Preliminaries

Our main goal is to introduce notation and terminology so that this book is self-contained.[1]

We assume you are familiar with basic concepts about sets (e.g.: subsets, union, Cartesian product, cardinality, equivalence classes, quotients, etc.), functions (e.g.: injectivity, surjectivity, inverses, (pre)image, etc.), logic (e.g.: quantifiers, implication) and proofs (e.g.: you can write, read and understand proofs),[2] and we will not recall anything here. However, we need to have a little talk about foundations.

Several times in our coverage of category theory, we will use the term **collection** in order to avoid set-theoretical paradoxes. Collections are supposed to behave just like sets except that we will never consider collections containing other collections. We do not make it more formal because there are many ways to do it[3] and none of them are relevant to this course. However, you still need to know why we cannot use sets as is usual in all other courses.

In short, there exist collections of objects that cannot be sets.[4] In our case, we will need to talk about the collection of all sets and the collection of all groups (among others) and they cannot form sets. For the former, it is easy to see because if S is the set of all sets, then it contains all its subsets and hence $\mathcal{P}(S) \subseteq S$, this leads to the contradiction $|\mathcal{P}(S)| \leq |S| < |\mathcal{P}(S)|$.

In the rest of this chapter, we cover the necessary background that we will use in the rest of the book. It is supposed to be a quick and (unfortunately) dry overview of stuff you may or may not have seen, so we will not dwell on explanations, intuitions and motivations.[5] You can safely skip these sections and come back whenever you click on a word or symbol that is defined here. We hope that this will save you from several trips to Wikipedia.

Abstract Algebra

Here we recall definitions, examples and results you may have seen in classes on abstract algebra or linear algebra.[6]

Monoids

Definition 1 (Monoid)**.** A **monoid** is set M equipped with a binary operation $\cdot : M \times M \to M$ called **multiplication** and an *identity* element[7] 1_M satisfying for all

[1] Especially with the heavy use of the `knowledge` package, I felt it was necessary to cover enough background material in order to have the least amount of external links in the book.

[2] The very first things usually taught in early undergraduate mathematics courses.

[3] Most commonly, people use classes or Grothendieck universes. If this sticky point worries you, we suggest you keep it in the back of your mind and go read `https://arxiv.org/pdf/0810.1279.pdf` when you are a bit more comfortable with category theory.

[4] Famous examples include the collection of ordinal numbers which, by the Burali–Forti paradox, cannot be a set and the collection of all sets that do not contain themselves which, by the Russel paradox, cannot be a set.

[5] Contrarily to the other chapters of this book.

[6] Monoids are not commonly covered, but they are simpler than groups and we need them at one point so we present them here.

[7] Some authors call 1_M the **unity** or the **neutral** element.

$x, y, z \in M$

$$(x \cdot y) \cdot z = x \cdot (y \cdot z) \quad \text{and} \quad 1_M \cdot x = x = x \cdot 1_M.$$

If it satisfies $\forall x, y \in M, x \cdot y = y \cdot x$, M is a **commutative monoid**.

Depending on the context, we will refer to a monoid either as M or $(M, \cdot)$ or $(M, \cdot, 1_M)$.

Remark 2. We will quickly drop the $\cdot$ symbol and denote multiplication with plain juxtaposition (i.e.: $xy := x \cdot y$) for monoids and other algebraic structures with a multiplication.

Examples 3. 1. For any set S, the set of function from S to itself form a monoid with the multiplication being composition and the identity being the identity map $s \mapsto s$.

2. The sets $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$ and $\mathbb{R}$ equipped with the operation of addition are all commutative monoids.

3. For any set S, the powerset $\mathcal{P}(S)$ has two simple monoid structures: one where the multiplication is $\cup$ and the identity if $\varnothing \subseteq S$ and the other where multiplication is $\cap$ and the identity is $S \subseteq S$.

Definition 4 (Homomorphism)**.** Let M and N be two monoids, a **monoid homomorphism** from M to N is a function $f : M \to N$ satisfying the following property:

$$f(1_M) = 1_N \quad \text{and} \quad \forall x, y \in M, f(xy) = f(x)f(y).$$

When f is a bijection, we call it a **monoid isomorphism**, say that M and N are **isomorphic** and denote $M \cong N$.

Definition 5 (Submonoid)**.** Given a monoid M, a **submonoid** of M is a subset $N \subseteq M$ containing 1_M that is closed under multiplication (i.e.: $\forall x, y \in N, x \cdot y \in N$).[8]

[8] This implies N is also a monoid with the multiplication and identity inherited from M.

Definition 6 (Kernel)**.** The **kernel** of a homomorphism $f : M \to N$ is the preimage of 1_N: $\ker(f) := f^{-1}(1_N)$. For any homomorphism f, $\ker(f)$ is a submonoid of M.[9]

[9] Similarly, the image of a homomorphism is also a submonoid.

Example 7. The inclusions $(\mathbb{N}, +) \to (\mathbb{Z}, +) \to (\mathbb{Q}, +) \to (\mathbb{R}, +)$ are all monoid homomorphisms with trivial kernel.[10] This implies this is also a chain of inclusions as submonoids.

[10] i.e.: the kernel only contains the identity.

Definition 8 (Monoid action)**.** Let M be a monoid and S a set, an (left) **action** of M on S is an operation $\star : M \times S \to S$ satisfying for all $x, y \in M$ and $s \in S$

$$(x \cdot y) \star s = x \star (y \star s) \quad \text{and} \quad 1_M \star s = s.$$

The data $(M, S, \star)$ will also be called an M–**set** and we may refer to it abusively with S.

Any monoid action has a **permutation representation** defined to be the map

$$\sigma_\star : M \to \Sigma_S = x \mapsto (s \mapsto x \star s).$$

Conversely, a map $\sigma : M \to \Sigma_S$ that satisfies $\sigma(1_M) = \mathrm{id}_S$ and $\sigma(xy) = \sigma(x) \circ \sigma(y)$ for any $x, y \in M$ gives rise to a monoid action $\star_\sigma$ defined by $x \star_\sigma s = \sigma(x)(s)$.[11]

[11] These are inverse operations, i.e.:

$$\sigma_{\star_\sigma} = \sigma \quad \text{and} \quad \star_{\sigma_\star} = \star.$$

Example 9. Any monoid M has a canonical left action on itself defined by $x \star m = xm$ for all $x, m \in M$.

Groups

Definition 10 (Group)**.** A **group** is set G equipped with a binary operation $\cdot : G \times G \to G$ called **multiplication**, an **inverse** operation $(-)^{-1} : G \to G$ and an **identity** element 1_G such that $(G, \cdot, 1_G)$ is a monoid and for all $x \in G$

$$x \cdot x^{-1} = 1_G = x^{-1} \cdot x.$$

If $(G, \cdot, 1_G)$ is a commutative monoid, we say that G is an **abelian group**.

Examples 11.
1. For any set S, the set of bijections from S to itself form a group with the multiplication being composition, the inverse being the set-theoretical inverse and the identity being the identity map $s \mapsto s$. We denote this group $\mathbf{\Sigma}_S$ and call it the group of **permutations** of S.[12]
2. The monoids on $(\mathbb{Z}, +)$, $(\mathbb{Q}, +)$ and $(\mathbb{R}, +)$ are also abelian groups with the inverse of x being $-x$.
3.

[12] For $n \in \mathbb{N}$, we denote Σ_n the group of permutations of $\{1, \dots, n\}$.

Definition 12 (Homomorphism)**.** Let G and H be two groups, a **group homomorphism** from G to H is a monoid homomorphism $f : G \to H$. It follows that[13]

$$\forall x \in G, f(x^{-1}) = f(x)^{-1}.$$

[13] For this, you need to show that inverses are unique.

When f is a bijection, we call it a **group isomorphism**, say that G and H are **isomorphic** and denote $G \cong H$.

Definition 13 (Subgroup)**.** Given a group G, a **subgroup** of G is a submonoid H of G closed under taking inverses (i.e.: $\forall x \in H, x^{-1} \in H$).[14]

[14] This implies H is also a group with the multiplication, inverse and identity inherited from G.

Definition 14 (Quotient)**.** Let G be a group and H a subgroup of G, the *quotient* G/H is the group whose elements are equivalence class of

Definition 15 (Kernel)**.** The **kernel** of a homomorphism $f : G \to H$ is the preimage of 1_H: $\ker(f) := f^{-1}(1_H)$. For any homomorphism f, $\ker(f)$ is a subgroup of G.[15]

[15] Similarly, the image of a homomorphism is also a subgroup.

Definition 16 (Group action)**.** Let G be a group and S a set, an (left) *action* of G on S is a (left) monoid action of G on S.

Example 17. Any group G has a canonical left action on itself defined by $x \star m = xm$ for all $x, m \in G$.

Rings

Fields

Vector Spaces

Order Theory

In this section, we briefly cover some early definitions and results from order theory. Since this subject is not usually taught in undergraduate courses, we spend a bit more time. In fact, we even introduce stuff we will not use later to make sure readers can get more familiar with the most important objects: posets and monotone functions.

Definition 18 (Poset)**.** A **poset** (short for partially ordered set) is a pair $(A, \leq)$ comprising a set A and a binary relation $\leq \subseteq A \times A$ that is

1. **reflexive** ($\forall x \in A, x \leq x$),
2. **transitive** ($\forall x, y, z \in A$ if $x \leq y$ and $y \leq z$ then $x \leq z$), and
3. **antisymmetric** ($\forall x, y \in A$ if $x \leq y$ and $y \leq x$ the $x = y$).

The relation is also called a **partial order**.[16]

[16] If antisymmetry is not satisfied, $\leq$ is called a **preorder**.

Examples 19. 1. The usual non-strict orders ($\leq$ and $\geq$) on $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$ and $\mathbb{R}$ are all partial orders. The strict orders do not satisfy reflexivity.

2. The divisibility relation $\mid$ on $\mathbb{N}$ satisfying $n \mid m$ whenever n divides m is a partial order.

3. For any set S, the powerset of S $\mathcal{P}(S)$ is a poset when equipped with the $\subseteq$ relation.

4. Any subset of a poset inherits a poset structure by restricting the partial order.

For any monoid M, there are three preorders defined by the so-called Green's relations:

$$\forall x, y \in Mx \leq_L y \Leftrightarrow \exists m \in M, x = my$$
$$\forall x, y \in Mx \leq_R y \Leftrightarrow \exists m \in M, x = ym$$
$$\forall x, y \in Mx \leq_J y \Leftrightarrow \exists m, m' \in M, x = mym'$$

Definition 20 (Monotone)**.** A function $f : (A, \leq_A) \to (B, \leq_B)$ between posets is **monotone** (or **order-preserving**) if for any $a, a' \in A$, $a \leq a' \implies f(a) \leq f(a')$.

