Introduction to Category Theory I
Categories and Functors

Bart Jacobs

iHub, Radboud University, Nijmegen, The Netherlands
bart@cs.ru.nl

February 12, 2025

Abstract. These notes introduce the notion of category, as collection of objects with arrows between them. Several basic examples are described. Also, the notion of functor is introduced, as an arrow between categories.

1 Introductory remarks

The common notions of mathematics, like function, relation, order etc. can all be formulated within set theory. The latter forms a basis for mathematics in which the $\in$-relation of inhabitation is taken as fundamental. Proof are done by reasoning with elements, that is, by "opening boxes".

Category theory provides an alternative foundational language. It is not based on elements, but on 'morphisms' (also called 'maps' or 'arrows'). In contrast to set theoretic proofs, categorical proofs don't involve any opening of boxes. One describes categorical structures and reasons about them solely by considering the morphisms going in and going out. In terminology from computer science: proofs proceed from specification, and not from implementation.

The best (actually, only) way to learn about category theory is to do many exercises. Proofs are not difficult, but require a principled way of thinking, keeping in mind in which universe one is working, and thus which definitions apply. At various places this text contains suggestions to the reader to check certain properties. It is recommended to do so.

2 Categories

A category consists of a collection of abstract arrows for which one has composition and identities. These arrows are between what are called objects. In this section we give the basic definition and consider a number of examples of categories. We often use open face capitals $\mathbb{A}$, $\mathbb{B}$, $\mathbb{C}$ for categories, where $\mathbb{C}$ should not be confused with the complex numbers.

Definition 1. *A* **category** $\mathbb{C}$ *consists of two collections*

$$\begin{array}{ll} \mathit{Obj}(\mathbb{C}) & \textit{the objects of } \mathbb{C} \\ \mathit{Arr}(\mathbb{C}) & \textit{the arrows (or morphisms or maps) of } \mathbb{C}. \end{array}$$

A morphism always has a domain and a codomain, which are objects. One writes

$$f\colon X \to Y \qquad \textit{or} \qquad X \xrightarrow{f} Y \qquad \textit{or} \qquad f \in \mathbb{C}(X,Y)$$

if X and Y are domain and codomain objects of the morphism f. We usually write $X \in \mathbb{C}$ for $X \in Obj(\mathbb{C})$. The collection $\mathbb{C}(X,Y)$ of maps from X to Y is called a hom-set.

As part of the structure of a category one requires that for each object X there is an **identity map***,*

$$id_X\colon X \to X$$

And for morphisms,

$$f\colon X \to Y \qquad \textit{and} \qquad g\colon Y \to Z$$

there is a **composite map***,*

$$g \circ f\colon X \to Z$$

One requires

$$f \circ id_X = f \qquad \textit{and} \qquad id_Y \circ f = f$$

for $f\colon X \to Y$, and

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

for $f\colon X \to Y$, $g\colon Y \to Z$ and $h\colon Z \to W$.

Equations between composites of arrows are often conveniently expressed by writing diagrams. For example, the identity laws above can be given by

$$\begin{tikzcd} X \arrow[r, "id_X"] \arrow[dr, "f"'] & X \arrow[d, "f"] \\ & Y \end{tikzcd} \qquad \begin{tikzcd} X \arrow[d, "f"'] \arrow[dr, "f"] & \\ Y \arrow[r, "id_Y"'] & Y \end{tikzcd}$$

Or, in a single diagram,

$$\begin{tikzcd} X \arrow[r, "id_X"] \arrow[dr, "f"'] & X \arrow[d, "f"] \arrow[dr, "f"] & \\ & Y \arrow[r, "id_Y"'] & Y \end{tikzcd}$$

The general convention is that all diagrams commute. This means that composites of all paths between two objects in a diagram are equal. For example the associativity of composition $\circ$ in the above definition may be expressed as

$$\begin{tikzcd} X \arrow[r, "f"] \arrow[d, "g\circ f"'] & Y \arrow[d, "h\circ g"] \\ W \arrow[r, "h"'] & Z \end{tikzcd}$$

Establishing the equality of two composites in such a diagram by following paths is called **diagram chasing**. It is very elementary but important in category theory. Proofs often consist of certain diagram chases. It is something one has to learn right in the beginning. But it's best to learn this by seeing how the relevant diagrams arise, and that's a bit hard to convey in a (static) text. In general, it is better to write diagrams then equations. This will become clear later on when we consider universal properties. Computer scientists tend to prefer equations, probably because these are easier to manipulate mechanically. But writing equations only, destroys much of category theory.

The convention that all such composites are equal is not absolute. Sometimes we wish to write

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

without assuming that f and g are equal. If they were, we would write a single arrow.

The most obvious example of a category is the category $\mathbf{Sets}$ of sets and functions. It has

objects	sets X
morphisms	$f\colon X \to Y$ in $\mathbf{Sets}$ are ordinary functions f from X to Y.

