2

ABSTRACT STRUCTURES

We begin with some remarks about category-theoretical definitions. These are characterizations of properties of objects and arrows in a category solely in terms of other objects and arrows, that is, just in the language of category theory. Such definitions may be said to be abstract, structural, operational, relational, or perhaps external (as opposed to internal). The idea is that objects and arrows are determined by the role they play in the category via their relations to other objects and arrows, that is, by their position in a structure and not by what they "are" or "are made of" in some absolute sense. The free monoid or category construction of the foregoing chapter was an example of one such definition, and we will see many more examples of this kind later; for now we start with some very simple ones. Let us call them *abstract characterizations*. We will see that one of the basic ways of giving such an abstract characterization is via a Universal Mapping Property or *UMP*.

2.1 Epis and monos

Recall that in **Sets**, a function $f : A \to B$ is called

injective if $f(a) = f(a')$ implies $a = a'$ for all $a, a' \in A$,
surjective if for all $b \in B$ there is some $a \in A$ with $f(a) = b$.

We have the following abstract characterizations of these properties:

Definition 2.1. In any category $\mathbf{C}$, an arrow

$$f : A \to B$$

is called a:

monomorphism, if given any $g, h : C \to A$, $fg = fh$ implies $g = h$,

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

epimorphism, if given any $i, j : B \to D$, $if = jf$ implies $i = j$.

$$\begin{tikzcd} A \arrow[r, "f"] & B \arrow[r, shift left, "i"] \arrow[r, shift right, "j"'] & D \end{tikzcd}$$

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 sets is monic just in case it is injective.*

Proof. Suppose $f : A \rightarrowtail B$. Let $a, a' \in A$ such that $a \neq a'$, and let $\{x\}$ be any given one-element set. Consider the functions

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

where

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

Since $\bar{a} \neq \bar{a'}$, it follows, since f is a monomorphism, that $f\bar{a} \neq f\bar{a'}$. Thus, $f(a) = (f\bar{a})(x) \neq (f\bar{a'})(x) = f(a')$. Whence f is injective.

Conversely, if f is injective and $g, h : C \to A$ are functions such that $g \neq h$, then for some $c \in C$, $g(c) \neq h(c)$. Since f is injective, it follows that $f(g(c)) \neq f(h(c))$, whence $fg \neq fh$. □

Example 2.3. In many categories of "structured sets" like monoids, the monos are exactly the "injective homomorphisms." More precisely, a homomorphism $h : M \to N$ of monoids is monic just if the underlying function $|h| : |M| \to |N|$ is monic, that is, injective by the foregoing. To prove this, suppose h is monic and take two different "elements" $x, y : 1 \to |M|$, where $1 = \{*\}$ is any one-element set. 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 $h \circ \bar{x}, h \circ \bar{y} : M(1) \to M \to N$, since h is monic. Thus, the corresponding "elements" $hx, hy : 1 \to N$ of N are also distinct, again by the UMP of $M(1)$.

$$\begin{tikzcd} M(1) \arrow[r, shift left, "\bar{x}"] \arrow[r, shift right, "\bar{y}"'] & M \arrow[r, "h"] & N \end{tikzcd}$$

$$\begin{tikzcd} 1 \arrow[r, shift left, "x"] \arrow[r, shift right, "y"'] & {|M|} \arrow[r, "|h|"] & {|N|} \end{tikzcd}$$

Conversely, if $|h| : |M| \to |N|$ is monic and $f, g : X \to M$ are any distinct homomorphisms, then $|f|, |g| : |X| \to |M|$ are distinct functions, and so $|h| \circ |f|, |h| \circ |g| : |X| \to |M| \to |N|$ are distinct, since $|h|$ is monic. Since therefore $|h \circ f| = |h| \circ |f| \neq |h| \circ |g| = |h \circ g|$, we also must have $h \circ f \neq h \circ g$.

A completely analogous situation holds, for example, for groups, rings, vector spaces, and posets. We shall see that this fact follows from the presence, in each of these categories, of certain objects like the free monoid $M(1)$.

Example 2.4. In a poset $\mathbf{P}$, every arrow $p \leq q$ is both monic and epic. Why?

Now, dually to the foregoing, the epis in **Sets** are exactly the surjective functions (exercise!); by contrast, however, in many other familiar categories they are not just the surjective homomorphisms, as the following example shows.

Example 2.5. In the category **Mon** of monoids and monoid homomorphisms, there is a monic homomorphism

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

where $\mathbb{N}$ is the additive monoid $(N, +, 0)$ of natural numbers and $\mathbb{Z}$ is the additive monoid $(Z, +, 0)$ of integers. We will show that this map, given by the inclusion $N \subset Z$ of sets, 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 N are equal, $f\mid_N = g\mid_N$, then $f = g$.

