Notes on Category Theory — Pieces

Indrjo Dedej

Last revision: 28th October 2023.

Contents

1	Adjointness	1
1.1	Definition	1
1.2	Units and counits	6
1.3	Triangle Identities	8
1.4	Adjunctions and limits	8
1.5	Exponentiation	10

1 Adjointness

1.1 Definition

Example 1 (Defining linear functions)**.** In Linear Algebra, we have a theorem which states linear functions are determined by the images of the elements of any base of the domain:

> Let V and W two vector spaces on some field k, with the first one having a base S; let us write i for the inclusion function $S \hookrightarrow V$. Then for every function $\phi : S \to W$ there exists one and only one linear function $f : V \to W$ such that
>
> $$\begin{tikzcd} S \arrow[r, hook, "i"] \arrow[dr, "\phi"'] & V \arrow[d, "f"] \\ & W \end{tikzcd}$$
>
> commutes.

That is the function

$$\begin{aligned} \mathbf{Vect}_k(V, W) &\to \mathbf{Set}(S, W) \\ f &\to fi \end{aligned} \tag{1.1}$$

is a bijection; in other words, this function post-composes linear functions with the inclusion of the base of the domain into the domain itself. Here, we write $\mathbf{Vect}_k$ and $\mathbf{Set}$ on purpose, because we want to walk a precise path. We have a function pointing to a vector space W, but set functions do not care about W being a vector space; instead, linear functions do! In some sense, in this example we see W 'downgraded' from the status of vector space to the one of barren set. On the other hand, from a set we construct an actual vector space — this is what being a base means. That being said, let us rearrange the diagram above:

$$\begin{tikzcd} S \arrow[r, "\text{generating}"] \arrow[d, "\phi"'] & V \arrow[d, "f"] \\ W & W \arrow[l, "\text{downgrading}"] \end{tikzcd}$$

where the horizontal arrows start from some category and land onto another one. The drawing just made is not meant to be a diagram in a strict sense, but an illustration on what is happening. If you are thinking the horizontal arrows are just a piece of a bigger picture, you are right: behind the scenes, two functors

$$\mathbf{Vect}_k \underset{G}{\overset{F}{\rightleftarrows}} \mathbf{Set}$$

are acting, where F is the functor that forgets and G the functor that takes sets and crafts a vector space from it and takes functions and gives linear functions. [Do I need to be more specific here? Perhaps, I may talk about such things elsewhere.] However, functors are a matter of morphisms too, so let us involve them too into this discourse. Let us call $\xi_{S,W}$ the function (1.1), as we will soon need thiss notation.

Take a function $\phi : S' \to S$ and a linear function $f : W \to W'$: in this case we have the function

$$\mathbf{Vect}_k(G(S), W) \to \mathbf{Vect}_k(G(S'), W')$$
$$g \to \left(fgG(\phi) : \begin{tikzcd} G(S) \arrow[r, "g"] & W \arrow[d, "f"] \\ G(S') \arrow[u, "G(\phi)"] & W' \end{tikzcd} \right)$$

The point is that we have a functor

$$\mathbf{Vect}_k(G(\quad), \quad) : \mathbf{Set}^{\mathrm{op}} \times \mathbf{Vect}_k \to \mathbf{Set}$$

that maps pairs (S, V) to sets $\mathbf{Vect}_k(G(S), V)$ and with respect to morphisms acts as just described above. It is not enough, giving the same ϕ and f the function

$$\mathbf{Set}(S, F(W)) \to \mathbf{Set}(S', F(W'))$$
$$\delta \to \left(F(f)\delta\phi : \begin{tikzcd} S \arrow[r, "\delta"] & F(W) \arrow[d, "F(f)"] \\ S' \arrow[u, "\phi"] & F(W') \end{tikzcd} \right)$$

and so a functor

$$\mathbf{Set}(\quad, F(\quad)) : \mathbf{Set}^{\mathrm{op}} \times \mathbf{Vect}_k \to \mathbf{Set}$$