Composition in this category is just function composition: for $X \xrightarrow{f} Y \xrightarrow{g} Z$ one has $g \circ f\colon X \to Z$ given by $(g \circ f)(x) = g(f(x))$. And the identity arrow $id_X\colon X \to X$ is the function with $(id_X)(x) = x$.

Notice that the domain and codomain form part of a morphism. So the assignments $n \mapsto n^2$, considered as functions $\mathbb{N} \to \mathbb{N}$ or as $\mathbb{N} \to \mathbb{R}$ are different morphisms in the category $\mathbf{Sets}$.

Let's consider some variations.

The category $\mathbf{PFun}$ is the category of sets and partial functions. It has

objects	sets X
morphisms	$f\colon X \to Y$ in $\mathbf{PFun}$ are partial functions; they take an element $x \in X$ and possibly assign a value $f(x) \in Y$. If it does, one writes $f(x)\downarrow$ and says '$f(x)$ is defined' or 'f converges at x'. Otherwise, if $f(x)$ is not defined, one writes $f(x)\uparrow$ or also $f(x) = \uparrow$.

Notice again that the domain X and codomain Y form part of a map $f\colon X \to Y$ in $\mathbf{PFun}$. Composition of $f\colon X \to Y$ and $g\colon Y \to Z$ is the usual composition of partial functions:

$$(g \circ f)(x) = \begin{cases} g(f(x)) & \text{if } f(x)\downarrow \text{ and } g(f(x))\downarrow \\ \uparrow & \text{otherwise} \end{cases}$$

Thus if $f, g\colon \mathbb{N} \rightrightarrows \mathbb{N}$ are

$$f(n) = \begin{cases} n & \text{if } n \text{ even} \\ \uparrow & \text{otherwise} \end{cases} \qquad \text{and} \qquad g(n) = 1$$

Then $g \circ f$ only converges at even numbers.

Identities in $\mathbf{PFun}$ are the usual identity functions. Associativity of composition is left as an exercise.

Here is another variation, the category $\mathbf{Sets}_\perp$ of pointed sets. It also contains partial functions as morphisms, but defined in a slightly different manner. Later on we shall be able to express that the categories $\mathbf{PFun}$ and $\mathbf{Sets}_\perp$ are essentially the same, by saying that they are isomorphic as categories.

The category $\mathbf{Sets}_\perp$ of pointed sets has

objects	sets X which have a distinguished element (or base point) $\perp_X \in X$
morphisms	$f\colon X \to Y$ in $\mathbf{Sets}_\perp$ are functions $f\colon X \to Y$ which preserve base points i.e. $f(\perp_X) = \perp_Y$.

Composition and identities are as in $\mathbf{Sets}$. Check for yourself that the category axioms hold.

We have seen partial computations form (morphisms in) a category. There is also a category $\mathbf{NonDet}$ whose morphisms capture non-deterministic computation.

objects	sets X
morphisms	$f\colon X \to Y$ in $\mathbf{NonDet}$ are functions $f\colon X \to \mathcal{P}(Y)$ producing for each input $x \in X$ a subset $f(x) \subseteq Y$ of outputs.

The identity $\mathit{id}_X\colon X \to X$ is given by singletons: $\mathit{id}_X(x) = \{x\}$. Composition of $f\colon X \to Y$ and $g\colon Y \to Z$ in $\mathbf{NonDet}$, that is, of $f\colon X \to \mathcal{P}(Y)$ and $g\colon Y \to \mathcal{P}(Z)$ is the function $g \circ f\colon X \to \mathcal{P}(Z)$ given by:

$$(g \circ f)(x) \coloneqq \{z \in Z \mid \exists y \in Y.\, y \in f(x) \text{ and } z \in g(y)\}.$$

Check yourself that this indeed forms a category!

In the above three examples objects are certain sets and morphisms are certain functions. This is by no means typical. In fact, for a novice to this subject it usually takes some time *not* to think of an arrow $f\colon X \to Y$ in an arbitrary category as a set theoretic function. The following three examples may help to get rid of this idea.

There is a category $\mathbf{Rel}$ of sets with relations as morphisms between them.

objects	sets X
morphisms	$R\colon X \to Y$ are relations $R \subseteq X \times Y$. One usually writes xRy for $\langle x, y\rangle \in R$.

As identity $\mathit{id}_X\colon X \to X$ morphism we take the relation

$$x(\mathit{id}_X)y \overset{\text{def}}{\Longleftrightarrow} x = y$$

And the composition of $R\colon X \to Y$ and $S\colon Y \to Z$ is the relation

$$x(S \circ R)z \overset{\text{def}}{\Longleftrightarrow} \exists y \in Y.\, xRy \text{ and } ySz.$$

Check the category axioms! Soon we can say that the categories $\mathbf{NonDet}$ and $\mathbf{Rel}$ are also isomorphic.