Note first that:

$$\begin{aligned} f(-n) &= f((-1)_1 + (-1)_2 + \cdots + (-1)_n) \\ &= f(-1)_1 * f(-1)_2 * \cdots * f(-1)_n \end{aligned}$$

and similarly for g. It therefore suffices to show that $f(-1) = g(-1)$. But

$$\begin{aligned} f(-1) &= f(-1) * u \\ &= f(-1) * g(0) \\ &= f(-1) * g(1-1) \\ &= f(-1) * g(1) * g(-1) \\ &= f(-1) * f(1) * g(-1) \\ &= f(-1+1) * g(-1) \\ &= f(0) * g(-1) \\ &= u * g(-1) \\ &= g(-1). \end{aligned}$$

Note that, 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 $x = y$.

Proposition 2.6. *Every iso is both monic and epic.*

Proof. Consider the following diagram:

$$\begin{tikzcd} A \arrow[r, shift left, "x"] \arrow[r, shift right, "y"'] & B \arrow[r, "m"] \arrow[dr, "1"'] & C \arrow[d, "e"] & \\ & & B \arrow[r, shift left] \arrow[r, shift right] & D \end{tikzcd}$$

If m is an isomorphism with inverse e, then $mx = my$ implies $x = emx = emy = y$. Thus, m is monic. Similarly, e cancels on the right and thus is epic. $\square$

In **Sets** the converse of the foregoing also holds: every mono-epi is iso. But this is not in general true, as shown by the example in monoids above.

2.1.1 Sections and retractions

We have just noted that any iso is both monic and epic. More generally, if an arrow

$$f : A \to B$$

has a left inverse

$$g : B \to A, \quad gf = 1_A$$

then f must be monic and g epic, by the same argument.

Definition 2.7. 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 *splitting* of e, and the arrow e is called a *retraction* of s. The object A is called a *retract* of X.

Since functors preserve identities, they also preserve *split* epis and *split* monos. Compare example 2.5 above in **Mon** where the forgetful functor

$$\mathbf{Mon} \to \mathbf{Set}$$

did not preserve the epi $\mathbb{N} \to \mathbb{Z}$.

Example 2.8. In **Sets**, every mono splits except those of the form

$$\emptyset \rightarrowtail A.$$

The condition that *every epi splits* is the categorical version of the axiom of choice. Indeed, consider an epi

$$e : E \twoheadrightarrow X.$$

We have the family 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 this means that $s(x) \in E_x$ for all $x \in X$.

Conversely, given a family 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 family.

The idea that a "family of objects" $(E_x)_{x \in X}$ can be represented by a single arrow $e : E \to X$ by using the "fibers" $e^{-1}(x)$ has much wider application than this, and will be considered further in Section 7.10.

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

$$
\begin{tikzcd}
 & E \arrow[d, "e"] \\
P \arrow[ur, dotted, "\bar{f}"] \arrow[r, "f"'] & X
\end{tikzcd}
$$

One says that f *lifts across* e. Any epi into a projective object clearly splits. Projective objects may be thought of as having a more "free" structure, thus permitting "more arrows."

The axiom of choice implies that all sets are projective, and it follows that free objects in many (but not all!) categories of algebras then are also projective. The reader should show that, in any category, any retract of a projective object is also projective.

2.2 Initial and terminal objects

We now consider abstract characterizations of the empty set and the one-element sets in the category **Sets** and structurally similar objects in general categories.

Definition 2.9. 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

$$C \to 1.$$

As in the case of monos and epis, note that there is a kind of "duality" in these definitions. Precisely, a terminal object in $\mathbf{C}$ is exactly an initial object in $\mathbf{C}^{\mathrm{op}}$. We will consider this duality systematically in the next chapter.

First, observe that since the notions of initial and terminal object are simple UMPs, such objects are uniquely determined up to isomorphism, just like the free monoids were.

Proposition 2.10. *Initial (terminal) objects are unique up to isomorphism.*

Proof. In fact, if C and C' are both initial (terminal) in the same category, then there is a *unique* isomorphism $C \to C'$. Indeed, suppose that 0 and $0'$ are both initial objects in some category $\mathbf{C}$; the following diagram then makes it clear that 0 and $0'$ are uniquely isomorphic.

$$\begin{tikzcd}
0 \arrow[r, "u"] \arrow[dr, "1_0"'] & 0' \arrow[d, "v"] \arrow[dr, "1_{0'}"] & \\
& 0 \arrow[r, "u"'] & 0'
\end{tikzcd}$$

For terminal objects, apply the foregoing to $\mathbf{C}^{\mathrm{op}}$. □

Example 2.11.

