Notes on Category Theory

Indrjo Dedej

Last revision: 5th October 2023.

Contents

0

Preamble

0.1 Symbolism for functions

If f is the name of a function, we write $f(x)$ the image of x. However, we may find ourselves writing fx or f_x to avoid an overwhelming use of brackets. The expression $f : X \to Y$ means that f is a function from the domain X to the codomain Y.

A literary device we will quite often take advantage is that of *currying*: from a function out of a product

$$f : A \times B \to C$$

we have, for $x \in A$, the functions

$$f(x, \quad) : B \to C\,, \; f(x, \quad)(y) := f(x, y).$$

The idea is to 'hold' the first variable at some value and let the second one vary: this is done by leaving a blank space to be filled with values from B. Of course, for $y \in B$, we introduce the functions

$$f(\quad, y) : A \to C\,, \; f(\quad, y)(x) := f(x, y).$$

Symbols like $\bullet$, $\cdot$ and $-$ can be employed instead of leaving an empty space: in fact, you may find written $f(x, \bullet)$, $f(x, \cdot)$ or $f(x, -)$ for example.

While leaving blank spaces or using placeholders may be acceptable in prefix notation, it may be a pain if used in combination with infix notation: consider, for example, the function $\mathbb{N} \to \mathbb{N}$ that takes each natural number to the corresponding successor and writing it as follows

$$\bullet + 1\,, \; \cdot + 1\,, \; - + 1\,, \; \ldots$$

or even

$$\underbrace{\quad}_{\text{blank space here, see?}} + 1.$$

In all those situations, we may forget placeholders and use parentheses as in

$$(+1)$$

and $(+1)(n)$ would be a synonym of $n + 1$.

Another way to introduce functions comes from Lambda Calculus. Suppose you are given some well-formed formula[1] Γ and some variable x: the symbol x may occur or not in Γ. Then we have

$$\lambda x.\Gamma$$

1. Written in a sensible way, that is following some syntactic rules. [More precision?]

called *lambda abstraction*. When we write an expression like this one, x is a *dummy* or a *bound* variable; in contrast, the other variables in Γ are said to be *free*. It works like the more familiar expressions

$$\forall x : \phi,\ \exists x : \phi,\ \lim_{x\to c} f(x) \text{ and } \int_\Omega f(x)\mathrm{d}x :$$

instead of the symbol x you may use another symbol that does not occur freely Γ without changing the meaning of the expression.

For example, we all agree that $\lambda x.x + k$ and $\lambda y.y + k$ have the same exact meaning, thus they are equivalent. But, what if instead of x we use k? In the formula $\lambda x.x + k$ the letter k already appears free, and we would have $\lambda k : k + k$, which is not the same.

In fact, the most basic operation you can perform with formulas involving variables is *substitution*: if you are given a formula Γ and a variable x that occurs freely in Γ, then we write $\Gamma[x/a]$ the formula Γ with the occurrences of x replaced by a, after eventually renaming all the dummy occurrences of a throughout the formula to avoid the issues illustrated before.

The operation of substitution is the way you can 'pass values' to such things and have returned another values:

$$(\lambda x.\Gamma)(a) := \Gamma[x/a].$$

It is clear the practical use of this formal device: if a is an element of set X and $(\lambda x.\Gamma)\,(a)$ is a member of another set Y, then we could write the function doing that assignment as $\lambda x.\Gamma$.

If you want, instead of $\lambda x.\Gamma$ you may write

$$x \mapsto \Gamma.$$

Observe that, in general, lambda abstractions do not have incorporated the information of domain and codomain, or in general it might not be inferred without doubt from the context. For example, what is $\lambda x.x + 1$? Without a hint from the context, it can be a function $\mathbb{R} \to \mathbb{R}$ or the successor function $\mathbb{N} \to \mathbb{N}$, and so on... To dispel any ambiguity, you can write explicitly something like this:

$$(\lambda n.n + 1) : \mathbb{N} \to \mathbb{N}.$$

0.2 Functions and equivalences

If X is a set and $\sim$ is an equivalence relation on X, we write $X/\!\sim$ or $\frac{X}{\sim}$ to indicate the set whose members are the sets

$$[x]_\sim := \{a \in X \mid a \sim x\} \quad \text{for } x \in X,$$

the equivalence classes under $\sim$. Sometimes we simply write $[x]$ when the name of the equivalence relation may be dropped without creating ambiguity.

Remember that equivalence classes form a partition, that is they are pairwise disjoint and their union is the entire set.

Annexed to that, there is the *canonical projection*

$$X \to X/\!\sim,\ x \to [x]_\sim.$$

Proposition 0.2.1 (Isomorphism Theorem of Set Theory)**.** Consider two sets X and Y, a function $f : X \to Y$ and an equivalence relation $\sim$ over X. Let also be $p : X \to X/{\sim}$ the canonical projection. If for every $a, b \in X$ such that $a \sim b$ we have $f(a) = f(b)$, then there exists one and only one function $\overline{f} : X/{\sim} \to Y$ such that

$$\begin{tikzcd} X \arrow[rr, "f"] \arrow[dr, "p"'] & & Y \\ & X/{\sim} \arrow[ur, "\overline{f}"'] & \end{tikzcd}$$

commutes. Moreover:

1. $\overline{f}$ is surjective if and only if so is f;
2. if also $a \sim b$ for every $a, b \in X$ such that $f(a) = f(b)$, then $\overline{f}$ is injective.

The name of this theorem will make you remember other isomorphism theorems. In Algebra, the *First Isomorphism Theorem* can be derived from this one. Consider, for example[2], two groups G and H a group homomorphism $f : G \to H$ and a normal subgroup N of G contained in $\ker f$. In that case, we have the equivalence relation $\sim_N$ on G defined by

$$x \sim_N y \quad \text{if and only if} \quad xa = y \text{ for some } a \in N$$

for $x, y \in G$. Further, being N normal in G, the set $G/N := G/{\sim_N}$ has as equivalence classes the lateral classes xN and has a group structure where the identity is N and the product of two lateral classes xN and yN is the lateral class $(xy)N$. In this case we have, if $x \sim_N y$, that is $xa = y$ for some $a \in N$,

$$f(y) = f(xa) = \underbrace{f(x)f(a) = f(x)}_{a \in N \subseteq \ker f}.$$

Hence, there is a unique function $\overline{f} : G/N \to H$ for which

$$\begin{tikzcd} G \arrow[rr, "f"] \arrow[dr, "p"'] & & H \\ & G/N \arrow[ur, "\overline{f}"'] & \end{tikzcd}$$

commutes. The function $\overline{f}$ is a group homomorphism and there are some other facts, but that takes us a bit away from the main topic of the section.

Exercise 0.2.2. Con you state something like that but for topological spaces and continuous functions? Remember the canonical projection $p : X \to X/{\sim}$ defines the topology for $X/{\sim}$ in which $U \subseteq X/{\sim}$ is open whenever so is $p^{-1}U$. (Here, $p^{-1}U$ is just the union of the equivalence classes in U.)

Proof of Proposition 0.2.1. Consider the relation

$$\overline{f} := \{(u, v) \in (X/{\sim}) \times Y \mid p(x) = u \text{ and } f(x) = v \text{ for some } x \in X\}:$$

we will show that it is actually a function from $X/{\sim}$ to Y. Picked any $u \in X/{\sim}$ (it is not empty), there is some $x \in u$ and then we have the element $f(x) \in Y$; in this case, $(u, f(x)) \in \overline{f}$. Now, let (u, v) and (u, v') be two any pairs of $\overline{f}$.

2. There are other First Isomorphism Theorems, one for rings and another one for modules.

Then $u = p(x)$ and $v = f(x) = v'$ for some $x \in u$, and so we conclude $v = v'$.
This function satisfies $\overline{f}p = f$, cause of its own definition.
Now, the uniqueness part comes. Assume you have a function $g : X/\sim \to Y$ such that $gp = f$: then for every $u \in X/\sim$ we have some $x \in u$ and

$$g(u) = g(p(x)) = f(x) = \overline{f}(p(x)) = \overline{f}(u),$$

that is $g = \overline{f}$. The most of the work is done now, whereas points (1) and (2) are immediate. □

Corollary 0.2.3. For X and Y sets, let $\sim_X$ and $\sim_Y$ be two equivalence relations on X and Y respectively and let $f : X \to Y$ be a function such that for every $a, b \in X$ such that $a \sim_X b$ we have $f(a) \sim_Y f(b)$. Then there exists one and only one function $\overline{f} : X/\sim_X \to Y/\sim_Y$ such that

$$\begin{tikzcd}
X \arrow[r, "f"] \arrow[d, "p_X"'] & Y \arrow[d, "p_Y"] \\
X/\sim_X \arrow[r, "\overline{f}"'] & Y/\sim_Y
\end{tikzcd}$$

commutes, where p_X and p_Y are the canonical projections. Moreover:

1. $\overline{f}$ is surjective if and only if so is f;
2. if also $a \sim_X b$ for every $a, b \in X$ such that $f(a) \sim_Y f(b)$, then $\overline{f}$ is injective.

Proof. Take the sets X and $Y/\sim_Y$ with the function $p_Y f : X \to Y/\sim_Y$ and use Proposition 0.2.1. □

[Functoriality here...]

Sometimes in Mathematics, generated equivalence relations are involved. Speaking in plain Set Theory terms: if you are given a set X and some $S \subseteq X \times X$, we define the equivalence relation *generated* by S as the smallest among the equivalence relations of X containing S. One can easily verify that such equivalence relation is the intersection of all the equivalence relations on X containing S. We write X/S or $\frac{X}{S}$ to mean the set X quotiented by the equivalence relation generated by R. One can use expressions like:

> On X consider the equivalence relation $\sim$ generated by (the family of statements)
> $$a_\lambda \sim b_\lambda \quad \text{for } \lambda \in I.$$
> (Of course, the a_λ-s the b_λ-s are elements of X.)

to say that:

> On X consider the equivalence relation $\sim$ generated by the set
> $$\{(a_\lambda, b_\lambda) \mid \lambda \in I\}.$$

That being said, it would be clear what we mean by writing

$$\frac{X}{a_\lambda \sim b_\lambda \text{ for } \lambda \in I}.$$

We will discuss about these constructions again in the chapter of limits and colimits.

Proposition 0.2.4. Let X and Y be two sets, $\sim$ an equivalence relation on X generated by $S \subseteq X \times X$ and $f : X \to Y$ any function. Then the following statements are equivalent:

1. for every $a, b \in X$, if $a \sim b$ then $f(a) = f(b)$
2. for every $a, b \in X$, if $(a, b) \in S$ then $f(a) = f(b)$.

Proof. The implication $(1) \Rightarrow (2)$ is trivially true. Consider on X the equivalence relation $\simeq$ define as: for all $a, b \in X$, $a \simeq b$ if and only if $f(a) = f(b)$. Here, $S \subseteq \simeq$, thus $\sim \subseteq \simeq$. □

The proposition just proved permits to rewrite Proposition 0.2.1 for generated equivalence relations.

Corollary 0.2.5. Consider two sets X and Y, a function $f : X \to Y$ and an equivalence relation over X generated by $S \subseteq X \times X$. If for every $(a, b) \in S$ we have $f(a) = f(b)$, then there exists one and only one function $\overline{f} : X/S \to Y$ such that

$$\begin{tikzcd} X \arrow[rr, "f"] \arrow[dr, "p"'] & & Y \\ & X/S \arrow[ur, "\overline{f}"'] & \end{tikzcd}$$

commutes. Here, $p : X \to X/S$ is the canonical projection.

1

Basic vocabulary

1.1 Categories

It is quite easy to make examples motivating the definition of categories and the evolution that follows through these pages.

Example 1.1.1 (Set Theory)**.** Here, we have *sets* and *functions*. Whereas the concepts of set ad membership are primitive, functions are formalised as follows: for A and B sets, a function from A to B is any $f \subseteq A \times B$ such that for every $x \in A$ there exists one and only one $y \in B$ such that $(x, y) \in f$. We write

$$f : A \to B \text{ or } A \xrightarrow{f} B$$

to say 'f is a function from A to B' and, for $x \in A$, we write $f(x)$ the element of B bound to x by f. Consecutive functions can be combined in a quite natural way: for A, B and C sets and functions

$$A \xrightarrow{f} B \xrightarrow{g} C,$$

the *composite* of g and f is the function

$$g \circ f : A \to C\,,\ g \circ f(x) := g\big(f(x)\big).$$

Informally speaking: f takes one input and gives one output; it is then passed to g, which then provides one result. Such operation is called *composition* and has some nice basic properties

1. Every set A has associated an *identity*

   $$\mathtt{1}_A : A \to A\,,\ \mathtt{1}_A(x) := x$$

   is such that for every set B and function $g : B \to A$ we have

   $$\mathtt{1}_A \circ g = g$$

   and for every set C and function $h : A \to C$ we have

   $$h \circ \mathtt{1}_A = h.$$

2. $\circ$ is *associative*, that is for A, B, C and D sets and

   $$A \xrightarrow{f} B \xrightarrow{g} C \xrightarrow{h} D$$

   functions, we have the identity

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

Example 1.1.2 (Topology)**.** A *topological space* is a set where some of its subsets have the status of 'open' sets. Being sets at their core, we have functions between topological spaces, but some of them are more interesting than others. Namely, *continuous functions* are functions that care about the label of open: for if X and Y are topological spaces, a function $f : X \to Y$ is said *continuous* whenever for every open set U of Y the set $f^{-1}U$ is an open set of X. Being function, consecutive continuous functions can be composed: is the resulting function continuous as well? Yes: if X, Y and Z are topological spaces and f and g continuous, for if U is open, then so is $(g \circ f)^{-1}U$.[1] We can state the following basic properties for the composition of continuous functions:

1. Every topological space A has associated the continuous function
$$1_A : A \to A,\ 1_A(x) := x$$
is such that for every topological space B and continuous function $g : B \to A$ we have
$$1_A \circ g = g$$
and for every topological space C and continuous function $h : A \to C$ we have
$$h \circ 1_A = h.$$
2. $\circ$ is *associative*, that is for A, B, C and D topological spaces and
$$A \xrightarrow{f} B \xrightarrow{g} C \xrightarrow{h} D$$
continuous functions, we have
$$(h \circ g) \circ f = h \circ (g \circ f).$$

That is: take the properties listed in the previous example and replace 'set' with 'topological space' and 'function' with 'continuous function'.

Exercise 1.1.3. In Measure Theory, we have *σ-algebras*, that is sets where some of its subsets are said to be *measurable*. We can define *measurable functions* too, that is functions that care about the property od being measurable as continuous functions do of the property of being open.[2] Of course, you can found other example of categories in Algebra, Linear Algebra, Geometry and Analysis. Go and catch as many as you can within you mathematical knowledge. And yes, it may be boring sometimes, and you are right, but as we progress there are remarkable differences from one category to another one.

It should be clear a this point what the pattern is:

Definition 1.1.4 (Categories)**.** A *category* amounts at assigning some things called *objects* and, for each couple of objects a and b, other things named *morphisms* from a to b. We write $f : a \to b$ to say that f is a morphism from a to b, where a is the *domain* of f and b the *codomain*. Besides, for a, b and c

1. It is a fact of Set Theory that for X, Y and Z sets and $f : X \to Y$ and $g : Y \to Z$ functions, we have $(g \circ f)^{-1}U = f^{-1}\left(g^{-1}U\right)$ for every $U \subseteq Z$.
2. Perhaps, you are taught that measurable functions are functions $f : \Omega \to \mathbb{R}$ from a measurable space Ω such that $f^{-1}(-\infty, a]$ is a measurable for every $a \in \mathbb{R}$. Anyway, $\mathbb{R}$ has the Borel σ-algebra, which is defined as the smallest of the σ-algebras containing the open subsets of $\mathbb{R}$ under the Euclidean topology. It can be easily shown that $f : \Omega \to \mathbb{R}$ is measurable if and only if $f^{-1}B$ is measurable for every Borel subset B of $\mathbb{R}$.

objects and $f : a \to b$ and $g : b \to c$ morphisms, there is associated the *composite morphism*

$$gf : a \to c.$$

All those things are regulated by the following axioms:

1. for every object x there is a morphism, 1_x, from x to x such that for every object y and morphism $g : y \to x$ we have

$$1_x g = g$$

   and for every object z and morphism $h : x \to z$ we have

$$h 1_x = h;$$

2. for a, b, c and d objects and morphisms

$$a \xrightarrow{f} b \xrightarrow{g} c \xrightarrow{h} d$$

   we have the identity

$$(hg)f = h(gf).$$

Sometimes, instead of 'morphism' you may find written 'map' or 'arrow'. The former is quite used outside Category Theory, whereas the latter refers to the fact that the symbol $\to$ is employed.

We have started with sets and functions, afterwards we have made an example based on the previous one; if you have accepted the invite of the exercise above, you have likely found categories where objects are sets at their core and morphisms are functions with extra property. We have given an abstract definition of categories because out there are many other categories that deserve attention.

Example 1.1.5 (Monoids are categories)**.** Consider a category $\mathcal{G}$ with a single object, that we indicate with a bare $\bullet$. All of its morphisms have $\bullet$ as domain and codomain: then any two morphisms are composable and the composite of two morphisms $\bullet \to \bullet$ is a morphism $\bullet \to \bullet$. This motivates us to proceed as follows: let G be the collection of the morphisms of $\mathcal{G}$ and consider the operation of composing morphism

$$G \times G \to G, \ (x, y) \to xy.$$

Being $\mathcal{G}$ a category implies this function is associative and $\mathcal{G}$ has the identity of $\bullet$, that is G has one element we call 1 and such that $f1 = 1f = f$ for every $f \in G$. In other words, we are saying G is a monoid. We say the single object category $\mathcal{G}$ *is* a monoid.
Conversely, take a monoid G and a symbol $\bullet$: make such thing acquire the status of object and the elements of G that of morphisms; in that case, the operation of G has the right to be called composition because the axioms of monoid say so. Here, $\bullet$ is something we care of just because by definition morphisms require objects and it has no role other than this.

In Mathematics, a lot of things are monoids, so this is nice. In particular, a *group* is a single object category where for every morphism f there is a morphism g such that gf and fg are the identity of the the unique object. We will deal with isomorphism later in this chapter.

Example 1.1.6 (Preordered sets are categories)**.** A *preordered set* (sometimes contracted as *proset*) consists of a set A and a relation $\leq$ on A such that:

1. $x \leq x$ for every $x \in A$;
2. for every $x, y, z \in A$ we have that if $x \leq y$ and $y \leq z$ then $x \leq z$.

Now we do this: for $x, y \in A$, whenever $x \leq y$ take $(a, b) \in A \times A$. We operate with these couples as follows:

$$(y, z)(x, y) := (x, z), \tag{1.1.1}$$

where $x, y, z \in A$. This definition is perfectly motivated by (2): in fact, if $x \leq y$ and $y \leq z$ then $x \leq z$, and so there is (x, z). By (1), for every $x \in A$ we have the couple (x, x), which has the following property: for every $y \in A$

$$\begin{aligned} (x, y)(x, x) &= (x, y) \quad \text{for every } y \in A \\ (x, x)(z, x) &= (z, x) \quad \text{for every } z \in A. \end{aligned} \tag{1.1.2}$$

Another remarkable feature is that for every $x_1, x_2, x_3, x_4 \in A$

$$\big((x_3, x_4)(x_2, x_3)\big)(x_1, x_2) = (x_3, x_4)\big((x_2, x_3)(x_1, x_2)\big) \tag{1.1.3}$$

We have a category indeed: its objects are the elements of A, the morphisms are the couples (x, y) such that $x \leq y$ and (1.1.1) gives the notion of composition; (1.1.2) says what are identities while (1.1.3) tells the compositions are associative.

Several things are prosets, so this is nice. Namely, *partially ordered sets*, or *posets*, are prosets where every time there are morphisms going opposite directions

$$a \rightleftarrows b$$

then $a = b$. Later in this chapter, we will meet *skeletal* categories.

Example 1.1.7 (Matrices)**.** We need to clarify some terms and notations before. Fixed some field k, for m and n positive integers, a *matrix* of type $m \times n$ is a table of elements of k arranged in m rows and n columns:

$$\begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{pmatrix}$$

If A is the name of a matrix, then $A_{i,j}$ is the element on the intersection of the ith row and the jth column. Matrices can be multiplied: if A and B are matrices of type $m \times n$ and $n \times r$ respectively, then AB is the matrix of type $m \times r$ where

$$(AB)_{i,j} := \sum_{p=1}^{n} A_{i,p} B_{p,j}.$$

Our experiment is this: consider the positive integers in the role of objects and, for m and n integers, the matrices of type $m \times n$ as morphisms from n to m; now, take AB as the composition of A and B. Let us investigate whether categorial axioms hold.

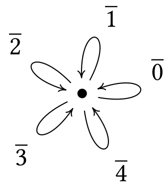


Figure 1.1. The group $\mathbb{Z}_5$ in a diagrammatic vest.

- For n positive integer, we have the *identity matrix* I_n, the one of type $n \times n$ defined by

$$(I_n)_{i,j} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

  One, in fact, can verify that such matrix is an 'identity' in categorial sense: for every positive integer m, an object, and for every matrix A of type $m \times n$, a morphism from n to m, we have

$$AI_n = A,$$

  that is composing A with I_n returns A; similarly, for every positive integer r and for every matrix B of type $r \times n$ we have

$$I_nB = B.$$

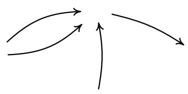
- For A, B and C matrices of type $m \times n$, $n \times r$ and $r \times s$ respectively, we have

$$(AB)C = A(BC).$$

  Again, this identity can be regarded under a categorial light.

The category of matrices over a field k just depicted is written $\mathbf{Mat}_k$. This example may seem quite useless, but it really does matter when you know there is the category of finite vector spaces $\mathbf{FDVect}_k$: just wait until we talk about equivalence of categories. [We may leave something to think about in the meantime, right?]

A diagram is a drawing made of 'nodes', that is empty slots, and 'arrows', that part from some nodes and head to other ones. Here is an example:



(1.1.4)

Nodes are the places where to put objects' names and arrows are to be labelled with morphisms' names. The next step is putting labels indeed, something like this:

$$\begin{tikzcd} & b \arrow[dr, "q"] & \\ a \arrow[ur, bend left, "f"] \arrow[ur, "g"'] & & d \\ & c \arrow[uu, "h"'] & \end{tikzcd} \tag{1.1.5}$$

The idea we want to capture is: having a scheme of nodes and arrows, as in (1.1.4), and then assigning labels, as in (1.1.5). Since diagrams serve to graphically show some categorial structure, there should exist the possibility to 'compose' arrows: two consecutive arrows

$$\begin{tikzcd} {} \arrow[r, bend left] & {} \arrow[r, bend right] & {} \end{tikzcd} \tag{1.1.6}$$

naturally yields that one that goes from the first node and heads to the last one; if in (1.1.6) we label the arrows with f and g, respectively, then the composite arrow is to be labelled with the composite morphism gf. That operation shall be associative and there should exist identity arrows too, that is arrows that represent and behave exactly as identity morphisms. In other words, our drawings shall care of the categorial structure.

