1 Preliminaries

1.1 Basic Topology

[1.1.1] Definition (*Category*)

A **category** C comprises:

(1) **Class of Objects.** a class of objects $\mathrm{Obj}(\mathsf{C})$.

(2) **Set of Morphisms.** for any two objects $X, Y \in \mathrm{Obj}(\mathsf{C})$, a *set* of morphisms $\mathsf{C}(X, Y)$ whose elements are maps $f : X \to Y$.

(3) **Law of Composition.** a composition rule such that, for any two morphisms $f : X \to Y$ and $g : Y \to Z$, there exists a morphism $gf : X \to Z$.

A category obeys the following rules:

(1) **Associativity of Composition.** That is, for morphisms $f : X \to Y$, $g : Y \to Z$, and $h : Z \to A$, we have that $(hg)f = h(gf)$.

(2) **Identity Morphisms.** For any two objects $X, Y \in \mathrm{Obj}(\mathsf{C})$ and a morphism $f : X \to Y$, there exist identity morphisms $\mathrm{id}_X : X \to X$ and $\mathrm{id}_Y : Y \to Y$ such that $f\mathrm{id}_X = f = \mathrm{id}_Y f$.

We say that C comprises a class of objects due to set-theoretic considerations as to how large a set can be, ideas behind sets of sets, etc. We purposefully treat the maps between two objects as a set for simplicity. Also note that, by the usual argument for identity maps being unique, identity morphisms are unique as well.

[1.1.2] Definition (*Left-Invertibility, Right-Invertibility, Invertibility, Isomorphism*)

Let C be a category, and let $X, Y \in \mathrm{Obj}(\mathsf{C})$ be arbitrary objects. Then $f : X \to Y$ is said to be **left-invertible** (resp. **right-invertible**) if there exists a morphism $g : Y \to X$ such that $gf = \mathrm{id}_X$ (resp. $h : Y \to X$ such that $fh = \mathrm{id}_Y$).

In the case where $f : X \to Y$ has both a left inverse and right inverse, then

$$g = g\mathrm{id}_Y = gfh = \mathrm{id}_X h = h,$$

and the single morphism $g = h$ is said to be the **inverse** of f. A morphism $f : X \to Y$ is said to be an **isomorphism** if it is invertible, and the objects are said to be **isomorphic** (denoted $X \cong Y$).

[1.1.3] Remark (*Isomorphism Classes*)

An isomorphism is a type of equivalence relation—given that it is reflexive, symmetric, and transitive—and so it forms equivalence classes, called isomorphism classes. For example, isomorphisms in Set are **bijections**, and two isomorphic sets are said to have the same cardinality; a **cardinal** is an isomorphism class of sets.

We shift our study of point-set topology from examining an object's structure, its points, and its elements. We instead observe how topological spaces, the objects of Top, behave in relation to other topological spaces. For any object $X \in \mathrm{Obj}(\mathsf{Top})$, we can characterize X by the morphisms that go in and out of X. This description completely characterizes X up to unique isomorphism; that is, if we have a description of how the morphisms in and out of X behave, we know the object belongs to a unique isomorphism class. Indeed, isomorphism is now our new form of equality: since two spaces that are isomorphic are essentially the same up to the labeling of their elements, we are well equipped to study topology through this categorical lens.

[1.1.4] Definition (*Pushforward, Pullback*)

Let C be a category with three objects $X, Y, Z \in \mathrm{Obj}(\mathsf{C})$, and let $f \in \mathsf{C}(X, Y)$. The **pushforward** of f is the map of sets $f_* : \mathsf{C}(Z, X) \to \mathsf{C}(Z, Y)$ such that $f_* : g \mapsto fg$. The **pullback** of f is the map of sets $f^* : \mathsf{C}(Y, Z) \to \mathsf{C}(X, Z)$ such that $f^* : g \mapsto gf$.

$$\begin{tikzcd} X \arrow[r, "f"] & Y \\ Z \arrow[u, "g"] \arrow[ur, "f_*(g)=fg"'] & \end{tikzcd} \qquad \begin{tikzcd} X \arrow[r, "f"] \arrow[dr, "f^*(g)=gf"'] & Y \arrow[d, "g"] \\ & Z \end{tikzcd}$$

Figure. The pushforward (left) and pullback (right) of f.

