{Nat}(G, F) \end{aligned}$$

where elements $X \in \mathcal{C}$ are to be interpreted as the elements $h^X \in \mathbf{Sets}^{\mathcal{C}^{\mathrm{op}}}$.

Note that, Strong Yoneda associates to each set in FX a natural transformation between h^X and F. If the functor F is representable, i.e., there exists $Y \in \mathcal{C}$ such that there exists a natural isomorphism,

$$F \xrightarrow[\alpha]{\cong} h^Y$$

Let $\mu(\alpha)$ be the corresponding element in $FY = \mathrm{Hom}_{\mathcal{C}}(Y, Y)$. The pair $(Y, \mu(\alpha))$ is called a universal object for F. It's such that for any other object $Z \in \mathcal{C}$, and each $g \in FX = \mathrm{Hom}_{\mathcal{C}}(X, Y)$ there exists a unique morphism $f : X \to Y$ such that,

$$Ff(\mu(\alpha)) = g.$$

2.2 | Limits & Colimits

The notion of limits and colimits is very important as they allow us to construct new objects and functors. Let $\mathcal{I}$ and $\mathcal{C}$ be two categories. An inductive system in $\mathcal{C}$ indexed by $\mathcal{I}$ is a functor,

$$F : \mathcal{I} \to \mathcal{C}.$$

The limit of a system is an object in $\mathcal{C}$ that is 'closest' to the system.

This can be formalised in the functor category as follows; Attach to each object $X \in \mathcal{C}$ the constant functor $c_X : \mathcal{I} \to \mathcal{C}$ that sends everything in $\mathcal{I}$ to X, and each morphism in $\mathcal{I}$ to the identity on X. A relation between an object X and the system F is a natural transformation between F and c_X. Such a natural transformation is called a cone. The collection of all such cones is the set of all naural transformations,

$$\begin{aligned} C_F : \mathcal{C}^{\text{op}} &\to \textbf{Sets} \\ X &\mapsto \text{Hom}_{\mathcal{C}^{\mathcal{I}}}(F, c_X). \end{aligned}$$

It's a contravariant functor from $\mathcal{C}$ to **Sets**. If the functor C_F is representable, there exists an object L such that,

$$C_F \cong h^L$$

So, in such case $C_F(X) \cong \text{Hom}_{\mathcal{C}}(X, L)$. In such a case we have, $\text{Hom}_{\mathcal{C}^{\mathcal{I}}}(F, c_X) \cong \text{Hom}_{\mathcal{C}}(X, L)$ and hence every cone must factor through L.[1] The representative L is called the limit of the inductive system, and is denoted by $\varinjlim F$.

A projective system in $\mathcal{C}$ indexed by $\mathcal{I}$ is a functor from $G : \mathcal{I}^{\text{op}} \to \mathcal{C}$. Similar to the inductive system, for projective system $G : \mathcal{I}^{\text{op}} \to \mathcal{C}$, we study the collection of cocones, i.e.,

$$\begin{aligned} C^G : \mathcal{C} &\to \textbf{Sets} \\ X &\mapsto \text{Hom}_{\mathcal{C}^{\mathcal{I}^{\text{op}}}}(c_X, G). \end{aligned}$$

If it's representable with representative M, $C^G(X) \cong \text{Hom}_{\mathcal{C}}(M, X)$. The limit is denoted by $\varprojlim G$.

For the case of locally small categories, we can define limits using limits in the category of sets, making it easier to work with. For **Sets**, we can define the limit in terms of the initial/terminal object,

$$\varprojlim F \coloneqq \text{Cone}(1, F) = \text{Nat}(1, F)$$

here the initial/terminal object is the set with one element. This is set, as we assumed the indexing category is small, i.e., the hom sets are small sets. Since we work with locally small categories, we could use this as definition for limit in the category of sets, then use this to define inductive and projective limits representably using hom sets of categories.

