Category Theory

Aryaman Maithani
https://aryamanmaithani.github.io/

2020-03-29 14:53:53+05:30

CONTENTS

§-1 INTRODUCTION

This is essentially a way to force myself to be productive and self-study Category Theory. I try to write these notes in a way that I'm teaching them to someone to help me understand it better.
That being said, I don't really think that these notes would be better than reading the book which I'm reading itself. The book being -

Category Theory by *Steve Awodey*.

I do skip some of the more advanced examples but in places I do add extra explanations as I feel necessary. So yeah, have fun.
A personal suggestion: whenever you see a proposition or lemma or anything that is followed by a proof, try doing it yourself. When you struggle with it, that's when the definitions *really* seep in. Due to this, you would find many proofs that are not exactly the same as the ones given in the book.

These notes will keep getting updated as I read more, possibly over the next two years. I'm also not proof-reading these, so there are bound to be many errors. If you find any, *please* let me know. You shall then get to feature in the last section - Acknowledgments, if you wish. Apart from typos, you may also give suggestions if you think that something was ambiguously phrased. This will help me in wording things better for more clarity.

That's how these notes could *really* end up adding something of value.

Lastly, if you're on a desktop/laptop and want to view this in your browser instead (so that you can select text or click hyperlinks), you may use the following link -

bit.ly/raw-cat

§0 Some Structures

§§0.1 Monoids

A *monoid* is a set M equipped with a binary operation $*$ and a distinguished element $u \in M$ satisfying:

- $a * (b * c) = (a * b) * c$, for all $a, b, c \in M$, and
- $a * u = a = u * a$ for all $a \in M$.

We may sometimes also refer to the monoid as $(M, *, u)$.
A monoid homomorphism $h : (M, *, u_M) \to (N, \cdot, u_N)$ is a function $h : M \to N$ satisfying

- $h(a * b) = h(a) \cdot h(b)$, for all $a, b \in M$, and
- $h(u_M) = h(u_N)$.

§§0.2 Groups

With the same notation as earlier, a monoid M is said to be *group* if for every $a \in M$, there exists some $b \in M$ such that $a * b = u = b * a$.

A group homomorphism is a monoid homomorphism between groups.

§§0.3 Preorders

A *preorder* is a set P equipped with a binary relation $\leq$ satisfying:

- $a \leq a$ for all $a \in P$, and
- $a \leq b$ and $b \leq c$ implies $a \leq c$ for all $a, b, c \in P$.

That is, the relation is reflexive and transitive.

§§0.4 Posets

A *poset* is a preorder P with the additional condition that $\leq$ is antisymmetric, that is,

$$a \leq b \text{ and } b \leq a \implies a = b \text{ for all } a, b \in P.$$

An order-preserving map between posets $(P, \leq_P)$ and $(Q, \leq_Q)$ is a function $f : P \to Q$ satisfying

$$p \leq_P q \implies f(p) \leq_Q f(q).$$

§§0.5 Boolean algebra

A *Boolean algebra* is a poset B equipped with distinguished elements $0, 1$, binary operations $a \vee b$ of "join" and $a \wedge b$ of "meet," and a unary operation $\neg b$ of "complementation." These are required to satisfy the conditions

$$\begin{aligned} 0 &\leq a \\ a &\leq 1 \\ a \leq c \text{ and } b \leq c \quad &\text{iff} \quad a \vee b \leq c \\ c \leq a \text{ and } c \leq b \quad &\text{iff} \quad c \leq a \wedge b \\ a \leq \neg b \quad &\text{iff} \quad a \wedge b = 0 \\ \neg\neg a &= a. \end{aligned}$$

A typical example of a Boolean algebra is the power-set $\mathcal{P}(X)$ of X. We have the following identifications:

$$0 = \varnothing, 1 = X, A \vee B = A \cup B, A \wedge B = A \cap B, \neg A = X \setminus A.$$

A Boolean homomorphism is a function $h : B \to B'$ between Boolean algebras that is an order-preserving map which preserves the addition structure, id est,

$$\begin{aligned} h(0) &= 0, \\ h(1) &= 1, \\ h(a \vee b) &= h(a) \vee h(b), \\ h(a \wedge b) &= h(a) \wedge h(b), \text{ and} \\ h(\neg a) &= \neg h(a). \end{aligned}$$

§1 CATEGORIES

§§1.1 Definition - Category

We avoid any technicalities of set theory and define what a category is.

Definition 1.1. A *category* consists of the following data:

- Objects: $A, B, C, \ldots$
- Arrows (or *morphisms*): $f, g, h, \ldots$
- For each arrow f, there are given objects

$$\mathrm{dom}(f), \mathrm{cod}(f)$$

called the *domain* and *codomain* of f. We write

$$f : A \to B$$

to indicate that $\mathrm{cod}(f) = A$ and $\mathrm{dom}(f) = B$.

- Given arrows $f : A \to B$ and $g : B \to C$, there is given an arrow

$$g \circ f : A \to C$$

called the *composite* of f and g.

- For each object A, there is given an arrow

$$1_A : A \to A$$

called the *identity arrow* of A.

Additionally, we require the above data to follow the following laws:

- Associativity:

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

for all $f : A \to B, g : B \to C, h : C \to D$.

- Unit:

$$f \circ 1_A = f = 1_B \circ f$$

for all $f : A \to B$.

The above highly imitates the behaviour of that of sets and functions. Indeed, that shall be our first example of a category. Many of our other examples will also consist of categories where the objects are "structured sets" and arrows "structure-preserving functions." However, these are not all. A category is *anything* that satisfies the above

definition.
Note that technically, it isn't correct to write $C \in \mathbf{C}$ or $(f : C \to D) \in \mathbf{C}$; however, we shall abuse notation and write so as the alternative is too annoying.

§§1.2 Examples

1. The category Set. The objects of this category are (all) sets and the arrows are (all) functions between sets. The composite of two arrows is defined as composition of functions. It can be verified that this is indeed a well defined composition. Lastly, given any object X (a set), the identity arrow of X is simply the identity function $1_X = \mathrm{id}_X : X \to X$.
   The above is all the data that's required to define a category. Now, it is be verified that Set is indeed a category with the above data. These are basic results from set theory and we leave this to the reader.

2. We may also consider the category $\mathsf{Set}_{\mathsf{fin}}$, the category consisting of finite sets and functions between them.
   In the same spirit, we may take other restrictions as well. For example, we may take only injective functions instead of functions. By noting that that the

composition of two injections is again an injection and that the identity map is an injection, we get that this is indeed a category.
Similarly, one may restrict the arrows to surjections.

3. As mentioned before, many categories arise from other mathematical structures. In these categories, the objects are "structured sets" and arrows "structure-preserving functions." The following table lists these.

Category	Objects	Arrows (Morphisms)
Top	Topological spaces	Continuous maps
Mon	Monoids	Monoid homomorphisms
Grp	Groups	Group homomorphisms
Ring	Rings	Ring homomorphisms
Field	Fields	Field extensions
$\mathsf{Vec}_{\Bbbk}$	Vector spaces over $\Bbbk$	Linear maps
Pos	Posets	Order-preserving maps
BA	Boolean algebra	Boolean homomorphisms

4. Preorders. A preorder is a set P equipped with a binary relation $\leq$ satisfying:

- $a \leq a$ for all $a \in P$, and
- $a \leq b$ and $b \leq c$ implies $a \leq c$ for all $a, b, c \in P$.

That is, the relation is reflexive and transitive.
Any preorder P can be recognised as a category with objects being the elements of P and a unique arrow

$$a \to b \text{ iff } a \leq b. \tag{1.1}$$

The compositions of arrows is defined in the way that will be forced from construction. One can verify that this is indeed a category.
Note very carefully these this is quite different from the earlier examples where the objects were "sets" and arrows were functions.
Conversely, given a category where there is at most one arrow between any two objects, one may define a preorder using (1.1).

5. A poset is a preorder P with the additional condition that $\leq$ is antisymmetric, that is,

$$a \leq b \text{ and } b \leq a \implies a = b \text{ for all } a, b \in P.$$

With the same construction as before, we see that a poset is also a category.

6. Finite categories. These are categories with finitely many arrows (and thus, finitely many objects as well). As the previous example shows posets to be categories, we can already get many examples. Let's look at some of them in particular.

   - The category $\mathbf{1}$ looks like this:

     $$*$$

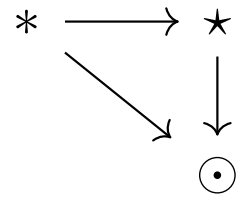
     It has one object and its identity arrow, which we do not draw.

   - The category $\mathbf{2}$ looks like this:

     $$* \longrightarrow \star$$

     It has two objects, their identity arrows, and exactly one arrow between them. Once again, the law of composition is clear.

   - The category $\mathbf{3}$ looks like this:

This is the category recognised by the poset $\{1, 2, 3\}$ with the typical ordering.

- The category $\mathbf{0}$ looks like this:

It has no objects and no arrows.
It is easy to specify finite categories - just take some objects and start putting arrows between them, but makes sure to put in the necessary identities and composites, as required by the axioms for a category. Also, if there are any any loops, then the need to be cut off by equations to keep the category finite. For example, consider the following specification:

(1.2)
$$A \underset{f}{\overset{e}{\rightleftarrows}} B$$

Unless we stipulate an equation like $ef = 1_A$, we will end up with infinitely many arrows $ef, efef, efefefef, \ldots$ This is still a category, just not a *finite* one. We will see this situation later again when we discuss free categories.

7. Let X be a set. We can regard X as a category $\mathbf{Dis}(X)$ by taking the objects to be the elements of X and taking the arrows to just be the required identity arrows, one for each $x \in X$. Such categories, in which the arrows are only identities, are called *discrete*. These categories are just special posets.

Before returning to more examples, we now define a “structure preserving” “function” between categories.

§§1.3 Definition - Functors

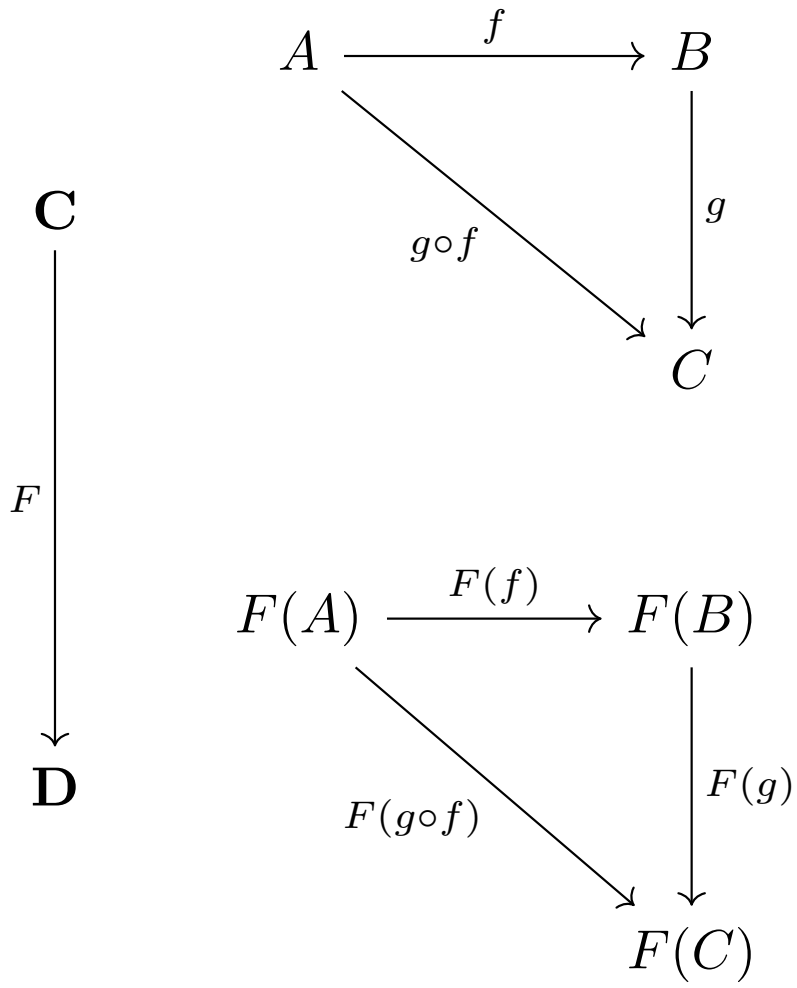
Definition 1.2. A functor

$$F : \mathbf{C} \to \mathbf{D}$$

between categories $\mathbf{C}$ and $\mathbf{D}$ is a mapping of objects (of $\mathbf{C}$) to objects (of $\mathbf{D}$) and arrows (of $\mathbf{C}$) to arrows (of $\mathbf{D}$), in such a way that

1. $F(f : A \to B) = F(f) : F(A) \to F(B)$,
2. $F(1_A) = 1_{F(A)}$, and
3. $F(g \circ f) = F(g) \circ F(f)$.