[1.1.5] Remark (*Intuition Behind Pushforward and Pullback*)

Consider the same setup as in the previous definition. For some $Z \in \mathrm{Obj}(\mathsf{C})$ and some morphism $f : X \to Y$, f induces a pushforward on maps from Z to X. Namely, if $g : Z \to X$ is a morphism, we can apply f so that $fg : Z \to Y$, and so the result is *pushed* to Y from X. On the other hand, f also induces a pullback on maps from Y to Z. Namely, if $g : Y \to Z$, we can apply f so that $gf : X \to Z$, and so the inputs are *pulled* back into X from Y.

[1.1.6] Theorem (*Morphism is Isomorphisms $\iff$ Pushforward and Pullback are Isomorphisms*)

The following are equivalent.

(1) $f : X \to Y$ is an isomorphism.

(2) The pushforward $f_* : \mathsf{C}(Z, X) \to \mathsf{C}(Z, Y)$ is an isomorphism of sets.

(3) The pullback $f^* : \mathsf{C}(Y, Z) \to \mathsf{C}(X, Z)$ is an isomorphism of sets.

Proof. We prove (1) $\iff$ (2) and note (1) $\iff$ (3) follows in a symmetric fashion.

(1) $\implies$ (2): Suppose $f : X \to Y$ is an isomorphism. Then let $g : Y \to X$ be its inverse, and note the pushforward $g_* : \mathsf{C}(Z, Y) \to \mathsf{C}(Z, X)$ is the inverse to the pushforward $f_* : \mathsf{C}(Z, X) \to \mathsf{C}(Z, Y)$.

(2) $\implies$ (1): Suppose $f_* : \mathsf{C}(Z, X) \to \mathsf{C}(Z, Y)$ is an isomorphism. Choose $Z = Y$, and note $f_* : \mathsf{C}(Y, X) \to \mathsf{C}(Y, Y)$. Since f_* is an isomorphism in Set, it is a bijection. By surjectivity there exists some $g : Y \to X$ such that $fg = \mathrm{id}_Y$. Moreover, note that $f_*(\mathrm{id}_X) = f$ and $f_*(gf) = (fg)f = (\mathrm{id}_Y)f = f$, and by injectivity of f_*, note that $gf = \mathrm{id}_X$. Thus g is f's inverse, completing the proof. ❦

Note that category theory studies not only the relationships between objects within a category, but also the maps between categories themselves. A map between categories is called a *functor*, which we will formally define later.

One might wonder if we can simply form a "category of categories," where the objects are categories and the morphisms are functors. Treating this naively causes three major issues:

(1) **Size.** We defined a category such that the collection of morphisms between any two objects must form a set. However, the collection of functors between two categories is often too massive to be considered a standard set.

(2) **Hierarchy.** Unlike basic objects (like sets or groups), categories have their own internal structure. Because of this, we can actually define mappings *between* functors (called natural transformations). A category of categories inherently possesses a "higher" hierarchy that a standard category cannot capture.

(3) **Rigidity.** In a standard category, two objects are isomorphic if there are morphisms F and G such that $G \circ F = \mathrm{id}$ and $F \circ G = \mathrm{id}$. If our objects are categories, this requires a strict one-to-one matching (bijection) of the objects inside them. This violates the philosophy of category theory, which only cares about objects up to isomorphism, not strict equality.

For example, consider the category of *all* finite-dimensional real vector spaces, and a second, smaller category containing only the standard spaces $\{\mathbb{R}^0, \mathbb{R}^1, \mathbb{R}^2, \ldots\}$. Conceptually, these convey the exact same mathematical information, since every finite-dimensional vector space is isomorphic to some $\mathbb{R}^n$. However, they cannot be strictly isomorphic because the first has uncountably many objects, while the second only has a countable amount. There is no way to pair them one-to-one.

Because of this rigidity, we avoid strict isomorphisms of categories and instead use a looser concept called an *equivalence of categories*, which allows us to treat categories as "the same" without requiring a strict matching of their objects.

[1.1.7] Definition (*Functor, Covariance and Contravariance*)

A **functor** F is a mapping from a category C to a category D that ensures the following data.

(1) For every $X \in \mathrm{Obj}(\mathsf{C})$, there is some $FX \in \mathrm{Obj}(\mathsf{D})$.

(2) For any two $X, Y \in \mathrm{Obj}(\mathsf{C})$ and any $f \in \mathsf{C}(X, Y)$, there is some $Ff \in \mathsf{D}(FX, FY)$.