1. In **Sets** the empty set is initial and any singleton set $\{x\}$ is terminal. Observe that **Sets** has just one initial object but many terminal objects (answering the question of whether $\mathbf{Sets} \cong \mathbf{Sets}^{\mathrm{op}}$).
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 **Groups**, the one-element group is *both* initial and terminal (similarly for the category of vector spaces and linear transformations, as well as the category of monoids and monoid homomorphisms). But in **Rings** (commutative with unit), the ring $\mathbb{Z}$ of integers is initial (the one-element ring with $0 = 1$ is terminal).
4. 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

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

There is also an equivalent, fully equational characterization not involving the ordering. A typical example of a Boolean algebra is the powerset $\mathcal{P}(X)$ of all subsets $A \subseteq X$ of a set X, ordered by inclusion $A \subseteq B$, and with the Boolean operations being the empty set $0 = \emptyset$, the whole set $1 = X$, union and intersection of subsets as join and meet, and the relative complement $X - A$ as $\neg A$. A familiar special case is the two-element

Boolean algebra $\mathbf{2} = \{0, 1\}$ (which may be taken to be the powerset $\mathcal{P}(1)$), sometimes also regarded as "truth values" with the logical operations of disjunction, conjunction, and negation as the Boolean operations. It is an initial object in the category **BA** of Boolean algebras. **BA** has as arrows the Boolean homomorphisms, i.e. functors $h : B \to B'$ that preserve the additional structure, in the sense that $h(0) = 0$, $h(a \vee b) = h(a) \vee h(b)$, etc. The one-element Boolean algebra (i.e. $\mathcal{P}(0)$) is terminal.

5. In a poset, an object is plainly initial iff it is the least element, and terminal iff it is the greatest element. Thus, for instance, any Boolean algebra has both. Obviously, a category *need not* have either an initial object or a terminal object; for example, 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 Generalized elements

Let us consider arrows into and out of initial and terminal objects. Clearly only certain of these will be of interest, but those are often especially significant.

A set A has an arrow into the initial object $A \to 0$ just if it is itself initial, and the same is true for posets. In monoids and groups, by contrast, every object has a unique arrow to the initial object, since it is also terminal.

In the category **BA** of Boolean algebras, however, the situation is quite different. The maps $p : B \to \mathbf{2}$ into the initial Boolean algebra $\mathbf{2}$ correspond uniquely to the so-called *ultrafilters* U in B. A *filter* in a Boolean algebra B is a nonempty subset $F \subseteq B$ that is closed upward and under meets:

$$\begin{array}{lll} a \in F \text{ and } a \leq b & \text{implies} & b \in F \\ a \in F \text{ and } b \in F & \text{implies} & a \wedge b \in F \end{array}$$

A filter F is *maximal* if the only strictly larger filter $F \subset F'$ is the "improper" filter, namely all of B. An *ultrafilter* is a maximal filter. It is not hard to see that a filter F is an ultrafilter just if for every element $b \in B$, either $b \in F$ or $\neg b \in F$, and not both (exercise!). Now if $p : B \to \mathbf{2}$, let $U_p = p^{-1}(1)$ to get an ultrafilter $U_p \subset B$. And given an ultrafilter $U \subset B$, define $p_U(b) = 1$ iff $b \in U$ to get a Boolean homomorphism $p_U : B \to \mathbf{2}$. This is easy to check, as is the fact that these operations are mutually inverse. Boolean homomorphisms $B \to \mathbf{2}$ are also used in forming the "truth tables" one meets in logic. Indeed, a row of a truth table corresponds to such a homomorphism on a Boolean algebra of formulas.

Ring homomorphisms $A \to \mathbb{Z}$ into the initial ring $\mathbb{Z}$ play an analogous and equally important role in algebraic geometry. They correspond to so-called *prime ideals*, which are the ring-theoretic generalizations of ultrafilters.

Now let us consider some arrows from terminal objects. For any set X, for instance, we have an isomorphism

$$X \cong \mathrm{Hom}_{\mathbf{Sets}}(1, X)$$

between elements $x \in X$ and arrows $\bar{x} : 1 \to X$, determined by $\bar{x}(*) = x$, from a terminal object $1 = \{*\}$. We have already used this correspondence several times. 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 space) P. In any category with a terminal object 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 are equal, $fa = ga$.

But be careful; this is not always the case! How many points are there of an object M in the category of monoids? That is, how many arrows of the form $1 \to M$ for a given monoid M? Just one! And how many points does a Boolean algebra have?

Because, in general, an object is not determined by its points, it is convenient to introduce the device of *generalized elements*. These are arbitrary arrows,

$$x : X \to A$$

(with arbitrary domain X), which can be regarded as *generalized* or *variable elements* of A. Computer scientists and logicians sometimes think of arrows $1 \to A$ as constants or closed terms and general arrows $X \to A$ as arbitrary terms. Summarizing:

Example 2.12.

1. 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.
2. By contrast, in **Mon**, for homomorphisms $h, j : M \to N$ we always have $hx = jx$, for all $x : 1 \to M$, since there's just one such point x. Thus monoids do not "have enough points."
3. But in any category **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 $fx = gx$ (why?). Thus, all objects have enough generalized elements.
4. In fact, it often happens that it is enough to consider generalized elements of just a certain form $T \to A$, that is, for certain "test" objects T. We shall consider this presently.

