1 Product

Definition 1.1 (Product). *Let $\mathcal{C}$ a category and $A, B \in \mathcal{C}$, $(A \times B, \pi_0, \pi_1)$ forms a product of A and B where $A \times B \in \mathcal{C}$, $\pi_0 : A \times B \to A$ and $\pi_1 : A \times B \to B$, if for any $X \in \mathcal{C}$ with $\pi_0' : X \to A$ and $\pi_1' : X \to B$, there is a unique arrow $u : X \to A \times B$ such that the following diagram commutes:*

$$
\begin{tikzcd}
X \arrow[dr, dashed, "u"] \arrow[drr, bend left, "\pi_1'"] \arrow[ddr, bend right, "\pi_0'"'] & & \\
& A \times B \arrow[r, "\pi_1"] \arrow[d, "\pi_0"] & B \\
& A &
\end{tikzcd}
$$

Furthermore, a product of A and B is a limit of diagram:

$$(A \qquad B)$$

One may trying to show that $0 \times X \simeq 0$ by:

$$
\begin{tikzcd}
0 \times X \arrow[dr, red, "\pi_0"] \arrow[drrr, red, bend left, "\pi_1"] \arrow[dddrr, bend right, "\pi_0"'] & & & \\
& 0 \arrow[dr, dashed, "!"] \arrow[drr, red, bend left, "!"] \arrow[ddr, bend right, "1_0"'] & & \\
& & 0 \times X \arrow[r, "\pi_1"] \arrow[d, "\pi_0"] & X \\
& & 0 &
\end{tikzcd}
$$

However, the red triangle needs not to commutes, that is, the arrow π_0 from $(0 \times X, \pi_0, \pi_1)$ to $(0, 1_0, !)$ may not exist.

Definition 1.2 (Product of Arrow). *Suppose $(A \times B, \pi_0, \pi_1)$ and $(C \times D, \pi_2, \pi_3)$ are two product, and $f : A \to C$, $g : B \to D$. The product of arrow $f \times g$ is a* *unique arrow from $A \times B$ to $C \times D$ such that the following diagram commutes:*

$$\begin{tikzcd}
A & A \times B \arrow[l, "\pi_0"'] \arrow[r, "\pi_1"] \arrow[d, dashed, "f\times g"] & B \arrow[d, "g"] \\
C & C \times D \arrow[l, "\pi_2"'] \arrow[r, "\pi_3"] & D
\arrow[from=1-1, to=2-1, "f"']
\end{tikzcd}$$

2 Exponential

Definition 2.1. *Let $\mathcal{C}$ a category. For any $B, C \in \mathcal{C}$, (C^B, ev) forms an exponential where $C^B \in \mathcal{C}$ and $ev : C^B \times B \to C$, if for any object $A \in \mathcal{C}$ and $f : A \times B \to C$, there is a unique $u : A \to C^B$ such that $f = ev \circ (u \times 1_B)$. In other words, the follow diagram commutes.*

$$\begin{tikzcd}
A \times B \arrow[dr, "f"] \arrow[dd, dashed, "u\times 1_B"] & \\
& C \\
C^B \times B \arrow[ur, "ev"] &
\end{tikzcd}$$

3 Pullback

Theorem 3.1. *Suppose we have two joined commuting squares like:*

$$\begin{tikzcd}
L \arrow[r, "f"] \arrow[d, "l"] & M \arrow[r, "g"] \arrow[d, "m"] & N \arrow[d, "n"] \\
X \arrow[r, "h"] & Y \arrow[r, "j"] & Z
\end{tikzcd}$$

Then:

1. *The outer rectangle is a pullback square if two inner squares are pullback squares.*

2. *The inner-left square is a pullback square if the ouer rectangle and the inner-right square are pullback squares.*

Proof.

1. For any (A, a, b) such that $j \circ h \circ a = n \circ b$, then there is a unique $u : A \to M$ such that $h \circ a = m \circ u$ and $b = g \circ u$. Then there is a unique $v : A \to L$ such that $l \circ a = v$ and $f \circ v = u$, which makes (A, a, b) against to the outer rectangle commutes.

$$\begin{tikzcd}
A \arrow[ddr, bend right, "a"'] \arrow[dr, dashed, "v"] \arrow[drr, dashed, "u"] \arrow[drrr, bend left, "b"] & & & \\
& L \arrow[r, "f"] \arrow[d, "l"] & M \arrow[r, "g"] \arrow[d, "m"] & N \arrow[d, "n"] \\
& X \arrow[r, "h"] & Y \arrow[r, "j"] & Z
\end{tikzcd}$$

2. For any (A, a, b) such that $h \circ a = m \circ b$, consider the inner-right pullback, then we have a unique $u : A \to M$ such that the diagram commutes:

$$\begin{tikzcd}
A \arrow[ddr, bend right, "a"'] \arrow[drr, dashed, "u"'] \arrow[drr, bend left, "b"] \arrow[drrr, bend left, "g \circ b"] & & & \\
& L \arrow[r, "f"] \arrow[d, "l"] & M \arrow[r, "g"] \arrow[d, "m"] & N \arrow[d, "n"] \\
& X \arrow[r, "h"] & Y \arrow[r, "j"] & Z
\end{tikzcd}$$