Example 21. You probably already know lots of monotone functions, but let us give two less intuitive examples. Let $f : S \to T$ be a function, the *image* map of f[17] is the function $\mathcal{P}(S) \to \mathcal{P}(T)$ defined by $S \supseteq X \mapsto f(X) := \{f(x) \mid x \in X\}$. When both powersets are equipped with the inclusion partial order, the image map is monotone because $X \subseteq X' \subseteq S$ implies $f(X) \subseteq f(X')$.

[17] Which we abusively denote f.

The *preimage* map is

$$f^{-1} : \mathcal{P}(T) \to \mathcal{P}(S) = T \supseteq Y \mapsto f^{-1}(Y) := \{y \in S \mid f(y) \in Y\}.$$

It is also order-preserving because $Y \subseteq Y' \subseteq T$ implies $f^{-1}(Y) \subseteq f^{-1}(Y')$.

Fact 22. *The composition of monotone functions between posets is monotone.*

Definition 23 (Dual)**.** The **dual order**[18] of a poset $(A, \leq)$, denoted $(A, \leq)^{\mathrm{op}}$, is the same set equipped with the converse relation $\geq$ defined by

[18] This definition lets us avoid many symmetric arguments.

$$\forall x, y \in A, x \geq y \Leftrightarrow y \leq x.$$

Definition 24. Let $(A, \leq)$ be a poset and $S \subseteq A$, then $a \in A$ is an **upper bound** of S if $\forall s \in S, s \leq a$. Moreover, $a \in A$ is a **supremum** of S, if it is a least upper bound, that is, a is an upper bound of S and for any upper bound a' of S, $a \leq a'$. A supremum of S is denoted $\bigvee S$, but when S contains only two elements, we use the infix notation $s_1 \vee s_2$ and call this a **join**.

A **lower bound** (resp. **infimum**/**meet**) of S is an upper bound (resp. supremum/join) of S in the dual order $(A, \leq)^{\mathrm{op}}$.[19] An infimum of S is denoted $\wedge S$ or $s_1 \wedge s_2$ in the binary case.

Proposition 25. *Infimums and supremums are unique when they exist.*[20]

Definition 26. A **complete lattice** comprises the data $(L, \wedge, \vee, \leq)$ where $(L, \leq)$ is a poset, and $\wedge, \vee : (\mathcal{P}(L), \subseteq) \to (L, \leq)$ are respectively infimum and supremum as defined above.[21] Observe that L has a smallest element $\vee\varnothing$ and a largest element $\wedge\varnothing$ (they are usually called **top** and **bottom** respectively).

Examples 27.
1. For any set S, $(\mathcal{P}(S), \subseteq)$ is a complete lattice: the supremum of a family of subsets is their union and the infimum is their intersection.
2. Defining supremums and infimums on the poset $(\mathbb{N}, |)$ is subtle. When $S \subseteq \mathbb{N}$ is non-empty, $\wedge S$ is the greatest common divisor of all elements in S and $\wedge\varnothing$ is 0 because any integer divides 0. For a finite and non-empty $S \subseteq \mathbb{N}$, $\vee S$ is the least common multiple of all elements in S. If S is infinite, then $\vee S$ is 0 and the supremum of the empty set is 1 because 1 divides any integer.

You might be wondering about possible posets where all infimums exist but not necessarily all supremums or vice-versa, it turns out that this is not possible as shown below.

Lemma 28. *Let $(L, \leq)$ be a poset, then the following are equivalent:*

(i) $(L, \wedge, \vee, \leq)$ is a complete lattice.

(ii) Any $S \subseteq L$ has a supremum.

(iii) Any $S \subseteq L$ has an infimum.

Proof. (i) $\implies$ (ii), (i) $\implies$ (iii) and (ii) + (iii) $\implies$ (i) are all trivial. Also, by using duality, we only need to prove (ii) $\implies$ (iii). For that, it suffices to note that for any $S \subseteq L$, $\wedge S = \bigvee\{a \in L \mid \forall s \in S, a \leq s\}$ is a suitable definition of the infimum.

Defined that way, $\wedge S$ is a lower bound of S because if $s < \wedge S$, then $s < a$ for some lower bound a of S[22], in particular $s \notin S$. Additionally, since we are taking the supremum over all lower bounds of S, no lower bound of S can be greater and we conclude that $\wedge S$ is indeed the infimum of S. $\square$

Definition 29 (Fixpoints)**.** Let $f : (L, \leq) \to (L, \leq)$, a **pre-fixpoint** of L is an element $x \in L$ such that $f(x) \leq x$. A **post-fixpoint** is an element $x \in L$ such that $x \leq f(x)$. A **fixpoint** (or **fixed point**) of f is a pre- and post-fixpoint.

Theorem 30 (Knaester-Tarski)**.** [23] *Let $(L, \wedge, \vee, \leq)$ be a complete lattice and $f : L \to L$ be monotone, then*

1. *The least fixpoint of f is $\mu f := \wedge\{a \in L \mid f(a) \leq a\}$.*
2. *The greatest fixpoint of f is $\nu f := \vee\{a \in L \mid a \leq f(a)\}$.*

[19] Explicityly, $a \in A$ is a lower bound of S if $\forall s \in S, a \leq s$. It is an infimum of S if, in addition to being a lower bound of S, any lower bound a' of S satisfies $a' \leq a$.

[20] This holds by antisymmetry.

[21] Notice that, by definition, these are monotone maps when the domain $\mathcal{P}(L)$ is equipped with the inclusion order. Moreover, if these functions are defined on all of $\mathcal{P}(L)$, all supremums and infimums exist in $(L, \leq)$.

[22] Because $\wedge S$ was the least upper bound for lower bounds of S.

[23] This is actually a weaker version of the Knaester-Tarski theorem which states that the fixpoints of a monotone f form a complete lattice.

Proof. 1. Any fixpoint of f is in particular a pre-fixpoint, thus μf, being a lower bound of all pre-fixpoints, is smaller than all fixpoints. Moreover, because for any pre-fixpoint $a \in L$, $f(\mu f) \leq f(a) \leq a$, $f(\mu f)$ is also a lower bound of the pre-fixpoints, so $f(\mu f) \leq \mu f$. We infer that $f(f(\mu f)) \leq f(\mu f)$, so $f(\mu f)$ is a pre-fixpoint and $\mu f \leq f(\mu f)$. We conclude that μf is a fixpoint by antisymmetry.

The proof of the second item is the proof of the first item done in the dual order.

2. Any fixpoint of f is in particular a post-fixpoint, thus νf, being an upper bound of post-fixpoints, is bigger than all fixpoints. Moreover, because for any post-fixpoint $a \in L$, $a \leq f(a) \leq f(\nu f)$, $f(\nu f)$ is an upper bound of the post-fixpoints, so $\nu f \leq f(\nu f)$. We infer that $f(\nu f) \leq f(f(\nu f))$, so $f(\nu f)$ is a post-fixpoint and $f(\nu f) \leq \nu f$. We conclude that νf is a fixpoint by antisymmetry.

□

Definition 31. Let $(A, \leq)$ be a poset, a **closure operator** on A is a map $c : A \to A$ that is

1. monotone,
2. **extensive** ($\forall x \in A, x \leq c(x)$), and
3. **idempotent** ($\forall x \in A, c(x) = c(c(x))$).[24]

[24] We will use this definition of idempotence in other contexts.

Example 32. The floor ($\lfloor - \rfloor$) and ceiling ($\lceil - \rceil$) operations are closure operators on $(\mathbb{R}, \geq)$ and $(\mathbb{R}, \leq)$ respectively.

Definition 33. Given two posets $(A, \leq)$ and $(B, \sqsubseteq)$, a **Galois connection** is a pair of monotone functions $l : A \to B$ and $r : B \to A$ such that for any $a \in A$ and $b \in B$,

$$l(a) \sqsubseteq b \Leftrightarrow a \leq r(b).$$

For such a pair, we write $l \dashv r : A \to B$.

Lemma 34. *Let $l \dashv r : A \to B$ be a Galois connection, then l and r are monotone.*

Proof. Assume towards a contradiction that $a < a'$ and $l(a) \not\sqsubseteq l(a')$, then because $l(a') \sqsubseteq l(a')$, we infer that $a' \leq r(l(a'))$ and thus, by transitivity, $a \leq r(l(a'))$. However, this contradicts the fact that $l(a) \not\sqsubseteq l(a')$ (using the $\Leftarrow$ of the Galois connection). We conclude that l is monotone.

A symmetric argument works to show that r is monotone. □

Example 35.

Lemma 36. *Let $l \dashv r : A \to B$ be a Galois connection, then $r \circ l : A \to A$ is a closure operator.*

Proof. Because r and l are monotone, $r \circ l$ is clearly monotone. Also, for any $a \in A$, $l(a) \sqsubseteq l(a)$ implying $a \leq r(l(a))$, so $r \circ l$ is extensive.

Now, in order to prove $r \circ l$ is idempotent, it is enough to show that[25]

$$r(l(a)) \geq r(l(r(l(a)))).$$

[25] The $\leq$ inequality follows by extensiveness.

Observe that since $r(b) \leq r(b)$ for any $b \in B$, we have $l(r(b)) \leq b$, thus in particular, with $b = l(a)$, we have $l(r(l(a))) \leq l(a)$. Applying r which is monotone yields the desired inequality. □

Topology

In this section, we introduce the basic terminology of topological spaces. Again we go a bit further than needed to help readers that first lear about topology here. We end this section by recalling some definitions about metric spaces.

Definition 37. A **topological space** is a pair (X, τ), where X is a set and $\tau \subseteq \mathcal{P}(X)$ is closed under arbitrary unions and finite intersections[26] whose elements are called **open sets** of X. We call τ a **topology** on X.

The **complement** of an open set U, denoted U^c, is said to be **closed**.[27]

In the sequel, fix a topological space (X, τ).

Lemma 38. *Let $(C_i)_{i\in I}$ be a family of closed sets of X, then $\cap_{i\in I} C_i$ is closed and if I is finite, $\cup_{i\in I} C_i$ is also closed.*[28]

Proof. Both statements readily follow from DeMorgan's laws and the fact that the complement of a closed set is open and vice-versa. For the first one, DeMorgan's laws yield

$$\bigcap_{i\in I} C_i = \left(\bigcup_{i\in I} C_i^c\right)^c,$$

and the LHS is the complement of a union of opens, so it is closed. For the second one, DeMorgan's laws yield

$$\bigcup_{i\in I} C_i = \left(\bigcap_{i\in I} C_i^c\right)^c,$$

and the LHS is the complement of a finite intersection of opens, so it is closed. □

Lemma 39. *A subset $A \subseteq X$ is open if and only if for any $x \in A$, there exists an open $U \subseteq A$ such that $x \in U$.*

Proof. $(\Rightarrow)$ For any $x \in A$, set $U = A$.