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

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

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 generalized elements."

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

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

we shall have $\alpha f = \beta g$ just if $\alpha f x = \beta g x$ for all generalized elements $x : X \to A$ (just take $x = 1_A : A \to A$).

Example 2.13. Generalized elements can be used to "reveal more structure" than do the constant elements. For example, consider the following posets X and A:

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

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

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

It is easy to see that f is both monic and epic in the category **Pos**; however, it is clearly not an iso. One would like to say that X and A are "different structures," and indeed, their being non-isomorphic says just this. But now, how to *prove* that they are *not* isomorphic (say, via some other $X \to A$)? In general, this sort of thing can be quite difficult.

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. Generalized elements provide an easy way to define invariants. For instance, the number of global elements of X and A is the same, namely the three elements of the sets. But consider instead the "**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, the posets cannot be isomorphic. In more detail, we can define for any poset P the numerical invariant

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

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

$$\begin{tikzcd} P \arrow[r, shift left, "i"] & Q \arrow[l, shift left, "j"] \end{tikzcd}$$

also gives an iso

$$\begin{tikzcd} \operatorname{Hom}(\mathbf{2}, P) \arrow[r, shift left, "i_*"] & \operatorname{Hom}(\mathbf{2}, Q) \arrow[l, shift left, "j_*"] \end{tikzcd}$$

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 $\operatorname{Hom}(X, -)$ is always a functor, and functors always preserve isos.

Example 2.14. As in the foregoing example, it is often the case that generalized 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 generalized 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 $\operatorname{Hom}_{\mathbf{Mon}}(\mathbb{N}, M)$ of all such figures, since

$$|M| \;\cong\; \operatorname{Hom}_{\mathbf{Sets}}(1, |M|) \;\cong\; \operatorname{Hom}_{\mathbf{Mon}}(\mathbb{N}, M).$$

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

Next we are going to see the categorical definition of a product of two objects in a category. This was first given by Mac Lane in 1950, and it is probably the earliest example of category theory being used to define a fundamental mathematical notion.

By "define" here I mean an abstract characterization, in the sense already used, in terms of objects and arrows in a category. And as before, we do this by giving a UMP, which determines the structure at issue up to isomorphism, as usual in category theory. Later in this chapter, we will have several other examples of such characterizations.

Let us begin by considering products of sets. Given sets A and B the *cartesian product* of A and B is the set of ordered pairs

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

Observe that there are two "coordinate projections"

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

with

$$\pi_1(a, b) = a, \qquad \pi_2(a, b) = b.$$

And indeed, given any element $c \in A \times B$ we have

$$c = (\pi_1 c, \pi_2 c).$$

The situation is captured concisely in the following diagram:

$$\begin{tikzcd} & 1 \arrow[dl, "a"'] \arrow[d, dotted, "{(a,b)}"] \arrow[dr, "b"] & \\ A & A \times B \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & B \end{tikzcd}$$

Replacing elements by generalized elements, we get the following definition.

Definition 2.15. 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

$$\begin{tikzcd} & X \arrow[dl, "x_1"'] \arrow[d, dotted, "u"] \arrow[dr, "x_2"] & \\ A & P \arrow[l, "p_1"] \arrow[r, "p_2"'] & B \end{tikzcd}$$

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

Remark 2.16. As in other UMPs, there are two parts:

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

Proposition 2.17. *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. Then, since Q is a product, there is a unique $i : P \to Q$ such that $q_1 \circ i = p_1$ and $q_2 \circ i = p_2$. Similarly, since P is a product, there is a unique $j : Q \to P$ such that $p_1 \circ j = q_1$ and $p_2 \circ j = q_2$.

$$\begin{tikzcd}
 & P \arrow[dl, "p_1"'] \arrow[d, dotted, "i"] \arrow[dr, "p_2"] & \\
A & Q \arrow[l, "q_1"] \arrow[r, "q_2"'] \arrow[d, dotted, "j"] & B \\
 & P \arrow[ul, "p_1"] \arrow[ur, "p_2"'] &
\end{tikzcd}$$

Composing, $p_1 \circ j \circ i = p_1$ and $p_2 \circ j \circ i = p_2$. Since also $p_1 \circ 1_P = p_1$ and $p_2 \circ 1_P = p_2$, it follows from the uniqueness condition that $j \circ i = 1_P$. Similarly, we can show $i \circ j = 1_Q$. Thus, $i : P \to Q$ is an isomorphism. □

If A and B have a product, we write

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

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

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

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

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

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 in two variables"; given any two generalized 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 arrows into products were (to be sure, they are related to the notion of an "exponential" Y^B, via "currying" $\lambda f : A \to Y^B$; we discuss this further in chapter 6).