If we want to formalize the idea just outlined, the definition of diagram sounds something like this:

Definition 1.1.8 (Diagrams)**.** A *diagram* in a category $\mathcal{C}$ is having:

- a scheme of nodes and arrows, that is a category $\mathcal{I}$;
- labels for nodes, that is for every object i of $\mathcal{I}$ one object x_i of $\mathcal{C}$;
- labels for arrows, that is for every pair of objects i and j of $\mathcal{I}$ and morphism $\alpha : i \to j$ of $\mathcal{I}$, one morphism $f_\alpha : x_i \to x_j$ of $\mathcal{C}$

with all this complying the following rules:

1. $f_{1_i} = 1_{x_i}$ for every i object of $\mathcal{I}$;
2. $f_\beta f_\alpha = f_{\beta\alpha}$, for α and β two consecutive morphisms of $\mathcal{I}$.

Rather than thinking diagrams abstractly — like in the form stated in the definition —, one usually draws them. In general, it is not a good idea to draw all the compositions. For example, consider four nodes and three arcs displayed as

$$\begin{tikzcd} a \arrow[dr, "f"] & & c \arrow[dr, "h"] & \\ & b \arrow[ur, "g"] & & d \end{tikzcd}$$

and draw all the compositions: you will convince yourself it may be a huge mess even for small diagrams. In fact, why waste an arrow to represent the composite gf in

$$\begin{tikzcd} a \arrow[dr, "f"] & & c \\ & b \arrow[ur, "g"'] & \end{tikzcd}$$

when gf is walking along f before and g then? Neither identities need to be drawn: we know every object has one and only one identity and thus the presence of an object automatically carries the presence of its identity.

[A finer formalisation of commutativity?] Consecutive arrows form a 'path'; in that case, we refer to the domain of its first arrow as the domain of the path and to the codomain of the last one as the codomain of the path. Two paths are said *parallel* when they share both domain and codomain. A diagram is said to be *commutative* whenever any pair of parallel paths yields the same composite morphism.

Let us express the categorial axioms in a diagrammatic vest. Let $\mathcal{C}$ be a category and x an object of $\mathcal{C}$. The fact that 1_x the identity of x can be translated as follows: the diagrams

$$\begin{tikzcd} & x \arrow[dd, "1_x"] \\ a \arrow[ur, "f"] \arrow[dr, "f"'] & \\ & x \end{tikzcd} \qquad \begin{tikzcd} x \arrow[dd, "1_x"'] \arrow[dr, "g"] & \\ & b \\ x \arrow[ur, "g"'] & \end{tikzcd} \tag{1.1.7}$$

commute for every a and b objects and f and g morphisms of $\mathcal{C}$. Associativity can be rephrased by saying:

$$\begin{tikzcd}
a \arrow[r, "f"] \arrow[dr, "gf"'] & b \arrow[dr, "hg"] & \\
& c \arrow[r, "h"'] & d
\end{tikzcd}$$

commutes for every a, b, c and d objects and f, g and h morphisms in $\mathcal{C}$.

Example 1.1.9 (Semigroup axioms)**.** A *semigroup* is a set X together with a function $\mu : X \times X \to X$ which is associative, that is

$$\mu(\mu(a,b),c) = \mu(a,\mu(b,c)) \text{ for every } a,b,c \in X.$$

The aim of this example is to see how can we put in diagrams all this. We have a triple of elements, to start with, $(a,b,c) \in X \times X \times X$. On the left side of the equality above, a and b are multiplied first, and the result is multiplied with c:

$$\begin{tikzcd}
X \times X \times X \arrow[r] & X \times X \arrow[r] & X \\
(a,b,c) \arrow[r, mapsto] & (\mu(a,b),c) \arrow[r, mapsto] & \mu(\mu(a,b),c)
\end{tikzcd}$$

It is best we some effort in naming these functions. While it is clear that $X \times X \to X$ is our μ, how do we write $X \times X \times X \to X$? There is notation for it: $\mu \times 1_X$.[3] Instead, on the other side, b and c are multiplied first, and then a is multiplied to their product:

$$\begin{tikzcd}
X \times X \times X \arrow[r] & X \times X \arrow[r] & X \\
(a,b,c) \arrow[r, mapsto] & (a,\mu(b,c)) \arrow[r, mapsto] & \mu(a,\mu(b,c))
\end{tikzcd}$$

The first function is $1_X \times \mu$ and the second one is μ. Thus the equality of the definition of semigroup is equivalent to the fact that the diagram

$$\begin{tikzcd}
X \times X \times X \arrow[r, "\mu \times 1_X"] \arrow[d, "1_X \times \mu"'] & X \times X \arrow[d, "\mu"] \\
X \times X \arrow[r, "\mu"'] & X
\end{tikzcd}$$

commutes.

Exercise 1.1.10. Recall that a monoid is a semigroup (X,μ) with $e \in X$ such that $\mu(x,e) = \mu(e,x) = x$ for every $x \in X$. We usually write a monoid as a triple (X,μ,e). A *group* is a monoid (G,μ,e) such that for every $x \in G$ there exists $y \in G$ such that $\mu(x,y) = \mu(y,x) = e$. Rewrite these structures using commutative diagrams. (The work about associativity is already done, so you should focus how to express the property of the identity in a monoid and the property of inversion for groups.) Also recall that a *monoid homomorphism*, then, from a monoid (X,μ,e_X) to a monoid (Y,λ,e_Y) is any function $f : X \to Y$ such that $f(\mu(a,b)) = \lambda(f(a),f(b))$ for every $a,b \in X$ and $f(e_X) = e_Y$. Use commutative diagrams. Observe that group homomorphisms are defined by requiring to preserve multiplication, whereas the the preservation of identities can be deduced.

[There is some remark on this.]

3. In general, if you have two functions $f : A_1 \to A_2$ and $g : B_1 \to B_2$, the function $f \times g : A_1 \times B_1 \to A_2 \times B_2$ is the one defined by $f \times g(a,b) = (f(a),g(b))$.

1.2 Foundations

Let us return at the beginning, namely the definition of category. Why not formulate it in terms of sets? That is, why don't muster the objects into a set, for any pair of objects, the morphisms into a set and writing compositions as functions?

Let us analyse what happens if we do that. A basic and quite popular fact that fatally crushes our hopes is:

there is no set of all sets.[4]

The first aftermath is that the existence of **Set** would not be legal, because otherwise a set would gather all sets.

Another example comes from both Algebra and Set Theory. In general, it's not a so profound result, but it is interesting for our discourse:

every pointed set $(X, 1)$ has an operation that makes it a group.[5]

Viz there exists no set of all groups, and then neither **Grp** would be supported.

As if the previous examples were not enough, Topology provides another irreducible case. Any set has the corresponding powerset, thus any set gives rise to at least one topological space. Our efforts are doomed, again: there is no set of all topological spaces, and so also **Top** would not be allowed!

It seems that using Set Theory requires the sacrifice of nice categories; and we do not want that, of course. From the few examples above one could surmise it is a matter of *size*: sets sometimes are not appropriate for collecting all the stuff that makes a category. Luckily, there is not a unique Set Theory and, above all, there is one that could help us.

The *von Neumann-Bernays-Gödel approach*, usually shortened as NBG, was born to solve size problems, and may be a good ground for our purposes. In NBG we have *classes*, the most general concept of 'collection'. But not all classes are at the same level: some, the *proper classes*, cannot be element of any class, whilst the others are the *sets*. Here is how the definition of category would look like.

Definition 1.2.1 (Categories)**.** A category $\mathcal{C}$ consists of:

- a class of objects, denoted $|\mathcal{C}|$;
- for every $a, b \in |\mathcal{C}|$, a class of morphisms from a to b, written as $\mathcal{C}(a, b)$;
- for every $a, b, c \in |\mathcal{C}|$, a composition, viz a function

$$\mathcal{C}(b,c) \times \mathcal{C}(a,b) \to \mathcal{C}(a,c)\,,\ (g,f) \to gf$$

with the following axioms:

1. for every $x \in |\mathcal{C}|$ there exists a $1_x \in \mathcal{C}(x,x)$ such that for every $y \in |\mathcal{C}|$ and $g \in \mathcal{C}(y,x)$ we have
$$1_x g = g$$
and for every $z \in |\mathcal{C}|$ and $h \in \mathcal{C}(x,z)$ we have
$$h 1_x = h;$$

4. If we want a set X to be the set of all sets, then it has all its subsets as elements, which is an absurd. In fact, Cantor's Theorem states that for every set X there is no surjective function $f : X \to 2^X$.

5. Actually, this fact is equivalent to the Axiom of Choice.

2. for $a, b, c, d \in |\mathcal{C}|$ and $f \in \mathcal{C}(a,b)$, $g \in \mathcal{C}(b,c)$ and $h \in \mathcal{C}(c,d)$ we have the identity
$$(hg)f = h(gf).$$

How does this double ontology of NBG actually apply at our discourse? For example, in NBG the class of all sets is a legit object: it is a proper class, because it cannot be an actual set. Thus, **Set** exists on NBG, and so exists **Grp**, **Top** and other big categories. Which is nice.

Hence, it is sensible to introduce some terms that distinguish categories by the size of their class of objects. [...]

[What can go wrong if $\mathcal{C}(a,b)$ are proper classes?]

1.3 Isomorphisms

[This sections requires a heavy rewriting.]

Let us step back to the origins. The categorial axioms state identities that deals with morphisms, since equality between morphisms is involved. For that reason, we shall regard these axioms as ones about morphisms, since objects barely appear as start/end point of morphisms.

Thus categories have a notion of sameness between morphisms, the equality, but nothing is said about objects. Of course, there is equality for objects too, but we can craft a better notion of sameness of objects. Not because equality is bad, but we shall look for something that can be stated solely in categorial terms. As usual, simple examples help us to isolate the right notion.

Cantor, the father of Set Theory, conducted its enquiry on cardinalities and not on equality of sets.

Example 1.3.1 (Isomorphisms of sets)**.** For A and B sets, there exists a bijective function $A \to B$ if and only if there exist two functions

$$\begin{tikzcd} A \arrow[r, shift left, "f"] & B \arrow[l, shift left, "g"] \end{tikzcd}$$

such that $gf = 1_A$ and $fg = 1_B$. In Set Theory, the adjective 'bijective' is defined by referring of the fact that sets are things that have elements:

for every $y \in B$ there is one and only one $x \in A$ such that $f(x) = y$.

In contrast,

there exist two functions $\begin{tikzcd} A \arrow[r, shift left, "f"] & B \arrow[l, shift left, "g"] \end{tikzcd}$ such that $gf = 1_A$ and $fg = 1_B$

is written in terms of functions and compositions of functions, that is it is written in a categorial language.

Example 1.3.2 (Isomorphisms in **Grp**)**.** [Yet to be TEXed...]

Example 1.3.3 (Isomorphisms in **Top**)**.** In Topology, things work a little differently. There are bijective continuous functions that that are not homeomorphisms. For instance,

$$f : [0, 2\pi) \to \mathbb{S}^1, \; f(x) := (\cos x, \sin x)$$

is continuous and bijective, but fails to be a homeomorphism because $\mathbb{S}^1$ is compact while $[0, 1)$ is not. 'Fortunately', in Topology there are two basic facts:

- bijective continuous functions that are also closed are homeomorphisms
- continuous functions from compact spaces to Hausdorff spaces are closed

As a consequence, **Top** has a subcategory in which bijections are homeomorphisms: the sub*category of compact Hausdorff spaces* **CHaus**.

Fine, there is some idea that we can formulate in categorial language.

Definition 1.3.4 (Isomorphic objects)**.** In a category $\mathcal{C}$, let a and b two objects and $f : a \to b$ a morphism. A morphism $g : b \to a$ of the same category is said *inverse* of f whenever $gf = 1_a$ and $fg = 1_b$. In that case

- $f : a \to b$ of $\mathcal{C}$ is an *isomorphism* when it has an inverse.
- a is said *isomorphic* to b when there is an isomorphism $a \to b$ in $\mathcal{C}$, and write $a \cong b$.

Lemma 1.3.5. Every morphism has at most one inverse.

That is it may not exist, but if it does it is unique. We write the inverse of f as f^{-1}.

Proof. Fixed a certain category $\mathcal{C}$ and given a morphism $f : a \to b$ with inverses $g_1, g_2 : b \to a$, we have $g_1 = g_1 1_b = g_1(fg_2) = (g_1 f)g_2 = 1_a g_2 = g_2$. □

Example 1.3.6 (Isomorphisms in $\mathbf{Mat}_k$)**.** In this category, morphisms are matrices with entries in some field k and isomorphisms are exactly invertible matrices. Recall that a square matrix A is said invertible whenever the is some matrix B of the same order such that $AB = BA = I$. From Linear Algebra, we know that a matrix A is invertible if and only if (for example) $\det A \neq 0$. One thing a careful reader may ask is: why restrict to only square matrices? We can easily prove that

if a matrix of type $m \times n$ has an inverse, then $m = n$.

This means for $\mathbf{Mat}_k$ that two different objects cannot be isomorphic, or equivalently isomorphic objects are equal.

Categories like this one have a dedicated name.

Definition 1.3.7 (Skeletal categories)**.** A category is said *skeletal* whenever its isomorphic objects are equal.

Exercise 1.3.8. Write **FinSet** for the category of finite sets and functions between sets. Find one skeleton.

Exercise 1.3.9. Find one skeleton of $\mathbf{FDVect}_k$.

1.4 Mono- and Epimorphisms

[This section has to be rewritten.]

Definition 1.4.1 (Monomorphisms and epimorphisms)**.** A morphism $f : a \to b$ of a category $\mathcal{C}$ is said to be:

- a *monomorphism* whenever if

$$\begin{tikzcd} c \arrow[r, bend left, "g_1"] \arrow[r, bend right, "g_2"'] & a \arrow[r, "f"] & b \end{tikzcd}$$

  commutes for every object c and morphisms $g_1, g_2 : c \to a$ of $\mathcal{C}$, then $g_1 = g_2$;

- an *epimorphism* whenever if

$$\begin{tikzcd} a \arrow[r, "f"] & b \arrow[r, bend left, "h_1"] \arrow[r, bend right, "h_2"'] & c \end{tikzcd}$$

  commutes for every object d and morphisms $h_1, h_2 : c \to a$ of $\mathcal{C}$, then $h_1 = h_2$;

[…]

Another way to express the things of the previous definition is this: $f : a \to b$ is a monomorphism whenever for every $c \in |\mathcal{C}|$ the function

$$\mathcal{C}(c,a) \to \mathcal{C}(c,b)\,,\ g \to fg \tag{1.4.8}$$

is injective. Similarly, $f : a \to b$ is an epimorphism when for every $d \in |\mathcal{C}|$ the function

$$\mathcal{C}(a,d) \to \mathcal{C}(b,d)\,,\ h \to hf \tag{1.4.9}$$

is injective. Category theorists call the functions (1.4.8) *precompositions* with f and (1.4.9) *postcompositions* with f.

1.5 Basic constructions

In this section, we will present the first and most basic constructions involving categories.

For $\mathcal{C}$ a category, its *dual* (or *opposite*) category is denoted $\mathcal{C}^{\mathrm{op}}$ and is described as follows. Here, the objects are the same of $\mathcal{C}$ and 'being a morphism $a \to b$' exactly means 'being a morphism $b \to a$ in $\mathcal{C}$'. In other words, passing from a category to its dual leaves the objects unchanged, whereas the morphisms have their verses reversed. To dispel any ambiguity, by 'reversing' the morphisms we mean that morphisms $f : a \to b$ of $\mathcal{C}$ can be found among the morphisms $b \to a$ of $\mathcal{C}^{\mathrm{op}}$ and, vice versa, morphisms $a \to b$ of $\mathcal{C}^{\mathrm{op}}$ among the morphisms $b \to a$ of $\mathcal{C}$. Nothing is actually constructed out of the blue. Some authors suggest to write f^{op} to indicate that one f once it has domain and codomain interchanged, but we do not do that here, because they really are the same thing but in different places. So, if f is the name of a morphism of $\mathcal{C}$, the name f is kept to indicate that morphism as a morphism of $\mathcal{C}^{\mathrm{op}}$; obviously, the same convention applies in the opposite direction. It may seem we are going to nowhere, but it makes sense when it comes to define the compositions in $\mathcal{C}^{\mathrm{op}}$: for $f : a \to b$ and $g : b \to c$ morphsisms of $\mathcal{C}^{\mathrm{op}}$ the composite arrow is so defined

$$gf := fg.$$

This is not a commutative property, though. Such definition is to be read as follows. At the left side, f and g are to be intended as morphisms of $\mathcal{C}^{\mathrm{op}}$ that are to be composed therein. Then the composite gf is calculated as follows:

1. look at f and g as morphisms of $\mathcal{C}$ and compose them as such: so $f : b \to a$ and $g : c \to b$ and $fg : c \to a$ according to $\mathcal{C}$;
2. now regard fg as a morphism of $\mathcal{C}^{\mathrm{op}}$: this is the value gf is bound to.

Let us see now whether the categorial axioms are respected. For x object of $\mathcal{C}^{\mathrm{op}}$ there is 1_x, which is a morphism $x \to x$ in either of $\mathcal{C}$ and $\mathcal{C}^{\mathrm{op}}$. For every object y and morphism $f : y \to x$ of $\mathcal{C}^{\mathrm{op}}$ we have

$$1_x f = f 1_x = f.$$

Similarly, we have that

$$g1_x = g$$

for every object z and morphism $g : x \to z$ of $\mathcal{C}^{\text{op}}$. Hence, 1_x is an identity morphism in $\mathcal{C}^{\text{op}}$ too. Consider now four objects and morphisms of $\mathcal{C}^{\text{op}}$

$$a \xrightarrow{f} b \xrightarrow{g} c \xrightarrow{h} d$$

and let us parse the composition

$$h(gf).$$

In $h(gf)$ regard both h and gf as morphisms of $\mathcal{C}$. In that case, $h(gf)$ is exactly $(gf)h$, where gf is fg once f and g are taken as morphisms of $\mathcal{C}$ and composed there. So $h(gf) = (fg)h$, where at left hand side compositions are performed in $\mathcal{C}$: being the composition is associative, $h(gf) = (fg)h = f(gh)$. We go back to $\mathcal{C}^{\text{op}}$, namely $f(gh)$ becomes $(gh)f$ and gh becomes hg, so that we eventually get the associativity

$$h(gf) = (hg)f.$$

It may seem hard to believe, but duality is one of the biggest conquest of Category Theory. [Talk more about duality here...] This construction may seem a useless sophistication for now, but later we will discover how this serves the scope to make functors encompass a broader class of constructions. However, as for now, let us see all this under a new light: what does duality mean for diagrammatic reasoning? Commuting triangles

$$\begin{tikzcd} & b \arrow[dd, "g"] \\ a \arrow[ur, "f"] \arrow[dr, "h"'] & \\ & c \end{tikzcd}$$

of $\mathcal{C}$ are exactly commuting triangles

$$\begin{tikzcd} & b \arrow[dl, "f"'] \\ a & \\ & c \arrow[uu, "g"'] \arrow[ul, "h"] \end{tikzcd}$$

in $\mathcal{C}^{\text{op}}$.

Exercise 1.5.1. In this way, it should be even more immediate to prove the two categorial axioms. Give it a try. Observe this approach is the mere translation of what we have conveyed with words above.

Example 1.5.2 (Dual prosets)**.** We already know how prosets are categories; let $(P, \leqslant)$ be one of them. Here the morphisms are exactly the pairs (b, a) such that $(a, b) \in \leqslant$. If we rephrase all this, we can introduce the dual relation $\geqslant$ defined by: $b \geqslant a$ if and only if $a \geqslant b$.

Exercise 1.5.3. Consider a single object category $\mathcal{G}$ — what is called moonid —. What is $\mathcal{G}^{\text{op}}$? What is $\mathcal{G}^{\text{op}}$ if $\mathcal{G}$ is a group?

The concept of duality for categories has one important consequence on statements written in a 'categorial language'. We do not need to be fully precise here: they are statements written in a sensible way using the usual logical connectives, names for objects, names for morphisms and quantifiers acting on such names.

Example 1.5.4. If we have a morphism $f : a \to b$ in some category $\mathcal{C}$, consider the statement

> For every object c of $\mathcal{C}$ and morphisms $g_1, g_2 : c \to a$ in $\mathcal{C}$, if $fg_1 = fg_2$ then $g_1 = g_2$.

If you remember, it is just said that f is a monomorphism. We operate a translation that doesn't modify the truth of the sentence: that is, if it is true, it remains so; it is false, it remains false.

> For every object c of $\mathcal{C}^{\mathrm{op}}$ and morphisms $g_1, g_2 : a \to c$ in $\mathcal{C}^{\mathrm{op}}$, if $g_1 f = g_2 f$ then $g_1 = g_2$.

If we regard f as a morphism $b \to a$ of $\mathcal{C}^{\mathrm{op}}$, then f is an epimorphism in $\mathcal{C}^{\mathrm{op}}$.

Let us try to settle this explicitly: if we have a categorial statement p, the dual of p — we may call p^{op} — is the statement obtained from p keeping the connectives and the quantifiers of p, whereas the other parts are replaced by their dual counterparts.

Example 1.5.5. [Anticipate products and co-products...]

Another useful construction is that of product of categories. Assuming we have two categories $\mathcal{C}_1$ and $\mathcal{C}_2$, the product $\mathcal{C}_1 \times \mathcal{C}_2$ is the category in which

- The objects are the pairs $(a, b) \in |\mathcal{C}_1| \times |\mathcal{C}_2|$.
- Being a morphism $(a_1, a_2) \to (b_1, b_2)$ means being a pair of two morphisms

$$\left(\begin{array}{c} a_1 \\ {\scriptstyle f_1}\downarrow \\ b_1 \end{array} , \begin{array}{c} a_2 \\ \downarrow{\scriptstyle f_2} \\ b_2 \end{array} \right)$$

  where f_1 is in $\mathcal{C}_1$ and f_2 in $\mathcal{C}_2$; we write such morphism as (f_1, f_2).
- The composition is defined component-wise

$$\left(\begin{array}{c} b_1 \\ {\scriptstyle g_1}\downarrow \\ c_1 \end{array} , \begin{array}{c} b_2 \\ \downarrow{\scriptstyle g_2} \\ c_2 \end{array} \right)\left(\begin{array}{c} a_1 \\ {\scriptstyle f_1}\downarrow \\ b_1 \end{array} , \begin{array}{c} a_2 \\ \downarrow{\scriptstyle f_2} \\ b_2 \end{array} \right) := (g_1 f_1, g_2 f_2).$$

Exercise 1.5.6. Verify categorial axioms hold for the product of two categories.