Thus, an arrow between objects gets mapped to an arrow between the images of the two objects. That is to say, a functor preserves domain and codomain. Similarly, it preserved identities and compositions.
This can be illustrated via the following picture:



Functors compose in the expected way. Moreover, this composition is associative. Also, every category $\mathbf{C}$ has an identity functor $1_{\mathbf{C}} : \mathbf{C} \to \mathbf{C}$.

This gives us another category Cat, the category with categories as objects and func-

tors as arrows.
Let us now turn back to examples.

§§1.4 Some more examples

1. We had previously seen that a poset P is in fact a category. The following question is natural to ask: Given posets P and Q, what is a functor between them?
   As it turns out and one can easily see, a functor $F : P \to Q$ is precisely an order preserving function from P to Q.
   The reader may compare this with the category Pos.

2. One may also consider monoids to be categories in the following way:
   A monoid is just a category with one element. The arrows of the category are the elements of the monoid. In particular, the identity arrow is the unit element. Composition of arrows is determined by the "product" (binary operation) $m \cdot n$ of the monoid.

For any set X, the set of functions from X to X, written as

$$\mathrm{Hom}_{\mathsf{Set}}(X, X)$$

is a monoid under the operation of composition. More generally, for any object C in any category $\mathbf{C}$, the set of arrows from C to C, written as $\mathrm{Hom}_{\mathbf{C}}(C, C)$, is a monoid under the composition operation of $\mathbf{C}$. The unit is the identity 1_C.

Once again, we have a correspondence between the arrows in Mon and functors between monoids regarded as categories. The correspondence is that they are the same!

3. *Forgetful functors.*
   Consider the functor $U : \mathsf{Grp} \to \mathsf{Set}$ which maps a group to its underlying set and a group homomorphism to the corresponding function between sets. This is clearly a functor. This is called the "forgetful functor" as it "forgets" the structure of the group. Similarly, one has forgetful functors from various different categories where the objects are structured sets.

§§1.5 Isomorphisms

Definition 1.3. In any category $\mathbf{C}$, an arrow $f : A \to B$ is called an *isomorphism*, if there is an arrow $g : b \to A$, called the *inverse* of f such that

$$g \circ f = 1_A \text{ and } f \circ g = 1_B.$$

Lemma 1.4. Inverses are unique.

Proof. Let $f : A \to B$ be an arrow and let $g_1, g_2 : B \to A$ be inverses of f. Observe that

$$g_1 = g_1 \circ 1_B = g_1 \circ (f \circ g_2) = (g_1 \circ f) \circ g_2 = 1_A \circ g_2 = g_2.$$

□

Since inverses are unique, we write the inverse of f as f^{-1}. (This is, of course, in the case that f is indeed an isomorphism.)

Definition 1.5. For objects A, B of $\mathbf{C}$, we say that A and B are *isomorphic* if there exists an isomorphism between them.
This is denoted by writing $A \cong B$.

Example 1.6. Let us consider some familiar categories where the objects are structured sets.

1. Set. Here, the isomorphisms are precisely the bijections.

2. $\mathsf{Mon}, \mathsf{Grp}, \mathsf{Ring}, \mathsf{Field}, \mathsf{Vec}_{\mathbb{k}}$. Here, the isomorphisms are precisely the (appropriate) homomorphisms[1] which are bijective.

3. Pos. Here, the isomorphisms are ***not*** the same as bijective order-preserving maps. While every isomorphism is indeed a bijective order-preserving map, the converse is not true.
   For example, consider the posets P_1 and P_2 defined on $\{a, b\}$. Let the order on P_1 be the discrete one. Let the order on P_2 be $a \leq b$. Consider the identity

[1] By "appropriate," we mean that one must consider *monoid* homomorphisms in the case of Mon and so on.

function as an arrow from P_1 to P_2. This is clearly a bijection that preserves the (trivial) order relations. However, this clearly is not an isomorphism of posets. (One may manually verify that the set-theoretic inverse is not an arrow in Pos. Later, we shall see a method that utilises the structure.)

4. Top. Once again, it is ***not*** the case that continuous bijections are isomorphisms.

The last two examples illustrate how isomorphisms need not always take the same form. The definition of isomorphism is our first example of an *abstract*, category theoretic definition of an important notion. It is abstract in the sense that it makes use only of the category theoretic notions, rather than some additional information about the objects and arrows.

As the reader may be familiar that one often defines a "group isomorphism" to be a bijective homomorphism. This definition makes use of the following fact - the inverse of a bijective homomorphism is once again a homomorphism. The advantage of our definition is that applies in *any* category.

§§1.6 Categories - New from Old

We now consider some constructions that help us create new categories from old.

1. *Product.* The product of two categories $\mathbf{C}$ and $\mathbf{D}$, written as

$$\mathbf{C} \times \mathbf{D}$$

has objects of the form (C, D), for $C \in \mathbf{C}$ and $D \in \mathbf{D}$, and arrows of the form

$$(f, g) : (C, D) \to (C', D')$$

for $f : C \to C' \in \mathbf{C}$ and $g : D \to D \in \mathbf{D}$. Composition and units are defined componentwise, that is,

$$\begin{aligned}(f, g) \circ (f', g') &= (f \circ f', g \circ g'),\\ 1_{(C,D)} &= (1_C, 1_D).\end{aligned}$$

There are two obvious *projection functors*

$$\mathbf{C} \xleftarrow{\pi_1} \mathbf{C} \times \mathbf{D} \xrightarrow{\pi_2} \mathbf{D}$$

defined by $\pi_1(C, D) = C$ and $\pi_1(f, g) = f$, and similarly for π_2.
As groups are monoids, one may recognise them as categories. In this case, the reader familiar with groups may recognise that for groups G and H, the product category $G \times H$ is the same as the (direct) group product $G \times H$ interpreted as a category.

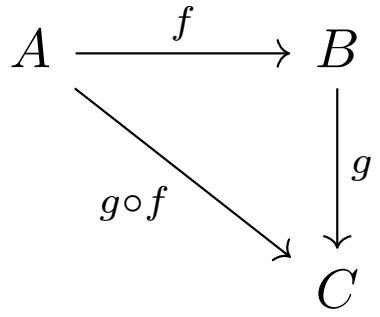
2. The *opposite* (or "dual") category $\mathbf{C}^{\mathrm{op}}$ of a category $\mathbf{C}$ has the same objects and arrow $f : C \to D$ in $\mathbf{C}^{\mathrm{op}}$ is an arrow $f : D \to C$ in $\mathbf{C}$. That is, $\mathbf{C}^{\mathrm{op}}$ is just $\mathbf{C}$ with all of the arrows formally turned around.
For easier notation, we shall write

$$f^* : D^* \to C^*$$

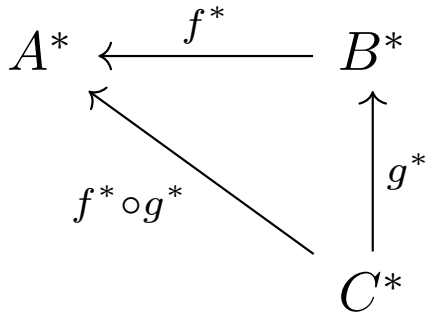
in $\mathbf{C}^{\mathrm{op}}$ for $f : C \to D$ in $\mathbf{C}$. With this notation, we have the units and composition rule as

$$1_{C^*} = (1_C)^*$$
$$f^* \circ g^* = (g \circ f)^*.$$

Thus, a diagram in $\mathbf{C}$



looks like this in $\mathbf{C}^{\mathrm{op}}$



3. The *arrow category* $\mathbf{C}^{\rightarrow}$ of a category $\mathbf{C}$ has the arrows of $\mathbf{C}$ as objects and

arrow g from $f : A \to B$ to $f' : A \to B'$ in $\mathbf{C}^{\to}$ is a "commutative square"

$$\begin{array}{ccc} A & \xrightarrow{g_1} & A' \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle f'} \\ B & \xrightarrow[g_2]{} & B' \end{array}$$

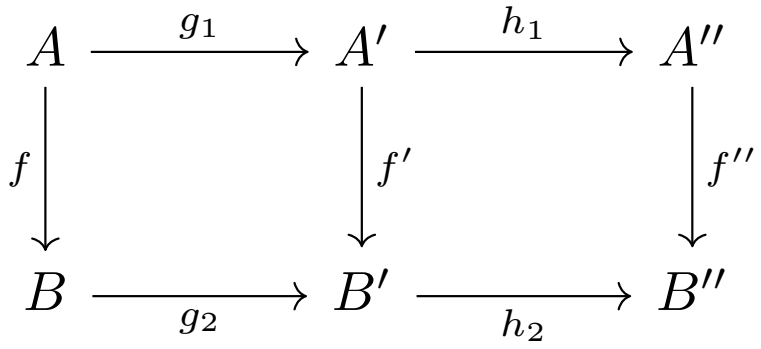
where g_1 and g_2 are arrows in $\mathbf{C}$. That is, such an arrow is a pair of arrows $g = (g_1, g_2)$ in $\mathbf{C}$ such that

$$g_2 \circ f = f' \circ g_1.$$

(A possible confusion that may arise in the mind of the reader is - why should such a commutative square exist? However, note that we are not claiming the existence of such a square. It could very well be possible that there is no arrow between two given objects in $\mathbf{C}^{\to}$.)
The identity arrow 1_f on an object $f : A \to B$ is the pair $(1_A, 1_B)$.
Composition of arrows is componentwise:

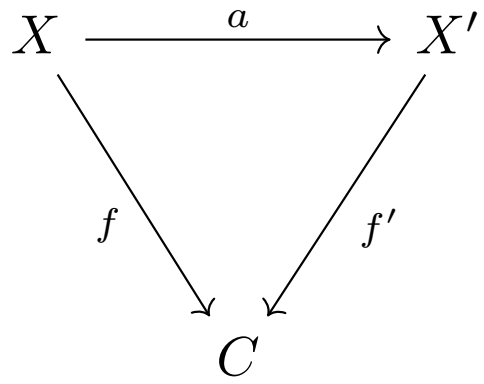
$$(h_1, h_2) \circ (g_1, g_2) = (h_1 \circ g_1, h_2 \circ g_2).$$