In this example $\mathbf{Rel}$ the objects are still sets. Next we'll have quite different objects. Recall that a **preorder** is a set X with a relation $\sqsubseteq$ on X which is

$$\begin{aligned} &\text{reflexive: } x \sqsubseteq x \\ &\text{transitive: } x \sqsubseteq y \text{ and } y \sqsubseteq z \implies x \sqsubseteq z. \end{aligned}$$

And $(X, \sqsubseteq)$ is a **poset** (partially ordered set) if $\sqsubseteq$ is additionally

$$\text{antisymmetric: } x \sqsubseteq y \text{ and } y \sqsubseteq x \implies x = y.$$

For any set X, the powerset $\mathcal{P}(X)$ with inclusion $\subseteq$ as order is a poset. To see a natural example of a preorder, consider the set $\mathbb{Z} \times \mathbb{N}$ of pairs of an integer and a natural number, and think of (n_1, n_2) as a representation for the rational $\frac{n_1}{n_2}$. Put an order $\preceq$ on $\mathbb{Z} \times \mathbb{N}$ by

$$(n_1, n_2) \preceq (m_1, m_2) \overset{\text{def}}{\iff} n_1 \cdot m_2 \leq m_1 \cdot n_2$$

where $\cdot$ and $\leq$ on the right are the usual multiplication and order. One gets a preorder $\preceq$. (One can turn $\preceq$ into a partial order by taking suitable equivalence classes. Then one gets the set $\mathbb{Q}$ of rational numbers, with usual partial order.)

Each preorder $(X, \sqsubseteq)$ determines a category with

objects	elements $x \in X$
morphisms	$x \to y$ exist if and only if $x \sqsubseteq y$. Further we say there is at most one arrow between any two objects. So we don't bother to give names to arrows in this category.

One indeed gets a category in this way: there are identity maps $x \to x$ since one has $x \sqsubseteq x$ by reflexivity. And if there are maps $x \to y$ and $y \to z$, then $x \sqsubseteq y$ and $y \sqsubseteq z$ so that $x \sqsubseteq z$ by transitivity; this yields $x \to z$. The identity and associativity requirements hold trivially (because there is at most one map between two objects).

A category $\mathbb{C}$ in which between any two objects there is at most one arrow may be identified with a preorder. On can order the objects of $\mathbb{C}$ by

$$X \sqsubseteq Y \iff \text{there is a map } X \to Y \text{ in } \mathbb{C}$$
$$(\text{i.e. } \mathbb{C}(X, Y) \neq \emptyset)$$

The fact that one has identities and composition in $\mathbb{C}$ yields that $\sqsubseteq$ forms a preorder on *Obj*$(\mathbb{C})$.

Thus, preorders correspond to certain degenerate categories.

A **monoid** is a set M together with an associative binary operation $\cdot\colon M \times M \to M$ for which one has a two-sided neutral (unit) element $u \in M$. Thus

$$x \cdot (y \cdot z) = (x \cdot y) \cdot z, \qquad u \cdot x = x, \qquad x \cdot u = x$$

for each $x, y, z \in M$. The monoid is called **commutative** if $x \cdot y = y \cdot x$ for all $x, y \in M$. Examples of commutative monoids are natural numbers with addition or multiplication, i.e. $(\mathbb{N}, +, 0)$ and $(\mathbb{N}, \cdot, 1)$.

An important example of a non-commutative monoid is the set $\mathcal{L}(X)$ of all finite lists / sequences of elements of X. Thus, elements of $\mathcal{L}(X)$ are lists $\langle x_1, \ldots, x_n \rangle$ of elements $x_i \in X$, including the empty list $\langle\rangle \in \mathcal{L}(X)$. This forms a monoid via concatenation $+\!\!+$ defined via juxtaposition:

$$\langle x_1, \ldots, x_n \rangle +\!\!+ \langle y_1, \ldots, y_m \rangle \coloneqq \langle x_1, \ldots, x_n, y_1, \ldots, y_m \rangle.$$

The empty sequence $\langle\rangle$ is the neutral element. Check for yourself that $(\mathcal{L}(X), +\!\!+, \langle\rangle)$ is a monoid and that it is not commutative. This $\mathcal{L}(X)$ is also called the **Kleene-star** $X^\star$ of X, or **free monoid** on X.

If every element x in a monoid has an inverse x^{-1} with respect to the multiplication of the monoid, then one has a **group**. Examples are $(\mathbb{Z}, +, 0, -(\cdot))$ or $(\mathbb{Q}-\{0\}, \cdot, 1, \frac{1}{(\cdot)})$.

Each monoid M determines a category with

objects	just a single object, say $*$
morphisms	$* \to *$ are elements $x \in M$.

One gets a category with the unit element $u \colon * \to *$ as identity map and with monoid composition $x \cdot y \colon * \to *$ as composite of morphisms $x \colon * \to *$ and $y \colon * \to *$.