$(\Leftarrow)$ For each $x \in X$, pick an open $U_x \subseteq A$ such that $x \in A$, then we claim $A = \cup_{x\in A} U_x$ which is open[29]. The $\subseteq$ inclusion follows because each $x \in A$ has a set U_x in the union that contains x. The $\supseteq$ inclusion follows because each term of the union is a subset of A by assumption. □

Lemma 40. *A subset $A \subseteq X$ is closed if and only if for any $x \notin A$, there exists an open U such that, $x \in U$ and $U \cap A = \varnothing$.*[30]

Definition 41. Given $A \subseteq X$, the **closure** of A, denoted A^- is the intersection of all closed sets containing A. One can show that A^- is the smallest closed set containing A.[31] Then, it follows that A is closed if and only if $A^- = A$.

Here are more easy results on the closure of a subset.

Lemma 42. *Given $A, B \subseteq X$ then the following statements hold:*

[26] For any family of open sets $\{U_i\}_{i\in I} \subseteq \tau$,

$$\bigcup_{i\in I} U_i \in \tau,$$

and if I is finite,

$$\bigcup_{i\in I} U_i \in \tau.$$

[27] Observe that both the empty set and the whole space are open and closed (sometimes referred to as **clopen**) because

$$\varnothing = \bigcup_{U\in\varnothing} U \text{ and } X = \bigcap_{U\in\varnothing} U \text{ and } \varnothing = X^c.$$

[28] This lemma gives an alternative to the axioms of Definition 37. Indeed, it is sometimes more convenient to define a topological space by giving its closed sets, and you can show the axioms about open sets still hold.

[29] Arbitrary unions of opens are open.

[30] This result is simply a restatement of the last one by setting $A = A^c$.

[31] A^- is closed because it is an intersection of closed sets and any closed sets containing A also contains A^- by definition.

1. $A \subseteq B \implies A^- \subseteq B^-$
2. $A \subseteq A^-$
3. $A^{--} = A^-$
4. $\emptyset^- = \emptyset$
5. $(A \cup B)^- = A^- \cup B^-$

Proof of Lemma 42. 1. By definition, B^- contains B, thus A, but B^- is closed, so it must contain A^-.

2. By definition.
3. A^- is closed, so its closure is itself.
4. 3 applied to $\emptyset$.
5. $\subseteq$ follows because the LHS is the smallest closed set containing $A \cup B$ and the RHS is closed and contains $A \cup B$.
$\supseteq$: Since the RHS is closed, we have $(A^- \cup B^-)^- = A^- \cup B^-$ implying that the RHS is the smallest closed set containing $A^- \cup B^-$. Then, since the LHS is a closed set containing A and B, it contains A^- and B^- and hence must contain the RHS.

□

Remark 43. If we view $\mathcal{P}(X)$ as partial order equipped with the inclusion relation, the previous lemma is about good properties of the function $(-)^- : \mathcal{P}(X) \to \mp(X)$. Namely, we showed in the first three points that it is a monotone, extensive and idempotent, and therefore it is a closure operator.[32]

[32] In fact, this is where the terminology comes from.

Definition 44. A subset $A \subseteq X$ is said to be **dense** (in X) if any non-empty open set intersects A non-trivially, that is, $\forall \emptyset \neq U \in \tau, A \cap U \neq \emptyset$.

Theorem 45 (Decomposition)**.** *Let $A \subseteq X$, then $A = A^- \cap (A \cup (A^-)^c)$, where A^- is closed and $A \cup (A^-)^c$ is dense. This results says that any subset of X can be decomposed into a closed and a dense set.*

Proof. The equality is clear[33] and A^- is closed by definition. It is left to show that $A \cup (A^-)^c$ is dense. Let $U \neq \emptyset$ be an open set. If U intersects A, we are done. Otherwise, we have the following equivalences:

$$U \cap A = \emptyset \Leftrightarrow A \subseteq U^c \Leftrightarrow A^- \subseteq U^c \Leftrightarrow U \subseteq (A^-)^c,$$

where the second $\Rightarrow$ holds because U^c is closed. We conclude $U \cap (A^-)^c \neq \emptyset$. □

[33] We use (in this order) distributivity of $\cap$ over $\cup$, the fact that a set and its complement intersect trivially and the inclusion $A \subseteq A^-$:

$$\begin{aligned} A^- \cap (A \cup (A^-)^c) &= (A^- \cap A) \cup (A^- \cap (A^-)^c) \\ &= A \cup \emptyset \\ &= A \end{aligned}$$

Lemma 46. *A subset $A \subseteq X$ is dense if and only if $A^- = X$.*

Proof. ($\Rightarrow$) Since $(A^-)^c$ is open but it intersects trivially the dense set A, it must be empty, thus A^- is the whole space.

($\Leftarrow$) Let U be an open set such that $U \cap A = \emptyset$, then A is contained in the closed set U^c, but this implies $A^- \subseteq U^c$,[34] thus U is empty. □

[34] Recall that the closure of A is the smallest closed set containing A.

Definition 47. Let $A \subseteq X$, the **interior** of A, denoted A° is the union of all open sets contained in A. Similarly to the closure, we can check that that A° is the largest open subset of A and thus that A is open if and only if $A = A^\circ$.[35]

[35] It also follows that $A \subseteq B \implies A^\circ \subseteq B^\circ$ and that $A^{\circ\circ} = A^\circ$.

We end this section by presenting a largely preferred way of defining a topology that avoid describing all open sets.

Definition 48 (Base)**.** Let X be a set, a **base** B is a set $B \subseteq \mathcal{P}(X)$ such that $X = \cup_{U \in B} U$ and any finite intersection of sets in B can be written as a union of sets in B.

Lemma 49. *Let X and $B \subseteq \mathcal{P}(X)$. If τ is the set of all unions of sets in B, then it is a topology on X. We say that τ is the topology **generated** by B.*

Proof. By assumption, we know that unions of opens are open and finite intersections of sets in B are open. It remains to show that finite intersections of unions of sets in B are also open. Let $U = \cup_{i\in I} U_i$ and $V = \cup_{j\in J} V_j$ with $U_i \in B$ and $V_j \in B$, then by distributivity, we obtain

$$U \cap V = \cup_{i\in I} U_i \bigcap \cup_{j\in J} V_j = \bigcup_{i\in I, j\in J} U_i \cap V_j,$$

so $U \cap V$ is open.[36] The lemma then follows by induction. □

[36] It is a union of opens.

In practice, instead of generating a topology from a base B, we start with any family $B_0 \subseteq \mathcal{P}(X)$ and let B be its closure under finite intersections, which satisfies the axioms of a base. Such a B_0 is often called a **subbase** for the topology generated by B.

Another very useful way to define topological spaces is to consider the topology induced by a metric.

Definition 50 (Metrics space)**.** A **metric space** (X, d) is a set X together with a function $d : X \times X \to \mathbb{R}$ called a **metric** with the following properties for $x, y, z \in X$:

1. $d(x, y) \geq 0$
2. $d(x, y) = 0 \Leftrightarrow x = y$
3. $d(x, y) = d(y, x)$
4. $d(x, y) \leq d(x, z) + d(z, y)$

Definition 51 (Non-expansive)**.** A function between metric spaces $f : (X, d_X) \to (Y, d_Y)$ is said to be **non-expansive**[37] if for all $x, x' \in X$,

$$d_Y(f(x), f(x')) \leq d_X(x, x').$$

[37] Also called 1**–Lipschitz** or **short**.

Fact 52. *The composition of any two non-expansive maps is non-expansive.*

Definition 53 (Open ball)**.** Let (X, d) be a metric space. Given a point $x \in X$ and a non-negative radius $r \in [0, \infty)$, the open ball of radius r centered at x is

$$B_r(x) := \{y \in X \mid d(x, y) < r.$$

Definition 54 (Induced topology)**.** Any metric space (X, d) has an *induced topology* generated by the set of all open balls of X.[38]

In this topology, a set $S \subseteq X$ is open if and only if every point $x \in S$ is contained in an open ball which is contained in S.[39]

[38] This topology is sometimes called the open ball topology.

[39] Equivalently, $\forall x \in S, \exists r > 0, B_r(x) \subseteq S$.

Definition 55 (Convergence)**.** Let (X, d) be a metric space, a sequence $\{p_n\}_{n\in\mathbb{N}} \subseteq X$ **converges** to $p \in X$ if

$$\forall \varepsilon > 0, \exists N \in \mathbb{N}, \forall n \geq N, d(p_n, p) < \varepsilon.$$

Definition 56 (Cauchy sequence)**.** Let (X, d) be a metric space, a sequence $\{p_n\}_{n\in\mathbb{N}} \subseteq X$ is called **Cauchy** if

$$\forall \varepsilon > 0, \exists N \in \mathbb{N}, \forall m, n \geq N \implies d(p_n, p_m) < \varepsilon.$$

Definition 57 (Completeness)**.** A metric space in which every Cauchy sequence converges is called **complete**.

Categories and Functors

Categories

Definition 58 (Oriented graph)**.** An **oriented graph** G consists of a collection of **nodes**/**objects** denoted G_0 and a collection of **arrows**/**morphisms** denoted G_1 along with two maps $s, t : G_1 \to G_0$, so that each arrow $f \in G_1$ has a **source** $s(f)$ and a **target** $t(f)$.

Definition 59 (Paths)**.** A **path** in an oriented graph G is a sequence of arrows $(f_1, \dots, f_k)$ that are **composable** in the sense that $t(f_i) = s(f_{i-1})$ for $i = 2, \dots, k$ as drawn below in (1). The collection of paths of length k[40] in G will be denoted G_k.

[40] The **length** of a path is the number of arrows in it.

$$\bullet \xrightarrow{f_k} \bullet \xrightarrow{f_{k-1}} \bullet \cdots \bullet \xrightarrow{f_2} \bullet \xrightarrow{f_1} \bullet \tag{1}$$

Observe that the notation indicating the direction of the path does not correspond to the usual notation in graph theory. The motivation for this divergence will come shortly as the composition of arrows in a category is defined. The main idea is that, conceptually, arrows coincide more closely with functions between mathematical objects rather than arrows between nodes of a graph.