mapping pairs (S, V) to sets $\mathbf{Set}(S, F(V))$ this time. That is we ended up with two functors

$$\mathbf{Set}^{\mathrm{op}} \times \mathbf{Vect}_k \to \mathbf{Set}.$$

Let us push our discourse a little further: the functions $\xi_{S,W}$, for S varying on sets and W on vector spaces over k, does form a natural isomorphism

$$\begin{tikzcd} \mathbf{Set}^{\mathrm{op}} \times \mathbf{Vect}_k \arrow[r, bend left, "\mathbf{Vect}_k(G(\quad),\quad)"] \arrow[r, bend right, "\mathbf{Set}(\quad,F(\quad))"'] \arrow[r, phantom, "\Downarrow\xi"] & \mathbf{Set} \end{tikzcd}$$

Observe, that being the $\xi_{S,W}$-s all isomorphisms, then we are done if we show that ξ is a natural transformation, viz

$$\begin{tikzcd}
\mathbf{Vect}_k(G(S), W) \arrow[r, "\xi_{S,W}"] \arrow[d, "\lambda h. fhG(\phi)"'] & \mathbf{Set}(S, F(W)) \arrow[d, "\lambda h. F(f)h\phi"] \\
\mathbf{Vect}_k(G(S'), W') \arrow[r, "\xi_{S',W'}"'] & \mathbf{Set}(S', F(W'))
\end{tikzcd}$$

commutes for every set S and S', vector space W and W', function $\phi : S' \to S$ and linear function $f : W \to W'$. To prove this, consider the inclusions $i : S \hookrightarrow G(S)$ and $i' : S' \hookrightarrow G(S')$. Thus

$$\left.\begin{matrix} (F(f)(hi)\phi)(v) \\ (fhG(\phi)i')(v) \end{matrix}\right\} = (f(h(\phi(v)))) \text{ for every } v \in S'$$

and we have concluded.

Exercise 2. In the chapter about functors we have dealt a bit of free stuff and functors that 'downgrade' objects and morphisms of a certain category to bare sets and functions respectively. For example, groups and homomorphisms are such as they live within **Grp**, but there is no place in **Set** for group structure (operation, identity and group axioms) and the property of preserving operations of homomorphisms.
In the chapter of limits and colimits, we have isolated universal properties enjoyed by those free stuff as initial objects in certain categories: pick some of those constructions and see if you can do something similar to the previous example. Preview: it is possible and the answer is in the next section, but now try and experiment a little.
This exercise is to get used to one specific pattern in doing things.

1. Given a set, prescribe a way to construct another object with structure. Luckily, it is not only about giving structure to sets but involves functions as well: hence functions of sets induce morphisms between the new objects. That is, the construction departs from **Set** and lands onto another category $\mathcal{E}$. The whole construction is a functor

$$\langle\cdot\rangle : \mathbf{Set} \to \mathcal{E}.$$

   If construction might be tricky sometimes, destruction is quite a rather simple task in comparison: that is we have a functor

$$U : \mathcal{E} \to \mathbf{Set}.$$

2. The situation now is two functors running in opposite directions

$$\begin{tikzcd}
\mathbf{Set} \arrow[r, shift left, "\langle\cdot\rangle"] & \mathcal{E} \arrow[l, shift left, "U"]
\end{tikzcd}$$

   The universal property enjoyed by the free objects in question gives a way to define a bijection

$$\mathbf{Set}(S, U(H)) \cong \mathcal{E}(\langle S\rangle, H),$$

one for each S and H. However, it is not just about bijections: the leap now is they form a natural isomorphism

$$\begin{tikzcd} \mathbf{Set}^{\mathrm{op}} \times \mathcal{E} \arrow[r, bend left, "\mathcal{E}(\langle\cdot\rangle,\ )"] \arrow[r, bend right, "\mathbf{Set}(\cdot, U(\ ))"'] \arrow[r, phantom, "\Downarrow"] & \mathbf{Set} \end{tikzcd}$$

Of course, the last point is the hardest part of the work, since it requires you to understand another actor. Although, considering another similar examples, you will realise that it is one way things are to be done.
One 'paedagogical' note: do not force yourself to follow the steps as enumerated above. The process is often triggered as one comes up with bijections like in the third point, and afterwards they work to construct suitable functors to make all the pieces of a wide puzzle fit to each other. Just take your time.

