MTH 333 Lecture Notes

Zachary Winkeler

Fall 2023

Contents

Chapter 1

Categories

1.1 Categories in general

1.1.1 Motivation: linear algebra

Recall that a **group** is a set G with a binary operation $\cdot$ which is

- **associative**, e.g. for all $a, b, c \in G$,

$$(a \cdot b) \cdot c = a \cdot (b \cdot c)\,,$$

- **unital**, e.g. there exists an element $1 \in G$ such that for all $a \in G$,

$$1 \cdot a = a \cdot 1 = a,$$

  and

- *invertible*, e.g. for all $a \in G$ there exists an element $a^{-1} \in G$ such that

$$a \cdot a^{-1} = a^{-1} \cdot a = 1\,.$$

Some examples of groups include

- $(\mathbb{Z}, +)$,
- $(\mathbb{Z}/n\mathbb{Z}, +)$,
- $((\mathbb{Z}/n\mathbb{Z})^\times, \cdot)$, and
- the **symmetric group** S_n.

One interesting place to find groups is linear algebra. For example, the set of all $n \times n$ real matrices is denoted $\mathrm{M}_n(\mathbb{R})$. While matrix multiplication is a unital associative binary operation on $\mathrm{M}_n(\mathbb{R})$, it is not a group structure because not every matrix has an inverse. On the other hand, the set of $n \times n$ *invertible* real matrices forms a group under matrix multiplication called the **general linear group**, denoted $\mathrm{GL}_n(\mathbb{R})$.

Groups are one example of an *algebraic structure*, which is actually a bit of a vague term for a set equipped with some operations. While $\mathrm{M}_n(\mathbb{R})$ is not a group under multiplication, it does form a weaker kind of algebraic structure called a *monoid*. A **monoid** is a set equipped with an associative unital binary operation, i.e. like a group but without requiring inverses.

Some other examples of monoids that are not groups include

- $(\mathbb{N}, +)$,
- $((\mathbb{Z}/n\mathbb{Z}), \cdot)$, and
- the set of functions $\{1, \ldots, n\} \to \{1, \ldots, n\}$ under composition.

What can we say about the set $\mathrm{M}(\mathbb{R})$ of all real matrices (of all sizes)? While this set does have a binary operation in matrix multiplication, it is only *sometimes* defined. For example,

$$\begin{pmatrix}1 & 2\end{pmatrix} \cdot \begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{pmatrix} = \begin{pmatrix}9 & 12 & 15\end{pmatrix},$$

but

$$\begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{pmatrix} \cdot \begin{pmatrix}1 & 2\end{pmatrix} = \ ??? \ .$$

The issue is that an $m \times n$ matrix can only be multiplied by an $n \times p$ matrix; in other words, matrix multiplication defines a *partial* operation on the set $\mathrm{M}(\mathbb{R})$. Therefore, $(\mathrm{M}(\mathbb{R}), \cdot)$ is not a monoid, but our first example of a *category*.

1.1.2 Categories in general

A **category** $\mathcal{C}$ is

- a collection $\mathrm{Obj}(\mathcal{C})$ of **objects**,
- a collection $\mathrm{Hom}(A, B)$ of **morphisms** for any $A, B \in \mathrm{Obj}(\mathcal{C})$, and
- an associative, unital operation $\circ : \mathrm{Hom}(B, C) \times \mathrm{Hom}(A, B) \to \mathrm{Hom}(A, C)$ for any $A, B, C \in \mathrm{Obj}(\mathcal{C})$.

Sometimes we call morphisms **arrows**, especially if we are trying to draw them. Our example in Section 1.1.1 involves the category $\mathbf{Vect}_{\mathbb{R}}$. The objects of $\mathbf{Vect}_{\mathbb{R}}$ are real vector spaces, and the morphisms $\mathbb{R}^n \to \mathbb{R}^m$ are linear transformations, which we can represent by $m \times n$ matrices. More examples of categories include:

- The category **Set** has sets as its objects, and if A and B are sets, $\mathrm{Hom}(A, B)$ is the collection of functions $A \to B$. The operation $\circ$ is defined to be composition of functions as you might expect.
- The category **Grp** has groups as its objects, and if G and H are groups, $\mathrm{Hom}(G, H)$ is the collection of group homomorphisms $G \to H$.
- The category **Grph** has graphs as its objects, and graph homomorphisms (functions that send vertices to vertices and preserve the edges) as its morphisms.
- The category **Top** has topological spaces as its objects, and continuous maps as its morphisms.

In all of these examples, the objects are sets with extra structure, and the morphisms are all the functions between those sets that preserve the structure. The word for this type of category is a *concrete category*, a term which we will sadly have to wait until later to rigorously define. For now, the important thing to notice is that sets and functions play a fundamental role in category theory.

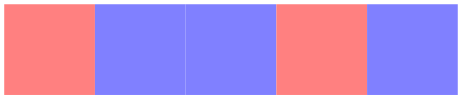
1.2 Examples of categories

Our examples above are all *concrete categories*, since the objects are sets with extra structure, and the morphisms are functions that preserve that structure. Some categories are not (obviously) concrete, on the other hand. There are (at least) two different perspectives on categories that one could adopt:

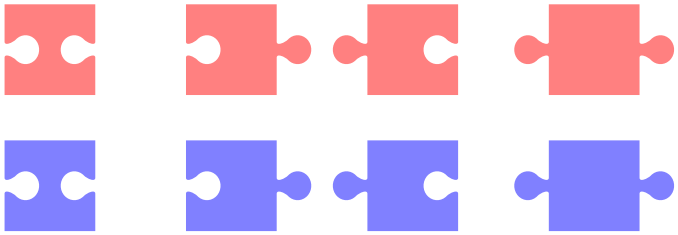
- *Object-oriented*: We want to study the objects, and the morphisms between them help us understand the objects better.
- *Morphism-oriented*: The morphisms are the actual "elements" of our category, and the purpose of the objects are to tell us when we can and can't compose the morphisms.

Our examples above support the first viewpoint rather nicely, but other examples might make more sense from the second perspective. For instance, any monoid (or group) M is a category with one object $\star$ and $\mathrm{Hom}(\star, \star) = M$, where $\circ$ is the monoid multiplication. The converse is true too; every category with only one object is actually a monoid!

Example. Imagine you have a bunch of square tiles that come in two colors, red and blue, and you want to arrange these tiles in a line. There are an infinite number of ways to do this; for example, you could make a line of three red tiles (RRR), a line of six tiles alternating colors ($RBRBRB$), etc. We might abbreviate the set of all such tile arrangements as $\{R, B\}^*$; this notation uses the **Kleene star**, which represents forming the *free monoid* generated by the set $\{R, B\}$.



Exercise 1.2.1. Now imagine that you're building a one-dimensional puzzle. You have a bunch of puzzle pieces that come in two colors, red and blue, and each piece may also have one of four shapes. The eight possible tiles are shown below:



You want to connect the puzzle pieces into strings, such that each piece fits nicely in the pieces beside it. For example, one string of puzzle pieces might look like this:

(a) The set of strings of puzzle pieces does not form a monoid. Why?

(b) The set of strings of puzzle pieces does, however, form a category. How many objects does this category have? What are the morphisms?

(c) Draw a picture of your category from part (b). (Since there are infinitely many strings of puzzle pieces, you probably can't draw *all* of it.)

Exercise 1.2.2. One way that we might depict binary operations is via multiplication tables. Each multiplication table below depicts a binary operation on the set $\{a, b, c\}$. Empty cells indicate that the row and column elements cannot be multiplied. For each table, determine if the operation satisfies the conditions for a group, monoid, and/or category. For each table that corresponds to a category, figure out how many objects it has and draw a picture of it.

(a)

	a	b	c
a	a	b	c
b	b	c	a
c	c	a	b

(b)

	a	b	c
a	a		
b		b	
c			c

(c)

	a	b	c
a	a	b	a
b	b	b	b
c	c	b	c

(d)

	a	b	c
a	a		
b	b		
c		b	c

(e)

	a	b	c
a	a		c
b		b	
c	c		c

1.3 Functors

1.3.1 Review

Recall that a category $\mathcal{C}$ is

- a collection $\mathrm{Obj}(\mathcal{C})$ of **objects**,
- a collection $\mathrm{Mor}(A, B)$ of **morphisms** for any $A, B \in \mathrm{Obj}(\mathcal{C})$, and
- an associative, unital operation $\circ : \mathrm{Mor}(B, C) \times \mathrm{Mor}(A, B) \to \mathrm{Mor}(A, C)$ for any $A, B, C \in \mathrm{Obj}(\mathcal{C})$.

Notably, each object $X \in \mathrm{Obj}(\mathcal{C})$ has an associated identity morphism $\mathrm{id}_X : X \to X$. Examples of categories include **Set**, $\mathbf{Vect}_{\mathbb{R}}$, **Grp**, any group G, any monoid M, and the one-dimensional puzzle piece example from last class.

One important perspective to keep in mind when studying categories is that sometimes you should think of the morphisms as the actual "elements" of the category, and the objects simply as obstructions to being able to compose morphisms.

One of the fun parts of category theory is that it draws connections between facts in seemingly unrelated parts of math.

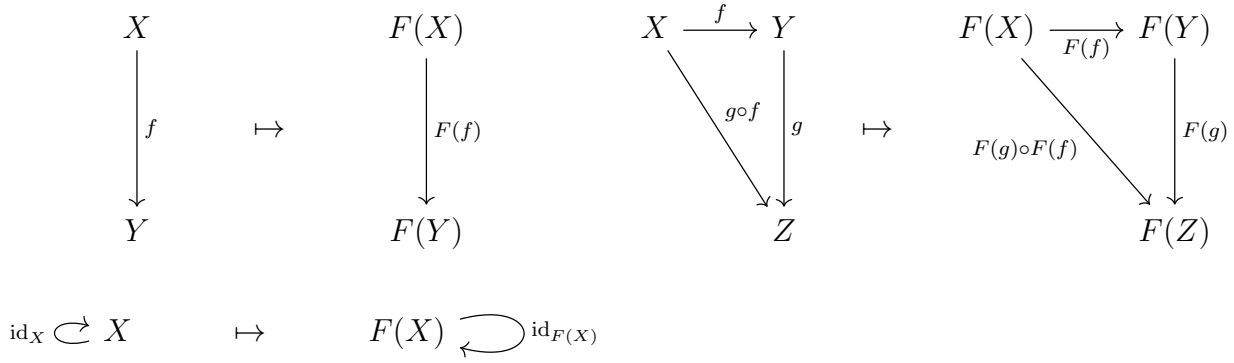
- Group theory: Every group is isomorphic to a subgroup of a symmetric group.
- Number theory: $\gcd(a, b) \cdot \mathrm{lcm}(a, b) = a \cdot b$.
- Propositional logic: $(P \wedge Q) \to R \iff P \to (Q \to R)$.
- Combinatorics: $|A \cup B| = |A| + |B| - |A \cap B|$.
- Set theory: $(A \cup B)^{\mathsf{c}} \cong A^{\mathsf{c}} \cap B^{\mathsf{c}}$.
- Functions: $\{\text{functions } A \to (B \to C)\} \cong \{\text{functions } A \times B \to C\}$.

1.3.2 Functors

In keeping with the spirit of studying maps between things, we may want to define maps between categories themselves! These are known as *functors*. A **functor** $F : \mathcal{C} \to \mathcal{D}$ is a map that sends objects to objects and morphisms to morphisms and respects the structure of a category i.e.

- if $f : X \to Y$ is a morphism in $\mathcal{C}$, then $F(f) : F(X) \to F(Y)$ is a morphism in $\mathcal{D}$,
- $F(\mathrm{id}_X) = \mathrm{id}_{F(X)}$, and
- $F(g \circ f) = F(g) \circ F(f)$.

We can visualize the above conditions with a few diagrams:



Functors are everywhere! For example, the determinant of the matrix is a functor $\det : \mathbf{Vect}_{\mathbb{R}} \to \mathbb{R}$, because

- we can think of $\mathbb{R}$ as a monoid under multiplication, so it is a category.
- $\det(T)$ is a real number, so it is a morphism in $\mathbb{R}$.
- $\det(\mathrm{id}_{\mathbb{R}^n}) = 1$.
- $\det(S \circ T) = \det(S)\det(T)$.

One can also check that

- If G and H are groups, then a functor $F : G \to H$ is just a group homomorphism.
- If M and N are monoids, then a functor $F : M \to N$ is a monoid homomorphism.

As another example, the *power set* operation defines a functor $\mathcal{P} : \mathbf{Set} \to \mathbf{Set}$. On a morphism $f : X \to Y$, we can define $\mathcal{P}(f)$ to be the map sending $U \subseteq X$ to $\mathcal{P}(U) \subseteq \mathcal{P}(Y)$. Then we can check that

- $\mathcal{P}(f) : \mathcal{P}(X) \to \mathcal{P}(Y)$ is indeed a function between sets,
- $\mathcal{P}(\mathrm{id}_X) = \mathrm{id}_{\mathcal{P}(X)}$, and
- $\mathcal{P}(g \circ f) = \mathcal{P}(g) \circ \mathcal{P}(f)$.

A **forgetful functor** is a common type of functor that "forgets" some structure on our objects. For example, the forgetful functor $\mathbf{Grp} \to \mathbf{Set}$ sends any group G to its underlying set. We have a similar forgetful functor from $\mathbf{Vect}_{\mathbb{R}}$ to $\mathbf{Set}$.

One consequence of the definition of a functor F is that it induces maps $\mathrm{Mor}(X, Y) \to \mathrm{Mor}(F(X), F(Y))$ for all objects X and Y. We say that the functor F is **faithful** if all of these induced maps are injective, and **full** if all of these induced maps are surjective. Forgetful functors are often faithful, but rarely full, since there may be other maps between objects that don't preserve their structure.

1.3.3 Posets

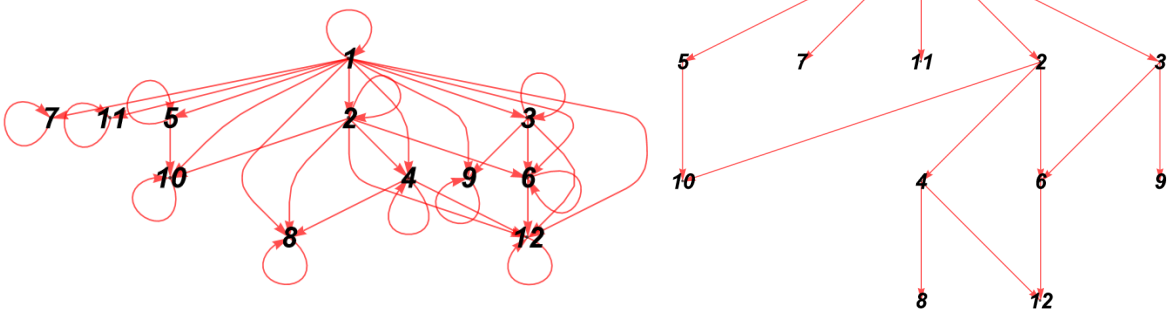
A **partially-ordered set** $(P, \leq)$ is a set P equipped with a binary relation $\leq$ that is

- reflexive, i.e. $p \leq p$,
- antisymmetric, i.e. $p \leq q$ and $q \leq p$ implies $p = q$, and
- transitive, i.e. $p \leq q$ and $q \leq r$ implies $p \leq r$.

Partially-ordered sets generalize **totally-ordered sets**, which contain the extra condition that one of $p \leq q$ or $q \leq p$ must always be true. Examples include

- any total order, e.g. $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$.
- $\mathbb{R}^2$ with the order $(x, y) \leq (r, s) \iff x \leq r \wedge y \leq s$.
- $\mathbb{N}$ with the divisibility order, i.e. $a \leq b \iff a \mid b$.
- $\mathcal{P}(X)$ for any set X, with $U \leq V \iff U \subseteq V$.

Any partially-ordered set $(P, \leq)$ can be made into a category $\mathcal{P}$: the objects of $\mathcal{P}$ are the elements of P, and there is a unique morphism $x \to y$ exactly when $p \leq q$; if $p \not\leq q$, there are no morphisms $p \to q$. We can visualize partial orders as directed graphs the same way that we would any other category; partial orders are especially nice since each set $\mathrm{Mor}(X, Y)$ either has zero or one elements, so there is no ambiguity about composition.



When drawing pictures of partial orders, we often omit the relations implied by reflexivity and transitivity (identity arrows and compositions of arrows), since they can be inferred from context and add visual clutter. A picture without these extra arrows is called a **Hasse diagram** of the partial order.

Exercise 1.3.1. Let $F : (\mathbb{N}, |) \to \mathcal{P}(\mathbb{N})$ be the functor that maps a natural number to its set of divisors.

(a) Prove that F is in fact a functor.

(b) Is F full?

(c) Is F faithful?

(d) Is F surjective on objects?

1.4 Constructions

1.4.1 Opposite Day

Let G be a non-commutative group, for example S_3. We can multiply elements of G to get other elements, and sometimes the order of multiplication might matter, e.g.

$$(13) \cdot (12) = (123)$$
$$(12) \cdot (13) = (132)\,.$$

Given any group G, we can define another group G^{op} with all the same elements as G, but where multiplication is *reversed*, i.e. if $ab = c$ in G, then $ba = c$ in G^{op}. We call G^{op} the **opposite group** of G. Denoting the multiplication in G^{op} by $\cdot'$, the above facts in S_3^{op} become

$$(13) \cdot' (12) = (132)$$
$$(12) \cdot' (13) = (123)\,.$$

Are S_3 and S_3^{op} different groups?

No! We can construct an isomorphism $\varphi : S_3 \to S_3^{\mathsf{op}}$ by swapping (123) and (132) and leaving the other four elements unchanged.

Also there is only one non-commutative group of order 6.

What can we say about G and G^{op} in general?

They are always isomorphic via the map $\phi : G \to G^{\mathsf{op}}$ sending $g \mapsto g^{-1}$.