However, if we replace u with b, we have $g \circ b = g \circ b$ and $h \circ a = m \circ b$, that means b can do u's job, but we know u is unique, so $b = u$. Now consider the outer pullback, we have a unique $v : A \to L$ such that the diagram commutes:

$$\begin{tikzcd}
A \arrow[dr, dashed, "v"] \arrow[drr, dashed, "u=b"] \arrow[drrr, bend left, "g\circ b"] \arrow[ddr, bend right, "a"'] & & & \\
& L \arrow[r, "f"] \arrow[d, "l"] & M \arrow[r, "g"] \arrow[d, "m"] & N \arrow[d, "n"] \\
& X \arrow[r, "h"] & Y \arrow[r, "j"] & Z
\end{tikzcd}$$

That is, $l \circ v = a$ and $g \circ f \circ v = g \circ b$, we claim that v is the unique factorization from $(A, a, u = b)$ to (L, l, f). It is obvious that $l \circ v = a$, we need to show $f \circ v = u = b$. We may use the trick we just used, we can see that $g \circ f \circ v = g \circ u$ and $m \circ f \circ v = h \circ l \circ v = h \circ a$. So $f \circ v$ can do b's job, so $f \circ v = b$.

For any arrow $w : A \to L$ such that $l \circ a = w$ and $f \circ w = b$, then we have also $g \circ f \circ w = g \circ b$, which implies w is the unique arrow from $A \to L$ such that the outer diagram commutes, so $w = v$.

□

Theorem 3.2. *A pullback square for the corner $D_1 \to V \leftarrow D_2$ is a product of $D_1 \to V$ and $V \leftarrow D_2$ in the slice category $\mathcal{C}/V$.*

Proof. Suppose (L, f, g) is the pullback of such corner, then we first need to show that there is an arrow $l : L \to V$ such that $s \circ f = l$ (therefore a morphism from (L, l) to (D_1, s)) and $t \circ g = l$ (a morphism from (L, l) to (D_2, t)).

Since (L, f, g) makes the pullback square commutes, we know $s \circ f = t \circ g$, therefore we let $l = s \circ f$ (or equivalently $t \circ g$).

We need to show that $((L, l), f, g)$ forms a product of (D_1, s) and (D_2, t), consider any $((A, a), b, c)$ where $a : A \to V$ such that $s \circ b = a$ and $t \circ c = a$. Just like l for L, a is redundant, so we may omit it. Now, the diagram looks like:

$$\begin{tikzcd}
A \arrow[drr, bend left, "c"] \arrow[dr, dashed, "!u"] \arrow[ddrr, bend left=15, "a"] \arrow[ddr, bend right, "b"'] & & \\
& L \arrow[r, "g"] \arrow[d, "f"'] \arrow[dr, "l"] & D_2 \arrow[d, "t"] \\
& D_1 \arrow[r, "s"] & V
\end{tikzcd}$$

Since (L, f, g) is a pullback, we know there is a unique $u : A \to L$ such that two triangle commutes. However, we must first show that u is an arrow from (A, a) to (L, l), that is, $l \circ u = a$. It is easy to see that $l \circ u = s \circ f \circ u = s \circ b = a$. □

Theorem 3.3. *if a category has all binary products and all equalizers for every pair of parallel arrows, then it has a pullback for any corners.*

Proof. Suppose $X \to Z \leftarrow Y$ a corner, then consider the product $X \times Y$:

$$\begin{tikzcd}
X \times Y \arrow[r, "\pi_1"] \arrow[d, "\pi_0"'] & Y \arrow[d, "t"] \\
X \arrow[r, "s"] & Z
\end{tikzcd}$$

Now, consider the equalizer for the parallel arrows $t \circ \pi_1$ and $s \circ \pi_0$:

$$\begin{tikzcd}
E \arrow[r, "e"] & X \times Y \arrow[r, shift left, "s\circ\pi_0"] \arrow[r, shift right, "t\circ\pi_1"'] & Z
\end{tikzcd}$$

We claim $(E, \pi_0 \circ e, \pi_1 \circ e)$ is a pullback of such corner. For any (F, f, g) such that the outer diagram commutes:

$$
\begin{tikzcd}
F \arrow[drrr, bend left, "b"] \arrow[dddr, bend right, "a"'] & & & \\
& E \arrow[rr, "\pi_1\circ e"] \arrow[dd, "\pi_0\circ e"'] \arrow[dr, dashed, "e"] & & Y \arrow[dd, "t"] \\
& & X\times Y \arrow[ur, "\pi_1"'] \arrow[dl, "\pi_0"] & \\
& X \arrow[rr, "s"] & & Z
\end{tikzcd}
$$

it is easy to see that there is a unique arrow $u : F \to X \times Y$ such that $\pi_0 \circ u = a$ and $\pi_1 \circ u = b$ since $X \times Y$ is a product. Then there is another unique arrow $v : F \to E$ such that $e \circ v = u$ since E is a equalizer.

$$
\begin{tikzcd}
F \arrow[drrr, bend left, "b"] \arrow[dddr, bend right, "a"'] \arrow[dr, dashed, "v"] \arrow[ddrr, dashed, bend left, "u"] & & & \\
& E \arrow[rr, "\pi_1\circ e"] \arrow[dd, "\pi_0\circ e"'] \arrow[dr, dashed, "e"] & & Y \arrow[dd, "t"] \\
& & X\times Y \arrow[ur, "\pi_1"'] \arrow[dl, "\pi_0"] & \\
& X \arrow[rr, "s"] & & Z
\end{tikzcd}
$$

Obviously, (commute) $\pi_0 \circ e \circ v = \pi_0 \circ u = a$ and $\pi_1 \circ e \circ v = \pi_1 \circ u = b$. (unique) If an arrow $w : F \to E$ can do the job, then $e \circ w : F \to X \times Y$ is another factorization from F to the product $X \times Y$, so $e \circ w = u$, but that means w is also a factorization from F to the equalizer E, which means $v = w$.