A functor must obey the following properties.

(1) $(Fg)(Ff) = F(gf)$ for any morphisms $f : X \to Y$ and $g : Y \to Z$.

(2) $F(\mathrm{id}_X) = \mathrm{id}_{FX}$ for any $X \in \mathrm{Obj}(\mathsf{C})$.

A functor F is said to be **covariant** if it maps the category C into D as defined above. A functor F is said to be **contravariant** if it maps the category C^{op} into D. Note that C and C^{op} have the same objects, so the first rule doesn't change, but the arrows are reversed, so every morphism $f : X \to Y$ in C is assigned a morphism $Ff : FY \to FX$ in D.

[1.1.8] Remark (*Utility of Contravariant Functors*)

Consider the category of vector spaces over a field k, whose objects are vector spaces over k and whose morphisms are linear maps $T : V \to W$. Consider also the category of dual vector spaces, where objects are of the form $\mathrm{Hom}(V, k)$ for some vector space V and morphisms are dual maps $T^* : W^* \to V^*$. Then consider a functor F between these two categories, one that sends $V \mapsto \mathrm{Hom}(V, k)$ and $T \mapsto T^*$. This is a **contravariant** functor: a morphism in the category of vector spaces is a linear map from V to W, but after the functor is applied, it becomes a map from $W^* = FW$ to $V^* = FV$, an obvious reversal of the morphism's direction.

[1.1.9] Example (*Examples of Functors*)

(1) Consider the functor $F = \mathsf{C}(X, -)$. When given a functor in this notation, objects $Z \in \mathrm{Obj}(\mathsf{C})$ are mapped to an object by replacing $-$ with the object in question, and so $FZ = \mathsf{C}(X, Z)$. Morphisms $f : Y \to Z$ are mapped to a morphism by replacing the $-$ with the domain and codomain, and so $Ff : \mathsf{C}(X, Y) \to \mathsf{C}(X, Z)$, which is exactly the pushforward $Ff = f_*$.

(2) The contravariant functor $\mathsf{C}(-, X)$ from C^{op} to Set maps objects in C to the set of all morphisms from itself to X, and it maps morphisms to their pullbacks. Indeed, if $Y \in \mathrm{Obj}(\mathsf{C})$, then the functor acts by $Y \mapsto \mathsf{C}(Y, X)$, taking Y to the set of all morphisms from Y to X. If $f : Y \to Z$ is a morphism, note that by contravariance it is mapped to a morphism $\mathsf{C}(Z, X) \to \mathsf{C}(Y, X)$, which is exactly the pullback.

(3) The functor $\mathrm{Hom}(-, k)$, where inputs are taken from the category of vector spaces, maps each vector space to its dual and maps each linear map to its dual map.

(4) A forgetful functor acts by forgetting the special features and structure of the domain. For example, a forgetful functor from Top to Set works by mapping all objects (topological spaces) in Top to a set without its topology, and it can map all morphisms (continuous maps) in Top to just a general function.

(5) A free functor is conceptually the "opposite" of a forgetful functor; it adds structure to a set in the most generic, canonical way possible. For example, a free functor from Set to Grp takes a set of generators S and maps it to the free group $F(S)$. We omit the formal definition for now, as it requires the machinery of adjoint functors (to be discussed later).

[1.1.10] Definition (*Faithful, Full, Fully Faithful Functors*)

Let F be a functor from category C to category D. Then the functor induces a map between hom-sets in C and D, defined by

$$\mathsf{C}(X, Y) \to \mathsf{D}(FX, FY) \qquad f \mapsto Ff$$

for any $X, Y \in \mathrm{Obj}(\mathsf{C})$. If the map is

(1) injective, F is called a **faithful functor**.

(2) surjective, F is called a **full functor**.

(3) bijective, F is called a **fully faithful functor**.

Note that fullness and faithfulness are defined entirely on the level of hom-sets (morphisms). They don't place restrictions on how the functor maps the objects of the category. If F is fully faithful, then every morphism $FX \to FY$ in D is the image of exactly one morphism $X \to Y$ in C under the induced map. Thus a fully faithful functor provides a lossless way to go between the the morphisms of the domain category and the codomain category. As mentioned though, there is no method prescribed by a fully faithful functor to go losslessly between the *objects* of the two categories. There is some restriction on how fully faithful functors map objects, but failure of injectivity is still possible.

