Math 395: Category Theory
Northwestern University, Lecture Notes

Written by Santiago Cañez

These are lecture notes for an undergraduate seminar covering Category Theory, taught by the author at Northwestern University. The book we roughly follow is "Category Theory in Context" by Emily Riehl. These notes outline the specific approach we're taking in terms the order in which topics are presented and what from the book we actually emphasize. We also include things we look at in class which aren't in the book, but otherwise various standard definitions and examples are left to the book. Watch out for typos! Comments and suggestions are welcome.

Contents

Introduction to Categories

Category theory provides a framework through which we can relate a construction/fact in one area of mathematics to a construction/fact in another. The goal is an ultimate form of abstraction, where we can truly single out what about a given problem is specific to that problem, and what is a reflection of a more general phenomenom which appears elsewhere. Practically this is done by phrasing as many constructions/facts as possible in terms of "arrows", where the point is that by doing so relations between different objects of mathematics are simpler to see.

Definition of a Category. The precise definition of a category is spelled-out in the book in Section 1.1. The key points is that a category consists of a collection of "objects", a collection of "morphisms" between objects, and a way to "compose" morphisms to get other morphisms in a way which is associative and admits identities.

Standard examples. The main examples of categories we'll care about can also be found in the book, such as:

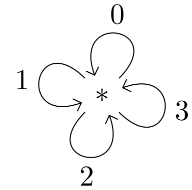
- **Set**, the category of sets where morphisms are given by ordinary functions,
- **Grp**, the category of groups where morphisms are given by group homomorphisms,
- **Top**, the category of topological spaces where morphisms are given by continuous maps,
- **Vect**, the category of vector spaces over some fixed field (which should be included as part of the notation) where morphisms are given by linear transformations.

Another example. All of the above are examples of what are called *concrete* categories. This is a term we'll define later, but it essentially means that the objects of the categories above are all sets possibly equipped with additional structure, and that the morphisms are just ordinary functions between sets possibly satisfying some additional condition. (For instance, an object in **Grp** is a set equipped with a group operation, and a morphism in **Grp** is simply a function $f : G \to H$ between groups which satisfies the additional condition that it be a group homomorphism.)

But, even though these types of examples are the main ones we'll care about, a category can be a much more general type of object. For instance, let us describe the category called $B\mathbb{Z}$. This category has only one object, which we will denote by $*$ and think of as a single "point". According to the definition of a category, we should then have, for this one object, a collection of morphisms $\mathrm{Mor}(*, *)$, which in this case we declare to be $\mathbb{Z}$:

$$\mathrm{Mor}(*, *) := \mathbb{Z}.$$

That is, we interpret each element of $\mathbb{Z}$ as being a "morphism" from $*$ to $*$. Now, of course, there is only one possible function from $*$ to $*$—namely the one that sends the single element $*$ of the domain to the single element $*$ of the codomain—so here we are considering "morphisms" which do not represent ordinary functions. Visually, we interpret each morphism as an arrow beginning and ending at the single object $*$:



Now, in order for this to be category we also need a notion of "composition", which in this case since there is only one object amounts to a map of the form:

$$\mathrm{Mor}(*,*) \times \mathrm{Mor}(*,*) \to \mathrm{Mor}(*,*),$$

which in our case becomes

$$\mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}.$$

We take this map to be ordinary addition $+$ on $\mathbb{Z}$. (So, "composing" the arrow labeled 1 in the picture above with the arrow labeled 2 results in the arrow labeled 3.) The point is that this is NOT "composition of functions" in the usual sense, but will give a valid "composition" operation in the category $B\mathbb{Z}$ as long as it satisfies the properties required in the definition of a category. One such requirement is that composition be associative, which is satisfied since $+$ is indeed associative. The second requirement is that exist a morphism $id_* \in \mathrm{Mor}(*,*)$ which acts as an identity for the "composition" we've defined. In this case, this identity morphism is $id_* = 0$, that is the integer 0 in $\mathrm{Mor}(*,*) = \mathbb{Z}$. Saying that this is the "identity morphism" with respect to "composition" in this case simply says that is the identity element for addition.

Thus we get a category $B\mathbb{Z}$ as claimed, albeit one where the notion of "morphism" and "composition" are more general than simply "function" and ordinary composition. In fact, we can already now give a "categorical" characterization of a well-known algebraic object: a (locally small) category one with a single object is precisely what is known as a *monoid*, which is a set equipped with an associative binary operation which admits an identity element. The monoid in question is precisely the set $\mathrm{Mor}(*,*)$, and the associative operation which turns this into a monoid is the "composition" of the category in question. (A *locally small* category is one where each collection of morphisms is actually a set, which is needed here in order to guarantee that $\mathrm{Mor}(*,*)$ is a set on which we can define a binary operation.) So for instance, any group can be viewed as a category with one object, where the group elements give the "morphisms" from that one object to itself. (Of course, a group gives a special type of category where every morphism is in fact "invertible". A small category where every morphism is invertible is called a *groupoid*, so a group is nothing but a groupoid with one object. We'll talk more about groupoids later on.)

A final example. Here is one final example we'll consider from time to time, which is again a category where the "morphisms" are not simply functions. The category **Rel** of *relations* has as objects ordinary sets, but as morphisms relations between sets:

$$f \in \mathrm{Mor}(A,B) \text{ is a relation from } A \text{ to } B\text{, which is a subset } f \text{ of } A \times B.$$

The usual *graph* of a function from A to B gives an example of such a relation, so we can say that **Rel** contains **Set** as a subcategory, but not all relations come from graphs of functions. In general, we can think of a relation as a "partially-defined, possibly multivalued function" if we interpret (a,b) being in the relation f as saying that "f sends a to b".

Composition in **Rel** comes from ordinary composition of relations: given a relation $f : A \to B$ and a relation $g : B \to C$, we define their composition $g \circ f : A \to C$ to be:

$$g \circ f := \{(a,c) \in A \times C \mid \text{there exists } b \in B \text{ such that } (a,b) \in f \text{ and } (b,c) \in g\}.$$

So, $g \circ f$ consists of all pairs of elements of A and C which are "related" to a common element of B. In the case where f and g are graphs of ordinary functions, this composition of relations reproduces the ordinary notion of composing functions: if $(a,b) \in f$ and $(b,c) \in g$, then $b = f(a)$

and $c = g(b)$ (here we abuse notation by denoting the function of which f is graph by the symbol f itself), so

$$c = g(b) = g(f(a)) = (g \circ f)(a),$$

meaning that $(a, c) \in g \circ f$ precisely when $c = (g \circ f)(a)$. Composition of relations is associative, and you can check that graphs of ordinary identity functions:

$$\text{graph}(id_A) = \{(a, a) \in A \times A \mid a \in A\}$$

give the identity morphisms. (I spent much time as graduate student trying to do "geometry" with a certain category of relations, so this is an example near and dear to my heart.)

Special Morphisms, Products

Isomorphisms. A morphism $f \in \text{Mor}(A, B)$ between two objects A and B in a category is an *isomorphism* or is *invertible* if it has an inverse: there exists a morphism $g \in \text{Mor}(B, A)$ such that $gf = id_A$ and $fg = id_B$, where $id_A \in \text{Mor}(A, A)$ and $id_B \in \text{Mor}(B, B)$ are the identity morphisms which are assumed to exist as part of the definition of a category. We say that A and B are *isomorphic*, and we write $A \cong B$, if there exists an isomorphism between them.

This general categorical notion of isomorphism reduces to the one you would expect in the standard examples: isomorphisms in **Set** are bijective functions, in **Grp** they are bijective group homomorphisms, in **Vect** they are invertible linear transformations, and in **Top** they are homeomoprhisms, i.e. continuous bijections with continuous inverses.

A category of metric spaces. Now, consider the category $\mathbf{Met}_C$ of metric spaces where morphisms are given by continuous functions. In this category the notion of "isomorphism" again means homeomorphism, so two metric spaces are isomorphic in this category when they are homeomorphic. However, often when considering whether or not two metric spaces are the "same" we have another definition in mind, that of two metric spaces being *isometric*: X is *isometric* to Y if there exists a bijective *isometry* between them, which is a bijective map $f : X \to Y$ which preserves distance in the sense that

$$d_Y(f(p), f(q)) = d_X(p, q) \text{ for all } p, q \in X.$$

After all, in a metric space you have more data than simply "open sets" (which is all that a homeomorphism detects), so it would be good to have a notion of "sameness" which incorporated the extra data of distance as well.

The goal is then to construct a category of metric spaces where "isomorphic" indeed means isometric, and we can do so simply by restricting the types of continuous maps we consider. The category **Met** has metric spaces as objects, but now a morphism $f : X \to Y$ is taken to be a function which does not "enlarge" distance in the sense that

$$d_Y(f(p), f(q)) \leq d_X(p, q) \text{ for all } p, q \in X.$$

Maps with this property are often called *metric maps*. Any metric map is actually continuous, so **Met** is a subcategory of $\mathbf{Met}_C$. (The subscript C used in $\mathbf{Met}_C$ is meant to denote "continuous", indicating that we take arbitrary continuous maps as morphisms.) In this category now, for $f : X \to Y$ to be an isomorphism requires that f and its inverse f^{-1} both be metric maps, so:

$$d_Y(f(p), f(q)) \leq d_X(p, q) \text{ and } d_X(f^{-1}(s), f^{-1}(t)) \leq d_Y(s, t)$$

for all $p, q \in X$ and $s, t \in Y$. But writing s and t as $s = f(p)$ and $t = f(q)$, the second condition becomes $d_X(p,q) \leq d_Y(f(s), f(t))$, and the two inequalities together thus imply

$$d_Y(f(p), f(q)) = d_X(p,q) \text{ for all } p, q \in X$$

so that $f : X \to Y$ is actually an isometry. The moral is that certain properties one might want can be achieved by changing the morphisms you consider.

Monomorphisms and epimorphisms. The definitions of *monomorphism* and *epimorphism* can be found in Section 1.2 of the book. The standard examples are those in **Set**, where the monomorphisms are the injective functions and the epimorphisms are the surjective functions. Proofs of these facts are left to the homework, as well as the interesting problem of determining the epimorphisms in **Met**, or even $\mathbf{Met}_C$, or **Haus** (the category of Hausdorff spaces where morphisms are arbitrary continuous maps), where the answer is that epimorphism means more than simply "continuous and surjective". (As opposed to all of **Top**, where epimorphism does indeed mean continuous and surjective.)

So, monomorphisms and epimorphisms should be treated as categorical analogues of "injections" and "surjections", although this analogy only goes so far. The thing to note is that we are thus able to give a complete definition of "injective" for ordinary functions which makes no reference to elements and is stated only in terms of the relation between an injective function and other functions; certainly proving that "monomorphism" means "injection" in **Set** requires working with elements, but stating the definition of an injection as a monomorphism can be done solely using "arrows".

Products. The definition of a *product* in a category shows up in Section 3.1 of the book, in the context of the more general notion known of a *limit*. We'll discuss this more general notion eventually, but for now we will only focus on products of two objects at a time. Here is what the more general notion boils down to in this case, which we'll take as our definition:

> A *product* of two objects A and B in a category is an object P together with morphisms $p_A : P \to A$ and $p_B : P \to B$ (which we call *projections*) which satisfy the following property: given any morphisms $f : C \to A$ and $g : C \to B$ into A and B from a common object C, there exists a unique morphism $h : C \to P$ such that $f = p_A \circ h$ and $g = p_B \circ h$. We express this by saying that the following diagram *commutes*:
>
> $$\begin{tikzcd} & C \arrow[d, dashed, "\exists! h"] \arrow[ddl, bend right, "f"'] \arrow[ddr, bend left, "g"] & \\ & P \arrow[dl, "p_A"'] \arrow[dr, "p_B"] & \\ A & & B \end{tikzcd}$$
>
> which means that composing any two composable arrows in the diagram results in the other arrow with the same domain and codomain. Concretely, there are two ways in this diagram to get from C to A, via f or via the composition $p_A \circ h$, and the requirement is that both give the same morphisms (i.e. $f = p_A \circ h$) and similarly for the two ways to get from C to B. The dashed arrow is the one which is required to exist (uniquely as denoted by the exclamation mark !) in the definition of product.

