Adjointness

Indrjo Dedej

Last revision: 7th June 2022.

Contents

1 Definition

Example 1.1 (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

$$\begin{tikzcd} \mathbf{Vect}_k \arrow[r, shift left, "F"] & \mathbf{Set} \arrow[l, shift left, "G"] \end{tikzcd}$$

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

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

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

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

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

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

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.

I'd love you to experiment by yourself a bit more before reading the definition of adjunction. The following exercise is already started, so you are 'just' required to finish it. Do not worry if you do not know how to go, read the definition and examine the other examples and return to this exercise. Any choice you do, it is worth to give it it a try now.

Exercise 1.2 (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:

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.

Now it is time to isolate the concept we have envisaged in the previous examples. Let $\mathcal{C}$ and $\mathcal{D}$ be two locally small [or something more?] categories and

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

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

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

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

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

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

Exercise 1.3. 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

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

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

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

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 1.4 (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.

[This part needs to be fixed.] Keeping the notation of the definition above, for $a \in |\mathcal{C}|$, $b \in |\mathcal{D}|$ and $f \in \hom_{\mathcal{C}}(a, R(b))$ we write $\alpha(f)$ for that element of $\hom_{\mathcal{D}}(L(a), b)$ which corresponds to f via the adjunction α. Since α is an isomorphism, it has the inverse $\alpha^{-1} : \hom_{\mathcal{D}}(L(a), b) \to \hom_{\mathcal{C}}(a, R(b))$: again, here $\alpha^{-1}(g)$ for $g \in \hom_{\mathcal{D}}(L(a), b)$ denotes the corresponding morphism via α^{-1}. Actually, the point is there is no preferred direction between the twos mentioned. In fact, for the most general purposes, more drastic conventions can be taken, that is using the same symbol for both α and α^{-1}:

$$\overline{a \xrightarrow{f} R(b)} = \left(L(a) \xrightarrow{\overline{f}} b \right), \quad \overline{L(a) \xrightarrow{g} b} = \left(a \xrightarrow{\overline{g}} R(b) \right).$$

Just pay attention about where things come from and where they are to go! Albeit this requires some caution, this notation has the great advantage that

$$\overline{\overline{p}} = p \text{ for every } p.$$

Now, we can explicitly express the naturality of the adjunction:

$$\overline{R(g)pf} = g\overline{p}L(f) \tag{1.2}$$

where $x, x' \in |\mathcal{C}|$, $y, y' \in |\mathcal{D}|$, $f \in \hom_{\mathcal{C}}(x', x)$, $g \in \hom_{\mathcal{D}}(y, y')$ and $p \in \hom_{\mathcal{C}}(x, R(y))$; equivalently,

$$\overline{gqL(f)} = R(g)\overline{q}f \tag{1.3}$$

where here $q \in \hom_{\mathcal{D}}(L(x), y)$.

2 Adjunctions, units and co-units

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 2.1. 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 $x{\downarrow}R$ for every $x \in |\mathcal{C}|$. Then, for $x \in |\mathcal{C}|$ and $y \in |\mathcal{D}|$, the functions

$$\mathcal{D}(Lx, y) \to \mathcal{C}(x, Ry)\,,\ f \to R(f)\eta_x$$

form a natural transformation $\mathcal{D}(L(\ \ ),\ \ ) \Rightarrow \mathcal{C}(\ \ , R(\ \ ))$.

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

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

By naturality $\eta_x f = RL(f)\eta_{x'}$, so the construction ends here. □

Proposition 2.2. Suppose given locally small categories $\mathcal{C}$ and $\mathcal{D}$, two functors $\mathcal{C} \underset{R}{\overset{L}{\rightleftarrows}} \mathcal{D}$ and an adjunction $\alpha : L \dashv R$. Then $\eta : 1_{\mathcal{C}} \Rightarrow RL$ is natural and $\eta_x : x \to RL(x)$ is initial in $x{\downarrow}R$ for every $x \in |\mathcal{C}|$.

Proposition 2.3. 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 $\alpha : L \dashv R$.Then the morphisms of $\mathcal{C}$

$$\eta_x : x \to RL(x)\,,\ \eta_x := \alpha\left(1_{L(x)}\right) \quad \text{where } x \in |\mathcal{C}|$$

do form a natural transformation $\eta : 1_{\mathcal{C}} \Rightarrow RL$; moreover, η_x is initial in $x{\downarrow}R$.

Proof. Yet to be TEX-ed... □

3 From units, counits and triangle identities to adjunctions

4 Limits from the view of adjunctions

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 being $\lambda : k_v \Rightarrow F$ 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

$$\begin{tikzcd} \mathcal{I} \arrow[r, bend left, "F"{name=U}] \arrow[r, bend right, "G"'{name=D}] & \mathcal{C} \arrow[from=U, to=D, Rightarrow, "\xi"] \end{tikzcd} \;;$$

$\lim F$ 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 $i \in |\mathcal{I}|$

$$\begin{tikzcd} F(i) \arrow[r, "\xi_i"] & G(i) \\ \lim F \arrow[u, "\lambda_i"] & \lim G \arrow[u, "\mu_i"'] \end{tikzcd} \tag{4.4}$$

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 (4.4), 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 $F \overset{\alpha}{\Longrightarrow} G \overset{\beta}{\Longrightarrow} H$. 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 4.1. 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...] □