That this works can be verified using the diagram



4. The *slice category* $\mathbf{C}/C$ of a category $\mathbf{C}$ over an object $C \in \mathbf{C}$ has

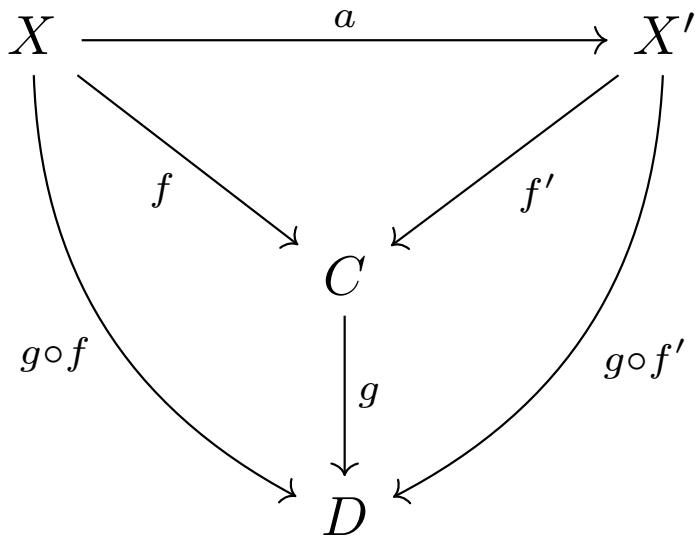
   - Objects: all arrows $f \in \mathbf{C}$ such that $\mathrm{cod}(f) = C$,
   - Arrows: an arrow a from $f : X \to C$ to $f' : X' \to C$ is an arrow $a : X \to X'$ in $\mathbf{C}$ such that $f' \circ a = f$, as indicated in



The identity arrows and compositions are induced from those of $\mathbf{C}$, just like the case of arrow category. Note that there is a functor $U : \mathbf{C}/C \to \mathbf{C}$ that "forgets about the base object C."
If $g : C \to D$ is a any arrow, then there is a composition functor,

$$g_* : \mathbf{C}/C \to \mathbf{C}/D$$

defined by $g_*(f) = g \circ f$, and similarly for the arrows in $\mathbf{C}/C$. (The arrows just go to themselves.)
The following diagram may be useful for the reader.



To repeat, we saw that given an object $C \in \mathbf{C}$, we get a category $\mathbf{C}/C$. Moreover, given an arrow $g : C \to D \in \mathbf{C}$, we get a functor $g_* : \mathbf{C}/C \to \mathbf{C}/D$.

Recalling that categories and functors are nothing but objects and arrows in Cat, this suggests that the above construction is in fact a functor. In fact, this is true as the reader may verify

$$\mathbf{C}/(-) : \mathbf{C} \to \mathsf{Cat}$$

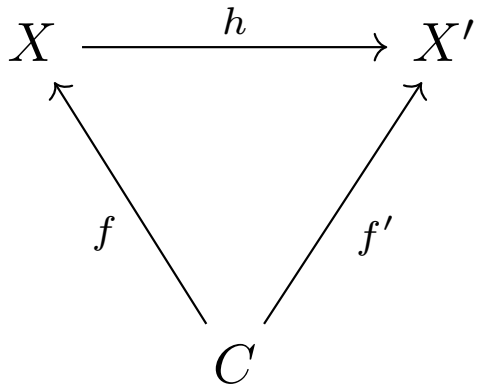
to be a functor. If $\mathbf{C} = \mathbf{P}$ is a poset category and $p \in \mathbf{P}$, then

$$\mathbf{P}/p \cong \downarrow(p),$$

that is, the slide category $\mathbf{P}/p$ is just the *principal ideal* $\downarrow(p)$ consisting of elements $q \in \mathbf{P}$ with $q \leq p$.

5. Similarly, the *coslice* category $C/\mathbf{C}$ of a category $\mathbf{C}$ under an object C of $\mathbf{C}$ has as objects all arrows f of $\mathbf{C}$ such that $\mathrm{dom}(f) = C$, and an arrow from $f : C \to X$ to $f' : C' \to X$ is an an arrow $h : X \to X'$ such that

$h \cdot f = f'$.

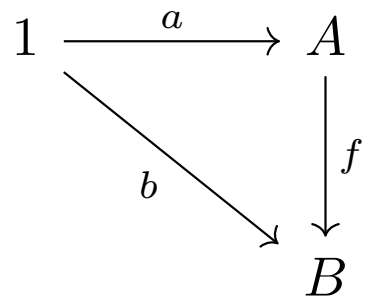


Example 1.7. The category Set_* of *pointed sets* consists of sets A with distinguished elements $a \in A$, and arrows $f : (A, a) \to (B, b)$ are functions $f : A \to B$ that preserves the "points," $f(a) = b$. This is isomorphic to the coslice category,

$$\mathsf{Set}_* \cong 1/\mathsf{Set}$$

of Set under any singleton $1 = \{*\}$. To see this, note that the functions $a : 1 \to A$ correspond uniquely to elements, $a(*) = a \in A$, and arrows $f : (A, a) \to (B, b)$

correspond exactly to commutative triangles:



§§1.7 Free categories

Free monoid. First we look at the concept of a free monoid.
First, we start with a set A which we shall call an "alphabet." We shall denote its elements as $a, b, c, \ldots$ and call them "letters," that is,

$$A = \{a, b, c, \ldots\}.$$

A *word* over A is a finite sequence of letters:

$$aryaman,\ integral,\ lenny,\ face,\ alkmdslkd, \ldots.$$

We write ε for the empty word, that is, the unique word over A of length zero. The "Kleene closure" of A is defined to be the set

$$A^* = \{\text{words over } A\}.$$

The above can easily be made into a monoid with composition (denoted by $*$) as concatenation. Accordingly, ε is the unit. Thus, A^* is a monoid, called the *free monoid* over the set A.
Any element $a \in A$ can also be regarded as a word of length one and hence, we have a function

$$i : A \to A^*$$

defined by $i(a) = a$, and called the "insertion of generators." The elements of A "generate" the free monoid, in the sense that every $w \in A^*$ is a $*-$product of as, that is, $w = a_1 * a_2 * \cdots * a_n$ for some $a_1, \ldots, a_n \in A$. One usually defines the property of being "free" in the following manner:
A monoid M is freely generated by $A \subset M$, if the following conditions hold:

1. Every element $m \in M$ can be written as a product of elements of A :

$$m = a_1 \cdot_M \cdots \cdot_M a_n, \quad a_i \in A.$$

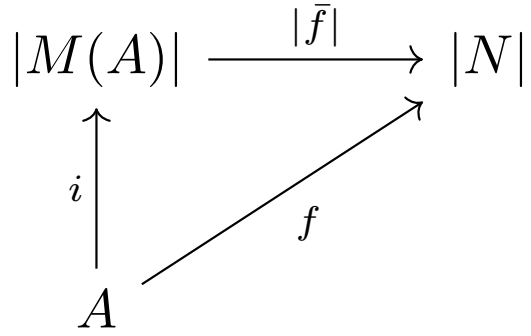
2. No "nontrivial" relations hold in M, that is, if $a_1 \cdots a_j = a'_1 \cdots a'_k$, then this is required by the axioms for monoids.

The second condition might seem a little vague and imprecise. We give a precise definition of "free" - capturing what is meant in the above - which avoids vagueness.
First, every monoid N has an underlying set $|N|$, and every monoid homomorphism $f : N \to M$ has an underlying function $|f| : |N| \to |M|$. This is nothing but the action of the forgetful functor seen earlier. "The" free monoid on a set A is, by definition, "the" monoid with the following so-called *universal mapping property* or UMP.

Universal Mapping Property of $M(A)$
There is a function $i : A \to |M(A)|$, and given any monoid N and any function $f : A \to |N|$, there is a *unique* monoid homomorphism $\bar{f} : M(A) \to N$ such that $|\bar{f}| \circ i = f$, all as indicated in the following diagram:
in Mon:

$$M(A) \overset{\bar{f}}{\dashrightarrow} N$$

in Set:



The reader is encouraged to prove the following proposition on their own.

Proposition 1.8. A^* has the UMP of the free monoid on A.

Proof. Given $f : A \to |N|$, define $f : A^* \to N$ by

$$f(\varepsilon) = u_N, \text{ the unit of } N,$$
$$f(a_1 \cdots a_i) = f(a_1) \cdot_N \cdots \cdot_N f(a_i).$$

Then, $\bar{f}$ is clearly a homomorphism with

$$\bar{f}(a) = f(a) \quad \text{for all } a \in A.$$

Now, we prove the uniqueness of $\bar{f}$.
If $g : A^* \to N$ also satisfies $g(a) = f(a)$ for all $a \in A$, then for all $a_1 \cdots a_i \in A^*$:

$$\begin{aligned} g(a_1 \cdots a_i) &= g(a_1 * \cdots * a_i) \\ &= g(a_1) \cdot_N \cdots \cdot_N g(a_i) \\ &= f(a_1) \cdot_N \cdots \cdot_N f(a_i) \\ &= \bar{f}(a_1) \cdot_N \cdots \cdot_N \bar{f}(a_i) \\ &= \bar{f}(a_1) \cdot_N \cdots \cdot_N g(a_i) \\ &= \bar{f}(a_1 \cdots a_i) \end{aligned}$$

Thus, $g = \bar{f}$ and hence, uniqueness is proved. □

Using the UMP, it is also easy to show that the free monoid $M(A)$ is determined uniquely, up to isomorphism, in the following sense.

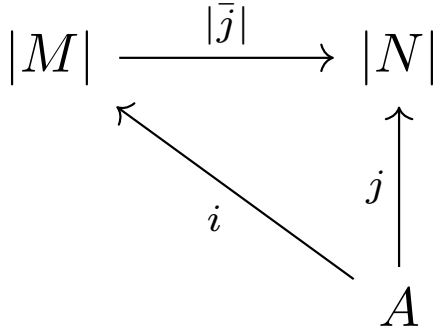
Proposition 1.9. Given monoids M and N with functions $i : A \to |M|$ and $j : A \to |N|$, each with the UMP of the free monoid of A, there is a (unique) monoid isomorphism $h : M \to N$ such that $|h|i = j$ and $|h^{-1}|j = i$.

Once again, the reader is encouraged to prove this on their own as it's a fun exercise.