So, the upshot is that for any C and any f and g in the diagram above, there exists a unique h which makes the diagram commute. The point is that a product gives a way to turn the two pieces of data $f : C \to A$ and $g : C \to B$ into the single piece of data $h : C \to P$ in a unique way. Being able to turn a possibly large amount of data into a single piece of data in this way is one of the benefits the "categorical" perspective provides. Products in categories are often denoted using the usual $A \times B$ notation, even though it is not necessarily true that a product is literally a set-theoretic Cartesian product, although it is in the standard examples. The map $h : C \to P$ required in the definition is then often denoted by $h = f \times g$.

Examples. In **Set**, the product of two objects A and B always exists and is indeed the ordinary Cartesian product $A \times B$. But of course, the definition of product also requires that data of the projection morphisms, which in this case are simply the usual ones:

$$p_A : (a, b) \mapsto a \text{ and } p_B : (a, b) \mapsto b.$$

Given $f : C \to A$ and $g : C \to B$, the map $h : C \to A \times B$ is $h(c) = (f(c), g(c))$, and you can check that these constructions satisfy the properties required in the definition. (Note that you would also have to argue that h defined in this way is the *only* map which can make the required diagram commute.)

Products in **Grp** are given by the usual direct product group structure: $G \times H$ with the group operation defined as

$$(g_1, h_1) \cdot (g_2, h_2) := (g_1 g_2, h_1 h_2).$$

The projection morphisms are the ordinary ones you would expect as well. But let us actually now verify that this group structure indeed is, among other possible group structures on $G \times H$, the one required in the definition of a product. Suppose $f : K \to G$ and $g : K \to H$ are group homomorphisms. The key point is that the map $h(k) = (f(k), g(k))$ which should satisfy the required product property should itself be a morphism in **Grp**, meaning a group homomorphism, and we claim that this is only true for the direct product group structure.

Indeed, we want it to be true that

$$h(k_1 k_2) = (f(k_1 k_2), g(k_1 k_2)) \text{ equal } h(k_1)h(k_2) = (f(k_1), g(k_1)) \cdot (f(k_2)g(k_2))$$

for all $k_1, k_2 \in K$ and some to-be-determined group structure $\cdot$ on $G \times H$. Since f and g are homomorphisms, this requirement becomes

$$(f(k_1)f(k_2), g(k_1)g(k_2)) = (f(k_1), g(k_1)) \cdot (f(k_2)g(k_2)),$$

which says that $\cdot$ should indeed be componentwise multiplication, *at least* on the *image* of f and g. But recall that the properties required of a product should hold for *any* K and *any* $f : K \to G$ and $g : K \to H$; in particular then, it should hold for instances when f and g are surjective, which shows that the componentwise-multiplication property above should in fact hold on all of $G \times H$, so that $\cdot$ is indeed the ordinary direct product structure.

Uniqueness. Notice that in the definition of a product in a category, we used the word "a" as opposed to "the": we spoke about *a* product of A and B instead of *the* product of A and B. So we now ask: are products unique? The answer is "no" if we interpret "unique" in the strictest possible sense, but "yes" if we interpret it correctly. For instance, in the example of **Grp**, take now N to be *any* group which is isomorphic to the direct product $G \times H$, say with isomorphism $\ell : N \to G \times H$. We claim that we can also turn N into a categorical product of G of H, as long as we define appropriate "projections" $N \to G$ and $N \to H$: take $N \to G$ to be the composition $p_G \circ \ell$ and $N \to H$ to be $p_H \circ \ell$. Given $f : K \to G$ and $g : K \to H$ as in the definition of a product, the map $h' : K \to N$ defined by

$$h' = \ell^{-1} \circ h$$

where $h : K \to G \times H$ is the usual $h = f \times g$ will satisfy the requirement needed to able to conclude that N is also a product of G and H in **Grp**. A similar trick works in any category, so that any object isomorphic a product can also itself be turned into a product.

The correct answer is that products are *unique up to unique isomorphism*, which means that given a product P of A and B, any other product P' of A and B will in fact be isomorphic to P, *and* in a unique way in the sense that there can exist only one isomorphism $P \to P'$. The phrase "unique up to unique isomorphism" is a common one we'll see again and again: objects defined by some categorical property are almost never genuinely unique in a strict sense, but they will essentially be as close to unique as possible.

So, suppose that P and P' are both products of A and B, with associated projection morphisms p_A, p_B for P and p'_A, p'_B for P'. Consider the diagram:

$$\begin{tikzcd}
 & P \arrow[d, dashed, "\exists! h"] \arrow[ddl, bend right, "p_A"'] \arrow[ddr, bend left, "p_B"] & \\
 & P' \arrow[dl, "p'_A"'] \arrow[dr, "p'_B"] & \\
A & & B
\end{tikzcd}$$

The morphism $h : P \to P'$ is the unique one guaranteed to exist by the fact that P' is a product of A and B. Commutativity of this diagram says that $p_A = p'_A \circ h$ and $p_B = p'_B \circ h$. A similar diagram with the roles of P and P' reversed which uses the fact that P is a product results in a unique morphism $h' : P' \to P$ satisfying $p'_A = p_A \circ h'$ and $p'_B = p_B \circ h'$.

Now, consider the following diagram:

$$\begin{tikzcd}
 & P \arrow[d, "h' \circ h"] \arrow[ddl, bend right, "p_A"'] \arrow[ddr, bend left, "p_B"] & \\
 & P \arrow[dl, "p_A"'] \arrow[dr, "p_B"] & \\
A & & B
\end{tikzcd}$$

This commutes, which we verify using the properties h and h' satisfy as follows:

$$p_A \circ (h' \circ h) = (p_A \circ h') \circ h = p'_A \circ h = p_A$$

and similar for the morphisms involving B. This shows that $h' \circ h : P \to P$ satisfies the requirement in the definition of P being a product. But of course, $id_P : P \to P$ satisfies this requirement as well, so since the map satisfying this requirement is assumed to be unique, we must have $h' \circ h = id_P$. Similar reasoning considering $h \circ h' : P' \to P'$ shows that $h \circ h' = id'_P$, so that h and h' are indeed inverses and hence that P and P' are isomorphic. The uniqueness of the isomorphism between them comes from the uniqueness of h in the construction above.

Abstract nonsense. It is common to say in the setting above that the fact that products are unique up to unique isomorphism follows from "abstract nonsense", which is a succinct way of saying that it follows solely from the definition of a categorical product itself and not any deeper mathematical content. In particular, many of the "uniqueness" properties which "products" have in the settings we might care about—Cartesian products in **Set**, direct products in **Grp**, product topologies in **Top**—really have nothing to do with how those specific products are defined, but are just consequences of the "abstract nonsense" of category theory.

But of course, we do not use the phrase "abstract nonsense" in a negative or dismissive way, since recognizing what types of properties are consequences of "abstract nonsense" goes a long way towards understanding well the given problem at hand. Soon, you too, will come to appreciate this notion of abstract nonsense.

Coproducts, Opposite Categories

Initial and terminal objects. The notions of *initial* and *terminal* objects in a category are defined in Section 1.6 of the book, where you can also find standard examples. Here I want to point out that initial and terminal objects, if they exist, are unique up to unique isomorphism, as another example of arguing via "abstract nonsense".

Suppose I and I' are both initial objects in some category. By the fact that I is initial there exists a unique morphism $f : I \to I'$, and by the fact that I' is initial there exists a unique morphism $g : I' \to I$. But then the composition $g \circ f : I \to I$ must equal $id_I : I \to I$ since the latter is the only morphism from I to itself because I is initial, and similarly $f \circ g = id_{I'}$. Thus I and I' are unique, and there is a unique isomorphism between them.

Coproducts. The dual concept to that of a product is the notion of a *coproduct.* (The term "dual" refers to the phenomena you get when you reverse all arrows in a given construction. For instance, the notion of a terminal object is dual to that of an initial object, and the notion of an epimorphism is dual to that of a monomorphism.) Here is the precise definition: a *coproduct* of an object A and an object B in a category is an object C together with morphisms $i_A : A \to C$ and $i_B : B \to C$ such that for any object D and morphisms $f : A \to D$ and $g : B \to D$, there exists a unique morphism $h : C \to D$ which makes the following diagram commute:

$$\begin{tikzcd} & D & \\ & C \arrow[u, dashed, "\exists! h"'] & \\ A \arrow[ur, "i_A"] \arrow[uur, bend left, "f"] & & B \arrow[ul, "i_B"'] \arrow[uul, bend right, "g"'] \end{tikzcd}$$

By abstract nonsense, if a coproduct of A and B exists it is unique up to unique isomorphism.

Examples. In **Set**, the coproduct of two sets A and B is the *disjoint union* $A \sqcup B$. (If you haven't seen the notion of a disjoint union before, it is essentially the same as the union, only that we don't care about whether A and B have any overlap: if they do, we treat the common elements as being "different". For instance, $\mathbb{Z} \sqcup \mathbb{Z}$ is not $\mathbb{Z}$, but consists of two copies of each integer; one 0 comes from the first copy of $\mathbb{Z}$, and another comes from the second, but these are different "zeroes".) Given $f : A \to D$ and $g \to D$, the unique map $A \sqcup B \to D$ is defined by applying f to elements of

A and g to elements of B. (Note that if we simply took the ordinary union $A \cup B$ as the candidate for the coproduct when A and B were not disjoint, the required map $h : A \cup B \to D$ would not exist since h would have to send any element $x \in A \cap B$ to both $f(x)$ and $g(x)$ in order to make the required diagram commutative.)

The coproduct of two objects X and Y in **Top** is again the disjoint union $X \sqcup Y$, equipped with the *disjoint union* topology: an open subset of $X \sqcup Y$ is defined to be the (disjoint) union of an open subset of X with an open subset of Y. The coproduct of two groups G and H in **Grp** is the *free product* $G * H$, which is further elaborated on in the homework.

The coproduct of two vector spaces V and W in **Vect** is the direct sum $V \oplus W$, which is defined to be the set of formal sums

$$v + w \text{ where } v \in V \text{ and } w \in W$$

where addition and scalar multiplication are defined "componentwise":

$$(v_1 + w_1) + (v_2 + w_2) := (v_1 + v_2) + (w_1 + w_2) \text{ and } r(v + w) = rv + rw.$$

This is isomorphic to the product $V \times W$ with operations also defined componentwise, but it is customary to speak about the "direct sum" $V \oplus W$ when referring to the coproduct as opposed to the product; we'll say why in a bit. The maps $i_V : V \to V \oplus W$ and $i_W : W \to V \oplus W$ required as part of the data of a coproduct are:

$$i_V : v \mapsto v + 0_W \text{ and } i_W : w \mapsto 0_V + w.$$

To verify that $V \oplus W$ is the correct coproduct, take any linear transformations $S : V \to U$ and $T : W \to U$ where U is some other vector space. We then need a (unique) linear transformation $h : V \oplus W \to U$ such that

$$S = h \circ i_V \text{ and } T = h \circ i_W.$$

Let us determine what this map must be, thereby not only showing that it exists but also that it is unique. Let $v + w \in V \oplus W$, which we can write as

$$v + w = (v + 0_W) + (0_V + w).$$

Since h is required to be linear, we have:

$$h(v + w) = h([v + 0_W] + [0_V + w]) = h(v + 0_W) + h(0_V + w).$$

The first term at the end is $(h \circ i_V)v$, which must thus be Sv, and the second term is $(h \circ i_W)w$, which must be Tw. Thus $h = S + T$ is the map

$$h = S + T : v + w \mapsto Sv + Tw,$$

which you can verify is indeed linear. Thus $V \oplus W$ does satisfy the property required of a coproduct.