In future, we will need to consider product categories of the form $\mathcal{C}^{\mathrm{op}} \times \mathcal{D}$. For objects, there is nothing weird to say; about morphism, observe that a morphism

$$\begin{array}{c} (a, b) \\ \downarrow{\scriptstyle (f,g)} \\ (a', b') \end{array}$$

is precisely the pair

$$\left(\begin{array}{c} a \\ f\big\uparrow \\ a' \end{array} , \begin{array}{c} b \\ \big\downarrow g \\ b \end{array} \right)$$

whose first component comes from $\mathcal{C}$ while the second one from $\mathcal{D}$.

Let as now talk about comma categories.

Example 1.5.7. Words are labels humans attach to things to refer to them. Different groups of speakers have developed different names for the surrounding world, which resulted in different languages. We can use sets to gather the words present in any language. Now, if we are given a set Ω of things and a set L of the words of a chosen language, then a function $\lambda : \Omega \to L$ can be seen as the act of labelling things with names.[6] We will call such functions as *vocabularies* for Ω, although this might not be the official name.
Languages do not live in isolation with others: if we know how to translate words, we can understand what speakers of other languages are saying. Imagine now you have two vocabularies

$$\begin{tikzcd} A & & B \\ & \Omega \arrow[ul, "\alpha"] \arrow[ur, "\beta"'] & \end{tikzcd}$$

To illustrate concept, think some $\omega \in \Omega$: if $\omega \in \Omega$ has the name $\alpha(\omega)$ under the vocabulary α and ω is called $\beta(\omega)$ according to β, then a translation would be a correspondence between $\alpha(\omega)$ and $\beta(\omega)$. We could say, a *translation* from α to β is a function $\tau : A \to B$ such that

$$\begin{tikzcd} A \arrow[rr, "\tau"] & & B \\ & \Omega \arrow[ul, "\alpha"] \arrow[ur, "\beta"'] & \end{tikzcd}$$

commutes. As you may expect, translations can be composed to obtain translations: if we have two translations η and θ as in the diagram

$$\begin{tikzcd} & B \arrow[dr, "\theta"] & \\ A \arrow[ur, "\eta"] & \Omega \arrow[l, "\alpha"] \arrow[u, "\beta"] \arrow[r, "\gamma"'] & C \end{tikzcd}$$

with the two triangles commuting, we have also the commuting

$$\begin{tikzcd} A \arrow[rr, "\theta\eta"] & & C \\ & \Omega \arrow[ul, "\alpha"] \arrow[ur, "\gamma"'] & \end{tikzcd}$$

6. Of course, our discourse is rather simplified here: everything in Ω has one and only one dedicated word, which is not always the case. In fact, the existence of synonyms within a language undermines the requirement of uniqueness. Further, a language might not have words for everything: for instance, German has the word Schilderwald, which has not a corresponding single word in English — if you want to explain the meaning it bears, you can say it is 'a street that is so overcrowded and rammed with so many street signs that you are getting lost rather than finding your way.'

This is interesting if we look at things with categories: objects are functions that have a domain in common, and we have selected as morphisms the functions between the codomains that make certain triangles commute. Other examples in such spirit follow.

Example 1.5.8 (Covering spaces)**.** Probably you have heard of covering spaces when you had to calculate the first homotopy group of $\mathbb{S}^1$. More generally, a *covering space* of a topological space X is continuous function $p : \widetilde{X} \to X$ with the following property: there is a open cover $\{U_i \mid i \in I\}$ of X such that every $p^{-1}U_i$ is the disjoint union of a family $\left\{V^i_j \mid j \in J\right\}$ of open subsets of $\widetilde{X}$ and the restriction of p to V^i_j is a homeomorphism $V^i_j \to U_i$ for every $i \in I$ and $j \in J$. [To be continued...]

Example 1.5.9 (Field extensions)**.** [Yet to be TEX-ed...]

1.6 Functors

Definition 1.6.1 (Functors)**.** A functor F from a category $\mathcal{C}$ to a category $\mathcal{D}$ is having the following functions, all indicated by F:

- one 'function on objects'

$$F : |\mathcal{C}| \to |\mathcal{D}| \,, \; x \to F(x)$$

- for every objects a and b, one 'function on morphisms'

$$F : \mathcal{C}(a,b) \to \mathcal{D}(F(a),F(b)) \,, \; f \to F(f)$$

such that

1. for every object x of $\mathcal{C}$ we have $F(1_x) = 1_{F(x)}$;
2. for every objects x, y, z and morphisms $f : x \to y$ and $g : y \to z$ of $\mathcal{C}$ we have $F(g)F(f) = F(gf)$.

To say that F is a functor from $\mathcal{C}$ to $\mathcal{D}$ we use $F : \mathcal{C} \to \mathcal{D}$, a symbolism that recalls that one of morphism in categories.

A first straightforward consequence of functoriality is contained in the following proposition.

Proposition 1.6.2. Let $F : \mathcal{C} \to \mathcal{D}$ be a functor. If f is an isomorphism of $\mathcal{C}$, then so is $F(f)$.

Proof. **Exercise 1.6.3**. □

As often happens, let us start with simple exmaples: in this context, the simplest ones can be obtained by choosing very simple categories.

Example 1.6.4 (Functors from sets)**.** Classes can be regarded as categories with no morphisms apart identities: in any category, every object carries its own identity, but if these are the only morphisms, they become redundant information. We will restrict our attention to classes that are actual sets.
So, what is a functor $F : \mathcal{S} \to \mathcal{C}$ out of a set $\mathcal{S}$? As functors do by definition, it maps objects to objects and morphisms to morphisms; but the only morphisms of $\mathcal{S}$ are identities, which are taken to identities of $\mathcal{C}$. Since F involves only objects and identities, F is just a families of objects of $\mathcal{C}$.[7] In particular, functors from sets to sets are just functions!

7. You probably are used to write $\{X_\alpha \mid \alpha \in I\}$ to indicate a family of sets. Actually, $\{X_\alpha \mid \alpha \in I\}$ is a function from the set of indexes I to some set the X_i's are picked from.

Example 1.6.5 (Functors from prosets)**.** Consider a functor $F : (A, \leq) \to \mathcal{C}$ out of a proset. We know that $(A, \leq)$ regarded as a category has at most one morphism for each ordered couple in $A \times A$. For that reason, let us adopt this notation: for every $i, j \in A$ such that $i \leq j$ indicate by $F_{i,j}$ the image of the unique morphism $i \to j$ of $(A, \leq)$ via F. That being said, our functor F is just a collection $\{F_i \mid i \in A\}$ with the morphisms $F_{i,j}$'s for $i, j \in A$ with $i \leq j$.
As a particular instance of this, let us examine functors

$$H : (\mathbb{N}, \leq) \to \mathcal{C}$$

with $\leq$ being the usual ordering of $\mathbb{N}$. For $i, j \in \mathbb{N}$ with $i \leq j$, the morphism $i \to j$ can be factored into consecutive morphisms

$$i \to j = (j - 1 \to j) \cdots (i + 1 \to i + 2)(i \to i + 1).$$

For that reason, our H 'is' just a sequence

$$H_0 \xrightarrow{\partial_0} H_1 \xrightarrow{\partial_1} \cdots \xrightarrow{\partial_{n+1}} H_n \xrightarrow{\partial_n} \cdots$$

of objects and morphisms in $\mathcal{C}$, where we have written ∂_j for $H_{j,j+1}$.

Exercise 1.6.6. Yes, diagrams are functors!

Example 1.6.7 (Monotonic functions)**.** We have met before, how a preordered set is a category; recall also the pure set-theoretic definition of this notion. For $(A, \leq_A)$ and $(B, \leq_B)$ preordered sets, a function $f : A \to B$ is said *monotonic* whenever for every $x, y \in A$ we have $f(x) \leq_B f(y)$ provided that $x \leq_A y$. In bare set-theoretic terms, this can be rewritten as follows: for every $x, y \in A$ such that $(x, y) \in \leq_A$, then $(f(x), f(y)) \in \leq_B$, where we make explicit the pairs, that are morphisms of the preordered sets seen as categories.

Example 1.6.8 (Monoid homomorphsisms)**.** We have previously seen that a monoid 'is' a single-object category. Consider now two such categories, say $\mathcal{G}$ and $\mathcal{H}$, and a functor $f : \mathcal{G} \to \mathcal{H}$ is. Denoting by $\bullet_{\mathcal{G}}$ and $\bullet_{\mathcal{H}}$ the object of $\mathcal{G}$ and $\mathcal{H}$ respectively, there is a unique possibility: mapping $\bullet_{\mathcal{G}}$ to $\bullet_{\mathcal{H}}$. The functorial axioms in that case are:

$$f(xy) = f(x)f(y)$$

for every morphisms x and y of $\mathcal{G}$ and

$$f(1_{\mathcal{G}}) = 1_{\mathcal{H}},$$

with $1_{\mathcal{G}}$ and $1_{\mathcal{H}}$ being the identities of $\mathcal{G}$ and $\mathcal{H}$ respectively. These two properties say that f is a monoid homomorphism; in this case there is also an equation that about objects but these two are a mere subtlety that adds nothing. It is easy to do the converse: a monoid homomorphism is a functor.

Let us remain to Algebra. What happens if the domain of a functor is a group and the codomain is **Set**? (Recall that 'group' is just a fancier abbreviation of 'single object groupoid'.) The answer is contained in the following example.

Example 1.6.9 (Group actions)**.** If $\mathcal{G}$ is a group, what is a functor $\delta : \mathcal{G} \to \mathbf{Set}$? Let us write G the set of its morphisms. The single object of $\mathcal{G}$ is mapped to one set X. Since all the morphisms of $\mathcal{G}$ are isomorphisms, then δ takes each of them to bijections from X to X. Thus we can say δ is the assignment of a certain set X and, for every g element of the group, of one isomorphism $\delta_g : X \to X$. Does this sound familiar? What we have described is a group *action* over the set X, that is a group isomorphism from G to the symmetric group of X.

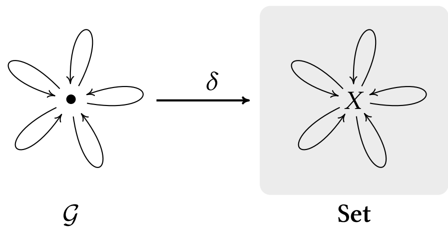


Figure 1.2. A group action as functor.

Example 1.6.10 (The category **Eqv**)**.** A *setoid* [nlab uses this term...] is having a set and an equivalence relation defined on it. If X is a set and $\sim$ an equivalence relation over X, the setoid amounting of these data is written as $(X, \sim)$. Any set X has of course its own equality, that we denote by $=_X$.[8]
For if X and Y sets, a function $f : X \to Y$ respects this rule by definition:

> for every $a, b \in X$, if $a =_X b$ then $f(a) =_Y b$.

We would like to replace the equalities above with equivalence relations: for if $(X, \sim_X)$ and $(Y, \sim_Y)$ are setoids, a *functoid* [ok, let me find/craft a nicer name...] from $(X, \sim_X)$ to $(Y, \sim_Y)$ is any function $f : X \to Y$ such that

> for every $a, b \in X$, if $a \sim_X b$ then $f(a) \sim_Y f(b)$.

Functoids are certain type of functions, and composing two of them as such returns a funtoid. Categorial axioms hold almost for free, so we really have a *category of setoids and functoids*, **Eqv**.
Let us now involve functoriality. There is a nice theorem:

> Let X and Y be two sets with $\sim_X$ and $\sim_Y$ equivalence relations on X and Y respectively. Then for every $f : X \to Y$ such that $f(a) \sim_Y f(b)$ for every $a, b \in X$ such that $a \sim_X b$, there exists one and only one $\phi : X/{\sim_X} \to Y/{\sim_Y}$ that makes
>
> $$\begin{tikzcd} X \arrow[r, "f"] \arrow[d, "\lambda a.[a]_X"'] & Y \arrow[d, "\lambda b.[b]_Y"] \\ X/{\sim_X} \arrow[r, "\phi"'] & Y/{\sim_Y} \end{tikzcd}$$
>
> commute. (The vertical functions are the canonical projections.)

This underpins the functor

$$\pi : \mathbf{Eqv} \to \mathbf{Set}$$

that maps setoids $(X, \sim)$ to sets $X/\sim$ and functoids $f : (X, \sim_X) \to (Y, \sim_Y)$ to functions

$$\pi_f : X/{\sim_X} \to Y/{\sim_Y}$$
$$\pi_f([a]_X) := [f(a)]_Y,$$

whose existence and uniqueness is claimed by the just mentioned Proposition.

Example 1.6.11 (Free groups)**.** Suppose given a *group alphabet* S, which is a set of things we decide to name 'letters'. Then a *group word* with system S is a string obtained by juxtaposition of a finite amount of 'x^1' and 'x^{-1}', where

8. In Set Theory, $=_X$ is the set $\{(a, a) \mid a \in X\}$.

$x \in S$. The *empty word* is obtained by writing no letter, and we shall denote it by something, say e; instead, the other words appear as

$$x_1^{\phi_1} \cdots x_n^{\phi_n},$$

with $x_1, \dots, x_n \in S$ and $\phi_1, \dots, \phi_n \in \{-1, 1\}$. [9] [10]
The length of a word is the number of letters it is made of. We define equality only on words having the same length: we say $x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ is equal to $y_1^{\beta_1} \cdots y_n^{\beta_n}$ whenever $x_i = y_i$ and $\alpha_i = \beta_i$ for every $i \in \{1, \dots, n\}$.
A group word $x_1^{\phi_1} \cdots x_n^{\phi_n}$ is said *irreducible* whenever $x_i^{\phi_i} \neq x_{i+1}^{-\phi_{i+1}}$ for every $i \in \{1, \dots, n-1\}$; the empty word is irreducible by convention. Let us write $\langle S \rangle$ the set of all irreducible words written using the alphabet S. It is natural to join two words by bare juxtaposition, but the resulting word may not be irreducible; this issue has to be fixed:

$$\begin{aligned}
&\cdot : \langle S \rangle \times \langle S \rangle \to \langle S \rangle \\
&e \cdot w := w \,, \; w \cdot e := w \\
&(x_1^{\lambda_1} \cdots x_m^{\lambda_m}) \cdot (y_1^{\mu_1} \cdots y_n^{\mu_n}) := \begin{cases} (x_1^{\lambda_1} \cdots x_{m-1}^{\lambda_{m-1}}) \cdot (y_2^{\mu_2} \cdots y_n^{\mu_n}) & \text{if } x_m^{\lambda_m} = y_1^{-\mu_1} \\ x_1^{\lambda_1} \cdots x_m^{\lambda_m} y_1^{\mu_1} \cdots y_n^{\mu_n} & \text{otherwise.} \end{cases}
\end{aligned}$$

Let us define a function that either reverses the order of the letters and changes each exponent to the other one:

$$i : \langle S \rangle \to \langle S \rangle \,, \; i\left(x_1^{\xi_1} \cdots x_i^{\xi_i} x_{i+1}^{\xi_{i+1}} \cdots x_n^{\xi_n}\right) := x_n^{-\xi_n} \cdots x_{i+1}^{-\xi_{i+1}} x_i^{-\xi_i} \cdots x_1^{-\xi_1}.$$

It is immediate to show that $w \cdot i(w) = i(w) \cdot x = e$ for every $w \in \langle S \rangle$. Only the associativity of $\cdot$ is a a bit tricky to prove. At this point we have endowed $\langle S \rangle$ with a group structure.
Thus from a set S we are able to build a group $\langle S \rangle$, that is called *free group* with base S, or group generated by S. Now, if take two sets S and T and a function $f : S \to T$, we have the group homomorphism

$$\langle f \rangle : \langle S \rangle \to \langle T \rangle \,, \; \langle f \rangle \, (x_1^{\delta_1} \cdots x_n^{\delta_n}) := \big(f(x_1)\big)^{\delta_i} \cdots \big(f(x_n)\big)^{\delta_n}.$$

It is immediate to demonstrate that we ended up with having a functor

$$\langle \;\; \rangle : \mathbf{Set} \to \mathbf{Grp}.$$

In future we will provide other examples involving free modules.

Example 1.6.12 (Free modules)**.** The explicit construction of the free *abelian* group given a set is simpler than that of free group in general. Since an abelian group is a $\mathbb{Z}$-module, let us show how to build a free module.
Let R be a ring and S be a set, as in the previous example. Intuitively, the module generated by S are linear combination of a finite amount of elements of S, that is expressions of the form

$$\sum_{i=1}^{n} \lambda_i x_i$$

9. Something that may irk you is that our words can be redundant, being consecutive repetitions of the same letter allowed. If you want, you can let exponents range over all the integers, but this needs you to modify what comes after.
10. Here, we can choose any pair of distinct symbols instead of -1 and 1. If we do so, we need a function that maps each of them into the other one. In this presentation we employ the function that takes one integer and returns its opposite.

for $n \in \mathbb{N}$, $\lambda_1, \dots, \lambda_n \in R$ and $x_1, \dots, x_n \in S$. Observe, however that the 'sum' here is just a formal expression: there is no link to the an operation of sum yet. We can rethink this linear combination as something more manageable during computations:

$$\sum_{x \in S} \lambda_x x$$

where $\lambda : S \to R$ is non zero for a finite amount of elements of S. Observe that it is just formalism: S may be an infinite set, but the sum $\sum_{x \in S} \lambda_x x$ is not to be understood as a series in Analysis; consider also $\lambda_x \neq 0$ for finitely many x, so if S is infinite, the most of the terms are useless. [Instead of using the device of 'formal expressions', we can define the module words as functions $\lambda : S \to R$ that assume non-zero values for a finite amount of elements. Isn't that the same stuff of a formal sum?]
Thus, let us write the explicit definition:

$$\langle S \rangle := \left\{ \sum_{x \in S} \lambda_x x \,\middle|\, \lambda : S \to R,\ \lambda_x \neq 0 \text{ for finitely many times} \right\}.$$

This is only the first step to make a module with such set. We give a sum

$$\begin{gathered} + : \langle S \rangle \times \langle S \rangle \to \langle S \rangle \\ \left(\sum_{x \in S} \alpha_x x \right) + \left(\sum_{x \in S} \beta_x x \right) := \sum_{x \in S} (\alpha_x + \beta_x) x \end{gathered}$$

and an external product

$$\begin{gathered} \cdot : R \times \langle S \rangle \to \langle S \rangle \\ \eta \cdot \left(\sum_{x \in S} \alpha_x x \right) := \sum_{x \in S} (\eta \alpha_x) x \end{gathered}$$

It is simple to verify that $\langle S \rangle$ is a R-module now.
So far, we only have an process that takes sets and emits R-modules: to make a functor, we also need to instruct how to construct a linear function from a simple function of sets. For $f : S \to T$, we give

$$\begin{gathered} \langle f \rangle : \langle S \rangle \to \langle T \rangle \\ \langle f \rangle \left(\sum_{x \in S} \lambda_x x \right) := \sum_{x \in S} \lambda_x f(x). \end{gathered}$$

It is simple to verify we have a functor

$$\langle \ \ \rangle : \mathbf{Set} \to \mathbf{Mod}_R.$$

Exercise 1.6.13. There is a plenty of 'free stuff' around that can give arise to functors like the one above. Find and illustrate some of them.

Example 1.6.14 (The First Homotopy Group)**.** A *pointed topological space* is a topological space X with one point $x_0 \in X$; we write it as (X, x_0). We define a *pointed continuous function* $(X, x_0) \to (Y, y_0)$ to be a continuous functions $X \to Y$ taking x_0 to y_0. Furthermore, composing such functions yields a pointed continuous function. So, really we have the category of pointed topological spaces, we denote by $\mathbf{Top}_*$.

$$\Omega(X, x_0) := \{\text{continuous } \phi : [0,1] \to X \mid \phi(0) = \phi(1) = x_0\}$$

and call its elements *loops* of X based at x_0. Two loops can be joined, that is traversing one loop after another one: for if $\phi, \psi \in \Omega(X, x_0)$ we introduce the loop $\phi * \psi : [0,1] \to X$ with

$$(\phi * \psi)(t) := \begin{cases} \phi(2t) & \text{if } t \leq \frac{1}{2} \\ \psi(2t-1) & \text{otherwise.} \end{cases}$$

This gives us the operation of *junction* of loops

$$* : \Omega(X, x_0) \times \Omega(X, x_0) \to \Omega(X, x_0).$$

Now it's time to find a suitable equivalence relation that is compatible with this operation. For if $\phi, \psi \in \Omega(X, x_0)$, we say ϕ is *homotopic* to ψ whenever there exists a *homotopy* from ϕ to ψ, viz a continuous function

$$H : [0,1] \times [0,1] \to X$$

such that $H(\ ,0) = \phi$, $H(\ ,1) = \psi$, $H(s,0) = H(s,1) = x_0$ for every $s \in [0,1]$. This relation is an equivalence one and it is compatible with $*$. It remains to verify some properties to define a group structure:

- $(\alpha * \beta) * \gamma$ is homotopic to $\alpha * (\beta * \gamma)$ for every $\alpha, \beta, \gamma \in \Omega(X, x_0)$.
- the paths $\alpha * c_{x_0}$, $c_{x_0} * \alpha$ and α are homotpic for every $\alpha \in \Omega(X, x_0)$; here, c_{x_0} is the loop defined by $c_{x_0}(t) = x_0$.
- $\alpha * \alpha^{-1}$, $\alpha^{-1} * \alpha$ and c_{x_0} are homotopic for every $\alpha \in \Omega(X, x_0)$; here, α^{-1} is the loop with $\alpha^{-1}(t) := \alpha(1-t)$.

We have now all the ingredients to introduce a group: define $\pi_1(X, x_0)$ to be the set obtained identifying homotopic elements of $\Omega(X, x_0)$; this set is a group once you consider the operation

$$\begin{aligned} \pi_1(X, x_0) \times \pi_1(X, x_0) &\to \pi_1(X, x_0) \\ ([\alpha], [\beta]) &\to [\alpha][\beta] := [\alpha * \beta]. \end{aligned}$$

Here, we have written $[\phi]$ for the set of loops homotopic to ϕ. Sometimes — especially if we are considering more topological spaces —, we need to specify the topological space we are taking loops, for example writing $[\phi]_X$. Now, it is the turn to define induced homomorphisms: for a pointed continuous function $f : (X, x_0) \to (Y, y_0)$ we have the group homomorphism

$$\pi_1(f) : \pi_1(X, x_0) \to \pi_1(Y, y_0)\,,\ \pi_1(f)[\phi]_X := [f\phi]_Y.$$

(You may have a look at Example 1.6.10.) Instead of $\pi_1(f)$, you may have been get used to f_*. In conclusion, we have just defined one functor

$$\pi_1 : \mathbf{Top}_* \to \mathbf{Grp}.$$

The *first fundamental group* is not just a group, and that the actual group is just a piece of larger picture.