Before we discuss fully faithful functors any further, we discuss some of the utility of functors in general.

[1.1.11] Theorem (*Functors Preserve Isomorphisms*)

Suppose $F : \mathsf{C} \to \mathsf{D}$ is a functor. If $X \cong Y$ in C, then $FX \cong FY$ in D.

Proof. Let $f : X \to Y$ be an isomorphism with inverse $g : Y \to X$. Applying the functor axioms:

- $g \circ f = \mathrm{id}_X \implies F(g) \circ F(f) = F(\mathrm{id}_X) = \mathrm{id}_{FX}$,
- $f \circ g = \mathrm{id}_Y \implies F(f) \circ F(g) = F(\mathrm{id}_Y) = \mathrm{id}_{FY}$.

Thus, $F(f)$ is an isomorphism with inverse $F(g)$. Note $F(f) : FX \to FY$ and $F(g) : FY \to FX$, and so $FX \cong FY$. ❦

[1.1.12] Remark (*Functors as Invariants*)

If two objects $X, Y \in \mathrm{Obj}(\mathsf{C})$ are "the same", then two objects $FX, FY \in \mathrm{Obj}(\mathsf{D})$ are also "the same." We say FX is an **invariant** of X. We say F computes the invariant for X: for any data shared between isomorphic objects in C, the functor encodes this in D. For example, consider the forgetful functor from Top to Set: this functor acts as an invariant since if any two topological spaces are homeomorphic, their sets are bijective. We may also consider the contrapositive, that if $FX \ncong FY$, then $X \ncong Y$. Once again, using the forgetful functor, one notes that if the underlying sets of topological spaces are not bijective, then they are certainly not homeomorphic.

Now noting that functors act by computing invariants of isomorphism classes in the domain, we ask the question of when are two invariants the same? We study this question by comparing functors.

[1.1.13] Definition (*Natural Transformation*)

Suppose F, G are two functors from C to D. A **natural transformation** η from F to G comprises morphisms $\eta_X : FX \to GX$ for each $X \in \mathrm{Obj}(\mathsf{C})$, where these morphisms make the diagram

$$\begin{tikzcd} FX \arrow[r, "Ff"] \arrow[d, "\eta_X"'] & FY \arrow[d, "\eta_Y"] \\ GX \arrow[r, "Gf"'] & GY \end{tikzcd}$$

commute (i.e., $\eta_Y Ff = Gf\eta_X$). If $\eta_X : FX \to GX$ is an isomorphism for each $X \in \mathrm{Obj}(\mathsf{C})$, we say η is a **natural isomorphism/natural equivalence**, and we say F and G are **naturally isomorphic**, denoted $F \cong G$. The collection of all natural transformations between two functors is denoted $\mathrm{Nat}(F, G)$.

Note that a natural transformation can be viewed as either a map between two functors, or a collection of maps between diagrams, one for each object in C. With these two perspectives, we hope to show that a functor can also be represented as a diagram, which turns out to be the case. While functors can be viewed as processes that map an object to an invariant, we will see later that they can be viewed as diagrams as well.

[1.1.14] Theorem (*Yoneda Lemma*)

For every object X in the category C and for every contravariant functor $F : C^{\text{op}} \to \text{Set}$, we have that

$$\text{Nat}(C(-, X), F) \cong FX.$$

That is, every element of FX is in bijection with a natural transformation $\eta : C(-, X) \to F$.

[1.1.15] Remark (*Yoneda's Lemma for* $C(-, Y)$)

We omit the proof (it follows readily from the diagram for natural transformations), but consider the special case of the contravariant functor $C(-, Y)$. Then we have that

$$\text{Nat}(C(-, X), C(-, Y)) \cong C(X, Y).$$

For any categories C, D, let D^C denote the category whose objects are functors $F : C \to D$ and whose morphisms are natural transformations $\eta : F \to G$. Now let $D = \text{Set}$, and consider the category $D^{C^{\text{op}}} = \text{Set}^{C^{\text{op}}}$. Note that the objects of this category are contravariant functors $F : C^{\text{op}} \to \text{Set}$, and a functor like this is called a **presheaf**. This category itself is quite important since it has nice properties, but for now we will discuss a special functor.