Definition 60 (Category)**.** An oriented graph $\mathbf{C}$ along with a **composition** map $\circ : \mathbf{C}_2 \to \mathbf{C}_1$ is a **category** if it satisfies the following properties:

1. For any $(f, g) \in \mathbf{C}_2$, $s(f \circ g) = s(g)$ and $t(f \circ g) = t(f)$. This is more naturally understood visually in (2).

$$\begin{tikzcd} \bullet \arrow[r, "g"'] \arrow[rr, bend left, "f\circ g"] & \bullet \arrow[r, "f"'] & \bullet \end{tikzcd} \tag{2}$$

2. For any $(f, g, h) \in \mathbf{C}_3$, $f \circ (g \circ h) = (f \circ g) \circ h$, namely, composition is associative.

3. For any object $A \in \mathbf{C}_0$, there exists an **identity** morphism $u_{\mathbf{C}}(A) \in \mathbf{C}_1$ with A as its source and target that satisfies $u_{\mathbf{C}}(A) \circ f = f$ and $g \circ u_{\mathbf{C}}(A) = g$, for any $f, g \in \mathbf{C}_1$ where $t(f) = A$ and $s(g) = A$.

If the third property of Definition 60 is not satisfied, $\mathbf{C}$ will be referred to as a **semicategory**. Some authors choose to explicit when a category *does* satisfy this property, qualifying it as unital, but this term also has other meanings, hence our preference for the first convention.

Remark 61 (Notation)*.* In general, we will denote categories with uppercase letters typeset with `\mathbf` ($\mathbf{C}$, $\mathbf{D}$, $\mathbf{E}$, etc.), their objects with uppercase letters (A, B, X, Y, Z, etc.) and their morphisms with lowercase letters (f, g, h, etc.). When the category is clear from the context, we denote the identity morphism id_A instead of $u_{\mathbf{C}}(A)$. We say that two morphisms are **parallel** if they have the same source and target.

Observe that since $\circ$ is associative, it induces a unique composition map on paths of any finite lengths, which we abusively denote $\circ : \mathbf{C}_k \to \mathbf{C}_1$. This lets us write $f_1 \circ f_2 \circ \cdots \circ f_k$ with no parentheses. Occasionally, we will mention the **composition of a path** or the **morphism that a path composes to** to mean the image of the path under this map.

Examples 62 (Boring examples)**.** It is really easy to construct a category by drawing its underlying oriented graph and inferring the definition of the composition from it. Starting from the very simple graph depicted in (3), we can infer the definition of a category with a single object and its identity morphism. This category is denoted $\mathbf{1}$, the composition is trivial since $\mathrm{id}_\bullet \circ \mathrm{id}_\bullet = \mathrm{id}_\bullet$.

$$\begin{tikzcd} \bullet \arrow[loop above] \end{tikzcd} \tag{3}$$

Similarly, we construct from the graph in (4) a category with two objects, their identity morphisms and nothing else. The composition is again trivial. This category will be denoted $\mathbf{1}+\mathbf{1}$.[41] More generally, for any collection $\mathbf{C}_0$, there is a category $\mathbf{C}$ whose collection of objects is $\mathbf{C}_0$ and whose collection of morphisms is $\mathbf{C}_1 := \{\mathrm{id}_X \mid X \in \mathbf{C}_0\}$. The composition map is completely determined by the third property in Definition 60.[42]

$$\begin{tikzcd} \bullet_1 \arrow[loop above] & \bullet_2 \arrow[loop above] \end{tikzcd} \tag{4}$$

[41] This name will be explained in

[42] i.e.: for any $X \in \mathbf{C}_0$, $\mathrm{id}_X \circ \mathrm{id}_X = \mathrm{id}_X$.

The graph in (5) corresponds to the category with objects $\{A, B\}$ and morphisms $\{\mathrm{id}_A, \mathrm{id}_B, f\}$.

$$\begin{tikzcd} A \arrow[loop left, "\mathrm{id}_A"] \arrow[r, "f"] & B \arrow[loop right, "\mathrm{id}_B"] \end{tikzcd} \tag{5}$$

The composition map is then completely determined by the properties of identity morphisms.[43] This category is denoted $\mathbf{2}$, note however that $\mathbf{1}+\mathbf{1} \neq \mathbf{2}$. Starting now, we will omit the identity morphisms from the diagrams (as is usual in the literature) for clarity reasons; they would hinder readability without adding information.

[43] i.e.: $f \circ \mathrm{id}_A = f$, $\mathrm{id}_B \circ f = f$, $\mathrm{id}_A \circ \mathrm{id}_A = \mathrm{id}_A$ and $\mathrm{id}_B \circ \mathrm{id}_B = \mathrm{id}_B$

It is not always as straightforward to construct a category from an oriented graph. For instance, if two distinct arrows have the same source and target, they must be explicitly drawn and the ambiguity in the composition must be dealt with. The graph in (6) is problematic: it has two distinct paths of length two starting at the top-left corner and ending at the bottom-right corner. Since the composition of these paths can be equal to any of the two distinct morphisms between these corners, there is no obvious category corresponding to this graph.

$$\begin{tikzcd} \bullet \arrow[r] \arrow[d] \arrow[dr, shift left] \arrow[dr, shift right] & \bullet \arrow[d] \\ \bullet \arrow[r] & \bullet \end{tikzcd} \tag{6}$$

Diagram (37) shows a very important example of a simple category that *handles* this problem.

$$\begin{tikzcd} \bullet \arrow[r] \arrow[d] & \bullet \arrow[d] \\ \bullet \arrow[r] & \bullet \end{tikzcd} \tag{7}$$

It is implicitly stating that the bottom and top paths compose to the same morphism, the latter is thus absent of the diagram. This category is called the **commutative square** and is denoted $\mathbf{2} \times \mathbf{2}$.[44]

[44] This notation will be explained in Definition 89.

Definition 63 (Commutativity)**.** The term **commutative** is generalized to arbitrary diagrams, it means that any two paths of length bigger than 1 which have the same source and target must compose to the same morphism.

Warning 64. Diagrams are not commutative by default. As our usage of commutative diagrams will ramp up in the following chapters,

Remark 65 (Convention). As you will see in this book, commutative diagrams are used quite a lot in category theory, yet there is no standard definition that everyone systematically uses.[45] For this reason, I decided to pick my favorite definition of commutativity which is somewhat uncommon (the constraint on the length is not usual).

[45] This does not really lead to many misunderstandings anyway because what is meant by a diagram is made clear from the context.

Before moving on to more interesting examples, we introduce the Hom notation.

Definition 66 (Hom)**.** Let $\mathbf{C}$ be a category and $A, B \in \mathbf{C}_0$ be objects, the collection of all morphisms going from A to B is

$$\mathrm{Hom}_{\mathbf{C}}(A, B) := \{f \in \mathbf{C}_1 \mid s(f) = A \text{ and } t(f) = B\}.$$

This leads to an alternative way of defining the morphisms of $\mathbf{C}$, namely, one can describe $\mathrm{Hom}_{\mathbf{C}}(A, B)$ for all $A, B \in \mathbf{C}_0$ instead of describing all of $\mathbf{C}_1$ at once. Defining the morphisms this way also means takes care of the source and target functions implicitly.

Remark 67 (Notation). Some authors choose to denote the collection of morphisms between A and B with $\mathbf{C}(A, B)$. We prefer to use the latter notation when working with 2–categories[46] to highlight the fact that $\mathbf{C}(A, B)$ has more structure. Other authors use hom with a lowercase "h", our choice here is arbitrary.

[46] c.f. Definition 209.

Definition 68 (Smallness)**.** A category $\mathbf{C}$ is called **small** if the collections of objects and morphisms are sets. If for all objects $A, B \in \mathbf{C}_0$, $\mathrm{Hom}_{\mathbf{C}}(A, B)$ is a set, $\mathbf{C}$ is said to be **locally small** and $\mathrm{Hom}_{\mathbf{C}}(A, B)$ is called a **hom–set**. A category that is not small can be referred to as **large**.

Example 69 (**Set**)**.** The category **Set** has the collection of sets as its objects and for any sets X and Y, $\mathrm{Hom}_{\mathbf{Set}}(X, Y)$ is the set of all the functions from X to Y. The composition map is given by composition of functions, the associativity follows from the definition and the identity maps serve as the identity morphisms. This category is locally small but not small.[47]

[47] By Russel's paradox.

Example 70. Let $(X, \leq)$ be a partially ordered set, then X can be viewed as a category with elements of X as its objects. For any $x, y \in X$, the hom–set $\mathrm{Hom}_X(x, y)$ contains a single morphism if $x \leq y$ and is empty otherwise. The identity morphisms arise from the reflexivity of $\leq$. Since every hom–set contains at most one element and $\leq$ is transitive, the composition map is completely determined. Detailing this out, if $f : x \to y$ and $g : y \to z$ are morphisms, then we know that $x \leq y$ and $y \leq z$. Thus, transitivity implies that $x \leq z$ and there is a unique morphism $x \to z$ so it must be $g \circ f$.

If a category corresponds to this construction for some poset, it is called **posetal**. In (8), we depict the posetal category associated to $(\mathbb{N}, \leq)$. The arrows between numbers n and $n+k$ are omitted for $k > 1$ as they can be inferred by the composition $n \leq n+1 \leq n+2 \leq \cdots \leq n+k$.

$$\overset{0}{\bullet} \longrightarrow \overset{1}{\bullet} \longrightarrow \overset{2}{\bullet} \longrightarrow \cdots \tag{8}$$

As a particular case of posetal categories, let (X, τ) be a topological space and note that the inclusion of open sets is a partial order on τ. Thus X has a corresponding posetal category. More explicitly, the objects are open sets and for any $U, V \in \tau$, the hom–set $\mathrm{Hom}_X(U, V)$ contains the inclusion map i_{UV} if $U \subseteq V$ and is empty otherwise. This category will be denoted $\mathcal{O}(X, \tau)$ or $\mathcal{O}(X)$.

Example 71 (Single object categories)**.** If a category $\mathbf{C}$ has a single object $*$, then the only morphisms go from $*$ to $*$. In particular, $\mathbf{C}_1 = \mathrm{Hom}_{\mathbf{C}}(*, *)$ and $\mathbf{C}_2 = \mathbf{C}_1 \times \mathbf{C}_1$. Then, the associativity of $\circ$ and existence of id_* makes $(\mathbf{C}_1, \circ)$ into a monoid.

Conversely, a monoid $(M, \cdot)$ can be represented by a single object category M, where $\mathrm{Hom}_M(*, *) = M$ and the composition map is the monoid operation.

Since many algebraic structures have an associative operation with an identity element, this yields a fairly general construction. The single object category associated to a monoid or group G will be denoted $\mathbf{B}(G)$ and referred to as the **delooping** of G.