Traditionally, functors of Definition 1.6.1 above are called 'covariant', because there are *contra*variant functors too. However, there is no sensible reason to maintain these two adjectives; at least, almost everyone agrees to not use the first adjective, whilst the second one still survives.

For if $\mathcal{C}$ and $\mathcal{D}$ are categories, a *contravariant functor* from $\mathcal{C}$ to $\mathcal{D}$ is just a functor $\mathcal{C}^{\mathrm{op}} \to \mathcal{D}$. It is best that we say what functors $F : \mathcal{C}^{\mathrm{op}} \to \mathcal{D}$ do. They map objects to objects and morphisms $f : a \to b$ of $\mathcal{C}^{\mathrm{op}}$ to morphisms $F(f) : F(a) \to F(b)$ of $\mathcal{D}$. But, remembering how dual categories are defined, what F actually does is this:

it maps objects of $\mathcal{C}$ to objects of $\mathcal{D}$, and morphisms $f : b \to a$ of $\mathcal{C}$ to morphisms $F(f) : F(a) \to F(b)$ of $\mathcal{D}$ (mind that a and b have their roles flipped).

Now, what about functoriality axioms? Neither with identities F does something different and the composite gf of $\mathcal{C}^{\mathrm{op}}$ is mapped to the composite $F(g)F(f)$ of $\mathcal{D}$. Again by definition of dual categories, this can be translated as follows:

the composite fg of $\mathcal{C}$ is mapped to $F(g)F(f)$ (notice here how f and g have their places switched).

You can think of contravariant functors as a trick to do what we want.

Example 1.6.15. The set of natural numbers $\mathbb{N}$ has the order relation of divisibility, that we denote $|$: regard this poset as a category. From Group Theory, we know that for every $m, n \in \mathbb{N}$ such that $m \mid n$ there is a homomorphism

$$f_{m,n} : \mathbb{Z}/n\mathbb{Z} \to \mathbb{Z}/m\mathbb{Z}\,, \; f_{m,n}(a + n\mathbb{Z}) := a + m\mathbb{Z}.$$

In fact, $\mathbb{Z}/m\mathbb{Z}$ is the kernel of the homomorphism

$$\pi_m : \mathbb{Z} \to \mathbb{Z}/m\mathbb{Z}\,, \; \pi_m(x) := x + m\mathbb{Z}$$

and, because $m \mid n$, we have $n\mathbb{Z} \subseteq m\mathbb{Z}$. In that case, some Isomorphism Theorem[11] justifies the existence of $f_{m,n}$. This offers us a nice functor:

$$F : (\mathbb{N}, |)^{\mathrm{op}} \to \mathbf{Grp}$$

that maps naturals n to groups $\mathbb{Z}/n\mathbb{Z}$ and $m \mid n$ to the homomorphism $f_{m,n}$ defined above.

[Clearly, this section needs more work...]

Functors can be composed — and I think at this point it is not a secret. Take $\mathcal{C}$, $\mathcal{D}$ and $\mathcal{E}$ categories and functors

$$\mathcal{C} \xrightarrow{F} \mathcal{D} \xrightarrow{G} \mathcal{E}.$$

The sensible way to define the composite functor $GF : \mathcal{C} \to \mathcal{E}$ is mapping the objects x of $\mathcal{C}$ to the objects $GF(x)$ of $\mathcal{E}$, and the morphisms $f : x \to y$ of $\mathcal{C}$ to the morphisms $GF(f) : GF(x) \to GF(y)$ of $\mathcal{E}$. That being set, the composition is associative and there is an identity functor too.

There are all the conditions, so what prevents us to consider a category — we can call **Cat** — that has categories as objects and functors as morphisms?

If we work upon NBG, we can think of any proper class as a category, for this statement have a closer look at Example 1.6.4. What happens now is that the class of objects of **Cat** has an element that is a proper class, which isn't legal in NBG.

Is a category of *locally small* categories and functors problematic? Take $\mathcal{C}$ such that $\mathcal{C}(a, b)$ is a proper class for some a and b objects: consider $\mathcal{C}/b$. In this case $|\mathcal{C}/b|$ is a proper class too, and here we go again.

Now what? If we stick to NBG, this is a limit we have to take into account. From now on, **Cat** is the category of *small* categories and functors between small categories.

11. How theorems are named sometimes varies, so for sake of clarity let us explicit the statement we are referring to: Let G and H be two groups, $f : G \to H$ an homomorphism and N some normal subgroup of G. Consider also the homomorphism $p_N : G \to G/N$, $p_N(x) := xN$. If $N \subseteq \ker f$ then there exists one and only one homomorphism $\overline{f} : G/N \to H$ such that $f = \overline{f} p_N$. (Moreover, $\overline{f}$ is surjective if and only if so is f.)

1.7 The hom functor

In a locally small [have we defined somewhere that?] category $\mathcal{C}$, take two morphisms

$$\begin{tikzcd} a_1 & & a_2 \arrow[d, "g_2"] \\ b_1 \arrow[u, "g_1"] & \text{and} & b_2 \end{tikzcd}$$

For $f : a_1 \to a_2$ we have the composite $g_2 f g_1 : b_1 \to b_2$, that is a function

$$\mathcal{C}(a_1, a_2) \to \mathcal{C}(b_1, b_2).$$

We will refer to this function using lambda calculus notation: $\lambda f.g_2 f g_1$. If we have

$$\begin{tikzcd} a_1 \arrow[r, "f"] & a_2 \arrow[d, "g_2"] \\ b_1 \arrow[u, "g_1"] & b_2 \arrow[d, "h_2"] \\ c_1 \arrow[u, "h_1"] & c_2 \end{tikzcd}$$

we can say that

$$\lambda f.(h_2 g_2) f (g_1 h_1) = (\lambda f'.h_2 f' h_1)(\lambda f.g_2 f g_1),$$

which can be derived by uniquely using the associativity of the composition. Another remarkable property can be obtained when $a_1 = b_1$, $g_1 = 1_{a_1}$, $a_2 = b_2$ and $g_2 = 1_{a_2}$:

$$\lambda f.1_{a_2} f 1_{a_1} = \lambda f.f = 1_{\mathcal{C}(a_1, b_1)}.$$

There is functoriality, to understand that, we need to package all this machinery in one functor. The functor we are looking for is

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

which takes every $(x, y) \in |\mathcal{C}^{\mathrm{op}} \times \mathcal{C}|$ to $\mathcal{C}(x, y)$ and

$$\hom_{\mathcal{C}} \left(\begin{array}{c} a_1 \\ {\scriptstyle g_1}\uparrow \\ b_1 \end{array} , \begin{array}{c} a_2 \\ \downarrow{\scriptstyle g_2} \\ b_2 \end{array} \right) \left(a_1 \xrightarrow{f} a_2 \right) := g_2 f g_1.$$

1.8 Constructions involving functors

[Yet to be TEXed...]

1.9 Natural transformations

For $\mathcal{C}$ and $\mathcal{D}$ categories and $F, G : \mathcal{C} \to \mathcal{D}$ functors, a *transformation* from F to G amounts at having for every $x \in |\mathcal{C}|$ one morphism $F(x) \to G(x)$ of $\mathcal{D}$. In other words, a transformation is aimed to measure the difference of two parallel functor by the unique means we have, viz morphisms.

In general, we stick to the following convention: if η is the name of a transformation from F to G, then η_x indicates the component $F(x) \to G(x)$ of the transformation.

We are not interested in all transformations, of course.

Definition 1.9.1 (Natural transformations)**.** A transformation η from a functor $F : \mathcal{C} \to \mathcal{D}$ to a functor $G : \mathcal{C} \to \mathcal{D}$ is said to be *natural* whenever for every $a, b \in |\mathcal{C}|$ and $f \in \mathcal{C}(a, b)$ the square

$$\begin{tikzcd} F(a) \arrow[r, "\eta_a"] \arrow[d, "F(f)"'] & G(a) \arrow[d, "G(f)"] \\ F(b) \arrow[r, "\eta_b"'] & G(b) \end{tikzcd}$$

commutes. This property is the 'naturality' of η.

There are some notations for referring to natural transformations: one may write $\eta : F \Rightarrow G$ or even

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

if they want to explicit also categories.

Natural transformations can be composed: taken two consecutive natural transformations

$$\begin{tikzcd} \mathcal{C} \arrow[r, bend left=60, "F"] \arrow[r, "G" description] \arrow[r, bend right=60, "H"'] \arrow[r, phantom, bend left=30, "\Downarrow\eta"] \arrow[r, phantom, bend right=30, "\Downarrow\theta"] & \mathcal{D} \end{tikzcd}$$

the transformation $\theta\eta$ that have the components $\theta_x \eta_x : F(x) \to H(x)$, for $x \in |\mathcal{C}|$ of $\mathcal{D}$ is natural. Such composition is associative. Moreover, for every functor $F : \mathcal{C} \to \mathcal{D}$ there is the natural transformation $1_F : F \Rightarrow F$ with components $1_{F(x)} : F(x) \to F(x)$, for $x \in |\mathcal{C}|$; they are identities in categorial sense:

$$\eta 1_F = \eta \text{ for every natural transformation } \eta : F \Rightarrow G$$
$$1_F \mu = \mu \text{ for every natural transformation } \mu : H \Rightarrow F.$$

All this suggests to, given two categories $\mathcal{C}$ and $\mathcal{D}$, form a category with functors $\mathcal{C} \to \mathcal{D}$ as objects and natural transformations as morphism, them being composable as explained above. [...]

[Consider `https://mathoverflow.net/q/39073`...]

1.10 Equivalent categories

Let us give a definition that will motivate our discourse.

Definition 1.10.1 (Full- and faithfulness)**.** A functor $F : \mathcal{C} \to \mathcal{D}$ is said *full*, respectively *faithful*, whenever for every $a, b \in |\mathcal{C}|$ the functions

$$F : \mathcal{C}(a, b) \to \mathcal{D}(F(a), F(b))$$

are surjective, respectively injective; we say that F is *fully faithful* [how lame, lol...] whenever it is both full and faithful.

What do we want 'two categories are the same' to mean? [Craft a nicer exposition... Let us try with categories being isomorphic first, and then with *essentially surjective* functors. Talk about *skeletons* of categories, and how can help to say whether two categories are equivalent.]

Example 1.10.2 (A functor $\mathbf{Mat}_k \to \mathbf{FDVect}_k$)**.** For k field, consider the functor

$$M : \mathbf{Mat}_k \to \mathbf{FDVect}_k$$

that maps $n \in |\mathbf{Mat}_k| = \mathbb{N}$ to $M(n) := k^n$ and $A \in \mathbf{Mat}_k(r,s)$ to the linear function

$$\begin{aligned} M_A &: k^r \to k^s \\ M_A(x) &= Ax. \end{aligned}$$

(Here the elements of k^n are matrices of type $n \times 1$.) [...]

1.11 The Yoneda Lemma

[This section is to be moved elsewhere...][Maybe, I should stick to *small* categories...] [Use `cramped` for some `tikzcds`...]

We have the *evaluation functor*

$$\mathrm{ev}_{\mathcal{C}} : \mathcal{C} \times [\mathcal{C}, \mathbf{Set}] \to \mathbf{Set}$$

that on objects

$$\mathrm{ev}_{\mathcal{C}}(x, F) := F(x)$$

and on morphisms

$$\mathrm{ev}\left(\begin{array}{c} a \\ \big\downarrow f \\ b \end{array}, \begin{array}{c} F \\ \eta \Downarrow \\ G \end{array}\right) := \eta_b F(f) = G(f)\eta_a.$$

Lemma 1.11.1 (A lemma for the Yoneda Lemma)**.** Let $\mathcal{C}$ be a locally small category. Then for every $x \in |\mathcal{C}|$ and functor $F : \mathcal{C} \to \mathbf{Set}$,

$$[\mathcal{C}, \mathbf{Set}](\mathcal{C}(x,-), F) \cong F(x).$$

In particular, the classes $[\mathcal{C}, \mathbf{Set}](\mathcal{C}(x,-), F)$ are actual sets.

Proof. For x and F as in the hypothesis, take functions

$$\lambda_{x,F} : [\mathcal{C}, \mathbf{Set}](\mathcal{C}(x,-), F) \to F(x)\,,\ \lambda_{x,F}(\alpha) := \alpha_x(1_x).$$

Now, for every $a \in F(x)$ we have the transformation $\mu_{x,F}(a)$ from $\mathcal{C}(x,\bullet)$ to F which has the components

$$\mathcal{C}(x,c) \to F(c)\,,\ f \to \big(F(f)\big)(a);$$

it is immediate to show that it is natural. Thus we have functions

$$\mu_{x,F} : F(x) \to [\mathcal{C}, \mathbf{Set}](\mathcal{C}(x,-), F).$$

We prove

$$\begin{aligned} \lambda_{x,F}\mu_{x,F} &= 1_{F(x)} \\ \mu_{x,F}\lambda_{x,F} &= 1_{[\mathcal{C},\mathbf{Set}](\mathcal{C}(x,-),F)}. \end{aligned}$$

In fact, for $a \in F(x)$ we have $\lambda_{x,F}\big(\mu_{x,F}(a)\big)$ is the component $\mathcal{C}(x,x) \to F(x)$ of $\mu_{x,F}(a)$ evaluated at 1_x, viz $1_{F(x)}(a) = a$. Besides, for if $\alpha : \mathcal{C}(x,\bullet) \Rightarrow F$ natural transformation we have $\mu_{x,F}\big(\lambda_{x,F}(\alpha)\big) = \mu_{x,F}\big(\alpha_x(1_x)\big)$ is the natural transformation $\mathcal{C}(x,\bullet) \Rightarrow F$ with components

$$\mathcal{C}(x,c) \to F(c)\,,\ f \to \big(F(f)\big)(\alpha_x(1_x)) = \alpha_c(f)$$

for $c \in |\mathcal{C}|$; that is $\mu_{x,F}\lambda_{x,F}(\alpha) = \alpha$. The proof is complete now. □

Let $\mathcal{C}$ be a locally small category. We have the functor

$$\mathcal{Y}_{\mathcal{C}} : \mathcal{C} \times [\mathcal{C}, \mathbf{Set}] \to \mathbf{Set}$$

given on objects as follows

$$\mathcal{Y}_{\mathcal{C}}(x, F) := [\mathcal{C}, \mathbf{Set}]\big(\mathcal{C}(x, -), F\big)$$

and on morphisms

$$\left[\mathcal{Y}_{\mathcal{C}}\left(\begin{array}{c} a \\ f\big\downarrow \\ b \end{array}, \begin{array}{c} F \\ \eta\Downarrow \\ G \end{array}\right)\right]\left(\begin{array}{c} \mathcal{C}(a,-) \\ \Downarrow\alpha \\ F \end{array}\right) := \left\{\mathcal{C}(b,c) \xrightarrow{\eta_c\alpha_c(\_f)} G(c) \,\middle|\, c \in |\mathcal{C}|\right\}.$$

Observe that Lemma 1.11.1 solves annoying size issues in the definition of $\mathcal{Y}_{\mathcal{C}}$ on objects. While the statement of this lemma is important for technical reasons, its proof guides us to the following completion.

Proposition 1.11.2 (Yoneda Lemma)**.** For $\mathcal{C}$ locally small category, $\mathcal{Y}_{\mathcal{C}} \cong \mathrm{ev}_{\mathcal{C}}$.

Proof. The transformation $\lambda : \mathcal{Y}_{\mathcal{C}} \Rightarrow \mathrm{ev}_{\mathcal{C}}$ having as components the functions $\lambda_{x,F}$ of the proof of Lemma 1.11.1 is natural, that is

$$\begin{tikzcd}
\mathcal{Y}_{\mathcal{C}}(a,F) \arrow[r, "\lambda_{a,F}"] \arrow[d, "\mathcal{Y}_{\mathcal{C}}(f,\eta)"'] & \mathrm{ev}_{\mathcal{C}}(a,F) \arrow[d, "\mathrm{ev}_{\mathcal{C}}(f,\eta)"] \\
\mathcal{Y}_{\mathcal{C}}(b,G) \arrow[r, "\lambda_{b,G}"'] & \mathrm{ev}_{\mathcal{C}}(b,G)
\end{tikzcd}$$

commutes for every $f \in \mathcal{C}(a,b)$ and $\eta \in [\mathcal{C}, \mathbf{Set}](F,G)$. In fact, for every natural transformation $\eta \in \mathcal{Y}_{\mathcal{C}}(a,F)$ we have

$$\mathrm{ev}_{\mathcal{C}}(f,\eta)\big(\lambda_{a,F}(\alpha)\big) = \eta_b\alpha_b\mathcal{C}(a,f)(1_a) = \eta_b\alpha_b f;$$

besides,

$$\lambda_{b,G}\big(\mathcal{Y}_{\mathcal{C}}(f,\eta)(\alpha)\big) = \eta_b\alpha_b(\_f)(1_b) = \eta_b\alpha_b f.$$

We can conclude λ is an isomorphism, as the proof of Lemma 1.11.1 tells us its components are isomorphisms. □

2

Limits and Colimits

To read this chapter you need to know what categories and a functors are, that is the first two chapters. About natural transformations, only the definition of natural transformation is required to formalize the definition of (co)limit.

2.1 Definition

[Expand this section with details about duality and TEX some initial exercises.]

Definition 2.1.1 (Limits & colimits)**.** Let $\mathcal{I}$ and $\mathcal{C}$ be two categories. For every object v of $\mathcal{C}$ we have the *constant functor*

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

where $k_v(i) := v$ for every object i and $k_v(f) := 1_v$ for every morphism f. [Later, in the chapter of the adjunctions, we will introduce one functor $\Delta : \mathcal{C} \to [\mathcal{I}, \mathcal{C}]$.] A *limit* of a functor $F : \mathcal{I} \to \mathcal{C}$ is some object v of $\mathcal{C}$ with a natural transformation $\lambda : k_v \Rightarrow F$ such that: for any object a of $\mathcal{C}$ and $\mu : k_a \Rightarrow F$ there is one and only one $f : a \to v$ of $\mathcal{C}$ such that

$$\begin{tikzcd} a \arrow[d, "f"'] \arrow[dr, "\mu_i"] & \\ v \arrow[r, "\lambda_i"'] & F(i) \end{tikzcd}$$

commutes for every i in $\mathcal{I}$. A *colimit*, instead, is an object u of $\mathcal{C}$ together with a $\chi : F \Rightarrow k_u$ that has the property: for every object b of $\mathcal{C}$ and $\xi : F \Rightarrow k_b$ there exists one and only one $g : u \to b$ of $\mathcal{C}$ that makes

$$\begin{tikzcd} u \arrow[d, "g"'] & F(i) \arrow[l, "\chi_i"'] \arrow[dl, "\xi_i"] \\ b & \end{tikzcd}$$

commute for every i in $\mathcal{I}$.

Example 2.1.2. We have already seen how a preordered set is a category; in this example let us employ $\mathbb{N}$ with the usual ordering $\leq$. First of all, let us figure out what cones and cocones of functors $H : \mathbb{N} \to \mathcal{C}$ are. Such functors, in other words, are sequence of objects and morphisms of $\mathcal{C}$ so arranged

$$H_0 \xrightarrow{\partial_0} H_1 \xrightarrow{\partial_2} \cdots \xrightarrow{\partial_{n-1}} H_n \xrightarrow{\partial_n} H_{n+1} \xrightarrow{\partial_{n+1}} \cdots$$

In this case, a cone on H is a collection $\{\alpha_i : A \to H_i \mid i \in \mathbb{N}\}$ such that $\alpha_j = \partial_{j-1} \cdots \partial_i \alpha_i$ for every $i, j \in \mathbb{N}$ such that $i < j$; remember that if $j > i$ there is no morphism $H_j \to H_i$ and, for $i \in |\mathcal{I}|$, the morphism $H_i \to H_i$ is the identity. [Continue after you have fixed some parts before.]

Example 2.1.3. [Rewrite.] Let $\mathcal{C}$ be a category and $\mathbf{1}$ a category that has one object and one morphism, and take a functor $f : \mathbf{1} \to \mathcal{C}$, some $v \in \mathcal{C}$ and the corresponding constant functor $k_v : \mathbf{1} \to \mathcal{C}$. A natural transformation $\zeta : k_v \Rightarrow f$ amounts of a single morphism $v \to \widetilde{f}$ of $\mathcal{C}$, where $\widetilde{f}$ indicates the image of the unique object of $\mathbf{1}$ via f. Thus, a limit of f is some $v \in |\mathcal{C}|$ and a morphism $\lambda : v \to \widetilde{f}$ of $\mathcal{C}$ such that: for every object u and morphism $\mu : u \to \widetilde{f}$ in $\mathcal{C}$, there is a unique morphism $u \to v$ of $\mathcal{C}$ that makes

$$\begin{tikzcd}
u \arrow[dr, "\mu"] \arrow[dd] & \\
& \widetilde{f} \\
v \arrow[ur, "\lambda"'] &
\end{tikzcd}$$

commute.

Exercise 2.1.4. What are colimts of functors $\mathbf{1} \to \mathcal{C}$?

Example 2.1.5. [Rewrite.] Consider a monoid (viz a single object category) $\mathcal{G}$: for the scope of this example we write G for the set of the morphisms of $\mathcal{G}$. Let $F : \mathcal{G} \to \mathbf{Set}$ be a functor, and let $\widehat{F}$ indicate the F-image of the unique object of $\mathcal{G}$ whilst, for $f \in G$, $\widehat{f}$ the function $F(f) : \widehat{F} \to \widehat{F}$. Now, being $k_X : \mathcal{G} \to \mathbf{Set}$ the functor constant at X, with X a set, a natural transformation $\lambda : F \Rightarrow k_X$ is a morphism $\lambda : \widehat{F} \to X$ such that $\lambda = \lambda \widehat{f}$ for every $f \in G$. These two things, the set X and the function λ, together are a colimit of F whenever

> for every set Y and function $\mu : \widehat{F} \to Y$ such that $\mu = \mu \widehat{f}$ for every $f \in G$ there exists one and only one function $h : X \to Y$ such that $\mu = h\lambda$.

