Category Theory

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

March 27, 2020

CONTENTS

§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.

§§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	Fields extensions
$\mathsf{Vec}_{\mathbb{k}}$	Vector spaces over $\mathbb{k}$	Linear maps
Pos	Posets	Order-preserving maps

4. Preoders. 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. We already have some examples of these from the previous case of posets. Let's look at some of them in particular.

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

  $$*$$

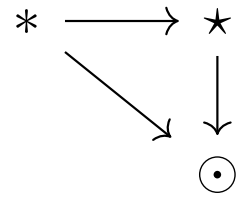
  It has one object and its identity arrow, which we do not draw.

- 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

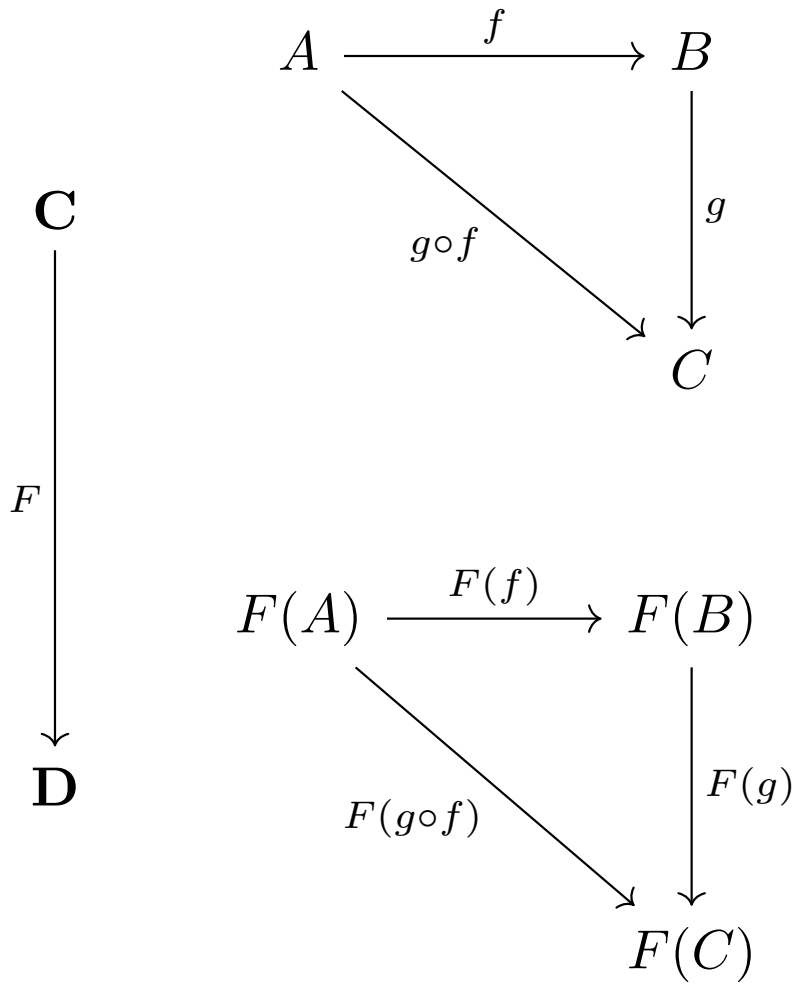
Definition 1.2. A functor

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

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 bijective (appropriate) homomorphisms.

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 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 does ***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.

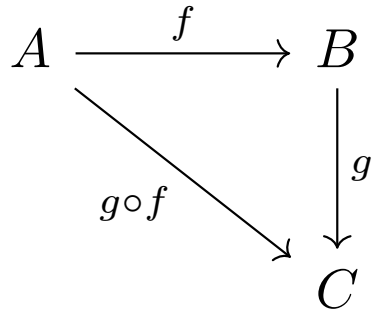
2. The *opposite* (or "dual") category $\mathbf{C}^{\mathrm{op}}$ of a category $\mathbf{C}$ has the same objects and arrow $f : C \to D$ in $\mathbf{C}^{\mathrm{op}}$ is an arrow $f : D \to C$ in $\mathbf{C}$. That is, $\mathbf{C}^{\mathrm{op}}$ is just $\mathbf{C}$ with all of the arrows formally turned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