The natural numbers can also be endowed with the monoid structure of addition, thus a particular instance of a single object category is the delooping of $(\mathbb{N}, +)$. Notice that this category is very different from the posetal category $(\mathbb{N}, \leq)$. In the former, $\mathbb{N}$ is in correspondence with the morphisms while in the latter, it is in correspondence with the objects.

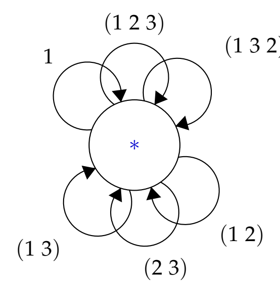


Figure 1: The delooping of the symmetric group S_3, aka $\mathbf{B}(S_3)$.

A lot of simple examples of large categories arise as subcategories of **Set**.

Definition 72 (Subcategory)**.** Let $\mathbf{C}$ be a category, a category $\mathbf{C}'$ is a **subcategory** of $\mathbf{C}$ if, the following properties are satisfied.

1. The objects and morphisms of $\mathbf{C}'$ are objects and morphisms of $\mathbf{C}$ (i.e.: $\mathbf{C}'_0 \subseteq \mathbf{C}_0$ and $\mathbf{C}'_1 \subseteq \mathbf{C}_1$).
2. The source and target maps of $\mathbf{C}'$ are the restrictions of the source and target maps of $\mathbf{C}$ on $\mathbf{C}'_1$ and for every morphism $f \in \mathbf{C}'_1$, $s(f), t(f) \in \mathbf{C}'_0$.
3. The composition map of $\mathbf{C}'$ is the restriction of the composition map of $\mathbf{C}$ on $\mathbf{C}'_2$ and for any $(f, g) \in \mathbf{C}'_2$, $f \circ_{\mathbf{C}'} g = f \circ_{\mathbf{C}} g \in \mathbf{C}'_1$.
4. The identity morphisms of objects in $\mathbf{C}'_0$ are the identity morphisms of objects in $\mathbf{C}_0$, i.e.: $u_{\mathbf{C}}(A) = u_{\mathbf{C}'}(A)$ when $A \in \mathbf{C}'_0$.

Intuitively, one can see $\mathbf{C}'$ as being obtained from $\mathbf{C}$ by removing some objects and morphisms, but making sure that no morphism is left with no source or no target and that no path is left without its composition.

Exercise 73 (NOW!)**.** Find an example of a category $\mathbf{C}$ and a category $\mathbf{C}'$ that satisfy the first three conditions but not the fourth.

See solution.

Definition 74 (Full and wide)**.** A subcategory $\mathbf{C}'$ of $\mathbf{C}$ is called **full** if for any objects $A, B \in \mathbf{C}'_0$, $\mathrm{Hom}_{\mathbf{C}'}(A,B) = \mathrm{Hom}_{\mathbf{C}}(A,B)$. It is called **wide** if $\mathbf{C}'_0 = \mathbf{C}_0$.

Examples 75 (Subcategories of **Set**)**.** One can view most of the theory studied in the first year of a typical mathematics curriculum through the lens of category theory as witnessed by the following list.

1. Since the composition of injective functions is again injective, the restriction of morphisms in **Set** to injective functions yields a wide subcategory of **Set**, denoted **SetInj**. Unsurprisingly, **SetSurj** can be constructed similarly.

2. Removing all infinite sets from **Set** yields the full subcategory of finite sets denoted **FinSet**.

3.

Example 76 (Concrete categories)**.** This second list of examples contains so-called concrete categories, which, informally, are categories of sets with extra structure.[48]

[48] A formal definition is given in

1. The category $\mathbf{Set}_*$ is the category of **pointed** sets. Its objects are sets with a distinguished element and its morphisms are functions that map distinguished elements to distinguished elements. More formally, $\mathbf{Set}_{*0}$ is the collection of pairs (X, x) where X is a set and $x \in X$. Moreover, for any two pointed sets (X, x) and (Y, y),

$$\mathrm{Hom}_{\mathbf{Set}_*}((X,x),(Y,y)) = \{f : X \to Y \mid f(x) = y\}.$$

   The identity morphisms and composition are defined as in **Set**, so the axioms of a category clearly hold after checking that if $f(X,x) \to (Y,y)$ satisfies $f(x) = y$ and $g : (Y,y) \to (Z,z)$ satisfies $g(y) = z$, then $(g \circ f)(x) = z$.

2. The category **Mon** is the category of monoids and their homomorphisms, let us detail the structure of **Mon**.[49] The objects are monoids, so $\mathbf{Mon}_0$ is the collection of all monoids, and the morphisms are monoid homomorphisms, so for any $M, N \in \mathbf{Mon}_0$, $\mathrm{Hom}_{\mathbf{Mon}}(M,N)$ is the set of homomorphisms from M to N. The composition in **Mon** is given by the composition of homomorphisms, we know it is well-defined because the composition of two homomorphisms is a homomorphism. Also, the composition is associative and the identity functions are homomorphisms, so we can define $u_{\mathbf{Mon}}(M) = \mathrm{id}_M$.

[49] These details are essentially the same for the categories in the rest of Example 76.

3. Similarly, the category of groups (resp. rings or fields) where the morphisms are group (resp. ring or field) homomorphisms is denoted **Grp** (resp. **Ring** or **Field**). The category of abelian groups (resp. commutative monoids or rings) is a full subcategories of **Grp** (resp. **Mon** or **Ring**) denoted **Ab** (resp. **CMon** or **CRing**).

4. Let k be a fixed field, the category of vector spaces over k where the morphisms are linear maps is denoted $\mathbf{Vect}_k$. The full subcategory of $\mathbf{Vect}_k$ consisting only of finite dimensional vector spaces is denoted $\mathbf{FDVect}_k$.

5. The category of partially ordered sets where morphisms are order-preserving functions is denoted **Poset**.

6. The category of topological spaces where morphisms are continuous functions is denoted **Top**.

Our last example is a large category which is not a subcategory of **Set**.

Example 77 (Rel). The category of sets and relations, denoted **Rel**, has as objects the collection of all sets and for any sets X and Y, $\mathrm{Hom}_{\mathbf{Rel}}(X,Y)$ is the set of relations between X and Y, that is, the powerset of of $X \times Y$. The composition of two relations $R \subseteq X \times Y$ and $S \subseteq Y \times Z$ is defined by

$$S \circ R = R; S := \{(x,z) \in X \times Z \mid \exists y \in Y, (x,y) \in R, (y,z) \in S\} \subseteq X \times Z.$$

One can check that this composition is associative and that, for any set X, the **diagonal relation** $\Delta_X = \{(x,x) : x \in X\} \subseteq X \times X$ is the identity with respect to this composition.

Remark 78. You can view **Set** as the subcategory of **Rel** where you only take the relations $R \subseteq X \times Y$ satisfying for any $x \in X$,

$$|\{y \in Y \mid (x,y) \in R\}| = 1.$$

Functors

The list above is far from exhaustive; there are many more mathematical objects that can fit in a category and this is a main reason for studying this subject. Indeed, categories encapsulate a natural structure that accurately represents the heart of several mathematical theories from a global and abstract perspective. Still, a category is almost never studied on its own since the abstraction it provides can make the properties of its objects more obscure. For instance, stating and proving Lagrange's theorem in the framework of **Grp** is quite more involved than in the classical way. Nevertheless, we will get to see in subsequent chapters that some surprising links can arise between seemingly unrelated subjects through the study of how different categories relate. The central tool for exhibiting these relations is a functor.

As we will show, a functor is a morphism of categories, thus, to motivate the definition, we can look at other morphisms we have encountered. A clear similarity between categories like **Set**, **Grp**, **Ring** or **Top** is that all the objects have some sort of structure that the morphisms preserve. Hence, we want to define a morphism that preserves the structure of a category, the latter being given by the source and target maps, the composition and the identities.

Definition 79 (Functor)**.** Let $\mathbf{C}$ and $\mathbf{D}$ be categories, a **functor** $F : \mathbf{C} \rightsquigarrow \mathbf{D}$ is a pair of maps $F_0 : \mathbf{C}_0 \to \mathbf{D}_0$ and $F_1 : \mathbf{C}_1 \to \mathbf{D}_1$ such that diagrams (9), (10) and (11) commute.[50]

$$\begin{tikzcd} \mathbf{C}_0 \arrow[d, "F_0"'] & \mathbf{C}_1 \arrow[l, "s"'] \arrow[r, "t"] \arrow[d, "F_1"] & \mathbf{C}_0 \arrow[d, "F_0"] \\ \mathbf{D}_0 & \mathbf{D}_1 \arrow[l, "s"] \arrow[r, "t"'] & \mathbf{D}_0 \end{tikzcd} \tag{9}$$

[50] F_2 is induced by the definition of F_1 with $(f,g) \mapsto (F_1(f), F_1(g))$.

$$\begin{tikzcd} \mathbf{C}_2 \arrow[r, "F_2"] \arrow[d, "\circ_{\mathbf{C}}"'] & \mathbf{D}_2 \arrow[d, "\circ_{\mathbf{D}}"] \\ \mathbf{C}_1 \arrow[r, "F_1"'] & \mathbf{D}_1 \end{tikzcd} \qquad (10) \qquad \begin{tikzcd} \mathbf{C}_0 \arrow[r, "F_0"] \arrow[d, "u_{\mathbf{C}}"'] & \mathbf{D}_0 \arrow[d, "u_{\mathbf{D}}"] \\ \mathbf{C}_1 \arrow[r, "F_1"'] & \mathbf{D}_1 \end{tikzcd} \qquad (11)$$

Remark 80 (Digesting diagrams). Commutative diagrams will be heavily employed to make clearer and more compact arguments.[51] However, it is an acquired skill to quickly grasp their meaning and make effective use of their advantages. Unpacking the above definition will help to understand it as well as getting better with manipulating diagrams.

A functor $F : \mathbf{C} \rightsquigarrow \mathbf{D}$[52] must satisfy the following properties.

i. For any $A, B \in \mathbf{C}_0$ and $f \in \mathrm{Hom}_{\mathbf{C}}(A, B)$, $F(f) \in \mathrm{Hom}_{\mathbf{D}}(F(A), F(B))$. This is equivalent to the commutativity of (9) which says $F_0(s(f)) = s(F_1(f))$ and $F_0(t(f)) = t(F_1(f))$.

ii. If $f, g \in \mathbf{C}_1$ are composable, then $F(f \circ_{\mathbf{C}} g) = F(f) \circ_{\mathbf{D}} F(g)$ by commutativity of (10).

iii. If $A \in \mathbf{C}_0$, then $u_{\mathbf{D}}(F(A)) = F(u_{\mathbf{C}}(A))$ (alternatively, $\mathrm{id}_{F(A)} = F(\mathrm{id}_A)$) by commutativity of (11).