[Is that thing even interesting?] [Write about functors $\mathcal{G} \to \mathbf{Set}$...]

[Write about duality here. Explain how limits and colimits are dual...]

The following is very basic property: limits of a same functor are are essentially the same.

Proposition 2.1.6. Let $F : \mathcal{I} \to \mathcal{C}$ be a functor. If $\{\eta_i : a \to F(i) \mid i \in |\mathcal{I}|\}$ and $\{\theta_i : b \to F(i) \mid i \in |\mathcal{I}|\}$ are limits of F, then $a \cong b$.

Proof. By definition of limit, we a have a unique $f : a \to b$ and a unique $g : b \to a$ making the triangles in

$$\begin{tikzcd}
a \arrow[dr, "\eta_i"] \arrow[dd, bend left, "f"] & \\
& F(i) \\
b \arrow[ur, "\theta_i"'] \arrow[uu, bend left, "g"] &
\end{tikzcd}$$

commute for every object i of $\mathcal{I}$. In this case,

$$\eta_i = \theta_i f = \eta_i(gf)$$
$$\theta_i = \eta_i g = \theta_i(fg)$$

Invoking again the universal property of limits, $gf = 1_a$ and $fg = 1_b$. □

Fortunately, there are few shapes that are both ubiquitous and simple. This section is dedicated to them, while in the successive one we will prove (Proposition 2.6.2) that if some simple functors have limits, then all the functors do have limits.

2.2 Terminal and initial objects

Definition 2.2.1 (Terminal & initial objects)**.** For $\mathcal{C}$ category, the limits of the empty functor $\varnothing \to \mathcal{C}$ are called *terminal objects* of $\mathcal{C}$, whereas the colimits *initial objects*.

Let us expand the definition above so that we can can look inside. A cone over the empty functor $\varnothing \to \mathcal{C}$ with vertex a is a natural transformation

$$\begin{tikzcd} \varnothing \arrow[r, bend left, "k_a"] \arrow[r, bend right] \arrow[r, phantom, "\Downarrow"] & \mathcal{C} \end{tikzcd} .$$

Here, the empty functor is k_a because there is at most one functor $\varnothing \to \mathcal{C}$. Again, because there must be a unique one, our natural transformation is the empty transformation, viz the one devoid of morphisms. A similar reasoning leads us to the following explicit definition of terminal and initial object.

Definition 2.2.2 (Terminal and initial objects, explicit)**.** Let $\mathcal{C}$ be a category.

- A *terminal object* of $\mathcal{C}$ is an object 1 of $\mathcal{C}$ such that for every object x of $\mathcal{C}$ there exists one and only one $x \to 1$ in $\mathcal{C}$.
- An *initial object* of $\mathcal{C}$ is an object 0 in $\mathcal{C}$ such that for every object x in $\mathcal{C}$ there exists one and only one morphism $0 \to x$ in $\mathcal{C}$.

Example 2.2.3 (Empty set and singletons)**.** It may sound weird, but for every set X there does exist a function $\varnothing \to X$; moreover, it is the unique one. To get this, think set-theoretically: a function is any subset of $\varnothing \times X$ that has the property we know. But $\varnothing \times X = \varnothing$, so its unique subset is $\varnothing$. This set is a function from $\varnothing$ to X since the statement

for every $a \in \varnothing$ there is one and only one $b \in X$ such that $(a, b) \in \varnothing$

is a *vacuous truth*. So $\varnothing$ is an initial object of **Set**. This case is quite particular, since the initial objects of **Set** are actually equal to $\varnothing$.
Now let us look for terminal objects in **Set**. Take an arbitrary set X: there is exactly one function from X to any singleton, that is singletons are terminal object of **Set**. Conversely, by Proposition 2.1.6, the terminal objects of **Set** must be singletons.

Exercise 2.2.4. Trivial groups — there is a unique way a singleton can be a group — are either terminal and initial objects of **Grp**.

Construction 2.2.5. Let us introduce a nice category that allows us to express some nice and simple facts in Mathematics. Let $\mathcal{C}$ and $\mathcal{J}$ two categories, a one of its objects and take a functor $F : \mathcal{J} \to \mathcal{C}$. We have the category $(a \downarrow F)$, this made:

- the objects are the morphisms $a \to F(x)$ of $\mathcal{C}$, with x being an object of $\mathcal{J}$;

- the morphisms from $f : a \to F(x)$ to $g : a \to F(y)$ are the morphisms $h : x \to y$ of $\mathcal{J}$ such that

$$\begin{tikzcd} & F(x) \arrow[dd, "F(h)"] \\ a \arrow[ur, "f"] \arrow[dr, "g"'] & \\ & F(y) \end{tikzcd}$$

  commutes;
- the composition is that of $\mathcal{J}$.

Example 2.2.6. [Rewrite this example in $\mathbf{Mod}_R$, for R ring? Yes...] In Linear Algebra we have a nice theorem:

> Let V be a vector space over a field k and $S \subseteq V$ a base. For every vector space W over k and function $\phi : S \to W$ there exists a unique 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.

In other words, this statement says that a linear function is completely determined by what it does with the vectors of S. We will consider now two functors

$$\mathbf{Set} \xrightarrow{\langle\cdot\rangle} \mathbf{Vect}_k \xrightarrow{U} \mathbf{Set}.$$

The first one takes a set S and produces the vector space on k

$$\langle S \rangle := \left\{ \sum_{x \in S} \lambda_x x \,\middle|\, \lambda : S \to k,\ \lambda_x \neq 0 \text{ for finitely many times} \right\}$$

(considered with two obvious operations). Furthermore, a function of sets $f : S \to T$ induces a linear function $\langle f \rangle : \langle S \rangle \to \langle T \rangle$ defined by

$$\langle f \rangle \left(\sum_{x \in S} \lambda_x x \right) := \sum_{x \in S} \lambda_x f(x)$$

where $\lambda : S \to k$ is almost always null. The functoriality of $\langle\cdot\rangle$ is just a matter of quick controls. [Do we really need all that machinery?] The functor U instead takes vector spaces and returns the correspondent set of vectors; we write $U(V) := V$, but observe that in **Set** we don't care anymore of the vector structure of V. Similarly, it takes linear functions and the return them: but, since U lands onto **Set**, who cares about linearity there? (We may say that U is the 'inclusion' of $\mathbf{Vect}_k$ into **Set**.) [Talk about forgetful functors elsewhere...] All this words allow us restate the aforementioned theorem as:

> For if S is a set, the inclusion $S \hookrightarrow \langle S \rangle$ is an initial object of $(S \downarrow U)$.

Exercise 2.2.7. In the previous example some details are omitted: you can be more talkative, though. However, it is really worth to think about such examples — not only because we will meet such pattern later under the vest of adjunctions. You may also look for another examples of similar kind, I'm sure you will find some.

Example 2.2.8 (Isomorphism Theorem for Set Theory)**.** We have defined **Eqv** earlier, recall it here. We have the functor

$$j : \mathbf{Set} \to \mathbf{Eqv}$$

that maps sets X to setoids X together with the equality relation, and functions $f : X \to Y$ to themselves. To get the mood for this example, sets *are* setoids where the equivalence relation is equality and functions *are* functoids between such setoids. In this case, the classical theorem

> Let X and Y be two sets and $\sim$ an equivalence relation on X. For every function $f : X \to Y$ such that $f(a) = f(b)$ for every $a, b \in X$ with $a \sim b$, there exists one and only one function $\overline{f} : X/\!\sim\, \to Y$ such that
>
> $$\begin{tikzcd} X \arrow[rr, "f"] \arrow[dr, "p"'] & & Y \\ & X/\!\sim \arrow[ur, "\overline{f}"'] & \end{tikzcd}$$
>
> commutes, where $p : X \to X/\!\sim$ is the canonical projection.

can be restated as follows

> the canonical projection $p : (X, \sim) \to X/\!\sim$ is initial in $(X, \sim) \downarrow j$.

Example 2.2.9 (Recursion)**.** In Set Theory, there is a nice theorem, the *Recursion Theorem*:

> Let $(\mathbb{N}, 0, s)$ be a Peano Model, where $0 \in \mathbb{N}$ and $s : \mathbb{N} \to \mathbb{N}$ is its successor function. For every pointed set X, $a \in X$ and $f : X \to X$ there exists one and only one function $x : \mathbb{N} \to X$ such that $x_0 = a$ and $x_{s(n)} = f(x_n)$ for every $n \in \mathbb{N}$.

Here, by Peano Model we mean a set $\mathbb{N}$ that has one element, we write 0, stood out and a function $s : \mathbb{N} \to \mathbb{N}$ such that, all this complying some rules:

1. s is injective;
2. $s(x) \neq 0$ for every $x \in \mathbb{N}$;
3. for if $A \subseteq \mathbb{N}$ has 0 and $s(n) \in A$ for every $n \in A$, then $A = \mathbb{N}$.

We show now how we can involve Category Theory in this case. First of all, we need a category where to work.
The statement is about things made as follows:

> a set X, one distinguished $x \in X$ and one function $f : X \to X$.

[Is there a name for these things?] We may refer to such new things by barely a triple (X, a, f), but we prefer something like this:

$$1 \xrightarrow{x} X \xrightarrow{f} X,$$

where 1 is any singleton, as usual. Peano Models are such things, with some additional properties. It is told about the existence and the uniqueness of a certain function. We do not want mere functions, of course: given

$$1 \xrightarrow{x} X \xrightarrow{f} X \text{ and } 1 \xrightarrow{y} Y \xrightarrow{g} Y,$$

we take the functions $r : X \to Y$ such that

$$\begin{tikzcd} & X \arrow[r, "f"] \arrow[d, "r"] & X \arrow[d, "r"] \\ 1 \arrow[ur, "x"] \arrow[dr, "y"'] & & \\ & Y \arrow[r, "g"'] & Y \end{tikzcd}$$

commutes and nothing else. [Is there a name for such functions?] These ones are the things we want to be morphisms. Suppose given

$$\begin{tikzcd} & X \arrow[r, "f"] \arrow[d, "p"] & X \arrow[d, "p"] \\ 1 \arrow[ur, "x"] \arrow[r, "y"] \arrow[dr, "z"'] & Y \arrow[r, "g"] \arrow[d, "q"] & Y \arrow[d, "q"] \\ & Z \arrow[r, "h"'] & Z \end{tikzcd}$$

where all the squares and triangles commute: thus we obtain the commuting

$$\begin{tikzcd} & X \arrow[r, "f"] \arrow[d, "qp"] & X \arrow[d, "qp"] \\ 1 \arrow[ur, "x"] \arrow[dr, "z"'] & & \\ & Z \arrow[r, "h"'] & Z \end{tikzcd}$$

This means that composing two morphisms as functions in **Set** produces a morphism. This is how we want composition to defined in this context. This choice makes the categorial axioms automatically respected. We call this category **Peano**. [Unless there is a better naming, of course.]
Being the environment set now, the Recursion Theorem becomes more concise:

Peano Models are initial objects of **Peano**.

By Proposition 2.1.6, any other initial object of **Peano** are isomorphic to some Peano Model: does this mean its initial objects are Peano Models? (Exercise.)

Exercise 2.2.10 (Induction ⇔ Recursion)**.** In **Set**, suppose you have $1 \xrightarrow{0} \mathbb{N} \xrightarrow{s} \mathbb{N}$, where s is injective and $s(n) \neq 0$ for every $n \in \mathbb{N}$. Demonstrate that the following statements are equivalent:

1. for if $A \subseteq \mathbb{N}$ has 0 and $s(n) \in A$ for every $n \in A$, then $A = \mathbb{N}$;
2. $1 \xrightarrow{0} \mathbb{N} \xrightarrow{s} \mathbb{N}$ is an initial object of **Peano**.

(1) ⇒ (2) proves the Recursion Theorem, whereas (2) ⇒ (1) requires you to codify a proof by induction into a recursion. Try it, it could be nice. [Prepare hints...]

Limits (colimits) are terminal (initial) objects of appropriate categories. Definition 2.2.2 does not make reference to limits and colimits as stated in Definition 2.1.1: thus, you can say what terminal and initial objects are and then tell what limits and colimits are in terms of terminal and initial objects.

Construction 2.2.11 (Category of cones)**.** For $\mathcal{C}$ category, let $F : \mathcal{I} \to \mathcal{C}$ be a functor. Then we define the *category of cones* over F as follows.

- The objects are the cones over F.
- For $\alpha := \left\{a \xrightarrow{\alpha_i} F(i)\right\}_{i\in|\mathcal{I}|}$ and $\beta := \left\{b \xrightarrow{\beta_i} F(i)\right\}_{i\in|\mathcal{I}|}$ two cones, the morphisms from α to β are the morphisms $f : a \to b$ of $\mathcal{C}$ such that

$$\begin{tikzcd} a \arrow[dr, "\alpha_i"] \arrow[dd, "f"'] & \\ & F(i) \\ b \arrow[ur, "\beta_i"'] & \end{tikzcd}$$

  commutes for every $i \in |\mathcal{I}|$.
- The composition of morphisms here is the same as that of $\mathcal{C}$.

We write such category as $\mathbf{Cn}_F$. We define also the *category of cocones* over F, written as $\mathbf{CoCn}_F$.

- The objects are the cocones over F.
- For $\alpha := \left\{F(i) \xrightarrow{\alpha_i} a\right\}_{i\in|\mathcal{I}|}$ and $\beta := \left\{F(i) \xrightarrow{\beta_i} b\right\}_{i\in|\mathcal{I}|}$ cocones, the morphisms from α to β are the morphisms $f : a \to b$ of $\mathcal{C}$ such that

$$\begin{tikzcd} & a \arrow[dl, "\alpha_i"'] \arrow[dd, "f"] \\ F(i) & \\ & b \arrow[ul, "\beta_i"] \end{tikzcd}$$

  commutes for every $i \in |\mathcal{I}|$.
- The composition of morphisms here is the same as that of $\mathcal{C}$.

It is quite immediate in either of the cases to show that categorial axioms are verified.

Proposition 2.2.12. For $\mathcal{C}$ category and $F : \mathcal{I} \to \mathcal{C}$ functor,

- limits of F are terminal objects of $\mathbf{Cn}_F$ and viceversa.
- colimits of F are initial objects of $\mathbf{CoCn}_F$ and viceversa.

Proof. This is **exercise 2.2.13**. □

2.3 Products and coproducts

Let $\mathcal{C}$ be a category and I a discrete category (that is a class). We have seen how functors $x : I \to \mathcal{C}$ are exactly families $\{x_i \mid i \in I\}$ of objects of $\mathcal{C}$. We call *(co)products* of $\{x_i \mid i \in I\}$ the (co)limits of $\{x_i \mid i \in I\}$. Let us put this definition into more explicit terms.

First of all, let us make clear what cones over a collection $\{x_i \mid i \in I\}$ are. For $p \in |\mathcal{C}|$ and $k_p : I \to \mathcal{C}$ the functor constant at p, a natural transformation

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

is just a family $\{p \to x_i \mid i \in I\}$ of morphisms of $\mathcal{C}$. In this fortunate case, the naturality condition automatically holds because I has no morphisms other than identities. Similarly, one can easily make explicit what cocones are.

Definition 2.3.1 (Products & coproducts)**.** Let $\mathcal{C}$ be a category. A *product* of a family $\{x_i \mid i \in I\}$ of objects in $\mathcal{C}$ is any family $\{\mathrm{pr}_i : p \to x_i \mid i \in I\}$ of morphisms of $\mathcal{C}$, usually called *projections*, respecting the following property: for every family $\{f_i : a \to x_i \mid i \in I\}$ of morphisms of $\mathcal{C}$ there exists one and only one $h : a \to p$ of $\mathcal{C}$ such that

$$\begin{tikzcd} a \arrow[d, "h"'] \arrow[dr, "f_i"] & \\ p \arrow[r, "\mathrm{pr}_i"'] & x_i \end{tikzcd}$$

commutes for every $i \in I$. A *coproduct* of $\{x_i \mid i \in I\}$ of objects of $\mathcal{C}$ is any family $\{\mathrm{in}_i : x_i \to q \mid i \in I\}$ of morphisms of $\mathcal{C}$, often referred to as *injections*, having the property: for every family $\{g_i : x_i \to b \mid i \in I\}$ of morphisms of $\mathcal{C}$ there exists one and only one $k : q \to b$ of $\mathcal{C}$ such that

$$\begin{tikzcd} b & \\ q \arrow[u, "k"] & x_i \arrow[ul, "g_i"'] \arrow[l, "\mathrm{in}_i"] \end{tikzcd}$$

commutes for every $i \in I$.

Example 2.3.2 (Infima and suprema in prosets)**.** Consider a proset $(\mathbb{P}, \leq)$ and a subset S of $\mathbb{P}$. In this instance, a product of S is some $p \in \mathbb{P}$ such that:

1. $p \leq x$ for every $x \in S$;
2. for every $p' \in \mathbb{P}$ such that $p' \leq x$ for every $x \in S$ we have $p' \leq p$.

If we have quick look to some existing mathematics, cones over S are what are called *lower bounds* of S. An *infimum* of S is any of the greatest lower bounds for S.
On the other hand, a coproduct of S is some $q \in \mathbb{P}$ such that:

1. $x \leq q$ for every $x \in S$;
2. for every $q' \in \mathbb{P}$ such that $x \leq q'$ for every $x \in S$ we have $q \leq q'$.

In other words, the cones over S are precisely the *upper bounds* of S. A *supremum* of S is any of the lowest upper bounds for S.
There is a dedicated notations for such elements, if $(\mathbb{P}, \leq)$ is a poset: the infimum of S is written as $\inf S$, whereas $\sup S$ is the supremum of S. If the elements of S are indexed, that is $S = \{x_i \mid i \in I\}$, then it is customary to write $\inf_{i \in I} x_i$ and $\sup_{i \in I} x_i$.

Exercise 2.3.3. Prosets provide some examples in which some subsets does not have infima or suprema.

Now let us turn our attention to a pair of quite ubiquitous constructs.

Example 2.3.4 (Cartesian product)**.** Given a family of sets $\{X_\alpha \mid \alpha \in \Gamma\}$, we have the corresponding *Cartesian product*

$$\prod_{\alpha \in \Gamma} X_\alpha := \left\{ f : \Gamma \to \bigcup_{\alpha \in \Gamma} X_\alpha \,\middle|\, f(\lambda) \in X_\lambda \text{ for every } \lambda \in \Gamma \right\},$$

whose elements are the *choices* from $\{X_\alpha \mid \alpha \in \Gamma\}$. As the name indicates, a choice f for every $\lambda \in \Gamma$ indicates one element of X_λ. Our product comes with the *projections*, one for each $\mu \in \Gamma$,

$$\begin{aligned} \mathrm{pr}_\mu : \prod_{\alpha\in\Gamma} X_\alpha &\to X_\mu \\ \mathrm{pr}_\mu(f) &:= f(\mu). \end{aligned}$$

Now, any family of functions $\{f : A \to X_\alpha \mid \alpha\}$ ca be compressed into one function

$$\prod_{\alpha\in\Gamma} f_\alpha : A \to \prod_{\alpha\in I} X_\alpha.$$

by defining $\left(\prod_{\alpha\in\Gamma} f_\alpha\right)(a)$ to be the function $\Gamma \to \bigcup_{\alpha\in\Gamma} X_\alpha$ mapping $\mu \in \Gamma$ to $f_\mu(a)$. It is simple to show that

$$\begin{tikzcd} A \arrow[drr, "f_\mu"] \arrow[d, "\prod_{\alpha\in\Gamma} f_\alpha"'] & & \\ \prod_{\alpha\in\Gamma} X_\alpha \arrow[rr, "\mathrm{pr}_\mu"'] & & X_\mu \end{tikzcd}$$

commutes for every $\mu \in \Gamma$. Moreover, $\prod_{\alpha\in\Gamma} f_\alpha$ is the only one that does this. Consider any function $g : A \to \prod_{\alpha\in\Gamma} X_\alpha$ with $f_\mu = \mathrm{pr}_\mu g$ for every $\mu \in \Gamma$: then for every $x \in A$ we have

$$\begin{aligned} \big(g(x)\big)(\mu) = \mathrm{pr}_\mu\big(g(x)\big) = f_\mu(x) = \\ = p_\mu\left(\left(\prod_{\alpha\in\Gamma} f_\alpha\right)(x)\right) = \left(\left(\prod_{\alpha\in\Gamma} f_\alpha\right)(x)\right)(\mu), \end{aligned}$$

that is $g = \prod_{\alpha\in\Gamma} f_\alpha$.

Exercise 2.3.5. It may be simple to reason about the Cartesian product of only two sets X_1 and X_2. In this case, the product is written as $X_1 \times X_2$ and its elements are represented as pairs $(a, b) \in X_1 \times X_2$ rather than functions $f : \{1, 2\} \to X_1 \cup X_2$ with $f(i) \in X_i$ for $i \in \{1, 2\}$. By setting things like this, the 'compression' of two functions $f_1 : A \to X_1$ and $f_2 : A \to X_2$ into a function $A \to X_1 \times X_2$ becomes more obvious.

Example 2.3.6 (Coproduct of sets)**.** For if $\{X_\alpha \mid \alpha \in \Lambda\}$ is a family of sets, we introduce the *disjoint union*

$$\sum_{\alpha\in\Lambda} X_\alpha := \bigcup_{\alpha\in\Lambda} X_\alpha \times \{\alpha\} = \{(x, \alpha) \mid \alpha \in \Lambda,\ x \in X_\alpha\}.$$

While the elements of every member of $\{X_\alpha \mid \alpha \in \Lambda\}$ are amalgamated in $\bigcup_{\alpha\in\Lambda} X_\alpha$, in the disjoint union $\sum_{\alpha\in\Lambda} X_\alpha$ the elements have attached a record of their provenience — in this case, the index of the set they come from. Because of this feature, the elements of $\sum_{\alpha\in\Lambda} X_\alpha$ are called *dependent pairs*. The disjoint union of $\{X_\alpha \mid \alpha \in \Lambda\}$ has one *injection* for each $\alpha \in \Lambda$:

$$\mathrm{in}_\mu : X_\mu \to \sum_{\alpha\in\Lambda} X_\alpha\,,\ \mathrm{in}_\mu(x) := (x, \mu).$$

Similarly to what we have done in the previous example, a family of functions $\{f_\alpha : X_\alpha \to A \mid \alpha \in \Lambda\}$ can be compressed into this one

$$\sum_{\alpha\in\Lambda} f_\alpha : \sum_{\alpha\in\Lambda} X_\alpha \to A$$
$$\left(\sum_{\alpha\in\Lambda} f_\alpha\right)(x,\mu) := f_\mu(x),$$

which, in other words, checks the provenience of an element and give it to an appropriate function f_α. This new function makes the diagram

$$\begin{tikzcd}
A & \\
& X_\mu \arrow[ul, "f_\mu"'] \arrow[dl, "\mathrm{in}_\mu"] \\
\sum_{\alpha\in\Lambda} X_\alpha \arrow[uu, "\sum_{\alpha\in\Lambda} f_\alpha"] &
\end{tikzcd}$$

commute for every $\mu \in \Lambda$, and it is the unique to do this.