So, the product of two objects in **Vect** is the same as their coproduct, only that we write the latter using "sum" notation. Later we will discuss products and coproducts of an arbitrary number of elements. We will see these as special cases of the general notions of *limits* and *colimits*, but they are defined via similar requirements as products and coproducts of two objects at a time. In the category of vector spaces, arbitrary products are given by the usual product $\prod_\alpha V_\alpha$ with "componentwise" addition and scalar multiplication, but it turns out that an arbitrary coproduct is given by the direct sum $\bigoplus_\alpha V_\alpha$, which is defined to be the subspace of $\prod_\alpha V_\alpha$ where only finitely many components are nonzero. (This is in a way analogous to how the product topology on an arbitrary number of topological spaces is defined.) We'll go over this later, but is the underlying reason why we use $\oplus$ instead of $\times$ when discussing coproducts.

Encoding data via products/coproducts. We can nicely summarize the definition of products and coproducts in the following way: the product of A and B is the object with the property that for any object D we have:

$$\mathrm{Mor}(D, A) \times \mathrm{Mor}(D, B) = \mathrm{Mor}(D, \text{product})$$

and the coproduct of A and B is the object such that for any object D we have:

$$\mathrm{Mor}(A, D) \times \mathrm{Mor}(B, D) = \mathrm{Mor}(\text{coproductx}, D).$$

The first reflects the fact that products give a unique way to turn two morphisms (f and g in the definition) mapping into A and B into a single morphism (h in the definition), and the second says that coproducts give a unique way to turn two morphisms mapping out of A and B into a single morphism.

Soon enough we will discuss the idea of *universality*, and the point is that products are "universal" for maps *into* A and B, and coproducts are universal for maps *from* A and B.

Opposite categories. Finally, we give a definition which might seem silly, but is actually quite useful. Given a category $\mathcal{C}$, the *opposite category* is the category $\mathcal{C}^{op}$ is the category with the same objects as $\mathcal{C}$, but where we defined $\mathrm{Mor}(A, B)$ in $\mathcal{C}^{op}$ to be $\mathrm{Mor}(B, A)$ in $\mathcal{C}$. To be more concrete, we consider each morphism $f : A \to B$ in $\mathcal{C}$ to instead be a morphism $f^{op} : B \to A$ in $\mathcal{C}^{op}$. (Perhaps it would be better to write this as $A \leftarrow B : f^{op}$.) The point is that we don't actually care about what f actually is, as, say a function/continuous map/homomorphism/whatever-we only care about its behavior as an "arrow". The opposite category $\mathcal{C}^{op}$ only retains information about how arrows relate to one another, nothing more.

Since arrows are reversed, products in $\mathcal{C}$ become coproducts in $\mathcal{C}^{op}$, and coproduts in $\mathcal{C}$ become coproducts in $\mathcal{C}^{op}$. Similarly, initial/terminal objects and monomorphisms/epimorphisms in $\mathcal{C}$ become terminal/initial objects and epimorphisms/monomorphisms in $\mathcal{C}^{op}$. Again, the notion of an opposite category might seem like a strange thing to look at, but we'll see that it does have uses.

Functors, Fullness and Faithfulness

One more product/coproduct example. Let X be a topological space and define its *category of open sets* $\mathrm{Op}(X)$ as follows. The objects of $\mathrm{Op}(X)$ are simply the open subsets of X, and the morphisms are given by inclusions; so to be concrete:

$$\mathrm{Mor}(U, V) = \begin{cases} \{\text{the unique inclusion } U \hookrightarrow V\} & \text{if } U \subseteq V \\ \emptyset & \text{otherwise.} \end{cases}$$

The fact that $U \subseteq V$ and $V \subseteq W$ implies $U \subseteq W$ says that the composition of two morphisms in this category is still a morphism in this category. The product of U and V in this category turns out to be their intersection (with "projection morphisms" $U \cap V \to U$ and $U \cap V \to V$ being inclusions), and the coproduct of U and V turns out to be their (ordinary) union. Note that we need X to be a topological space in order to guarantee that $U \cap V$ and $U \cup V$ are still objects in this category. More generally, the product of *finitely* many objects in $\mathrm{Op}(X)$ exists and is their intersection, and the coproduct of an *arbitrary* number of objects exists and is their union. (We say that $\mathrm{Op}(X)$ has finite products and arbitrary coproducts.)

Given a set X and a collection $\mathcal{T}$ of subsets of X, we can define a similar category with elements of $\mathcal{T}$ as objects (I guess we should assume that $\emptyset, X \in \mathcal{T}$) and inclusions as morphisms. We then have that this category has finite products and arbitrary coproducts if and only if $\mathcal{T}$ is a topology on X, so that we can rephrase the definition of a topological space itself solely in categorical terms. This perspective on what a "topology" is makes it simpler to state the definition of what's called a *sheaf* on a topological space, which is an important construction in various areas of analysis and geometry. We might discuss this a bit later on.

Functors. As with most things in mathematics, we should care not only about categories as mathematical structures on their own, but also about "maps" between these structures which "preserve" said structure. The notion of a *(covariant) functor* between categories provides such "maps" in the case at hand, where a functor is a way of sending objects to objects and morphism to morphisms, in a way which preserves compositions and identities. The precise definition is in Section 1.3 of the book, as is the definition of a *contravariant functor* between categories, which is a functor which reverses the direction of arrows, as opposed to "covariant" ones which preserve directions. (It is common to use the term "functor" without qualification to mean one which is covariant. Note that any functor can be made covariant by working with the opposite category: a contravariant functor $\mathcal{C} \to \mathcal{D}$ is the same as a covariant functor $\mathcal{C}^{op} \to \mathcal{D}$.)

All of the standard examples we looked at are in the book: forgetful functors, power set functors, the dual space functor in **Vect**, etc. Of particular interest is Example 1.3.9 in the book, which shows that various types of group actions in different contexts-the usual notion of a group action on a set, the notion of a continuous action of a group on a topological space, and the notion of a linear action of a group on a vector space (also known as a *representation* of the group)-are all encoded by a functor with domain category BG.

Adjoint functors. Let $F : \mathbf{Top} \to \mathbf{Set}$ be the forgetful functor, and let $D : \mathbf{Set} \to \mathbf{Top}$ be the functor which sends a set S to the space S equipped with the discrete topology and a function $S \to S'$ between sets to the same function only now viewed as a continuous map $S \to S'$ between discrete spaces. These two functors satisfy the following property in relation to one another: giving a morphism $D(S) \to Y$ in **Top** is the same as giving a morphism $S \to F(Y)$ in **Set**, or more symbolically

$$\mathrm{Mor}_{\mathbf{Top}}(D(S), Y) = \mathrm{Mor}_{\mathbf{Set}}(S, F(Y)).$$

We say that D is *left adjoint* to F, and that F is *right adjoint* to D. The point is that adjoint functors can be used to move data back and forth between two categories.

We'll talk more about adjoint functors later on, but for now here is one more example. Now take $F : \mathbf{Grp} \to \mathbf{Set}$ to again be the forgetful functor, and take $G : \mathbf{Grp} \to \mathbf{Set}$ to be the *free group* functor which sends a set S to the free group generated by S: $G(S)$ has as elements "words" $s_1 \ldots s_n$ made up of elements of S with the group operation being "concatenation". A basic fact is that giving a group homomorphism from $G(S)$ to some other group H is the *same* as giving an ordinary function from S to $F(H)$, which is the required adjoint property:

$$\mathrm{Mor}_{\mathbf{Grp}}(G(S), H) = \mathrm{Mor}_{\mathbf{Set}}(S, F(H)).$$

Faithful and full functors. The definitions of what it means for a functor to be *faithful* and *full* are in Section 1.5 of the book, where faithful functors are ones which are injective on morphisms and full functors are ones which are surjective on morphisms, but where in both cases we refer only to morphisms which occur between the objects $F(A)$ and $F(B)$ in the target category corresponding to fixed objects A and B in the source category. (In general, a functor could send two objects A, B in the source category to the same object $F(A) = F(B)$ in the target category, and thus there could be two morphisms $A \to C$ and $B \to C$ which are sent to the same morphism $F(A) = F(B) \to F(C)$. Such a thing could still possibly be "faithful" since the non-injectivity here is arising from *different* objects in the source.)

Faithfullness and fullness will be nice properties going forward, and here is one hint at how they might be useful. Say we have a functor $F : \mathcal{C} \to \mathcal{D}$. The problem is to determine properties on F which will imply it send products in $\mathcal{C}$ to products in $\mathcal{D}$, or coproducts in $\mathcal{C}$ to coproducts in $\mathcal{D}$. So, given the former product diagram below, we want the second diagram to also be a product diagram:

$$\begin{tikzcd}
 & C \arrow[d, dashed, "\exists! h"] \arrow[ddl, bend right, "f"'] \arrow[ddr, bend left, "g"] & & & & F(C) \arrow[d, dashed, "Fh"] \arrow[ddl, bend right, "Ff"'] \arrow[ddr, bend left, "Fg"] & \\
 & P \arrow[dl, "p_A"'] \arrow[dr, "p_B"] & & \arrow[r, "F"] & \; & F(P) \arrow[dl, "F(p_A)"'] \arrow[dr, "F(p_B)"] & \\
A & & B & & F(A) & & F(B)
\end{tikzcd}$$

Of course, in order $F(P)$ to be an actual product of $F(A)$ and $F(B)$ on the right, a similar commutative diagram property would have to hold for *all* objects in $\mathcal{D}$ in place of $F(C)$ and all morphisms $D \to F(A)$ and $D \to F(B)$, not just those of the form Ff and Fg respectively. One way to guarantee this is to require that all objects in $\mathcal{D}$ are of the form $F(C)$ and then for all morphisms $F(C) \to F(A)$ and $F(C) \to F(B)$ to be of the form Ff, Fg; the former is true when the functor F is *surjective on objects*, and the latter is true when F is full. We are not saying that these are the *only* conditions under which F might preserve products, but it definitely seems to give a sufficient condition at least.

Almost, there is one more thing to require: the uniqueness of the morphism $Fh : F(C) \to F(P)$ making the diagram on the right commute. Suppose there were two such morphisms Fh and Fh', where h and h' are both morphisms $C \to P$ in $\mathcal{C}$. The commutativity of the diagram on the right gives, for instance:

$$Ff = F(p_A)Fh \text{ and } Ff = F(p_A)Fh',$$

which the functorial properites of F turns into

$$Ff = F(p_A \circ h) \text{ and } Ff = (p_A \circ h').$$

We want h and h' to make the first diagram commute, in order to be able to apply uniqueness of the morphism $C \to P$. But this requires know that $f = p_A \circ h$ and $f = p_A \circ h'$, which would follow from F being *faithful*. In this case, it follows that h and h' both make the first diagram commute (after we go through a similar argument with g), so the fact that P is a product guarantees that $h = h'$, and hence that $Fh = Fh'$ so that $F(P)$ is an actual product on the right.

So, we conclude that a functor which is full, faithful, and surjective on objects will send products to products. (Actually, we can relax this a bit by requiring not that every object in $\mathcal{D}$ be in the literal image of F, but only that it be isomorphic to something in the image. A functor with this property is said to be *essentially surjective*, which is a concept will see in a related context soon enough.) Again, this is not an "if and only if" statement, since functors can preserve products (or coproducts) without being full, faithful, and surjective on objects, but it at least gives a sufficient condition. Later we will see other ways to guarantee that functors preserve products, which are related to the existence of adjoints for that given functor.