So $(E, \pi_0 \circ e, \pi_1 \circ e)$ is a pullback of such corner. $\square$

Theorem 3.4. *If a category has a terminal object and has a pullback for every corner, then it has all binary product.*

Theorem 3.5. *If a category has a terminal object and has a pullback for every corner, then it has a equalizer for every parallel arrwos.*

Proof. Suppose $s, t : X \to Y$ are parallel arrows, then the following diagram commutes:

$$
\begin{tikzcd}
X \arrow[dr, dashed, "\langle s,t \rangle"] \arrow[drr, bend left, "t"] \arrow[dd, bend right, "s"'] & & \\
& Y \times Y \arrow[r, "\pi_1"] \arrow[d, "\pi_0"] & Y \\
& Y &
\end{tikzcd}
$$

Note that we have $Y \times Y$ since this category has all binary products. Then consider this corner:

$$
\begin{tikzcd}
& Y \arrow[d, "\langle 1_Y, 1_Y \rangle"'] \\
X \arrow[r, "\langle s,t \rangle"] & Y \times Y
\end{tikzcd}
$$

We have an object E, $e : E \to X$ and $f : E \to Y$ such that the square commutes:

$$
\begin{tikzcd}
E \arrow[r, "f"] \arrow[d, "e"'] & Y \arrow[d, "\langle 1_Y, 1_Y \rangle"'] \\
X \arrow[r, "\langle s,t \rangle"] & Y \times Y
\end{tikzcd}
$$

(Proof comes from textbook until here)

We can see that $\pi_0 \circ \langle s, t \rangle \circ e = s \circ e$ while $\pi_0 \circ \langle 1_Y, 1_Y \rangle \circ f = 1_Y \circ f = f$, therefore $s \circ e = f$, similarly $t \circ e = f$, so $s \circ e = t \circ e$. We claim E is the equalizer for the parallel arrows $s, t : X \to Y$. For any (F, e') such that $s \circ e' = t \circ e'$, then we have a unique arrow $u : F \to E$ such that this diagram commutes:

$$
\begin{tikzcd}
F \arrow[dr, dashed, "u"] \arrow[drr, bend left, "s\circ e'=t\circ e'"] \arrow[ddr, bend right, "e'"'] & & \\
& E \arrow[r, "f"] \arrow[d, "e"'] & Y \arrow[d, "\langle 1_Y, 1_Y \rangle"] \\
& X \arrow[r, "\langle s,t \rangle"] & Y \times Y
\end{tikzcd}
$$

where $e \circ u = e'$. Suppose $v : F \to E$ where $e \circ v = e'$, then $f \circ v = s \circ e \circ v = s \circ e'$. $\square$

$$\Big| -|x-z|_X + |y-z|_X \Big|_{\approx \cap \approx} \geq \epsilon$$

4 Fiber and Fibration

I am trying to understand fiber, fibration and pullback with my stupid brain.

4.1 Fiber

I will use "intuitive" rather than "definition" cause I really don't understand fiber.

Intuitive 4.1 (Fiber)**.** *Suppose we are in a space (i.e.* **Set***), and a mapping $f : A \to B$, then for some point $b \in B$, the inverse image of b, which is exactly $f^{-1}(b)$, is called a fiber of f over b.*

We can treat a product as a pullback with apex 1, the terminal object:

$$
\begin{tikzcd}
C \arrow[dr, dashed, "!u"] \arrow[drr, bend left, "g"] \arrow[ddr, bend right, "f"'] & & \\
& A \times B \arrow[r, "\pi_1"] \arrow[d, "\pi_0"'] & B \arrow[d, "!"] \\
& A \arrow[r, "!"'] & 1
\end{tikzcd}
$$

We can treat A as the fiber against to the only point in 1, same for B. Now, what if we replace 1 with something else?

$$
\begin{tikzcd}
C \arrow[dr, dashed, "!u"] \arrow[drr, bend left, "g"] \arrow[ddr, bend right, "f"'] & & \\
& A \times_I B \arrow[r, "\pi_1"] \arrow[d, "\pi_0"'] & B \arrow[d, "t"] \\
& A \arrow[r, "s"'] & I
\end{tikzcd}
$$

For every point $i \in I$, we have fiber $A_i \subseteq A$ and $B_i \subseteq B$, which can form a product $A_i \times B_i$. We may sum all these products, and finally get $A \times_I B$, this is why the pullback is sometimes called *fiber product*.

We can also pick certain fiber from this pullback:

$$
\begin{tikzcd}
\varphi \arrow[r, "i"] \arrow[d, "!"'] & A \arrow[d, "f"] \\
1 \arrow[r, "x"'] & B
\end{tikzcd}
$$

The morphism $x : 1 \to B$ is a global element, which "pick" an element of B, then i must maps φ to the fiber of f over point x, which should be a injection.

The collection of fiber (the source of the morphism/the domain of the function) is called *fiber bundle*.