Consider two functors, $F : \mathcal{I} \to \mathcal{C}$ and $G : \mathcal{I}^{\text{op}} \to \mathcal{C}$. For any object $X \in \mathcal{C}$ we can construct the composite functor,

$$\mathcal{I} \xrightarrow{F} \mathcal{C} \xrightarrow{h_X} \textbf{Sets},$$

giving rise to the functors, $\widehat{F}_X \coloneqq \text{Hom}_{\mathcal{C}}(F(-), X)$, and $\widehat{G}_X \coloneqq \text{Hom}_{\mathcal{C}}(X, G(-))$. These is an inductive and projective systems respectively in the category of sets,

$$\widehat{F}_X : \mathcal{I}^{\text{op}} \to \textbf{Sets},$$

It's easy to see that it's a functor from $\mathcal{I}^{\text{op}}$ to **Sets** using the following diagram,

$$\begin{tikzcd} i \arrow[d, "f"] & F_i \arrow[d, "F(f)"'] \arrow[dr] & & G_i \arrow[d, "G(f)"'] & X \arrow[l] \arrow[dl] \\ j & F_j \arrow[r] & X & G_j & \end{tikzcd}$$

and the projective limit exists. The limit of this inductive system F, denoted by $\varinjlim F$, can be defined as the representative of the functor,

$$X \mapsto \varprojlim \widehat{F}_X.$$

[1] Intuitively the limit is the 'closest' object to the system. The notion of closeness comes from morphisms, so if there exists any other object with morphisms to the system, then it must be 'farther' than the limit, or in terms of morphisms there must exist a morphism between this object and the limit, and hence the morphisms to the system must factor through the limit.

So, we have directly by definition,

$$\mathrm{Hom}_{\mathcal{C}}(\varinjlim F, X) \cong \varprojlim(\mathrm{Hom}_{\mathcal{C}}(F, X)).$$

and similarly for the projective systems,

$$\mathrm{Hom}_{\mathcal{C}}(X, \varprojlim G) \cong \varprojlim(\mathrm{Hom}_{\mathcal{C}}(X, G)).$$

This can be translated as follows, for all objects $X \in \mathcal{C}$ and all family of morphisms $f_i : X \to F_i$, in $\mathcal{C}$ such that for all $f \in \mathrm{Hom}_{\mathcal{I}}(i, j)$, with $f_j = f_i \circ F(f)$ factors uniquely through $\varprojlim F$.

$$\begin{tikzcd}
& C \arrow[d, dashed] \arrow[ddl, "f_i"'] \arrow[ddr, "f_j"] & \\
& \varprojlim F \arrow[dl, "\varphi_i"'] \arrow[dr, "\varphi_j"] & \\
F_i \arrow[rr, "F(f)"'] & & F_j
\end{tikzcd}$$

This might be the reason for naming it cones. Similarly, projective limits can be written in terms of universal property as,

$$\begin{tikzcd}
G_i \arrow[rr, "G(f)"] \arrow[dr, "\varphi_i"] \arrow[ddr, "f_i"'] & & G_j \arrow[dl, "\varphi_j"'] \arrow[ddl, "f_j"] \\
& \varinjlim G \arrow[d, dashed] & \\
& L &
\end{tikzcd}$$

Note that if $\mathcal{I}$ admits terminal object t, then the limit $\varprojlim F$ corresponds to the object $F(t)$. Note that indexing sets usually have terminal objects.

A category $\mathcal{C}$ is called complete if it has all small limits, it's called cocomplete if it has all small colimits. In such a case, limit is the functor,

$$\varprojlim : \mathcal{C}^{\mathcal{I}} \to \mathcal{C}$$

Limits are essential tool to construct new objects and new functors.

Limits and colimits exist in the category of sets, so the main goal of doing all this Yoneda and representable functors stuff is to use this structure of the category of sets to study more general categories which do not have limits and colimits. This allows us to evade the problem by going to the functor category.

2.2.1 | Products & Coproducts

Let $\mathcal{C}$ be a category and consider a family $\{X_i\}_{i \in I}$ of objects of $\mathcal{C}$ indexed by a set I, then we can consider the contravariant functor,

$$G : Y \mapsto \prod_{i \in I} \mathrm{Hom}_{\mathcal{C}}(Y, X_i)$$

The product on the right side is the standard product in the category of sets. Assuming the functor is representable, i.e., there exists an object P such that, $G(Y) = \mathrm{Hom}_{\mathcal{C}}(Y, P)$. This is called the product, denoted by, $\prod_{i\in I} X_i$. So by definition we have,

$$\mathrm{Hom}_{\mathcal{C}}(Y, \prod_{i\in I} X_i) = \prod_{i\in I} \mathrm{Hom}_{\mathcal{C}}(Y, X_i)$$