Proof. Using j and the UMP of M, we get a monoid homomorphism $\bar{j} : M \to N$ with the following diagrams:
in Set:

$$M \overset{\bar{j}}{\dashrightarrow} N$$

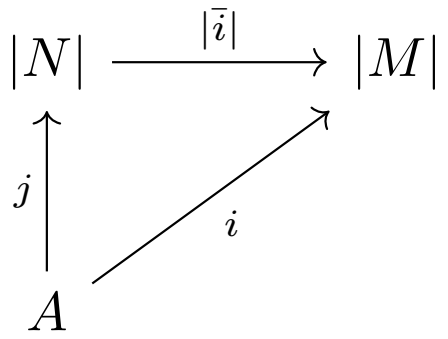
in Mon:



On the other hand, using i and the UMP of N we get a monoid homomorphism $\bar{i} : N \to M$ with the following diagrams:
in Set:

$$N \overset{\bar{j}}{\dashrightarrow} M$$

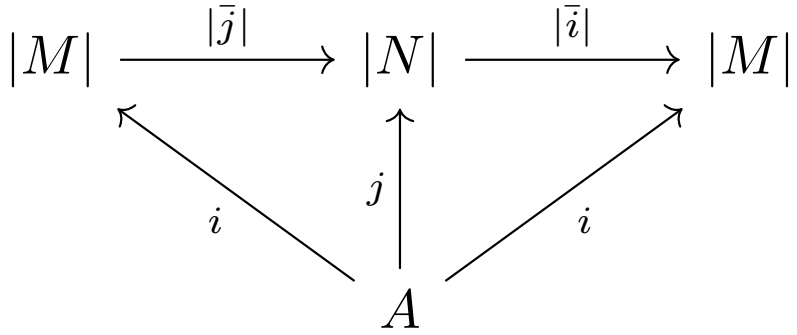
in Mon:



Composing the above arrows gives us the diagrams: in Set:

$$M \overset{\overline{j}}{\dashrightarrow} N \overset{\overline{j}}{\dashrightarrow} M$$

in Mon:

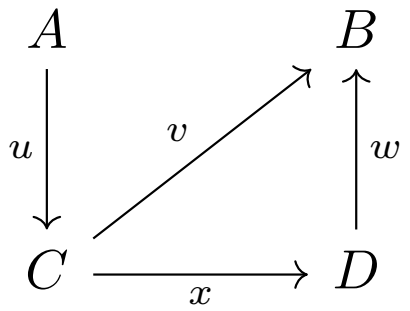


Now, look at the monoid homomorphism $\overline{i} \circ \overline{j} : M \to M$.
It has the property that $|\overline{i} \circ \overline{j}|i = i$. As $1_M : M \to M$ has this property, by the

uniqueness part of the UMP of M, it follows that $\overline{i} \circ \overline{j} = 1_M$. Exchanging the roles of M and N shows that $\overline{j} \circ \overline{i} = 1_N$.
This finishes our proof. □

(How do we get that this isomorphism is unique?)

Free category. Now, we want to do the same thing for categories in general. Instead of underlying sets, categories have underlying graphs, so let us review these first.
A *directed graph* consists of vertices and edges, each of which is directed, that is, each edge has a "source" and a "target" vertex.



We draw graphs just like categories, but there is no composition of edges, and there are no identities.
Thus, a graph consists of two sets, E (edges) and V (vertices), and two functions,

$s : E \to V$ (source) and $t : E \to V$ (target). Thus, in Set, a graph is just a configuration of objects and arrows of the form

$$E \underset{t}{\overset{s}{\rightrightarrows}} V$$

Now, every graph "generates" a category, the *free category* on G. This is similar in spirit to the construction of the free monoid on a set. There we created the words by writing letters one after the other. We do the same here but adjoining arrows only if the source and target matches. To be more precise, the free category $\mathbf{C}(G)$ is defined by taking the vertices of G as objects, and the *paths* in G as arrows, where a path is a finite sequence of edges $e_1, \ldots, e_n$ such that $t(e_i) = s(e_{i+1})$, for all $i = 1, \ldots, n$. We write the arrows of $\mathbf{C}(G)$ in the form $e_n e_{n-1} \ldots e_1$.

$$v_0 \xrightarrow{e_1} v_1 \xrightarrow{e_2} \ldots \xrightarrow{e_n} v_n$$

Put

$$\begin{aligned} \mathrm{dom}(e_n \ldots e_1) &= s(e_1) \\ \mathrm{cod}(e_n \ldots e_1) &= t(e_n) \end{aligned}$$

and define composition by concatenation:

$$e_n \dots e_1 \circ e'_m \dots e'_1 = e_n \dots e_1 e'_m \dots e'_1.$$

For each vertex v, we have an "empty path" denoted by 1_v, which is to be the identity arrow at v.

We will show that the $\mathbf{C}(G)$ so defined also has a UMP. Before that, we make a slight digression.

First, we observe that any category $\mathbf{C}$ can be described with a diagram like this:

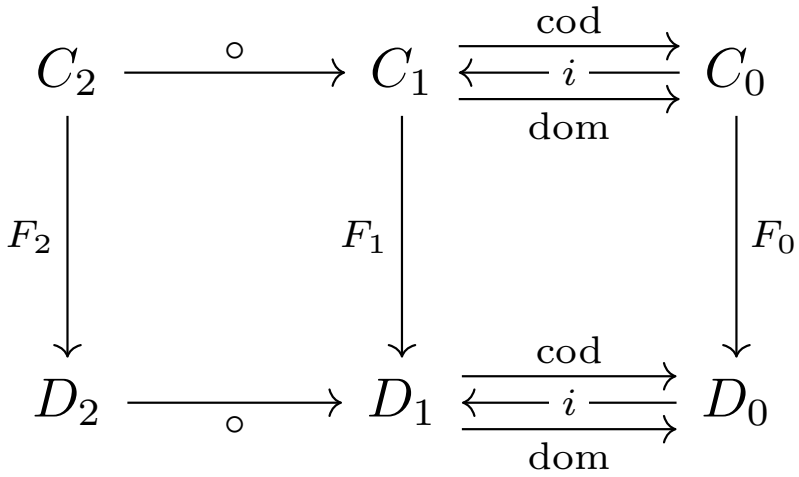
$$C_2 \xrightarrow{\quad\circ\quad} C_1 \underset{\xleftarrow{\quad i \quad}}{\overset{\overset{\mathrm{cod}}{\longrightarrow}}{\underset{\mathrm{dom}}{\longrightarrow}}} C_0$$

where C_0 is the collection of objects of $\mathbf{C}$, C_1 the arrows, i is the identity operation, and C_2 is the collection $\{(f, g) \in C_1 \times C_1 : \mathrm{cod}(f) = \mathrm{dom}(g)\}$. Then, a functor $F : \mathbf{C} \to \mathbf{D}$ from $\mathbf{C}$ to another category $\mathbf{D}$ is a pair of functions

$$F_0 : C_0 \to D_0$$
$$F_1 : C_1 \to D_1$$

such that each similarly labeled square in the following diagram commutes:

(1.3)



where $F_2(f, g) = (F_1(f), F_1(g))$.
Now let us describe a *homomorphism of graphs*,

$$h : G \to H.$$

We need a pair of functions $h_0 : G_0 \to H_0, h_1 : G_1 \to H_1$ making the two squares commute (once with ts and once with ss) in the following diagram commute:

(1.4)

$$\begin{array}{ccc} G_1 & \overset{t}{\underset{s}{\rightrightarrows}} & G_0 \\ \downarrow{\scriptstyle h_1} & & \downarrow{\scriptstyle h_0} \\ H_1 & \overset{t}{\underset{s}{\rightrightarrows}} & H_0 \end{array}$$

With this in place, we can now describe the forgetful functor

$$U : \mathsf{Cat} \to \mathsf{Graphs}$$

as sending the category

$$C_2 \xrightarrow{\circ} C_1 \underset{\mathrm{dom}}{\overset{\mathrm{cod}}{\underset{\longrightarrow}{\overset{\longrightarrow}{\longleftarrow i}}}} C_0$$

to the underlying graph

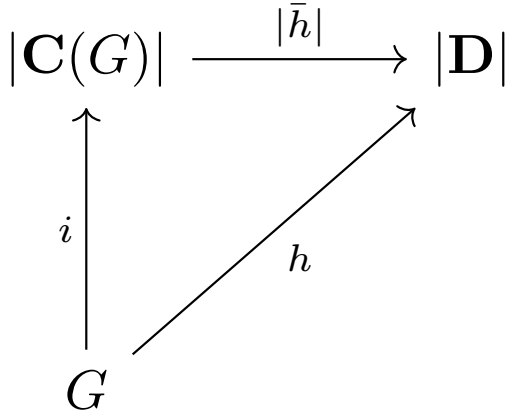
$$C_1 \underset{\text{dom}}{\overset{\text{cod}}{\rightrightarrows}} C_0.$$

And similarly, the effect of U is described functors by erasing some of the arrows of (1.3) to get a diagram like (1.4).
Recall how we had defined the UMP of free monoid using the forgetful functor. We shall do the same in this case. Borrowing the same notation, we shall write $|\mathbf{C}| = U(\mathbf{C})$, et cetera, for the underlying graph of a category $\mathbf{C}$.
The free category on a graph now has the following UMP.

Universal mapping property of $\mathbf{C}(G)$.
There is a graph homomorphism $i : G \to |\mathbf{C}(G)|$, and given any category $\mathbf{D}$ and any graph homomorphism $h : G \to |\mathbf{D}|$, there is a unique functor $\bar{h} : \mathbf{C}(G) \to \mathbf{D}$ with $|\bar{h}| \circ i = h$.
in Cat:

$$\mathbf{C}(G) \overset{\bar{h}}{\dashrightarrow} \mathbf{D}$$

in Graph:



Example 1.10. The free category on a graph with just one vertex is just a free monoid on the set of edges. The free category on a graph with only vertices (no edges) is the discrete category on the set of vertices of G. The free category on a graph with two vertices and one edge between them is the finite category $\mathbf{2}$. The free category on a graph of the form

$$A \underset{f}{\overset{e}{\rightleftarrows}} B$$

has (in addition to the identity arrows) the infinitely many arrows:

$$e, f, ef, fe, efe, fef, \ldots$$

Recall we had seen the above graph earlier ((1.2)) when we were discussing finite categories.

§2 Abstract Structures

§§2.1 Epis and Monos

Definition 2.1. In any category C, an arrow

$$f : A \to B$$

is called a

- *monomorphism*, if given any $g, h : C \to A$, $fg = fh$ implies $g = h$.
- *epimorphism*, if given any $i, j : B \to D$, $if = jf$ implies $i = j$.

$$C \underset{h}{\overset{g}{\rightrightarrows}} A \xrightarrow{f} B \underset{j}{\overset{i}{\rightrightarrows}} D$$

We often write $f : A \rightarrowtail B$ if f is a monomorphism and $f : A \twoheadrightarrow B$ if f is an epimorphism.

Proposition 2.2. A function $f : A \to B$ between is monic iff f is injective.

Proof. $(\implies)$ Suppose that f is monic. We show that f is injective.
Let $a, a' \in A$ be such that $f(a) = f(a')$.
Consider $g : A \to A$ defined as

$$g(x) = \begin{cases} x & x \neq a \\ a' & x = a \end{cases}$$

and let $h : A \to A = 1_A$.
Clearly, one sees that $fg = f = f1_A$. As f is monic, we have that $1_A = g$. In particular, $g(a) = 1_A(a)$ which yields $a' = a$.
$(\impliedby)$ Suppose that f in injective. We show that f is monic.
Let $g, h : C \to A$ be arrows such that $fg = fh$. Let $a \in A$ be arbitrary. Then, $fg(a) = fh(a)$. As f is injective, this yields $g(a) = h(a)$. □

Before going ahead, we may make the following observation for proving that $f : A \to B$ is injective.

Proposition 2.3. Let $f : A \to B$ be a function. Let $1 = \{*\}$ be any one-element set.

Suppose that $fg \neq fh$ whenever $g, h : 1 \to A$ are distinct functions.
Then, f is injective.