Coproduct Examples, Concreteness

Products and coproducts for metric spaces. The product of two objects in **Met**, the category of metric spaces with morphisms given by metric maps, is the Cartesian product $X \times Y$ equipped with the box metric, and the coproduct of X and Y does not exist if both X and Y are nonempty. This was the topic of a student presentation, and here are some proof sketches.

Given metric spaces (X, d_X) and (Y, d_Y), the box metric d on $X \times Y$ is defined by

$$d((x_1, y_1), (x_2, y_2)) = \max\{d_X(x_1, x_2), d_Y(y_1, y_2)).$$

(The name "box" metric comes from the fact that in $\mathbb{R} \times \mathbb{R} = \mathbb{R}^2$ or $\mathbb{R}^3$, open balls indeed looks like "boxes".) The key requirement needed in the definition of a product is given morphisms $S \to X$ and $S \to Y$, that the map

$$h : S \to X \times Y \text{ defined by } h(s) = (f(s), g(s))$$

actually be a morphism in **Met**. This turns into the requirement that

$$d((f(s), g(s)), ((f(t), g(t))) = \max\{d_X(f(s), f(t)), d_Y(g(s), g(t))\} \leq d_S(s, t) \text{ for all } s, t \in S$$

given the fact that

$$d_X(f(s), f(t)) \leq d_S(s, t) \text{ and } d_Y(g(s), g(t)) \leq d_S(s, t) \text{ for all } s, t \in S.$$

For the latter inequalities to imply the former, d *must* (at least up to isometry) be given by the box metric. Note that the two other "standard" metrics one might put on a product—the *taxicab* $d_X + d_Y$ and *Euclidean* $\sqrt{d_X^2 + d_Y^2}$ metrics—do not satisfy this required "metric map" property, and so do not give the product in **Met**.

To see that coproducts do not exist if X, Y are nonempty, suppose instead (P, d_P) was a coproduct for X and Y with inclusion morphisms $i_X : X \to P$ and $i_Y : Y \to P$. For any $n \in \mathbb{N}$ take $\{0, n\}$ to be a metric space with the absolute value metric d, and let $f : X \to \{0, n\}$ and $g : Y \to \{0, n\}$ be the constant function 0 and the constant function n respectively. Since P is a coproduct, there exists $h : P \to \{0, n\}$ such that

$$h \circ i_X = f \text{ and } h \circ i_Y = g.$$

But in order for h to actually be a morphism in **Met**, we must then have:

$$d(h(i_X(x), i_Y(y)) \leq d_P(i_X(x), i_Y(y)) \text{ for any } x \in X, y \in Y,$$

which translates into

$$n = |0 - n| \leq d_P(i_X(x), i_Y(y)).$$

But this should then be true for any $n \in \mathbb{N}$, which would imply that the distance in P between an element coming from X and one coming from Y should be infinite, which is nonsense. (Note that if you altered your definition of "metric" to allow for infinite distance, then a coproduct would exist: the disjoint union $X \sqcup Y$ where you defined the distance between elements of X and Y to be infinite.)

Coproducts for groups. The coproduct of two groups G and H in **Grp** is the free product $G * H$, which is defined to be the set of all words consisting of elements of G and H in an alternating fashion with concatenation as the group operation. (When we have to elements of G or two elements of H next to each other, with use their respective group operations to turn these into single elements of G or H.) This was also the topic of a student presentation, and here is the basic idea. Given group homomorphisms $g : G \to N$ and $k : H \to N$, that $G * H$ is the coproduct comes down to showing that the map

$$h : G * H \to N \text{ defined by } h(g_1 h_1 \dots g_n h_n) = f(g_1)k(h_1) \cdots f(g_n)k(h_n),$$

and similarly on other possible words in $G * H$, is a group homomorphism, which follows essentially from the definition of the group operation on $G * H$; indeed, this group operation arises precisely from this requirement.

Now, if we instead tried to use $G \times H$ with its standard inclusions $G \to G \times H$ and $H \to G \times H$ as the coproduct, the issue is that the analogous map

$$h : G \times H \to N \text{ defined by } h((g, h)) = f(g)k(h)$$

would NOT be a homomorphism, because $h((g_1, h_1)(g_2, h_2))$ is not equal to $h(g_1, h_1)h(g_2, h_2)$ by the way in which the group operation is defined on the direct product $G \times H$. However, if we strict our category to only consist of abelian groups, so that the N's we consider in the definition of coproduct are also abelian, then $G \times H$ does serve as a coproduct we can always rewrite $f(g_1)k(g_1) \cdots f(g_n)k(g_n)$ as $f(g_1) \cdots f(g_n)k(h_1) \cdots k(h_n)$ in N. I'll leave it to you to flesh out all of these details.

Concrete categories. The categories **Set**, **Top**, **Vect**, and **Grp** all have the property that their objects are simply *sets* possibly equipped with extra structure, and that their morphisms are ordinary set-theoretic *functions* possibly satisfying some additional requirement. Such categories are nice since we can often use intuition about sets and functions in their study. A *concrete category* is intuitively one where we can think of objects as being "sets" and morphisms as being "functions", only that we have to interpret this in the right way.

Here is the definition: a *concrete category* is a category $\mathcal{C}$ equipped with a faithful functor $F : \mathcal{C} \to \mathbf{Set}$, which is part of the data. The idea is that such a functor gives us way to turn an object $A \in \mathrm{Ob}(\mathcal{C})$ into a set $F(A)$ and a morphism $f \in \mathrm{Mor}_{\mathcal{C}}(A, B)$ into a function $Ff \in \mathrm{Mor}_{\mathbf{Set}}(F(A), F(B))$, where the "faithful" requirement says that we do not lose information about f when doing so, so that we can (essentially) learn everything we need to know about f by studying Ff instead. In the four examples above, the required faithful functor is simply the forgetful functor, and indeed in general we think of F as being a "forgetful functor" for the concrete category $(\mathcal{C}, F)$.

Examples. Here are two more examples of concrete categories. Let G be a group and consider the category BG with a single object and elements of G morphisms. As written, this does not appear to be concrete since morphisms are not honest functions $* \to *$, but the point is that this category can indeed by "concretized" by specifying an appropriate "forgetful functor". A functor $F : BG \to \mathbf{Set}$ is the data of a (left) action of G on a set $S := F(*)$, and saying this functor is faithful is what it means to say that this action is faithful: different $g, h \in G$ act on S in different ways, or equivalently that there exists $s \in S$ such that $gs \neq hs$. Any group has at least one faithful left action—namely the action on itself given by left multiplication—to BG can always be equipped with a faithful functor to **Set**. This is all just a way of phrasing the fact that any group is isomorphic to a subgroup of a permutation group.

The category **Rel** of relations can also be "concretized", even though as written morphisms $A \to B$ are not honest functions. Take $F : \mathbf{Rel} \to \mathbf{Set}$ to be the power set functor, which sends an object in **Rel** to its power set and a relation $f : A \to B$ to the induced function $Ff : P(A) \to P(B)$ defined by:

$$(Ff)(S) := \{b \in B \mid \text{ there exists } a \in S \text{ such that } (a, b) \in f\}.$$

In other words, if you think of the relation $f \subseteq A \times B$ as a "possibly not everywhere-defined, possibly multivalued function", $(Ff)(S)$ is analogous to taking the image of S. That F is faithful will be left to the homework.

In fact, most categories you'll see can be made concrete by specifying an appropriate forgetful functor, but not all categories can be made concrete in this way. Here is the standard example of a non-concrete category, which we mention only to say there is such a thing but which we won't do anything more with since understanding it better requires knowledge of algebraic topology. The *homotopy category* of topological spaces is the category **hTop** whose objects are topological spaces and whose morphisms are given by equivalence classes of homotopic maps:

$$\mathrm{Mor}_{\mathbf{hTop}}(X, Y) := \{\text{homotopy classes of continuous maps } X \to Y\}.$$

(Saying that two maps are homotopic means that you can "deform" one into the other, but we'll leave the precise definition to a course in algebraic topology.) It turns out that no functor from **hTop** to **Set** can be faithful, so that **hTop** cannot be made concrete, but this is actually a deep and difficult thing to show, so we make no attempt to do so here.

Natural Isomorphisms, Representability

Natural transformations. The definition of a *natural transformation* $\eta : F \Rightarrow G$ between two (both covariant or both contravariant) functors $F, G : \mathcal{C} \to \mathcal{D}$ is given in Section 1.4 of the book. The key idea is that a natural transformation gives a way to turn data about F into data about G in a way which is compatible with all possible morphisms, as expressed by the commutativity of the diagrams

$$\begin{tikzcd} F(A) \arrow[r, "Ff"] \arrow[d, "\eta_A"] & F(B) \arrow[d, "\eta_B"] \\ G(A) \arrow[r, "Gf"] & G(B) \end{tikzcd}$$

arising from morphisms $f : A \to B$ (or $f : B \to A$ in the contravariant case) in $\mathcal{C}$. In the case of a *natural isomoprhism* $\eta : F \Rightarrow G$, so that each $\eta_A : F(A) \to G(A)$ is actually an isomorphism in $\mathcal{D}$, the point is not only that F and G produce isomorphic objects, but they do so in a way which is compatible with all possible category-theoretic data.

The first standard example of a natural isomorphism is the one between the identity functor on the category of finite-dimensional and the functor which sends a finite-dimensional vector space V to V^{**}, the dual of its dual, induced by the isomorphism $V \to V^{**}$ defined by

$$v \mapsto (\text{the linear map on } V^* \text{ which sends } f \text{ to } f(v)).$$

This was the topic of a student presentation, and can be found in Section 1.4 of the book. The essential reason why this isomorphism is "natural" is that $V \to V^{**}$ can be defined without reference to a basis. Contrast this with the the functor which sends V to its (single) dual V^*: it is still true that finite-dimensional vector space is isomorphic to its dual, but specifying such an isomorphism requires additional data, which prevents the identity functor from being "naturally isomorphic" to the single dual functor.

Some history. At this point it makes sense to say a bit about the interesting origins of category theory in the 1950's. Eilenberg and Mac Lane where studying cohomology (or possible homology, one of those two) in algebraic topology, which associates algebraic data (a group, ring, or similar) to a topological space in a way which encodes some of the topology. They had two cohomological constructions which ended up proving isomorphic results, but they noticed that not only were the resulting objects the same, but the actual constructions themselves were in some sense the "same". They thus then needed a way to formalize this notion, and invented the concept of a natural transformation to do so, where they interpreted two constructions as being the same as being compatible with all morphisms.

But this then leads to the question: what types of objects should a natural transformation occur between? Eilenberg and Mac Lane then had to invent the notion of a "functor" to describe what a natural transformation went *from* and what it went *to*. In their motivating setting, they invented the notion of a functor to describe more formally properties which their cohomological constructions were supposed to satisfy. But this then leads to the question: what types of objects should a functor map between? In other words, what type of data should a functor (cohomological construction in their example) take as input and should it output? They finally had to thus invent the notion of a "category" to describe the source and target of a functor, so that their cohomological constructions could be interpreted as functors from a category of topological spaces to a category of algebraic objects.

Thus, historically, natural transformations came first, then functors, and then categories! As with most things in mathematics, categories thus arose from attempts to encode what was being seen in some concrete examples in a more general and formal setting.