Example 3 (Prescribing functions via currying)**.** [Yet to be TEXed...]

Let us explicitly state all the concepts. Let $\mathcal{C}$ and $\mathcal{D}$ be two locally small categories and

$$\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "L"] & \mathcal{D} \arrow[l, bend left, "R"] \end{tikzcd}$$

two functors. We have then the functor

$$\hom_{\mathcal{C}}(\ \ , R(\ \ )) : \mathcal{C}^{\mathrm{op}} \times \mathcal{D} \to \mathbf{Set}$$

that maps objects (x, y) to $\hom_{\mathcal{C}}(x, R(y))$ and pairs of morphisms

$$\left(\begin{array}{c} (x,y) \\ {\scriptstyle (f,g)}\big\downarrow \\ (x',y') \end{array} \right) = \left(\begin{array}{ccc} x & & y \\ {\scriptstyle f}\big\uparrow & , & \big\downarrow{\scriptstyle g} \\ x' & & y' \end{array} \right)$$

to functions

$$\begin{aligned} \hom_{\mathcal{C}}(x, R(y)) &\to \hom_{\mathcal{C}}(x', R(y')) \\ h &\to R(g)hf \end{aligned}$$

We have also the functor

$$\hom_{\mathcal{D}}(L(\ \ ),\ \ ) : \mathcal{C}^{\mathrm{op}} \times \mathcal{D} \to \mathbf{Set}$$

that maps (x, y) to $\hom_{\mathcal{C}}(L(x), y)$ and pairs of morphisms

$$\left(\begin{array}{c} (x,y) \\ {\scriptstyle (f,g)}\big\downarrow \\ (x',y') \end{array} \right) = \left(\begin{array}{ccc} x & & y \\ {\scriptstyle f}\big\uparrow & , & \big\downarrow{\scriptstyle g} \\ x' & & y' \end{array} \right)$$

to functions

$$\begin{aligned} \hom_{\mathcal{D}}(L(x), y) &\to \hom_{\mathcal{D}}(L(x'), y') \\ h &\to ghL(f). \end{aligned}$$

Exercise 4. The functors just mentioned can be defined as composition of others, one of which is already known. We recall it here. For if $\mathcal{C}$ is a locally small category, the functor

$$\hom_{\mathcal{C}} : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathbf{Set}$$

described as follows:

- objects (x, y) are mapped into sets $\hom_{\mathcal{C}}(x, y)$;
- morphisms $(f, g) : (x, y) \to (x', y')$ of $\mathcal{C}^{\mathrm{op}} \times \mathcal{C}$, viz pairs

$$\left(\begin{array}{c} x \\ {\scriptstyle f}\big\uparrow \\ x' \end{array} , \begin{array}{c} y \\ \big\downarrow{\scriptstyle g} \\ y' \end{array} \right)$$

with the first morphism regarded as one of $\mathcal{C}$, into functions

$$\begin{aligned} \hom_{\mathcal{C}}(x, y) &\to \hom_{\mathcal{C}}(x', y') \\ h &\to ghf \end{aligned}$$

Now, suppose given two functors

$$F_1 : \mathcal{C}_1 \to \mathcal{D}_1 \text{ and } F_2 : \mathcal{C}_2 \to \mathcal{D}_2.$$