Exercise 2.3.7. Prove $\bigcup_{\alpha\in\Lambda} X_\alpha$ is a coproduct if the X_α-s are pairwise disjoint. By the way, $\bigcup_{\alpha\in\Lambda} X_\alpha$ is isomorphic to an appropriate quotient of $\sum_{\alpha\in\Lambda} X_\alpha$. Part of the exercise is to find an equivalence relation $\sim$ on $\sum_{\alpha\in\Lambda} X_\alpha$ and a function

$$\sum_{\alpha\in\Lambda} X_\alpha \to \bigcup_{\alpha\in\Lambda} X_\alpha$$

which maps $\sim$-equivalent elements to the same element.

Exercise 2.3.8. Haskell natively offers the function

```
either :: (a -> c) -> (b -> c) -> Either a b -> c
```

How do `Either a b` and this function fit in the current topic? If you accept this little exercise, remember `Either a b` is defined to be either `Left a` or `Right b`. We haven't talked about the category of types, but it is not be that unseen.

Example 2.3.9 (Product of topological spaces)**.** Consider now a family of topological spaces $\{X_i \mid i \in I\}$ and let us see if we can have a product of topological space in the sense of the Definition above.
In order to talk about product topological space we shall determine a topology over the set $\prod_{i\in I} X_i$. From the Example 2.3.4, we have a nice machinery, but it is all about sets and functions! We define the *product topology* — sometimes called 'Tychonoff topology' — as the smallest among the topologies for $\prod_{i\in I} X_i$ for which all the projections $\mathrm{pr}_j : \prod_{i\in I} X_i \to X_j$ of the Example 2.3.4 are continuous.
The question is now: do these continuous functions form a product in **Top**? Taking a family of continuous functions $\{f_i : A \to X_i \mid i \in I\}$ and looking at the 'underground' **Set**, there does exist one function $\widehat{f} : A \to \prod_{i\in I} X_i$ such that $f_i = \mathrm{pr}_i \widehat{f}$ for every $i \in I$, but we do not know if it is continuous! To give an answer, let us consider the family

$$\mathcal{T} := \left\{ U \subseteq \prod_{i\in I} X_i \text{ open} \,\middle|\, \widehat{f}^{-1} U \text{ is open in } A \right\} :$$

the idea is that if we demonstrate $\mathcal{T}$ is a topology for $\prod_{i\in I} X_i$ and $\mathcal{T}$ makes all the pr_i's continuous, then we can conclude the continuity of $\widehat{f}$. The first part is immediate, so let us focus on the remaining part. If we take an open subset V of X_j, the open subset $\mathrm{pr}_j^{-1} V$ of the product is in $\mathcal{T}$, because $f_j^{-1}V = \widehat{f}^{-1}\left(\mathrm{pr}_j^{-1} V\right)$ is open in A.

Example 2.3.10 (Coproduct of topological spaces)**.** As in the previous example, we move from Example 2.3.6. In Topology, it is maybe more customary to use

$$\coprod_{i\in I} X_i \text{ instead of } \sum_{i\in I} X_i$$

when $\{X_i \mid i \in I\}$ is a family of topological spaces. However, if we do not give a topology to $\coprod_{i\in I} X_i$, this object remains a bare set. In analogy to what happened with the Cartesian product, we prescribe the open subsets of $\coprod_{i\in I} X_i$ by making reference to the injections $\mathrm{in}_j : X_j \to \coprod_{i\in I} X_i$:

> we define a subset A of $\coprod_{i\in I} X_i$ to be open if and only if $\mathrm{in}_j^{-1} A$ is an open subset of X_j for every $j \in I$.

Let us recycle the universal property enjoyed by the family of the injections, that is for every family $\{g_i : X_i \to A \mid i \in I\}$ of continuous functions there exists one function $\widetilde{g} : \coprod_{i\in I} X_i \to A$ such that $g_j = \widetilde{g}\,\mathrm{in}_j$ for every $j \in I$. Furthermore, $\widetilde{g}$ is continuous: if $U \subseteq A$ is open, then so are the subsets $g_j^{-1}U \subseteq X_j$; consequently $g_j^{-1}U = \mathrm{in}_j^{-1}\left(\widetilde{g}^{-1}U\right)$ for every $j \in I$, which implies $\widetilde{g}^{-1}U$ is open.

Sometimes, products and coproducts can be isomorphic, as in the following example.

Example 2.3.11 (Product and coproduct of modules)**.** [Yet to be TEX-ed...]

Let us talk about *finite* products, that is products of a finite set of objects. The following arguments will be useful when we will deal with finite completeness of categories. Keep an eye on Figure 2.1.

Proposition 2.3.12 (Finite products, reduction from left)**.** Let $\mathcal{C}$ be a category a finite set $\{x_1, \dots, x_n\}$, with $n \geq 2$, of objects of $\mathcal{C}$. Let $x_1 \xleftarrow{l_2} p_2 \xrightarrow{r_2} x_2$ be one of the products of $\{x_1, x_2\}$ and let $p_i \xleftarrow{l_{i+1}} p_{i+1} \xrightarrow{r_{n+1}} x_{i+1}$ be one of the products of $\{p_i, x_{i+1}\}$. Then the morphisms

$$\begin{aligned} l_2 \cdots l_n &: p_n \to x_1 \\ r_j l_{j+1} \cdots l_n &: p_n \to x_j \quad \text{for } j \in \{2, \dots, n-1\} \\ r_n &: p_n \to x_n \end{aligned}$$

of $\mathcal{C}$ do form a product of $\{x_1, \dots, x_n\}$.

Proof. The proof is conducted by induction on $n \geq 2$. The case $n = 2$ is the base case of our recursive definition. To proceed with the inductive step, let reduction from left (Proposition 2.3.12)

$$\begin{tikzcd}
 & & & & & p_n \arrow[dl, "l_n"'] \arrow[dr, "r_n"] & \\
 & & & & p_{n-1} \arrow[dl, "l_{n-1}"'] \arrow[dr, "r_{n-1}"] & & x_n \\
 & & & \cdots \arrow[dl, "l_4"'] \arrow[dr, "\cdots"] & & x_{n-1} & \\
 & & p_3 \arrow[dl, "l_3"'] \arrow[dr, "r_3"] & & \cdots & & \\
 & p_2 \arrow[dl, "l_2"'] \arrow[dr, "r_2"] & & x_3 & & & \\
x_1 & & x_2 & & & &
\end{tikzcd}$$

reduction from right (Proposition 2.3.13)

$$\begin{tikzcd}
 & p_n \arrow[dl, "l_n"'] \arrow[dr, "r_n"] & & & & & \\
x_1 & & p_{n-1} \arrow[dl, "l_{n-1}"'] \arrow[dr, "r_{n-1}"] & & & & \\
 & x_2 & & \cdots \arrow[dl, "\cdots"'] \arrow[dr, "r_4"] & & & \\
 & & \cdots & & p_3 \arrow[dl, "l_3"'] \arrow[dr, "r_3"] & & \\
 & & & x_{n-2} & & p_2 \arrow[dl, "l_2"'] \arrow[dr, "r_2"] & \\
 & & & & x_{n-1} & & x_n
\end{tikzcd}$$

Figure 2.1. Finite products recursively constructed.

us picture the situation:

$$\begin{tikzcd}
& & p_{n+1} \arrow[dl, "l_{n+1}"'] \arrow[drr, "r_{n+1}"] & & \\
& p_n \arrow[dl, "l_2\cdots l_n"'] \arrow[d, "r_j l_{j+1}\cdots l_n"] \arrow[dr, "r_n"] & & & x_{n+1} \\
x_1 & x_j & x_n & & \\
& & & a \arrow[ulll, "f_1"] \arrow[ull, "f_j"] \arrow[ul, "f_n"] \arrow[uull, dotted, "h"] \arrow[uuul, dotted, "k"'] \arrow[uur, "f_{n+1}"'] &
\end{tikzcd}$$

where $j \in \{2, \dots, n-1\}$, a is an arbitrary object with morphisms $f_1, \dots, f_n, f_{n+1}$. By the universal property of product, we have

$$\begin{aligned}
f_1 &= l_1 \cdots l_n h \\
f_j &= r_j l_{j+1} \cdots l_n h \\
f_n &= r_n h
\end{aligned}$$

for one and only one $h : a \to p_n$. Again by the universal property of product.

$$\begin{aligned}
h &= l_{n+1} k \\
f_{n+1} &= r_{n+1} k
\end{aligned}$$

for a unique $k : a \to p_{n+1}$. Thus

$$\begin{aligned}
f_1 &= l_1 \cdots l_n l_{n+1} k \\
f_j &= r_j l_{j+1} \cdots l_n l_{n+1} k \\
f_n &= r_n l_{n+1} k \\
f_{n+1} &= r_{n+1} k
\end{aligned}$$

and we have concluded. □

Proposition 2.3.13 (Finite products, reduction from right)**.** Let $\mathcal{C}$ be a category a finite set $\{x_1, \dots, x_n\}$, with $n \geq 2$, of objects of $\mathcal{C}$. Let $x_{n-1} \xleftarrow{l_2} p_2 \xrightarrow{r_2} x_n$ be one of the products of $\{x_{n-1}, x_n\}$ and let $x_{n-i} \xleftarrow{l_{i+1}} p_{i+1} \xrightarrow{r_{n+1}} p_i$ be one of the products of $\{x_{n-i}, p_i\}$. Then the morphisms

$$\begin{aligned}
r_2 \cdots r_n &: p_n \to x_n \\
l_j r_{j+1} \cdots r_n &: p_n \to x_{n-j+1} \quad \text{for } j \in \{2, \dots, n-1\} \\
r_n &: p_n \to x_1
\end{aligned}$$

of $\mathcal{C}$ do form a product of $\{x_1, \dots, x_n\}$.

Proof. This is **exercise 2.3.14**. You should expect some work like in the proof of Proposition 2.3.12. □

Corollary 2.3.15 (Associativity of product)**.** In a category $\mathcal{C}$, let

$$x_1 \xleftarrow{p_1} x_1 \times x_2 \xrightarrow{p_2} x_2$$

a product of $\{x_1, x_2\}$,

$$x_1 \times x_2 \xleftarrow{p_{1,2}} (x_1 \times x_2) \times x_3 \xrightarrow{p_3} x_3$$

a product of $\{x_1 \times x_2, x_3\}$,

$$x_2 \xleftarrow{q_2} x_2 \times x_3 \xrightarrow{q_3} x_3$$

a product of $\{x_2, x_3\}$,

$$x_1 \xleftarrow{q_1} x_1 \times (x_2 \times x_3) \xrightarrow{q_{2,3}} x_2 \times x_3$$

a product of $\{x_1, x_2 \times x_3\}$.

$$\begin{tikzcd}
 & & (x_1 \times x_2) \times x_3 \arrow[dl, "p_{1,2}"'] \arrow[ddrr, "p_3"] & & \\
 & x_1 \times x_2 \arrow[dl, "p_1"'] \arrow[dr, "p_2"] & & & \\
x_1 & & x_2 & & x_3 \\
 & & & x_2 \times x_3 \arrow[ul, "q_2"] \arrow[ur, "q_3"'] & \\
 & & x_1 \times (x_2 \times x_3) \arrow[uull, "q_1"] \arrow[ur, "q_{2,3}"'] & &
\end{tikzcd}$$

Then

$$(x_1 \times x_2) \times x_3 \cong x_1 \times (x_2 \times x_3).$$

Proof. It follows from Proposition 2.3.12 and Proposition 2.3.13. □

Corollary 2.3.16. A category has all finite products if and only if has a terminal object and all binary products.

Proof. One implication is easy. For the opposite one: terminal objects are empty products; an object with identity is a product of itself; if you are given at least two objects, either of Proposition 2.3.12 and Proposition 2.3.13 tell you finite product are consecutive binary products. □

Exercise 2.3.17. In a category with terminal object 1, we have $a \times 1 \cong 1 \times a \cong a$.

2.4 Pullbacks and pushouts

Let $\mathcal{C}$ be a category. The limits of the functors

$$\left(\begin{tikzcd}[column sep=small] \bullet \arrow[dr] & & \bullet \arrow[dl] \\ & \bullet & \end{tikzcd} \right) \to \mathcal{C}$$

are called *pullbacks*. Dually, the colimits of the functors

$$\left(\begin{tikzcd}[column sep=small] & \bullet \arrow[dl] \arrow[dr] & \\ \bullet & & \bullet \end{tikzcd} \right) \to \mathcal{C}$$

are said *pushouts*. More explicitly:

Definition 2.4.1 (Pullbacks & pushouts, explicit)**.** Let $\mathcal{C}$ a category and a pair of morphisms with the same codomain

$$\begin{tikzcd} a \arrow[dr, "f"'] & & b \arrow[dl, "g"] \\ & c & \end{tikzcd} \tag{2.4.1}$$

in $\mathcal{C}$. A *pullback* of (2.4.1) is any pair of morphisms with a common domain

$$\begin{tikzcd} & p \arrow[dl, "h"'] \arrow[dr, "k"] & \\ a & & b \end{tikzcd}$$

in $\mathcal{C}$ such that:

- the square

$$\begin{tikzcd} & p \arrow[dl, "h"'] \arrow[dr, "k"] & \\ a \arrow[dr, "f"'] & & b \arrow[dl, "g"] \\ & c & \end{tikzcd}$$

  commutes
- for every

$$\begin{tikzcd} & p' \arrow[dl, "h'"'] \arrow[dr, "k'"] & \\ a & & b \end{tikzcd}$$

  in $\mathcal{C}$ making

$$\begin{tikzcd} & p \arrow[dl, "h'"'] \arrow[dr, "k'"] & \\ a \arrow[dr, "f"'] & & b \arrow[dl, "g"] \\ & c & \end{tikzcd}$$

  commute there exists one and only one $r : p' \to p$ in $\mathcal{C}$ such that

$$\begin{tikzcd} & p' \arrow[dl, "h'"'] \arrow[dr, "k'"] \arrow[d, dashed, "r"] & \\ a & p \arrow[l, "h"] \arrow[r, "k"'] & b \end{tikzcd}$$

  commutes.

Assuming now we have a pair of morphisms with the same domain

$$\begin{tikzcd} & c \arrow[dl, "f"'] \arrow[dr, "g"] & \\ a & & b \end{tikzcd} \tag{2.4.2}$$

in $\mathcal{C}$, a *pushout* of (2.4.2) is any pair of morphisms in $\mathcal{C}$ with a common codomain

$$\begin{tikzcd} a \arrow[dr, "h"'] & & b \arrow[dl, "k"] \\ & q & \end{tikzcd}$$

such that:

- the square

$$\begin{tikzcd} & c \arrow[dl, "f"'] \arrow[dr, "g"] & \\ a \arrow[dr, "h"'] & & b \arrow[dl, "k"] \\ & q & \end{tikzcd}$$

  commutes
- for every

$$\begin{tikzcd} a \arrow[dr, "h'"'] & & b \arrow[dl, "k'"] \\ & q' & \end{tikzcd}$$

  in $\mathcal{C}$ making

$$\begin{tikzcd} & c \arrow[dl, "f"'] \arrow[dr, "g"] & \\ a \arrow[dr, "h'"'] & & b \arrow[dl, "k'"] \\ & q & \end{tikzcd}$$

  commute there exists one and only one $s : q \to q'$ in $\mathcal{C}$ such that

$$\begin{tikzcd} a \arrow[r, "h"] \arrow[dr, "h'"'] & q \arrow[d, dashed, "s"] & b \arrow[l, "k"'] \arrow[dl, "k'"] \\ & q' & \end{tikzcd}$$

  commutes.

The definitions above become can be more concise though: pullbacks (pushouts) are products (coproducts) in certain categories.

Proposition 2.4.2. A pullback of

$$\begin{tikzcd} a \arrow[dr, "f"'] & & b \arrow[dl, "g"] \\ & c & \end{tikzcd}$$

in $\mathcal{C}$ is any of the products that pair of morphisms in $\mathcal{C}{\downarrow}c$. Dually, a pushout of

$$\begin{tikzcd} & c \arrow[dl, "f"'] \arrow[dr, "g"] & \\ a & & b \end{tikzcd}$$

in $\mathcal{C}$ is any of the coproducts such pair of morphisms in $c{\downarrow}\mathcal{C}$.

Proof. This is **exercise 2.4.3**. $\square$

Exercise 2.4.4. Let $\mathcal{C}$ be a category with initial object 0 and terminal object 1. What are pullbacks of a pair of morphisms with codomain 1? What are pushouts of a pair of morphisms with domain 0?

Example 2.4.5 (Pullbacks in **Set**)**.** Now we consider sets and functions as in

$$\begin{tikzcd} & A \arrow[d, "f"] \\ B \arrow[r, "g"'] & C \end{tikzcd}$$

with the aim to find a pullback of it. From the set

$$D := \{(a,b) \in A \times B \mid f(a) = g(b)\}$$

we can make the functions

$$p : D \to A\,,\ p(a,b) := a$$
$$q : D \to B\,,\ q(a,b) := b$$

Hence we can draw at least the commuting square

$$\begin{tikzcd} D \arrow[r, "p"] \arrow[d, "q"'] & A \arrow[d, "f"] \\ B \arrow[r, "g"'] & C \end{tikzcd}$$

Now consider

$$\begin{tikzcd} D' \arrow[drr, bend left, "p'"] \arrow[ddr, bend right, "q'"'] & & \\ & D \arrow[r, "p"] \arrow[d, "q"'] & A \arrow[d, "f"] \\ & B \arrow[r, "g"'] & C \end{tikzcd}$$

with $fp' = gq'$. This hypothesis implies that $(p'(x), q'(x)) \in D$ for every $x \in D'$, and allows us to introduce the function

$$r : D' \to D\,,\ r(x) := (p'(x), q'(x))$$

which is such that $pr = p'$ and $qr = q'$. Finally, r is the unique one to do so, which fact is immediate for how r is defined.

Let us remain in **Set**. Consider a function $f : A \to B$ and the diagram

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

where we have duplicated f. The example above tells us we have the pullback square

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

with $R_f := \{(a,b) \in A \times A \mid f(a) = f(b)\}$. This subset of $A \times A$ is a certain equivalence relation over A, namely the *kernel relation* of f. [Did we mention kernel relations in the intro?]

There is nothing special of **Set** that prevents us to generalize it to any category $\mathcal{C}$: we define the *kernel relation* of a $f : a \to b$ in $\mathcal{C}$ to be any of the pullbacks of

$$\begin{tikzcd} & a \arrow[d, "f"] \\ a \arrow[r, "f"'] & b \end{tikzcd}$$

As soon as we meet coequalizers, we will have the tool to express the quotient X/R_f in a categorial fashion, and thus to motivate the general concept of *quotient object.*

Example 2.4.6 (Pushouts in **Set**). Recall what we have done in Example 2.3.6, but change a bit the notation. Take a family of two sets A_1 and A_2: write $A_1 + A_1$ instead of using the $\sum$ or $\coprod$ notation, write `left` and `right` in place of in_1 and in_2, respectively.
To get started, let us consider sets and functions

$$\begin{tikzcd} & C \arrow[dl, "f"'] \arrow[dr, "g"] & \\ A & & B \end{tikzcd}$$

Let us draw a diagram

$$\begin{tikzcd} C \arrow[r, "f"] \arrow[d, "g"'] & A \arrow[d, "\texttt{left}"] \\ B \arrow[r, "\texttt{right}"'] & A+B \end{tikzcd} \tag{2.4.3}$$

By definition of $A+B$, it can be $\texttt{left}f(x) \neq \texttt{right}g(x)$ for some $x \in C$. However, we can make them 'equal' under an adequate equivalence relation $\sim$: the smallest in which, for $x \in C$, the elements $\texttt{left}f(x)$ and $\texttt{right}g(x)$ are identified; that is we define $\sim$ to be the smallest equivalence relation containing

$$R := \{(\texttt{left}f(x), \texttt{right}g(x)) \mid x \in C\}.$$

In this case, let us write p the projection $A+B \to \frac{A+B}{\sim}$. The new square is

$$\begin{tikzcd} C \arrow[r, "f"] \arrow[d, "g"'] & A \arrow[d, "h"] \\ B \arrow[r, "k"'] & \frac{A+B}{\sim} \end{tikzcd} \tag{2.4.4}$$

with $h := p\texttt{left}$ and $k := p\texttt{right}$ and it is commutative. Now, pick

$$\begin{tikzcd} C \arrow[r, "f"] \arrow[d, "g"'] & A \arrow[d, "h"] \arrow[ddr, bend left, "h'"] & \\ B \arrow[r, "k"'] \arrow[drr, bend right, "k'"'] & \frac{A+B}{\sim} & \\ & & X \end{tikzcd}$$

such that $h'f = k'g$. By the universal property of coproduct, we have the function

$$h' + k' : A + B \to X$$

such that $(h' + k')\,\texttt{left} = h'$ and $(h' + k')\,\texttt{right} = k'$. Taken, for any $x \in C$, one $(\texttt{left}f(x), \texttt{right}g(x)) \in R$, we have

$$(h' + g')(\texttt{right}f(x)) = h'f(x)$$
$$(h' + g')(\texttt{left}g(x)) = k'g(x)$$

which are equal for every $x \in C$, by assumption. [Talk about generated equivalence relations and what follows.] Thus the triangle

$$\begin{tikzcd} A+B \arrow[rr, "h'+k'"] \arrow[dr, "p"'] & & X \\ & \frac{A+B}{\sim} \arrow[ur, dashed] & \end{tikzcd}$$

commutes for exactly one dashed function. This function is the one we are looking for. [Complete...]

Exercise 2.4.7. In the previous example, what is $\frac{A+B}{\sim}$ if $C = A \cap B$ and f and g are just the inclusions of C in A and B respectively? [There is other material to put here...]

Exercise 2.4.8. Go back to Example 2.4.6. What if we started our discourse from

$$\begin{tikzcd} C \arrow[r, "f"] \arrow[d, "g"'] & A \arrow[d, hook] \\ B \arrow[r, hook] & A \cup B \end{tikzcd}$$

instead?