Representable functors. Representable functors are defined in Section 2.1 of the book, but we'll give the required definitions here in a, hopefully, simpler to digest manner. The point is that any object A in a (locally small) category $\mathcal{C}$ gives two "natural" functors $\mathcal{C} \to \mathbf{Set}$, one covariant and the other contravariant, which, as we'll see, pretty much encode all data about A itself. We define $h^A : \mathcal{C} \to \mathbf{Set}$ to be the covariant functor defined by

$$h^A(B) := \mathrm{Mor}(A, B) \text{ on objects, and}$$

$$h^A(B \xrightarrow{f} C) := \text{ the map } \mathrm{Mor}(A, B) \xrightarrow{h^A f} \mathrm{Mor}(A, C) \text{ defined by } g \mapsto f \circ g.$$

We define $h_A : \mathcal{C} \to \mathbf{Set}$ to the contravariant functor defined by

$$h_A(B) := \mathrm{Mor}(B, A) \text{ on objects, and}$$

$$h_A(B \xrightarrow{f} C) := \text{ the map } \mathrm{Mor}(C, A) \xrightarrow{h^A f} \mathrm{Mor}(B, A) \text{ defined by } g \mapsto g \circ f.$$

(The functor h_A is called the *functor of points* of A, and we'll usually think of it as being a covariant functor $h_A : \mathcal{C}^{op} \to \mathbf{Set}$. The functor h^A does not have a standard name.) So, h^A is defined by morphisms *from* A, and h_A is defined by morphisms *into* A. We'll see later the sense in which such functors encode all information about the object A.

A functor $F : \mathcal{C} \to \mathbf{Set}$ is *representable* if is naturally isomorphic to such a functor, so $F \cong h^A$ for some A in the covariant case, and $F \cong h_A$ for some A in the contravariant case. The idea we'll build towards is that we can essentially treat such a functor as being A itself, so that representable functors can be thought as being actual objects in $\mathcal{C}$. For now, here is a key example we've seen before, which we can now formulate using the language of representability.

Given $A, B \in \mathrm{Ob}(\mathcal{C})$, let $F : \mathcal{C} \to \mathbf{Set}$ be the contravariant functor defined by

$$F(C) := \mathrm{Mor}(C, A) \times \mathrm{Mor}(C, B) \text{ and } F(C \xrightarrow{f} D) = [(g, k) \mapsto (g \circ f, k \circ f)]$$

where $(g, k) \mapsto (g \circ f, k \circ f)$ gives a map $\mathrm{Mor}(D, A) \times \mathrm{Mor}(D, B) \to \mathrm{Mor}(C, A) \times \mathrm{Mor}(C, B)$, i.e. $F(D) \to F(C)$. In order for this functor to be representable requires the existence of an object in $\mathcal{C}$ satisfying

$$\mathrm{Mor}(C, A) \times \mathrm{Mor}(C, B) \cong \mathrm{Mor}(C, \text{representing object})$$

in a way which is compatible with all morphisms, but we have actually already seen this notion elsewhere: it is essentially the definition of a product $A \times B$ for A and B! Indeed, we claim that F is naturally isomorphic to $h_{A\times B}$, as we now show.

First, to define a natural isomorphism $h_{A\times B} \cong F$ requires for each $C \in \mathrm{Ob}(\mathcal{C})$ a choice of a bijection (i.e. isomorphism in $\mathbf{Set}$)

$$\eta_C : h_{A\times B}(C) \xrightarrow{\cong} F(C),$$

which in our case looks like

$$\eta_C : \mathrm{Mor}(C, A \times B)) \xrightarrow{\cong} \mathrm{Mor}(C, A) \times \mathrm{Mor}(C, B).$$

Take this to be the map

$$(C \xrightarrow{f} A \times B) \mapsto (p_A \circ f, p_B \circ f)$$

where $p_A : A \times B \to A$ and $p_B : A \times B \to B$ are the two projection morphisms in the definition of a product. The definition of product implies that this map is invertible, since given $k : C \to A$ and $\ell : C \to B$, there exists a unique $h : C \to A \times B$ such that $k = p_A \circ h$ and $\ell = p_B \circ h$. For these bijections to define a natural isomorphism $h_{A\times B} \cong F$ requires that they be compatible with all morphisms in $\mathcal{C}$, which means that given any $g : C \to D$ in $\mathcal{C}$, the following diagram should commute:

$$\begin{tikzcd}
h_{A\times B}(C) \arrow[d, "\eta_C"] & h_{A\times B}(D) \arrow[l, "h_{A\times B}(g)"'] \arrow[d, "\eta_D"] \\
F(C) & F(D) \arrow[l, "Fg"']
\end{tikzcd}$$

which in the case at hand looks like:

$$\begin{tikzcd}
\mathrm{Mor}(C, A \times B) \arrow[d, "\eta_C"] & \mathrm{Mor}(D, A \times B) \arrow[l, "- \circ g"'] \arrow[d, "\eta_D"] \\
\mathrm{Mor}(C, A) \times \mathrm{Mor}(C, B) & \mathrm{Mor}(D, A) \times \mathrm{Mor}(D, B) \arrow[l, "(- \circ g, - \circ g)"']
\end{tikzcd}$$

Take an element $k : D \to A \times B$ in the upper right. Applying the map on top gives the element $k \circ g$ in the upper left, and then applying η_C on the left side gives

$$(p_A \circ (k \circ g), p_B \circ (k \circ g))$$

in the lower left. Instead, starting with $k : D \to A \times B$ in the upper right, applying η_D on the right gives $(p_A \circ k, p_B \circ k)$, and applying the map on the bottom gives

$$((p_A \circ k) \circ g, (p_B \circ k) \circ g)$$

in the lower left, which is the same as the previous element in the lower left by the associativity of composition. Hence the isomorphisms η_C do define a natural isomorphism between $h_{A \times B}$ and F. Again, the point is that you can then interpret the functor F as being the "same" as the product $A \times B$ since both encode the same data.

More Representable Examples

Coproducts. As in the case of products, the definition of coproducts can also be phrased in terms of a representable functor. Given objects A and B in $\mathcal{C}$, let $F : \mathcal{C} \to \mathbf{Set}$ be the covariant functor defined by

$$F(C) := \mathrm{Mor}(A, C) \times \mathrm{Mor}(B, C) \text{ and } F(C \xrightarrow{f} D) = [(g, k) \mapsto (f \circ g, f \circ k)]$$

where the latter is a function $\mathrm{Mor}(A, C) \times \mathrm{Mor}(B, C) \to \mathrm{Mor}(A, D) \times \mathrm{Mor}(B, D)$. This functor is representable if and only if a coproduct $A \sqcup B$ for A and B exists, and the representing object is that coproduct; the key is the requirement that we have bijections

$$\mathrm{Mor}(A, C) \times \mathrm{Mor}(B, C) \cong \mathrm{Mor}(A \sqcup B, C)$$

for all C, which is essentially the defining property a coproduct for A and B should have.

Power set functor, contravariant version. Let $P : \mathbf{Set} \to \mathbf{Set}$ be the contravariant power set functor, defined on objects by sending a set to its power set, and on morphisms by sending $f : A \to B$ to the preimage function $Pf : P(B) \to P(A)$ given by $S \mapsto f^{-1}(S)$. In order for this to be representable requires the existence of a set X with natural bijections

$$P(A) \cong \mathrm{Mor}(A, X)$$

for all sets A; that is, a set X so that mapping into X is the same data as specifying a subset of the domain. We take $X = \{0, 1\}$ to be a two-element set, and the required bijections

$$\eta_A : P(A) \to \mathrm{Mor}(A, \{0, 1\}) \text{ to be given by } S \mapsto \chi_S$$

where $X_S : A \to \{0, 1\}$ is the *indicator* or *characteristic* function S defined by sending elements of S to 1 and elements of $A - S$ to 0. The inverse of η_A sends a function $g : A \to \{0, 1\}$ to the preimage $g^{-1}(1) \subseteq A$.

To verify that the bijections η_A define the data of a natural isomorphism $\eta : P \Rightarrow h_{\{0,1\}}$, we check the commutativity of some diagrams. Given a function $f : A \to B$, consider:

$$\begin{tikzcd}
P(A) \arrow[d, "\eta_A"] & P(B) \arrow[l, "Pf = f^{-1}"'] \arrow[d, "\eta_B"] \\
h_{\{0,1\}}(A) = \mathrm{Mor}(A, \{0,1\}) & h_{\{0,1\}}(B) = \mathrm{Mor}(B, \{0,1\}) \arrow[l, "- \circ f"']
\end{tikzcd}$$

Take $S \in P(B)$ in the upper right. Applying the map on top gives $f^{-1}(S) \in P(A)$, and then applying the map on the left gives $\chi_{f^{-1}(S)} \in \operatorname{Mor}(A, \{0,1\})$. On the other hand, applying the map on the right to $S \in P(B)$ gives $\chi_S \in \operatorname{Mor}(B, \{0,1\})$, and then applying the map on the bottom gives $\chi_S \circ f \in \operatorname{Mor}(A, \{0,1\})$. The two resulting functions $\chi_{f^{-1}(S)}, \chi_S \circ f$ in the lower left are:

$$\chi_{f^{-1}(S)}(x) = \begin{cases} 1 & x \in f^{-1}(S) \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \chi_S(f(x)) = \begin{cases} 1 & f(x) \in S \\ 0 & \text{otherwise,} \end{cases}$$

so these are the same since $x \in f^{-1}(S)$ if and only if $f(x) \in S$. Thus the diagram above commutes, so P is naturally isomorphic to $h_{\{0,1\}}$ and hence P is representable.

Power set functor, covariant version. Now consider the covariant power set functor, still denoted by $P : \mathbf{Set} \to \mathbf{Set}$, which again sends a set to its power set but now sends a function $f : A \to B$ to the image function $Pf : P(A) \to P(B)$ defined by $S \mapsto f(S)$. In order for this to be representable there would have to exist a set X such that $P \cong h^X$, which translates into having natural bijections

$$P(S) \cong \operatorname{Mor}(X, S)$$

for all sets S. However, in this case there is no natural choice for X as there is no natural way to detect subsets of S by mapping *into* S, as opposed to the case of $P(S) \cong \operatorname{Mor}(S, \{0,1\})$.

That no such X exists can be made precise using a simple argument: when $S = \emptyset$, $P(S)$ has cardinality 1, so X would have to be empty as well since otherwise $\operatorname{Mor}(X, S)$ would have cardinality 0; but if X is empty, then $\operatorname{Mor}(\emptyset, S)$ always cardinality 1 regardless of S, which is clearly not true of $P(S)$. Thus the covariant power set functor is not representable.

Forgetful on Top. The forgetful functor $F : \mathbf{Top} \to \mathbf{Set}$ is representable. Indeed, a representing object would have to satisfy

$$X \cong \operatorname{Map}(object, X)$$

as sets for any topological space X (where Map denotes the set of continuous maps), and it is easy to see that a single point satisfies this property: a continuous map $f : pt \to X$ is completely determined by the element $f(pt)$ of X, and any such element gives a continuous map in this way. Of course, checking that $F \cong h^{pt}$ requires checking that various diagrams commute, but this is simple to work out in this case, so we omit it here.

Extracting a topology. With an eye towards the idea that the functors h^A, h_A should capture all information about an object A in a category, note that the result above shows that in **Top**, h_X first of all captures all information about X as a *set*, since the set X can be extracted from $h_X(pt) = \operatorname{Map}(pt, X)$.