4.2 Fibration

Some intuitive comes from this article.

Intuitive 4.2. *A fibration works like an indexed family (i.e. a function $I \to A$), but do it in fiber way (i.e. a function $A \to I$).*

4.3 Base-change Functor

These section is related to *The Dao of FP*

We can also treat the morphism on right-hand side as a fibration, and the bottom-left corner a base (the target of a fibration):

$$\begin{tikzcd} ? \arrow[r, dashed] \arrow[d, dashed] & E \arrow[d, "p"] \\ A \arrow[r, "f"'] & B \end{tikzcd}$$

Then we can treat $E \xrightarrow{p} B$ as an object in the slice category $\mathcal{C}/B$, similarly, the left-hand side morphism an object in $\mathcal{C}/A$. Then we can define a base-change functor $f^* : \mathcal{C}/B \to \mathcal{C}/A$ such that:

$$\begin{tikzcd} f^*E \arrow[r, dashed, "g"] \arrow[d, dashed, "f^*p"'] & E \arrow[d, "p"] \\ A \arrow[r, "f"'] & B \end{tikzcd}$$

a pullback.

We denote f^*E as the source of f^*p, it doesn't mean that f^* accept a object in $\mathcal{C}$.

We need to define the action of base-change functor on the morphism of $\mathcal{C}/B$:

$$
\begin{tikzcd}
f^*E' \arrow[rrr, dashed, bend left, "g'"] \arrow[dr, "f*p'"'] & f^*E \arrow[l, "?"'] \arrow[r, dashed, "g"] \arrow[d, "f^*p"'] & E \arrow[r, "e"] \arrow[d, "p"] & E' \arrow[dl, "p'"] \\
& A \arrow[r, "f"'] & B &
\end{tikzcd}
$$

Two commute triangles are the morphisms in $\mathcal{C}/A$ and $\mathcal{C}/B$.

Since f^*E' is a pullback of $A \xrightarrow{B} \xleftarrow{E'}$, it tips us that we can find the commute square below to get the morphism we want:

$$
\begin{tikzcd}
f^*E \arrow[r] \arrow[d] & E' \arrow[d, "p'"] \\
A \arrow[r, "f"'] & B
\end{tikzcd}
$$

If we look the last diagram carefully, we can find this square commutes:

$$
\begin{tikzcd}
f^*E \arrow[r, dashed, "g"] \arrow[d, "f^*p"'] & E \arrow[r, "e"] \arrow[d, "p"] & E' \arrow[dl, "p'"] \\
A \arrow[r, "f"'] & B &
\end{tikzcd}
$$

therefore

$$
\begin{tikzcd}
f^*E \arrow[dr, dashed, "f^*e"] \arrow[drr, bend left, "e \circ g"] \arrow[ddr, bend right, "f^*p"'] & & \\
& f * E' \arrow[r, "g'"] \arrow[d, "f^*p'"'] & E' \arrow[d, "p'"] \\
& A \arrow[r, "f"'] & B
\end{tikzcd}
$$

The functoriality follows the fact that f^*e is unique that makes the diagram commutes.

5 Functors

Definition 5.1 (Full)**.** *A functor $F : \mathcal{C} \to \mathcal{D}$ is called full, if for any $a, b \in \mathcal{C}$, the mapping on morphism $F : \mathcal{C}(a, b) \to \mathcal{D}(Fa, Fb)$ is surjective.*

Definition 5.2 (Faithful)**.** *A functor $F : \mathcal{C} \to \mathcal{D}$ is called faithful, if for any $a, b \in \mathcal{C}$, the mapping on morphism $F : \mathcal{C}(a, b) \to \mathcal{D}(Fa, Fb)$ is injective.*

Definition 5.3 (Essentially Full)**.** *A functor $F : \mathcal{C} \to \mathcal{D}$ is called Essentially full, if for any $a \in \mathcal{C}$, the mapping on object $F : \mathcal{C} \to \mathcal{D}$ is surjective.*

Theorem 5.1. *Suppose $F : \mathcal{C} \to \mathcal{D}$ a functor, and $f : a \to b$ a morphism in $\mathcal{C}$. Then f is an isomorphism iff Ff is an isomorphism.*

Proof. ($\Rightarrow$) We claim $F(f^{-1}) : Fb \to Fa$ is an inverse, we can see that $F(f^{-1} \circ f) = F(id_a) = id_{Fa}$ and $F(f \circ f^{-1}) = F(id_b) = id_{Fb}$.

($\Leftarrow$) Suppose Fg is the inverse of Ff, and we can retrieve g from Fg cause F is full faithful. Then $F(g \circ f) = Fg \circ Ff = id_{Fa} = F(id_a)$ therefore $g \circ f = id_a$ since F is full faithful, similar to $F(f \circ g)$, so f is indeed an isomorphism. □

Corollary 5.1. *Suppose $F : \mathcal{C} \to \mathcal{D}$ is full and faithful, show that F is injective on object.*

Proof. Trivial by previous theorem. □

Note that a commuting diagram applied to a functor is still commutes, due to the functoriality:

$$
\begin{tikzcd}
a \arrow[r, "f"] \arrow[d, "i"'] & b \arrow[d, "j"] \\
c \arrow[r, "g"'] & d
\end{tikzcd}
\quad \Rightarrow \quad
\begin{tikzcd}
Fa \arrow[r, "Ff"] \arrow[d, "Fi"'] & Fb \arrow[d, "Fj"] \\
Fc \arrow[r, "Fg"'] & Fd
\end{tikzcd}
$$