The subscript on F is often omitted, as is common in the literature, because it is always clear whether F is applied to an object or a morphism.

Examples 81 (Boring examples)**.** As usual, a few trivial constructions arise.

1. For any category $\mathbf{C}$, the **identity functor** $\mathrm{id}_{\mathbf{C}} : \mathbf{C} \rightsquigarrow \mathbf{C}$ is defined by letting $(\mathrm{id}_{\mathbf{C}})_0$ and $(\mathrm{id}_{\mathbf{C}})_1$ be identity maps on $\mathbf{C}_0$ and $\mathbf{C}_1$ respectively.

2. Let $\mathbf{C}$ be a category and $\mathbf{C}'$ a subcategory of $\mathbf{C}$, the **inclusion functor** $\mathcal{I} : \mathbf{C}' \rightsquigarrow \mathbf{C}$ is defined by letting $\mathcal{I}_0$ be the inclusion map $\mathbf{C}'_0 \hookrightarrow \mathbf{C}_0$ and $\mathcal{I}_1$ be the inclusion map $\mathbf{C}'_1 \hookrightarrow \mathbf{C}_1$.

3. Let $\mathbf{C}$ and $\mathbf{D}$ be categories and X be an object in $\mathbf{D}$, the **constant functor** $X : \mathbf{C} \rightsquigarrow \mathbf{D}$ is defined by letting $X_0(A) = X$ for any $A \in \mathbf{C}_0$ and $X_1(f) = \mathrm{id}_X$ for any $f \in \mathbf{C}_1$.

Examples 82 (Less boring)**.** Functors with the domain being one of $\mathbf{1}$, $\mathbf{2}$ or $\mathbf{2} \times \mathbf{2}$ (cf. Example 62) are a bit less boring. Let the codomain be a category $\mathbf{C}$ and let us analyze these functors.

- Let $F : \mathbf{1} \rightsquigarrow \mathbf{C}$, F_0 assigns to the single object $\bullet \in \mathbf{1}_0$ an object $F(\bullet) \in \mathbf{C}_0$. Then, by commutativity of (11), F_1 is completely determined by $\mathrm{id}_\bullet \mapsto \mathrm{id}_{F(\bullet)}$. We conclude that functors of this type are in correspondence with objects of $\mathbf{C}$.

- Let $F : \mathbf{2} \rightsquigarrow \mathbf{C}$, F_0 assigns to A and B, two objects $FA, FB \in \mathbf{C}_0$ and F_1's action on identities is fixed. Still, there is one choice to make for $F_1(f)$ which must be a morphism in $\mathrm{Hom}_{\mathbf{C}}(FA, FB)$. Therefore, F sums up to a choice of two objects in $\mathbf{C}$ and a morphism between them. In other words, functors of this type are in correspondence with morphisms in $\mathbf{C}$.[53]

[51] This is especially true when using a blackboard or pen and paper because it makes it easier to point at things. Sadly, I cannot point at things on this pdf you are reading.

[52] The $\rightsquigarrow$ (`\rightsquigarrow`) notation for functors is not that common, they are usually denoted with plain arrows because they are morphisms. Still, we feel it is useful to have a special treatment for functors until you get accustomed to them. The squiggly arrow notation is sometimes used for Kleisli morphisms which we cover in.

[53] After picking a morphism, the source and target are determined.

- Similarly (we leave the details as an exercise), functors of type $F : \mathbf{2} \times \mathbf{2} \rightsquigarrow \mathbf{C}$ are in correspondence with commutative squares inside the category $\mathbf{C}$.[54]

[54] i.e.: pairs of pairs of composable morphisms $((f,g),(f',g')) \in \mathbf{C}_2 \times \mathbf{C}_2$ satisfying $f \circ g = f' \circ g'$.

Remark 83 (Functoriality). We will use the term **functorial** as an adjective to qualify transformations that behave like functors and **functoriality** to refer to the property of behaving like a functor.

Throughout the rest of this book, the goal will essentially be to grow our list of categories and functors with more and more interesting examples and perhaps exploit their properties wisely. Before pursuing this objective, we give important definitions analogous to injectivity and surjectivity of functions.

Definition 84 (Full and faithful)**.** Let $F : \mathbf{C} \rightsquigarrow \mathbf{D}$ be a functor. For $A, B \in \mathbf{C}_0$, denote the restriction of F_1 to $\mathrm{Hom}_{\mathbf{C}}(A,B)$ with

$$F_{A,B} : \mathrm{Hom}_{\mathbf{C}}(A,B) \to \mathrm{Hom}_{\mathbf{D}}(F(A),F(B)).$$

- If $F_{A,B}$ is injective for any $A, B \in \mathbf{C}_0$, then F is **faithful**.
- If $F_{A,B}$ is surjective for any $A, B \in \mathbf{C}_0$, then F is **full**.
- If $F_{A,B}$ is bijective for any $A, B \in \mathbf{C}_0$, then F is **fully faithful**.

Remark 85*.* These notions never mention the action of F on objects, so they cannot lead to a notion of isomorphism of categories.

Examples 86. For the following examples, we leave to the reader the easy and irrelevant to category theory task of proving they are actually functors.

1. The **powerset functor** $\mathcal{P} : \mathbf{Set} \rightsquigarrow \mathbf{Set}$ sends a set X to its powerset $\mathcal{P}(X)$ and a function $f : X \to Y$ to the image map $\mathcal{P}(f) : \mathcal{P}(X) \to \mathcal{P}(Y)$, the latter sends a subset $S \subseteq X$ to
$$\mathcal{P}(f)(S) = f(S) := \{f(s) \mid s \in S\} \subseteq Y.$$
The powerset functor is faithful because the same image map cannot arise from two different functions[55], it is not full because lots of functions $\mathcal{P}(X) \to \mathcal{P}(Y)$ are not image maps. One can also argue by cardinality because (when $|X|, |Y| \geq 2$)
$$|\mathrm{Hom}_{\mathbf{Set}}(X,Y)| = |Y|^{|X|} < |\mathcal{P}(Y)|^{|\mathcal{P}(X)|} = |\mathrm{Hom}_{\mathbf{Set}}(\mathcal{P}(X),\mathcal{P}(Y))|.$$

[55] Indeed, if $f(x) \neq g(x)$, then $f(\{x\}) \neq g(\{x\})$.

2. While in general the inclusion functor of a subcategory is not interesting,[56] there are some distinguished cases. For instance, when considering subcategories of **Set** such as those mentioned in Example 75, the inclusion functor gets a fancier denomination, namely, the **forgetful** functor (denoted U for underlying). This is because, morally, the functor is forgetting about the inner structure of the objects and morphisms and it outputs the underlying sets and functions. The forgetful functor $U : \mathbf{Grp} \rightsquigarrow \mathbf{Set}$ sends a group $(G, \cdot, 1_G)$ to its underlying set G, *forgetting about the operation and identity*. It sends a group homomorphism $f : G \to H$ to the underlying function, *forgetting about the homomorphism properties*.

[56] Still, one can check the inclusion functor is always faithful and it is full if and only if the subcategory is full.

3. It is also sometimes useful to consider *intermediate* forgetful functors. For example, $U : \mathbf{Ring} \rightsquigarrow \mathbf{Ab}$ sends a ring $(R, +, \cdot, 1_R, 0_R)$ to the abelian group $(R, +, 0_R)$, *forgetting about multiplication and* 1_R. It sends a ring homomorphism $f : R \to S$ to the same underlying function seen as a group homomorphism.[57]

4. In some cases, there is a canonical way to go in the opposite direction of the forgetful functor, that is, the free functor. For $\mathbf{Grp}$, the free functor $F : \mathbf{Set} \rightsquigarrow \mathbf{Grp}$ sends a set to the free group generated by this set and a function $f : X \to Y$ to the unique group homomorphism $F(X) \to F(Y)$ that restricts to f on the set of generators.

   Later in the book, when covering adjunctions, we will study a strong relation between the forgetful functor U and the free functor F that will generalize to other mathematical structures.

5. Let $(X, \leq)$ and $(Y, \sqsubseteq)$ be posets, and $F : X \rightsquigarrow Y$ be a functor between their posetal categories. For any $a, b \in X$, if $a \leq b$, then $\mathrm{Hom}_X(a, b)$ contains a single element, thus $\mathrm{Hom}_Y(F(a), F(b))$ must contain a morphism as well,[58] or equivalently $F(a) \sqsubseteq F(b)$. This shows that F_0 is an order-preserving function on the posets.

   Conversely, any order-preserving function between X and Y will correspond to a unique functor since there is only one morphism in all the hom–sets.[59]

6. Let G and H be groups and $\mathbf{B}(G)$ and $\mathbf{B}(H)$ be their respective deloopings, then the functors $F : \mathbf{B}(G) \rightsquigarrow \mathbf{B}(H)$ are exactly the group homomorphisms from G to H.[60]

7. For any group G, the functors $F : \mathbf{B}(G) \rightsquigarrow \mathbf{Set}$ are in correspondence with left actions of G. Indeed, if $S = F(*)$, then

$$F_1 : G = \mathrm{Hom}_{\mathbf{B}(G)}(*, *) \to \mathrm{Hom}_{\mathbf{Set}}(S, S)$$

   is such that $F(gh) = F(g) \circ F(h)$ for any $g, h \in G$ and $F(1_G) = \mathrm{id}_S$.[61] Moreover, since for any $g \in G$,

$$F(g^{-1}) \circ F(g) = F(g^{-1}g) = F(1_G) = \mathrm{id}_S = F(1_G) = F(gg^{-1}) = F(g) \circ F(g^{-1}),$$

   the function $F(g)$ is a bijection (its inverse if $F(g^{-1})$) and we conclude F_1 is the permutation representation of the group action defined by $g \star s = F(g)(s)$ for all $g \in G$ and $s \in S$.

8. In the previous example, replacing $\mathbf{Set}$ with $\mathbf{Vect}_k$, one obtains k-linear representations of G instead of actions of G.[62]

Remark 87. From this long (and yet hardly exhaustive) list, one might get the feeling that every important mathematical transformation is a functor. This is not the case so we wanted to show where functoriality can fail and hopefully give you a bit of intuition about why they fail.

[57] It can do that because part of the requirements for ring homomorphisms is to preserve the underlying additive group structure.

[58] The image of the element in $\mathrm{Hom}_X(a, b)$ under F.