Proof. Let $a, a' \in A$ be such that $a \neq a'$. Consider the functions

$$\bar{a}, \bar{a'} : 1 \to A$$

where

$$\bar{a}(*) = a, \quad \bar{a'}(x) = a'.$$

Since $\bar{a} \neq \bar{a'}$, it follows from our hypothesis that $f\bar{a} \neq f\bar{a'}$. Thus, $f(a) = (f\bar{a})(x) \neq (f\bar{a'})(x) = f(a')$. Hence, it follows that f is injective. □

Using this proposition, one may give a slightly simpler proof of $(\implies)$ of Proposition 2.2.

Example 2.4. In many categories of "structured sets" like monoids, the monos are exactly the "injective homomorphisms". More precisely, a homomorphism $h : M \to N$ is monic precisely if the underlying function $|h| : M \to N$ is injective. (By the above, it is the same as saying $|h|$ is monic.)

To see that $|h|$ being injective $\implies$ h is monic, one may consider the earlier proof.
Conversely, let $h : M \to N$ be monic. We show that $|h|$ is monic.
Suppose $x, y : 1 \to |M|$ are two different "elements" (arrows), where $1 = \{*\}$, any one-element set. By Proposition 2.3, it suffices to prove that $|h|x, |h|y : 1 \to N$ are also distinct.
By the UMP of the free monoid $M(1)$, there are distinct corresponding homomorphisms $\bar{x}, \bar{y} : M(1) \to M$, with distinct composites $h \circ \bar{x}, h \circ \bar{y} : M(1) \to N$, since h is monic. Thus, the corresponding "elements" $|h| \circ x, |h| \circ y : 1 \to N$ are also distinct, again by the UMP of $M(1)$.

Example 2.5. In a poset $\mathbf{P}$, every arrow $p \leq q$ is both monic and epic. This follows trivially from the fact that given any two objects, there is at most one arrow from one to the other.

Dually to the foregoing, the epis in Set are exactly the surjective functions. This is a fun exercise that is left to the reader. However, unlike the previous case in Mon, the epis are not just the surjective homomorphisms!

Example 2.6. In the category Mon of monoids and monoid homomorphisms, there is a monoid homomorphism

$$\mathbb{N} \rightarrowtail \mathbb{Z}$$

where $\mathbb{N}$ is the additive monoid of nonnegative integers and $\mathbb{Z}$ is the additive monoid of integers. We show that this map, given by the inclusion, is also epic in Mon by showing that the following holds:
Given any monoid homomorphisms $f, g : (|\mathbb{Z}|, +, 0) \to (M, *, u)$, if the restrictions to $|\mathbb{N}|$ are equal, then $f = g$.
The restrictions to $|\mathbb{N}|$ being equal already tells us that $f(n) = g(n)$ whenever $n \geq 0$. Let us assume that $n < 0$. One then notes

$$\begin{aligned} f(n) &= f(n) * u \\ &= f(n) * g(0) \\ &= f(n) * g(-n+n) \\ &= f(n) * g(-n) * g(n) \\ &= f(n) * f(-n) * g(n) \\ &= f(0) * g(n) \\ &= u * g(n) \\ &= g(n) \end{aligned}$$

From an algebraic point of view, a morphism e is epic if and only if e cancels on the right: $xe = ye$ implies $x = y$. Dually, m is monic if and only if m cancels on the left: $mx = my$ implies $x = y$.

Note that, of course, this does not mean that either e or m is invertible. (Even if they're cancel-able from both sides.)

However, *if* a morphism *is* invertible, then it is clearly a mono and an epi. This leads to the next proposition.

Proposition 2.7. Every iso is both monic and epic.

Proof. Let e be an isomorphism with m as its inverse. Let x and y be arrows such that $xe = ye$. Then, one has $xem = yem$ or $x = y$, as desired. Similarly, $ef = eg$ implies that $f = g$. □

In Set, the converse also holds: every mono-epi is iso. But this is not true in the general case, as Example 2.6 showed us.

2.1.1 Sections and retractions

We just noted that any iso is both monic and epic. However, carefully looking at the above proof tells us the following more general result:
Let $f : A \to B$ and $g : B \to A$ be arrows such that $gf = 1_A$. Then, f is monic and g epic.
This leads to the following definition.

Definition 2.8. A *split* mono (epi) is an arrow with a left (right) inverse. Given arrows $e : X \to A$ and $s : A \to X$ such that $es = 1_A$, the arrow s is called a *section* or a *splitting* of e, and the arrow e is called a *retraction* of s. The object A is called a *retract* of X.

Clearly every *split* mono (epi) is also a mono (epi) but the converse is not true in general. Since functors preserve identities, they also preserve *split* epis and monos. Compare this with Example 2.6 where the forgetful functor $\mathsf{Mon} \to \mathsf{Set}$ clearly does not preserve the epi $\mathbb{N} \rightarrowtail \mathbb{Z}$. (Thus, this also serves as an example of an epi that does not split.)

Example 2.9. In Set, every mono splits except those of the form

$$\varnothing \rightarrowtail A.$$

To see this, let $f : X \rightarrowtail A$ be a mono with $X \neq \varnothing$. Pick some $x_0 \in X$ and define $g : A \twoheadrightarrow X$ as

$$g(a) = \begin{cases} b & a = f(b) \\ x_0 & a \neq f(b) \text{ for any } b \in X \end{cases}$$

In view of f being monic (and hence, injective), the above function is well defined and the reader may verify that $gf = 1_X$.
Let us look at the more interesting condition of an epi splitting.

Lemma 2.10. In Set, the condition that every epi splits is equivalent to the axiom of choice.

Proof. First, let us assume the axiom of choice. Consider an epi

$$e : E \twoheadrightarrow X.$$

If $X = \varnothing$, then $E = \varnothing$ and e is its own inverse. Let us now assume that $X \neq \varnothing$. Then, we have the following nonempty collection of nonempty sets:

$$E_x = e^{-1}\{x\}, \quad x \in X.$$

A choice function for this family $(E_x)_{x \in X}$ is exactly a splitting of e, that is, a function $s : X \to E$ such that $es = 1_X$, since that means that $s(x) \in E_x$ for all $x \in X$. Conversely, assume that every epi splits. Given a nonempty collection of nonempty sets,

$$(E_x)_{x \in X}$$

take $E = \{(x, y) \mid x \in X, y \in E_x\}$ and define the epi $e : E \twoheadrightarrow X$ by $(x, y) \mapsto x$. A splitting s of e then determines a choice function for the collection. □

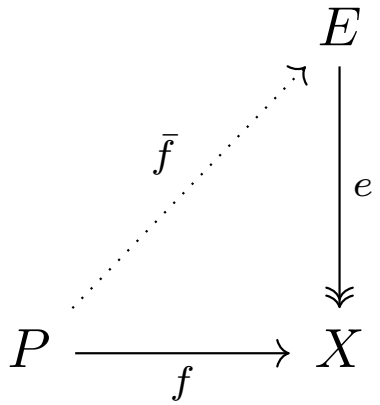
Corollary 2.11. The Axiom of Choice is equivalent to the following statement: Given any surjective map $s : A \twoheadrightarrow B$, there exists a map $f : B \to A$ such that $sf = 1_B$.

2.1.2 Projective objects

A notion related to the existence of "choice functions" is that of being "projective."

Definition 2.12. An object P is said to be *projective* if for any epi $e : E \twoheadrightarrow X$ and arrow $f : P \to X$ there is some (not necessarily unique) arrow $\bar{f} : P \to E$ such that $e \circ \bar{f} = f$, as indicated in the following diagram:

(2.1)



One says that f *lifts across* e.

Proposition 2.13. Any epi into a projective object splits.

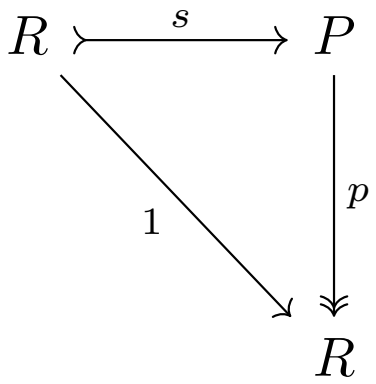
Proof. Let P be a projective object and $e : E \twoheadrightarrow P$ an epi. Consider (2.1) with $X = P$ and $f = 1_P$. The lift $\overline{1_P}$ is clearly a right inverse of e. □

Proposition 2.14. The axiom of choice implies that all sets are projective.

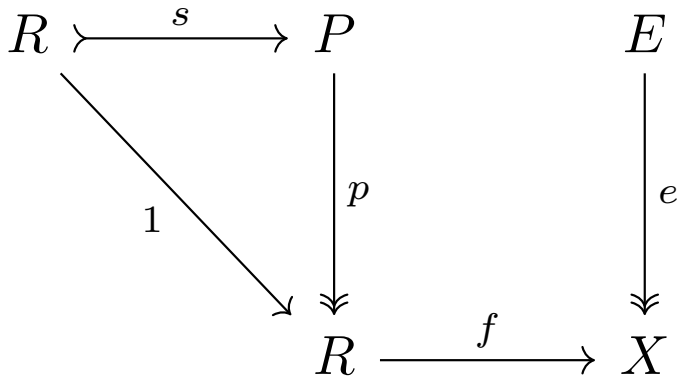
Proof. Consider E, P, X, e, f as in the Definition 2.12. By Lemma 2.10, we see that there exists $s : X \to E$ such that $es = 1_X$. One sees that $\bar{f} = sf$ has the desired property. □

Proposition 2.15. Any retract of a projective object is also projective.

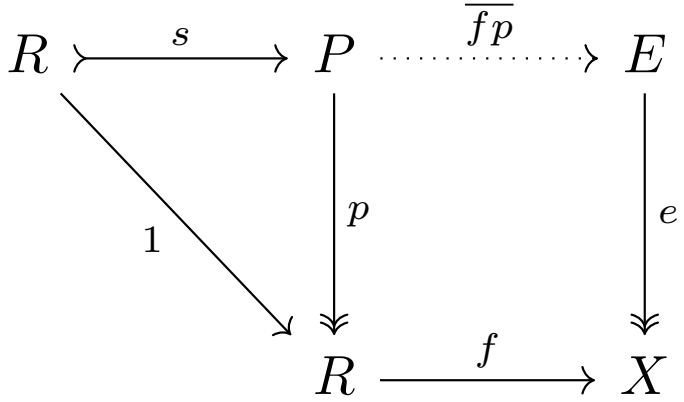
Proof. Let P be a projective object and R a retract of P. Let p, s be as pictured.



Let $e : E \twoheadrightarrow X$ be an epi and let $f : R \to X$ be an arrow. We show that e lifts across f. The information so far can be pictured as follows:



Now, since P is projective, we get that e lifts across $fp : P \to E$. Let this lift be $\overline{fp}$ as shown.



Then, $\bar{f} = \overline{fp}s$ is the desired lift of e across f. To see this, one observes that the

above diagram commutes, that is,

$$\begin{aligned} e(\overline{fp}s) &= (e\overline{fp})s \\ &= (fp)s \\ &= f(ps) \\ &= f(1) \\ &= f. \end{aligned}$$

□

§§2.2 Initial and terminal objects

Definition 2.16. In any category $\mathbf{C}$, an object

- 0 is *initial* if for any object C there is a unique morphism,

$$0 \to C,$$

- 1 is *terminal* if for any object C there is a unique morphism,

$$C \to 1.$$

One may note that a terminal object in $\mathbf{C}$ is precisely an initial object in $\mathbf{C}^{\mathrm{op}}$. Note that a category need not have an terminal or initial object. A category may also have one more than of either. However, the following proposition tells us that they must be isomorphic.