Definition 5.4 (Natural Transform)**.** *Suppose $F, G : \mathcal{C} \to \mathcal{D}$ are functors, then $\alpha : F \Rightarrow G$ is called a natural transform from F to G, if:*

- *For any $x \in \mathcal{C}$, $\alpha_x : Fx \to Gx$ a morphism in $\mathcal{D}$.*
- *Furthermore, for any morphism $f : x \to y$ in $\mathcal{C}$, the following square commutes:*

$$
\begin{tikzcd}
Fx \arrow[r, "Ff"] \arrow[d, "\alpha_x" description] & Fy \arrow[d, "\alpha_y" description] \\
Gx \arrow[r, "Gf"] & Gy
\end{tikzcd}
$$

The one of composition of two natural transforms is vertical composition:

$$
\begin{tikzcd}
\mathcal{C} \arrow[r, bend left=50, "F"{name=F}] \arrow[r, "G"{name=G}] \arrow[r, bend right=50, "H"'{name=H}] & \mathcal{D}
\arrow[Rightarrow, from=F, to=G, "\alpha"]
\arrow[Rightarrow, from=G, to=H, "\beta"]
\end{tikzcd}
$$

which is indeed a natural transform cause:

$$\begin{tikzcd}
Fx \arrow[r, "Ff"] \arrow[d, "\alpha_x"] \arrow[dd, bend right, dashed, "(\beta\circ\alpha)_x"'] & Fy \arrow[d, "\alpha_y"] \arrow[dd, bend left, dashed, "(\beta\circ\alpha)_y"] \\
Gx \arrow[r, "Gf"] \arrow[d, "\beta_x"] & Gy \arrow[d, "\beta_y"] \\
Hx \arrow[r, "Hf"] & Hy
\end{tikzcd}$$

the outer diagram commutes.

There is another way to compose two natural transforms, the horizontal composition:

$$\begin{tikzcd}
\mathcal{C} \arrow[r, bend left, "F"] \arrow[r, bend right, "F'"'] \arrow[r, phantom, "\Downarrow\alpha"] & \mathcal{D} \arrow[r, bend left, "G"] \arrow[r, bend right, "G'"'] \arrow[r, phantom, "\Downarrow\beta"] & \mathcal{E}
\end{tikzcd}$$

We would expect that there is a natural transform $\beta \cdot \alpha : G \circ F \Rightarrow G' \circ F'$, but how? Firstly, we have the following diagram commutes cause α is a natural transform:

$$\begin{tikzcd}
Fx \arrow[r, "Ff"] \arrow[d, "\alpha_x"] & Fy \arrow[d, "\alpha_y"] \\
F'x \arrow[r, "F'f"] & F'y
\end{tikzcd}$$

Then we apply it to the functor G.

$$\begin{tikzcd}
G(Fx) \arrow[r, "G(Ff)"] \arrow[d, "G(\alpha_x)"] & G(Fy) \arrow[d, "G(\alpha_y)"] \\
G(F'x) \arrow[r, "G(F'f)"] & G(F'y)
\end{tikzcd}$$

It is similar to what we want, beside the bottom arrow, it is time to use β.

$$\begin{tikzcd}
G(Fx) \arrow[r, "G(Ff)"] \arrow[d, "G(\alpha_x)"] & G(Fy) \arrow[d, "G(\alpha_y)"] \\
G(F'x) \arrow[r, "G(F'f)"] \arrow[d, "\beta_{F'x}"] & G(F'y) \arrow[d, "\beta_{F'y}"] \\
G'(F'x) \arrow[r, "G'(F'f)"] & G'(F'y)
\end{tikzcd}$$

And the $\beta_{F'\_} \circ G(\alpha_\_)$ is the definition of $\beta \cdot \alpha$.

Also, if one of the natural transform is the identity transform, say $id_G \cdot \alpha$, then it can be denoted by $G \cdot \alpha$. Notice that $G \cdot \alpha$ has type $G \circ F \Rightarrow G \circ F'$, which "modifies" only one side.

You can see that the horizontal composition is much different from vertical composition, the former one is much like a product of morphism (if you treat $\circ$ as some kind of product):

$$\alpha : F \Rightarrow F'$$
$$\beta : G \Rightarrow G'$$
$$\beta \cdot \alpha : F \circ G \Rightarrow F' \circ G'$$

While the later one is much like a composition of morphism:

$$\alpha : F \Rightarrow G$$
$$\beta : G \Rightarrow H$$
$$\beta \circ \alpha : F \Rightarrow H$$

It looks like we can write horizontal composition in vertical composition of two horizontal compositions:

$$\begin{tikzcd}
\mathcal{C} \arrow[r, bend left, "F"] \arrow[r, bend right, "F'"'] \arrow[r, phantom, "\Downarrow\alpha"] & \mathcal{D} \arrow[r, bend left, "G"] & \mathcal{E}
\end{tikzcd} \quad \text{and} \quad \begin{tikzcd}
\mathcal{C} \arrow[r, bend right, "F'"'] & \mathcal{D} \arrow[r, bend left, "G"] \arrow[r, bend right, "G'"'] \arrow[r, phantom, "\Downarrow\beta"] & \mathcal{E}
\end{tikzcd}$$