In general, given any category $\mathcal{C}$, the **opposite category** $\mathcal{C}^{\mathsf{op}}$ has an object X' for every object X of $\mathcal{C}$, and a morphism $f' : Y' \to X'$ for every morphism $f : X \to Y$ of $\mathcal{C}$. The composition operation $\circ'$ in $\mathcal{C}^{\mathsf{op}}$ is defined such that if $f \circ g = h$ in $\mathcal{C}$, then $g' \circ' f' = h'$ in $\mathcal{C}^{\mathsf{op}}$. Note that not only is the order of composition swapped, but each morphism also has swapped sources and targets!

Remark 1.4.1. If $f : X \to Y$ is a function, we often call X the *domain* and Y the *codomain* of f. When talking about categories, sometimes we use the words **source** and **target** instead. These are mostly interchangeable; perhaps source and target feel less **Set**-focused.

One reason why we might define opposite categories is to talk about functors which *reverse the order of morphisms*. More specifically, one might define a **contravariant functor** $F : \mathcal{C} \to \mathcal{D}$ to be a mapping of objects to objects and morphisms $f : X \to Y$ to morphisms $F(f) : F(Y) \to F(X)$ that respects but reverses composition: $F(g \circ f) = F(f) \circ F(g)$. However, we can already express this construction using ordinary (**covariant**) functors and the opposite category. A contravariant functor $F : \mathcal{C} \to \mathcal{D}$ is just a functor $\mathcal{C}^{\mathsf{op}} \to \mathcal{D}$, or equivalently, a functor $\mathcal{C} \to \mathcal{D}^{\mathsf{op}}$.

1.4.2 Products and Disjoint Unions

Given two categories $\mathcal{C}$ and $\mathcal{D}$, we can form their **product category** $\mathcal{C} \times \mathcal{D}$. The objects of $\mathcal{C} \times \mathcal{D}$ are pairs $(C, D) \in \text{Obj}(\mathcal{C}) \times \text{Obj}(\mathcal{D})$, and the morphisms are pairs $(f, g) : (C, D) \to (X, Y)$ of morphisms $f : C \to X$ and $g : D \to Y$. Composition is defined component-wise, i.e. $(f, g) \circ (h, i) = (f \circ h, g \circ i)$. If $\mathcal{C} = G$ and $\mathcal{D} = H$ are groups, then the product category coincides with the direct product $G \times H$.

We can also form the **disjoint union category** $\mathcal{C} \sqcup \mathcal{D}$. The objects are $\text{Obj}(\mathcal{C}) \sqcup \text{Obj}(\mathcal{D})$ and the morphisms are inherited from each category individually. Visually, this corresponds to drawing a picture of both categories side-by-side.

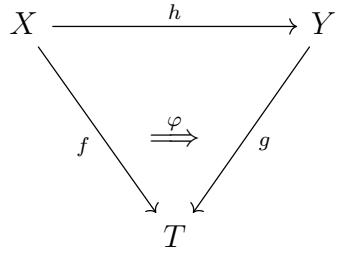
Remark 1.4.2. The **disjoint union** $X \sqcup Y$ of two sets X and Y represents taking their union while ignoring whether or not any elements of X are "the same" as elements of Y. One way to formally define this is as

$$X \sqcup Y = (X \times \{1\}) \cup (Y \times \{2\})$$

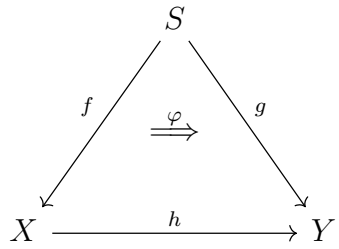
i.e. the elements of $X \sqcup Y$ are pairs $(x, 1)$ for $x \in X$ or $(y, 2)$ for $y \in Y$. This definition essentially forces elements of X and Y to be distinct.

1.4.3 Slices and Coslices

Let $\mathcal{C}$ be a category, and let T be an object in $\mathcal{C}$. The **slice category** $\mathcal{C}/T$ (read "$\mathcal{C}$ over T") has an object for every *morphism* whose target is T. If $f : X \to T$ and $g : Y \to T$ are two objects of $\mathcal{C}/T$, a morphism $\varphi : f \to g$ is a map $h : X \to Y$ making the triangle below commute, i.e. $g \circ h = f$.



Analogously, the **coslice category** $S/\mathcal{C}$ (read "$\mathcal{C}$ under S") has an object for every morphism whose source is S. If $f : S \to X$ and $g : S \to Y$ are two objects of $S/\mathcal{C}$, a morphism $\phi : f \to g$ is a map $h : X \to Y$ making the triangle below commute, i.e. $h \circ f = g$.



1.5 Isos, Monos, and Epis

1.5.1 Isomorphisms

In any category $\mathcal{C}$, an **isomorphism** $f : X \to Y$ is a morphism with a *two-sided inverse*, i.e. a morphism $g : Y \to X$ such that

$$g \circ f = \mathrm{id}_X \quad \text{and} \quad f \circ g = \mathrm{id}_Y \ .$$

We often denote a two-sided inverse g by f^{-1}. Intuitively, isomorphisms connect objects that are "basically the same". For example,

- the isomorphisms in **Set** are the bijections,
- the isomorphisms in **Grp** are the group isomorphisms, and
- the isomorphisms in $\mathbf{Vect}_{\mathbb{R}}$ are represented by invertible matrices.

If we consider a group G as a category, then every arrow is an isomorphism! If we consider a monoid M as a category, then the isomorphisms are exactly the invertible elements.

It may be tempting to think of isomorphisms and bijections in concrete categories as equivalent notions; however, this is only true in one direction! *Isomorphisms are always bijections, but bijections are not always isomorphisms.*

To illustrate this notion, let's think about posets. One natural notion of morphism between posets is *order homomorphism*. An **order homomorphism** $f : P \to Q$ is a function $P \to Q$ that respects the partial order

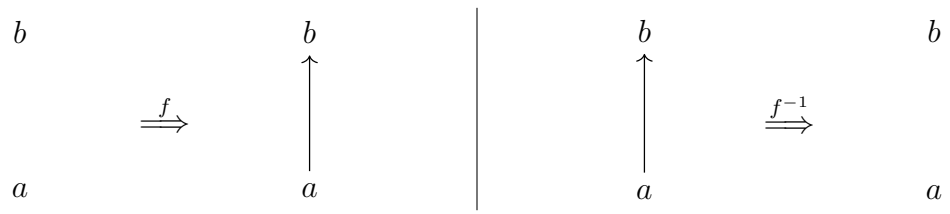
$$p \leq q \implies f(p) \leq f(q) \, .$$

In order for f to be an isomorphism of posets, f^{-1} needs to satisfy the same condition:

$$f(p) \leq f(q) \implies f^{-1}(f(p)) \leq f^{-1}(f(q)) \, .$$

Therefore, if $f : P \to Q$ is an **order isomorphism**, then $p \leq q \iff f(p) \leq f(q)$.

Now, consider two very basic posets. The poset P has underlying set $\{a, b\}$ with only the required relations $a \leq a$ and $b \leq b$. The poset Q has the same underlying set with only one additional relation $a \leq b$. The identity function on $\{a, b\}$ induces an order homomorphism $f : P \to Q$, which respects $\leq$ and is clearly bijective. However, f is not an isomorphism, since the inverse function f^{-1} does *not* respect $\leq$: $a \leq b$ in Q, but $a \not\leq b$ in P.



1.5.2 Monos and Epis

In our definition of isomorphism, it was important that $g : Y \to X$ was a *two-sided* inverse to $f : X \to Y$. There are many situations where $g \circ f = \mathrm{id}_X$ or $f \circ g = \mathrm{id}_Y$ are true individually, but not both at the same time. For example, let $f : \mathbb{N} \to \mathbb{Z}$ be given by $f(x) = x$ and let $g : \mathbb{Z} \to \mathbb{N}$ be given by $g(y) = |y|$. Then $(g \circ f)(x) = |x|$, which is the identity on $\mathbb{N}$. On the other hand, $(f \circ g)(y) = |y|$, which is *not* the identity on $\mathbb{Z}$. To summarize,

$$g \circ f = \mathrm{id}_{\mathbb{N}}, \quad \text{but} \quad f \circ g \neq \mathrm{id}_{\mathbb{Z}} \ .$$

Therefore, we say that g is a **left inverse** of f, and f is a **right inverse** of g. However, neither of f or g has a two-sided inverse. The set-theoretic terminology here is that f is injective and g is surjective, but neither is bijective.

There are multiple ways to generalize injective and surjective to morphisms in general categories, based on the following equivalences.

Lemma 1.5.1. *The following are equivalent for a function $f : X \to Y$ with $X \neq \emptyset$:*

(a) f is injective: $f(x_1) = f(x_2) \implies x_1 = x_2$ for all $x_1, x_2 \in X$.

(b) f is left-cancellative: $f \circ h_1 = f \circ h_2 \implies h_1 = h_2$ for all functions $h_1, h_2 : W \to X$.

(c) f has a left inverse: $g \circ f = \mathrm{id}_X$ for some $g : Y \to X$.

Lemma 1.5.2. *The following are equivalent for a function $f : X \to Y$ with $X \neq \emptyset$:*

(a) f is surjective: for all $y \in Y$, there exists an $x \in X$ such that $f(x) = y$.

(b) f is right-cancellative: $h_1 \circ f = h_2 \circ f \implies h_1 = h_2$ for all functions $h_1, h_2 : Y \to Z$.

(c) f has a right inverse: $f \circ g = \mathrm{id}_Y$ for some $g : Y \to X$.

We can't directly generalize (a) in either lemma to categories, since the objects of our category may not have "elements" in the traditional sense. On the other hand, each (b) and (c) definition generalizes nicely, since they only refer to properties of morphisms. In fact, it turns out that (b) and (c) produce distinct generalizations.

Generalizing part (b) of both lemmas, if a map $f : X \to Y$ is left-cancellative, i.e. $f \circ h_1 = f \circ h_2 \implies h_1 = h_2$, we call f **monic**, or a **monomorphism**. Dually, if $f : X \to Y$ is right-cancellative, i.e. $h_1 \circ f = h_2 \circ f$, we call g **epic**, or an **epimorphism**. These names are often shortened to simply "monos" or "epis".

On the other hand, generalizing part (c) of both lemmas gives us the definitions of *split monos and epis*. A **split monomorphism** $f : X \to Y$ is a morphism with a left inverse, i.e. $g \circ f = \mathrm{id}_X$ for some $g : Y \to X$. Dually, a **split epimorphism** $f : X \to Y$ has a right inverse, i.e. $f \circ g = \mathrm{id}_Y$ for some $g : Y \to X$.

Chapter 2

Limits

2.1 Universal properties

2.1.1 Initial, Terminals, and Zeroes

An **initial object** $\emptyset$ in a category $\mathcal{C}$ is an object such that there is exactly one morphism

$$\emptyset \to X$$

for every object X. Dually, a **terminal object** 1 in $\mathcal{C}$ is an object such that there is exactly one morphism

$$X \to 1$$

for every object X. Here are some examples:

Category	Initial object	Terminal object
Set	$\{\}$	$\{\star\}$
Grp	0	0
$\mathbf{Vect}_{\mathbb{R}}$	$\mathbb{R}^0$	$\mathbb{R}^0$
Ring	$\mathbb{Z}$	0
Field	–	–
$\mathbf{Field}_p$	$\mathbb{Z}/p\mathbb{Z}$	–

Sometimes, an object 0 in a category is both an initial *and* a terminal object. In this case, we call it a **zero object**. On the other hand, sometimes a category may not have any initial (or terminal) objects.

What we can say is that if an initial (or terminal) object exists, then it is *unique up to unique isomorphism*. What this means is that any two initial (or terminal) objects are isomorphic, and there is only one isomorphism between them! For example, the one-element set $\{\star\}$ is a terminal object in **Set**, and so is the one-element set $\{x\}$. These two sets are isomorphic, and in fact there is only one isomorphism between them: the map that sends $\star \mapsto x$.

2.1.2 Intro to Universal Properties

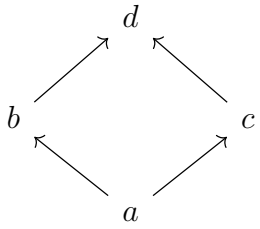
The definition of initial and terminal objects is our first example of a *universal property*. A **universal property** is a characterization of an object by the relationship that it has to every other object in a category.

Example. If A and B are sets, the intersection $A \cap B$ satisfies the universal property that, for any set C such that $C \subseteq A$ and $C \subseteq B$, $C \subseteq A \cap B$.

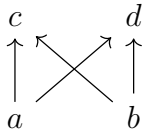
Example. The free group F_n satisfies the universal property that, for any group G, morphisms $F_n \to G$ are in bijection with n-tuples of elements of G.

Exercise 2.1.1. For each poset (or Hasse diagram representing a poset), identify the initial objects and/or terminal objects, if they exist.

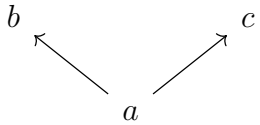
(a)



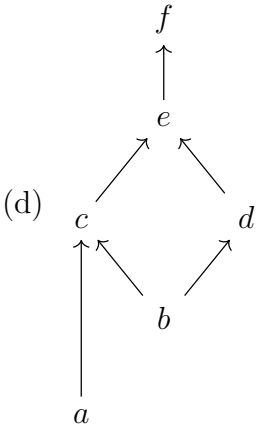
(b)



(c)



(d)



(e) $\mathbb{N}$ with the usual ordering $\leq$.

(f) $\mathbb{Z}$ with the usual ordering $\leq$.

(g) $\mathbb{N}$ with $a \leq b \iff a \mid b$.

(h) $\mathbb{Z}$ with $a \leq b \iff a \mid b$.

Exercise 2.1.2. Do any posets (not just the ones listed here) have zero objects?

The initial object of a poset is called the least element, or bottom, and is denoted $\bot$. The terminal object of a poset is called the greatest element, or top, and is denoted $\top$. A poset with both an initial object and a terminal object is called a bounded poset.

2.2 Products

2.2.1 Intro to Universal Properties

Recall that last class we discussed initial and terminal objects; an *initial object* $\emptyset$ satisfies the property that there is a unique map $\emptyset \to X$ for any object X. Dually, a *terminal object* 1 satisfies the property that there is a unique map $X \to 1$ for any object X. The definition of initial and terminal objects is our first example of a *universal property*. A **universal property** is a characterization of an object by the relationship that it has to every other object in a category.

Example. If A and B are sets, the intersection $A \cap B$ satisfies the universal property that, for any set C such that $C \subseteq A$ and $C \subseteq B$, $C \subseteq A \cap B$.

Example. The free group F_n satisfies the universal property that, for any group G, morphisms $F_n \to G$ are in bijection with n-tuples of elements of G.

One way that universal properties are used in category theory is to define constructions that don't depend on our objects having "elements", i.e. generalize notions from **Set** to an arbitrary category $\mathcal{C}$. For example, let's think about the product of two sets $A \times B$. Given A and B, it is easy to define the Cartesian product:

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

However, how can we define $A \times B$ in **Set** *without referencing elements*? In other words, is there a way to define $A \times B$ purely in terms of objects and morphisms?

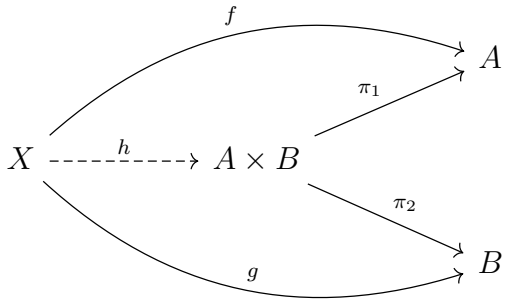
2.2.2 Products in general

Let A and B be two sets, let $A \times B$ be their Cartesian product, and let X be some other set. If we have two functions $f : X \to A$ and $g : X \to B$, then we can combine them into a single function $f \times g : X \to A \times B$ defined to be $(f \times g)(x) = (f(x), g(x))$. Conversely, given a function $h : X \to A \times B$, we can define maps $f : X \to A$ and $g : X \to B$ to be $f(x) = \pi_1(h(x))$ and $g(x) = \pi_2(h(x))$. Here, π_1 and π_2 are the projection maps

$$\begin{aligned} \pi_1 &: A \times B \to A & \pi_1(a, b) &= a \\ \pi_2 &: A \times B \to B & \pi_2(a, b) &= b\,. \end{aligned}$$

One can prove that this correspondence is actually a bijection between maps $X \to A \times B$ and pairs of maps $X \to A$ and $X \times B$. In other words, specifying a map $X \to A$ and a map $X \to B$ is basically the same as specifying a map $X \to A \times B$. This is the property that we will use to generalize the product of sets to other categories!

In any category $\mathcal{C}$, the **product** of two objects A and B, if it exists, is an object $A \times B$ with maps $\pi_1 : A \times B \to A$ and $\pi_2 : A \times B \to B$ such that for any two morphisms $f : X \to A$ and $g : X \to B$, there is a unique morphism $h : X \to A \times B$ with $\pi_1 \circ h = f$ and $\pi_2 \circ h = g$.



Note that the maps π_1 and π_2 are part of the definition of the product. For some examples of products,

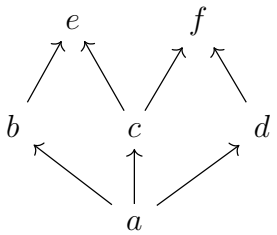
- the product in **Set** is the Cartesian product $A \times B$,
- the product in **Grp** is the product group $G \times H$,
- the product in $\mathbf{Vect}_{\mathbb{R}}$ is the direct sum $V \oplus W$,
- the product in **Ring** is the product ring $R \times S$.

In all of these cases, products exist and they are basically what you would expect them to be: the underlying set is the Cartesian product, and the extra structure is extended in a sensible way.

Sometimes, products may not exist; for example, there is no product in **Field** or $\mathbf{Field}_p$. In other cases, products exist but may not look anything like the Cartesian product of sets. Let's look at products in posets to see instances of this phenomenon. Given a poset $(P, \leq)$, we can consider it as a category where the objects are the elements of P as usual. Let $a, b \in P$ be two elements.