We claim that h^X actually captures all information about X as a topological space, since the topology on X can be extracted from h^X in the following way. As we saw in the power set example, we have a bijection

$$P(X) \cong \operatorname{Mor}(X, \{0,1\}).$$

Now, equip $\{0,1\}$ with the topology in which $\{1\}$ is open but $\{0\}$ is not. Then a continuous map $f : X \to \{0,1\}$ is fully determined by the preimage $f^{-1}(1)$, since to be continuous only requires that this single preimage be open in X. Moreoever, any open subset of X arises in this way by taking f to be the indicator function of that open set. Thus we get a bijection

$$\operatorname{Op}(X) \cong \operatorname{Map}(X, \{0,1\})$$

where $\mathrm{Op}(X)$ denotes the set of open subsets of X, i.e. the topology on X. Hence the topology on X can be extracted from h^X as claimed, so h_X and h^X do encode all information about X.

This fact can also be interpreted as saying that the contravariant functor which sends a topological space X to its collection of open subsets (and which sends a continuous map to the map induced by taking preimages) is representable with representing object $\{0, 1\}$ equipped with the topology given above. This will be worked out in a homework problem.

Forgetful on Grp. The forgetful functor **Grp** $\to$ **Set** is also representable. In this case a representing object should be a group giving set-theoretic bijections

$$G \cong \mathrm{Hom}(object, G)$$

for all groups G, where Hom denotes the set of group homomorphisms. The group $\mathbb{Z}$ works:

$$G \cong \mathrm{Hom}(\mathbb{Z}, G)$$

since a homomorphism $\phi : \mathbb{Z} \to G$ is completely determined by $\phi(1)$ since $\mathbb{Z}$ is generated by 1, and there are no restrictions on what $\phi(1) \in G$ can actually be. The inverse of the desired bijection is thus $\phi \mapsto \phi(1)$, and it is simple to check that morphisms behave appropriately so that the forgetful functor in this setting is indeed isomoprhic to $h^{\mathbb{Z}}$.

Forgetful on Ring. The forgetful functor **Ring** $\to$ **Set** is also representable, where by **Ring** we mean the category of rings with identity elements and where morphisms (i.e. ring homomorphisms) are required to preserve identities. Here we need an object satisfying

$$R \cong \mathrm{Hom}(object, R)$$

for all $R \in \mathrm{Ob}(\mathbf{Ring})$, and we can see that $\mathbb{Z}[x]$, the ring of polynomials in the variable x with coefficients in $\mathbb{Z}$, works. Indeed, given a ring homomorphism $\phi : \mathbb{Z}[x] \to R$, the fact that ϕ is required to preserve identities fully determines $\phi(n)$ for any $n \in \mathbb{Z}$ (since $\phi(n) = n\phi(1)$), and hence since ϕ preserves multiplication its behavior is fully determined by $\phi(x)$, which can be any element of R. The inverse of the bijection we want is thus $\phi \mapsto \phi(x)$, and it will follow that this forgetful functor is isomorphic to $h^{\mathbb{Z}[x]}$.

Equivalences between Categories

Definitions iso and equiv. Now that we have a notion of a "morphism" between categories, we can talk about the sense in which two categories are the "same". The first possible definition is the "obvious" one you might expect given the definition of "isomorphism" we've seen in every other context until now: we say that $\mathcal{C}$ is *isomorphic* to $\mathcal{D}$ if there exist functors $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$ such that $G \circ F = id_{\mathcal{C}}$ and $F \circ G = id_{\mathcal{D}}$, where $id_{\mathcal{C}}$ and $id_{\mathcal{D}}$ are identity functors.

However, this definition turns out to be too restrictive. Indeed, the requirement that the given compositions *equal* identity functors says that given an object A in $\mathcal{C}$, the object $G(F(A))$ be *literally the same* as A itself, and similarly for objects in $\mathcal{D}$. But we know that two objects in a category can be thought of as being the "same" without being literally the same as long as they are isomorphic. For instance, products/coproducts are not unique in the literal sense, but only in the sense that they are isomorphic in a unique way. To take another example, the definition of a functor being representable does not say that $F(B)$ literally be the same as $h^B(A)$, only that they be isomorphic; for sure, the power set $P(S)$ of a set S is not literally the same as the set of functions $S \to \{0, 1\}$—it is only the "same" in the sense that they encode the same data.

Thus, it makes sense to relax the requirement that A and $G(F(A))$ in the definition of isomorphic categories be equal to the requirement that they be isomorphic. This leads to the following, better, definition of two categories being the "same": $\mathcal{C}$ is *equivalent* to $\mathcal{D}$ if there exist functors $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$ such that $G \circ F \cong id_{\mathcal{C}}$ and $F \circ G \cong id_{\mathcal{D}}$. So, we require only that, for instance, the functor $G \circ F$ be naturally isomorphic to the identity functor on $\mathcal{C}$, which says that for any object A, $G(F(A))$ be isomorphic to A in a natural way. We'll see that this truly is the better notion of what it means for two categories to be the "same", and it reflects the idea that all categorical properties of one category can be transported to the other.

Alternate characterization of equivalences. A functor $F : \mathcal{C} \to \mathcal{D}$ implementing an equivalence between categories is called, unsurprisingly, and *equivalence*, and the "inverse" functor $G : \mathcal{D} \to \mathcal{C}$ is still referred to as being its *inverse*, even though it is not technically an "inverse" in the sense that compositions give literal identities. But, as in the case of **Set**, where "invertible" can be characterized without reference to an inverse as "injective" and "surjective", so too can equivalences between categories be characterized without explicit reference to an inverse:

> A functor $F : \mathcal{C} \to \mathcal{D}$ is an equivalence if and only it is full, faithful, and essentially surjective.

We have seen the notions of full and faithful before, and *essentially surjective* means that any object in $\mathcal{D}$ is isomorphic to something in the image of F: for all $D \in \mathrm{Ob}(\mathcal{D})$ there exits $C \in \mathrm{Ob}(\mathcal{C})$ such that $F(C) \cong D$. Justifying this result was the topic of a student presentation, and can be found in the book in Section 1.5. (Technically, to construct an "inverse" of F requires that we work with *small* categories, but whatever.) The bulk of the real work goes into showing that the "inverse" functor constructed is actually a functor, which comes down to some abstract nonsense argument via commutative diagrams.

Equivalences preserve everything. We previously saw the notions of full, faithful, and essentially surjective back when thinking about what properties a functor could have which would guarantee it sent products to products, so the result we proved back then was that equivalences preserve products. Similarly, equivalences preserve coproducts, and the proof is basically the same after reversing arrows. A clean way of deriving this is as follows: if $F : \mathcal{C} \to \mathcal{D}$ is an equivalence, then so is $F^{op} : \mathcal{C}^{op} \to \mathcal{D}^{op}$, so since F^{op} preserves products, F preserves coproducts because coproducts in a category become products in the opposite category.

In general, an equivalence between categories will preserve whatever categorical notions you want: monomorphisms, epimorphisms, initial objects, terminal objects, etc, so that equivalent categories really do have essentially the "same" properties.

Equivalence example. Here is a first example (also found in the book) of equivalent categories which are not isomorphic. Let $\mathbf{FVect}_{\mathbb{R}}$ be the category whose objects are finite-dimensional real vector spaces and whose morphisms are linear transformations. Let $\mathbf{Mat}_{\mathbb{R}}$ be the *category of matrices* defined as follows:

- objects in $\mathbf{Mat}_{\mathbb{R}}$ are nonnegative integers,
- a morphism $n \to m$ is an $m \times n$ matrix with real entries,
- composition in $\mathbf{Mat}_{\mathbb{R}}$ is given by matrix multiplication: given $B : n \to m$ and $A : m \to k$—so B is an $m \times n$ matrix and A is a $k \times m$ matrix—the composition $A \circ B : n \to k$ is the $k \times n$ matrix AB,
- identity morphisms in $\mathrm{Mat}_{\mathbb{R}}$ are given by identity matrices.

We claim that $\mathbf{FVect}_{\mathbb{R}}$ and $\mathbf{Mat}_{\mathbb{R}}$ are equivalent. First note that they certainly cannot be isomoprhic: isomorphisms in $\mathbf{Mat}_{\mathbb{R}}$ exist only between two objects which are literally the same (i.e. an $m \times n$ matrix can be invertible only when $m = n$), but isomorphisms in $\mathbf{FVect}_n$ can exist between objects which are not literally the same. So, there are in a sense not enough objects in $\mathbf{Mat}_{\mathbb{R}}$ to allow it to be isomorphic to $\mathbf{FVect}_{\mathbb{R}}$.

To construct an equivalence, choose one and for all an isomorphism $T_V : V \to \mathbb{R}^{\dim V}$ for every object in $\mathbf{FVect}_{\mathbb{R}}$, or equivalently choose once and for all a basis for every V. Define $F : \mathbf{FVect}_{\mathbb{R}} \to \mathbf{Mat}_{\mathbb{R}}$ by:

$$V \mapsto \dim V \text{ on objects, and}$$

$$(T : V \to W) \mapsto (\text{the standard matrix of } S_W \circ T \circ S_V^{-1} : \mathbb{R}^{\dim V} \to \mathbb{R}^{\dim W}) \text{ on morphisms.}$$

Equivalently, F sends T to the matrix of T relative to the chosen bases for V and W. It is straightforward to check that this is a functor. The inverse functor $G : \mathbf{Mat}_{\mathbb{R}} \to \mathbf{Vect}_{\mathbb{R}}$ is given by:

$$n \mapsto \mathbb{R}^n \text{ on objects, and}$$

$$(m \times n \text{ matrix } A) \mapsto (\text{the linear transformation } A : \mathbb{R}^n \to \mathbb{R}^m \text{ induced by } A) \text{ on morphisms.}$$

One can check that G is a functor and that $F \circ G$ and $G \circ F$ are naturally isomorphic to identities; or instead, one can argue that F alone is full, faithful, and essentially surjective. The point is that $G(F(V))$ gives back $\mathbb{R}^{\dim V}$, so not literally V itself but only something isomorphic to V, which is why this gives an equivalence of categories and not an isomorphism.

Yoneda Lemma, Functors as Objects

Compact Hausdorff spaces and C^*-algebras. Last time we gave an example of equivalent categories which were not isomorphic, but in many ways that example is unsatisfactory: when doing linear algebra, people for sure often work with matrix representations of linear transformations as the example from last would suggest, but no one in their right mind literally thinks about such things in terms of the equivalence $\mathbf{FVect}_{\mathbb{R}} \to \mathbf{Mat}_{\mathbb{R}}$ itself. In other words, this equivalence is essentially "cooked up" for the sake of giving an example, but it does not really give any insight into the actual mathematics of linear algebra.

So, with that in mind, we will describe another equivalence which actually *does* have some real meaning in the sense of producing new ways of thinking about some mathematics. Given a compact Hausdorff space X, we let $C(X)$ denote the space of continuous, complex-valued functions on X:

$$C(X) := \{f : X \to \mathbb{C} \mid f \text{ is continuous}\}.$$

We can equip the set $C(X)$ with various additional structures. First, it is a complex vector space under ordinary addition and scalar multiplication of functions. Second, we can multiply two functions together:

$$(fg)(p) := f(p)g(p),$$

which turns $C(X)$ into what's called an *algebra* over $\mathbb{C}$. (We won't give the definition of an "algebra", but the point is that this multiplication is in some sense "compatible" with the vector space structure.) Next, any element f of $C(X)$ has a *conjugate* element $\bar{f}$ obtained by taking complex conjugates of the values of f:

$$\bar{f}(p) := \overline{f(p)}.$$

And finally, we can equip $C(X)$ with a *norm* via

$$\|f\| := \sup_{p \in X} |f(p)|$$

where $|\cdot|$ denotes complex absolute value. (This supremum exists since X is compact by the Extreme Value Theorem.) All of these structures (vector space, algebra multiplication, conjugation, norm) turn $C(X)$ into what's called a *C^*-algebra*. We won't define this precisely, but the definition encodes various ways in which these structures are compatible. Actually, $C(X)$ is a *unital, commutative* C^*-algebra, where "unital" means that the multiplication has an identity element (namely the constant function 1), and commutative means that $fg = gf$. There is a natural notion of homomorphism between C^*-algebras, which are maps between them which behave nicely with all the structures involved. In particular, given a continuous map $f : X \to Y$ between compact Hausdorff spaces, we get a C^*-algebra homomorphism $f^* : C(Y) \to C(X)$ given by pre-composition with f: f^* sends $g \in C(Y)$ to $g \circ f \in C(X)$.