Exercise 2.4.9 (Gluing topological spaces)**.** If you are given two spaces X and A, a subspace $E \subseteq A$ and a continuous function $f : E \to X$, then $X \sqcup_f A$ denotes the disjoint union $X \sqcup A$ where every $x \in E$ is identified to $f(x)$, that is

$$X \sqcup_f A := \frac{X \sqcup A}{x \sim f(x) \text{ for } x \in E}.$$

Find a pushout square

$$\begin{tikzcd} E \arrow[r, hook] \arrow[d, "f"'] & A \arrow[d] \\ X \arrow[r] & X \sqcup_f A \end{tikzcd}$$

in **Top**. The exercise requires you to work about the topologies involved and about continuity.

Example 2.4.10 (CW complexes)**.** In Topology, several spaces often employed are homotopic — or even homeomorphic — to other spaces glued together. Although you can glue everything to everything, very simple spaces to attach are disks $\mathbb{D}^n := \{x \in \mathbb{R}^n \mid \|x\| \leq 1\}$ along their boundaries $\mathbb{S}^{n-1} := \partial\mathbb{D}^n$. (Pay attention to superscripts.) For any topological space X, we can perform the following recursive construction:

- Let X_0 be the space X but with the discrete topology.
- For $n \in \mathbb{N}$, from topological space X_n we prescribe the construction of another space X_{n+1}. If we are given a family $\{D_\alpha \mid \alpha \in \Lambda\}$ of copies of $\mathbb{D}^{n+1}$ and collection of continuous functions

  $$\{f_\alpha : \partial D_\alpha \to X_n | \alpha \in \Lambda\}$$

  then we can consider the following topological space

  $$X_{n+1} := \frac{X_n \sqcup \coprod_{\alpha\in\Lambda} D_\alpha}{x \sim f_\alpha(x) \text{ for } \alpha \in \Lambda \text{ and } x \in \partial D_\alpha}.$$

In other words, X_{n+1} is X_n with $(n+1)$-dimensional disks attached to it along their boundaries.

As always, we are striving to find some universal property worth of consideration. A square comes easily if you consider the inclusions running in parallel $\mathbb{S}^n \hookrightarrow \mathbb{D}^{n+1}$ and $X_n \hookrightarrow X_{n+1}$ together with $\mathbb{D}^{n+1} \hookrightarrow X_{n+1}$ of the construction above. The other pieces are the attaching maps: indeed we have a commuting square

$$\begin{tikzcd}
\coprod_{\alpha\in\Lambda} \partial D_\alpha \arrow[r, hook] \arrow[d, "\coprod_{\alpha\in\Lambda} f_\alpha"'] & \coprod_{\alpha\in\Lambda} D_\alpha \arrow[d, hook] \\
X_n \arrow[r, hook] & X_{n+1}
\end{tikzcd}$$

This square is a pushout one in **Top**, which is easy to prove.

Exercise 2.4.11 (Spheres are CW complexes)**.** Consider the case in which X_0 is a single point space and so are the spaces X_i for $1 \le i \le n-1$. Such situation can be achieved by attaching no disk for a while; afterwards, attach one disk $\mathbb{D}^n$ along $\mathbb{S}^{n-1}$ to X_{n-1}. Hence we have a homeomorphism $X_n \cong \mathbb{D}^n/\mathbb{S}^{n-1}$, but you will show something more, that is $X_n \cong \mathbb{S}^n$.
In your Topology course, you might have managed to show this as follows:

1. You have constructed a surjective continuous function $f : \mathbb{D}^n \to \mathbb{S}^n$ and considered the quotient space $\mathbb{D}^n/\sim_f$ where $\sim_f$ is the kernel relation [talk about kernel relations!] of f. This relation is not a random relation: for every $x, y \in \mathbb{D}^n$, we have $x \sim_f y$ if and only if $x = y$ or $x, y \in \mathbb{S}^{n-1}$. As consequence, $\mathbb{D}^n/\sim_f = \mathbb{D}^n/\mathbb{S}^{n-1}$.
2. Thanks to the universal property of quotients, the function $\phi : \mathbb{D}^n/\mathbb{S}^{n-1} \to \mathbb{S}^n$ such that $f = \phi p_n$, with the p_n the canonical projection, is continuous and bijective. Now, recalling that continuous functions from compact spaces to Hausdorff spaces are closed, conclude that indeed ϕ is a homeomorphism.

The aim of this exercise it that you can arrive to the same result in a different manner. If you can recollect your memories or retrieve your notes, see if you can recycle the f above and write a pushout square

$$\begin{tikzcd}
\mathbb{S}^{n-1} \arrow[r, hook] \arrow[d, "!"'] & \mathbb{D}^n \arrow[d, "f"] \\
X_{n-1} \arrow[r] & \mathbb{S}^n
\end{tikzcd} \tag{2.4.5}$$

If you do not know how $\mathbb{D}^n/\mathbb{S}^{n-1} \cong \mathbb{S}^n$, it does not matter since you will force yourself to search for a pushout square like (2.4.5).

Exercise 2.4.12 ($\mathbb{RP}^n$ is a CW complex)**.** In Topology, the n-th *real projective space* is

$$\mathbb{RP}^n := \frac{\mathbb{R}^{n+1} \setminus \{0\}}{x \sim \lambda x \text{ for } x \in \mathbb{R}^{n+1}, \lambda \in \mathbb{R}}$$

which is known to be homeomorphic to the sphere $\mathbb{S}^n$ which has the antipodal points identified:

$$\frac{\mathbb{S}^n}{x \sim -x \text{ for } x \in \mathbb{S}^n}.$$

This observation is the key for the coming arguments. In fact, $\mathbb{S}^n$ is the boundary of $\mathbb{D}^{n+1}$ and we already have a continuous function $p_n : \mathbb{S}^n \to \mathbb{RP}^n$

that attaches the the disc to the projective space along the boundary. Find a pushout square of the form

$$\begin{tikzcd}
\mathbb{S}^n \arrow[r, hook] \arrow[d, "p_n"'] & \mathbb{D}^{n+1} \arrow[d, "?"] \\
\mathbb{RP}^n \arrow[r, hook] & \mathbb{RP}^{n+1}
\end{tikzcd}.$$

Topology, again, but combined with Group Theory.

Example 2.4.13 (Seifert-van Kampen Theorem)**.** Suppose given a topological space X, two open subsets $A, B \subseteq X$ such that $A \cup B = X$ and one point x_0 of $A \cap B$. Let us denote by i_A, i_B, j_A and j_B the group morphisms induced by the inclusions $A \cap B \hookrightarrow A$, $A \cap B \hookrightarrow B$, $A \hookrightarrow X$ and $B \hookrightarrow X$, respectively. If A, B and $A \cap B$ are path-connected then,

$$\begin{tikzcd}
 & \pi_1(A, x_0) \arrow[dr, "j_A"] & \\
\pi_1(A \cap B, x_0) \arrow[ur, "i_A"] \arrow[dr, "i_B"'] & & \pi_1(X, x_0) \\
 & \pi_1(B, x_0) \arrow[ur, "j_B"'] &
\end{tikzcd}$$

is a pushout square of **Grp**.

[The Pullback Lemma is dropped here without a precise plan to embed it nicely with examples and further development. It's an issue that must be fixed.]

Proposition 2.4.14 (The Pullback Lemma)**.** In a category $\mathcal{C}$ consider a diagram

$$\begin{tikzcd}
\bullet \arrow[r, "a"] \arrow[d, "d"'] \arrow[dr, phantom, "Q_1"] & \bullet \arrow[r, "b"] \arrow[d, "g"] \arrow[dr, phantom, "Q_2"] & \bullet \arrow[d, "c"] \\
\bullet \arrow[r, "e"'] & \bullet \arrow[r, "f"'] & \bullet
\end{tikzcd}$$

where the perimetric rectangle commutes and the square on the right is a pullback one. Then that on the left is a pullback square is and only if so is the outer rectangle.

Proof. Let us assume Q_1 is a pullback square first. Consider any choice of h and k such that $ch = f(ek)$:

$$\begin{tikzcd}
\bullet \arrow[drrr, bend left, "h"] \arrow[ddr, bend right, "k"'] & & & \\
 & \bullet \arrow[r, "a"] \arrow[d, "d"'] \arrow[dr, phantom, "Q_1"] & \bullet \arrow[r, "b"] \arrow[d, "g"] \arrow[dr, phantom, "Q_2"] & \bullet \arrow[d, "c"] \\
 & \bullet \arrow[r, "e"'] & \bullet \arrow[r, "f"'] & \bullet
\end{tikzcd}$$

Being Q_2 a pullback square, there exists one and only one l such that $h = bl$ and $gl = ek$.

$$\begin{tikzcd}
\bullet \arrow[drrr, bend left, "h"] \arrow[ddr, bend right, "k"'] \arrow[drr, "l"] & & & \\
 & \bullet \arrow[r, "a"] \arrow[d, "d"'] \arrow[dr, phantom, "Q_1"] & \bullet \arrow[r, "b"] \arrow[d, "g"] \arrow[dr, phantom, "Q_2"] & \bullet \arrow[d, "c"] \\
 & \bullet \arrow[r, "e"'] & \bullet \arrow[r, "f"'] & \bullet
\end{tikzcd}$$

We have just said that this square in red commutes:

$$\begin{tikzcd}
\bullet \arrow[drrr, bend left, "h"] \arrow[drr, bend left, "l"] \arrow[ddr, bend right, "k"'] & & & \\
& \bullet \arrow[r, "a"'] \arrow[d, "d"] \arrow[dr, phantom, "Q_1"] & \bullet \arrow[r, "b"'] \arrow[d, "g"] \arrow[dr, phantom, "Q_2"] & \bullet \arrow[d, "c"] \\
& \bullet \arrow[r, "e"'] & \bullet \arrow[r, "f"'] & \bullet
\end{tikzcd}$$

Now, being Q_1 a pullback square, we have one m such that$l = am$ and $k = dm$:

$$\begin{tikzcd}
\bullet \arrow[drrr, bend left, "h"] \arrow[drr, bend left, "l"] \arrow[ddr, bend right, "k"'] \arrow[dr, "m"] & & & \\
& \bullet \arrow[r, "a"'] \arrow[d, "d"] \arrow[dr, phantom, "Q_1"] & \bullet \arrow[r, "b"'] \arrow[d, "g"] \arrow[dr, phantom, "Q_2"] & \bullet \arrow[d, "c"] \\
& \bullet \arrow[r, "e"'] & \bullet \arrow[r, "f"'] & \bullet
\end{tikzcd}$$

At this point, we have $bam = bl = h$ and $dm = k$. To conclude the first half of the theorem, you have to pick any m' making commute the triangles in green:

$$\begin{tikzcd}
\bullet \arrow[drrr, bend left, "h"] \arrow[drr, bend left, "am"] \arrow[ddr, bend right, "k"'] \arrow[dr, "m'"] & & & \\
& \bullet \arrow[r, "a"'] \arrow[d, "d"] \arrow[dr, phantom, "Q_1"] & \bullet \arrow[r, "b"'] \arrow[d, "g"] \arrow[dr, phantom, "Q_2"] & \bullet \arrow[d, "c"] \\
& \bullet \arrow[r, "e"'] & \bullet \arrow[r, "f"'] & \bullet
\end{tikzcd}$$

Being Q_2 a pullback square, we have $am' = am$. In conclusion, being Q_1 a pullback square too, we have $m = m'$. [Finer explanation here...] □

Exercise 2.4.15. Prove the remaining part of the theorem above.

Example 2.4.16 (Character functions)**.** Consider a subset A of some larger set X. You sure know a simple function called *character function* with just says if an element of X is a member of A:

$$\chi_A : X \to \{\texttt{true}, \texttt{false}\}\ ,\ \chi_A(x) := \begin{cases} \texttt{true} & \text{if } x \in A \\ \texttt{false} & \text{otherwise} \end{cases}$$

From now on, let us write Ω to mean $\{\texttt{true}, \texttt{false}\}$. As always, let us draw what we have:

$$\begin{tikzcd}
A \arrow[r, hook] & X \arrow[d, "\chi_A"] \\
& \Omega
\end{tikzcd}$$

with $A \hookrightarrow X$ being the usual inclusion. The composition of such functions is function constant to $\texttt{true}$. We know that constant functions are such because they can be factored through some function $A \to 1$ [write about this explicitly somewhere], which results in a commuting square

$$\begin{tikzcd}
A \arrow[r, hook] \arrow[d, "!"'] & X \arrow[d, "\chi_A"] \\
1 \arrow[r, "\lambda x.\texttt{true}"'] & \Omega
\end{tikzcd}$$

Well, this square is a pullback square. Of course, that is not all we have to say. [To be continued.]

2.5 Equalizers and coequalizers

For $\mathcal{C}$ category, the limits of functors

$$(\bullet \rightrightarrows \bullet) \to \mathcal{C}$$

are called *equalizers*. The colimits are called *coequalizers* instead.

Definition 2.5.1 (Equalizers & coequalizers, explicit)**.** Let $\mathcal{C}$ be a category and

$$a \underset{g}{\overset{f}{\rightrightarrows}} b \tag{2.5.6}$$

a pair of morphisms in $\mathcal{C}$. An *equalizer* of this pair is any morphism $i : c \to a$ such that:

- the diagram

$$c \xrightarrow{i} a \underset{g}{\overset{f}{\rightrightarrows}} b$$

  commutes
- for every $i' : c' \to a$ of $\mathcal{C}$ making

$$c' \xrightarrow{i'} a \underset{g}{\overset{f}{\rightrightarrows}} b$$

  commute, there is one and only one $k : c' \to c$ in $\mathcal{C}$ such that

$$\begin{tikzcd} c' \arrow[d, "k"'] \arrow[dr, "i'"] & \\ c \arrow[r, "i"'] & a \end{tikzcd}$$

  commutes.

Dually, a *coequalizer* of the pair (2.5.6) is any morphism $j : b \to d$ such that

- the diagram

$$a \underset{g}{\overset{f}{\rightrightarrows}} b \xrightarrow{j} d$$

  commutes;
- for every $j' : b \to d'$ of $\mathcal{C}$ making

$$a \underset{g}{\overset{f}{\rightrightarrows}} b \xrightarrow{j'} d'$$

  commute, there exists one and only one $h : d \to d'$ such that

$$\begin{tikzcd} & d \arrow[d, "h"] \\ b \arrow[ur, "j"] \arrow[r, "j'"'] & d' \end{tikzcd}$$

  commutes.

Before we analyse some example, the following lemma, may be quite useful to guide us.

Lemma 2.5.2. Equalizers are monomorphisms. Dually, coequalizers are epimorphisms.

Proof. Consider

$$\begin{tikzcd} e' \arrow[r, shift left, "s"] \arrow[r, shift right, "t"'] & e \arrow[r, "i"] & a \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & b \end{tikzcd}$$

with i equalizer of f and g and $is = it$. We can redraw this diagram as follows:

$$\begin{tikzcd} e \arrow[dr, "i"] & & \\ & a \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & b \\ e' \arrow[uu, bend left, "s"] \arrow[uu, bend right, "t"'] \arrow[ur, "is=it"'] & & \end{tikzcd}$$

In this case, we have $f(is) = f(it) = g(is) = g(it)$. Thus, by definition of equalizer, it must be $s = t$. $\square$

How could this be of aid? For example, in **Set** this means we have to look for inclusions in the domain of the domain of the parallel arrows. That is the case, indeed.

Example 2.5.3 (Equalizers in **Set**)**.** In **Set**, consider two functions

$$\begin{tikzcd} X \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & Y \end{tikzcd} .$$

The subset

$$E := \{x \in X \mid f(x) = g(x)\}$$

has the inclusion in X, we call it $i : E \hookrightarrow X$. Of course, we have $fi = gi$, so one part of the work is done. Now, let us take a commuting diagram

$$\begin{tikzcd} E' \arrow[r, "i'"] & X \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & Y \end{tikzcd} .$$

It follows that $i'(x) \in E$ for every $x \in E'$. Hence, we shall consider the function

$$h : E' \to E\,,\ h(x) := i'(x).$$

It is immediate now that $i : E \hookrightarrow X$ is an equalizer of f and g.

If we want one example of coequalizer in **Set**, we have to look for some surjection out of the codomain of the given parallel arrows.

Example 2.5.4 (Coequalizers in **Set**)**.** In **Set**, consider two functions

$$\begin{tikzcd} X \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & Y \end{tikzcd} .$$

Having a commuting diagram like

$$\begin{tikzcd} X \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & Y \arrow[r, "j"] & A \end{tikzcd}$$

means that we must look for some $j : Y \to A$ such that, for $x \in X$, the elements $f(x)$ and $g(x)$ are brought to the same element of A. The thing works if take A to be the quotient
$$\frac{Y}{f(x) \sim g(x) \text{ for } x \in X}$$
and define j as quotient map $Y \to Y/\sim$. It only remains to verify the universal property of coequalizer. Take any $j' : Y \to Z$ such that $j'f = j'g$.

$$\begin{tikzcd} & & Y/\sim \arrow[d, dashed] \\ X \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & Y \arrow[ur, "j"] \arrow[r, "j'"'] & Z \end{tikzcd}$$

The existence of the unique function $Y/\sim \to Z$ easily follows from Corollary 0.2.5.

Exercise 2.5.5. Retrieve Example 2.4.6 and prepare to combine it with the example we have just made. The square (2.4.3) doesn't even commute, but $A + B$ with the two injections is a coproduct, not a random thing out there. You can rearrange that diagram too

$$\begin{tikzcd} C \arrow[r, shift left, "\mathtt{left} f"] \arrow[r, shift right, "\mathtt{right} g"'] & A + B \end{tikzcd}$$

and summon the canonical projection $A + B \to \frac{A+B}{\sim}$ which is a coequalizer. At this point, we have the commutative square (2.4.4), which we proved to be a pushout square. Luckily, this works in general, and it is up to you to realize why and make the dual of the result too.
In a category $\mathcal{C}$, suppose you have a coproduct

$$a \xleftarrow{\mathtt{left}} a + b \xrightarrow{\mathtt{right}} b$$

a square

$$\begin{tikzcd} c \arrow[r, "f"] \arrow[d, "g"'] & a \arrow[d, "\mathtt{left}"] \\ b \arrow[r, "\mathtt{right}"'] & a + b \end{tikzcd}$$

and a coequalizer $p : a + b \to d$ of $\mathtt{left} f$ and $\mathtt{right} g$. Prove that

$$\begin{tikzcd} c \arrow[r, "f"] \arrow[d, "g"'] & a \arrow[d, "p\mathtt{left}"] \\ b \arrow[r, "p\mathtt{right}"'] & d \end{tikzcd}$$

is a pushout square in $\mathcal{C}$.

Exercise 2.5.6. In a category $\mathcal{C}$, consider two parallel morphisms

$$\begin{tikzcd} a \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & b \end{tikzcd}$$

and a product

$$\begin{tikzcd} a & a \times b \arrow[l, "p_a"'] \arrow[r, "p_b"] & b \end{tikzcd}$$

The universal property of products gives the morphisms $\overline{f}, \overline{g} : a \to a \times b$ such that $p_a\overline{f} = 1_a$, $p_b\overline{f} = f$, $p_a\overline{g} = 1_a$ and $p_b\overline{g} = g$:

$$\begin{tikzcd} & a \arrow[dl, "1_a"'] \arrow[d, shift right, "\overline{f}"'] \arrow[d, shift left, "\overline{g}"] \arrow[dr, "g"] \arrow[dr, shift right, "f"'] & \\ a & a \times b \arrow[l, "p_a"] \arrow[r, "p_b"'] & b \end{tikzcd}$$

Consider the pullback square

$$\begin{tikzcd} c \arrow[r, "m"] \arrow[d, "n"'] & a \arrow[d, "\overline{f}"] \\ a \arrow[r, "\overline{g}"'] & a \times b \end{tikzcd}$$

and prove that:

1. $m = n$.
2. m is equalizer of f and g.

Proposition 2.5.7. Categories with binary products/coproducts have equalizers/coequalizers if and only if have pullbacks/pushouts. That is: in a category with products/coproducts, pullbacks/pushouts can be obtained by equalizers/coequalizers and vice versa.

Ok, let us step back to Lemma 2.5.2, for there is something curious. Equalizers are monomorphisms, but what are the consequences if an equalizer is in addition epic?

Proposition 2.5.8. An epic equalizer is an isomorphism. Dually, a monic epimonomorphism is an isomorphism.

Proof. Assume $i : e \to a$ is an epic equalizer of

$$\begin{tikzcd} a \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & b \end{tikzcd}$$

From $fi = gi$, we can derive $f = g$, being i epic. Hence

$$\begin{tikzcd} a \arrow[r, "1_a"] & a \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & b \end{tikzcd}$$

commutes: by the universal property of equalizers, $1_a = ik$ for a unique $k : a \to e$. Moreover, by simple computation $iki = i = i1_e$, which implies that $ki = 1_e$ since i is monic. In conclusion, i is invertible. □

How could this become even interesting to us? If a category is such that every monomorphism is an equalizer of some pair of parallel arrows, then there monic and epic morphisms are ismorphisms, which we know it does not occur in every category.

Example 2.5.9. In **Set** that phenomenon does occur, let us look in it more closely. The problem we have can be stated as follows: take and injective function $f : A \to B$ and find two parallel functions parting from B to which f is an equalizer. We know, how to construct equalizers in **Set** and, even though that is not what we want, that example may guide our exploration. The problem we want to solve requires to find a certain set C and two certain functions $h, k : B \to C$. [Yet to be TEXed...]

2.6 (Co)Completeness

[Some of the parts here are to be rewritten...] [Better notation needed here...] Consider a functor $F : \mathbf{I} \to \mathcal{C}$. Let I be the *underlying discrete category* of $\mathbf{I}$ [say something about that elsewhere...] and $X : I \to \mathcal{C}$ the functor introduced as $X_\lambda := F_\lambda$ for $\lambda \in I$ [the *underlying discrete functor*...]. In other words, X is just F without no morphism $F_f : X_\alpha \to X_\beta$, where $f : \alpha \to \beta$ in $\mathbf{I}$, since I itself has not any morphism apart the identities. For the same reason, any cone $c : k_v \Rightarrow X$ is a just cone $k_x \Rightarrow F$ that cannot care about the morphisms F_f; precisely, $c : k_v \Rightarrow X$ is just a family $\{v \to X_\lambda \mid \lambda \in I\}$, while $k_v \Rightarrow F$ is the same family but also satisfying the naturality condition:

$$\begin{tikzcd} & X_\alpha \arrow[dd, "F_f"] \\ v \arrow[ur, "c_\alpha"] \arrow[dr, "c_\beta"'] & \\ & X_\beta \end{tikzcd}$$

commutes for every α, β and $f : \alpha \to \beta$ in $\mathbf{I}$. So you should expect that in general categories having products cannot guarantee the existence of limits.