Note that $C(-, X)$ is a contravariant functor from C^{op} to Set, and so it is an object in our category $\text{Set}^{C^{\text{op}}}$. Let us consider the functor $y : C \to \text{Set}^{C^{\text{op}}}$, which sends objects $X \mapsto C(-, X)$ and sends morphisms $f : X \to Y$ to a natural transformation η whose components are all pushforwards of f. That is, for each $Z \in \text{Obj}(C)$, we have that $\eta_Z : C(Z, X) \to C(Z, Y)$, which is exactly the pushforward of f when considering a map $g : Z \to X$.

Consider the map $f \mapsto yf$, which is a map $C(X, Y) \to \text{Set}^{C^{\text{op}}}(FX, FY)$, which is equivalently rewritten as $C(X, Y) \to \text{Nat}(C(-, X), C(-, Y))$. Note that $\text{Nat}(C(-, X), C(-, Y)) \cong C(X, Y)$ by Yoneda Lemma. We can show that f is the inverse of the bijection $\Phi(\eta) = \eta_X(\text{id}_X)$, and so f is a bijection.

These are just examples, but the overarching example is clear: an object can be studied completely by all the morphisms into and out of it.

We will now shift over to basic categorical results in set theory.

[1.1.16] Definition (*Monomorphisms, Epimorphisms*)

Suppose $f : X \to Y$ is a morphism. We say f is a **monomorphism** if f is left cancellative. That is, for any two morphisms $g_1, g_2 : Z \to X$, we have that

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

Conversely, we say f is an **epimorphism** if f is right cancellative. That is, for any two morphisms $g_1, g_2 : Y \to Z$, we have that

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

We denote a monomorphism by $\hookrightarrow$ and an epimorphism by $\twoheadrightarrow$.

[1.1.17] Example (*Monomorphisms and Epimorphisms in* Set)

Note that monomorphisms coincide with injections and epimorphisms coincide with surjections in Set.

If f is an injective function and $f(g_1(x)) = f(g_2(x))$, it is clear that $g_1(x) = g_2(x)$ by injectivity, and so f is left cancellative (and thus monic). On the other hand, we can prove that if f is left cancellative, f is injective by contraposition. If f is not injective, then there exists $x_1 \neq x_2$ for which $f(x_1) = f(x_2)$. Let g_1, g_2 be maps such that $g_k(x) = x_k$, and so $f(g_1(x)) = f(g_1(x))$ but $g_1(x) \neq g_2(x)$, and so f is not left cancellative. Thus a map in Set is injective if and only if it is monic.

If f is a surjective function and $g_1(f(x)) = g_2(f(x))$ for every x, then since every y in the codomain is some $f(x)$, we have $g_1(y) = g_2(y)$ for every y. Thus f is right cancellative (and thus epic). Conversely, we show that if f is not surjective, it is not epic. If f is not surjective, there is some $y_0 \in Y$ not in the image of f. Define $g_1, g_2 : Y \to \{0, 1\}$ such that both maps agree on the image of f (e.g., they both map every $y \in f(X)$ to 1), but $g_1(y_0) = 1$ and $g_2(y_0) = 0$. Then $g_1 f = g_2 f$ but $g_1 \neq g_2$, so f is not right cancellative.

A map that is both monic and epic in Set is also an isomorphism, since it forces a unique inverse for the map. This is not always true in other categories. For example, a continuous bijection in Top is monic and epic, but it is not a homeomorphism (and thus not an isomorphism).

[1.1.18] Definition (*Initial, Final Objects*)

Suppose C is a category. An object $X \in \mathrm{Obj}(\mathsf{C})$ is called **initial** if there is a unique map from X to every other object in C. Conversely, X is called **final** if there is a unique map from every other object in C to X.

It is immediate from the definition that initial and final objects are unique up to unique isomorphism, but we omit the proof due to its ease.

[1.1.19] Example (*Initial and Final Objects in* Set)

The empty set is the initial object in Set. There is exactly one map $f : \{\} \to X$ for any set X, which is the "empty map", or in other words the map that does nothing. Conversely, all singletons $\{*\}$ are terminal in Set. For any set X, there is exactly one map $f : X \to \{*\}$, which is the constant map $f(x) = *$ for every $x \in X$. Note that singletons are unique up to unique isomorphism in Set, abiding by our previous intuition.

[1.1.20] Example (*Cartesian Product in* Set, *Categorically*)

Recall the Cartesian product of two sets $X \times Y$ is the set of all ordered pairs (x, y), with $x \in X$ and $y \in Y$. We have a concrete characterization of what the structure inside a Cartesian product looks like, but we want a more complete characterization. We want to be able to describe morphisms into or out of $X \times Y$. We can do this by interpreting the product categorically.