We thus have a contravariant functor **CHaus** $\to$ **ucC*-Alg** from the category of compact Hausdorff spaces and the category of unital, commutative C^*-algebras. It turns out that *any* unital, commutative C^*-algebra arises in this way, in the sense that given any unital, commutative C^*-algebra B there exists a compact Hausdorff space X such that $C(X) \cong B$ as C^*-algebras, and that morphisms between these algebra arises from continuous maps in the manner describe above. This says that the functor above is essentially surjective, full, and it turns out that it is also faithful, so that it is an equivalence! Thus, all information about compact Hausdorff spaces is completely encoded in information about unital, commtuative C^*-algebras, and indeed one could learn everything there is to know about a compact Hausdorff space X from its C^*-algebra $C(X)$ of functions. The fact that the categories **CHaus** and **ucC*-Alg** are equivalent is the starting point in the subject known as *noncommutative geometry*, which we'll say something about a bit later on. If we drop the requirement that our algebras be unital, it turns out that the category of commutative C^*-algebras in general is equivalent to the category of locally compact Hausdorff spaces via essentially the same functor, where we take as morphisms in the category of locally compact Hausdorff spaces to be continuous functions which "vanish at infinity".

We are only introducing C^*-algebras to give an example of an interesting, and actually useful equivalence between categories, but just for the fun of it we'll note the following. Examples of noncommutative C^*-algebras come from what are known as "algebras of bounded operators on Hilbert spaces", which show up in quantum mechanics as the things which describe physically observable quantities like position, momentum, and energy. (In fact, *all* C^*-algebras arise from Hilbert spaces in this way.) Thus, C^*-algebras show up in mathematical formulations of quantum mechanics, and the notion of *noncommutative geometry* mentioned above is at the core of some attempts to understand the elusive concept known as "quantum geometry".

Yoneda Lemma. The Yoneda Lemma states that given a (covariant) functor $F : \mathcal{C} \to \mathbf{Set}$, where $\mathcal{C}$ is locally small, for any $A \in \mathrm{Ob}(\mathcal{C})$ there exists a bijection

$$F(A) \cong \mathrm{Nat}(h^A, F),$$

where the right side denotes the set of natural transformations from h^A to F. A similar result holds in the contravariant case, where we get bijections $F(A) \cong \mathrm{Nat}(h_A, F)$. (There is also a sense in which these bijections are themselves natural, but we'll ignore this bit.) The proof of the Yoneda Lemma was the topic of a student presentation, and can be found in Section 2.2 of the book.

Here we will get a bit philosophical, and talk about what the point of the Yoneda Lemma actually is, and later the sense in which it says that we can view arbitrary functors as "non-existent objects” in a category. First, we mention what we will call the “functor of points” approach to mathematics: a mathematical object can be fully characterized by all morphisms *into* it. That is, knowledge of $\mathrm{Mor}(S, A)$ for all S is equivalent to knowledge of A itself. Indeed, this is an idea we mentioned previously in relation to the functors h_A—which we previously said is referred to as being the *functor of points* of A—and now we seek to push this idea further. In some examples, literal points of A can be extracted from h_A, as in the case of the bijection $X \cong h_X(pt) = \mathrm{Map}(pt, X)$ for a topological space X, or $G \cong h_G(\mathbb{Z}) = \mathrm{Mor}(\mathbb{Z}, G)$ for a group G. Moreover, in these examples, the rest of the (topological) structure on X or (group) structure on G can also be extracted from h_X and h_G respectively. In general, we can view $\mathrm{Mor}(S, A)$ as in a sense being “points of A parameterized by S”, which comes from the idea that in, say, **Set**, $\mathrm{Mor}(S, A) = A^S$ is “the product of S-many copies of A”, so literally points of A “parameterized” by S. In various areas of mathematics, an element of $\mathrm{Mor}(S, A)$ is thought of as being a “generalized” point of A, so that the functor of points h_A detects generalized points of A.

So, an object A is fully encoded by the functor h_A. Now we claim that morphisms between objects are also encoded by such functors. Indeed, in the case where $F = h_B$ is itself a functor of points, the Yoneda result $\mathrm{Nat}(h_A, F) \cong F(A)$ becomes:

$$\mathrm{Nat}(h_A, h_B) \cong h_B(A) = \mathrm{Mor}(A, B),$$

which says precisely that morphisms between the objects A and B can be obtained solely from the functors h_A and h_B. (This fact is one reason why one should prefer the contravariant functor h_A to the covariant functor h^A: in the covariant case, $\mathrm{Nat}(h^A, F) \cong F(A)$ for $F = h^B$ becomes $\mathrm{Nat}(h^A, h^B) \cong h^B(A) = \mathrm{Mor}(B, A)$, so that morphisms between two objects get *reversed* when characterizing them in terms of representable functors.)

Pushing our philosophy further, we now know that not only objects, but also morphisms between objects are fully encoded by the functors h_A. The final step is to note that we should not only care about objects and morphisms between them, but that an important part of the data of an object A is also how *arbitrary* contravariant functors $F : \mathcal{C} \to \mathbf{Set}$ behave *on* A, and that this too can be obtained from h_A precisely via the Yoneda result that $F(A) \cong \mathrm{Nat}(h_A, F)$! So, objects, morphisms between objects, *and* how arbitrary functors behave on objects are *all* encoded in knowledge of the functors h_A, so that these functors truly do capture in the most general sense possible *all* information about A.

Functors as nonexistent objects. The discussion above is all well and good, but it still doesn’t quite say why we should care about the fact that we can study on object A by studying h_A instead, in that we still haven’t spoken about what type of insight about A can be gleamed from h_A that wasn’t readily apparent in A itself already. Now we argue that the true point of the Yoneda Lemma is that it gives a way to think about arbitrary functors $F : \mathcal{C}^{op} \to \mathbf{Set}$ as “objects” in $\mathcal{C}$ itself, or really as objects in an “enlargement” of $\mathcal{C}$. Certainly for those functors $F \cong h_A$ which are representable, we can indeed think about F as simply being the object A, but now we claim that even *non-representable* F can be thought of as being “nonexistent” objects in $\mathcal{C}$. The upshot is that often in mathematics we come across things which do not actually exist, but the Yoneda Lemma gives a way to (indirectly) study such things anyway by interpreting them as functors instead.

The starting point, of course, is that an actual object A is the same as h_A, as we have explained above. Thus, by analogy, we can interpret a non-representable functor F as being an “object” in $\mathcal{C}$ as long as we know what it means to “map” *into* F from other things which are honestly objects in $\mathcal{C}$. That is, if F were indeed representable, then $F(A) \cong \mathrm{Nat}(h_A, F) \cong \mathrm{Mor}(A, F)$ would characterize the functor of points of F (here we abuse notation by using the same letter F to denote both a representable functor and the object which represents it), so now we take $\mathrm{Mor}(A, F) := F(A)$ as the *definition* of the "functor of points" of the non-existent object F! We are thus taking elements of $F(A)$ as being what it *means* to "map" *from* A *into* F, even when F is not representable. The fact that the Yoneda Lemma gives a way to define maps from A into F thus says that F itself is equipped with the same data that a literal object of A would be equipped with, so why shouldn't we interpret F as a being a type of "object" in $\mathcal{C}$ as well?

To phrase this another way, note that the Yoneda Lemma gives rise to a functor:

$$\mathcal{C} \to \text{Fun}(\mathcal{C}^{op}, \mathbf{Set}),$$

where $\text{Fun}(\mathcal{C}^{op}, \mathbf{Set})$ denotes the category whose objects are functors from $\mathcal{C}^{op}$ to $\mathbf{Set}$ and morphisms are natural transformation, defined on objects by $A \mapsto h_A$ and on morphisms by the bijection $\text{Mor}(A, B) \cong \text{Nat}(h_A, h_B)$ arising from the Yoneda Lemma. That this is actually bijection says that this functor is full and faithful, so that we can view $\mathcal{C}$ as being a *subcategory* of the functor category $\text{Fun}(\mathcal{C}^{op}, \mathbf{Set})$. (We cay say that $\mathcal{C}$ is isomorphic to the subcategory of $\text{Fun}(\mathcal{C}^{op}, \mathbf{Set})$ consisting of the functors h_A, or that it is equivalent to the subcategory of $\text{Fun}(\mathcal{C}^{op}, \mathbf{Set})$ consisting of representable functors. This functor $\mathcal{C} \to \text{Fun}(\mathcal{C}^{op}, \mathbf{Set})$ is usually called the *Yoneda embedding* of $\mathcal{C}$ into $\text{Fun}(\mathcal{C}^{op}, \mathbf{Set})$.) Thus, we can view $\text{Fun}(\mathcal{C}^{op}, \mathbf{Set})$ as an "enlargment" of $\mathcal{C}$, and the idea we are trying to make sense of is that other objects in this "larger" category should be interpreted as "non-existent objects" of $\mathcal{C}$.

Just how far can we actually push this idea that arbitrary functors $\mathcal{C}^{op} \to \mathbf{Set}$ should be thought of as being "objects" of $\mathcal{C}$? Pretty far, in fact. For instance, often times in geometry one comes across a type of "smooth" space on which some group acts, and one is interested in doing "calculus" on the resulting quotient space, which is the space of orbits under the given group action. The problem is that such quotient spaces are very often (usually in fact) not "smooth" themselves, so that the tools from calculus are not readily available. The solution is to nonetheless treat this quotient as if it were "smooth", and attempt to do calculus anyway! To make this precise, this "non-smooth" space actually turns out to have a natural interpretation as a "smooth functor", and it is through this functor which we can view this non-smooth space as actually being smooth after all; so, we are forced to expand our notion of what we mean by "smooth space" by allowing things which are not spaces in the usual sense, which certainly leads to more abstraction, but the benefit is that it actually makes sense to do calculus on such "spaces" interpreted as functors.

The discussion above is pretty vague, so here is one final, more concrete example. Recall that equivalence

$$\mathbf{CHaus}^{op} \to \mathbf{ucC^*\text{-}Alg}$$

between the category of compact Hausdorff spaces and the category of unital, commutative C^*-algebras mentioned at the start. We now claim that a *non-commutative* (unital) C^*-algebra should be interpreted as the algebra of functions on some non-existent "noncommutative compact Hausdorff space", and it is this idea which the subject of *noncommutative geometry* makes precise. So, let B be a non-commutative C^*-algebra. Then the functor of points h_B gives a functor

$$\mathbf{ucC^*\text{-}Alg} \to \mathbf{Set}$$

which sends $C(X)$ (where X is some compact Hausdorff space) to $\text{Hom}(C(X), B)$, where Hom denotes homomorphisms of C^*-algebras, whatever that means. Composing with the previous functor, we get a functor we'll denote by B itself:

$$B : \mathbf{CHaus}^{op} \to \mathbf{Set},$$

which on objects at least is given by $X \mapsto \text{Hom}(C(X), B)$. This functor is not representable since B is not commutative, but the philosophy espoused above suggests we should think of it as giving a way to think of B as an "object" in **CHaus** anyway, i.e. as a "noncommutative compact Hausdorff space". We *define* a "continuous map" $X \to B$ from a true compact Hausdorff space X into this "non-existent" one B to be precisely the data of a C^*-algebra homomorphism $B \to C(X)$. Again, if B were actually commutative, this would literally be true by the equivalence of categories given at the start. It turns out that many constructions in **CHaus** can then be carried over to such a functor B, giving a way to do "geometry" on this non-existent space solely in terms of the non-commutative C^*-algebra B itself. Good stuff.