- The product $a \times b$, if it exists, is some element c with maps $c \to a$ and $c \to b$.
  - We can translate these maps as inequalities $c \leq a$ and $c \leq b$.
- Furthermore, the product c should satisfy the property that given any other $x \in P$ with maps $x \to a$ and $x \to b$, there is a unique map $x \to c$ that commutes with the projections.
  - We can translate this part as saying that if $x \leq a$ and $x \leq b$, then $x \leq c$.

Exercise 2.2.1. Find the products $a \times b$, $b \times d$, $d \times e$, and $e \times f$ in the poset shown below.



Essentially, in a poset $(P, \leq)$, the product of a and b is some element c that is smaller than (or equal to) a and b, but bigger than (or equal to) any other element x that is also smaller than (or equal to) a and b. To use a more familiar name, this element c is the **greatest lower bound** of a and b.

Exercise 2.2.2. Let $\mathcal{U}$ be some set, and consider the poset $(\mathcal{P}(\mathcal{U}), \subseteq)$ of subsets of $\mathcal{U}$ ordered by inclusion.

(a) Draw this poset for $\mathcal{U} = \{1, 2\}$.

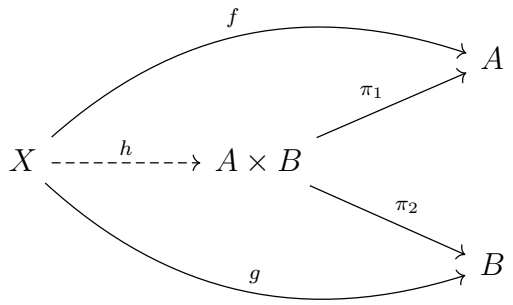
(b) What operation does the categorical product correspond to for a general set $\mathcal{U}$?

Exercise 2.2.3. Consider the poset consisting of the two truth values F and T, with the ordering $F \leq T$. What operation does the product correspond to in this category?

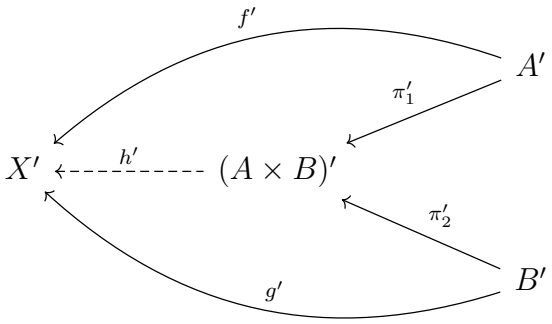
2.3 Coproducts

2.3.1 Duality

Last class, we talked about *products* in a category. The product of two objects A and B, is, if it exists, an objects $A \times B$ such that for any pair of morphisms $X \to A$ and $X \to B$, there exists a unique morphism $X \to A \times B$ making the diagram below commute.



Let's look at this diagram in the *opposite category*. In that case, the product $A \times B$ would correspond to some object $(A \times B)'$ that satisfies a different condition: for every pair of morphisms $f' : A' \to X'$ and $g' : B' \to X'$, there exists a unique morphism $h' : (A \times B)' \to X'$ such that the "opposite diagram" below commutes.



This object $(A \times B)'$ is no longer the product of A' and B'; instead it is their *coproduct*. In general, if we define some kind of construction in a category $\mathcal{C}$, then we get a dual construction in $\mathcal{C}^{\mathsf{op}}$. This notion is called **duality**, and is often colloquially thought of as "reversing the arrows".

Initial and terminal objects are dual in the same sense. An initial object $\emptyset$ in $\mathcal{C}$ corresponds to a terminal object in $\mathcal{C}^{\mathsf{op}}$:

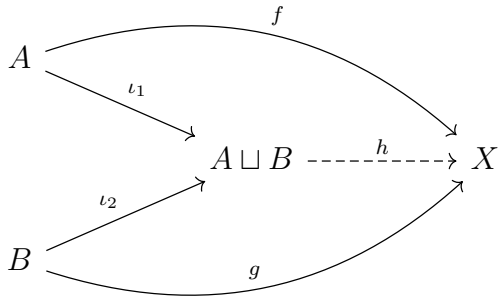
$$\emptyset \dashrightarrow X \qquad \text{vs.} \qquad \emptyset' \dashleftarrow X'$$

and dually, a terminal object 1 in $\mathcal{C}$ corresponds to an initial object in $\mathcal{C}^{\mathsf{op}}$:

$$X \dashrightarrow 1 \qquad \text{vs.} \qquad X' \dashleftarrow 1'\,.$$

2.3.2 Coproducts

The **coproduct** of two objects A and B in a category $\mathcal{C}$ is an object $A \sqcup B$ along with two inclusion maps $\iota_1 : A \to A \sqcup B$ and $\iota_2 : B \to A \sqcup B$, such that for any pair of maps $f : A \to X$ and $g : B \to X$, there exists a unique map $h : A \sqcup B \to X$ making the diagram below commute:



Coproducts exist in **Set**; the coproduct of two sets A and B is their **disjoint union** $A \sqcup B$, which represents taking their union while ignoring whether or not any elements of A are "the same" as elements of B. One way to formally define this is as

$$A \sqcup B = (A \times \{1\}) \cup (B \times \{2\})$$

i.e. the elements of $A \sqcup B$ are pairs $(a, 1)$ for $a \in A$ or $(b, 2)$ for $b \in B$. This definition essentially forces elements of A and B to be distinct.

One interesting thing to note about coproducts is that they often look a lot weirder than products in categories of algebraic structures, and might not be the "obvious thing" to construct. For example, the disjoint union of two groups is not a group; for one, you would have two identity elements, which is a problem! Instead, the coproduct $G \sqcup H$ is the **free product** $G \star H$:

- The elements of $G \star H$ are sequences of elements ("words") of $G \cup H$.
- We consider two words the same if they have the same **reduced word** $g_1 h_1 g_2 h_2 \ldots$, obtained by repeated combining adjacent elements from the same factor using the group operation.
- We identify both 1_G and 1_H with $1_{G \star H}$.
- The group operation in $G \star H$ is given by concatenating two words together.

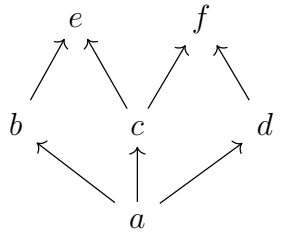
For example, if $G = S_3$ and $H = (\mathbb{Z}, +)$, then the product of the elements $a = ((132), -5, (13))$ and $b = ((13), 2)$ in the free product $G \star H$ is

$$\begin{aligned} ab &= ((132), -5, (13), (13), 2) \\ &= ((132), -5, 2) \\ &= ((132), 3)\,. \end{aligned}$$

2.3.3 Lattices

As usual, let's look at coproducts in posets. If $(P, \leq)$ is a poset considered as a category, then the coproduct of elements a and b in P is an element c such that $a \leq c$ and $b \leq c$, and for any x with $a \leq x$ and $b \leq x$, then $c \leq x$. The element c is, perhaps unsurprisingly, the **least upper bound** of a and b. This makes sense given that the product of a and b is the greatest lower bound, and that products and coproducts are dual notions.

Exercise 2.3.1. Find the coproducts $a \sqcup b$, $b \sqcup c$, $b \sqcup d$, and $e \sqcup f$ (if they exist) in the poset shown below.



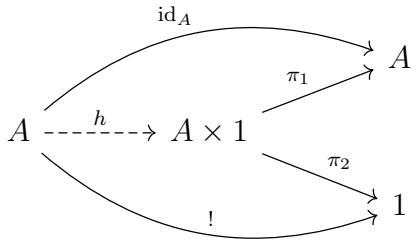
The product $a \times b$ in a poset is more commonly called the **meet** of a and b, denoted $a \wedge b$. Analogously, the coproduct $a \sqcup b$ in a poset is more commonly called the **join** of a and b, denoted $a \vee b$. A poset in which the meet and join of any two elements exist is called a **lattice**.

2.3.4 A First Look at Monoidal Categories

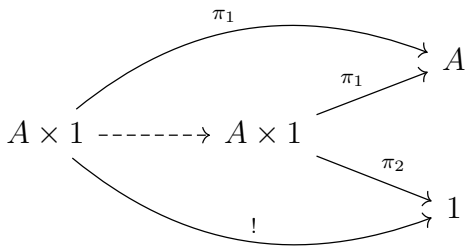
When we think of categories as algebraic structures, we are usually referring to the composition operation on morphisms as the "multiplication". However, products give us a way to multiply objects! (Sometimes.) Furthermore, if a category $\mathcal{C}$ with products has a terminal object 1, then that terminal object operates like a unit for the binary product $\times$!

Theorem 2.3.2. *For any object A, $A \times 1 \cong A$.*

Proof. We can use the universal property of the product $A \times 1$ with $f = \mathrm{id}_A$ and g the unique map $! : A \to 1$ to deduce that the map $h : A \to A \times 1$ is split mono, i.e. it has a left inverse which is given by $\pi_1 : A \times 1 \to A$.



Now, we would like to show that $h \circ \pi_1 = \mathrm{id}_{A\times 1}$. To do this, we will again use the universal property of the product, this time with the maps $\pi_1 : A \times 1 \to A$ and the unique map $! : A \times 1 \to 1$.



Which map could the dashed arrow be? Your first thought might be the identity $\mathrm{id}_{A\times 1}$. Indeed, this morphism does make the diagram commute, since $\pi_1 \circ \mathrm{id}_{A\times 1} = \pi_1$ and $\pi_2 \circ \mathrm{id}_{A\times 1} = \pi_2$. However, $h \circ \pi_1$ could also work just as well!

$$\begin{aligned} \pi_1 \circ (h \circ \pi_1) &= (\pi_1 \circ h) \circ \pi_1 \\ &= \mathrm{id}_A \circ \pi_1 \\ &= \pi_1 \end{aligned}$$

But the dashed arrow is supposed to be unique! Therefore, the map $\pi_1 \circ h$ must be equal to the identity! This concludes our proof that $h : A \to A \times 1$ is an isomorphism. □

2.4 Elements

One of the principles of generalizing constructions from **Set** is that "a good definition should involve sets and functions, but not elements of sets". This is because the objects of a given category may not *be* sets with extra structure, and instead might be elements of a poset, or arbitrary symbols like $\star$ or 😊. With that being said, what if we could define elements themselves, without using elements?

Definition 2.4.1. Let $\mathcal{C}$ be a category with a terminal object 1. A **global element** of an object X in $\mathcal{C}$ is a morphism $1 \to X$.

For example, the global elements of a set X in **Set** are maps $\{*\} \to X$, which are in bijection with elements of X. The global elements of a poset P in **Pos** are maps[1] $\{*\} \to P$, which are also in bijection with elements of P.

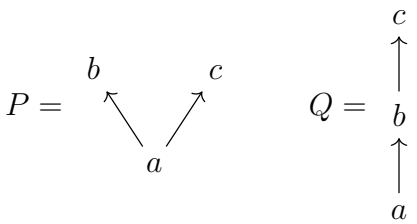
While this definition works well for some categories, it also fails for others. In the category **Grp** of groups and group homomorphisms, the terminal object is the trivial group $\{*\}$, but this object is also initial, so there is only ever one morphism $\{*\} \to G$ for any G. In the category $\mathbf{Vect}_{\mathbb{R}}$ of vector spaces and linear maps, the terminal object $\mathbb{R}^0$ is again also initial. However, sometimes we can modify the definition of global element to still recover some version of the "elements" of an object.

Definition 2.4.2. Let $\mathcal{C}$ be any category. A **generalized element** of an object X is a morphism $Y \to X$ for any object Y.

While generalized elements are significantly more general than global elements, in many cases it turns out that you can get a lot of information out of specific choices of "test object" Y. In $\mathbf{Vect}_{\mathbb{R}}$, letting $Y = \mathbb{R}^1$ actually recovers what we would probably think of as the "elements" of V. Specifically, there is a bijection $V \cong \mathrm{Mor}(\mathbb{R}^1, V)$ of sets.

Another advantage of generalized elements is that they provide a nice set of *invariants* for objects of a category. An **invariant** is some property of an object that is preserved by isomorphism. Therefore, if we want to prove that two objects X_1 and X_2 are not isomorphic, it suffices to show that some invariant distinguishes them. For a fixed Y, the set of global elements $Y \to X$ is an invariant of X up to isomorphism. Therefore, if $\mathrm{Mor}(Y, X_1)$ and $\mathrm{Mor}(Y, X_2)$ are not in bijection, then X_1 and X_2 must be non-isomorphic.

Example. Consider the posets P and Q depicted below:



Both P and Q have three global elements, so we cannot use this invariant to distinguish them. On the other hand, we can look at maps from other posets into these. For example, let I be the poset $\{0, 1\}$ with $0 \leq 1$. There are five distinct morphisms $I \to P$, and six distinct morphisms $I \to Q$. Therefore, P and Q must be non-isomorphic as posets.

[1] Here, we are implicitly giving the set $\{*\}$ its unique poset structure, i.e. $* \leq *$.

2.5 Subobjects

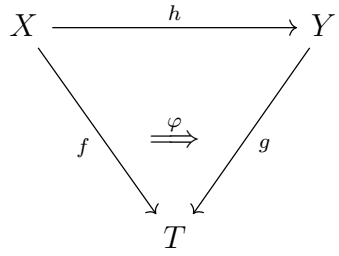
2.5.1 Products and Disjoint Unions

Given two categories $\mathcal{C}$ and $\mathcal{D}$, we can form their **product category** $\mathcal{C} \times \mathcal{D}$. The objects of $\mathcal{C} \times \mathcal{D}$ are pairs $(C, D) \in \text{Obj}(\mathcal{C}) \times \text{Obj}(\mathcal{D})$, and the morphisms are pairs $(f, g) : (C, D) \to (X, Y)$ of morphisms $f : C \to X$ and $g : D \to Y$. Composition is defined component-wise, i.e. $(f, g) \circ (h, i) = (f \circ h, g \circ i)$. If $\mathcal{C} = G$ and $\mathcal{D} = H$ are groups, then the product category coincides with the direct product $G \times H$.

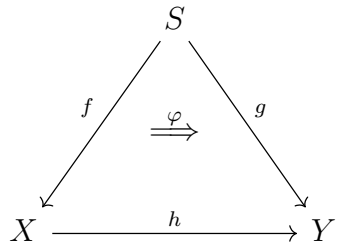
We can also form the **disjoint union category** $\mathcal{C} \sqcup \mathcal{D}$. The objects are $\text{Obj}(\mathcal{C}) \sqcup \text{Obj}(\mathcal{D})$ and the morphisms are inherited from each category individually. Visually, this corresponds to drawing a picture of both categories side-by-side.

2.5.2 Slices and Coslices

Let $\mathcal{C}$ be a category, and let T be an object in $\mathcal{C}$. The **slice category** $\mathcal{C}/T$ (read "$\mathcal{C}$ over T") has an object for every *morphism* whose target is T. If $f : X \to T$ and $g : Y \to T$ are two objects of $\mathcal{C}/T$, a morphism $\varphi : f \to g$ is a map $h : X \to Y$ making the triangle below commute, i.e. $g \circ h = f$.



Analogously, the **coslice category** $S/\mathcal{C}$ (read "$\mathcal{C}$ under S") has an object for every morphism whose source is S. If $f : S \to X$ and $g : S \to Y$ are two objects of $S/\mathcal{C}$, a morphism $\phi : f \to g$ is a map $h : X \to Y$ making the triangle below commute, i.e. $h \circ f = g$.



2.5.3 Subcategories

A **subcategory** $\mathcal{D}$ of $\mathcal{C}$ is a category with "some of the objects and morphisms from $\mathcal{C}$, but perhaps not all of them". We have already seen many examples of subcategories before, for example:

- Any diagram involving sets could be thought of as a subcategory of **Set**.

- The category Mono($\mathcal{C}$) that you constructed as homework is a subcategory of $\mathcal{C}$.

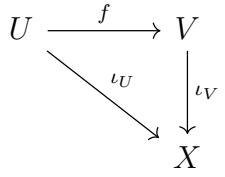
We will have a more rigorous definition of a subcategory soon!

2.5.4 Subobjects in general

Now, we will generalize the idea of a subset to general categories. As usual, we will be making use of definitions that do not involve elements.

Definition 2.5.1. A **subobject** of an object X in a category $\mathcal{C}$ is a monomorphism $U \to X$ for any object U.

Definition 2.5.2. A morphism from the subobject $\iota_U : U \to X$ to the subobject $\iota_V : V \to X$ is a map $f : U \to V$ such that $\iota_U = \iota_V \circ f$. Visually, this says that the diagram below commutes.



Definition 2.5.3. Given a category $\mathcal{C}$ and an object X, there is a category $\mathrm{Sub}_{\mathcal{C}}(X)$ whose objects are subobjects of X in $\mathcal{C}$ and whose morphisms are morphisms of subobjects.

One consequence of these definitions is that, while $\{1, 2\}$ and $\{1, 3\}$ are isomorphic as sets, the inclusion maps $\{1, 2\} \to \{1, 2, 3\}$ and $\{1, 3\} \to \{1, 2, 3\}$ are not isomorphic as subsets of $\{1, 2, 3\}$. On the other hand, if $U \to X$ and $V \to X$ are isomorphic as subobjects of X, then $U \cong V$ as objects of $\mathcal{C}$.

Note that $\mathrm{Sub}_{\mathcal{C}}(X)$ is a subcategory of $\mathcal{C}/X$; specifically, it is the subcategory where the objects are only the monomorphisms whose target is X, instead of *all* morphisms whose target is X.

The category $\mathrm{Sub}_{\mathcal{C}}(X)$ of subobjects of X is very close to being a partial order. The only issue is that there may be subobjects that are isomorphic but not equal. For example, in the category of sets, the inclusion map $\{0\} \to \{0, 1\}$ and the map $\{5\} \to \{0, 1\}$ sending 5 to 0 are isomorphic, but not the same map. Therefore, $\mathrm{Sub}_{\mathcal{C}}(X)$ is not partially-ordered. since it does not satisfy the asymmetry condition. However, it does still satisfy the reflexivity and transitivity conditions. We call such a category **preordered**.