Conversely, every category $\mathbb{C}$ in which there is only one object may be identified with a monoid $(\mathit{Arr}(\mathbb{C}), \circ, id)$. Thus monoids also correspond to certain degenerate categories.

We'll see many more examples of categories in due course. Often we'll have categories of "mathematical structures" and "structure preserving maps". As typical examples one has a category $\mathbf{Mon}$ of monoids and mononid homomorphisms. Explicitly, the category $\mathbf{Mon}$ has

objects monoids $(M, \cdot, u)$

morphisms $(M, \cdot, u) \longrightarrow (N, \bullet, v)$ are functions $f \colon M \to N$ between the underlying sets satisfying

$$f(x \cdot y) = f(x) \bullet f(y) \qquad \text{and} \qquad f(u) = v.$$

Similarly, there is a category $\mathbf{Sp}$ of topological spaces $X = (X, \mathcal{O}(X))$ and continuous functions (i.e. $f \colon X \to Y$ in $\mathbf{Sp}$ if for each open $o \in \mathcal{O}(X)$ one has $f^{-1}(o) \in \mathcal{O}(Y)$).

Also there are categories $\mathbf{PreOrd}$ (and $\mathbf{PoSets}$) of preorders (and posets) with monotone functions. That is, $f \colon (X, \sqsubseteq) \to (Y, \preceq)$ if $f \colon X \to Y$ is a function between the underlying sets which satisfies $x \sqsubseteq x' \implies f(x) \preceq f(x')$.

Actually, it is good mathematical practice everytime one sees a new mathematical structure, to ask oneself what are the corresponding structure preserving mappings. Then one can form a category with these structures as objects and with appropriate morphisms between them. Sometimes, for more complicated mathematical structures (like

C^*-algebras) there is a choice of morphism, but usually there is a canonical concept of a structure preserving map.

We shall be using the following notational conventions. Arbitrary categories will be written as $\mathbb{A}, \mathbb{B}, \mathbb{C}, \mathbb{D}, \ldots$ in open face, but specific ones like $\mathbf{Sets}$ or $\mathbf{Sp}$ in bold face. Objects will be written in uppercase $X, Y, Z, \ldots, U, V, W, \ldots$ or $A, B, C, \ldots$ and morphisms in lower case $f, g, h, \ldots, a, b, c, \ldots$ but sometimes also with greek letters $\varphi, \psi, \chi, \ldots$

In an arbitrary category $\mathbb{C}$ one says that two objects X, Y are **isomorphic** (and one writes $X \cong Y$ then) if there are maps $f\colon X \to Y$ and $g\colon Y \to X$ in $\mathbb{C}$ with $g \circ f = id$ and $f \circ g = id$. This gives the usual notion of isomorphism of sets, groups or topological spaces, when we consider isomorphy in the categories of sets, groups, or spaces.

Finally we mention two ways to form new categories from old.

Definition 2. *Given two categories $\mathbb{C}, \mathbb{D}$ one can form the* **product category** $\mathbb{C} \times \mathbb{D}$ *with*

objects	*pairs (X, U) with $X \in \mathbb{C}$ and $U \in \mathbb{D}$*
morphisms	*$(X, U) \to (Y, V)$ are pairs (f, g) of maps where $f\colon X \to Y$ in $\mathbb{C}$ and $g\colon U \to V$ in $\mathbb{D}$.*

Identities and composition in $\mathbb{C} \times \mathbb{D}$ are obtained componentwise.

Definition 3. *For a category $\mathbb{C}$ one can define the* **opposite category** $\mathbb{C}^{op}$ *by reverting all arrows. Thus $\mathbb{C}^{op}$ has*

objects	$X \in \mathbb{C}$
morphisms	*$X \xrightarrow{f} Y$ in $\mathbb{C}^{op}$ are $Y \xrightarrow{f} X$ in $\mathbb{C}$.*

Identities and composition are inherited from $\mathbb{C}$.

Now we can form categories like $\mathbf{Sets}^{\mathrm{op}} \times \mathbf{Mon}$ or $\mathbf{Sets}^2 = \mathbf{Sets} \times \mathbf{Sets}$.

3 Functors

Mathematical structures generally come equipped with a notion of structure preserving mappings, like monoid and monoid homomorphisms or rings and ring homomorphisms. These structures and their mappings can then be organized in a category. (But there may be more than one sensible notion of mapping, depending on which aspect of the structure one wishes to preserve.) Here we consider appropriate mappings of categories. They are called 'functors'. They preserve the relevant structure of categories: domain, codomain, identities and composition. In this section we define such functors and give many examples.