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

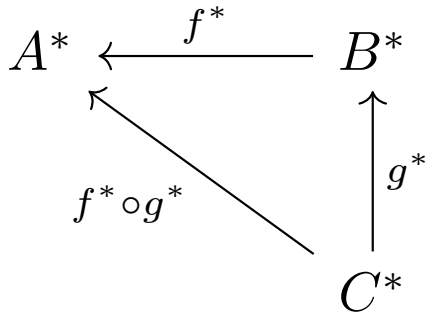
   in $\mathbf{C}^{\mathrm{op}}$ for $f : C \to D$ in $\mathbf{C}$. With this notation, we have the units and composition rule as

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

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"

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

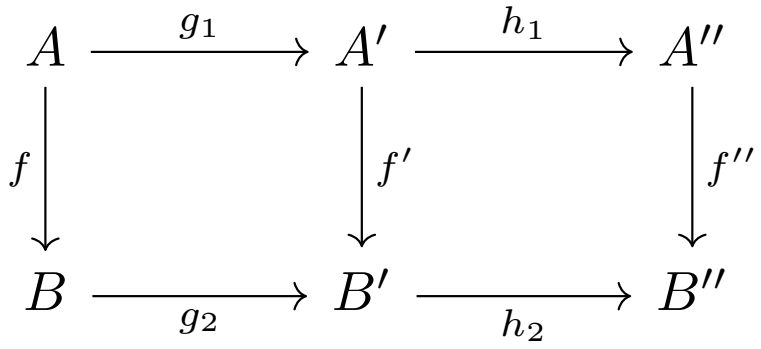
where g_1 and g_2 are arrows in $\mathbf{C}$. That is, such an arrow is a pair of arrows $g = (g_1, g_2)$ in $\mathbf{C}$ such that

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

(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 component-wise:

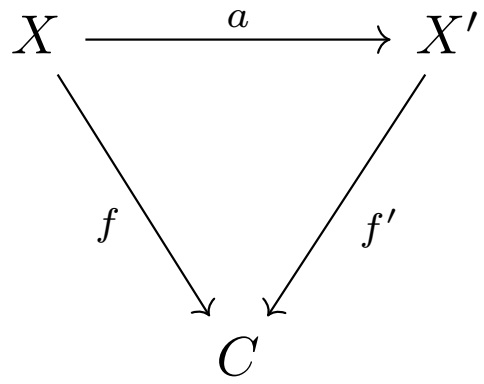
$$(h_1, h_2) \circ (g_1, g_2) = (h_1 \circ g_1, h_2 \circ g_2).$$

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

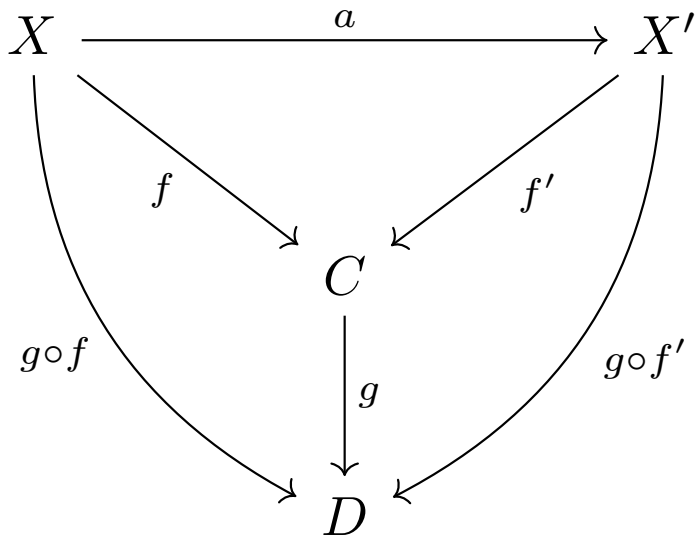
   - Objects: all arrows $f \in \mathbf{C}$ such that $\mathrm{cod}(f) = C$,
   - Arrows: an arrow a from $f : X \to C$ to $f' : X' \to C$ is an arrow $a : X \to X'$ in $\mathbf{C}$ such that $f' \circ a = f$, as indicated in