2.5 Examples of products

1. We have already seen cartesian products of sets. Note that if we choose a different definition of ordered pairs $\langle a, b\rangle$ we get different sets

$$A \times B \quad \text{and} \quad A \times' B$$

each of which is (part of) a product, and so are isomorphic. For instance, we could set:

$$\langle a, b\rangle = \{\{a\}, \{a, b\}\}$$
$$\langle a, b\rangle' = \langle a, \langle a, b\rangle\rangle$$

2. Products of "structured sets" like monoids or 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 $\{\langle g, h\rangle \mid g \in G,\ h \in H\}$ 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 ones $\langle g, h\rangle \mapsto g$ (or h).

3. Similarly, for categories $\mathbf{C}$ and $\mathbf{D}$, we already defined the category of pairs 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). (Check this: verify the UMP for the product category so defined.)

As special cases, we also get products of posets and of monoids as products of categories. (Check this: the projections and unique paired function are always monotone and so the product of posets, constructed in **Cat**, is also a product in **Pos**, and similarly for **Mon**.)

4. Let P be a poset 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,

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

then we need

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

Do you recognize this operation $p \times q$? It is just what is usually called the *greatest lower bound*: $p \times q = p \wedge q$. Many other order-theoretic notions are also special cases of categorical ones, as we shall see later.

5. (For those who know something about Topology.) Let us show that the product of two *topological spaces* X, Y, as usually defined, really is a product in **Top**, the category of spaces and continuous functions. Thus suppose we have spaces X and Y and the product spaces $X \times Y$ with its projections

$$X \xleftarrow{p_1} X \times Y \xrightarrow{p_2} Y$$

Recall that $O(X \times Y)$ is generated by basic open sets of the form $U \times V$ where $U \in O(X)$ and $V \in O(Y)$, so every $W \in O(X \times Y)$ is a union of such basic opens.

- Clearly p_1 is continuous, since $p_1^{-1}U = U \times Y$.
- Given any continuous $f_1 : Z \to X, f_2 : Z \to Y$, let $f : Z \to X \times Y$ be the function $f = \langle f_1, f_2 \rangle$. We just need to see that f is continuous.

- Given any $W = \bigcup_i (U_i \times V_i) \in O(X \times Y)$, $f^{-1}(W) = \bigcup_i f^{-1}(U_i \times V_i)$, so it suffices to show $f^{-1}(U \times V)$ is open. But

$$\begin{aligned} f^{-1}(U \times V) &= f^{-1}((U \times Y) \cap (X \times V)) \\ &= f^{-1}(U \times Y) \cap f^{-1}(X \times V) \\ &= f^{-1} \circ p_1^{-1}(U) \cap f^{-1} \circ p_2^{-1}(V) \\ &= (f_1)^{-1}(U) \cap (f_2)^{-1}(V) \end{aligned}$$

where $(f_1)^{-1}(U)$ and $(f_2)^{-1}(V)$ are open, since f_1 and f_2 are continuous.

The following diagram concisely captures the situation at hand:

$$\begin{tikzcd} & O(Z) & \\ O(X) \arrow[ur, "f_1^{-1}"] \arrow[r, "p_1^{-1}"'] & O(X \times Y) \arrow[u, dotted, "f^{-1}"'] & O(Y) \arrow[ul, "f_2^{-1}"'] \arrow[l, "p_2^{-1}"] \end{tikzcd}$$

6. (For those familiar with type theory.) Let us consider the *category of types* of the (simply typed) λ-calculus. The λ–calculus is a formalism for the specification and manipulation of functions, based on the notions of "binding of variables" and functional evaluation. For example, given the real polynomial expression $x^2 + 2y$, in the λ-calculus one writes $\lambda y.x^2 + 2y$ for the function $y \mapsto x^2 + 2y$ (for each fixed value x), and $\lambda x \lambda y.x^2 + 2y$ for the function-valued function $x \mapsto (y \mapsto x^2 + 2y)$.

   Formally, the λ-calculus consists of:

- Types: $A \times B,\ A \to B, \ldots$ (generated from some basic types)
- Terms:

$$\begin{array}{rl} x, y, z, \ldots : A & \text{(variables for each type } A) \\ a : A,\ b : B, \ldots & \text{(possibly some typed constants)} \\ \langle a, b \rangle : A \times B & (a : A,\ b : B) \\ \mathrm{fst}(c) : A & (c : A \times B) \\ \mathrm{snd}(c) : B & (c : A \times B) \\ ca : B & (c : A \to B,\ a : A) \\ \lambda x.b : A \to B & (x : A,\ b : B) \end{array}$$

- Equations:

$$\begin{aligned} \mathrm{fst}(\langle a, b\rangle) &= a \\ \mathrm{snd}(\langle a, b\rangle) &= b \\ \langle \mathrm{fst}(c), \mathrm{snd}(c)\rangle &= c \\ (\lambda x.b)a &= b[a/x] \\ \lambda x.cx &= c \quad (\text{no } x \text{ in } c) \end{aligned}$$

The relation $a \sim b$ (usually called *$\beta\eta$-equivalence*) on terms is defined to be the equivalence relation generated by the equations, and renaming of bound variables:

$$\lambda x.b = \lambda y.b[y/x] \quad (\text{no } y \text{ in } b)$$

The category of types $\mathbf{C}(\lambda)$ is now defined as follows:

- objects: the types,
- arrows $A \to B$: closed terms $c : A \to B$, identified if $c \sim c'$,
- identities: $1_A = \lambda x.x$ (where $x : A$),
- composition: $c \circ b = \lambda x.c(bx)$.