Definition 4. *Let $\mathbb{C}$ and $\mathbb{D}$ be categories. A* **functor** $F\colon \mathbb{C} \to \mathbb{D}$ *consists of two mappings,*

$$\mathit{Obj}(\mathbb{C}) \longrightarrow \mathit{Obj}(\mathbb{D}), \qquad\qquad \mathit{Arr}(\mathbb{C}) \longrightarrow \mathit{Arr}(\mathbb{D})$$

both written as F, which preserve domain and codomain:

$$f\colon X \to Y \text{ in } \mathbb{C} \quad\Longrightarrow\quad F(f)\colon F(X) \to F(Y) \text{ in } \mathbb{D}$$

and also identities and composition,

$$F(id_X) = id_{F(X)}, \qquad\qquad F(g \circ f) = F(g) \circ F(f).$$

Or, in diagrams,

$$\begin{tikzcd} F(X) \arrow[r, "F(id_X)"] \arrow[d, equal] & F(X) \arrow[d, equal] \\ F(X) \arrow[r, "id_{F(X)}"] & F(X) \end{tikzcd} \qquad\qquad \begin{tikzcd} F(X) \arrow[r, "F(f)"] \arrow[dr, "F(g\circ f)"'] & F(Y) \arrow[d, "F(g)"] \\ & F(Z) \end{tikzcd}$$

We often spare on parentheses and write FX for $F(X)$ and Ff for $F(f)$.

A functor $F\colon \mathbb{C} \to \mathbb{C}$ from a category to itself is called an ***endofunctor****.*

Here is a first set of examples.

Example 5. 1. Taking powersets $X \mapsto \mathcal{P}(X)$ can be extended to an endofunctor $\mathcal{P}\colon \mathbf{Sets} \to \mathbf{Sets}$. What is required is that a function $f\colon X \to Y$ must be turned into a function $\mathcal{P}(f)\colon \mathcal{P}(X) \to \mathcal{P}(Y)$. It must turn a subset of X into a subset of Y. This can be done via what is called the **image**:

$$\mathcal{P}(f)\Big(U \subseteq X\Big) \;\coloneqq\; \Big(\{f(x) \mid x \in U\} \subseteq Y\Big).$$

We have to check that $\mathcal{P}$ preserves identities and composition, i.e. that $\mathcal{P}(\mathit{id}) = \mathit{id}$ and $\mathcal{P}(g \circ f) = \mathcal{P}(g) \circ \mathcal{P}(f)$. This is easy. We check things formally, in detail, for an arbitrary subset $U \subseteq X$.

$$\begin{aligned} \mathcal{P}(\mathit{id})(U) &= \{\mathit{id}(x) \mid x \in U\} \\ &= \{x \mid x \in U\} \\ &= U \\ &= \mathit{id}(U) \\ \mathcal{P}(g \circ f)(U) &= \{(g \circ f)(x) \mid x \in U\} \\ &= \{g(f(x)) \mid x \in U\} \\ &= \{g(y) \mid y \in \{f(x) \mid x \in U\}\} \\ &= \mathcal{P}(g)\Big(\mathcal{P}(f)(U)\Big) \\ &= \big(\mathcal{P}(g) \circ \mathcal{P}(f)\big)(U). \end{aligned}$$

2. We fix a set A and consider the operation $F(X) = A \times X = \{(a, x) \mid a \in A \text{ and } x \in X\}$. We claim that this F forms a functor $\mathbf{Sets} \to \mathbf{Sets}$. Indeed, for a function $f\colon X \to Y$ we can define $F(f)\colon F(X) \to F(Y)$ as $F(f)(a, x) = (a, f(x))$. One can now check that $F(\mathit{id}) = \mathit{id}$ and $F(g \circ f) = F(g) \circ F(f)$.
In such simple cases the definition on morphisms is often left implicit. One then simply says things like: consider the functor $X \mapsto A \times X$ for a fixed set A. Further details are left implicit.

3. There is a "graph" functor
$$G\colon \mathbf{Sets} \longrightarrow \mathbf{Rel}$$
which maps an object $X \in \mathbf{Sets}$ to $X \in \mathbf{Rel}$ (so it's the identity on objects) and maps a function $f\colon X \to Y$ in $\mathbf{Sets}$ to the relation $G(f) = \{\langle x, y\rangle \mid f(x) = y\} \subseteq X \times Y$. Then G clearly preserves domain and codomain. As to identities we have
$$\begin{aligned} G(\mathit{id}_X) &= \{\langle x, y\rangle \in X \times X \mid \mathit{id}_X(x) = y\} \\ &= \{\langle x, x\rangle \mid x \in X\} \\ &= \mathit{id}_{G(X)}\colon X \longrightarrow X \text{ in } \mathbf{Rel}. \end{aligned}$$
And for $f\colon X \to Y$ and $g\colon Y \to Z$,
$$\begin{aligned} G(g \circ f) &= \{\langle x, z\rangle \mid g(f(x)) = z\} \\ &= \{\langle x, z\rangle \mid \exists y \in Y.\, f(x) = y \text{ and } g(y) = z\} \\ &= \{\langle x, z\rangle \mid \exists y \in Y.\, xG(f)y \text{ and } yG(g)z\} \\ &= G(g) \circ G(f). \end{aligned}$$
Notice that the first composition $\circ$ is in $\mathbf{Sets}$ and the second one is in $\mathbf{Rel}$. Thus one has be keenly aware in which category (universe) we are working.