Proposition 2.17. Initial (terminal) objects are unique up to isomorphism.

Proof. We shall prove the statement for initial objects.
Let C and C' be initial objects. Let $i : C \to C'$ and $i' : C' \to C$ be the unique morphisms between these objects.
Consider the morphism $i \circ i'$. It is a morphism from C' to C'. Note that $1_{C'}$ is a morphism from C' to C'. As C' is an initial object, there is only one morphism from C' to itself. Thus, $i \circ i' = 1_{C'}$. A similar argument shows that $i' \circ i = 1_C$ and thus, i is an isomorphism. □

Example 2.18.

1. In Set, the empty set is initial and any singleton set $\{x\}$ is terminal. Observe that Set has just one initial object but many terminal objecs.

2. In Cat, the category $\mathbf{0}$ (no objects and no arrows) is initial and the category $\mathbf{1}$ (one object and its identity arrow) is terminal.

3. In Grp, the one-element group is both initial and terminal. (Same for $\mathsf{Vec}_{\mathbb{k}}$ and Mon.) But in Ring (commutative with unit), the ring $\mathbb{Z}$ is initial and the one-element ring is terminal.

4. Recall Boolean algebra. Another familiar example of the two-element Boolean algebra $\mathbf{2} = \{0, 1\}$ (which may be taken to be the power-set $\mathcal{P}(1)$). It is an initial object in the category BA of Boolean algebras and boolean homomorphisms.
   The one-element structure (i.e., $\mathcal{P}(0)$) is terminal

5. In a poset (viewed as a category), an object is initial iff is the least and terminal iff is the greatest. Thus, for instance, any Boolean algebra has both. (Note that we are viewing the Boolean algebra as a category in itself. Different from what we did in the previous example.)
   On the other hand, the poset $(\mathbb{Z}, \leq)$ has neither.

6. For any category $\mathbf{C}$ and any object $X \in \mathbf{C}$, the identity arrow $1_X : X \to X$

is a terminal object in the slice category $\mathbf{C}/X$ and an initial object in the coslice category $X/\mathbf{C}$.

§§2.3 Generalised elements

Let us consider arrows into and out of initial and terminal objects.
A set A has an arrow into the initial object 0 precisely if it is itself initial and the same is true for poset categories. In monoids and groups, by contrast, every object has a unique arrow to the initial object, since it is also terminal.
Let us consider some arrows from terminal objects. For any set X, for instance, we have an isomorphism

$$X \cong \mathrm{Hom}_{\mathsf{Set}}(1, X)$$

between elements $x \in X$ and arrows $\bar{x} : 1 \to X$, determined $\bar{x}(*) = x$, from a terminal object $1 = \{*\}$. A similar situation holds in posets and in topological spaces, where the arrows $1 \to P$ correspond to elements of the underlying set of a poset or topological space. In any category with terminal

objects 1, such arrows $1 \to A$ are often called *global elements*, or *points*, or *constants* of A. In sets, posets, and spaces, the general arrows $A \to B$ are determined by what they do to the points of A, in the sense that two arrows $f, g : A \to B$ are equal if for every point $a : 1 \to A$ the composites fa and ga are equal.

However, this is not always the case! Recall that in Mon, the terminal object 1 is also an initial object and hence, given any monoid M, there is a unique arrow $1 \to M$. (This is to say that M has one point.) In the category BA, there is no arrow $1 \to B$ if $B \neq 1$. (This is to say that B has no points.)

Thus, in general, an object is not determined by its points. For this reason, it is convenient to introduce the device of *generalised elements*. There are arbitrary arrows

$$x : X \to A$$

(with arbitrary domain X), which can be regarded as *generalised* or *variable elements* of A.

We summarise the above in the following list of examples:

Example 2.19.

- Consider arrows $f, g : P \to Q$ in Pos. Then, $f = g$ iff for all $x : 1 \to P$, we have $fx = gx$. In this sense, posets "have enough points" to separate the arrows.
- By contrast, in Mon, for homomorphisms $h, j : M \to N$, we always have $hx = jx$, for all $x : 1 \to M$, since there is just one such point x. However, we do have examples of distinct homomorphisms between two monoids. Thus, monoids do not "have enough points."
- But in any category $\mathbf{C}$, and for any arrows $f, g : C \to D$, we always have $f = g$ iff for all $x : X \to C$, it holds that $fx = gx$.
  The "only if" part is trivial. To see the "if" part, consider $X = C$ and $x = 1_C$.
  Thus, all objects have enough generalised elements.
- In fact, it often happens that it is enough to consider generalised elements of just a certain form $T \to A$, that is, for certain "test" objects T. We shall consider this presently.

Generalised elements are also good for "testing" for various conditions. Consider, for instance, diagrams of the following shape:

$$X \underset{x'}{\overset{x}{\rightrightarrows}} A \xrightarrow{f} B$$

The arrow f is monic iff $x \neq x'$ implies $fx \neq fx'$ for all x, x', that is, just if f is "injective on generalised elements."

Similarly, in any category $\mathbf{C}$, to test whether a square commutes,

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ {\scriptstyle g}\downarrow & & \downarrow{\scriptstyle \alpha} \\ D & \xrightarrow[\beta]{} & C \end{array}$$

we shall have $\alpha f = \beta g$ just if $\alpha f x = \beta g x$ for all generalised elements $x : X \to A$. (why?)

Example 2.20. Generalised elements can also be used to "reveal more structure" than do the constant elements. For example, consider the following posets X and A :

$$X = \{x \leq y, y \leq z\},$$
$$A = \{a \leq b \leq c\}.$$

There is an order preserving bijection $f : X \to A$ defined by

$$f(x) = a, \quad f(y) = b, \quad f(z) = c.$$

It is easy to see that f is both monic and epic in the category Pos; however, it is clearly not an iso. In fact, no map between them is an iso. However, how would one *prove* this? In this case, it is not that difficult to do this via "brute force."

One way to prove that two objects are not isomorphic is to use "invariants": attributes that are preserved by isomorphisms. If two objects differ by an invariant, they cannot be isomorphic. For instance, the number of global elements of X and A is the same, namely the three elements of the set. But consider instead the "$\mathbf{2}-$elements" $\mathbf{2} \to X$, from the poset $\mathbf{2} = \{0 \leq 1\}$ as a "test-object." Then X has 5 such elements, and A has 6. Since these numbers are invariants (why?), the posets cannot be isomorphic. In more detail, we can define for any poset P the numerical invariant

$$|\operatorname{Hom}(\mathbf{2}, P)| = \text{ the number of elements of } \operatorname{Hom}(\mathbf{2}, P).$$

Then if $P \cong Q$, it is easy to see that $|\mathrm{Hom}(\mathbf{2}, P)| = |\mathrm{Hom}(\mathbf{2}, Q)|$, since any isomorphism

$$P \underset{j}{\overset{i}{\rightleftarrows}} Q$$

also gives an isomorphism

$$\mathrm{Hom}(\mathbf{2}, P) \underset{j_*}{\overset{i_*}{\rightleftarrows}} \mathrm{Hom}(\mathbf{2}, Q)$$

defined by composition:

$$i_*(f) = if,$$
$$j_*(g) = jg,$$

for all $f : \mathbf{2} \to P$ and $g : \mathbf{2} \to Q$. Indeed, this is a special case of the very general fact that $\mathrm{Hom}(X, -)$ is always a functor, and functors always preserve isomorphisms.

Example 2.21. As in the foregoing example, it is often the case that generalised elements $t : T \to A$ "based at" certain objects T are especially revealing. We can think

of such elements geometrically as "figures of shape T in A," just as an arrow $\mathbf{2} \to P$ in posets is a figure of shape $p \leq p'$ in P. For instance, as we have already seen, in the category of monoids, the arrows from the terminal monoid are entirely uninformative, but those from the free monoid on one generator $M(1)$ suffice to distinguish homomorphisms, in the sense that two homomorphisms $f, g : M \to M'$ are equal if their composites with all such arrows are equal. Since we know that $M(1) = \mathbb{N}$, the monoid of natural numbers, we can think of generalised elements $M(1) \to M$ based at $M(1)$ as "figures of shape $\mathbb{N}$" in M. In fact, by the UMP of $M(1)$, the underlying set $|M|$ is therefore (isomorphic to) the collection $\mathrm{Hom}_{\mathsf{Mon}}(\mathbb{N}, M)$ of all such figures, since

$$|M| \cong \mathrm{Hom}_{\mathsf{Set}}(1, |M|) \cong \mathrm{Hom}_{\mathsf{Mon}}(\mathbb{N}, M).$$

In this sense, a map from a monoid is determined by its effect on all of the figures of shape $\mathbb{N}$ in the monoid.

§§2.4 Products

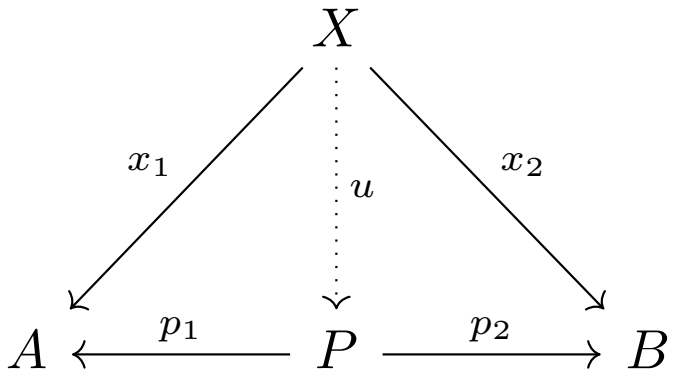
Definition 2.22. In any category $\mathbf{C}$, a *product diagram* for the objects A and B consists of an object P and arrows

$$A \xleftarrow{p_1} P \xrightarrow{p_2} B$$

satisfying the following UMP:
Given any diagram of the form

$$A \xleftarrow{x_1} X \xrightarrow{x_2} B$$

there exists a unique $u : X \to P$ making the diagram



commute, that is, such that $x_1 = p_1 u$ and $x_2 = p_2 u$.

Remark 2.23. As in other UMPs, there are two parts:

- *Existence*: There is some $u : X \to P$ such that $x_1 = p_1 u$ and $x_2 = p_2 u$.
- *Uniqueness*: Given any $v : X \to P$, if $p_1 v = x_1$ and $p_2 v = x_2$, then $v = u$.

Proposition 2.24. Products are unique up to isomorphism.

Proof. Suppose

$$A \xleftarrow{p_1} P \xrightarrow{p_2} B$$

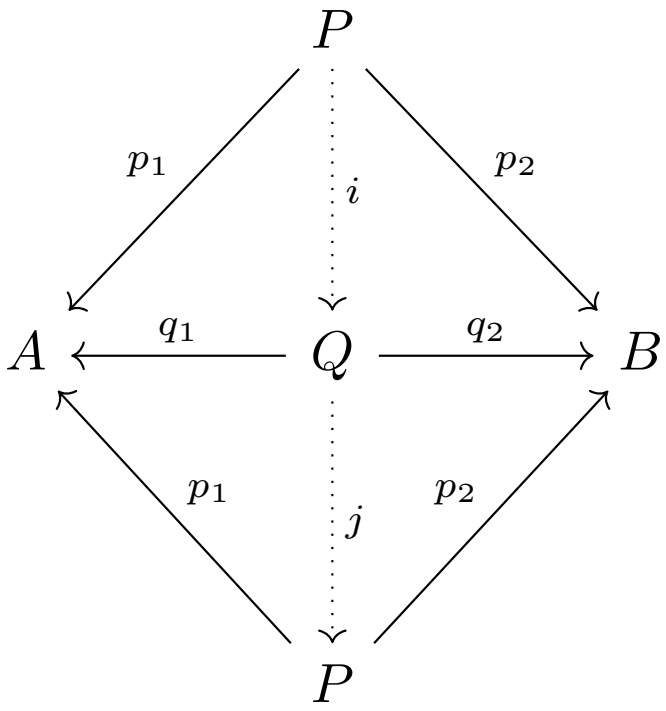
and

$$A \xleftarrow{q_1} Q \xrightarrow{q_2} B$$

are products of A and B. Since Q is a product, we get a (unique) morphism $i : P \to Q$ making the upper triangle of (2.2) commute. Similarly, since P is a product we get a

(unique) morphism $j : Q \to P$ making the lower triangle of (2.2) commute.