Given any preorder $(P, \leq)$, we can force it to be a partial order by replacing the elements of P by their equivalence classes under the relation $x \sim y \iff x \leq y$ and $y \leq x$. Therefore, while preorders may initially seem to be much more general than partial orders, it turns out that they are very related. We usually ask for sets to be partially-ordered rather than preordered just for convenience; it is easier conceptually to assume that any two isomorphic objects are also the same!

Remark 2.5.4. The category of sets is rather special in that we could also have defined a subset of X as a morphism $X \to \{0, 1\}$ to a two-object set. In other words, the set of monomorphisms $U \to X$ (up to isomorphism) is in bijection with $\mathrm{Mor}(X, \{0, 1\})$. This relationship can be summarized by saying that the category **Set** has a *subobject classifier* $\{1, 2\}$. Not many categories have one of these, but those that do tend to have interesting properties.

2.6 Pullbacks

2.6.1 Motivation: Databases

Imagine that you are working on a project involving data about colleges. You might have a database `SCHOOLS` with a row for each college, and columns that list information about the schools.

Name	City	State	Students	...
Smith College	Northampton	MA	2,566	...
Amherst College	Amherst	MA	1,971	...
Hampshire College	Amherst	MA	465	...
Mount Holyoke College	South Hadley	MA	2,214	...
UMass Amherst	Amherst	MA	23,947	...
...	...	...	...	...

Additionally, let's say that you also have a database `CITIES` containing information about different cities.

Name	State	Population	...
Northampton	MA	29,311	...
Amherst	MA	37,819	...
South Hadley	MA	17,806	...
...	...	...	...

Perhaps we are interested in some statistic that depends on information from both tables; for example, maybe we want to know the ratio of each school's student population to the population of their city. Query langauges like SQL can perform what are called *cross joins*, which essentially take the Cartesian product of the rows in the two tables to produce a third table.

Name	City	State	Students	Name	State	Population	...
Smith College	Northampton	MA	2,566	Northampton	MA	29,311	...
Smith College	Northampton	MA	2,566	Amherst	MA	37,819	...
Smith College	Northampton	MA	2,566	South Hadley	MA	17,806	...
Amherst College	Amherst	MA	1,971	Northampton	MA	29,311	...
Amherst College	Amherst	MA	1,971	Amherst	MA	37,819	...
Amherst College	Amherst	MA	1,971	South Hadley	MA	17,806	...
Hampshire College	Amherst	MA	465	Northampton	MA	29,311	...
...	...	...	...	...	...	...	...

Figure 2.6.1: A cross join of the two tables.

We could definitely use this table to look up our statistic, but it also contains a bunch of extra unnecessary rows. For example, in row 2, why do we care about the population of Amherst if Smith College is in Northampton? The only actually relevant rows to our question are rows 1 and 5 highlighted above (as well as some other ones not shown).

In practice, this problem would actually be solved with an *inner join* in SQL, not a cross join. The inner join starts with the Cartesian product of the two tables, but only keeps a

subset of the rows that have matching columns (according to some specified definition of "matching"). In the example above, we may want to specify that we only want the rows of the table where the `City` column of the `SCHOOLS` table matches the `Name` column of the `CITIES` table.

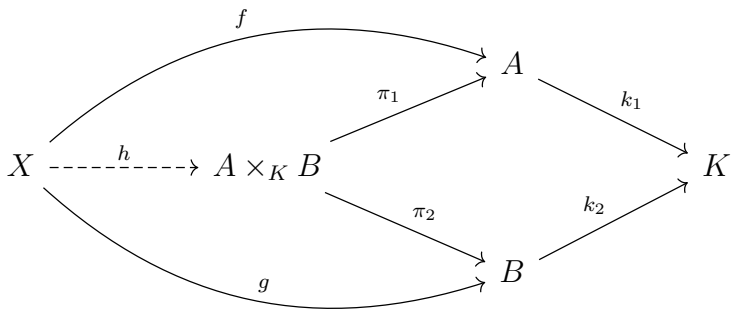
Name	City	State	Students	Name	State	Population	...
Smith College	Northampton	MA	2,566	Northampton	MA	29,311	...
Amherst College	Amherst	MA	1,971	Amherst	MA	37,819	...
Hampshire College	Amherst	MA	465	Amherst	MA	37,819	...
Mount Holyoke College	South Hadley	MA	2,214	South Hadley	MA	17,806	...
UMass Amherst	Amherst	MA	23,947	Amherst	MA	37,819	...
...	...	...	...	...	...	...	...

Figure 2.6.2: An inner join on the condition `SCHOOLS.City = CITIES.Name`.

This idea of taking a subset of a Cartesian product is the construction that we will be discussing today in the context of general categories.

2.6.2 Pullbacks in general

Let A and B be objects of a category $\mathcal{C}$. Furthermore, choose some maps $k_1 : A \to K$ and $k_2 : B \to K$ for some object K. The **pullback** of A and B along K, if it exists, is an object $A \times_K B$ with maps $\pi_1 : A \times_K B \to A$ and $\pi_2 : A \times_K B \to B$ such that for any other object X with maps $f : X \to A$ and $g : X \to B$ commuting with k_1 and k_2, there is a unique map $h : X \to A \times_K B$.



Intuitively, $k_1 : A \to K$ and $k_2 : B \to K$ are "keys" that tell us when elements of A and elements of B have "something in common". Therefore, the pullback $A \times_K B$ is like the product of A and B, except that it only contains information about pairs (a, b) where a and b have "something in common".

In the category of sets, this intuitive explanation is actually completely correct: $A \times_K B$ consists of all pairs (a, b) such that $k_1(a) = k_2(b)$. In symbols:

$$A \times_K B = \{(a, b) \in A \times B \mid k_1(a) = k_2(b)\}.$$

For example, let $A = B = K = \mathbb{N}$, let $k_1 : A \to K$ be $k_1(a) = a^2$, and let $k_2 : B \to K$ be $k_2(b) = b^3$. Then the product $A \times_K B$ consists of all pairs (a, b) such that $a^2 = b^3$:

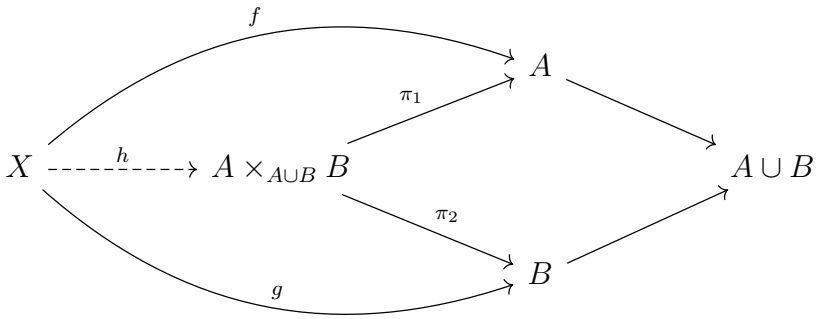
$$A \times_K B = \{(1, 1), (8, 4), (27, 9), (64, 16), \dots\}$$

2.6.3 Posets

Pullbacks in posets are actually kind of boring. If we consider a poset P as a category, then to construct a pullback we should start with two elements $a, b \in P$ and an element k such that $a \leq k$ and $b \leq k$. The pullback is an object $a \times_k b$ such that $a \times_k b \leq a$, $a \times_k b \leq b$, and for any x such that $x \leq a$ and $x \leq b$, $x \leq a \times_k b$. Sadly, this is just the product, which is just the meet: $a \times_k b = a \times b = a \wedge b$. The extra information involving the element k didn't change anything since morphisms in posets are unique.

2.6.4 Sets

Instead, let's consider another example involving **Set**. Let A and B be sets, and let $k_1 : A \to A \cup B$ and $k_2 : B \to A \cup B$ be the inclusions into their union. What is the pullback in this case?



As we defined before, the pullback of sets can be computed with a nice set-builder formula:

$$\begin{aligned} A \times_{A \cup B} B &= \{(a, b) \in A \times B \mid k_1(a) = k_2(b)\} \\ &= \{(a, b) \in A \times B \mid a = b\} \\ &= \{(x, x) \mid x \in A \text{ and } x \in B\} \\ &= \{(x, x) \mid x \in A \cap B\} \end{aligned}$$

Therefore, the pullback $A \times_{A \cup B} B$ is isomorphic to the intersection $A \cap B$ as sets. This is rather different than the ordinary product $A \times B$, which is just the Cartesian product. We have seen the intersection $A \cap B$ before, though, as the product in the *poset category* of subsets ordered by inclusion. Therefore, one could also think of the pullback as a generalization of the intersection that works in the full category of sets.

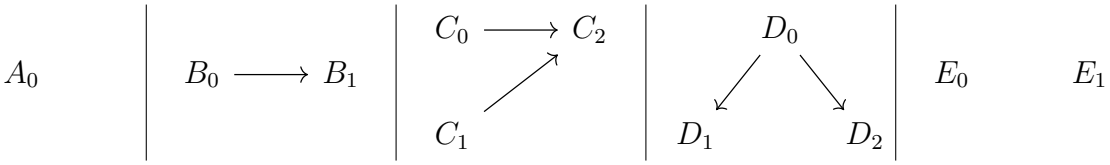
2.7 Limits

2.7.1 Diagrams and Cones

So far, we have relied on *commutative diagrams* as visual tools for understanding compositions between different morphisms. Now, we will give a precise definition of what a "diagram" really is.

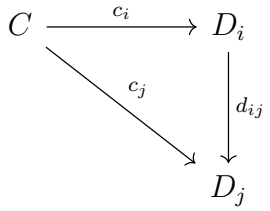
Definition 2.7.1. A **diagram of type $\mathcal{J}$ in a category $\mathcal{C}$** is a functor $D : \mathcal{J} \to \mathcal{C}$.

The category $\mathcal{J}$ of a diagram $D : \mathcal{J} \to \mathcal{C}$ called the *index category* of the diagram. Usually, index categories are very simple categories, with only a couple objects and morphisms. We might index the objects of $\mathcal{J}$ and their images in $\mathcal{C}$ by natural numbers, e.g. $D(J_i) = D_i$. Some examples of common index categories include:

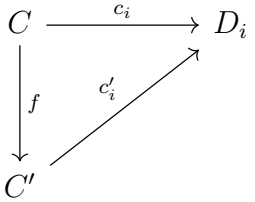


Some of these diagrams have special names; for example, diagrams shaped like the fourth index category above are often called *spans* or *roofs*. Diagrams shaped like the third index category are called *cospans*.

Given a diagram $D : \mathcal{J} \to \mathcal{C}$ in some category $\mathcal{C}$, we can talk about a *cone* on that diagram. A **cone** $(C, \{c_i\}_i)$ on D is a choice of object C in $\mathcal{C}$ and maps $c_i : C \to D_i$ commuting with the morphisms in $D(\mathcal{J})$. More specifically, we require that for each morphism $d_{ij} : D_i \to D_j$, we have that $d_{ij} \circ c_i = c_j$.



We can also talk about a **morphism of cones**. A morphism between cones $(C, \{c_i\}_i)$ and $(C', \{c'_i\})$ is a morphism f in $\mathcal{C}$ such that $f \circ c_i = c'_i$ for all i. Visually, f causes the diagram below to commute.



Putting these two ideas together, we get that the collection of all cones on a particular diagram itself forms a category. This category is denoted $\mathrm{Cone}(D)$; its objects are cones on D, and its morphisms are the morphisms of cones described above.

2.7.2 Limits in general

The purpose of defining cones is to allow us to unify many of the concepts we have talked about so far. For example, one way to describe the slice category $\mathcal{C}/T$ is as the category of cones on the constant diagram on T. Additionally, it turns out that terminal objects, products, and pullbacks are all specific examples of what are called *limits*.

A **limit** L of a diagram $D : \mathcal{J} \to \mathcal{C}$ is a terminal object in $\text{Cone}(D)$.

- Unfolding the definition of a cone, this means that L comes equipped with maps to all the objects in the image of D, and that these maps commute with the morphisms in the image of D.
- Unfolding the definition of a terminal object, this also means that any other object X that satisfies the first bullet point above has a unique morphism to L.

Let's see how limits generalize the concepts we have already discussed:

- Terminal objects are limits on the *empty diagram* $\epsilon : \emptyset \to \mathcal{C}$. Any object of $\mathcal{C}$ can be considered as a cone on ϵ, so a terminal object in $\text{Cone}(\epsilon)$ is just a terminal object in $\mathcal{C}$.
- Products are limits on the diagram D consisting of two disjoint objects A and B. A cone on D is a choice of object X with maps $X \to A$ and $X \to B$, so the terminal object in $\text{Cone}(D)$ is some object $A \times B$ with maps $A \times B \to A$ and $A \times B \to B$ such that there is a unique map $X \to A \times B$ for any X.
- Pullbacks are limits on *cospans*, i.e. diagrams D consisting of two objects A and B with morphisms to a third object K.

While limits may seem so much more general than these examples, it turns out that they are not any harder to find.

Theorem 2.7.2. *A category $\mathcal{C}$ has all finite limits if and only if it has all pullbacks and a terminal object.*

A "finite limit" is simply a limit over a diagram with finitely-many objects. But wait, there are diagrams with infinitely-many objects?

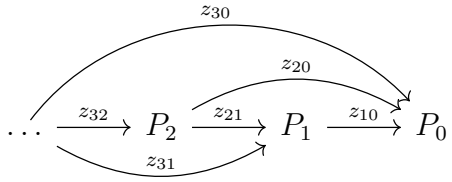
2.7.3 Example: Power Series

A *formal power series* is an expression of the form

$$a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \dots$$

which may contain infinitely many terms, where the coefficients a_i are real numbers. The set of all formal power series forms a ring $\mathbb{R}[\![x]\!]$, i.e. we can add and multiply these expressions just like polynomials.

In many ways, formal power series are like "infinite polynomials". We can make this intuition precise using a limit in the category **Ring** of all rings. Let $P_i = \mathbb{R}[x]/(x^i\mathbb{R}[x])$ be the ring of polynomials of degree $< i$. There are projection maps $z_{ij} : P_i \to P_j$ for $i \geq j$ given by removing terms of degree higher than j. We can then define $\mathbb{R}[\![x]\!]$ as the limit of the diagram



consisting of all the P_i and all the projections z_{ij} between them.

Unfolding the definition of a limit, this means that the ring of power series has maps $z_{\infty i} : \mathbb{R}[\![x]\!] \to P_i$ for all i, and that these maps commute with the other projections. This kind of makes sense! Given any power series, we can always truncate it at a particular power to get a polynomial; this is often done with Taylor series in calculus to get a "good enough" approximation. This explains why $\mathbb{R}[\![x]\!]$ should be a cone over this diagram. It is less obvious why it should be the *terminal cone* on this diagram; essentially, any other ring with "truncation" maps should factor through $\mathbb{R}[\![x]\!]$.

2.7.4 Limits in Set

As with pullbacks, we can give a more concrete definition of limits if we restrict our attention to diagrams in the category **Set**. Unlike with pullbacks, this definition is still rather unwieldy.

The limit of a diagram $D : \mathcal{J} \to \mathbf{Set}$ is the set

$$L = \{(x_0, x_1, \dots) \in D_0 \times D_1 \times \cdots \mid d_{ij}(x_i) = x_j \text{ for all } d_{ij}\}\,.$$

Essentially, an element of the limit L is like a choice of an element of each set in the diagram, such that the maps d_{ij} in the diagram identify the elements with each other.

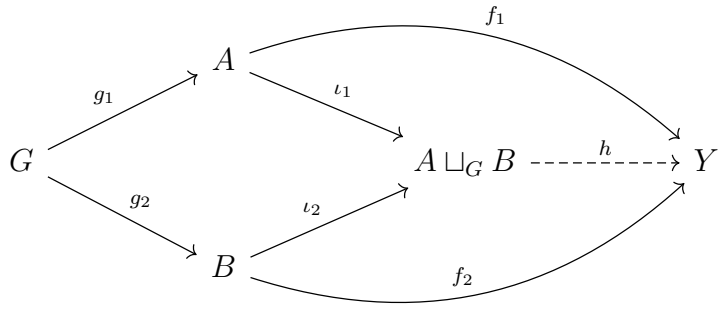
2.8 Colimits

Recall that last time, we generalized the concepts of terminal objects, products, and pullbacks using *limits*. A **limit** is a terminal object in the category $\mathrm{Cone}(D)$ of cones on some diagram $D : \mathcal{J} \to \mathcal{C}$. Today, we will generalize initial objects, coproducts, and *pushouts*, using constructions called *colimits*.

Wait, what is a pushout?

2.8.1 Pushouts

Pushouts are the dual concept to pullbacks. More specifically, given objects A and B of some category $\mathcal{C}$, as well as maps $g_1 : G \to A$ and $g_2 : G \to B$ from some object G, the **pushout** is an object $A \sqcup_G B$ equipped with maps $\iota_1 : A \to A \sqcup_G B$ and $\iota_2 : B \to A \sqcup_G B$ (commuting with g_1 and g_2). The pushout is also required to satisfy the universal property that, for any object Y with maps $f_1 : A \to Y$ and $f_2 : B \to Y$ (commuting with g_1 and g_2), there is a unique map $h : A \sqcup_G B \to Y$.



In the category of sets, we can describe the pushout more explicitly as

$$A \sqcup_G B = (A \sqcup B)/\sim$$

where $\sim$ is the equivalence relation generated by setting $(a, 1) \sim (b, 2)$ if and only if $g_1(x) = a$ and $g_2(x) = b$ for some $x \in G$. Intuitively, we are "gluing" the sets A and B together along the set G; each element of G tells us to identify an element of A with an element of B.

Example. Consider $\mathcal{C} = \mathbf{Set}$, and let $G = A \cap B$ with the inclusion maps $g_1 : A \cap B \to A$ and $g_2 : A \cap B \to B$. Then the pushout $A \sqcup_{A \cap B} B$ is isomorphic to the union $A \cup B$.