We define the *product functor*

$$F_1 \times F_2 : \mathcal{C}_1 \times \mathcal{C}_2 \to \mathcal{D}_1 \times \mathcal{D}_2$$

as follows: it maps objects (a_1, a_2) to $(F_1(a_1), F_2(a_2))$ and morphisms

$$\left(\begin{array}{c} x_1 \\ {\scriptstyle f_1}\big\downarrow \\ y_1 \end{array} , \begin{array}{c} x_2 \\ {\scriptstyle f_2}\big\downarrow \\ y_2 \end{array} \right)$$

into

$$\left(\begin{array}{c} F_1(x_1) \\ {\scriptstyle F_1(f_1)}\big\downarrow \\ F_1(y_1) \end{array} , \begin{array}{c} F_2(x_2) \\ {\scriptstyle F_2(f_2)}\big\downarrow \\ F_2(y_2) \end{array} \right)$$

Definition 5 (Adjunctions)**.** Consider two locally small categories $\mathcal{C}$ and $\mathcal{D}$ and two functors $\mathcal{C} \underset{R}{\overset{L}{\rightleftarrows}} \mathcal{D}$. An *adjunction* from L to R is a natural isomorphism

$$\begin{tikzcd} \mathcal{C}^{\mathrm{op}} \times \mathcal{D} \arrow[r, bend left, "\hom_{\mathcal{C}}(-, R(\cdot))"] \arrow[r, bend right, "\hom_{\mathcal{D}}(L(-), \cdot)"'] \arrow[r, phantom, "\Downarrow\alpha"] & \mathbf{Set} \end{tikzcd}.$$

We write such natural isomorphism as $\alpha : L \dashv R$, and say L is the *left adjoint*, whereas R is the *right* one.

We say L is the 'left' adjoint and R is the 'right' one because when we write the bijections

$$\mathcal{D}(L(x), y) \cong \mathcal{C}(x, R(y))$$

L is applied to the left argument in $\mathcal{D}(L(x), y)$ whereas R is applied to the right argument in $\mathcal{C}(x, R(y))$.

Exercise 6 (Inspired by Haskell[1])**.** [Borrow some Haskell notation?] We define the category of partial functions, written as **Par**. Here objects are sets and morphisms are partial functions. For A and B sets, a *partial function* from A to B is relation $f \subseteq A \times B$ with this property:

1. A programming language. It is not bad you know something about it.

for every $x \in A$ and $y_1, y_2 \in B$, if $(x, y_1) \in f$ and $(x, y_2) \in f$ then $y_1 = y_2$.

We want to compose partial functions as well: provided $f \in \mathbf{Par}(A, B)$ and $g \in \mathbf{Par}(B, C)$,

$$gf := \{(x, y) \in A \times C \mid (x, z) \in f \text{ and } (z, y) \in g \text{ for some } z \in B\}\,.$$

It is immediate to verify **Par** complies the rules that make it a category.
The thing important here is this: suppose given a partial function $f : A \to B$, every $x \in A$ may have one element of B bound — in this case, we write it $f(x)$ — or none. The key of the exercise is: what if we considered 'no value' as an admissible output value? Provided two sets A and B and a partial function $f : A \to B$, we assign an actual function

$$\overline{f} : A \to B + 1\,,\ \overline{f}(x) := \begin{cases} f(x) & \text{if } x \text{ has an element of } B \text{ bound} \\ * & \text{otherwise} \end{cases}$$

where $1 := \{*\}$ with $*$ designating the absence of output. It is quite simple to show that

$$\mathbf{Par}(A, B) \to \mathbf{Set}(A, B + 1)\,,\ f \to \overline{f}$$

is a bijection for every couple of sets A and B. Now it's up to you to categorify this by considering two suitable functors

$$\mathbf{Set} \underset{\texttt{Maybe}}{\overset{I}{\rightleftarrows}} \mathbf{Par}\ .$$

It should be simple to guess how is defined I. As for `Maybe`, you do not need to know Haskell: if you do, fine, otherwise you are learning something new.

1.2 Units and counits

In the first example of the introduction, we isolated the concept of adjunction from that one of initial objects of certain categories: We now isolate and formalize this process. We will do the converse too. As result, we end up having two equivalent ways to work with adjointness.