Let us verify that this is a well-defined category:
Unit laws:

$$\begin{aligned} c \circ 1_B &= \lambda x(c((\lambda y.y)x)) = \lambda x(cx) = c \\ 1_C \circ c &= \lambda x((\lambda y.y)(cx)) = \lambda x(cx) = c \end{aligned}$$

Associativity:

$$\begin{aligned} c \circ (b \circ a) &= \lambda x(c((b \circ a)x)) \\ &= \lambda x(c((\lambda y.b(ay))x)) \\ &= \lambda x(c(b(ax))) \\ &= \lambda x(\lambda y(c(by))(ax)) \\ &= \lambda x((c \circ b)(ax)) \\ &= (c \circ b) \circ a \end{aligned}$$

This category has binary products. Indeed, given types A and B, let

$$p_1 = \lambda z.\mathrm{fst}(z), \quad p_2 = \lambda z.\mathrm{snd}(z) \quad (z : A \times B).$$

And given a and b as in

$$\begin{tikzcd} & X \arrow[dl, "a"'] \arrow[d, dotted, "{(a,b)}"] \arrow[dr, "b"] & \\ A & A \times B \arrow[l, "p_1"] \arrow[r, "p_2"'] & B \end{tikzcd}$$

let

$$(a, b) = \lambda x.\langle ax, bx\rangle.$$

Then

$$\begin{aligned} p_1 \circ (a, b) &= \lambda x(p_1((\lambda y.\langle ay, by\rangle)x)) \\ &= \lambda x(p_1\langle ax, bx\rangle) \\ &= \lambda x(ax) \\ &= a. \end{aligned}$$

Similarly, $p_2 \circ (a, b) = b$.

Finally, if $c : X \to A \times B$ also has

$$p_1 \circ c = a, \qquad p_2 \circ c = b$$

then

$$\begin{aligned} (a, b) &= \lambda x.\langle ax, bx\rangle \\ &= \lambda x.\langle (p_1 \circ c)x, (p_2 \circ c)x\rangle \\ &= \lambda x.\langle (\lambda y(p_1(cy)))x, (\lambda y(p_2(cy)))x\rangle \\ &= \lambda x.\langle (\lambda y((\lambda z.\mathrm{fst}(z))(cy)))x, (\lambda y((\lambda z.\mathrm{snd}(z))(cy)))x\rangle \\ &= \lambda x.\langle \lambda y(\mathrm{fst}(cy))x, \lambda y(\mathrm{snd}(cy))x\rangle \\ &= \lambda x.\langle \mathrm{fst}(cx), \mathrm{snd}(cx)\rangle \\ &= \lambda x.(cx) \\ &= c. \end{aligned}$$

Remark 2.18. The λ-calculus had another surprising interpretation, namely as a system of notation for proofs in propositional calculus; this is known as the "Curry-Howard" correspondence. Briefly, the idea is that one interprets types as propositions (with $A \times B$ being conjunction and $A \to B$ implication) and terms $a : A$ as proofs of the proposition A. The term-forming rules such as

$$\frac{a : A \qquad b : B}{\langle a, b\rangle : A \times B}$$

can then be read as annotated rules of inference, showing how to build up labels for proofs inductively. So, for instance, a natural deduction proof such as

$$\cfrac{\cfrac{\cfrac{[A] \qquad [B]}{A \times B}}{B \to (A \times B)}}{A \to (B \to (A \times B))}$$

with square brackets indicating cancellation of premisses, is labeled as follows:

$$\dfrac{\dfrac{\dfrac{[x : A] \qquad [y : B]}{\langle x, y\rangle : A \times B}}{\lambda y.\langle x, y\rangle : B \to (A \times B)}}{\lambda x \lambda y.\langle x, y\rangle : A \to (B \to (A \times B))}$$

The final "proof term" $\lambda x \lambda y.\langle x, y\rangle$ thus records the given proof of the "proposition" $A \to (B \to (A \times B))$, and a different proof of the same proposition would give a different term.

Although one often speaks of a resulting "isomorphism" between logic and type theory, what we in fact have here is simply a functor from the category of proofs in the propositional calculus with conjunction and implication (as defined in example 10.), into the category of types of the lambda calculus. The functor will not generally be an isomorphism unless we impose some further equations between proofs.