This isomorphism can be translated into the universal property definition as follows, given an object Y and a family of morphisms $f_i : Y \to X_i$ this family factorizes uniquely through $\prod_{i\in I} X_i$, visualized by the diagram,

$$\begin{tikzcd} & Y \arrow[dl, "f_i"'] \arrow[d, dashed, "\exists! h"'] \arrow[dr, "f_j"] & \\ X_i & \prod_{i\in I} X_i \arrow[l, "\pi_i"] \arrow[r, "\pi_j"'] & X_j \end{tikzcd}$$

The order of I is unimportant as composition with a permutation of I also belongs to the same hom set. If all $X_i = X$ then this is denoted by X^I.

Similarly we can consider the functor,

$$Y \mapsto \prod_{i\in I} \mathrm{Hom}_{\mathcal{C}}(X_i, Y)$$

This is a covariant functor. Assuming it's representable there exists an object C such that, $F(Y) = \mathrm{Hom}_{\mathcal{C}}(C, Y)$. The representative C is denoted by $\coprod_{i\in I} X_i$ and by definition we have,

$$\mathrm{Hom}_{\mathcal{C}}(\coprod_{i\in I} X_i, Y) = \prod_{i\in I} \mathrm{Hom}_{\mathcal{C}}(X_i, Y).$$

This isomorphism can be translated as follows, given an object Y and a family of morphisms $f_i : X_i \to Y$ this family factorizes uniquely through $\coprod_{i\in I} X_i$, visualized by the diagram,

$$\begin{tikzcd} X_j \arrow[r, "\epsilon_j"] \arrow[dr, "f_j"'] & \coprod_{i\in I} X_i \arrow[d, dashed, "h"'] & X_i \arrow[l, "\epsilon_i"'] \arrow[dl, "f_i"] \\ & Y & \end{tikzcd}$$

In algebra, for modules, etc. the coproduct is denoted by $\oplus$, and is called direct sum. It follows directly from definition that,

$$\mathrm{Hom}_{\mathcal{C}}(\coprod_{i\in I} X_i, Y) = \mathrm{Hom}_{\mathcal{C}}(Y, \prod_{i\in I} X_i)$$

When the indexing category is discrete, i.e., the only morphisms are the identity morphisms, the limit and colimit in such a case corresponds to products and coproducts.

2.2.2 | Kernel & Cokernel

For sets, the kernel of two maps s, t is defined as the set $\ker(s, t) = \{x \in S \mid s(x) = t(x)\}$. Using this, for any two maps $f, g : Y \rightrightarrows Z$, we have set maps,

$$\mathrm{Hom}_{\mathcal{C}}(X, Y) \to \mathrm{Hom}_{\mathcal{C}}(X, Z)$$

given by the action, $h \mapsto f \circ h$. Using these set maps we can define the functor,

$$Y \mapsto \ker\big(\mathrm{Hom}_{\mathcal{C}}(X,Y) \rightrightarrows \mathrm{Hom}_{\mathcal{C}}(X,Z)\big).$$

This is a covariant functor from the category $\mathcal{C}$ to **Sets**. Assuming this functor is representable, the representative denoted by $\ker(f,g)$ is called the equalizer of f, g.

This isomorphism can be translated as follows, given an object X and morphisms $i : X \to Y$ and $j : X \to Z$ such that $i \circ f = j \circ g$, uniquely factors through $\ker(f,g)$, visualized by the diagram,

$$\begin{tikzcd}
X \arrow[d, dashed, "\exists!"] \arrow[dr, "i"] \arrow[drr, bend left, "j"] & & \\
\ker(f,g) \arrow[r, "e"'] & Y \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & Z
\end{tikzcd}$$