[59] Given $f : (X, \leq) \to (Y, \sqsubseteq)$ order-preserving, the corresponding functor between the posetal categories of X and Y acts like f of the objects and sends a morphism $a \to b$ to the unique morphism $f(a) \to f(b)$ which exists because $a \leq b \implies f(a) \sqsubseteq f(b)$.

[60] Similarly for the deloopings of monoids.

[61] This is because gh is the composite of g and h in $\mathbf{B}(G)$ and 1_G is the identity morphism in $\mathbf{B}(G)$.

[62] You might know about linear representations, we just mention them in passing.

For instance, let us define $F : \mathbf{FDVect}_k \rightsquigarrow \mathbf{Set}$ which assigns to any vector space over k a choice of basis. There is no non-trivializing way to define an action of F on linear maps which make F into a functor. Another example of a non-functor is given by the center of a group in **Grp**: a morphism of group $H \to G$ does not necessarily send the center of H in the center of G (take for instance $S_2 \hookrightarrow S_3$).

In this chapter, we introduced a novel structure, namely categories, that functors preserve.[63] Since we also introduced several categories where objects had some structure that morphisms preserve, it is reasonable to wonder whether categories are also part of a category. In fact, the only missing ingredient is the composition of functors (we already know what the source and target of a functor is and every category has an identity functor). After proving the following proposition, we end up with the category **Cat** where objects are categories and morphisms are functors. In order to avoid paradoxes of the Russel kind, it is essential to restrict **Cat** to contain only small categories.

[63] In fact, we defined functors so that they preserve the structure of categories.

Proposition 88. *Let $F : \mathbf{C} \rightsquigarrow \mathbf{D}$ and $G : \mathbf{D} \rightsquigarrow \mathbf{E}$ be functors and $G \circ F : \mathbf{C} \rightsquigarrow \mathbf{E}$ be their* ***composition*** *defined by $G_0 \circ F_0$ on objects and $G_1 \circ F_1$ on morphisms. Then, $G \circ F$ is a functor.*

Proof. One could proceed with a really hands-on proof and show that $G \circ F$ satisfies the three necessary properties in a straightforward manner. While this should not be too hard, the proof will end up involving objects, morphisms and the composition from all three different categories. This can easily lead to confusion or worse, boredom!

Instead, we will use the diagrams we introduced in the first definition of a functor. From the functoriality of F and G, we get two sets of three diagrams and combining them yields the diagrams for $G \circ F$.[64]

[64]

$$\begin{tikzcd}
\mathbf{C}_0 \arrow[d, "F_0"'] & \mathbf{C}_1 \arrow[l, "s"'] \arrow[r, "t"] \arrow[d, "F_1"] & \mathbf{C}_0 \arrow[d, "F_0"] \\
\mathbf{D}_0 \arrow[d, "G_0"'] & \mathbf{D}_1 \arrow[l, "s"'] \arrow[r, "t"] \arrow[d, "G_1"] & \mathbf{D}_0 \arrow[d, "G_0"] \\
\mathbf{E}_0 & \mathbf{E}_1 \arrow[l, "s"'] \arrow[r, "t"] & \mathbf{E}_0
\end{tikzcd} \tag{12}$$

$$\begin{tikzcd}
\mathbf{C}_2 \arrow[r, "F_2"] \arrow[d, "\circ_{\mathbf{C}}"'] & \mathbf{D}_2 \arrow[r, "G_2"] \arrow[d, "\circ_{\mathbf{D}}"] & \mathbf{E}_2 \arrow[d, "\circ_{\mathbf{E}}"] \\
\mathbf{C}_1 \arrow[r, "F_1"'] & \mathbf{D}_1 \arrow[r, "G_1"'] & \mathbf{E}_1
\end{tikzcd} \tag{13} \qquad
\begin{tikzcd}
\mathbf{C}_0 \arrow[r, "F_0"] \arrow[d, "u_{\mathbf{C}}"'] & \mathbf{D}_0 \arrow[r, "G_0"] \arrow[d, "u_{\mathbf{D}}"] & \mathbf{E}_0 \arrow[d, "u_{\mathbf{E}}"] \\
\mathbf{C}_1 \arrow[r, "F_1"'] & \mathbf{D}_1 \arrow[r, "G_1"'] & \mathbf{E}_1
\end{tikzcd} \tag{14}$$

To finish the proof, you need to convince yourself that combining commutative diagrams in this way yields commutative diagrams. We proceed with a proof by example. Take diagram (14), we know the left and right square are commutative because F and G are functors. To show that the rectangle also commutes, we need to show the top path and bottom path from $\mathbf{C}_0$ to $\mathbf{E}_1$ compose to the same function. Here is the derivation:[65]

[65] In this case, both the diagram and the derivation are fairly simple. This will not stay true in the rest of the book, but the complexity of diagrams will grow way slower than the complexity of derivations and we will mostly omit the latter for this reason.

$$\begin{aligned} G_1 \circ F_1 \circ u_{\mathbf{C}} &= G_1 \circ u_{\mathbf{D}} \circ F_0 && \text{left square commutes} \\ &= u_{\mathbf{E}} \circ G_0 \circ F_0 && \text{right square commutes} \end{aligned}$$

□

Since functors are also a new structure, one might expect that there are transformations between functors that preserve it. It is indeed the case, they are called natural transformations and they are the main subject of Chapter **??**. Moreover, although we will not cover it, there is a whole tower of abstraction that one could build in this way and it is the subject of study of higher category theory.

Products

There is one last thing we want to mention to end this chapter. We have defined two new mathematical objects, categories and functors and presented several examples of each. By defining products, we give you access to an unlimited amount of new categories and functors you can construct from known ones.[66]

[66] This is akin to product of groups, diret sums of vector spaces, etc.

Definition 89 (Product categories)**.** Let $\mathbf{C}$ and $\mathbf{D}$ be two categories, the **product** of $\mathbf{C}$ and $\mathbf{D}$, denoted $\mathbf{C} \times \mathbf{D}$, is the category whose objects are pairs of objects in $\mathbf{C}_0 \times \mathbf{D}_0$ and for any two pairs $(X, Y), (X', Y') \in (\mathbf{C} \times \mathbf{D})_0$,[67]

$$\mathrm{Hom}_{\mathbf{C}\times\mathbf{D}}((X,Y),(X',Y')) := \mathrm{Hom}_{\mathbf{C}}(X,X') \times \mathrm{Hom}_{\mathbf{D}}(Y,Y').$$

[67] Explicitly, a morphism $(X, Y) \to (X', Y')$ is a pair of morphisms $X \to X'$ and $Y \to Y'$.

The identity morphisms and the composition are defined componentwise, i.e.: $\mathrm{id}_{(X,Y)} = (\mathrm{id}_X, \mathrm{id}_Y)$ and if $(f, f') \in \mathbf{C}_2$ and $(g, g') \in \mathbf{D}_2$ are two composable pairs, then $(f, g) \circ (f', g') = (f \circ f', g \circ g')$.[68]

[68] We leave you to check that this defines the composition of all morphisms in $\mathbf{C} \times \mathbf{D}$. Namely, if (f, g) and (f', g') are composable, then f, f' and g, g' are composable.
See solution.

Exercise 90. Show that the assignment $\Delta_{\mathbf{C}} : \mathbf{C} \rightsquigarrow \mathbf{C} \times \mathbf{C} = X \mapsto (X, X)$ is functorial, i.e.: give its action on morphisms and show it satisfies the relevant axioms. We call $\Delta_{\mathbf{C}}$ the **diagonal functor**.

Definition 91 (Product functor)**.** Let $F : \mathbf{C} \rightsquigarrow \mathbf{C}'$ and $G : \mathbf{D} \rightsquigarrow \mathbf{D}'$ be two functors, the **product** of F and G, denoted $F \times G$, is defined componentwise on objects and morphisms, i.e.: for any $(X, Y) \in (\mathbf{C} \times \mathbf{D})_0$ and $(f, g) \in (\mathbf{C} \times \mathbf{D})_1$

$$(F \times G)(X, Y) = (FX, GY) \text{ and } (F \times G)(f, g) = (Ff, Gg).$$

Let us check this defines a functor.

i. By definition of $\mathbf{C}' \times \mathbf{D}'$, (Ff, Gg) is a morphism from (FX, GY) to (FX', GY').

ii. For $(f, f') \in \mathbf{C}_2$ and $(g, g') \in \mathbf{D}_2$, we have

$$\begin{aligned} (F \times G)((f,g) \circ (f',g')) &= (F \times G)(f \circ f', g \circ g') \\ &= (F(f \circ f'), G(g \circ g')) \\ &= (Ff \circ Ff', Gg \circ Gg') \\ &= (Ff, Gg) \circ (Ff', Gg') \\ &= (F \times G)(f,g) \circ (F \times G)(f',g'). \end{aligned}$$

iii. Since F and G preserve identity morphisms, we have

$$(F \times G)(\mathrm{id}_{(X,Y)}) = (F \times G)(\mathrm{id}_X, \mathrm{id}_Y) = (F\mathrm{id}_X, G\mathrm{id}_Y) = (\mathrm{id}_{FX}, \mathrm{id}_{GY}) = \mathrm{id}_{(FX,GY)}.$$

Exercise 92 (NOW!). Let $F : \mathbf{C} \times \mathbf{C}' \to \mathbf{D}$ be a functor. For $X \in \mathbf{C}_0$, we define $F(X,-) : \mathbf{C}' \rightsquigarrow \mathbf{D}$ on objects by $Y \mapsto F(X,Y)$ and on morphisms by $g \mapsto F(\mathrm{id}_X, g)$. Show that $F(X,-)$ is a functor. Define $F(-,Y)$ similarly.

See solution.

We will often use $-$ as a **placeholder** for an input so that the latter remains nameless. For instance, $f(-,-)$ means f takes two inputs. The type of the inputs and outputs will be made clear in the context.

Exercise 93. Let $F : \mathbf{C} \times \mathbf{C}' \to \mathbf{D}$ be an action defined on objects and morphisms which is not necessarily a functor. Show that if for any $X \in \mathbf{C}_0$ and $Y \in \mathbf{C}'_0$, $F(X,-)$ and $F(-,Y)$ as defined above are functors, then F is a functor. In other words, the functoriality of F can be proven componentwise.

See solution.

In the next chapters, we will present other interesting constructions, but we will stop here for now.

Duality

The concept of duality is ubiquitous throughout mathematics. It can relate two perspectives of the same object as for dual vector spaces, two complementary problems such as a maximization and a minimization linear program and even two seemingly unrelated fields like topology and logic (cf. Stone dualities). While this vague principle of duality is the foundation of many groundbreaking results, the duality in question here is categorical duality and it is a bit more precise.