Proposition 7. Suppose given two locally small categories $\mathcal{C}$ and $\mathcal{D}$, two functors

$$\mathcal{C} \underset{R}{\overset{L}{\rightleftarrows}} \mathcal{D}$$

and a natural transformation $\eta : 1_{\mathcal{C}} \Rightarrow RL$ such that $\eta_x : x \to RL(x)$ is initial in $x{\downarrow}R$ [did we introduce comma categories?] for every $x \in |\mathcal{C}|$. Then, for $x \in |\mathcal{C}|$ and $y \in |\mathcal{D}|$, the functions

$$\begin{aligned} \mathcal{D}(L(x), y) &\to \mathcal{C}(x, R(y)) \\ f &\to R(f)\eta_x \end{aligned}$$

form an adjunction $L \dashv R$.

Proof. The fact that η_x is initial object implies that these function are all bijective. Now, we just need to verify the transformation is natural. Take $x, x' \in |\mathcal{C}|$, $y, y' \in |\mathcal{D}|$, $f \in \mathcal{C}(x', x)$ and $g \in \mathcal{D}(y, y')$ and examine the square

$$\begin{tikzcd} \mathcal{D}(L(x), y) \arrow[r, "u \to R(u)\eta_x"] \arrow[d, "u \to guL(f)"'] & \mathcal{C}(x, R(y)) \arrow[d, "v \to R(g)vf"] \\ \mathcal{D}(L(x'), y') \arrow[r, "v \to R(v)\eta_{x'}"'] & \mathcal{C}(x', R(y')) \end{tikzcd}$$

Taken $u \in \mathcal{D}(L(x), y)$, we perform the following calculations

$$\begin{aligned} R(g)R(u)\eta_x f &= R(gu)\eta_x f \\ R(guL(f))\eta_{x'} &= R(gu)RL(f)\eta_{x'} \end{aligned}$$

By the naturality of η, we have $\eta_x f = RL(f)\eta_{x'}$, and thus the construction ends here. □

Proposition 8. Suppose now you have locally small categories $\mathcal{C}$ and $\mathcal{D}$, functors

$$\begin{tikzcd} \mathcal{C} \arrow[r, shift left, "L"] & \mathcal{D} \arrow[l, shift left, "R"] \end{tikzcd}$$

and an adjunction $L \dashv R$. For $x \in |\mathcal{C}|$ write $\eta_x : x \to RL(x)$ the morphism in $\mathcal{C}$ corresponding to $1_{L(x)}$ of $\mathcal{D}$. Then the morphisms $\eta_x : x \to RL(x)$ form a natural transformation $\eta : 1_{\mathcal{C}} \Rightarrow RL$. Moreover, η_x is initial in $x{\downarrow}R$.

Proof. Let us write the adjunction of the statement above as

$$\overline{} : \mathcal{D}(L(\ \ ),\ \ ) \Rightarrow \mathcal{C}(\ \ , R(\ \ )).$$

We verify that

$$\begin{tikzcd} x \arrow[r, "\eta_x"] \arrow[d, "f"'] & RL(x) \arrow[d, "RL(f)"] \\ y \arrow[r, "\eta_y"'] & RL(y) \end{tikzcd}$$

commutes for every f in $\mathcal{C}$. In fact,

$$\begin{aligned} RL(f)\eta_x &= RL(f)\overline{1_{L(x)}}1_x = \overline{L(f)1_{L(x)}1_{L(x)}} = \overline{L(f)} \\ \eta_y f &= R(1_{L(y)})\overline{1_{L(y)}}f = \overline{1_{L(y)}1_{L(y)}L(f)} = \overline{L(f)}. \end{aligned}$$

It remains to show that the morphisms $\eta_x : x \to RL(x)$ are initial in $x{\downarrow}R$. In $\mathcal{C}$ we draw

$$\begin{tikzcd} x \arrow[r, "\eta_x"] \arrow[dr, "g"'] & RL(x) \\ & R(y) \end{tikzcd}$$

We know that there is one and only one $h : L(x) \to y$ such that $g = \overline{h}$. Then

$$g = \overline{h1_{L(x)}L(1_x)} = R(h)\overline{1_{L(x)}}1_x = R(h)\eta_x. \qquad \square.$$

[Co-units version.]