In the two degenerate cases of categories (preorders and monoids) functors turn out to be well-known mappings.

Proposition 6. *1. Consider two preorders $(X, \sqsubseteq)$ and $(Y, \preceq)$ as categories (with at most one morphism between two objects). A functor $F\colon (X, \sqsubseteq) \to (Y, \preceq)$ is then nothing but a function $F\colon X \to Y$ between the underlying sets, which is **monotone**: $x \sqsubseteq x' \implies F(x) \preceq F(x')$.*

2. Consider two monoids $(M, \cdot, u)$ and $(N, \bullet, v)$ as categories (with only one object). A functor $F\colon (M, \cdot, u) \to (N, \bullet, v)$ is then a monoid homomorphism: a function $F\colon M \to N$ between the underlying sets which preserves multiplication and units: $F(x \cdot y) = F(x) \bullet F(y)$ and $F(u) = v$. ❑

This result has no substance at all. Just unravelling the definitions yields the statements.

Lemma 7. *For each category $\mathbb{C}$ there is an identity functor $\mathit{id}\colon \mathbb{C} \to \mathbb{C}$ which maps $X \mapsto X$ and $(f\colon X \to Y) \mapsto (f\colon X \to Y)$.*

Also, for functors $F\colon \mathbb{C} \to \mathbb{D}$ and $G\colon \mathbb{D} \to \mathbb{E}$ there is a composite functor $G \circ F\colon \mathbb{C} \to \mathbb{E}$ given by
$$\begin{aligned} X &\mapsto G(F(X)) \\ \Big(f\colon X \to Y\Big) &\mapsto \Big(G(F(f))\colon G(F(X)) \to G(F(Y))\Big) \end{aligned}$$
Usually we simply write GFX and GFf for $G(F(X))$ and $G(F(f))$.

Thus we can form a category with categories as objects and functors as morphisms. We write $\mathbf{Cat}$ for this category. □

(A reader with some acquaintance with paradoxes in set theory may start to feel a bit uncomfortable at this point: if there is a category of categories, then this category must be an object of itself. The way to avoid such circularity is to say that $\mathbf{Cat}$ is the category of **small** categories, where a category $\mathbb{C}$ is called small if its collections *Obj*$(\mathbb{C})$ and *Arr*$(\mathbb{C})$ of objects and arrows are small sets—as opposed to proper classes. But at the moment we don't care too much about these distinctions.)

An important class of functors are so-called **forgetful functors**. They forget part of the structure. For example, there are forgetful functors,

$$\mathbf{Mon} \longrightarrow \mathbf{Sets}, \qquad \text{and} \qquad \mathbf{PoSets} \longrightarrow \mathbf{PreOrd} \longrightarrow \mathbf{Sets}$$

The first of these maps a monoid $(M, \cdot, u)$ to its underlying set M and forgets about the monoid structure. The second one first maps a poset $(X, \sqsubseteq)$ to the preorder $(X, \sqsubseteq)$ by forgetting about anti-symmetry, and then to its underlying set X by forgetting about the order altogether. Note that in going from posets to preorders one forgets a property and not some structure.

A functor $F\colon \mathbb{C} \to \mathbb{D}$ is called **faithful** if it is injective on morphisms, i.e.

$$F(f) = F(g)\colon FX \to FY \text{ in } \mathbb{D} \implies f = g\colon X \to Y \text{ in } \mathbb{C}$$

And F is called **full** if for each $h\colon FX \to FY$ in $\mathbb{D}$ there is an $f\colon X \to Y$ in $\mathbb{C}$ with $h = Ff$.

One says that $\mathbb{C}$ is a **subcategory** of $\mathbb{D}$ if there are inclusions,

$$\mathit{Obj}(\mathbb{C}) \hookrightarrow \mathit{Obj}(\mathbb{D}), \qquad \mathit{Arr}(\mathbb{C}) \hookrightarrow \mathit{Arr}(\mathbb{D})$$

forming a functor $\mathbb{C} \hookrightarrow \mathbb{D}$. This simply means that objects and arrows of $\mathbb{C}$ are objects and arrows of $\mathbb{D}$ and that identities and composition in $\mathbb{C}$ are as in $\mathbb{D}$. One says that $\mathbb{C}$ is a **full subcategory** of $\mathbb{D}$ if the inclusion $\mathbb{C} \hookrightarrow \mathbb{D}$ is a full functor. This means that for $X, Y \in \mathbb{C}$ one has $\mathbb{C}(X, Y) = \mathbb{D}(X, Y)$. For example, the category of posets is a full subcategory of the category of preorders, that is, the inclusion functor $\mathbf{PoSets} \hookrightarrow \mathbf{PreOrd}$ is full and faithful.