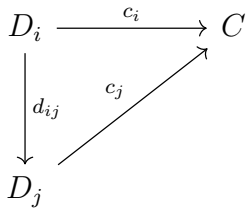
Example. Consider $\mathcal{C} = \mathbf{Set}$, and let $G = \emptyset$ with the unique maps g_1 and g_2 (since G is initial). Then the pushout $A \sqcup_\emptyset B$ is isomorphic to the disjoint union $A \sqcup B$.

Example. Consider $\mathcal{C} = \mathbf{Set}$, and let $G = \{*\}$. We can consider A and B as pointed sets $(A, g_1(*))$ and $(B, g_2(*))$. Then the pushout $A \sqcup_{\{*\}} B$ is isomorphic to their coproduct in the category $\mathbf{Set}_*$. This is sometimes called the **wedge sum** of the two pointed sets.

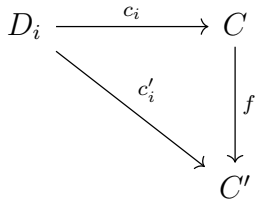
2.8.2 Colimits in general

We defined a limit as a terminal object in the category $\mathrm{Cone}(D)$ of cones on a diagram $D : \mathcal{J} \to \mathcal{C}$. The dual definition, then, is that a **colimit** is an *initial object* in the category $\mathrm{Cocone}(D)$ of *cocones* on a diagram $D : \mathcal{J} \to \mathcal{C}$.

A **cocone** on a diagram $D : \mathcal{J} \to \mathcal{C}$ is an object C of $\mathcal{C}$ along with maps $c_i : D_i \to C$ for all objects D_i in the diagram D, such that the maps c_i commute with the morphisms in D.



A **morphisms of cocones** from $(C, \{c_i\}_i)$ to $(C', \{c_i'\})$ is a map $f : C \to C'$ commuting with the maps c_i and c_i' for all i, i.e.



is a commutative diagram. Like cones, cocones on a fixed diagram D and cocone morphisms for a category $\mathrm{Cocone}(D)$.

Colimits generalize many of the constructions we have seen so far.

- Initial objects are colimits on the *empty diagram* $\epsilon : \emptyset \to \mathcal{C}$. Any object of $\mathcal{C}$ can be considered as a cocone on ϵ, so an initial object in $\mathrm{Cocone}(\epsilon)$ is just an initial object in $\mathcal{C}$.

- Coproducts are colimits on the diagram D consisting of two disjoint objects A and B. A cocone on D is a choice of object Y with maps $A \to Y$ and $B \to Y$, so the initial object in $\mathrm{Cocone}(D)$ is some object $A \sqcup B$ with maps $A \to A \sqcup B$ and $B \to A \sqcup B$ such that there is a unique map $A \sqcup B \to Y$ for any Y.

- Pushouts are colimits on *spans*, i.e. diagrams D consisting of two objects A and B with morphisms from a third object G.

As before, we also have a theorem about when categories have colimits.

Theorem 2.8.1. *A category $\mathcal{C}$ has all finite colimits if and only if it has all pushouts and an initial object.*

2.8.3 Example: Infinite Sequences

2.8.4 Colimits

Let's say that we have a diagram in $\mathbf{Vect}_{\mathbb{R}}$ consisting of an inclusion of $\mathbb{R}^0$ into $\mathbb{R}^1$, $\mathbb{R}^1$ into $\mathbb{R}^2$, and so on.

$$\mathbb{R}^0 \xrightarrow{\iota_0} \mathbb{R}^1 \xrightarrow{\iota_1} \mathbb{R}^2 \xrightarrow{\iota_2} \dots$$

More specifically, we could define the map $\iota_i : \mathbb{R}^i \to \mathbb{R}^{i+1}$ by sending the standard basis vector $\vec{e}_i$ in $\mathbb{R}^i$ to the standard basis vector $\vec{e}_i$ in $\mathbb{R}^{i+1}$; essentially, this inclusion map just adds a zero to the end of each vector in $\mathbb{R}^i$.

What is the colimit of this sequence? We're looking for some vector space C with inclusions from each $\mathbb{R}^i$ into C, such that inclusions into any other space factor through C. It turns out that we can think of elements of this space as infinitely-long vectors with only finitely-many nonzero entries. Let's call this space $\mathbb{R}^\infty$. This space can also be obtained as the coproduct of infinitely-many copies of $\mathbb{R}^1$.

2.8.5 Limits

Now, instead consider the diagram in $\mathbf{Vect}_{\mathbb{R}}$ consisting of projection maps $\mathbb{R}^i \to \mathbb{R}^{i-1}$.

$$\dots \xrightarrow{\pi_3} \mathbb{R}^2 \xrightarrow{\pi_2} \mathbb{R}^1 \xrightarrow{\pi_1} \mathbb{R}^0$$

More specifically, we could define the map $\pi_i : \mathbb{R}^i \to \mathbb{R}^{i-1}$ as projecting onto the first $i-1$ coordinates of our space, i.e. dropping the last component of a vector in $\mathbb{R}^i$.

What is the limit of this sequence? We're looking for some vector space L with projections to each $\mathbb{R}^i$, such that projections from any other space factor through L. It turns out that we can think of elements of this space as infinitely-long vectors, where any number of components can be nonzero. Since this is just the vector space of infinite sequences of real numbers, we can call it $\mathbb{R}^{\mathbb{N}}$. This space can also be obtained as the product of infinitely-many copies of $\mathbb{R}^1$.

2.8.6 Another Example

We can use the above techniques to construct "infinite" versions of a lot of different algebraic objects. For example, we can consider the inclusions $S_i \to S_j$ of symmetric groups, where $i \leq j$. If we write elements of S_i in cycle notation, these inclusions are induced by the identity map. We can consider the colimit of the diagram

$$S_0 \xrightarrow{\iota_0} S_1 \xrightarrow{\iota_1} S_2 \xrightarrow{\iota_2} \dots$$

to build the group of "permutations on an infinite set with finite support". Note that this is *not* isomorphic to the group $\mathrm{Aut}(\mathbb{N})$ of automorphisms of a countably-infinite set. The latter contains permutations that "move" infinitely-many elements, whereas the former does not.

Chapter 3

Functors

3.1 Categories, again

3.1.1 Limits and Colimits in Set

As with pullbacks, we can give a more concrete definition of limits if we restrict our attention to diagrams in the category **Set**. Unlike with pullbacks, this definition is still rather unwieldy.

The limit of a diagram $D : \mathcal{J} \to \mathbf{Set}$ is the set

$$L = \{(x_0, x_1, \dots) \in D_0 \times D_1 \times \cdots \mid d_{ij}(x_i) = x_j \text{ for all } d_{ij}\}\,.$$

Essentially, an element of the limit L is like a choice of an element of each set in the diagram, such that the maps d_{ij} in the diagram identify the elements with each other.

Dually, the colimit of a diagram $D : \mathcal{J} \to \mathbf{Set}$ is the set

$$C = (D_0 \sqcup D_1 \sqcup \dots)/\sim\,,$$

where $\sim$ is the equivalence relation generated by all equations of the form $x = d_{ij}(x)$ for $x \in D_i$ and some morphism $d_{ij} : D_i \to D_j$. Essentially, an element of the colimit C is like a choice of an element of *one* set in the diagram, modulo the relation identifying elements related by the maps d_{ij}.

3.1.2 Small and Large Categories

Now that we have become familiar with categories, we are starting to run into more "meta" questions like "Is there a category of all categories?" Before we talk about this, we should first question the perhaps most fundamental category, **Set**.

Let's review the definition of a category from the first week. A **category** $\mathcal{C}$ is

- a collection $\mathrm{Obj}(\mathcal{C})$ of **objects**,
- a collection $\mathrm{Mor}(A, B)$ of **morphisms** for any $A, B \in \mathrm{Obj}(\mathcal{C})$, and
- an associative, unital operation $\circ : \mathrm{Mor}(B, C) \times \mathrm{Mor}(A, B) \to \mathrm{Mor}(A, C)$ for any $A, B, C \in \mathrm{Obj}(\mathcal{C})$.

It is actually important that we used the word "collection" here instead of set. The objects of **Set** *cannot* themselves form a set, since that would imply that we have a "set of all sets". This is a construction that most versions of set theory prefer to avoid, so that our categories are paradox-free.

On the other hand, we have definitely seen categories where the objects actually do form a set. Many examples like groups or monoids have only a single object. Other examples, like posets, have a finite number of objects, or maybe a countably-infinite number of objects. Such categories with the property that $\mathrm{Obj}(\mathcal{C})$ and $\mathrm{Mor}(\mathcal{C})$ are sets are called **small** categories. Here, we are using $\mathrm{Mor}(\mathcal{C})$ to denote the collection of all morphisms in $\mathcal{C}$, regardless of source and target. Categories that do not satisfy this property are called **large**.

Therefore, with these sizes issues sorted out, we can actually talk about a "category of categories", as long as we restrict our categories a bit. The **category of small categories** **Cat** has an object for each small category, and a morphism $\mathcal{C} \to \mathcal{D}$ for each functor $F : \mathcal{C} \to \mathcal{D}$. It is worth noting that, while the objects of **Cat** are small categories, **Cat** itself is a large category.

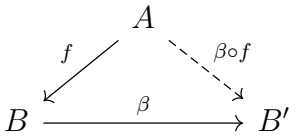
Let's go back to **Set**. Even though $\mathrm{Obj}(\mathbf{Set})$ is not a set, if we specify two sets A and B, then the collection of all functions $A \to B$ actually does form a set. Therefore, if we just wanted to generalize the properties of the category of sets, we could have defined a category to have a *set* of morphisms $\mathrm{Mor}(A, B)$ for each pair of objects (A, B). Categories in which $\mathrm{Mor}(A, B)$ is always a set are called **locally small** categories.

As an easy first example, all small categories are also locally small. For examples of locally small categories that are not small, we can look at categories whose objects are algebraic structures that can be "built" on any set. For example, **Pos**, **Top**, and **Group** are all large, but locally small.

Question. Is **Cat** locally small?

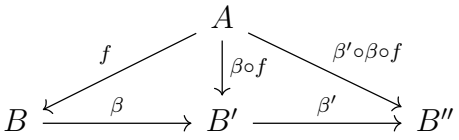
3.2 Representable Functors

Let $\mathcal{C}$ be a locally small category, so that $\mathrm{Mor}(A, B)$ is a set for each pair of objects A and B. Choose particular objects A and B. If we also choose a morphism $\beta : B \to B'$, then this induces a function $\mathrm{Mor}(A, \beta) : \mathrm{Mor}(A, B) \to \mathrm{Mor}(A, B')$. This function is defined as $\mathrm{Mor}(A, \beta)(f) = \beta \circ f$.

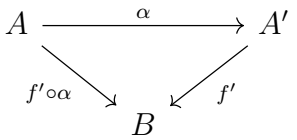


If we fix an object A, but treat the object B as a parameter, we get that $\mathrm{Mor}(A, -)$ is a functor $\mathcal{C} \to \mathbf{Set}$. This functor sends an object B to the set $\mathrm{Mor}(A, B)$, and a morphism $\beta : B \to B'$ to the function $\mathrm{Mor}(A, \beta) : \mathrm{Mor}(A, B) \to \mathrm{Mor}(A, B')$. This functor is called a **representable functor**; we say that the functor is represented by the object A.

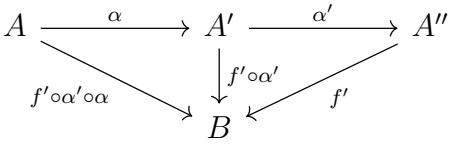
To check that $\mathrm{Mor}(A, -)$ is a functor, we would need to show that it respects the identity morphisms and compositions. Here is a picture of what this might look like for compositions.



Similarly, we could instead fix B, and treat A as a variable parameter. If we have some morphism $\alpha : A \to A'$, then it induces a function $\mathrm{Mor}(\alpha, B) : \mathrm{Mor}(A', B) \to \mathrm{Mor}(A, B)$; note the swapped direction! This function is defined as $\mathrm{Mor}(\alpha, B)(f') = f' \circ \alpha$.



In this case, we no longer get that $\mathrm{Mor}(-, B)$ is a (covariant) functor $\mathcal{C} \to \mathbf{Set}$, since it swaps the order of morphisms!



Instead, this is a *contravariant* functor, i.e. a functor $\mathcal{C}^{\mathsf{op}} \to \mathbf{Set}$. This functor is also called a representable functor; sometimes people use the term "corepresentable" to indicate the contravariance.

3.2.1 Limit Preservation

A functor $F : \mathcal{C} \to \mathcal{C}'$ is said to **preserve limits** if for any diagram $D : \mathcal{J} \to \mathcal{C}$ with a limit L, the object $F(L)$ is a limit of the diagram $F \circ D : J \to \mathcal{C}'$. The functor should also respect the projections $L \to D_i$ for each object D_i of the diagram D. Functors that preserve limits are called **continuous**. If we use the notation $\varprojlim D$ to denote the limit of the diagram D, then this relationship is sometimes written

$$F(\varprojlim D) = \varprojlim F(D)\,.$$

Functors can also **preserve colimits** in the same sense, i.e.

$$F(\varinjlim D) = \varinjlim F(D)\,,$$

where $\varinjlim D$ denotes the colimit of the diagram D. Such functors are called **cocontinuous**.

We can find many examples of continuous functors by looking at sets of morphisms.

Theorem 3.2.1. *For any object A, the covariant representable functor $\mathrm{Mor}(A, -)$ preserves limits.*

For example, in the category of sets, there is a bijection

$$\mathrm{Mor}(A, B \times C) \to \mathrm{Mor}(A, B) \times \mathrm{Mor}(A, C)\,.$$

On the other hand, while one might expect the corepresentable functors to preserve colimits, they in fact do something a bit weirder.

Theorem 3.2.2. *For any object B, the contravariant representable functor $\mathrm{Mor}(-, B)$ maps colimits to limits.*

For example, in the category of sets, there is a bijection

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

Here, the colimit $\sqcup$ is turned into a limit $\times$.

3.3 Natural transformations

Let $\mathcal{C}$ be a locally-small category. Recall that last time we discussed how to think of the morphism set $\mathrm{Mor}(A, B)$ as a functor $\mathrm{Mor}(-,-) : \mathcal{C}^{\mathsf{op}} \times \mathcal{C} \to \mathbf{Set}$. If we fix the first parameter, we get the covariant functor $\mathrm{Mor}(A, -)$ *represented* by the object A. If we instead fix the second parameter, we get the contravariant functor $\mathrm{Mor}(-, B)$ *represented* by the object B.

Consider the power set functor $\mathcal{P}(-) : \mathbf{Set} \to \mathbf{Set}$ on the category of sets. This functor is not of the form $\mathrm{Mor}(A, -)$ or $\mathrm{Mor}(-, B)$, since subsets are not maps from or into a fixed object. On the other hand, for any set X, we have an isomorphism $\mathcal{P}(X) \cong \mathrm{Mor}(X, \{0, 1\})$. Intuitively, we can think of the functor $\mathcal{P}(-)$ as being "isomorphic" to a representable functor, even if it isn't equal to one. Before we define isomorphisms of functors, though, we first need to talk about what a map between functors should look like!

3.3.1 Natural transformations in general

Let $\mathcal{C}$ and $\mathcal{D}$ be categories, and let $F, G : \mathcal{C} \to \mathcal{D}$ be two functors. A **natural transformation** $\theta : F \to G$ is, for every object $C \in \mathcal{C}$, a morphism $\theta_C : F(C) \to G(C)$ in $\mathcal{D}$ that commutes with the images of the morphisms from $\mathcal{C}$. More specifically, for any $f : C \to C'$ in $\mathcal{C}$, the following diagram should commute.

$$\begin{array}{ccc} F(C) & \xrightarrow{\theta_C} & G(C) \\ \Big\downarrow{\scriptstyle F(f)} & & \Big\downarrow{\scriptstyle G(f)} \\ F(C') & \xrightarrow{\theta_{C'}} & G(C') \end{array}$$

The maps θ_C are called the **components** of θ.

3.3.2 Example: $\mathrm{id}_{\mathbf{Set}} \to \mathrm{Mor}(\{*\}, -)$

For example, let $F = \mathrm{id}_{\mathbf{Set}} : \mathbf{Set} \to \mathbf{Set}$ be the identity functor on $\mathbf{Set}$, and let $G = \mathrm{Mor}(\{*\}, -) : \mathbf{Set} \to \mathbf{Set}$ be the covariant functor represented by $\{*\}$. The functor $\mathrm{Mor}(\{*\}, -)$ sends a set to its global elements, which should be isomorphic to the original set. We can define a natural transformation $\theta : \mathrm{id}_{\mathbf{Set}} \to \mathrm{Mor}(\{*\}, -)$ by defining a component θ_X for each set X. Specifically, we can send $x \in X$ to the constant function $\mathrm{const}_x : \{*\} \to X$, i.e. $\theta_X(x) = \mathrm{const}_x$.

$$\begin{array}{ccc} X & \xrightarrow{x \mapsto \mathrm{const}_x} & \mathrm{Mor}(\{*\}, X) \\ \Big\downarrow{\scriptstyle f} & & \Big\downarrow{\scriptstyle \mathrm{Mor}(\{*\}, f)} \\ Y & \xrightarrow{y \mapsto \mathrm{const}_y} & \mathrm{Mor}(\{*\}, Y) \end{array}$$

3.3.3 Example: $\mathrm{id}_{\mathbf{Set}} \to \mathcal{P}(-)$

Let $F = \mathrm{id}_{\mathbf{Set}} : \mathbf{Set} \to \mathbf{Set}$ be the identity functor and let $G = \mathcal{P}(-) : \mathbf{Set} \to \mathbf{Set}$ be the power set functor. We can define a natural transformation $\theta : \mathrm{id}_{\mathbf{Set}} \to \mathcal{P}(-)$ on a set X to be $\theta_X(x) = \{x\}$.

$$\begin{array}{ccc} X & \xrightarrow{x \mapsto \{x\}} & \mathcal{P}(X) \\ \Big\downarrow{\scriptstyle f} & & \Big\downarrow{\scriptstyle \mathcal{P}(f)} \\ Y & \xrightarrow{y \mapsto \{y\}} & \mathcal{P}(Y) \end{array}$$

3.3.4 Example: $\mathrm{Mor}(-, \{0,1\}) \to \mathcal{P}(-)$

Before we can define a natural transformation between these two functors, we should first deal with an issue of variance; specifically, $\mathrm{Mor}(-, \{0,1\})$ is a *contravariant* functor, whereas $\mathcal{P}(-)$, as we have defined it, is *covariant*. Therefore, the former has source $\mathbf{Set}^{\mathsf{op}}$ and the latter has source $\mathbf{Set}$, so we cannot even define a natural transformation between them! Thankfully, it turns out that there are actually two power set functors, and we can just use the contravariant one. Define the **contravariant power set functor** $\mathcal{P} : \mathbf{Set}^{\mathsf{op}} \to \mathbf{Set}$ to assign to each set X its power set $\mathcal{P}(X)$, and to each function $f : X \to Y$ the inverse image function $\mathcal{P}(f) : \mathcal{P}(Y) \to \mathcal{P}(X)$ defined by $\mathcal{P}(f)(B \subseteq Y) = f^{-1}(B) \subseteq A$.