Proposition 9. Suppose given two locally small categories $\mathcal{C}$ and $\mathcal{D}$, two functors

$$\begin{tikzcd} \mathcal{C} \arrow[r, shift left, "L"] & \mathcal{D} \arrow[l, shift left, "R"] \end{tikzcd}$$

and a natural transformation $\theta : LR \Rightarrow 1_{\mathcal{D}}$ such that $\theta_y : LR(y) \to y$ is terminal in $L{\downarrow}y$ for every $y \in |\mathcal{C}|$. Then, for $x \in |\mathcal{C}|$ and $y \in |\mathcal{D}|$, the functions

$$\begin{aligned} \mathcal{C}(x, R(y)) &\to \mathcal{D}(L(x), y) \\ f &\to \theta_y L(f) \end{aligned}$$

form an adjunction $L \dashv R$.

Proposition 10. Suppose now you have locally small categories $\mathcal{C}$ and $\mathcal{D}$, functors

$$\mathcal{C} \underset{R}{\overset{L}{\rightleftarrows}} \mathcal{D}$$

and an adjunction $L \dashv R$. For $y \in |\mathcal{C}|$ write $\theta_y : LR(y) \to y$ the morphism in $\mathcal{D}$ corresponding to $1_{R(y)}$ of $\mathcal{C}$. Then the morphisms θ_y form a natural transformation $\theta : LR \Rightarrow 1_{\mathcal{D}}$. Moreover, θ_y is terminal in $L{\downarrow}y$.

Exercise 11. Prove the last two theorems.

[Yet there is something left to say...]

1.3 Triangle Identities

[Yet to be TEXed...]

1.4 Adjunctions and limits

Let $\mathcal{I}$ and $\mathcal{C}$ be two categories. For every $v \in |\mathcal{C}|$ we have the *constant functor*

$$k_v : \mathcal{I} \to \mathcal{C}$$

where $k_v(i) := v$ for every $i \in |\mathcal{I}|$ and $k_v(f) := 1_v$ for every morphism f of $\mathcal{I}$. Recall that $\lambda : k_v \Rightarrow F$ being a limit of a functor $F : \mathcal{I} \to \mathcal{C}$ means:

> for every $\mu : k_v \Rightarrow F$ there exists one and only one $f : a \to v$ of $\mathcal{C}$ such that $\mu_i = \lambda_i f$ commutes for every object i of $\mathcal{I}$.

That is, if you put it in other words, it sounds like:

> there is a bijection
> $$\mathcal{C}(a, v) \to [\mathcal{I}, \mathcal{C}](k_a, F)$$
> taking $f : a \to v$ to the natural transformation
> $$\lambda_\bullet f : k_a \Rightarrow F\,,\ \lambda_\bullet f(i) := \lambda_i f.$$

There is a smell of adjunction situation here. Let us start with finding an appropriate pair of functors

$$\mathcal{C} \rightleftarrows [\mathcal{I}, \mathcal{C}]\ .$$

One functor is already suggested:

$$\Delta : \mathcal{C} \to [\mathcal{I}, \mathcal{C}]$$

takes $x \in |\mathcal{C}|$ to the functor $\mathcal{I} \to \mathcal{C}$ that maps every object to x and every morphism to 1_x; then for $i \in |\mathcal{I}|$ define

$$\Delta\left(x \xrightarrow{f} y\right)$$

to be the natural transformation $\Delta(x) \Rightarrow \Delta(y)$ amounting uniquely of f.