Often one defines categories as full subcategories by only saying what the objects are. For example, there is the full subcategory $\mathbf{FinSets} \hookrightarrow \mathbf{Sets}$ of finite sets. This means that $\mathbf{FinSets}$ has finite sets as objects and (all) ordinary functions between them. Similarly, there are full subcategories $\mathbf{FinGrp} \hookrightarrow \mathbf{Grp}$ and $\mathbf{Ab} \hookrightarrow \mathbf{Grp}$ of finite and abelian (*i.e.* commutative) groups.

One way to think about a functor is as a mapping which takes objects in one category and turns them into objects in another category. Moreover it transforms the corresponding mappings. The latter is expressed by saying that the mapping is 'functorial'. We list a series of examples.

Example 8. 1. For categories $\mathbb{C}$, $\mathbb{D}$ we can form the product category $\mathbb{C} \times \mathbb{D}$ as in Definition 2. It comes equipped with two **projection functors**

$$\mathbb{C} \xleftarrow{\mathit{Fst}} \mathbb{C} \times \mathbb{D} \xrightarrow{\mathit{Snd}} \mathbb{D}$$

which map

$$X \leftarrow\!\shortmid (X, U) \mapsto U.$$

Also there is a **diagonal functor**

$$\varDelta\colon \mathbb{C} \longrightarrow \mathbb{C} \times \mathbb{C} \qquad \text{by} \qquad X \mapsto (X, X).$$

2. Let $\mathbf{Mon}$ be the category of monoids and monoid homomorphisms. The latter are functions between the underlying sets that preserves multiplication and neutral elements. There is a forgetful functor $\mathbf{Mon} \to \mathbf{Sets}$. We show that there is also a functor $\mathcal{L}\colon \mathbf{Sets} \to \mathbf{Mon}$ in the other direction, given by the list operation that we mentioned earlier.
   The claim is that this assignment $X \mapsto (\mathcal{L}(X), +\!\!+, \langle\rangle)$ is functorial. For a function $f\colon X \to Y$ in $\mathbf{Sets}$ we can define a function $\mathcal{L}(f)\colon \mathcal{L}(X) \to \mathcal{L}(Y)$ by:

   $$\mathcal{L}(f)\Big(\langle x_1, \ldots, x_n\rangle\Big) \coloneqq \langle f(x_1), \ldots, f(x_n)\rangle.$$

   In functional programming this is called the list-map operation. This $\mathcal{L}(f)$ forms a monoid homomorphism. Clearly, $\mathcal{L}(f)(\langle\rangle) = \langle\rangle$. Further:

   $$\begin{aligned}
   &\mathcal{L}(f)\Big(\langle x_1, \ldots, x_n\rangle +\!\!+ \langle x'_1, \ldots, x'_m\rangle\Big)\\
   &= \mathcal{L}(f)\Big(\langle x_1, \ldots, x_n, x'_1, \ldots, x'_m\rangle\Big)\\
   &= \langle f(x_1), \ldots, f(x_n), f(x'_1), \ldots, f(x'_m)\rangle\\
   &= \langle f(x_1), \ldots, f(x_n)\rangle +\!\!+ \langle f(x'_1), \ldots, f(x'_m)\rangle\\
   &= \mathcal{L}(f)\Big(\langle x_1, \ldots, x_n\rangle\Big) +\!\!+ \mathcal{L}(f)\Big(\langle x'_1, \ldots, x'_m\rangle\Big).
   \end{aligned}$$

   Hence, $\mathcal{L}(f)$ is a morphism $(\mathcal{L}(X), +\!\!+, \langle\rangle) \to (\mathcal{L}(Y), +\!\!+, \langle\rangle)$ in the category $\mathbf{Mon}$ of monoids.
   We are not done yet! We still have to check that $\mathcal{L}$ preserves identities and composition, i.e. that $\mathcal{L}(\mathit{id}) = \mathit{id}$ and $\mathcal{L}(g \circ f) = \mathcal{L}(g) \circ \mathcal{L}(f)$. This is easy and left to the interested reader.
   We conclude that saying that we have a list functor $\mathcal{L}\colon \mathbf{Sets} \to \mathbf{Mon}$ is a compact statement that contains a lot of information. By the way, we can now see that $\mathcal{L}$ is also a functor $\mathbf{Sets} \to \mathbf{Sets}$ via composition of functors in:

   $$\begin{tikzcd}
   \mathbf{Sets} \arrow[r, "\mathcal{L}"] \arrow[dr, "\mathcal{L}"'] & \mathbf{Mon} \arrow[d, "\text{forgetful}"] \\
   & \mathbf{Sets}
   \end{tikzcd}$$