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

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

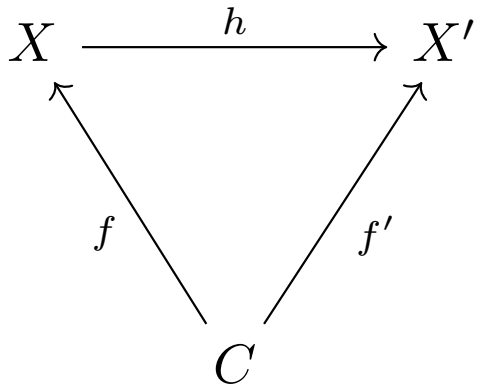
to be a functor. If $\mathbf{C} = \mathbf{P}$ is a poset category and $p \in \mathbf{P}$, then

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

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

$h \cdot f = f'$.

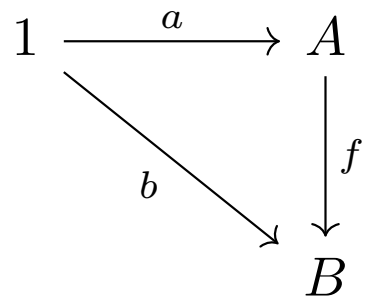


Example 1.7. The category Set_* of *pointed sets* consists of sets A with distinguished elements $a \in A$, and arrows $f : (A, a) \to (B, b)$ are functions $f : A \to B$ that preserves the "points," $f(a) = b$. This is isomorphic to the coslice category,

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

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 :

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

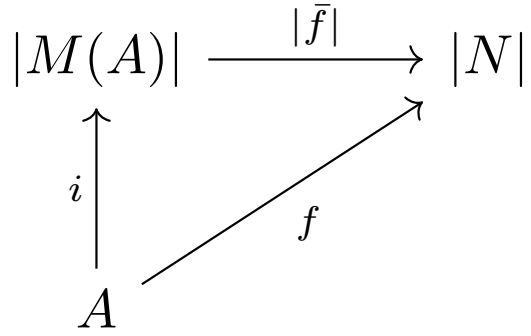
2. No "nontrivial" relations hold in M, that is, if $a_1 \cdots a_j = a'_1 \cdots a'_k$, then this is required by the axioms for monoids.

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.

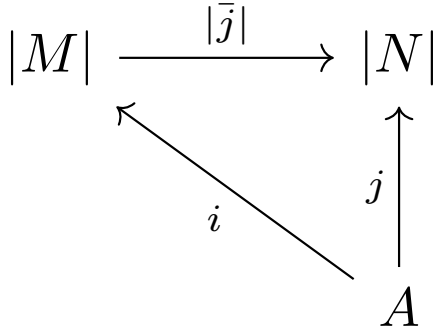
Proposition 1.9. Given monoids M and N with functions $i : A \to |M|$ and $j : A \to |N|$, each with the UMP of the free monoid of A, there is a (unique) monoid isomorphism $h : M \to N$ such that $|h|i = j$ and $|h^{-1}|j = i$.

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:

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

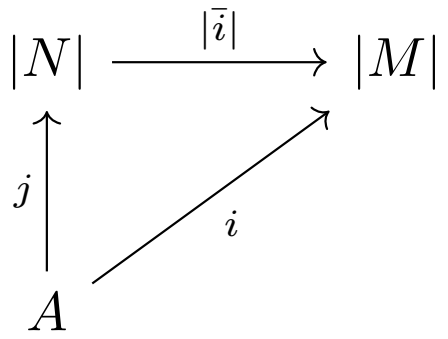
in Mon:



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:

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

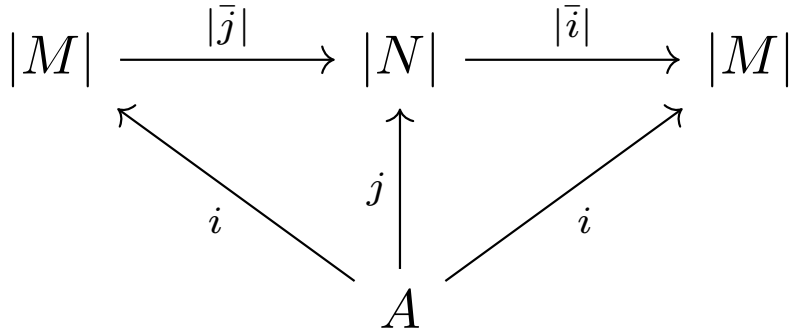
in Mon:



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

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

in Mon:

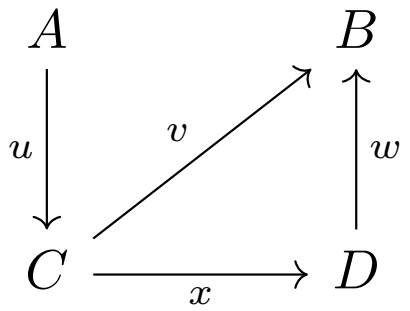


Now, look at the monoid homomorphism $\overline{i} \circ \overline{j} : M \to M$.
It has the property that $|\overline{i} \circ \overline{j}|i = i$. As $1_M : M \to M$ has this property, by the

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:

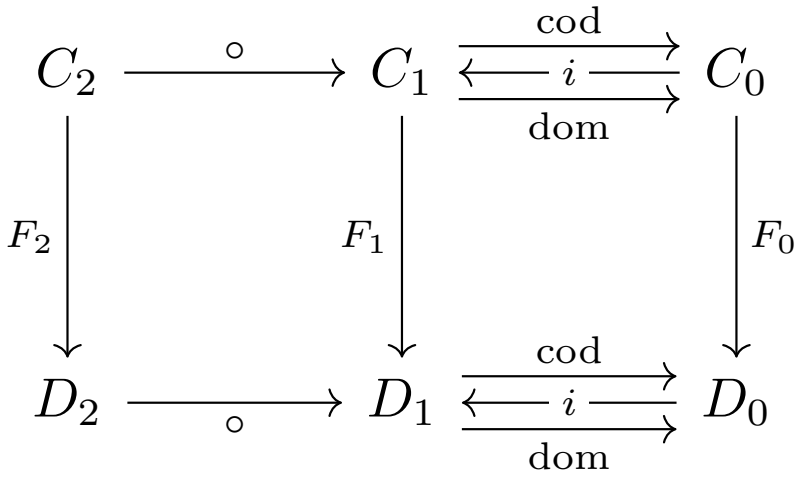
$$C_2 \xrightarrow{\quad\circ\quad} C_1 \underset{\xleftarrow{\quad i \quad}}{\overset{\overset{\mathrm{cod}}{\longrightarrow}}{\underset{\mathrm{dom}}{\longrightarrow}}} C_0$$

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

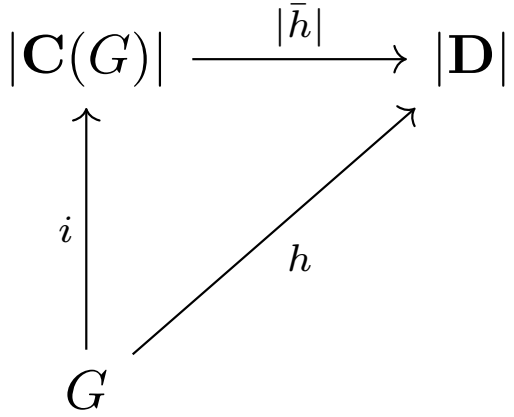
$$C_1 \underset{\text{dom}}{\overset{\text{cod}}{\rightrightarrows}} C_0.$$

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 $\mathbf{Graph}$:



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 $M(1)$.