To be able to describe kernel and cokernel we have to first have a zero object, i.e,. an object that's both initial and terminal. An object Z is called a zero object if for any object A, there exists a unique morphism $Z \to A$ and a unique morphism $A \to Z$. It's unique upto isomorphism and denoted by 0. Between any two objects $A, B \in \mathcal{C}$, there exists a unique morphism $0_{A,B}$ given by the composition,

$$A \to 0 \to B$$

In this case, the kernel of a map f is defined as the equalizer of the maps $f, 0 : \mathcal{C} \to \mathcal{C}$, $\ker(f) = \ker(f, 0)$. The kernel of a map $f : Y \to Z$ is a morphism $\iota : \ker(f) \to A$ such that $f \circ \iota = 0_{\ker(f),B}$ and any other morphism $i : X \to Y$ with $f \circ i = 0_{K,B}$ uniquely factors through $\ker(f)$, visualized by the diagram,

$$\begin{tikzcd}
X \arrow[d, dashed, "e"] \arrow[dr, bend left, "i"] & & \\
\ker(f) \arrow[r, "\iota"'] & Y \arrow[r, "f"'] & Z
\end{tikzcd}$$

Here we have not written the zero morphism from X to Z. Similarly we can define coequalizer and cokernel. Given two maps $f, g : Y \rightrightarrows Z$, we have set maps, $\mathrm{Hom}_{\mathcal{C}}(Y,X) \to \mathrm{Hom}_{\mathcal{C}}(Z,X)$ given by the action, $h \mapsto h \circ f$. Coequalizer is the representative of the functor,

$$Y \mapsto \ker\big(\mathrm{Hom}_{\mathcal{C}}(Y,X) \rightrightarrows \mathrm{Hom}_{\mathcal{C}}(Z,X)\big).$$

This can be visualized by the diagram,

$$\begin{tikzcd}
Y \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] \arrow[drr, bend right, "l"'] & Z \arrow[r, "\iota"] \arrow[dr, "k"] & \mathrm{coker}(f,g) \arrow[d, dashed, "e"] \\
& & X
\end{tikzcd}$$

The cokernel of a morphism f is a morphism $\iota : X \to \mathrm{coker}(f)$ with $\iota \circ f = 0_{A,\mathrm{coker}(f)}$, and for any morphism $k : B \to L$ with $k \circ f = 0_{A,L}$ will factor uniquely through $\mathrm{coker}(f)$.

$$\begin{tikzcd}
Y \arrow[r, "f"] & Z \arrow[r, "\iota"] \arrow[dr, bend right, "k"] & \mathrm{coker}(f) \arrow[d, dashed, "e"] \\
& & X
\end{tikzcd}$$

For a category $\mathcal{I}$, we have two natural maps, $\mathrm{Cod} : \coprod \mathrm{Hom}_{\mathcal{I}}(I, J) \to \mathcal{I}$ that sends $f : I \to J$ to $I \in \mathcal{I}$ and the map $\mathrm{Dom} : \coprod \mathrm{Hom}_{\mathcal{I}}(I, J) \to \mathcal{I}$ that sends $f : I \to J$ to J.

Let $\mathcal{C}$ be a category with products and kernels. Now suppose $F : \mathcal{I}^{\mathrm{op}} \to \mathcal{C}$ is a projective system, for each $f : i \to j$, we get two morphisms in $\mathcal{C}$, we get two parallel maps, $\widehat{f} = f \circ \pi$ and $\widehat{g} = \mathbb{1}_{F(i)} \circ \pi_{F(i)}$, where π is the projection from the product $F(i) \prod F(j)$ to $F(i)$,

$$F(i) \prod F(j) \underset{\widehat{g}}{\overset{\widehat{f}}{\rightrightarrows}} F(i).$$

Using this we obtain two morphisms,

$$\textstyle\prod_{i \in \mathcal{I}} F(i) \underset{\widehat{g}}{\overset{\widehat{f}}{\rightrightarrows}} \displaystyle\prod_{f \in \coprod \mathrm{Hom}_{\mathcal{I}}(I,J), I, J \in \mathcal{I}} F(\mathrm{Cod}(f)).$$

It can then be showed that the colimit,

$$\varprojlim_{\mathcal{I}} F = \ker(\widehat{f}, \widehat{g}).$$

and similarly for limit

$$\varinjlim_{\mathcal{I}} G = \mathrm{coker}(\widehat{f}, \widehat{g}).$$

To prove this, one has to prove this for the category of sets, and then the proof for general case follows as it's defined representably in terms of limit for sets, see [3] f. So, if $\mathcal{C}$ possesses kernels and products it possesses colimits, and similarly if it possesses cokernels and coproducts, it possesses limits.