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

$$\begin{tikzcd}
A \arrow[d, "f"'] & A \times A' \arrow[l, "p_1"'] \arrow[r, "p_2"] & A' \arrow[d, "f'"] \\
B & B \times B' \arrow[l, "q_1"] \arrow[r, "q_2"'] & B'
\end{tikzcd}$$

with indicated products. Then we write

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

for $f \times f' = \langle f \circ p_1, f' \circ p_2\rangle$. Thus, both squares in the following diagram commute.

$$\begin{tikzcd}
A \arrow[d, "f"'] & A \times A' \arrow[l, "p_1"'] \arrow[r, "p_2"] \arrow[d, dotted, "f \times f'"] & A' \arrow[d, "f'"] \\
B & B \times B' \arrow[l, "q_1"] \arrow[r, "q_2"'] & B'
\end{tikzcd}$$

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

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

as the reader can easily check, using the UMP of the product. A category which has a product for every pair of objects is said to *have binary products.*

We can also define ternary products

$$A_1 \times A_2 \times A_3$$

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 each of the three i's.) Plainly, such a condition can be formulated for any number of factors.

It is clear, however, that if a category has binary products, then it has all finite products with two or more factors; for instance, one could set

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

to satisfy the UMP for ternary products. On the other hand, one could instead have taken $A \times (B \times C)$ just as well. This shows that the binary product operation $A \times B$ is associative up to isomorphism, for we must have

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

by the UMP of ternary products.

Observe also that a terminal object is a "null-ary" product, that is, a product of no 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 can also define the product of a family of objects $(C_i)_{i \in I}$ indexed by *any* set I, by giving a UMP for "I-ary products" analogous to those for nullary, unary, binary, and n-ary products. We leave the precise formulation of this UMP as an exercise.

Definition 2.19. 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 (small) products* if every set of objects in $\mathbf{C}$ has a product.

2.7 Hom-sets

In this section, we assume that all categories are locally small.

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*. Note that any arrow $g : B \to B'$ in $\mathbf{C}$ induces a function:

$$\begin{aligned} \mathrm{Hom}(A, g) &: \mathrm{Hom}(A, B) \to \mathrm{Hom}(A, B') \\ (f : A \to B) &\mapsto (g \circ f : A \to B \to B') \end{aligned}$$

Thus, $\mathrm{Hom}(A, g) = g \circ f$; one sometimes writes g_* instead of $\mathrm{Hom}(A, g)$, so

$$g_*(f) = g \circ f.$$

Let us show that this determines a functor

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

called the (covariant) *representable functor* of A. We need to show that

$$\mathrm{Hom}(A, 1_X) = 1_{\mathrm{Hom}(A,X)}$$

and that

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

Taking an argument $x : A \to X$, we clearly have

$$\begin{aligned} \mathrm{Hom}(A, 1_X)(x) &= 1_X \circ x \\ &= x \\ &= 1_{\mathrm{Hom}(A,X)}(x) \end{aligned}$$

and

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

We will study such representable functors much more carefully later. For now we just want to see how one can use Hom-sets to give another formulation of the definition of products.

For any object P, a pair of arrows $p_1 : P \to A$ and $p_2 : P \to B$ determine an element (p_1, p_2) 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.

$$
\begin{tikzcd}
 & X \arrow[dl, "x_1"'] \arrow[d, "x"] \arrow[dr, "x_2"] & \\
A & P \arrow[l, "p_1"] \arrow[r, "p_2"'] & B
\end{tikzcd}
$$

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

$$\vartheta_X(x) = (x_1, x_2) \tag{2.1}$$

This function ϑ_X can be used to express concisely the condition of being a product as follows.

Proposition 2.20. *A diagram of the form*

$$
\begin{tikzcd}
A & P \arrow[l, "p_1"] \arrow[r, "p_2"'] & B
\end{tikzcd}
$$

is a product for A and B iff for every object X, the canonical function ϑ_X given in (2.1) *is an isomorphism,*

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

Proof. Examine the UMP of the product: it says exactly that for every element $(x_1, x_2) \in \mathrm{Hom}(X, A) \times \mathrm{Hom}(X, B)$, there is a unique $x \in \mathrm{Hom}(X, P)$ such that $\vartheta_X(x) = (x_1, x_2)$, that is, ϑ_X is bijective. $\square$

Definition 2.21. Let $\mathbf{C}$, $\mathbf{D}$ be categories with binary products. A functor $F : \mathbf{C} \to \mathbf{D}$ is said to *preserve binary products* if it takes every product diagram

$$
\begin{tikzcd}
A & A \times B \arrow[l, "p_1"] \arrow[r, "p_2"'] & B
\end{tikzcd}
\qquad \text{in } \mathbf{C}
$$

to a product diagram

$$
\begin{tikzcd}
FA & F(A \times B) \arrow[l, "Fp_1"] \arrow[r, "Fp_2"'] & FB
\end{tikzcd}
\qquad \text{in } \mathbf{D}.
$$

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 iso.

For example, the forgetful functor $U : \mathbf{Mon} \to \mathbf{Sets}$ preserves binary products.