Now, let $F = \mathrm{Mor}(-, \{0,1\})$ be the contravariant functor represented by $\{0,1\}$, and let $G = \mathcal{P}(-)$ be the contravariant power set functor. We can define a natural transformation $\theta : \mathrm{Mor}(-, \{0,1\}) \to \mathcal{P}(-)$ via components $\theta_X : \mathrm{Mor}(X, \{0,1\}) \to \mathcal{P}(X)$. Specifically, let $\theta_X(\varphi) = \varphi^{-1}(1)$.

$$\begin{array}{ccc} \mathrm{Mor}(X, \{0,1\}) & \xrightarrow{\varphi \mapsto \varphi^{-1}(1)} & \mathcal{P}(X) \\ \Big\uparrow{\scriptstyle \mathrm{Mor}(f, \{0,1\})} & & \Big\uparrow{\scriptstyle \mathcal{P}(f)} \\ \mathrm{Mor}(Y, \{0,1\}) & \xrightarrow{\psi \mapsto \psi^{-1}(1)} & \mathcal{P}(Y) \end{array}$$

3.3.5 Example: $X \times Y \to Y \times X$

Let $\mathcal{C}$ be any category with binary products. Let $F = - \times - : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ be the product functor, and let $G = - \times - : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ also be the product, but reversing the order of the inputs. We can define a natural transformation $\theta : F \to G$ on an object (X, Y) of $\mathcal{C} \times \mathcal{C}$ to be the map $\theta_{(X,Y)} : X \times Y \to Y \times X$ written in components as (π_2, π_1).

$$\begin{array}{ccc} X \times Y & \xrightarrow{(\pi_2, \pi_1)} & Y \times X \\ \Big\downarrow{\scriptstyle f \times g} & & \Big\downarrow{\scriptstyle g \times f} \\ W \times Z & \xrightarrow{(\pi_2, \pi_1)} & Z \times W \end{array}$$

3.4 Natural isomorphisms

Let $\mathcal{C}$ and $\mathcal{D}$ be categories, and let $F, G : \mathcal{C} \to \mathcal{D}$ be two functors. Recall that last time we defined a **natural transformation** $\theta : F \to G$ to be, for every object $C \in \mathcal{C}$, a morphism $\theta_C : F(C) \to G(C)$ in $\mathcal{D}$ that commutes with the images of the morphisms from $\mathcal{C}$. More specifically, for any $f : C \to C'$ in $\mathcal{C}$, the following diagram should commute.

$$\begin{array}{ccc} F(C) & \xrightarrow{\theta_C} & G(C) \\ \Big\downarrow{\scriptstyle F(f)} & & \Big\downarrow{\scriptstyle G(f)} \\ F(C') & \xrightarrow{\theta_{C'}} & G(C') \end{array}$$

3.4.1 Properties of natural transformations

Natural transformations can be composed. If $\theta : F \to G$ and $\eta : G \to H$ are natural transformations, then their composition $\eta \circ \theta : F \to H$ is given in components by $(\eta \circ \theta)_C = \eta_C \circ \theta_C$.

For any categories $\mathcal{C}$ and $\mathcal{D}$, we can define the **functor category** $\mathrm{Fun}(\mathcal{C}, \mathcal{D})$; the objects of this category are functors $F : \mathcal{C} \to \mathcal{D}$, and the morphisms are natural transformations $\theta : F \to G$.

We can now define what it means for two functors F and G to be "isomorphic". Specifically, $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{C} \to \mathcal{D}$ are **naturally isomorphic** if and only if they are isomorphic as objects in the functor category $\mathrm{Fun}(\mathcal{C}, \mathcal{D})$. This means that there is a natural transformation $\theta : F \to G$ with an inverse $\eta : G \to F$ such that

$$\eta \circ \theta = \mathrm{id}_F \qquad \text{and} \qquad \theta \circ \eta = \mathrm{id}_G \ .$$

Thankfully, there is an easier way to check if a natural transformation is a natural isomorphism.

Theorem 3.4.1. *A natural transformation $\theta : F \to G$ is a natural isomorphism if and only if each component θ_C is an isomorphism.*

Therefore, if $F, G : \mathcal{C} \to \mathbf{Set}$ are two functors, then a natural transformation $\theta : F \to G$ is an isomorphism if and only if $\theta_C : F(C) \to G(C)$ is a bijection for all C.

This lets us talk about what a representable functor is in general. A functor $F : \mathcal{C} \to \mathbf{Set}$ is **representable** if and only if it is naturally isomorphic to $\mathrm{Hom}(A, -)$ or $\mathrm{Hom}(-, B)$ for some object A or B.

In our examples from last class, we have essentially already shown that the identity functor $\mathrm{id}_{\mathbf{Set}} \cong \mathrm{Mor}(\{*\}, -)$ is representable, as well as the contravariant power set functor $\mathcal{P}(-) \cong \mathrm{Mor}(-, \{0, 1\})$. As another example, on this week's homework, you are basically showing that the forgetful functor $U : \mathbf{Grp} \to \mathbf{Set}$ is representable. In fact, many forgetful functors are representable; we have also already shown that the forgetful functor $V : \mathbf{Vect}_{\Bbbk} \to \mathbf{Set}$ is representable as well whenever we talked about generalized elements, with $V \cong \mathrm{Mor}(\mathbb{R}^1, -)$.

3.4.2 Limit preservation

A functor $F : \mathcal{C} \to \mathcal{C}'$ is said to **preserve limits** if for any diagram $D : \mathcal{J} \to \mathcal{C}$ with a limit L, the object $F(L)$ is a limit of the diagram $F \circ D : J \to \mathcal{C}'$. The functor should also respect the projections $L \to D_i$ for each object D_i of the diagram D. Functors that preserve limits are called **continuous**. If we use the notation $\varprojlim D$ to denote the limit of the diagram D, then this relationship is sometimes written

$$F(\varprojlim D) = \varprojlim F(D)\,.$$

Functors can also **preserve colimits** in the same sense, i.e.

$$F(\varinjlim D) = \varinjlim F(D)\,,$$

where $\varinjlim D$ denotes the colimit of the diagram D. Such functors are called **cocontinuous**.

We can find examples of (co)continuous functors by looking at representable functors.

Theorem 3.4.2. *Covariant representable functors preserve limits.*

For example, in the category of sets, there is a bijection

$$\mathrm{Mor}(A, B \times C) \to \mathrm{Mor}(A, B) \times \mathrm{Mor}(A, C)\,.$$

On the other hand, while one might expect corepresentable functors to preserve colimits, they in fact do something a bit weirder.

Theorem 3.4.3. *Contravariant representable functors map colimits to limits.*

For example, in the category of sets, there is a bijection

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

Here, the colimit $\sqcup$ is turned into a limit $\times$.

As another example, let's look at the contravariant representable functor $\mathcal{P} : \mathbf{Set}^{\mathsf{op}} \to \mathbf{Set}$. Since we know that this functor sends colimits to limits, we can deduce that

- $\mathcal{P}(\emptyset) \cong \{*\}$,
- $\mathcal{P}(A \sqcup B) \cong \mathcal{P}(A) \times \mathcal{P}(B)$ for two sets A and B, and
- $\mathcal{P}(A \sqcup_G B) \cong \mathcal{P}(A) \times_{\mathcal{P}(G)} \mathcal{P}(B)$ assuming we have maps $G \to A$ and $G \to B$ for some set G.

3.5 Equivalence of categories

Let **FinCard** be the category whose objects are the sets $[n] = \{1, 2, \ldots, n\}$ for each $n \in \mathbb{N}$, and whose morphisms are the functions bewteen them. This is a full subcategory of **Set**. For comparison, let **FinSet** be the category whose objects are finite sets and whose morphisms are functions between them. This is also a full subcategory of **Set**. Are these two categories isomorphic?

If **FinCard** and **FinSet** were isomorphic, we would be able to find functors $F : \mathbf{FinCard} \to \mathbf{FinSet}$ and $G : \mathbf{FinSet} \to \mathbf{FinCard}$ such that $G \circ F = \mathrm{id}_{\mathbf{FinCard}}$ and $F \circ G = \mathrm{id}_{\mathbf{FinSet}}$. However, such functors cannot exist! For one, **FinCard** is a small category, whereas **FinSet** is a large (but locally small) category. Therefore, the collections of objects of these two categories cannot be in bijection with one another.

At the same time, these two categories are rather similar. We can come up with an inclusion functor $F : \mathbf{FinCard} \to \mathbf{FinSet}$ rather easily. While F doesn't have an inverse, if we weaken the requirements a bit, we can find a functor $G : \mathbf{FinSet} \to \mathbf{FinCard}$ that is "like an inverse", in that $G \circ F \cong \mathrm{id}_{\mathbf{FinCard}}$ and $F \circ G \cong \mathrm{id}_{\mathbf{FinSet}}$. Here, $\cong$ denotes that these compositions are naturally isomorphic to the identity functors, but may not be equal to them. The functors F and G are called **weak inverses**. The functor G sends a set S with cardinality n to the set $[n]$. Furthermore, G actually chooses a specific isomorphism $S \to [n]$ for each S, so that we can make coherent choices of where to send morphisms between sets. Since we are choosing isomorphisms $S \to [n]$ for each set, this construction requires the axiom of choice, which is not incredibly important but maybe worth noting. The functors F and G described above exhibit an *equivalence of categories* between **FinCard** and **FinSet**.

Definition 3.5.1. An **equivalence of categories** is a pair of functors $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$ such that $G \circ F \cong \mathrm{id}_{\mathcal{C}}$ and $F \circ G \cong \mathrm{id}_{\mathcal{D}}$.

Two categories $\mathcal{C}$ and $\mathcal{D}$ are **equivalent** if they are related by such a pair of functors. We denote this relationship by $\mathcal{C} \simeq \mathcal{D}$.

The functor F above is full and faithful, which is to say that for any objects $[m]$ and $[n]$ of **FinCard**, the induced map $\mathrm{Mor}([m], [n]) \cong \mathrm{Mor}(F([m]), F([n]))$ is a bijection. Furthermore, while F is not surjective on objects, every object of **FinSet** is *isomorphic* to an object in the image of F, since every finite set has a cardinality. This property has a special name.

Definition 3.5.2. A functor $F : \mathcal{C} \to \mathcal{D}$ is **essentially surjective** if and only if every object D of $\mathcal{D}$ is isomorphic to $F(C)$ for some object C of $\mathcal{C}$.

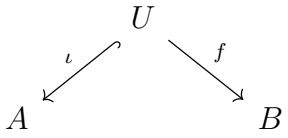
Essentially surjective functors can be thought of as "surjective on objects, up to isomorphism". There is a theorem relating this property to equivalences of categories.

Theorem 3.5.3. *Given a functor $F : \mathcal{C} \to \mathcal{D}$, the following are equivalent:*

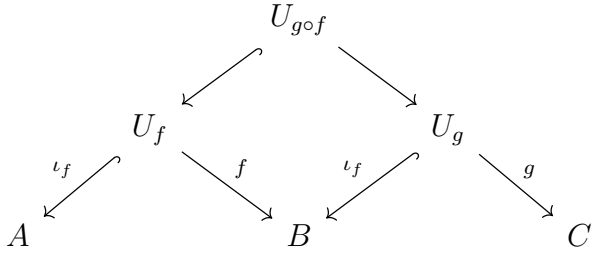
- *F has a weak inverse, and therefore induces an equivalence of categories.*
- *F is full, faithful, and essentially surjective.*

3.5.1 Example: Partial functions

A **partial function** $f : A \to B$ is a subset $U \subseteq A$ along with a function $f : U \to B$. We can think of f as a function on A that is only "partially defined". Visually, we can think of the inclusion map $\iota : U \to A$ and the function $f : U \to B$ as the two maps in a span.



How do we compose partial functions? Given partial functions $f : A \to B$ defined on $U_f \subseteq A$ and $g : B \to C$ defined on $U_g \subseteq B$, we can get a partial function $g \circ f : A \to C$ by defining $U_{g\circ f} = f^{-1}(U_g)$. Diagrammatically, we can see this as the pullback of the two spans.



There is a category **Par** whose objects are sets and whose morphisms are partial functions. While this may seem like an entirely new category, it turns out that it is actually equivalent to the category $\mathbf{Set}_*$ of pointed sets and based maps. Let $F : \mathbf{Par} \to \mathbf{Set}_*$ be the map that takes a set X and appends a base point $*$, i.e. $F(X) = X \sqcup \{*\}$; this set is often abbreviated X_+. Given a partial function $f : X \to Y$, we can define $F(f)$ to be

$$F(f)(x) = \begin{cases} f(x) & x \in U_f \\ * & \text{otherwise} \end{cases}.$$

The functor F has a weak inverse $G : \mathbf{Set}_* \to \mathbf{Par}$ defined on objects as $G((X, x_0)) = X \setminus \{x_0\}$. On a morphism $f : (X, x_0) \to (Y, y_0)$, we define the partial map $G(f) : X \setminus \{x_0\} \to Y \setminus \{y_0\}$ to be defined whenever $x \neq x_0$ and $f(x) \neq y_0$.

Note that $G \circ F$ is actually the identity on **Par**, since F adds a distinguished point which is then removed by G, and similarly for the morphisms. However, $F \circ G$ is not the identity on $\mathbf{Set}_*$, since we are removing the basepoint of (X, x_0) and replacing it with $*$. While (X, x_0) is certainly isomorphic to $(X \setminus \{x_0\} \sqcup \{*\}, *)$, these sets are not equal. Nevertheless, these isomorphisms tell us that $F \circ G$ is naturally isomorphic to $\mathrm{id}_{\mathbf{Set}_*}$. Therefore, **Par** and $\mathbf{Set}_*$ are equivalent as categories.

3.6 Equivalence, continued

3.6.1 Skeletons

Recall the categories **FinCard** and **FinSet** from last class. The objects of **FinCard** were sets $[n] = \{1, \ldots, n\}$, and the objects of **FinSet** were finite sets. Both categories had functions as their morphisms. We saw that these two categories were equivalent, even though they are not equal.

One special property that **FinCard** has is that $[m] \cong [n]$ only when $m = n$. In other words, if two objects of **FinCard** are isomorphic, then they are actually the same object. Categories satisfying this property are called **skeletal**.

Theorem 3.6.1. *Every category $\mathcal{C}$ is equivalent to a skeletal subcategory $\mathcal{D}$ of $\mathcal{C}$.*

Proof. We can use the axiom of choice to choose, for each isomorphism class of objects in $\mathcal{C}$, a representative object. The category $\mathcal{D}$ is then the full subcategory on these representatives. □

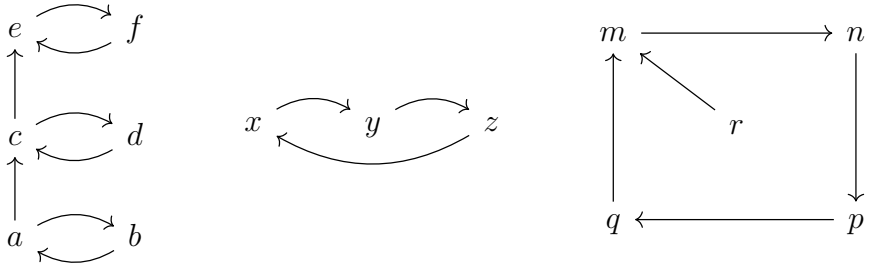
Some examples:

- **FinCard** is a skeletal subcategory of **FinSet**.
- The category **Set** of all sets has a skeletal subcategory **Card** with one object for every possible cardinality that a set could have.
- The category $\mathbf{FinVect}_{\mathbb{R}}$ of finite-dimensional real vector spaces has a skeletal subcategory whose objects are the vector spaces $\mathbb{R}^n$ for each n.

3.6.2 Preorders

A **preorder** on a set X is a reflexive and transitive relation $\leq$. Preorders are like partial orders without the antisymmetry condition. A set equipped with a preorder is sometimes called a "proset", analogous to the poset terminology.

We can think of preordered sets as categories in the same way as posets; given a preordered set $(X, \leq)$, we construct a category $\mathcal{X}$ with an object x for each $x \in X$, and a morphism $x \to y$ for each relation $x \leq y$. For example, here are some Hasse diagrams of preorders.



Unlike with partial orders, Hasse diagram of preorders are not unique! For example, in the middle diagram above, we could instead have added arrows $y \to x$ and $z \to y$ instead of the single arrow $z \to x$, and we would have another non-redundant set of relations.

Lack of asymmetry means that preorders may have elements x and y such that $x \leq y$ and $y \leq x$, but $x \neq y$. Let's denote the relationship $x \leq y$ and $y \leq x$ by $x \sim y$. We can show that $\sim$ is an equivalence relation!

- Reflexivity: It is always true that $x \leq x$, so $x \sim x$.
- Symmetry: If $x \sim y$, then $x \leq y$ and $y \leq x$, so $y \sim x$.
- Transitivity: If $x \sim y \sim z$, then $x \leq y$, $y \leq x$, $y \leq z$ and $z \leq y$. Therefore, by transitivity of $\leq$, we get that $x \leq z$ and $z \leq x$, so $x \sim z$.