From now on, assume $\mathcal{I}$ is small and every functor $\mathcal{I} \to \mathcal{C}$ has a limit. Now, in spite of not being strictly unique [‘strictly unique’... huh?], all the limits of a given functor are isomorphic, so are the vertices: let us indicate by $\lim F$ the vertex of any of the limits of F. Now, take a natural transformation

$$\begin{tikzcd} \mathcal{I} \arrow[r, bend left, "F"] \arrow[r, bend right, "G"'] \arrow[r, phantom, "\Downarrow\xi"] & \mathcal{C} \end{tikzcd} ;$$

$\lim F$ is the vertex of some limit

$$\left\{ \lim F \xrightarrow{\lambda_i} F(i) \,\middle|\, i \in |\mathcal{I}| \right\}$$

and $\lim G$ is the vertex of a certain limit

$$\left\{ \lim G \xrightarrow{\mu_i} G(i) \,\middle|\, i \in |\mathcal{I}| \right\}.$$

If we display all the stuff we have gathered so far, we have for $i \in |\mathcal{I}|$

$$\begin{tikzcd} F(i) \arrow[r, "\xi_i"] & G(i) \\ \lim F \arrow[u, "\lambda_i"] & \lim G \arrow[u, "\mu_i"'] \end{tikzcd} \tag{1.2}$$

The universal property of limits ensures that there is one and only one morphism $\lim F \to \lim G$ making the above diagram a commuting square. Let us call this morphism $\lim \xi$. We have a functor

$$\lim : [\mathcal{I}, \mathcal{C}] \to \mathcal{C}$$

indeed. If you take $F = G$ and η the identity of the functor F in (1.2), then

$$\lim 1_F = 1_{\lim F},$$

obtained by uniquely employing the universal property of limit. Now take three functors $F, G, H : \mathcal{I} \to \mathcal{C}$ and two natural transformations $F \xRightarrow{\alpha} G \xRightarrow{\beta} H$. To these functors are associated the respective limits

$$\left\{ \lim F \xrightarrow{\lambda_i} F(i) \,\middle|\, i \in |\mathcal{I}| \right\}$$

$$\left\{ \lim G \xrightarrow{\mu_i} G(i) \,\middle|\, i \in |\mathcal{I}| \right\}$$

$$\left\{ \lim H \xrightarrow{\eta_i} H(i) \,\middle|\, i \in |\mathcal{I}| \right\}$$

so that we have commuting squares glued together:

$$\begin{tikzcd} F(i) \arrow[r, "\alpha_i"] & G(i) \arrow[r, "\beta_i"] & H(i) \\ \lim F \arrow[u, "\lambda_i"] \arrow[r, "\lim \alpha"'] & \lim G \arrow[u, "\mu_i"] \arrow[r, "\lim \beta"'] & \lim H \arrow[u, "\eta_i"'] \end{tikzcd}$$

We have for every $i \in |\mathcal{I}|$

$$\eta_i \lim \beta \lim \alpha = \beta_i \mu_i \lim \alpha = \beta_i \alpha_i \lambda_i;$$

then, by how it is defined the limit of a natural transformation, it must be

$$\lim(\beta\alpha) = \lim \beta \lim \alpha.$$

The following proposition pushes all this discourse to a conclusion.

Proposition 12. There is an adjunction

$$\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "\Delta"] \arrow[r, phantom, "\perp"] & {[\mathcal{I},\mathcal{D}]} \arrow[l, bend left, "\lim"] \end{tikzcd}$$

Proof. [Yet to be TEX-ed...] □

1.5 Exponentiation

Let $\mathcal{C}$ be a category with binary products. Consider one object c of $\mathcal{C}$. For a object in $\mathcal{C}$ choose

$$\begin{tikzcd} & a \times c \arrow[dl, "p_a"'] \arrow[dr, "q_a"] & \\ a & & c \end{tikzcd}$$

to be any of the products of a and c. That being said, we will work now to construe a functor

$$(\times c) : \mathcal{C} \to \mathcal{C}.$$

As the notation hints, we make the convention

$$(\times c)(x) := x \times c$$

for x object of $\mathcal{C}$. [In the zeroth chapter, remember to introduce this kind of notation.] Now, we shall involve morphisms too. [In the section of products, remember to talk about the the product of morphisms.] If we take any $f : a \to b$ of $\mathcal{C}$, then we instruct $(\times c)$ on morphisms as follows:

$$(\times c)(f) := f \times 1_c.$$

[In the section of products, remember to talk about the the product of morphisms.] Functoriality, in this case, directly descends from what we have said in the section about products. [Remember to TEX that part too.]