2.2.3 | Pullback or Fibered Product

Let $\mathcal{I}$ be the indexing category with three objects X, Y, Z and two morphisms, $Y \leftarrow X \to Z$ then for functors $F : \mathcal{I} \to \mathcal{C}$, pullback $FY \times_{FX} FZ$ is defined to be the limit of this functor. In terms of universal property, a pullback for a diagram

$$FY \xrightarrow{f} FX \xleftarrow{g} FZ$$

in a category $\mathcal{C}$ is the commutative square with vertex $FY \times_{FX} FZ$ such that any other commutative square factors through it, i.e.,

$$\begin{tikzcd}
P \arrow[dr, dashed, "\exists!"] \arrow[drr, bend left, "f_P"] \arrow[ddr, bend right, "g_P"'] & & \\
& FY \times_{FX} FZ \arrow[r, dashed] \arrow[d, dashed] & FZ \arrow[d, "g"] \\
& FY \arrow[r, "f"'] & FX
\end{tikzcd}$$

The limit is called the fibered product. The categories that have the fibered product are called fibered categories. In case of **Sets** the pullback always exist because limits exist and the pullback consists of all elements (x, y) such that $f(x) = g(y)$.

Similarly, a pushforward corresponds to the limit of the functor $G : \mathcal{I}^{\mathrm{op}} \to \mathcal{C}$ as above,

$$\begin{tikzcd}
FX \arrow[r, "f"] \arrow[d, "g"'] & FZ \arrow[d, dashed] \arrow[ddr, bend left, "f_P"] & \\
FY \arrow[r, dashed] \arrow[drr, bend right, "g_P"'] & FY \coprod_{FX} FZ \arrow[dr, dashed, "\exists!"] & \\
& & C
\end{tikzcd}$$

2.2.4 | Exponentiation

The categorical notions of product and coproduct correpsond to the arithematic operations such as multiplication and addition. We can similarly talk about exponentiation. In the category of sets, **Sets**, for $X, Z \in \mathcal{C}$, Z^X is the function set consisting of all functions $h : X \to Z$. Here we have the bijection,

$$\mathrm{Hom}(Y \times X, Z) \to \mathrm{Hom}(Y, Z^X).$$

for a function, $f : Y \times X \to Z$, this map sends each $y \in Y$ to the function $f(y, -) \in Z^X$. Conversely given a function $f' : Y \to Z^X$, we can define a map $f(y, x) = f'(y)(x)$. So,

$$\mathrm{Hom}(Y \times X, Z) \cong \mathrm{Hom}(Y, Z^X)$$

or equivalently, $(-)^X$ is the right adjoint of $(-) \times X$. By setting $Y = 1$, we obtain,

$$Z^X \cong \mathrm{Hom}(1, Z^X) \cong \mathrm{Hom}(X, Z).$$

Let $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$ be two functors, they are called an adjoint pair if

$$\mathrm{Hom}_{\mathcal{D}}(F(X), Y) = \mathrm{Hom}_{\mathcal{C}}(X, G(Y))$$

for all $X \in \mathcal{C}$ and $Y \in \mathcal{D}$. F is a left adjoint to G and G is a right adjoint to F. This is denoted by, $F \dashv G$. Adjoints are unique upto isomorphism and is the representative of the functor,

$$X \mapsto \mathrm{Hom}_{\mathcal{D}}(F(X), Y)$$

This gives us, $\mathrm{Hom}_{\mathcal{C}}(G(X), G(Y)) \cong \mathrm{Hom}_{\mathcal{D}}(F \circ G(X), Y)$, and, $\mathrm{Hom}_{\mathcal{D}}(F(X), F(Y)) \cong \mathrm{Hom}_{\mathcal{C}}(X, G \circ F(Y))$.

References

[1] S Mac Lane, Categories for the Working Mathematician, Springer, 1998

[2] S Mac Lane, L Moerdijk, Sheaves in Geometry and Logic: A First Introduction to Topos Theory, Springer, 1992

[3] M Kashiwara, P Schapira, Categories and Sheaves, Springer, 2000