Furthermore, the preorder $\leq$ respects this equivalence relation. This means that the equivalence classes of $\sim$ are also preordered by $\leq$; in fact, the equivalence classes form a partial order!

Theorem 3.6.2. *Any preordered set $\mathcal{X}$ is equivalent as a category to a poset $\mathcal{P}$.*

3.6.3 Groupoids

A **group** is a category with one object in which every morphism is an isomorphism. In the same vein, a **groupoid** is a category with potentially any number of objects in which every morphism is an isomorphism.

If groups represent symmetries, then groupoids can be thought of as representing symmetries of things with multiple "states". For example, the dihedral group D_4 represents the symmetries of a 2D square in 3D space. This group has eight elements; four rotations, and four more rotations followed by a flip. Now, let's assume that the two faces of the square are painted different colors. If we flip the square, we now no longer have a symmetry; instead, flips bring us to a different "state". However, we can still encode this behavior in a groupoid, by having two objects; one for each face of the square.

If $\mathrm{Mor}(X, Y)$ is nonempty for any two objects X and Y of a groupoid $\mathcal{G}$, then we say that $\mathcal{G}$ is **connected**. In a connected groupoid, one can show that any two objects have isomorphic endomorphism monoids.

Remark 3.6.3. In fact, the endomorphisms of an object are not just a monoid, they actually form a group. Invertible endomorphisms of an object are called **automorphisms**, and the collection of all automorphisms of an object X form the **automorphism group** $\mathrm{Aut}(X)$. Therefore, we might say that any two objects in a connected groupoid have isomorphic automorphism groups.

In some sense, groupoids are not much different than groups.

Theorem 3.6.4. *Any nonempty connected groupoid $\mathcal{G}$ is equivalent as a category to some group G.*

This group G can be taken to be the automorphism group of any object of $\mathcal{G}$.

3.7 Adjoints

3.7.1 Free Vector Spaces

Recall that a **basis** for a vector space V is a set $B = \{\vec{\beta_i}\}_{i \in I}$ of vectors such that any $\vec{v} \in V$ can be written as a linear combination $c_1\vec{\beta_1} + \dots c_n\vec{\beta_n}$ of vectors in B. Usually, we think of bases as things that we can find given vector spaces, e.g. if $V = \mathbb{R}^4$, then we know that we can pick the standard basis $\{\vec{e}_1, \vec{e}_2, \vec{e}_3, \vec{e}_4\}$.

However, we can also go the other direction! Given a set $S = \{s_1, \dots, s_n\}$. We can define the **free vector space** $\mathbb{R}^S$ on the set S to be the set of all **formal linear combinations** $c_1 s_1 + \cdots + c_n s_n$. Here, the word "formal" indicates that we are not assuming that the elements of S were previously equipped with any notion of addition or scalar multiplication. Instead, we are simply *defining* a vector space $\mathbb{R}^S$ for which S is a basis.

For example, a few elements of the free vector space on the set $S = \{a, b, c\}$ are c, $2a + b$, and $\frac{1}{2}a - b + c$. This vector space is isomorphic, but not equal, to $\mathbb{R}^3$.

As another example, we can think of the vector space $\mathbb{R}[x]$ of polynomials in one variable as the free vector space generated by the set $\{1, x, x^2, \dots\}$. Any polynomial in x is a linear combination of these elements, and there are no linear dependencies between them.

This "free vector space" construction is also **functorial**; if we have a function $f : S \to S'$ of sets, then we get an induced linear transformation $\mathbb{R}^S \to \mathbb{R}^{S'}$ by sending

$$c_1 s_1 + \cdots + c_n s_n \mapsto c_1 f(s_1) + \cdots + c_n f(s_n)\,.$$

We may denote this functor

Now, let's think about how we often define maps between vector spaces. If we have two vector spaces V and W, and we want to define a linear transformation $T : V \to W$, we often do so by specifying $T(\vec{\beta_i})$ for each basis vector β_i in a basis for V. In other words, we are specifying a *function* from a basis B of V to the underlying set of the vector space W. It is actually a rather remarkable fact that this is equivalent to specifying a linear transformation. We can formalize this correspondence using the forgetful functor $U : \mathbf{Vect}_{\mathbb{R}} \to \mathbf{Set}$.

Theorem 3.7.1. *For any vector space W, there is a bijection*

$$\mathrm{Mor}(\mathbb{R}^S, W) \cong \mathrm{Mor}(S, U(W))\,.$$

In fact, this bijection extends to natural isomorphisms in either parameter, i.e.

$$\mathrm{Mor}(\mathbb{R}^-, W) \cong \mathrm{Mor}(-, U(W))$$

and

$$\mathrm{Mor}(\mathbb{R}^S, -) \cong \mathrm{Mor}(S, U(-))\,.$$

Given this kind of relationship, we would say that the functors $\mathbb{R}^- : \mathbf{Set} \to \mathbf{Vect}_{\mathbb{R}}$ and $U : \mathbf{Vect}_{\mathbb{R}} \to \mathbf{Set}$ are *adjoint functors*.

Definition 3.7.2. Two functors $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$ are **adjoint** if and only if there exist isomorphisms

$$\mathrm{Mor}(F(X), Y) \cong \mathrm{Mor}(X, G(Y))$$

for all objects X of $\mathcal{C}$ and Y of $\mathcal{D}$. Furthermore, these isomorphisms should extend to a natural isomorphism $\mathrm{Mor}(F(-),-) \cong \mathrm{Mor}(-,G(-))$ as functors $\mathcal{C}^{\mathsf{op}} \times \mathcal{D} \to \mathbf{Set}$. The functor F is called the **left adjoint**, and G is called the **right adjoint**.

3.7.2 Currying

Let's consider a pair of adjoint functors from **Set** to itself. Fix a set B, and let $F = - \times B$ be the functor that sends any input set A to $A \times B$. Let $G = \mathrm{Mor}(B, -)$ be the functor that sends any input set C to $\mathrm{Mor}(B, C)$. These two functors are adjoint, which is to say that for any sets A and C, we have the following natural isomorphism:

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

Intuitively, this tells us that a function from $A \times B \to C$ is the same as a function from A to $\mathrm{Mor}(B, C)$. We can explicitly write down a bijection θ on an input $\varphi \in \mathrm{Mor}(A \times B, C)$ as

$$\theta(\varphi)(a)(b) = \phi(a, b) \,.$$

The inverse θ^{-1} on $\psi \in \mathrm{Mor}(A, \mathrm{Mor}(B, C))$ is

$$\theta^{-1}(\psi)(a, b) = \psi(a)(b) \,.$$

Applying θ to a function is known as **currying** in computer science.

3.7.3 Free-forgetful Adjunctions

Our first example of the free vector space functor $\mathbb{R}^-$ is actually a special case of a **free-forgetful adjunction**. This is the general term for any situation where a forgetful functor $U : \mathcal{D} \to \mathcal{C}$ has a left adjoint $F : \mathcal{C} \to \mathcal{D}$. "Forgetful functor" is really an informal designation, but usually the objects of $\mathcal{D}$ are thought of as "objects of $\mathcal{C}$ with extra struture".

For example, the forgetful functor $U : \mathbf{Grp} \to \mathbf{Set}$ has a left adjoint $F : \mathbf{Set} \to \mathbf{Grp}$ given by the **free group functor**. This functor takes as input a set S and constructs the group $F(S)$ generated by the elements of S. In other words, elements of $F(S)$ are sequences of elements of S with concatenation as the group operation, subject to some relations.

More examples of forgetful functors with "free" left adjoints:

- $U : \mathbf{Set}_* \to \mathbf{Set}$
- $U : \mathbf{Ab} \to \mathbf{Set}$
- $U : \mathbf{Ab} \to \mathbf{Grp}$
- $U : \mathbf{Pos} \to \mathbf{Set}$
- $U : \mathbf{Ring} \to \mathbf{Grp}$ (the group of units functor)

3.7.4 Limit Preservation

Theorem 3.7.3. *If $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$ are a pair of adjoint functors, where F is the left adjoint and G is the right adjoint, then F preserves colimits, and G preserves limits.*

For example, the above theorem combined with the free-forgetful adjunction between **Set** and **Grp** tells us that

$$F(X \sqcup Y) \cong F(X) \star F(Y) \qquad \text{and} \qquad U(G \times H) \cong U(G) \times U(H) \,.$$

3.8 Cartesian closed categories

3.8.1 Adjoints

Recall that two functors $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$ are **adjoint** if and only if there exist isomorphisms

$$\mathrm{Mor}(F(X), Y) \cong \mathrm{Mor}(X, G(Y))$$

for all objects X of $\mathcal{C}$ and Y of $\mathcal{D}$. Furthermore, these isomorphisms should extend to a natural isomorphism $\mathrm{Mor}(F(-), -) \cong \mathrm{Mor}(-, G(-))$ as functors $\mathcal{C}^{\mathsf{op}} \times \mathcal{D} \to \mathbf{Set}$. The functor F is called the **left adjoint**, and G is called the **right adjoint**.

We have many examples of adjoint pairs of functors, including

- the product functor $- \times B : \mathbf{Set} \to \mathbf{Set}$ and the set-of-morphisms functor $\mathrm{Mor}(B, -) : \mathbf{Set} \to \mathbf{Set}$,
- the free vector space functor $\mathbb{R}^{-} : \mathbf{Set} \to \mathbf{Vect}_{\mathbb{R}}$ and the forgetful functor $U : \mathbf{Vect}_{\mathbb{R}} \to \mathbf{Set}$, and
- the abelianization functor $-^{\mathrm{ab}} =: \mathbf{Grp} \to \mathbf{Ab}$ and the forgetful functor $U : \mathbf{Ab} \to \mathbf{Grp}$.

Today, we're going to look at more examples like the first one above.

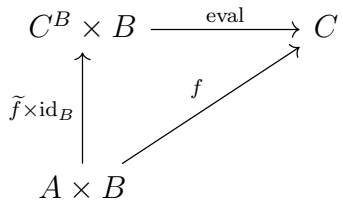
3.9 Cartesian closed categories in general

A category $\mathcal{C}$ which has finite products is called **Cartesian closed** if the product functor $- \times B : \mathcal{C} \to \mathcal{C}$ has a right adjoint for all objects B. When it exists, this adjoint is often denoted $-^B$ or $[B, -]$. In formulas, this means that

$$Mor(A \times B, C) \cong \mathrm{Mor}(A, C^B) \qquad \text{i.e.} \qquad \mathrm{Mor}(A \times B, C) \cong \mathrm{Mor}(A, [B, C])$$

for all objects A, B, and C of $\mathcal{C}$. The functor $[-, -] : \mathcal{C}^{\mathsf{op}} \times \mathcal{C} \to \mathcal{C}$ is often called the **internal hom** or **exponential**. Note that $[-, -]$ is different from $\mathrm{Mor}(-, -)$ when $\mathcal{C} \neq \mathbf{Set}$, as the latter is a functor $\mathcal{C}^{\mathsf{op}} \times \mathcal{C} \to \mathbf{Set}$. Therefore, the right adjoint to the product functor gives us a way to think about the morphisms as forming an object of the category, instead of just a set.

We can define what it means for an object C^B to be an exponential in more concrete terms as well. Given two objects B and C of $\mathcal{C}$, the exponential is an object C^B along with an "evaluation" morphism $\mathrm{eval} : C^B \times B \to C$. Furthermore, the exponential should satisfy the universal property that for any object A and morphism $f : A \times B \to C$, there is a unique morphism $\widetilde{f} : A \to C^B$ such that the following diagram commutes.



3.9.1 Propositions

One interesting example of a Cartesian closed category is the category of logical propositions[1]. Let **Prop** denote the category whose objects are logical propositions, and let there be a morphism from P to Q exactly when we can prove Q assuming P holds. The product in **Prop** is the *logical and* operator, i.e. $P \times Q = P \wedge Q$. The exponential in **Prop** is the *conditional*, i.e. $Q^P = P \implies Q$. Note that in previous examples, the exponential was like a morphism set with extra structure, whereas this case is a bit different.

3.9.2 Posets

The category **Pos** of posets is also cartesian closed! Given two order homomorphisms $f, g : P \to Q$, we say that $f \leq g$ if and only if $f(p) \leq g(p)$ for all $p \in P$. With this ordering, the set $\mathrm{Mor}(P, Q)$ is itself a poset $[P, Q]$, and we have natural isomorphisms

$$\mathrm{Mor}(P \times Q, R) \cong \mathrm{Mor}(P, [Q, R])$$

for all posets P, Q, and R.

3.9.3 Closed categories

Some categories have exponential objects, but they do not arise as adjunct functors of the product. In general, a category with exponential objects is called a **closed category**; we add the "Cartesian" modifier only when the exponentials and products are related by an adjunction.

[1] There are many such ways to think about logic in categorical terms, but this is a relatively simple one.

3.10 Monoidal categories

3.10.1 Review

Recall that two functors $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$ are *adjoint* iff we have a natural isomorphism

$$\mathrm{Mor}(F(X), Y) \cong \mathrm{Mor}(X, G(Y))$$

for all objects X of $\mathcal{C}$ and Y of $\mathcal{D}$. *Cartesian closed categories* are those that have the property that the product functor $- \times B$ has a right adjoint; we call this adjoint the *exponential* or *internal hom*, and denote it $-^B$ or $[B, -]$.

Some examples of Cartesian closed categories included **Set**, **Pos**, and **Prop**.

It is actually true that in any Cartesian closed category, we can strengthen the adjunction above between the product and exponential to be an isomorphism *in* $\mathcal{C}$, i.e.

$$[X \times Y, Z] \cong [X, [Y, Z]]$$

or, written another way,

$$Z^{X \times Y} \cong (Z^Y)^X .$$

3.10.2 Vector Spaces

One category that we might expect to be Cartesian closed is $\mathbf{Vect}_{\mathbb{R}}$, the category of real vector spaces and linear maps. For example, there is a nice way to consider $\mathrm{Mor}(V, W)$ as a vector space $[V, W]$ itself, since we can add and scale linear transformations. However, we run into issues trying to make such an adjunction between $- \times V$ and $[V, -]$ work.

Consider three vector spaces $\mathbb{R}^a$, $\mathbb{R}^b$, and $\mathbb{R}^c$. If $\mathbf{Vect}_{\mathbb{R}}$ were Cartesian closed, then we would expect that

$$\mathrm{Mor}(\mathbb{R}^a \times \mathbb{R}^b, \mathbb{R}^c) \cong \mathrm{Mor}(\mathbb{R}^a, [\mathbb{R}^b, \mathbb{R}^c]) .$$

Note that $\mathbb{R}^a \times \mathbb{R}^b \cong \mathbb{R}^{a+b}$ and $[\mathbb{R}^b, \mathbb{R}^c] \cong \mathbb{R}^{bc}$. Swapping $\mathrm{Mor}(-,-)$ for $[-,-]$ additionally tells us that

$$[\mathbb{R}^{a+b}, \mathbb{R}^c] \cong [\mathbb{R}^a, \mathbb{R}^{bc}] .$$

This cannot hold as an isomorphism in $\mathbf{Vect}_{\mathbb{R}}$, since $[\mathbb{R}^{a+b}, \mathbb{R}^c] \cong \mathbb{R}^{(a+b)c}$, while $[\mathbb{R}^a, \mathbb{R}^{bc}] \cong \mathbb{R}^{abc}$. Therefore, $\mathbf{Vect}_{\mathbb{R}}$ cannot be a Cartesian closed category.

However, there is a kind of "product" on vector spaces that *does* satisfy this property. The **tensor product** $V \otimes W$ of two vector spaces V and W satisfies the property that $\dim(V \otimes W) = \dim(V) \dim(W)$, so in particular $\mathbb{R}^a \otimes \mathbb{R}^b \cong \mathbb{R}^{ab}$. Given bases $\{v_1, \ldots, v_a\}$ of V and $\{w_1, \ldots, w_b\}$ of W, we can construct $V \otimes W$ as the free vector space on the basis

$$\{v_i \otimes w_j \mid 1 \le i \le a, 1 \le j \le b\} .$$

With this kind of product, we do in fact have an adjunction between $- \otimes V$ and $[V, -]$, i.e.

$$[U \otimes V, W] \cong [U, [V, W]]$$

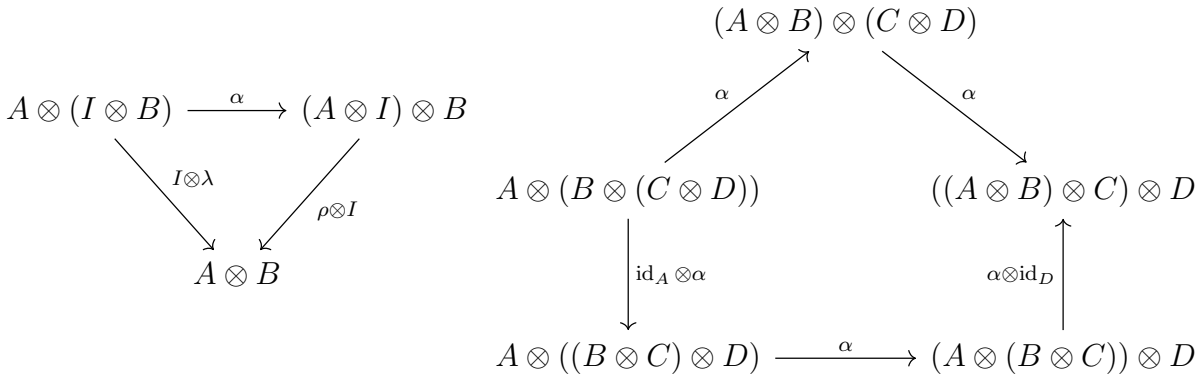
for all vector spaces U, V, and W.

3.10.3 Monoidal Categories in general

A **monoidal category** is a category $\mathcal{C}$ equipped with

- a "product" functor $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$,
- a "unit" object I of $\mathcal{C}$,
- a natural "associator" isomorphism $\alpha : A \otimes (B \otimes C) \cong (A \otimes B) \otimes C$,
- a natural "left unitor" isomorphism $\lambda : I \otimes A \cong A$, and
- a natural "right unitor" isomorphism $\rho : A \otimes I \cong A$.