Corollary 2.22. *For any object X in a category $\mathbf{C}$ with products, the (covariant) representable functor*

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

preserves products.

Proof. For any $A, B \in \mathbf{C}$, the foregoing proposition 2.20 says that there is a canonical isomorphism:

$$\mathrm{Hom}_{\mathbf{C}}(X, A \times B) \cong \mathrm{Hom}_{\mathbf{C}}(X, A) \times \mathrm{Hom}_{\mathbf{C}}(X, B)$$

□

2.8 Exercises

1. Show that a function between sets is an epimorphism if and only if it is surjective. Conclude that the isos in **Sets** are exactly the epi-monos.
2. Show that in a poset category, all arrows are both monic and epic.
3. (Inverses are unique) If an arrow $f : A \to B$ has inverses $g, g' : B \to A$ (i. e. $g \circ f = 1_A$ and $f \circ g = 1_B$, and similarly for g'), then $g = g'$.
4. With regard to a commutative triangle,

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

   in any category $\mathbf{C}$, show

   a. if f and g are isos (resp. monos, resp. epis), so is h;
   b. if h is monic, so is f;
   c. if h is epic, so is g;
   d. (by example) if h is monic, g need not be.

5. Show that the following are equivalent for an arrow

$$f : A \to B$$

   in any category.

   (a) f is an isomorphism;
   (b) f is both a mono and a split epi;
   (c) f is both a split mono and an epi;

(d) f is both a split mono and a split epi.

6. Show that a homomorphism $h : G \to H$ of graphs is monic just if it is injective on both edges and vertices.
7. Show that in any category, any retract of a projective object is also projective.
8. Show that all sets are projective (use the axiom of choice).
9. Show that the epis among posets are the surjections (on elements), and that the one-element poset $\mathbf{1}$ is projective.
10. Show that sets, regarded as discrete posets, are projective in the category of posets (use the foregoing exercises). Give an example of a poset that is not projective. Show that every projective poset is discrete, i.e. a set. Conclude that **Sets** is (isomorphic to) the "full subcategory" of projectives in **Pos**, consisting of all projective posets and all monotone maps between them.
11. Let A be a set. Define an *A-monoid* to be a monoid M equipped with a function $m : A \to U(M)$ (to the underlying set of M). A morphism $h : (M, m) \to (N, n)$ of A-monoids is to be a monoid homomorphism $h : M \to N$ such that $U(h) \circ m = n$ (a commutative triangle). Together with the evident identities and composites, this defines a category A-**Mon** of A-monoids.

    Show that an initial object in A-**Mon** is the same thing as a free monoid $M(A)$ on A. (Hint: compare their respective UMPs.)
12. Show that for any Boolean algebra B, Boolean homomorphisms $h : B \to \mathbf{2}$ correspond exactly to ultrafilters in B.
13. In any category with binary products, show directly that:

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

14. (a) For any index set I, define the product $\prod_{i \in I} X_i$ of an I-indexed family of objects $(X_i)_{i \in I}$ in a category, by giving a UMP generalizing that for binary products (the case $I = 2$).

    (b) Show that in **Sets**, for any set X the set X^I of all functions $f : I \to X$ has this UMP, with respect to the "constant family" where $X_i = X$ for all $i \in I$, and thus

$$X^I \cong \prod_{i \in I} X$$

15. Given a category $\mathbf{C}$ with objects A and B, define the category $\mathbf{C}_{A,B}$ to have objects (X, x_1, x_2), where $x_1 : X \to A$, $x_2 : X \to B$, and with arrows $f : (X, x_1, x_2) \to (Y, y_1, y_2)$ being arrows $f : X \to Y$ with $y_1 \circ f = x_1$ and $y_2 \circ f = x_2$.

    Show that $\mathbf{C}_{A,B}$ has a terminal object just in case A and B have a product in $\mathbf{C}$.

16. In the category of types $\mathbf{C}(\lambda)$ of the λ-calculus, determine the product functor $A, B \mapsto A \times B$ explicitly. Also show that, for any fixed type A, there is a functor $A \to (-) : \mathbf{C}(\lambda) \to \mathbf{C}(\lambda)$, taking any type X to $A \to X$.
17. In any category $\mathbf{C}$ with products, define the *graph* of an arrow $f : A \to B$ to be the monomorphism

$$\Gamma(f) = \langle 1_A, f \rangle : A \rightarrowtail A \times B$$

(why is this monic?). Show that for $\mathbf{C} = \mathbf{Sets}$ this determines a functor $\Gamma : \mathbf{Sets} \to \mathbf{Rel}$ to the category $\mathbf{Rel}$ of relations, as defined in the exercises to chapter 1. (To get an actual relation $R(f) \subseteq A \times B$, take the image of $\Gamma(f) : A \rightarrowtail A \times B$.)
18. Show that the forgetful functor $U : \mathbf{Mon} \to \mathbf{Sets}$ from monoids to sets is representable. Infer that U preserves all (small) products.