(2.2)



Note that the composite $j \circ i$ is a morphism from P to P which has the following property: $p_1 \circ j \circ i = p_1$ and $p_2 \circ j \circ i = p_2$. Note that 1_P also has the same property, id est, $p_1 \circ 1_P = p_1$ and $p_2 \circ 1_P = p_2$. Since P is a product, there is a such unique morphism and thus, $j \circ i = 1_P$. Interchanging the roles of P and Q gives us $i \circ j = 1_Q$ and hence, i and j are the desired isomorphisms. $\square$

If A and B have a product, we write

(2.3) $$A \xleftarrow{p_1} A \times B \xrightarrow{p_2} B$$

for one such product. Then given X, x_1, x_2 as in the definition, we write

(2.4) $$\langle x_1, x_2 \rangle \text{ for } u : X \to A \times B.$$

Note, however, that a pair of objects may have many different products in a category. For example, given a product $A \times B, p_1, p_2$, and any iso $h : A \times B \to Q$, the diagram $Q, p_1 \circ h, p_2 \circ h$ is also a product of A and B.
Now an arrow *into* a product

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

is "the same thing" as a pair of arrows

$$f_1 : X \to A, \quad f_2 : X \to B.$$

(This follows from the UMP.)
So, we can essentially forget about such arrows, in that they are uniquely determined

by pairs of arrows. But something useful *is* gained if a category has products; namely,, consider arrows *out* of the product,

$$g : A \times B \to Y.$$

Such a g is a "function of two variables"; given any two generalised elements $f_1 : X \to A$ and $f_2 : X \to B$, we have an element $g\langle f_1, f_2\rangle : X \to Y$. Such arrows $g : A \times B \to Y$ are not "reducible" to anything more basic, the way that products into arrows were.

§§2.5 Examples of Products

1. Set. In this category, every pair of objects does have a product. It is the usual Cartesian product (along with the projections). Given sets A and B, the *cartesian product* of A and B is the set of ordered pairs

$$A \times B = \{(a, b) \mid a \in A, b \in B\}.$$

The projections are the following two maps

$$A \xleftarrow{\pi_1} A \times B \xrightarrow{\pi_2} B$$

defined as

$$a \xleftarrow{\pi_1} (a, b) \xrightarrow{\pi_2} b$$

Moreover, given any pair of functions $f : X \to A$ and $g : X \to B$, there is a unique $h : X \to A \times B$ that makes the required diagram commute. It is given by

$$h(x) = (f(x), g(x)).$$

Note that if we choose a different definition of ordered pairs, we get different sets. They will, of course, be isomorphic.

2. Products of "structured sets" like monoids and groups can *often* be constructed as products of the underlying sets with *componentwise* operations: If G and H are groups, for instance, $G \times H$ can be constructed by taking the underlying set of $G \times H$ to be the set $\{\langle g, h \mid g \in G, h \in H\rangle\}$ and defining the binary operation by

$$\langle g, h\rangle \cdot \langle g', h'\rangle = \langle g \cdot g', h \cdot h'\rangle$$

the unit by

$$u = \langle u_G, u_H\rangle$$

and inverses by

$$\langle a, b\rangle^{-1} = \langle a^{-1}, b^{-1}\rangle.$$

The projection homomorphisms $G \times H \to G$ (or H) are the evident one $\langle g, h\rangle \mapsto g$ (or h).

Note that this need not always work in all categories with "structured sets." For example, the cartesian product of two fields with componentwise operations is not a field. (Not just "not necessarily," it's *never* a field.)

3. For categories $\mathbf{C}$ and $\mathbf{D}$, we already defined the category of objects and arrows,

$$\mathbf{C} \times \mathbf{D}.$$

Together with the evident projection functors, this is indeed a product in Cat (when $\mathbf{C}$ and $\mathbf{D}$ are "small").

As special cases, we also get the products of posets and of monoids as product of categories. For them to indeed be the products in Pos and Mon, one has to check that the product category does indeed take the form of a poset-category or a monoid-category. Moreover, the projections and the unique paired functions have to be checked to be monotone/monoid homomorphisms.

4. Let P be a poset (considered as a category) and consider a product of elements $p, q \in P$. We must have projections

$$p \times q \leq p,$$
$$p \times q \leq q,$$

and if for any element x, we have

$$x \leq p, \quad \text{and} \quad x \leq q,$$

then we need

$$x \leq p \times q.$$

The above operation $\times$ can be recognised as the "meet" operation. $p \times q$ is just what is usually the *greatest lower bound* of p and q. In more familiar notation, we have $p \times q = p \wedge q$.

5. One can show that the product of topological spaces, as usually defined, is indeed the product in Top.

§§2.6 Categories with products

Let $\mathbf{C}$ be a category that has a product diagram for every pair of objects. Suppose we have objects and arrows

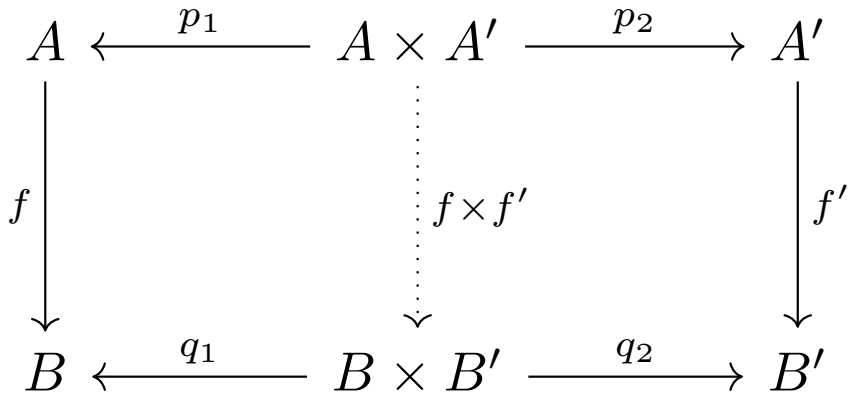
$$\begin{array}{ccccc} A & \xleftarrow{p_1} & A \times A' & \xrightarrow{p_2} & A' \\ \downarrow{\scriptstyle f} & & & & \downarrow{\scriptstyle f'} \\ B & \xleftarrow{q_1} & B \times B' & \xrightarrow{q_2} & B' \end{array}$$

with indicated products. Then, we write

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

for the arrow $f \times f'$ obtained by the UMP of $B \times B'$ and the arrows $f \circ p_1 : A \times A' \to B$ and $f \circ p_2 : A \times A' \to B$. That is, the unique $f \times f'$ making both squares in the

following diagram commute:



In this way, if we choose a product for each pair of objects, we get a functor

$$\times : \mathbf{C} \times \mathbf{C} \to \mathbf{C}.$$

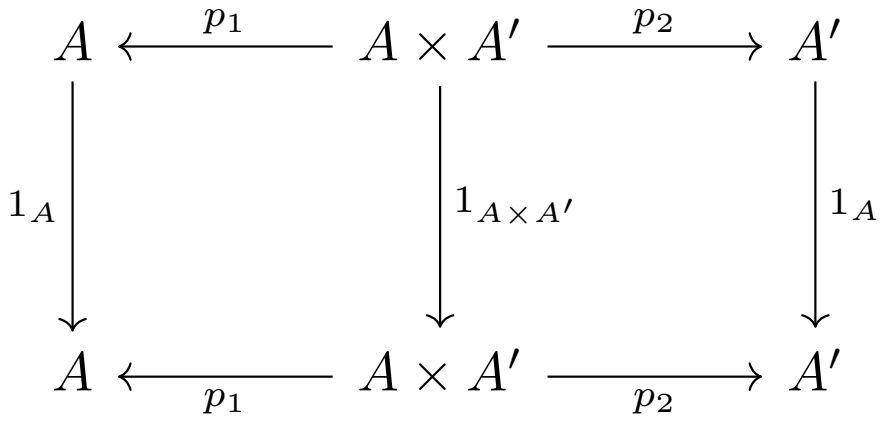
This is defined on objects by sending (A, A') to $A \times A'$ and $(f, f') : (A, A') \to (B, B')$ to $f \times f' : A \times A' \to B \times B'$.

Let us show that this is indeed a functor.

The domain and codomain is clearly preserved, by construction.

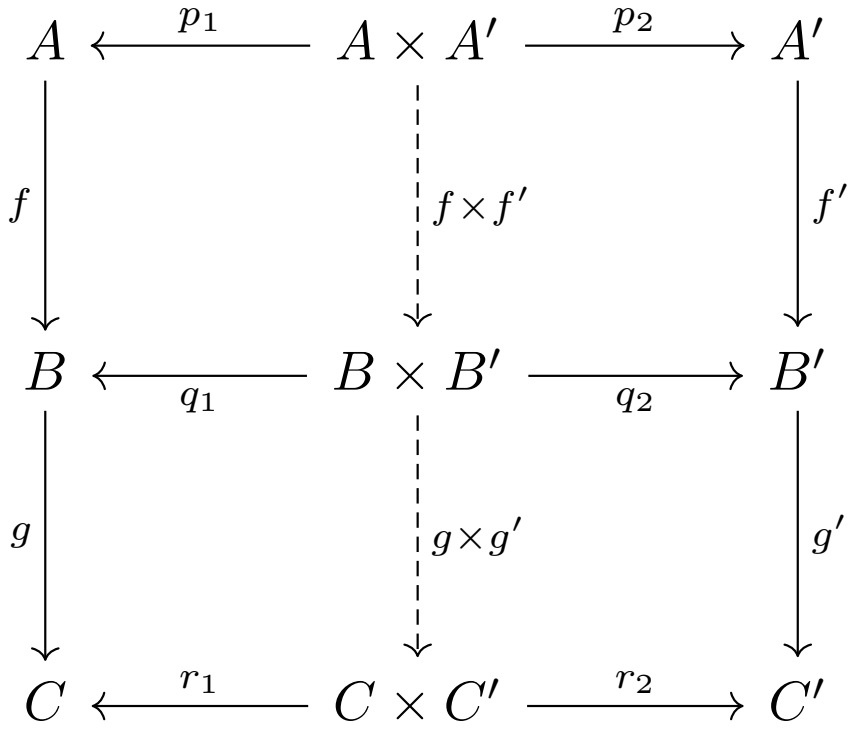
Let us show that it preserves units. Let $(A, A') \in \mathbf{C} \times \mathbf{C}$. We wish to show that $1_A \times 1_{A'} : A \times A' \to A \times A'$ is the identity arrow. For this, we note that the following

diagram commutes (why?):



By the uniqueness clause of the UMP, we get that $1_A \times 1_{A'} = 1_{A \times A'}$.
To show that this map preserves preserves composition, one may observe diagram (2.5) and use the UMP of $C \times C'$.