Definition 13 (Exponential object)**.** In the category $\mathcal{C}$ that has binary products, the *exponential object* of two objects a and b of $\mathcal{C}$ is any

- object of $\mathcal{C}$, we write as b^a
- a morphism $\mathrm{ev} : b^a \times a \to b$, the *evaluation*

such that ev is a terminal object of $(\times a) \downarrow b$. A category $\mathcal{C}$ is said to 'have exponentials' whenever for every $a, b \in |\mathcal{C}|$ there is in $\mathcal{C}$ the corresponding exponential object.

We can involve adjunctions in this discourse now! In fact, the definition gives a bijection

$$\mathcal{C}(a \times c, b) \cong \mathcal{C}\left(a, b^c\right)$$

for $a, b, c \in |\mathcal{C}|$. Thus, first of all, we shall try to arrive to a situation like this

$$\begin{tikzcd} \mathcal{C} \arrow[r, shift left, "(\times c)"] & \mathcal{C} \arrow[l, shift left, "?"] \end{tikzcd}$$

so that we can see if $(\times c) \dashv ?$.
The functor labelled with a question mark comes from a close analysis of the the new objects just introduced. Consider a morphism $f : a \to b$ and a third

object c of $\mathcal{C}$. Assume also, $\mathcal{C}$ has exponentials. If we write ev_a and ev_b for the evaluations associated to a^c and b^c, we we have

$$\begin{tikzcd} a^c \times c \arrow[d, "\mathrm{ev}_a"'] & b^c \times c \arrow[d, "\mathrm{ev}_b"] \\ a \arrow[r, "f"'] & b \end{tikzcd}.$$

Composing ev_a and f and the definition of exponential objects yield a unique morphism $a^c \to b^c$ of $\mathcal{C}$, we denote f^c, making

$$\begin{tikzcd} a^c \times c \arrow[r, "f^c \times 1_c"] \arrow[d, "\mathrm{ev}_a"'] & b^c \times c \arrow[d, "\mathrm{ev}_b"] \\ a \arrow[r, "f"'] & b \end{tikzcd}$$

commute. Indeed, we have the functor $\square^c : \mathcal{C} \to \mathcal{C}$ with

$$\square^c(x) := x^c$$

for x object of $\mathcal{C}$ and

$$\square^c(f) := f^c$$

for $f : a \to b$ in $\mathcal{C}$.

Exercise 14. Under the hypothesis that $\mathcal{C}$ has exponentials, you can provide functors $c^\bullet : \mathcal{C}^{\mathrm{op}} \to \mathcal{C}$, using exponential objects. This exercise is not essential for this section.

The conclusion is the following theorem.

Proposition 15. $(\times c) \dashv \square^c$ for every $c \in |\mathcal{C}|$.

Proof. There is not much work left to do: we have two functors running in opposite directions and we have bijections $\mathcal{C}(a \times c, b) \to \mathcal{C}(a, b^c)$, one for every $c \in |\mathcal{C}|$; we have just to verify the naturality condition. □

Proposition 16. A category $\mathcal{C}$ with binary products has exponentials if and only if for every $c \in |\mathcal{C}|$ the functor $(\times c) : \mathcal{C} \to \mathcal{C}$ has a right adjoint. [More details...]

Proof. At this point, one of the implications is already demonstrated. The remaining can be readily derived using the constructions of the section about units and counits. [That section is a mess...] Indeed, the evaluation morphisms

$$\mathrm{ev} : a^c \times c \to a$$

are terminal objects of $[(\times c) \circ \square^c] \downarrow a$ and form a natural transformation

$$(\times c) \circ \square^c \Rightarrow 1_{\mathcal{C}}.$$

[More details...] [Make additions to the section about units and co-units.] □

[And now... Cartesian closed categories?]