We form a product $X \times Y$ by taking two sets X, Y together with two projection maps $\pi_X(x, y) = x$ and $\pi_Y(x, y) = y$. We characterize the product of two sets by the property that if we have any two maps $f_X : Z \to X$ and $f_Y : Z \to Y$, we can factor these maps through the product by $f_X = \pi_X \circ h$ and $f_Y = \pi_Y \circ h$, where $h : Z \to X \times Y$ is unique. We can represent this with a commutative diagram.

$$\begin{tikzcd} & Z \arrow[d, "h"] \arrow[ddl, bend right, "f_X"'] \arrow[ddr, bend left, "f_Y"] & \\ & X \times Y \arrow[dl, "\pi_X"] \arrow[dr, "\pi_Y"'] & \\ X & & Y \end{tikzcd}$$

Note that h forces this diagram to commute. If we drew f_X, f_Y out of Z and π_X, π_Y out of $X \times Y$, the diagram wouldn't be "connected" (thus wouldn't commute) until a map was drawn from $Z \to X \times Y$. The universal property of a product completely characterizes every single map going into $X \times Y$ from some set by the relationships $\pi_X \circ h = f_X$ and $\pi_Y \circ h = f_Y$. Using this, we have a straight forward proof that all the sets that satisfy this universal property are unique up to unique isomorphism, and in the language of category theory, we have that $X \times Y$ is well defined.

Note that we can extend the definition of a product from a product of two objects to an arbitrary product by an indexing set Λ, simply by assigning the product a projection map π_λ for each $\lambda \in \Lambda$. This is one nice aspect of the categorical product: it scales nicely. Its main utility, however, is characterizing products in different categories. If a construction in some other category satisfies this universal property, we have established a concrete link between the product in Set with whatever construction is described by the universal property in the other category.

[1.1.21] Example (*Coproduct in* Set, *Categorically*)

Recall the coproduct of two sets $X \sqcup Y$ (often called the disjoint union) is the set of all elements belonging to either X or Y, typically distinguished by indexing tags: $X \sqcup Y = (X \times \{0\}) \cup (Y \times \{1\})$. We have a concrete characterization of what the structure inside a disjoint union looks like, but we want a more complete characterization. We want to be able to describe morphisms into or out of $X \sqcup Y$. We can do this by interpreting the coproduct categorically.

We form a coproduct $X \sqcup Y$ by taking two sets X, Y together with two canonical injection maps $i_X(x) = (x, 0)$ and $i_Y(y) = (y, 1)$. We characterize the coproduct of two sets by the property that if we have any two maps $f_X : X \to Z$ and $f_Y : Y \to Z$, we can factor these maps through the coproduct by $f_X = h \circ i_X$ and $f_Y = h \circ i_Y$, where $h : X \sqcup Y \to Z$ is unique. We can represent this with a commutative diagram.

$$\begin{tikzcd} X \arrow[dr, "i_X"] \arrow[ddr, bend right, "f_X"'] & & Y \arrow[dl, "i_Y"'] \arrow[ddl, bend left, "f_Y"] \\ & X \sqcup Y \arrow[d, "h"] & \\ & Z & \end{tikzcd}$$

Note that h forces this diagram to commute. If we drew f_X, f_Y into Z and i_X, i_Y into $X \sqcup Y$, the diagram wouldn't be "connected" (thus wouldn't commute) until a map was drawn from $X \sqcup Y \to Z$. The universal property of a coproduct completely characterizes every single map going out of $X \sqcup Y$ to some set by the relationships $h \circ i_X = f_X$ and $h \circ i_Y = f_Y$. Using this, we have a straightforward proof that all the sets that satisfy this universal property are unique up to unique isomorphism, and in the language of category theory, we have that $X \sqcup Y$ is well defined.

Note that we can extend the definition of a coproduct from a coproduct of two objects to an arbitrary coproduct by an indexing set

Λ, simply by assigning the coproduct an injection map i_λ for each $\lambda \in \Lambda$. This is one nice aspect of the categorical coproduct: it scales nicely. Its main utility, however, is characterizing coproducts in different categories. If a construction in some other category satisfies this universal property, we have established a concrete link between the coproduct in Set with whatever construction is described by the universal property in the other category.

2 Examples and Constructions