(2.5)



One may now define a ternary product

$$A_1 \times A_2 \times A_3$$

with an analogous UMP: there are three projections $p_i : A_1 \times A_2 \times A_3 \to A_i$, and for any object X and three arrows $x_i : X \to A_i$, there is a unique arrow $u : X \to$

$A_1 \times A_2 \times A_3$ such that $p_i u = x_i$ for $i = 1, 2, 3$. Clearly, such a condition can be formulated for any (finite?) number of factors.
It is clear that if a category has binary products, then it has all finite products with two or more factors; for instance, one could set

$$A \times B \times C = (A \times B) \times C$$

with the appropriate projections. On the other hand, one could have taken $A \times (B \times C)$ as well. It can be seen that both of these satisfy the ternary UMP. However, by the ternary UMP, we also then have

$$(A \times B) \times C \cong A \times (B \times C),$$

id est, the binary product operation is associative up to isomorphism.
Observe that a terminal object is a "nullary" product, a product of zero objects:
Given no objects, there is an object 1 with no maps, and given any other object X and no maps, there is a unique arrow

$$! : X \to 1$$

making nothing further commute.
Similarly, any object A is the *unary product* of A with itself one time.

Finally, one may also define the product of a family of objects $(C_i)_{i\in I}$ indexed by *any* set I in the following manner:
The product $(C_i)_{i\in I}$ is an object P along with a family $(p_i)_{i\in I}$ of arrows

$$p_i : P \to C_i$$

satisfying the following UMP:
Given any object X and family $(x_i)_{i\in I}$ of arrows

$$x_i : X \to C_i,$$

there exists a unique $u : X \to P$ such that $p_i u = x_i$ for every $i \in I$.

Definition 2.25. A category $\mathbf{C}$ is said to *have all finite products*, if it has a terminal object and all binary products (and therewith products of any finite cardinality). The category $\mathbf{C}$ *has all products* if every set of objects in $\mathbf{C}$ has a product.

§§2.7 Hom-sets

In this (sub)section, we assume that all categories are locally small, that is given any two objects in this category, the collection of morphisms from one to the other is in fact

a set.

Recall that in any category $\mathbf{C}$, given any objects A and B, we write

$$\mathrm{Hom}(A, B) = \{f \in \mathbf{C} \mid f : A \to B\}$$

and call such a set of arrows a *Hom-set*. Clearly, any element in the above set is an arrow with domain A and codomain B. Thus, we may compose it with an arrow $g : B \to B'$ to get a an arrow $g \circ f : A \to B'$, and arrow with domain A and codomain B'. This is just an elaborate of saying that any arrow $g : B \to B'$ induces a function

$$\mathrm{Hom}(A, g) : \mathrm{Hom}(A, B) \to \mathrm{Hom}(A, B')$$

defined in the above manner, id est,

$$(f : A \to B) \mapsto (g \circ f : A \to B').$$

(Note very carefully that the above sends an arrow to another arrow.)

Thus, we now have seen two things of the form $\mathrm{Hom}(A, -)$; when $-$ is an object B, we get a *set* $\mathrm{Hom}(A, B)$, an object in Set, when $-$ is an arrow $g : B \to B'$, we get a *function* $\mathrm{Hom}(A, g)$, an arrow in Set. Moreover, it's a function from $\mathrm{Hom}(A, \mathrm{dom}\, g)$ to $\mathrm{Hom}(A, \mathrm{cod}\, g)$. The attentive reader should now see what this is screaming - the

creation of a functor!
Indeed, now we show that

$$\mathrm{Hom}(A, -) : \mathbf{C} \to \mathsf{Set}$$

is a functor. (The definition of this map is precisely what was written above.)
That it preserves domain and codomain follows from construction. We now show that it preserves units and compositions.
Units:
Let $B \in \mathbf{C}$ be an object and 1_B its identity arrow. We show that $\mathrm{Hom}(A, 1_B) : \mathrm{Hom}(A, B) \to \mathrm{Hom}(A, B)$ is the identity map $1_{\mathrm{Hom}(A,B)}$.
This is direct; indeed, let $f \in \mathrm{Hom}(A, B)$. Then,

$$\begin{aligned} \mathrm{Hom}(A, 1_B)(f) &= 1_B \circ f \\ &= f. \end{aligned}$$

Compositions:
Let $g : B \to C$ and $h : C \to D$ be arrows in $\mathbf{C}$. We show that

$$\mathrm{Hom}(A, h \circ g) = \mathrm{Hom}(A, h) \circ \mathrm{Hom}(A, g).$$

Note that both sides are functions from $\mathrm{Hom}(A, B)$ to $\mathrm{Hom}(A, C)$. Let $f \in \mathrm{Hom}(A, B)$. Then,

$$\begin{aligned}
\mathrm{Hom}(A, h \circ g)(f) &= (h \circ g) \circ f \\
&= h \circ (g \circ f) \\
&= h \circ (\mathrm{Hom}(A, g)(f)) \\
&= \mathrm{Hom}(A, h)(\mathrm{Hom}(A, g)(f)) \\
&= (\mathrm{Hom}(A, h) \circ \mathrm{Hom}(A, b))(f).
\end{aligned}$$

This completes the proof.

Thus, $\mathrm{Hom}(A, -)$ is indeed a functor. We shall this the (covariant) *representable functor* of A. We will study such representable functors later. For now, we see how one can use Hom $-$sets to give another formulation of the definition of products.

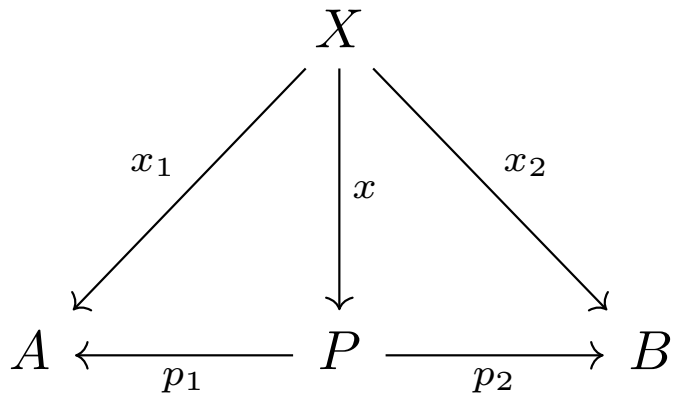
For *any* object P, a pair of arrows of arrows $p_1 : P \to A$ and $p_2 : P \to B$ determine an element of the set

$$\mathrm{Hom}(P, A) \times \mathrm{Hom}(P, B).$$

Now, given any arrow

$$x : X \to P$$

composing with p_1 and p_2 gives a pair of arrows $x_1 = p_1 \circ x : X \to A$ and $x_2 = p_2 \circ x : X \to B$, as indicated in the following diagram:



(We emphasise that in the above discussion, P is *any* object. Not necessarily a product.)

In this way, we have a function

$$\vartheta_X = (\mathrm{Hom}(X, p_1), \mathrm{Hom}(X, p_2)) : \mathrm{Hom}(X, P) \to \mathrm{Hom}(X, A) \times \mathrm{Hom}(X, B)$$

defined by

$$\vartheta_X(x) = (x_1, x_2). \tag{2.6}$$

Note very carefully that the above function depends on P, p_1, p_2.

Using this function, we can express the condition of being a product concisely as follows.

Proposition 2.26. A diagram of the form

$$A \xleftarrow{p_1} P \xrightarrow{p_2} B$$

is a product for A and B iff for every object X, the canonical function ϑ_X given in (2.6) is an isomorphism.

Proof. The proof follows from observing that the above condition is just a rephrasing of the UMP of the product. Let $(x_1, x_2) \in \mathrm{Hom}(X, A) \times \mathrm{Hom}(X, B)$ be arbitrary. We recall from Remark 2.23, the two conditions of product:

- *Existence*: There is some $x : X \to P$, id est, $x \in \mathrm{Hom}(X, P)$ such that $x_1 = p_1 x$ and $x_2 = p_2 x$. This is iff ϑ_X is surjective.
- *Uniqueness*: Given any $v : X \to P$, if $p_1 v = x_1$ and $p_2 v = x_2$, then $v = x$. This is iff ϑ_X is injective.

The result follows. (Recall that an isomorphism in Set is the same as a bijection.) □

Definition 2.27. Let $\mathbf{C}, \mathbf{D}$ be categories with binary products. A function $F : \mathbf{C} \to \mathbf{D}$ is said to *preserve binary products* if it takes every product

$$A \xleftarrow{p_1} A \times B \xrightarrow{p_2} B \quad \text{in } \mathbf{C}$$

to a product diagram

$$FA \xleftarrow{p_1} F(A \times B) \xrightarrow{p_2} FB \quad \text{in } \mathbf{D}. \tag{2.7}$$

Note that in the above, it is not sufficient that $F(A \times B)$ is isomorphic to $FA \times FB$. We also require Fp_1 and Fp_2 to act like the arrows that give us the UMP. Thus, we must have an isomorphism that is "good enough".

Let us see more elaborately when the above is indeed a product diagram in $\mathbf{D}$. As $\mathbf{D}$ has binary products, consider the product $FA \times FB$ with the projections q_1 and q_2 to FA and FB, respectively.

Consider the unique arrow $i : F(A \times B) \to FA \times FB$ obtained via the UMP of $FA \times FB$ using the arrows Fp_1 and Fp_2. Then, (2.7) is a product diagram iff i is an isomorphism.

A rough sketch of the proof is as follows: For the forward direction, it will follow from the UMP as in the proof of Proposition 2.24.

For the reverse direction, let X be an arbitrary object with arrows into A and B, use i^{-1} and UMP of $FA \times FB$ to get the arrow $x : X \to F(A \times B)$ as desired by the definition of product.

To summarise the discussion, I shall write what the book writes concisely as:
It follows that F preserves products just if

$$F(A \times B) \cong FA \times FB$$

"canonically," that is, iff the canonical "comparison arrow"

$$\langle Fp_1, Fp_2 \rangle : F(A \times B) \to FA \times FB$$

in $\mathbf{D}$ is an isomorphism.
(Recall the definition of $\langle Fp_1, Fp_2 \rangle$ from (2.4).)
For example, the forgetful functor $U : \mathsf{Mon} \to \mathsf{Set}$ preserves products.

Corollary 2.28. For any object X in a category $\mathbf{C}$ with products, the (covariant) representable functor

$$\mathrm{Hom}_{\mathbf{C}}(X, -) : \mathbf{C} \to \mathsf{Set}$$

preserves products.

Proof. By our above observation, we want to show that:
for any $A, B \in \mathbf{C}$, $\mathrm{Hom}(X, A \times B)$ and $\mathrm{Hom}(X, A) \times \mathrm{Hom}(X, B)$ are canonically isomorphic (in Set).

By Proposition 2.26, we precisely have that there is a canonical isomorphism

$$\mathrm{Hom}(X, A \times B) \cong \mathrm{Hom}(X, A) \times \mathrm{Hom}(X, B).$$

□

§3 Acknowledgments

Here's a list of the people who have helped me make the notes better. I'm thankful to them. The count after the name denotes the number of changes made due to their suggestions - these include both typos and pointing out places where the phrasing could be improved.
The names are listed in chronological order based on the first suggestion.

1. Ishan Kapnadak: 1
2. Amit Rajaraman: 11