This structure is required to satisfy the "triangle identity" and "pentagon identity", which say that the following diagrams commute for all objects A, B, C, and D.



Here are some examples of monoidal categories.

- Every category with finite products is a monoidal category, with $\otimes = \times$ and $I = 1$.
- Every category with finite coproducts is a monoidal category, with $\otimes = \sqcup$ and $I = \emptyset$.
- As we saw on the previous page, the category of real vector spaces is monoidal, with $\otimes = \otimes$ and $I = \mathbb{R}^1$.
- The category $\mathbf{Set}_*$ of pointed sets is a monoidal category, with $\otimes = \wedge$ (the **smash product** $A \wedge B = (A \times B)/(A \sqcup B)$) and $I = 2$.
- Any monoid M can be turned into a monoidal category $\mathcal{M}$ by letting the objects of $\mathcal{M}$ be the elements of M, the morphisms of $\mathcal{M}$ be the identities, the monoidal operation $\otimes$ be the multiplication in M, and the unit be the unit in M.

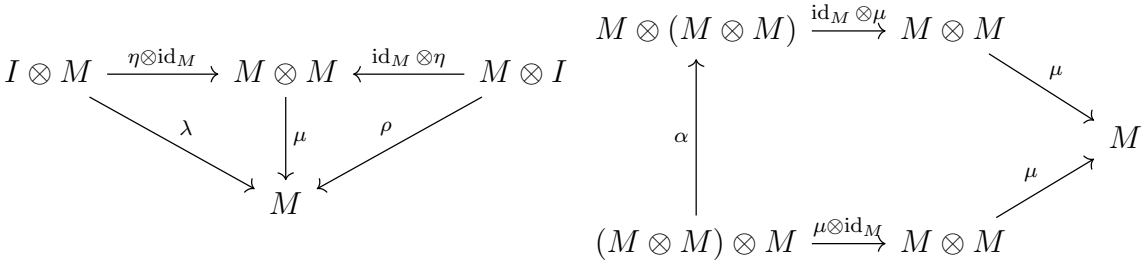
If a monoidal category $\mathcal{C}$ also has exponential objects that are adjoint to the product $\otimes$, then we call $\mathcal{C}$ a **closed monoidal category**. These include all Cartesian closed categories, as well as the examples above of real vector spaces and pointed sets.

3.10.4 Monoids in Monoidal Categories

Given a monoidal category $\mathcal{C}$, we can define what it means for an object in $\mathcal{C}$ to be itself a monoid! A **monoid in** $\mathcal{C}$ is an object M of $\mathcal{C}$ equipped with

- a map $\mu : M \otimes M \to M$ and
- a map $\eta : I \to M$

satisfying a few laws that say that μ is associative and unital with respect to the associator α and unitors λ and ρ.



Monoids in a category describe what it means to give an object some extra structure involving multiplication. For example:

- A monoid in **Set** is just a monoid.
- A monoid in the category **Ab** of abelian groups is a ring.
- A monoid in $\mathbf{Vect}_{\mathbb{R}}$ is called an **associative $\mathbb{R}$-algebra**. As an example, the complex numbers $\mathbb{C}$ are an $\mathbb{R}$-algebra if we think of them as vectors in $\mathbb{R}^2$ with complex multiplication.
- A monoid in the category **Mon** of monoids is a *commutative* monoid.

This last fact follows from a famous proof called the **Eckmann-Hilton argument**, which says that if a set M is a monoid in two ways, and these ways are compatible with each other, then both monoid structures are actually the same commutative monoid. Essentially, if $(M, \circ, 1_\circ)$ and $(M, \star, 1_\star)$ are both monoids, and these operations are compatible with each other in the sense that

$$(a \circ b) \star (c \circ d) = (a \star c) \circ (b \star d)\,,$$

then $1_\circ = 1_\star$ and $\circ = \star$ and $\circ = \star$ is commutative.

The Eckmann-Hilton argument. First, we will prove that the identities are the same.

$$\begin{aligned} 1_\star &= 1_\star \star 1_\star \\ &= (1_\star \circ 1_\circ) \star (1_\circ \circ 1_\star) \\ &= (1_\star \star 1_\circ) \circ (1_\circ \star 1_\star) \\ &= 1_\circ \circ 1_\circ \\ &= 1_\circ \end{aligned}$$

Now, we don't need to distinguish between $1_\star$ and $1_\circ$, so we'll just call them 1. Next, we will prove that the operations are the same.

$$\begin{aligned} a \star b &= (a \circ 1) \star (1 \circ b) \\ &= (a \star 1) \circ (1 \star b) \\ &= a \circ b \end{aligned}$$

At this point, we will just use $\circ$ to denote either operation, since they are equal. Finally, we will prove that the operation is commutative.

$$\begin{aligned} a \circ b &= (1 \circ a) \circ (b \circ 1) \\ &= (1 \circ b) \circ (a \circ 1) \\ &= b \circ a \end{aligned}$$

□

3.11 The Yoneda embedding

3.11.1 Presheaves

Recall that a functor $\mathcal{C} \to \mathbf{Set}$ is *representable* if it is naturally isomorphic to one of the form $\mathrm{Mor}(X, -)$ for some $X \in \mathcal{C}$. Dually, a functor $\mathcal{C}^{\mathsf{op}} \to \mathbf{Set}$ is representable iff it is naturally isomorphic to one of the form $\mathrm{Mor}(-, Y)$ for some $Y \in \mathcal{C}$. Today, we will be interested in functors with the same signature, representable or not.

How can we think about a functor $F : \mathcal{C} \to \mathbf{Set}$? What is such an object really telling us? When $\mathcal{C}$ is a poset, one perspective to take is that a functor $\mathcal{C} \to \mathbf{Set}$ is a "filtered set"; each piece $F(p)$ is the image of some poset element $p \in \mathcal{C}$, and if $p \leq q$, then $F(p) \subseteq F(q)$. We can recover the "underlying set" by taking the union of these pieces, i.e.

$$\bigcup_{p \in \mathcal{C}} F(p)\,.$$

When $\mathcal{C}$ is a general category, the "parameterized set" interpretation can still be useful; we have a set $F(X)$ for each object X, and maps $F(X) \to F(Y)$ for each map $X \to Y$ that tell us how to identify elements of $F(X)$ with $F(Y)$.

If we fix some category $\mathcal{C}$, we can consider the *functor category* $\mathbf{Set}^{\mathcal{C}}$. Recall that the objects of this category are functors $\mathcal{C} \to \mathbf{Set}$, and the morphisms are natural transformations. More explicitly, if $F, G : \mathcal{C} \to \mathbf{Set}$ are two functors, then a natural transformation $\theta : F \to G$ is a choice of component function $\theta_X : F(X) \to G(X)$ for each object X of $\mathcal{C}$.

Let's talk about a couple interesting examples of functor categories of this form. If Γ is the two-object category

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

from the homeworks, then we have already seen that a functor $\Gamma \to \mathbf{Set}$ is just a (directed multi)graph, and therefore the functor category $\mathbf{Set}^{\Gamma}$ is a category of graphs and graph homomorphisms.

If Δ is the category of finite ordinals, then $\mathbf{Set}^{\Delta^{\mathsf{op}}}$ is the category of *simplicial sets*, which are objects used in algebraic topology to compute invariants such as homology groups. More concretely, a finite ordinal is a totally-ordered set $[n] = \{1 \leq \cdots \leq n\}$, so the category Δ looks like

$$[0] \longrightarrow [1] \;\substack{\longrightarrow\\ \longleftarrow\\ \longrightarrow}\; [2] \;\substack{\longrightarrow\\ \longleftarrow\\ \longrightarrow\\ \longleftarrow}\; \cdots\,.$$

Therefore, the image of a functor $\Delta^{\mathsf{op}} \to \mathbf{Set}$ would look like

$$X_0 \longleftarrow X_1 \;\substack{\longrightarrow\\ \longleftarrow\\ \longrightarrow}\; X_2 \;\substack{\longleftarrow\\ \longrightarrow\\ \longleftarrow\\ \longrightarrow}\; \cdots\,.$$

Intuitively, simplicial sets are like shapes that are built out of points, lines, triangles, and tetrahedra of any dimension. These maps tell us when one shape is the boundary of another, for example.

3.11.2 The Yoneda embedding in general

Recall that for any locally small category $\mathcal{C}$, we have a set-of-morphisms functor

$$\mathrm{Mor}(-,-) : \mathcal{C}^{\mathsf{op}} \times \mathcal{C} \to \mathbf{Set}\,.$$

It turns out that **Cat** is itself Cartesian closed, so we can transform the two-input functor above into a single-input contravariant functor

$$g : \mathcal{C}^{\mathsf{op}} \to \mathbf{Set}^{\mathcal{C}}$$

which sends an object X in $\mathcal{C}$ to the functor $\mathrm{Mor}(X,-)$. We can also transform our two-input functor into a covariant functor

$$h : \mathcal{C} \to \mathbf{Set}^{\mathcal{C}^{\mathsf{op}}}$$

by fixing the other argument instead, i.e. $h(Y) = \mathrm{Mor}(-,Y)$. By convention, this functor h is the one that we will be focusing on. It is called the **Yoneda embedding**, as it turns out that h is full, faithful, and injective on objects (any functor satisfying these properties is called an **embedding**). Intuitively, h lets us view $\mathcal{C}$ as a subcategory of $\mathbf{Set}^{\mathcal{C}^{\mathsf{op}}}$. This is rather surprising; it is not immediately obvious that $\mathcal{C}$ should be a subcategory of $\mathbf{Set}^{\mathcal{D}}$ for *any* category $\mathcal{D}$, much less the choice $\mathcal{D} = \mathcal{C}^{\mathsf{op}}$.

As a specific example, let's consider the case that $\mathcal{C}$ is a group G, which has one object $*$ and a morphism $* \to *$ for each $g \in G$. A functor $\alpha : G \to \mathbf{Set}$ is a choice of set X and a function $\alpha_g : X \to X$ for each $g \in G$; such a structure is called a G-**set**, or an **action** of the group G on the set X. Every group acts on itself via left multiplication, i.e. $\alpha : G \to \mathbf{Set}$ sends $*$ to the underlying set of G, and $\alpha_g(x) = gx$. Dually, every group also acts on itself via right multiplication, as long as we remember to reverse the arrows: $\alpha : G^{\mathsf{op}} \to \mathbf{Set}$ sends $*$ to the underlying set of G, and $\alpha_g(x) = xg$.

The Yoneda embedding gives us a functor $h : G \to \mathbf{Set}^{G^{\mathsf{op}}}$, and evaluating this functor on $*$ gives us a functor $h(*) : G^{\mathsf{op}} \to \mathbf{Set}$. We can think of $h(*)$ as identifying the elements of G^{op} with automorphisms of some set, specifically G itself, with G^{op} acting by right multiplication. Therefore, h identifies elements of G with automorphisms of $h(*)$ that commute with right multiplication, which are exactly the automorphisms induced by left multiplication.

Proof. We want to show that the automorphisms of G that commute with right multiplication are exactly the left multiplications. Let $f : G \to G$ commute with right multiplication. Then

$$f(xg) = f(x)g$$

for all $x, g \in G$. Then

$$f(1g) = f(1)g$$

for all $g \in G$, so f is given by left multiplication by $f(1)$. □

This is the essence of **Cayley's theorem**, which says that any finite group G is a subgroup of S_n for $n = |G|$. More generally, any group G is a subgroup of $\mathrm{Aut}(X)$ where X is the cardinality of the underlying set of G.

3.12 The Yoneda lemma

Recall that the Yoneda embedding is the functor

$$\mathrm{Yo} : \mathcal{C} \to \mathbf{Set}^{\mathcal{C}^{\mathrm{op}}}$$

for a locally small category $\mathcal{C}$. This functor sends an object Y in $\mathcal{C}$ to $\mathrm{Mor}(-, Y)$ in $\mathbf{Set}^{\mathcal{C}^{\mathrm{op}}}$. A consequence of the Yoneda lemma is that this functor is full, faithful, and injective on objects; we call such a functor an **embedding**.

3.12.1 Group Actions

Let G be a group, thought of as a one-object category. Recall that a functor $\alpha : G \to \mathbf{Set}$ picks out a set $X = \alpha(*)$ and assigns elements $g \in G$ to automorphisms $\alpha_g : X \to X$. We call this algebraic structure a **G-set** or **group action of G on X**. A natural transformation θ between two functors $\alpha, \beta : G \to \mathbf{Set}$ is called an **equivariant map**; it consists of a single component $\theta_* : X \to X$ that commutes with the actions α and β in the sense that $\theta_*(\alpha_g(x)) = \beta_g(\theta_*(x))$. When the actions α and β are implicit, we often write this as $\theta_*(gx) = g\theta_*(x)$.

While a given group G may act on many different sets X, every group acts on itself by left multiplication, i.e. $X = G$ and $\alpha_g(h) = gh$. The existence of this action is essentially the content of Cayley's theorem.

Theorem 3.12.1 (Cayley's theorem)**.** *Every group is a subgroup of a symmetric group.*

It turns out that we can recover this statement from the Yoneda embedding, with $\mathcal{C} = G$. Specifically, we have a functor

$$\mathrm{Yo} : G \to \mathbf{Set}^{G^{\mathsf{op}}}$$

that sends the object $*$ of G to a functor $\mathrm{Yo}(*) : G^{\mathsf{op}} \to \mathbf{Set}$. This functor $\mathrm{Yo}(*)$ sends the object $*$ of G^{op} to the underlying set of G, and sends an element $g \in G^{\mathsf{op}}$ to the automorphism of G given by right multiplication, i.e. $\mathrm{Yo}(*)(g)(h) = hg$. What does Yo do to morphisms of G? It sends $g \in G$ to a natural transformation $\mathrm{Yo}(g) : \mathrm{Yo}(*) \to \mathrm{Yo}(*)$, which is an equivariant map $G \to G$ commuting with right multiplication. It turns out that all such maps must come from *left multiplication* by some element of g.

Proof. We want to show that the automorphisms of G that commute with right multiplication are exactly the left multiplications. Let $f : G \to G$ commute with right multiplication. Then

$$f(xg) = f(x)g$$

for all $x, g \in G$. Then

$$f(1g) = f(1)g$$

for all $g \in G$, so f is given by left multiplication by $f(1)$. □

Therefore, the Yoneda embedding identifies elements of G with automorphisms of G. Additionally, it states that these automorphisms commute with right multiplication by elements of G, which adds slightly more structure to our original statement of Cayley's theorem.

3.12.2 Posets

What does the Yoneda embedding say about when our category $\mathcal{C}$ is a poset $\mathcal{P}$? In this case, we have a functor

$$\mathrm{Yo} : \mathcal{P} \to \mathbf{Set}^{\mathcal{P}^{\mathsf{op}}}$$

that sends $p \in \mathcal{P}$ to a functor $\mathrm{Yo}(p) : \mathcal{P}^{\mathsf{op}} \to \mathbf{Set}$, which we can think of as a filtered set. The Yoneda embedding also sends a relation $p \to q$ to a natural transformation $\theta : \mathrm{Yo}(p) \to \mathrm{Yo}(q)$, which is a filtered map between these filtered sets. Note that $\mathcal{P}$ may not have originally been defined as a poset of sets and inclusions! Intuitively, we can think about the Yoneda embedding as telling us that every poset is isomorphic to one that consists of sets and inclusions; in this way, it generalizes Cayley's theorem to a wide range of other types of categories.

3.12.3 The Yoneda lemma in general

The **Yoneda lemma** says that for any locally small category $\mathcal{C}$, functor $F : \mathcal{C} \to \mathbf{Set}^{\mathsf{op}}$, and object X of $\mathcal{C}$, there is an isomorphism

$$\mathrm{Nat}(\mathrm{Mor}(-, X), F) \cong F(X)\,.$$

Here, $\mathrm{Nat}(-,-)$ is the set of natural transformations between the two functors; basically the set $\mathrm{Mor}(-,-)$ in the category of categories $\mathbf{Cat}$. Using this notation as well as the Yoneda embedding, we can rewrite this as

$$\mathrm{Mor}_{\mathbf{Cat}}(\mathrm{Yo}(X), F) \cong F(X)\,.$$

A first consequence of the Yoneda lemma comes by setting $F = \mathrm{Yo}(Y)$ for some other object Y of $\mathcal{C}$. This tells us that

$$\mathrm{Mor}_{\mathbf{Cat}}(\mathrm{Yo}(X), \mathrm{Yo}(Y)) \cong \mathrm{Mor}(X, Y)$$

which implies that the Yoneda embedding is full and faithful. This also implies that not only is the Yoneda embedding injective on objects, but it is even injective on objects *up to isomorphism*. If $\mathrm{Yo}(X) \cong \mathrm{Yo}(Y)$, then this isomorphism is in the image of some invertible $f : X \to Y$ in $\mathcal{C}$.

As another consequence of the Yoneda lemma, we obtain an interesting way to prove isomorphisms. For example, if we want to prove that $X \times (Y \sqcup Z) \cong (X \times Y) \sqcup (X \times Z)$, we can look at maps from these objects to any generic object A using the (contravariant) Yoneda embedding $\mathcal{C}^{\mathsf{op}} \to \mathbf{Set}^{\mathcal{C}}$.

$$\begin{aligned}
\mathrm{Mor}(X \times (Y \sqcup Z), A) &= \mathrm{Mor}(Y \sqcup Z, A^X) \\
&= \mathrm{Mor}(Y, A^X) \times \mathrm{Mor}(Z, A^X) \\
&= \mathrm{Mor}(X \times Y, A) \times \mathrm{Mor}(X \times Z, A) \\
&= \mathrm{Mor}((X \times Y) \sqcup (X \times Z), A)
\end{aligned}$$

The general idea of this kind of calculation is to "enlarge" the category $\mathcal{C}$ to a bigger one $\mathbf{Set}^{\mathcal{C}^{\mathsf{op}}}$ that has nicer properties, do math in that category, then bring our results back to $\mathcal{C}$. It is a bit like using complex numbers to solve a real-valued problem.