3. (Slightly technical / advanced) There is a standard way to turn a preorder into a poset by taking suitable equivalence classes. This passage can be extended to morphisms and yields a functor $\mathbf{PreOrd} \to \mathbf{PoSets}$ in the following way. Given a preorder $(X, \sqsubseteq)$, define an equivalence relation $\sim$ on X by

   $$x \sim y \overset{\text{def}}{\Longleftrightarrow} x \sqsubseteq y \text{ and } y \sqsubseteq x$$

and let $X/\!\sim\; = \{[x] \mid x \in X\}$ be the associated set of equivalence classes—with $[x] = \{y \in X \mid x \sim y\}$. Then $X/\!\sim$ is a poset with as ordering

$$[x] \sqsubseteq [y] \overset{\text{def}}{\Longleftrightarrow} x \sqsubseteq y \text{ in } X.$$

This order is well-defined in the sense that it's independent of the choice of representatives: if $x \sim x'$ and $y \sim y'$ then $[x] \sqsubseteq [y]$ if and only if $[x'] \sqsubseteq [y']$. And this order on $X/\!\sim$ is clearly anti-symmetric.
We claim that the assignment $(X, \sqsubseteq) \mapsto (X/\!\sim, \sqsubseteq)$ is functorial, i.e. that it extends to morphisms. Indeed if $f\colon (X, \sqsubseteq) \to (Y, \preceq)$ is a morphism in $\mathbf{PreOrd}$ (i.e. f is a monotone function $X \to Y$), then we can define a function $X/\!\sim\, \to Y/\!\sim$ by $[x] \mapsto [f(x)]$. The details that one gets a well-defined monotone function, and that identities and composition are preserved, are left as an exercise.

4. (For mathematics students) Next we will define the so-called **Alexandroff** functor $\mathcal{A}\colon \mathbf{PreOrd} \to \mathbf{Sp}$ which maps a preorder to its set of elements with the Alexandroff topology. This gives an example of a full and faithful functor.
But first we need some preparatory definitions. Let $(X, \sqsubseteq)$ be a preorder. A subset $a \subseteq X$ is called an **upper set** if it is upper closed: if $x \in a$ and $x \sqsubseteq y$, then $y \in a$. For each element $x \in X$ there is the principal upperset

$$\uparrow\! x = \{y \in X \mid x \sqsubseteq y\}$$

of elements above x. The set of all uppersets in X will be written as $\mathcal{O}_{\mathcal{A}}(X)$. It is closed under arbitrary unions and intersections and thus forms a topology on X. We write $\mathcal{A}(X) = (X, \mathcal{O}_{\mathcal{A}}(X))$. We thus have the object part of the functor $\mathcal{A}\colon \mathbf{PreOrd} \to \mathbf{Sp}$.
For two preorders $(X, \sqsubseteq)$ and $(Y, \preceq)$ and a function $f\colon X \to Y$ one has

$$\begin{array}{c} f \text{ is monotone } (X, \sqsubseteq) \to (Y, \preceq), \text{ i.e. a morphism in } \mathbf{PreOrd} \\ \Updownarrow \\ f \text{ is continuous } (X, \mathcal{O}_{\mathcal{A}}(X)) \to (Y, \mathcal{O}_{\mathcal{A}}(Y)), \text{ i.e. a morphism in } \mathbf{Sp} \end{array}$$

This gives a tight correspondence between an order-theoretic and a topological property. The proof goes as follows.

($\Downarrow$) Assume f is monotone. For an upper set $b \subseteq Y$ we have to show that $f^{-1}(b) = \{x \mid f(x) \in b\}$ is an upper set of X. Assume therefore that $y \sqsupseteq x \in f^{-1}(b)$. Then $f(x) \in b$ and $f(y) \succeq f(x)$ by monotonicity. Hence $f(y) \in b$, since b is an upper set, and thus $y \in f^{-1}(b)$.

($\Uparrow$) Assume now that f is continuous, i.e. f^{-1} maps uppersets to uppersets. We show that f is monotone: for $x, y \in X$ one has $x \sqsubseteq y \implies f(x) \preceq f(y)$. Assume $x \sqsubseteq y$ but not $f(x) \not\preceq f(y)$. Then $f(x) \in \{z \in Y \mid z \not\preceq g(y)\}$. The latter is an upperset of Y, so

$$f^{-1}(\{z \in Y \mid z \not\preceq f(y)\}) = \{w \in X \mid f(w) \not\preceq f(y)\}$$

is an upper set of X. It contains x by assumption. Since $x \sqsubseteq y$, it must also contain y. But that's impossible. Hence $f(x) \preceq f(y)$.

We get a functor $\mathcal{A}\colon \mathbf{PreOrd} \to \mathbf{Sp}$ which is the identity on morphisms. Therefore we use the above implication $(\Downarrow)$. This functor is then obviously faithful; it's full by the implication $(\Updownarrow)$.