In symbol, it is $G(\alpha_-)$ (the former one) and $\beta_{F'-}$ (the later one), and finally $\beta_{F'-} \circ G(\alpha_-)$, which is exactly the horizontal composition $\beta \cdot \alpha$. Similarly, we might suppose there is another definition of horizontal composition: $G'(\alpha_-) \circ \beta_{F-}$ which is the vertical composition of:

$$\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "F"] & \mathcal{D} \arrow[r, bend left, "G"] \arrow[r, bend right, "G'"'] \arrow[r, phantom, "\Downarrow\beta"] & \mathcal{E} \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} \mathcal{C} \arrow[r, bend left, "F"] \arrow[r, bend right, "F'"'] \arrow[r, phantom, "\Downarrow\alpha"] & \mathcal{D} \arrow[r, bend right, "G'"'] & \mathcal{E} \end{tikzcd}$$

The corresponding diagram would be: apply the naturality diagram of α to G', then put β above it.

6 Yoneda

This chapter combines arguments from some books:

- The Dao of FP
- The Joy of Abstraction

Definition 6.1. $H_x = \mathcal{C}(-, x)$ *and* $H^x = \mathcal{C}(x, -)$.

Take H^x as an example, it sends $\mathcal{C}$ to **Set**, the interesting part is the mapping on morphism. For any morphism $f : a \to b$ of $\mathcal{C}$, H^f must be a mapping $\mathcal{C}(x, a) \to \mathcal{C}(x, b)$, we can see that $g \mapsto f \circ g$ would be a choice.

We have to show that it satifies the functoriality:

- $H^{id_a}(g) = id_a \circ g = g$
- $H^{f \circ g}(h) = (f \circ g) \circ h = f \circ (g \circ h) = H^f(g \circ h) = H^f(H^g(h)) = (H^f \circ H^g)(h)$.

Similar to H_x, the only difference is that H_x is a contrafunctor.

Suppose $f : a \to b$ an isomorphism, we can see that H^x gives an isomorphism between two hom-sets: $\mathcal{C}(x, a)$ and $\mathcal{C}(x, b)$.

Furthermore, we can no more fix x, that is, make $H_\bullet$ (or $H^\bullet$) a functor from $\mathcal{C}$ to $[\mathcal{C}^{op}, \mathbf{Set}]$, a functor to a functor!

The problem we need to solve is that what should $H_\bullet$ do on a morphism $f : a \to b$. Since $H_\bullet$ produce a functor, H_f must produce a natural transformation between H_a and H_b. Suppose $x, y \in \mathcal{C}$ and $g : x \to y$, note that H_a and H_b are contrafunctor, so we need to reverse the arrows!

$$\begin{tikzcd}
H_a(y) \arrow[r, "H_a(g)"] \arrow[d, "(H_f)_y"'] & H_a(x) \arrow[d, "(H_f)_x"] \\
H_b(y) \arrow[r, "(H_b)(g)"'] & H_b(x)
\end{tikzcd}$$

and we can unfold the definitions

$$\begin{tikzcd}
\mathcal{C}(y,a) \arrow[r, "\mathcal{C}(g,a)"] \arrow[d, "(H_f)_y"'] & \mathcal{C}(x,a) \arrow[d, "(H_f)_x"] \\
\mathcal{C}(y,b) \arrow[r, "\mathcal{C}(g,b)"'] & \mathcal{C}(x,b)
\end{tikzcd}$$

and suppose $s \in \mathcal{C}(y,a)$, we know the top-right corner would be $s \circ g$ since $\mathcal{C}(g,a) = - \circ g$ (same to $\mathcal{C}(g,b)$). In order to construct an arrow in $\mathcal{C}(y,b)$, we can pre-compose the arrow $f : a \to b$. Then the bottom-left corner would be $f \circ s$, and the bottom-right corner would be $(f \circ s) \circ g$ (by left-bottom path) and $f \circ (s \circ g)$ (by top-right path), which is exactly the same!

$$\begin{tikzcd}
s \arrow[r, mapsto, "- \circ g"] \arrow[d, mapsto, "f \circ -"'] & s \circ g \arrow[d, mapsto, "f \circ -"] \\
f \circ s \arrow[r, mapsto, "- \circ g"'] & (f \circ s) \circ g = f \circ (s \circ g)
\end{tikzcd}$$

Note that the condition here $(f \circ -) \circ (- \circ g) = (- \circ g) \circ (f \circ -)$ is the naturality condition which is mentioned in *The Dao of FP*. A bijection between two hom-sets $\alpha_x = \mathcal{C}(x,a) \to \mathcal{C}(x,b)$ that satifies the naturality condition $\alpha_y \circ (- \circ g) = (- \circ g) \circ \alpha_x$ can retrieve the isomorphism between a and b. This will be unsurprised if we notice that such bijection with naturality condition forms a natural transformation, then we can retrieve the morphism (not isomorphism yet) from it. The morphism becomes iso- when we know $H_\bullet$ is full and faithful (see below and chapter *functor*),

Definition 6.2. *$H_\bullet$ is called Yoneda embedding.*

Theorem 6.1. *Shows $H_\bullet$ is an embedding by showing it is full and faithful.*