Equalizers and Coequalizers

More on Yoneda and functors. Let us mention two more things based on the Yoneda Lemma. First, consider what the Yoneda Lemma says in the case of the category BG for a group G. In this case there is only one representable functor h_* to consider, where $*$ is the unique object of BG. The functor $h_* : BG \to \mathbf{Set}$ behaves on objects via $h_*(*) = \mathrm{Mor}(*, *) = G$, and on morphisms by

$$g \mapsto (h \mapsto hg)$$

where the latter is the map $\mathrm{Mor}(*, *) \to \mathrm{Mor}(*, *)$ given by "pre-composition" with g. Thus the functor $h_* : BG \to \mathbf{Set}$ is precisely the one which encodes the right action of G on itself by multiplication. The Yoneda Embedding

$$BG \to \mathrm{Fun}(BG^{op}, \mathbf{Set})$$

thus "embeds" this action into a larger category of "actions". The image of this embedding actually lands in the subcategory of functors F such that $F(*) = G$ is the underlying set of the given group, so we get an embedding of BG into the functors which encode *permutations* of the set G. Hence, the Yoneda Embedding in this case amounts to an injective group homomorphism $G \to \mathrm{Perm}(G)$, and the fact that such a group homomorphism exists is precisely the statement of *Cayley's Theorem* from group theory. In this way then, the Yoneda Embedding can be viewed as a vast generalization of Cayley's Theorem, which says that an arbitrary category can be viewed as subcategory of a "category of symmetries" of itself.

Second, if G is still a group, recall that the functor h_G is meant to encode the same data as G. But if H is another group, the set of morphisms $h_G(H) = \mathrm{Hom}(H, G)$ can itself be given a group structure defined by pointwise multiplication: the product of $\phi, \psi \in h_G(H)$ is defined by $(\phi\psi)(h) := \phi(h)\psi(h)$. The idea is that these group structures all together uniquely characterize the group structure on G itself. Now, suppose we had a non-representable functor $F : \mathbf{Grp}^{op} \to \mathbf{Set}$, which we wanted to view as a "non-existent" object in **Grp**; so, we want to view F as a type of "group". How can we make this precise?

The idea is, as in the case where F actually is representable, a "group" structure on F should be characterized by group structures on each $F(A) =: \mathrm{Mor}(A, F)$ which are all compatible in some sense. But a group structure on each set $F(A)$ amounts to functions

$$F(A) \times F(A) \to F(A) \text{ for each } A$$

which satisfy the group axioms. The required "compatibility" of these is expressed by saying that the data of all such group operations forms a natural transformation $F \times F \Rightarrow F$, where $F \times F$ denotes the functor which acts on objects as $A \mapsto F(A) \times F(A)$. Thus, in order to think of F as a type of "group", what we actually need is a "multiplication" $F \times F \Rightarrow F$ expressed as a natural transformation which obeys something similar to the group axioms. We'll talk about such things later on, where the point is that F should be what's called a *group object* in a category of functors. We'll see that the definition of an ordinary group can be expressed solely in terms of arrows and diagrams, so that we can take such diagrams as the *definition* of "group" in other categories. In this way, the idea that F can be thought of as a "non-existent group" will be made precise.

Universal properties. Now we return to considering various constructions, where the common thread underlying all is that of defining an object via a *universal property*. We have already seen two examples of such properties: the one defining products and the one defining coproducts. In the case of a product $A \times B$ for $A, B \in \mathrm{Ob}(\mathcal{C})$, the point is that we have morphisms $A \times B \to A$ and $A \times B \to B$, and the definition of product says precisely that $A \times B$ equipped with these morphisms is "universal" among *all* morphisms from an arbitrary object D into A and B, in the sense that the morphisms $D \to A$ and $D \to B$ can be obtained in a unique way from the morphisms $A \times B \to A$ and $A \times B \to B$ defining the product. Indeed, this is the point behind the "there exists a unique morphism $h : D \to A \times B$" part of the definition of $A \times B$, which does describe how to obtain $D \to A$ and $D \to B$ from the unique $h : D \to A \times B$.

Said another way, the point is that we have a diagram

$$\begin{tikzcd} & A \times B \arrow[dl, "p_A"'] \arrow[dr, "p_B"] & \\ A & & B \end{tikzcd}$$

which is universal among all such diagrams in the sense that any other diagram

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

can be obtained from the first in a unique way via a map $D \to A \times B$:

$$\begin{tikzcd} & D \arrow[dl, "f = p_A \circ h"'] \arrow[dr, "g = p_B \circ h"] \arrow[rrr, dashed, "\exists! h"] & & & A \times B \arrow[dl, "p_A"'] \arrow[dr, "p_B"] & \\ A & & B & A & & B \end{tikzcd}$$

Similarly, the definition of a coproduct $A \sqcup B$ for A and B says that we have a diagram of the form

$$\begin{tikzcd} A \arrow[dr, "i_A"] & & B \arrow[dl, "i_B"'] \\ & A \sqcup B & \end{tikzcd}$$

which is universal among all such diagrams: there is a unique way to obtain any

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

from the first:

$$\begin{tikzcd}
A \arrow[dr, "i_A"] & & B \arrow[dl, "i_B"'] & A \arrow[dr, "f = h \circ i_A"'] & & B \arrow[dl, "g = h \circ i_B"] \\
& A \sqcup B \arrow[rrr, dashed, "\exists! h"] & & & D &
\end{tikzcd}$$

By abstract nonsense, whenever an object satisfies a universal property, it will be unique up to unique isomorphism. (The term *universal property* of *universal element* can be defined more precisely using the Yoneda Lemma and the language of representable functors, but for us the informal idea expressed above will be enough. You can check the book for a more precise formulation if interested. The point is that a universal property will be characterized by some diagram, and the functor which assigns to an object all possible such diagrams will be representable the the object satisfying that universal property, if it exists.)

Equalizers and coequalizers. The first universal properties we will consider beyond that of products or coproducts are the ones defining the notion of an *equalizer* and a *coequalizer*. Given a pair of morphisms

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

an *equalizer* for the pair is an object E and morphism

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

satisfying $f \circ e = g \circ e$, which is universal among all such things; a *coequalizer* for the pair is an object Q and morphism

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

satisfying $q \circ f = q \circ g$, which is universal among all such things. (We say that e "equalizes" f and g, and q "coequalizes" f and g.) To be clear, saying that the equalizer diagram is universal among all such diagrams means that given

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

where $f \circ e' = g \circ e'$, there exists a unique morphism $E' \to E$ such that the diagram for E' can be obtained from the one for E; i.e. so that the following commutes:

$$\begin{tikzcd} E' \arrow[r, dashed, "\exists!"] \arrow[rr, bend left, "e'"] & E \arrow[r, "e"] & A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \end{tikzcd}$$

The same is true for the definition of a coequalizer, only with all arrows reversed.

The main examples are in **Set**, where the equalizer is given by

$$E = \{a \in A \mid f(a) = g(a)\}$$

and the coequalizer is the quotient (i.e. set of equivalence classes)

$$Q = B/\sim$$

where $\sim$ is the equivalence generated by the requirement that $f(a) \sim g(a)$ for all $a \in A$.

Regular monomorphisms and epimorphisms. It turns out that the morphism $E \to A$ defining an equalizer is always a monomorphism, and the morphism $B \to Q$ defining a coequalizer is always an epimorphism, as you will show on an upcoming homework set. In the case of **Set**, the equalizer E is a *subset* of A, and the coequalizer Q is a *quotient* of B. We say that a monomorphism $E \to A$ is a *regular monomorphism* if it is the equalizer of some pair of morphisms from A into B, and an epimorphism $B \to Q$ is a *regular* epimorphism if it is the coequalizer of some pair of morphisms A into B. We will see that regular epimorphisms can be thought of a generalizations of various "quotient" constructions, and regular monomorphisms are generalizations of "sub" constructions. For instance, a regular monomorphism $Y \to X$ in Top is precisely (up to homeomorphism) the inclusion of a subspace $Y \subseteq X$, and in **Grp** regular monomorphisms give subgroups.

Some Functor Properties, An Equivalence Example

Some Functor Properties. Problem 5 from Homework 2 was the topic of a student presentation. Concretely, this problem covered Exercises 1.5.iv, 1.6.iii, and 1.6.iv in the book, which deal with functors which preserve/reflect/create various types of morphisms. The point was to understand how to use faithfulness/fullness conditions.

An Equivalence Example. Problem 4 from Homework 2 was the topic of another student presentation, which is Exercise 1.5.ii in the book. This asked to show that the category of finite pointed sets $\mathbf{Fin}_*$ was equivalent to the opposite of a certain category Γ defined by Segal: Γ has finite sets as objects, and a morphism $S \to T$ is by definition a function

$$f : S \to P(T)$$

from S into the power set of T such that if $s_1 \neq s_2 \in S$, then $f(s_1) \cap f(s_2) = \emptyset$. Here we describe the desired equivalence, but will leave checking all details to the reader.

For $n \geq 0$, let $[n] := \{0, 1, 2, \ldots, n\}$. For simplicity, we will take our finite sets to be of the form $S_n := [n] - \{0\}$, which is enough since any finite set is isomorphic to one of these. We will then take our finite pointed sets to be $[n]$ for $n \geq 1$, where $0 \in [n]$ will always denote the specified basepoint. The functor $F : \Gamma \to \mathbf{Fin}_*$ is defined as:

$$S_n \mapsto [n] \text{ on objects, and}$$

$$(f : S_n \to P(S_m)) \mapsto (Ff : [m] \to [n]) \text{ on morphisms}$$

where $Ff : [m] \to [n]$ is given by

$$(Ff)(i) = \begin{cases} j & \text{if } i \in f(j) \text{ for some } j \in S_n \\ 0 & \text{if no such } j \text{ exists.} \end{cases}$$

Note that Ff is well-defined since if there is a $j \in S_n$ such that $i \in f(j)$, there is only one such j by the requirement that $f(a_1) \cap f(a_2) = \emptyset$ when $a_1 \neq a_2$. The point is that we look at the subsets

$f(j) \subseteq S_m$ for $j \in S_n$ to see if any contain $i \in [m]$: if so, we send i to that j; if not we send i to 0. In particular, $(Ff)(0) = 0$ since 0 appears in none of the S_n's, so Ff does preserve the basepoint.

Define the functor $G : \mathbf{Fin}_* \to \Gamma$ by

$$[n] \mapsto S_n \text{ on objects, and}$$

$$(g : [n] \to [m]) \mapsto (k \mapsto g^{-1}(k) - \{0\}) \text{ on morphisms}$$

where $g^{-1}(k) - \{0\} \subseteq P(S_n)$ simply denotes the preimage of $k \in S_m$ under g with zero excluded, which defines a function $S_m \to P(S_n)$ as required of a morphism in Γ. The check that F and G are functors, and that they are "inverse" to one another, is, as stated before, left to the reader.

Segal's Category, Coequalizer Examples

Segal's Category.

"Spaces" via functors.

Quotient groups as coequalizers.

Quotient spaces as coequalizers.

Limits and Colimits

Diagrams.

Cones and cocones.

Limits and colimits.

Examples.

Pullbacks and pushouts.

More on Limits/Colimits

Constructing limits in Set.

Back to pullbacks.

Inverse limits.

Gluing constructions.

More Limit/Colimit Examples

Coproducts in Vect.

Supremums and infimums.

$\mathbb{R}^\infty$ and S^∞.

Solenoids.