Let us indicate by $\{p_\lambda : P \to X_\lambda \mid \lambda \in I\}$ one of the products of X: we have the morphisms

$$\begin{tikzcd} & X_\alpha \arrow[dr, "F_f"] & \\ P \arrow[ur, "p_\alpha"] \arrow[rr, "p_\beta"'] & & X_\beta \end{tikzcd}$$

that run in parallel for $\alpha, \beta \in I$ and $f : \alpha \to \beta$ in $\mathbf{I}$. Observe that for every $\beta \in I$ there is one morphism $P \to X_\beta$, namely p_β. On the other hand, for $\alpha \in I$ there may be one $f : \alpha \to \beta$ or more or even none of such; this means in general we do not have only one F_f present in the last diagram. To be safe, we will consider may copies of X_β as needed, so that there is one $p_\alpha F_f$ going towards its own copy of X_β. [Use a better notation here.] For that scope, consider the set

$$J := \bigcup_{\alpha,\beta \in I} \mathbf{I}(\alpha, \beta)$$

and the functor $\widetilde{X} : J \to \mathcal{C}$ with $\widetilde{X}_f$ defined to be X_β where β is the codomain of f. If $\{q_f : Q \to \widetilde{X}_f \mid f \in J\}$ is one of the products of $\widetilde{X}$, then we have one morphism $r : P \to Q$ such that $q_f r = F_f p_\alpha$ for every $\alpha \in I$ and morphism $f \in J$ with domain α, and one morphism $s : P \to Q$ such that $q_f s = p_\beta$ for every $\beta \in I$ and morphism $f \in J$ with codomain β.

We have thus constructed two parallel morphisms

$$\begin{tikzcd} P \arrow[r, shift left, "r"] \arrow[r, shift right, "s"'] & Q \end{tikzcd}$$

Assume there is an equalizer $i : L \to P$ of the pair r and s. We are going to show that:

the morphisms $p_\alpha i : L \to P_\alpha$ for $\alpha \in I$ do form a limit for F.

First of all, we verify that they form a natural transformation $k_L \Rightarrow X$. In fact,

$$F_f p_\alpha i = \underbrace{q_f r i = q_f s i}_{i \text{ equalizer of } r \text{ and } s} = p_\beta i$$

for every $\alpha, \beta \in I$ and $(f : a \to b) \in J$. We consider now any natural transformation $j : k_{L'} \Rightarrow F$ and show the existence and the uniqueness of a morphism $h : L' \to L$ such that

$$\begin{tikzcd} L' \arrow[d, "h"'] \arrow[dr, "j_\alpha"] & \\ L \arrow[r, "p_\alpha i"'] & X_\alpha \end{tikzcd}$$

commutes for every $\alpha \in I$. Forming the morphisms p_α for $\alpha \in I$ a product in $\mathcal{C}$, let $g : L' \to P$ in $\mathcal{C}$ be the morphism such that $p_\alpha g = j_\alpha$ for every $\alpha \in I$. We can arrange a picture like this:

$$\begin{tikzcd} L \arrow[r, "i"] & P \arrow[r, shift left, "r"] \arrow[r, shift right, "s"'] \arrow[d, "p_\alpha"] & Q \arrow[d, "q_f"] \\ L' \arrow[ur, "g"] \arrow[r, "j_\alpha"'] & X_\alpha \arrow[r, "F_f"'] & X_\beta \end{tikzcd}$$

Here, we have

$$q_f r g = F_f p_\alpha g = \underbrace{F_f j_\alpha = j_\beta}_{\substack{j : k_{L'} \Rightarrow F \text{is a} \\ \text{natural transformation}}} = p_\beta g = q_f s g.$$

Being the family of the morphisms $q_f : Q \to \widetilde{X}_f$ a product, we must have $st = rt$. And being $i : L \to P$ an equalizer of r and s, it must be $g = ih$ for a unique $h : L' \to L$. Hence, $j_\alpha = p_\alpha g = (p_\alpha i)h$, that is h works fine for our scope. To conclude, let $h' : L' \to L$ such that $(p_\alpha i)h = (p_\alpha i)h'$ for every $\alpha \in I$: by the universal property of products, $ih = ih'$; but, being equalizers monomorphisms, we can conclude $h = h'$.

Definition 2.6.1. A category $\mathcal{C}$ is said *(co)complete* whenever any functor $\mathbf{I} \to \mathcal{C}$ has a (co)limit. $\mathcal{C}$ is said *finitely (co)complete* when every functor $\mathbf{I} \to \mathcal{C}$ with $\mathbf{I}$ finite admits a (co)limit.

[No concerns about the size of **I**? It is important.] In general, it may be difficult to demonstrate that a certain category is complete. We have just proved a criterion that may be of aid:

Proposition 2.6.2 (Completeness Theorem)**.** Categories that have products and equalizers are complete.

A special place is for finite (co)limits.

Proposition 2.6.3 (Finite Completeness Theorem I)**.** Categories having terminal objects, binary products and equalizers are finitely complete. [Write a definition for 'finite completeness'.]

Proof. Use Corollary 2.3.16 and the argument to prove the Completeness Theorem. ☐

Actually, we have another finite completeness theorem, which requires some preliminary work.

Proposition 2.6.4 (Finite Completeness Theorem II)**.** Categories that have terminal objects and pullbacks are finitely complete.

Proof. Use the previous Lemma and the Finite Completeness Theorem I. □

Let us sum all up in one corollary:

Corollary 2.6.5 (Finite Completeness Theorem)**.** For any category, the following facts are equivalent:

1. it is finitely complete
2. it has a terminal, binary products and equalizers
3. it has a terminal object and pullbacks

2.7 Other exercises

Here is a collection of exercises that do have a precise placement in the the previous sections, for they deal with the interplay between different kinds of limits and with the previous chapters in a manner that would burden the flow of the discourse.

Exercise 2.7.1. Consider a category with binary products, and for $a, b \in |\mathcal{C}|$ write the corresponding products as

$$a \xleftarrow{p_{a,b}} a \times b \xrightarrow{q_{a,b}} b$$

1. Construct the functor $(c\times) : \mathcal{C} \to \mathcal{C}$. Without much other effort, you can define functors $(\times c) : \mathcal{C} \to \mathcal{C}$ as well.
2. Can you construct natural transformations $(a\times) \Rightarrow (b\times)$ and $(\times a) \Rightarrow (\times b)$?

Dually, if in a category $\mathcal{C}$ with binary coproducts you have coproducts

$$a \xrightarrow{l_{a,b}} a + b \xleftarrow{r_{a,b}} b$$

you can:

1. Construct functors $(c+)$ and $(+c)$ from $\mathcal{C}$ to $\mathcal{C}$.
2. Construct natural transformations $(a+) \Rightarrow (b+)$ and $(+a) \Rightarrow (+b)$.

Exercise 2.7.2 (Elements of objects)**.** [This part needs a heavy rewrite.] Let us take advantage of a basic fact about sets:

$$X \cong \mathbf{Set}(1, X) \text{ for every set } X.$$

In general, the isomorphism relation above is not made possible by a unique bijection, but there is one really meaningful for us: the function that takes $x \in X$ to the function $\widehat{x} : 1 \to X$ mapping the unique element of 1 into x. The great deal here is that functions $1 \to X$ inspect X and this isomorphism just outlined identifies every x to $\widehat{x}$.
Let us step back for a moment and turn our attention to the act of defining functions. To define a function $f : X \to Y$, one writes an expression like

$$f(x) := \Gamma \tag{2.7.7}$$

with Γ being a formula that may contain the symbol x or not. By writing something like (2.7.7), you are prescribing the images of each element of the domain. This deeply relies on these two facts:

1. Sets are things you can look inside.
2. We have a principle that guarantees the function we are defining in such manner is uniquely determined:

   Given two functions $f_1, f_2 : X \to Y$, if we have $f_1(a) = f_2(a)$ for every $a \in X$, then it must be $f_1 = f_2$.

   This is crucial, since once you have assigned a function as in (2.7.7), it cannot behave any different from what prescribed.

How this can be interesting to us at this point? First of all, 1 is a terminal object. We have showed earlier how elements of X can be thought as functions $1 \to X$. In this case, the application of f to x is the mere composition

$$fx : 1 \xrightarrow{x} X \xrightarrow{f} Y.$$

The principle aforementioned can be restated as:

If the diagram

$$\begin{tikzcd} 1 \arrow[r, "x"] & X \arrow[r, bend left, "f_1"] \arrow[r, bend right, "f_2"'] & Y \end{tikzcd}$$

commutes for every $x : 1 \to X$, then $f_1 = f_2$.

[To be continued...]

Exercise 2.7.3 (Natural number objects)**.** Assume you have a category $\mathcal{C}$ with terminal object 1 and with *natural number object*

$$1 \xrightarrow{0} \mathbb{N} \xrightarrow{s} \mathbb{N}$$

[I should spend more words for natural number objects.] It is not necessarily the $\mathbb{N}$ in **Set** you are used too. Prove the *primitive recursion theorem*:

If

$$\mathbb{N} \xleftarrow{p_{\mathbb{N}}} \mathbb{N} \times X \xrightarrow{p_X} X$$

is a product of $\mathcal{C}$, then for every $x_0 : 1 \to X$ and $f : \mathbb{N} \times X \to X$ in $\mathcal{C}$ there is one and only one $x : \mathbb{N} \to X$ such that

$$\begin{tikzcd} & \mathbb{N} \arrow[r, "s"] \arrow[d, "\langle 1_{\mathbb{N}}, x \rangle"] & \mathbb{N} \arrow[d, "x"] \\ 1 \arrow[ur, "0"] \arrow[r, "\langle 0, x_0 \rangle"'] & \mathbb{N} \times X \arrow[r, "f"'] & X \end{tikzcd}$$

[We just used some notation we have never used before here.]

Indeed, in a category with a terminal object and binary products, *recursion theorem* is equivalent to *primitive recursion theorem*. [Prepare a hint.]

Exercise 2.7.4 (Epic-monic factorization)**.** Consider in category $\mathcal{C}$ one pullback square

$$\begin{tikzcd} r \arrow[r, "s"] \arrow[d, "t"'] & a \arrow[d, "f"] \\ a \arrow[r, "f"'] & b \end{tikzcd}$$

and let $p : a \to a/r$ be a coequalizer of s and t.

1. Thanks to the universal property of coequalizers, there is exactly one $\overline{f} : a/r \to b$ in $\mathcal{C}$ that satisfies $f = \overline{f}p$. Show that $\overline{f}$ is monic.
2. The factorization $\overline{f}p$ is an *epic-monic factorization* of f. Consider now $q : a \to c$ epic and $g : c \to b$ monic in $\mathcal{C}$ such that $f = gq$, that is another epic-monic factorization of f. Show that there exists one and only one $h : a/r \to c$ such that

$$\begin{tikzcd} & a/r \arrow[dd, "h"] \arrow[dr, "\overline{f}"] & \\ a \arrow[ur, "p"] \arrow[dr, "q"'] & & b \\ & c \arrow[ur, "g"'] & \end{tikzcd}$$

   commutes. Show that moreover h is an isomorphism.
3. Explore by yourself: f is monic if and only if ....
4. Apply all that above to some concrete example. Does this sound familiar now?

Consider in category $\mathcal{C}$ one pushout square

$$\begin{tikzcd} a \arrow[r, "f"] \arrow[d, "f"'] & b \arrow[d, "u"] \\ b \arrow[r, "v"'] & z \end{tikzcd}$$

and let $\operatorname{im} f : f(a) \to b$ be an equalizer of u and v. Make the dual of above. (Spoiler: again about epic-monic factorizations.)

Exercise 2.7.5 (Equivalence relations)**.** [Yet to be TEXed...]

Exercise 2.7.6 (Subobject classifiers and some properties)**.** [Yet to be TEXed...]

3

Adjointness

3.1 Definition

Example 3.1.1 (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

$$\begin{aligned} \mathbf{Vect}_k(V, W) &\to \mathbf{Set}(S, W) \\ f &\to fi \end{aligned} \tag{3.1.1}$$

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 (3.1.1), 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

$$\mathbf{Vect}_k(G(S), W) \to \mathbf{Vect}_k(G(S'), W')$$

$$g \to \left(fgG(\phi) : \begin{array}{ccc} G(S) & \xrightarrow{g} & W \\ {\scriptstyle G(\phi)}\uparrow & & \downarrow{\scriptstyle f} \\ G(S') & & W' \end{array} \right)$$

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

$$\mathbf{Set}(S, F(W)) \to \mathbf{Set}(S', F(W'))$$

$$\delta \to \left(F(f)\delta\phi : \begin{array}{ccc} S & \xrightarrow{\delta} & F(W) \\ {\scriptstyle \phi}\uparrow & & \downarrow{\scriptstyle F(f)} \\ S' & & F(W') \end{array} \right)$$

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

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

and we have concluded.

Exercise 3.1.2. 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 3.1.3 (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

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

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

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

to functions

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

Exercise 3.1.4. 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

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

  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 3.1.5 (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 3.1.6 (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:

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

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

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

$$\begin{tikzcd} \mathbf{Set} \arrow[r, shift left, "I"] & \mathbf{Par} \arrow[l, shift left, "\mathtt{Maybe}"] \end{tikzcd}.$$

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.

3.2 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 3.2.1. 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 $\eta : 1_{\mathcal{C}} \Rightarrow RL$ 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. $\square$

Proposition 3.2.2. 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 $x \in |\mathcal{C}|$ 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,

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

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

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

[Co-units version.]

Proposition 3.2.3. 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 $\theta : LR \Rightarrow 1_{\mathcal{D}}$ 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

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

form an adjunction $L \dashv R$.

Proposition 3.2.4. 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 3.2.5. Prove the last two theorems.

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

3.3 Triangle Identities

[Yet to be TEXed...]

3.4 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 $i \in |\mathcal{I}|$

$$\begin{tikzcd} F(i) \arrow[r, "\xi_i"] & G(i) \\ \lim F \arrow[u, "\lambda_i"] & \lim G \arrow[u, "\mu_i"'] \end{tikzcd} \tag{3.4.2}$$

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 (3.4.2), 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 3.4.1. 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...] □

3.5 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 3.5.1 (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 3.5.2. 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 3.5.3. $(\times c) \dashv \square^c$ 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 3.5.4. 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?]

4

Introduction to Topoi

4.1 Subobject classifiers

Throughout the current section, we assume $\mathcal{E}$ is a category with initial object 1. That being the setting, we can give the following definition.

Definition 4.1.1. A *subobject classifier* for $\mathcal{E}$ is any morphism $t : 1 \to \Omega$ such that: for every monomorphism $f : a \to b$ of $\mathcal{E}$ there is one and only one morphism $\chi_f : b \to \Omega$ in $\mathcal{E}$ for which there is a pullback square

$$\begin{tikzcd} a \arrow[r, "f"] \arrow[d, "!"'] & b \arrow[d, "\chi_f"] \\ 1 \arrow[r, "t"'] & \Omega \end{tikzcd} \tag{4.1.1}$$

That is we can assign to every monomorphism $f : a \to b$ the morphism $\chi_f : b \to \Omega$ satisfying the property of the definition. Let us introduce then some symbolism: for $b \in |\mathcal{E}|$ we write $\mathrm{Sub}_{\mathcal{E}}\, b$ for the class of all the monomorphisms of $\mathcal{E}$ with codomain b. Hence we can introduce the function

$$\chi : \mathrm{Sub}_{\mathcal{E}}\, b \to \mathcal{E}(b, \Omega)$$

with χ_f defined to be that morphism $b \to \Omega$ for which there is a pullback square as the diagram (4.1.1).

It is worth to observe $\mathrm{Sub}_{\mathcal{E}}(b)$ has a natural structure of preorder: for

$$\begin{tikzcd} a_1 \arrow[dr, "f_1"'] & & a_2 \arrow[dl, "f_2"] \\ & b & \end{tikzcd}$$

monomorphisms of $\mathcal{E}$, write $f_1 \leq f_2$ to say there is some $h : a_1 \to a_2$ in $\mathcal{E}$ for which

$$\begin{tikzcd} a_1 \arrow[rr, "h"] \arrow[dr, "f_1"'] & & a_2 \arrow[dl, "f_2"] \\ & b & \end{tikzcd}$$

commutes. Note that, being here f_1 and f_2 monomorphisms, there is at most one h as such and it is a monomorphism as well.

We show now the relation $\simeq$ on $\mathrm{Sub}_{\mathcal{E}}\, b$ defined by

$$f_1 \simeq f_2 \text{ if and only if } f_1 \leq f_2 \text{ and } f_2 \leq f_1$$

for $f_1, f_2 \in \mathrm{Sub}_{\mathcal{E}}\, b$ is an equivalence relation. [...]

Yes, $\mathrm{Sub}_{\mathcal{E}}\, b$ is the full subcategory of $\mathcal{E} \downarrow b$ whose objects are all the monomorphisms of $\mathcal{E}$ with codomain b, and whose isomorphism relation is $\simeq$.

Proposition 4.1.2. Let

$$\begin{tikzcd} a_1 \arrow[dr, "f_1"'] & & a_2 \arrow[dl, "f_2"] \\ & b & \end{tikzcd}$$

be monomorphisms. $\chi_{f_1} = \chi_{f_2}$ if and only if $f_1 \simeq f_2$.

Proof. Assume $\chi_{f_1} = \chi_{f_2}$. By definition of subobject classifiers, χ_{f_1} is the morphism for which

$$\begin{tikzcd} a_1 \arrow[r, "f_1"] \arrow[d, "!"'] & b \arrow[d, "\chi_{f_1}"] \\ 1 \arrow[r, "t"'] & \Omega \end{tikzcd} \qquad \begin{tikzcd} a_2 \arrow[r, "f_2"] \arrow[d, "!"'] & b \arrow[d, "\chi_{f_1}"] \\ 1 \arrow[r, "t"'] & \Omega \end{tikzcd}$$

are pullback squares. Consequently, we must infer that there is one isomorphism $h : a_1 \to a_2$ such that $f_1 = f_2 h$. Hence $f_1 \le f_2$, and $f_2 \le f_1$ too, because $f_1 h^{-1} = f_2$.

For the remaining part of the proof, let us write $!_1$ the unique morphism $a_1 \to 1$ and $!_2$ the unique morphism $a_2 \to 1$. Also remember that triangles

$$\begin{tikzcd} a_1 \arrow[rr] \arrow[dr, "!_1"'] & & a_2 \arrow[dl, "!_2"] \\ & 1 & \end{tikzcd} \qquad \begin{tikzcd} a_1 \arrow[dr, "!_1"'] & & a_2 \arrow[ll] \arrow[dl, "!_2"] \\ & 1 & \end{tikzcd}$$

always commute.

Now we suppose $f_1 \simeq f_2$. The plan for the proof is: if we show that

$$\begin{tikzcd} a_1 \arrow[r, "f_1"] \arrow[d, "!_1"'] & b \arrow[d, "\chi_{f_2}"] \\ 1 \arrow[r, "t"'] & \Omega \end{tikzcd}$$

is a pullback square, then, being $\chi_{f_1} : b \to \Omega$ the one for which there is a pullback square like this, we can conclude $\chi_{f_1} = \chi_{f_2}$. First of all such square commutes: if we call h the morphism $a_1 \to a_2$ such that $f_1 = f_2 h$, then

$$\chi_{f_2} f_1 = \chi_{f_2} f_2 h = t!_2 h = t!_1.$$

Consider

$$\begin{tikzcd} \bullet \arrow[drr, bend left, "u"] \arrow[ddr, bend right, "v"'] & & \\ & a_1 \arrow[r, "f_1"] \arrow[d, "!_1"'] & b \arrow[d, "\chi_{f_2}"] \\ & 1 \arrow[r, "t"'] & \Omega \end{tikzcd}$$

where $\chi_{f_2} u = tv$. Being

$$\begin{tikzcd} a_2 \arrow[r, "f_2"] \arrow[d, "!_2"'] & b \arrow[d, "\chi_{f_2}"] \\ 1 \arrow[r, "t"'] & \Omega \end{tikzcd}$$

a pullback square we have one $z : \bullet \to a_2$ for which $f_2 z = u$ and $!_2 z = v$. From the assumption $f_1 \simeq f_2$, we have $f_2 \leq f_1$, that is $f_2 = f_1 q$ for some $q : a_2 \to a_1$. Then $u = f_1 qz$ and $v =!_1 qz$. Let us see if qz is what we are looking for.

$$\begin{tikzcd}
\bullet \arrow[dr, shift left, "qz"] \arrow[dr, shift right, "r"'] \arrow[drr, bend left, "u"] \arrow[ddr, bend right, "v"'] & & \\
& a_1 \arrow[r, "f_1"] \arrow[d, "!_1"'] & b \arrow[d, "\chi_{f_2}"] \\
& 1 \arrow[r, "t"'] & \Omega
\end{tikzcd}$$

where we suppose $!_1 r = v$ and $f_1 r = u$. Being f_1 a monomorphism, the sole second identity is enough to conclude $r = qz$. $\square$

Bibliography

[AHS06] J. Adámek, H. Herrlich and G.E. Strecker. *Abstract and Concrete Categories: The Joy of Cats*. Reprints in Theory and Applications of Categories, 2006. URL: http://www.tac.mta.ca/tac/reprints/articles/17/tr17.pdf.

[Bra17] Tai-Danae Bradley. *One-Line Proof: Fundamental Group of the Circle*. 2017. URL: https://www.math3ma.com/blog/one-line-proof-fundamental-group-of-the-circle.

[Gol06] R. Goldblatt. *Topoi: The Categorial Analysis of Logic*. Dover Books on Mathematics. Dover Publications, 2006.

[Law05] F.W. Lawvere. *An elementary theory of the category of sets (long version) with commentary*. Reprints in Theory and Applications of Categories, 2005. URL: http://www.tac.mta.ca/tac/reprints/articles/11/tr11.pdf.

[Lei12] T. Leinster. 'Rethinking set theory'. In: (2012). DOI: 10.48550/ARXIV.1212.6543. URL: https://arxiv.org/abs/1212.6543.

[Lei16] T. Leinster. *Basic Category Theory*. arXiv, 2016. DOI: 10.48550/ARXIV.1612.09375. URL: https://arxiv.org/abs/1612.09375.

[LR03] F.W. Lawvere and R. Rosebrugh. *Sets for Mathematics*. Cambridge University Press, 2003.

[Rie17] E. Riehl. *Category Theory in Context*. Aurora: Dover Modern Math Originals. Dover Publications, 2017. URL: https://math.jhu.edu/~eriehl/context/.

[Str11] J. Strom. *Modern Classical Homotopy Theory*. Graduate studies in mathematics. American Mathematical Society, 2011.

[Uni13] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study, 2013. URL: https://homotopytypetheory.org/book.