Proof. (Full) For any $a, b \in \mathcal{C}$, suppose $\alpha : [\mathcal{C}^{op}, \mathbf{Set}](H_a, H_b)$ a morphism (natural transformation). Use the Yoneda trick, we have $\alpha_a : \mathcal{C}(a, a) \to \mathcal{C}(a, b)$ and then $\alpha_a(id_a) : \mathcal{C}(a, b)$. As we see the definition of $H_\bullet$ on morphism, we should expect that α has form $f \circ -$ for some $f : a \to b$. But how coincident, we have a morphism $\alpha_a(id_a) : \mathcal{C}(a, b)$. So we claim $H_{\alpha_a(id_a)} = \alpha$. (In the other hand, if α has form $f \circ -$, then $\alpha_a(id_a) = f \circ id_a = f$). For any $x \in \mathcal{C}$, we need to show $(H_{\alpha_a(id_a)})_x = \alpha_x : \mathcal{C}(x, a) \to \mathcal{C}(x, b)$. So we suppose $g \in \mathcal{C}(x, a)$, then the following diagram commutes since α is natural:

$$\begin{tikzcd}
\mathcal{C}(a,a) \arrow[r, "-\circ g"] \arrow[d, "\alpha_a"'] & \mathcal{C}(x,a) \arrow[d, "\alpha_x"] \\
\mathcal{C}(a,b) \arrow[r, "-\circ g"'] & \mathcal{C}(x,b)
\end{tikzcd}$$

Then

$$\begin{tikzcd}
id_a \arrow[r, mapsto, "-\circ g"] \arrow[d, mapsto, "\alpha_a"'] & id_a \circ g = g \arrow[d, mapsto, "\alpha_x"] \\
\alpha_a(id_a) \arrow[r, mapsto, "-\circ g"'] & \alpha_a(id_a) \circ g = \alpha_x(g)
\end{tikzcd}$$

is a proof of $H_{\alpha_a(id_a)}(g) = \alpha_a(id_a) \circ g = \alpha_x(g)$.

(Faithful) Suppose $f \circ - H_f = H_g = g \circ -$, then $f = f \circ id_a = H_f(id_a) = H_g(od_a) = g \circ id_a = g$. $\square$

As we see in the proof of $H_\bullet$ is a full functor, the natural transformation α at some x (therefore any $x \in \mathcal{C}$) is completely determined by the value $\alpha_a(id_a)$, cause for any g, we have $\alpha_a(id_a) \circ g = \alpha_x(g)$.

We may rename H_b with F, then

$$\begin{tikzcd}
\mathcal{C}(a,a) \arrow[rrr, "\mathcal{C}(g,a)"] \arrow[ddd, "\alpha_a"'] & & & \mathcal{C}(x,a) \arrow[ddd, "\alpha_x"] \\
& id_a \arrow[r, mapsto, "-\circ g"] \arrow[d, mapsto, "\alpha_a"'] & id_a \circ g = g \arrow[d, mapsto, "\alpha_x"] & \\
& \alpha_a(id_a) \arrow[r, mapsto, "Fg"'] & (Fg)(\alpha_a(id_a)) = \alpha_x(g) & \\
Fa \arrow[rrr, "Fg"'] & & & Fx
\end{tikzcd}$$

It seems that H_b can be replaced with any functor in $[\mathcal{C}^{op}, \mathbf{Set}]$, furthermore, the natural transformation is still determined by $\alpha_a(id_a)$ (and $\alpha_a(id_a)$ is determined by α, trivial though, but it implies that there is a precisely corresponding).

Theorem 6.2 (Yoneda Lemma)**.** *Show that the natural transformation between H_a and any functor $F \in [\mathcal{C}^{op}, \boldsymbol{Set}]$ correspond precisely to the elements of Fa. In other words,*

$$[\mathcal{C}^{op}, \boldsymbol{Set}](H_a, F) \cong Fa$$

Proof. The arrow $f : [\mathcal{C}^{op}, \mathbf{Set}](H_a, F) \to Fa$ is obvious by the Yoneda trick.

$$\alpha \mapsto \alpha_a(id_a) : [\mathcal{C}^{op}, \mathbf{Set}](H_a, F) \to Fa$$

For any $x \in Fa$, as we see before, we may expect there is a natural transformation α such that $\alpha_a(id_a) = x$, then the action on other objects is completely determined by $\alpha_a(id_a)$ as we see before.

$$x, g \mapsto (Fg)(x) : Fa \to [\mathcal{C}^{op}, \mathbf{Set}](H_a, F)$$

Note that $(F(id_a))(x) = id_{Fa}(x) = x$.

We need to show that they are inverse to each other. For any natural transformation α:

$$[\mathcal{C}^{op}, \mathbf{Set}](H_a, F) \xrightarrow{\quad f \quad} Fa \xrightarrow{\quad f^{-1} \quad} [\mathcal{C}^{op}, \mathbf{Set}](H_a, F)$$

$$\alpha \xmapsto{\quad -_a(id_a) \quad} \alpha_a(id_a) \xmapsto{\quad (F-)(-) \quad} (F-)(\alpha_a(id_a))$$

And we can see $(Fg)(\alpha_a(id_a)) = \alpha_x(g)$ for any $x \in \mathcal{C}$ and $g \in \mathcal{C}(x, a)$ by the same way as the proof which shows $H_\bullet$ is a full functor.

In another direction, we get:

$$Fa \xrightarrow{\quad f^{-1} \quad} [\mathcal{C}^{op}, \mathbf{Set}](H_a, F) \xrightarrow{\quad f \quad} Fa$$

$$x \xmapsto{\quad (F-)(-) \quad} (F-)(x) \xmapsto{\quad -_a(id_a) \quad} F(id_a)(x)$$

It is obvious that $F(id_a)x = id_{Fa}x = x$. $\square$

7 Adjoint

Theorem 7.1. *Show that the hom-functor preserves limit, that is, for any $Y \in \mathcal{C}$ and diagram $\mathcal{D}$, we have:*

$$\mathrm{Lim}(\mathcal{C}(Y, \mathcal{D}_-)) \cong \mathcal{C}(Y, \mathrm{Lim}\,\mathcal{D})$$

Proof. The idea comes from *The Dao of FP* and nlab.

We may consider the cone with singleton set as vertex:

$$\begin{tikzcd}
 & 1 \arrow[d, "u"] \arrow[ddl, bend right, "\mathrm{const}_{p_i}"'] \arrow[ddr, bend left, "\mathrm{const}_{p_j}"] & \\
 & \mathrm{Lim}(\mathcal{C}(Y, \mathcal{D}_-)) \arrow[dl] \arrow[dr] & \\
\mathcal{C}(Y, \mathcal{D}_i) \arrow[rr, "\mathcal{C}(Y, \mathcal{D}_f)"] & & \mathcal{C}(Y, \mathcal{D}_j)
\end{tikzcd}$$

where const_{p_i} is the function that takes a morphism $p_i \in \mathcal{C}(Y, \mathcal{D}_i)$.

We know there is a one-to-one corresponding between u and the pair $\langle \mathrm{const}_{p_i}, \mathrm{const}_{p_j} \rangle$:

$$[\mathcal{J}, \mathbf{Set}](\Delta_1, \mathcal{C}(Y, \mathcal{D}_-)) \cong \mathbf{Set}(1, \mathrm{Lim}(\mathcal{C}(Y, \mathcal{D}_-)))$$

or we can simplify the equation by defining $F_j = \mathcal{C}(Y, \mathcal{D}_j) : \mathcal{J} \to \mathbf{Set}$.

$$[\mathcal{J}, \mathbf{Set}](\Delta_1, F) \cong \mathbf{Set}(1, \mathrm{Lim}\, F)$$

We may recall that the hom-set maps out from 1 is isomorphic to the target, that is:

$$\mathbf{Set}(1, \mathrm{Lim}\, F) \cong \mathrm{Lim}\, F$$

Similarly, the cone with 1 as vertex is a pair of selection of $\mathcal{C}(Y, \mathcal{D}_i)$, it forms a cone with Y as vertex:

$$\begin{tikzcd}
 & Y \arrow[dl, bend right, "p_i"'] \arrow[dr, bend left, "p_j"] & \\
\mathcal{D}_i \arrow[rr, "\mathcal{D}_f"] & & \mathcal{D}_j
\end{tikzcd}$$

the diagram is indeed commute since

$$\begin{tikzcd}
 & 1 \arrow[dl, bend right, "\mathrm{const}_{p_i}"'] \arrow[dr, bend left, "\mathrm{const}_{p_j}"] & & & & 1 \arrow[dl, mapsto, bend right, "\mathrm{const}_{p_i}"'] \arrow[dr, mapsto, bend left, "\mathrm{const}_{p_j}"] & \\
\mathcal{C}(Y, \mathcal{D}_i) \arrow[rr, "\mathcal{C}(Y, \mathcal{D}_f)"] & & \mathcal{C}(Y, \mathcal{D}_j) & \Rightarrow & p_i \arrow[rr, mapsto, "\mathcal{D}_f \circ -"] & & p_j
\end{tikzcd}$$

Also we can make a pair of selection of $\mathcal{C}(Y, \mathcal{D}_-)$ from a cone with Y as vertex.

Then the cone of $\mathcal{C}(Y, \mathcal{D}_-)$ with vertex 1, is isomorphic to the cone of $\mathcal{D}_-$ with vertex Y:

$$[\mathcal{J}, \mathbf{Set}](\Delta_1, F) \cong [\mathcal{J}, \mathcal{C}](\Delta_Y, D)$$

While the later one is naturally isomorphic to the limit of $\mathcal{J}_\bullet$:

$$[\mathcal{J}, \mathcal{C}](\Delta_Y, D) \cong \mathcal{C}(Y, \operatorname{Lim} D)$$

Finally, we have:

$$\begin{gathered}
\operatorname{Lim} \mathcal{C}(Y, \mathcal{D}_-) \\
\cong \\
\mathbf{Set}(1, \operatorname{Lim}(\mathcal{C}(Y, \mathcal{D}_-))) \\
\cong \\
[\mathcal{J}, \mathbf{Set}](\Delta_1, \mathcal{C}(Y, \mathcal{D}_-)) \\
\cong \\
[\mathcal{J}, \mathcal{C}](\Delta_Y, D) \\
\cong \\
\mathcal{C}(Y, \operatorname{Lim} D)
\end{gathered}$$

□

Dually, we also have:

$$\operatorname{Lim}(\mathcal{C}(\mathcal{D}_-, Y)) \cong \mathcal{C}(\operatorname{Colim} \mathcal{D}, Y)$$

8 Monad

Some function in programming have *side effect*, for example:

- Partial: undefined, represent by Maybe.
- Exception:
- Nondetermini