Informally, there is nothing more to say than "Take all the diagrams in a definition/theorem, reverse the arrows and reap the benefits of the dual concept/result." The more formal version will follow after we first exhibit the principle in action.

Recall that, intuitively, a functor is a structure preserving transformation between categories. A simple example we have seen was functors between posets that were order-preserving functions. However, as a consequence, one might conclude that order-reversing functions impair the structure of a poset, which feels arbitrary. The same happens between deloopings of groups because anti-homomorphisms[69] cannot arise as functors between such categories.

[69] An **anti-homomorphism** $f : G \to H$ is a function satisfying $f(gg') = f(g')f(g)$ and $f(1_G) = f(1_H)$.

There are two options to remedy this discrepancy between intuition and formalism; both have duality as a guiding principle.

Contravariant Functors

By modifying Definition 79 to require that $F(f)$ goes in the opposite direction, we obtain a contravariant functor. Incidentally, what we defined as a functor then is also called a **covariant** functor.

Definition 94 (Contravariant functor)**.** Let $\mathbf{C}$ and $\mathbf{D}$ be categories, a **contravariant functor** $F : \mathbf{C} \rightsquigarrow \mathbf{D}$ is a pair of maps $F_0 : \mathbf{C}_0 \to \mathbf{D}_0$ and $F_1 : \mathbf{C}_1 \to \mathbf{D}_1$ making diagrams (15), (16) and (17) commute.[70]

[70] Where F_2' is now induced by the definition of F_1 with $(f, g) \mapsto (F_1(g), F_1(f))$.

$$\begin{tikzcd}
\mathbf{C}_0 \arrow[d, "F_0"'] & \mathbf{C}_1 \arrow[l, "s"'] \arrow[r, "t"] \arrow[d, "F_1"] & \mathbf{C}_0 \arrow[d, "F_0"] \\
\mathbf{D}_0 & \mathbf{D}_1 \arrow[l, "t"] \arrow[r, "s"'] & \mathbf{D}_0
\end{tikzcd} \tag{15}$$

$$\begin{tikzcd} \mathbf{C}_2 \arrow[r, "F_2'"] \arrow[d, "\circ_{\mathbf{C}}"'] & \mathbf{D}_2 \arrow[d, "\circ_{\mathbf{D}}"] \\ \mathbf{C}_1 \arrow[r, "F_1"'] & \mathbf{D}_1 \end{tikzcd} \qquad (16) \qquad \begin{tikzcd} \mathbf{C}_0 \arrow[r, "F_0"] \arrow[d, "u_{\mathbf{C}}"'] & \mathbf{D}_0 \arrow[d, "u_{\mathbf{D}}"] \\ \mathbf{C}_1 \arrow[r, "F_1"'] & \mathbf{D}_1 \end{tikzcd} \qquad (17)$$

In words, F must satisfy the following properties.

i. For any $A, B \in \mathbf{C}_0$, if $f \in \mathrm{Hom}_{\mathbf{C}}(A, B)$ then $F(f) \in \mathrm{Hom}_{\mathbf{D}}(F(B), F(A))$.

ii. If $f, g \in \mathbf{C}_1$ are composable, then $F(f \circ g) = F(g) \circ F(f)$.

iii. If $A \in \mathbf{C}_0$, then $u_{\mathbf{D}}(F(A)) = F(u_{\mathbf{C}}(A))$.

Examples 95. Just like their covariant counterparts, contravariant functors are quite numerous. Here are a few simple ones, we leave you to check that they satisfy the diagrams above.

1. Contravariant functors $F : (X, \leq) \rightsquigarrow (Y, \sqsubseteq)$ correspond to order-reversing functions between the posets X and Y while contravariant functors $F : \mathbf{B}(G) \rightsquigarrow \mathbf{B}(H)$ correspond to anti-homomorphisms between the groups G and H.

2. The contravariant powerset functor $\widehat{\mathcal{P}} : \mathbf{Set} \rightsquigarrow \mathbf{Set}$ sends a set X to its powerset $\mathcal{P}(X)$ and a function $f : X \to Y$ to the pre-image map $\widehat{\mathcal{P}}(f) : \mathcal{P}(Y) \to \mathcal{P}(X)$, the latter sends a subset $S \subseteq Y$ to

$$\widehat{\mathcal{P}}(f)(S) = f^{-1}(S) := \{x \in X \mid f(x) \in S\} \subseteq X.$$

Next, there is a couple of functors that are key to understand the philosophy put forward by category theory.[71]

[71] We will talk more about it when covering the Yoneda lemma in Chapter ??.

Example 96 (Hom functors)**.** Let $\mathbf{C}$ be a locally small category and $A \in \mathbf{C}_0$ one of its objects.[72] We define the covariant and contravariant Hom functors from $\mathbf{C}$ to $\mathbf{Set}$.

[72] We need local smallness to have functors into $\mathbf{Set}$.

1. The covariant functor $\mathrm{Hom}_{\mathbf{C}}(A, -) : \mathbf{C} \rightsquigarrow \mathbf{Set}$ sends an object $B \in \mathbf{C}_0$ to the hom–set $\mathrm{Hom}_{\mathbf{C}}(A, B)$ and a morphism $f : B \to B'$ to the function

$$\mathrm{Hom}_{\mathbf{C}}(A, f) : \mathrm{Hom}_{\mathbf{C}}(A, B) \to \mathrm{Hom}_{\mathbf{C}}(A, B') = g \mapsto f \circ g.$$

This function is called **post-composition by** f and is denoted $f \circ (-)$ or f_*. Let us show $\mathrm{Hom}_{\mathbf{C}}(A, -)$ is a covariant functor.

i. For any $f \in \mathbf{C}_1$, it is clear from the definitions that

$$\mathrm{Hom}_{\mathbf{C}}(A, s(f)) = s(\mathrm{Hom}_{\mathbf{C}}(A, f)) \text{ and } \mathrm{Hom}_{\mathbf{C}}(A, t(f)) = t(\mathrm{Hom}_{\mathbf{C}}(A, f)).$$

ii. For any $(f_1, f_2) \in \mathbf{C}_2$, we claim that

$$\mathrm{Hom}_{\mathbf{C}}(A, f_1 \circ f_2) = \mathrm{Hom}_{\mathbf{C}}(A, f_1) \circ \mathrm{Hom}_{\mathbf{C}}(A, f_2).$$

In the L.H.S., an element $g \in \mathrm{Hom}_{\mathbf{C}}(A, s(f_1 \circ f_2))$ is mapped to $(f_1 \circ f_2) \circ g$ and in the R.H.S., an element $g \in \mathrm{Hom}_{\mathbf{C}}(A, s(f_2))$ is mapped to $f_1 \circ (f_2 \circ g)$. Since $s(f_1 \circ f_2) = s(f_2)$ and composition is associative, we conclude that the two maps are the same.

iii. For any $B \in \mathbf{C}_0$, the post-composition by $u_{\mathbf{C}}(B)$ is defined to be the identity,[73] hence (11) also commutes.

[73] Namely, for any $f : A \to B$, $u_{\mathbf{C}}(B) \circ f = f$.

2. The contravariant functor $\mathrm{Hom}_{\mathbf{C}}(-, A) : \mathbf{C} \rightsquigarrow \mathbf{Set}$ sends an object $B \in \mathbf{C}_0$ to the hom–set $\mathrm{Hom}_{\mathbf{C}}(B, A)$ and a morphism $f : B \to B'$ to the function

$$\mathrm{Hom}_{\mathbf{C}}(f, A) : \mathrm{Hom}_{\mathbf{C}}(B', A) \to \mathrm{Hom}_{\mathbf{C}}(B, A) = g \mapsto g \circ f.$$

This function is called **pre-composition by** f and is denoted $(-) \circ f$ or f^*. Let us show $\mathrm{Hom}_{\mathbf{C}}(-, A)$ is a contravariant functor.

i. For any $f \in \mathbf{C}_1$, it is clear from the definitions that

$$\mathrm{Hom}_{\mathbf{C}}(s(f), A) = t(\mathrm{Hom}_{\mathbf{C}}(f, A)) \text{ and } \mathrm{Hom}_{\mathbf{C}}(t(f), A) = s(\mathrm{Hom}_{\mathbf{C}}(f, A)).$$

ii. For any $(f_1, f_2) \in \mathbf{C}_2$, we claim that

$$\mathrm{Hom}_{\mathbf{C}}(f_1 \circ f_2, A) = \mathrm{Hom}_{\mathbf{C}}(f_2, A) \circ \mathrm{Hom}_{\mathbf{C}}(f_1, A).$$

In the L.H.S., an element $g \in \mathrm{Hom}_{\mathbf{C}}(t(f_1 \circ f_2), A)$ is mapped to $g \circ (f_1 \circ f_2)$ and in the R.H.S., an element $g \in \mathrm{Hom}_{\mathbf{C}}(t(f_1), A)$ is mapped to $(g \circ f_1) \circ f_2$. Since $t(f_1 \circ f_2) = t(f_1)$ and composition is associative, we conclude that the two maps are the same.

iii. For any $B \in \mathbf{C}_0$, pre-composition by $u_{\mathbf{C}}(B)$ is defined to be the identity,[74] hence (17) also commutes.

[74] Namely, for any $f : B \to A$, $f \circ u_{\mathbf{C}}(B) = f$.

We will not dwell too long on contravariant functors as we will see right away how they can be avoided.

Opposite Category

Another way to deal with order-reversing maps $(X, \leq) \to (Y, \subseteq)$ is to consider the reverse order on X and a covariant functor $(X, \geq) \rightsquigarrow (Y, \subseteq)$. This also works for anti-homomorphisms by constructing the opposite group G^{op} in which the operation is reversed, namely $g \cdot^{\mathrm{op}} h = hg$. The opposite category is a generalization of these constructions.

Definition 97 (Opposite category)**.** Let $\mathbf{C}$ be a category, we denote the **opposite category** with $\mathbf{C}^{\mathrm{op}}$ and define it by[75]

[75] Intuitively, we reverse the direction of all morphisms in $\mathbf{C}$ and reverse the order of composition as well.

$$\mathbf{C}_0^{\mathrm{op}} = \mathbf{C}_0,\ \mathbf{C}_1^{\mathrm{op}} = \mathbf{C}_1,\ s^{\mathrm{op}} = t,\ t^{\mathrm{op}} = s,\ u_{\mathbf{C}^{\mathrm{op}}} = u_{\mathbf{C}}$$

with the composition defined by $f^{\mathrm{op}} \circ^{\mathrm{op}} g^{\mathrm{op}} = (g \circ f)^{\mathrm{op}}$.[76] This naturally leads to the following contravariant functor $(-)^{\mathrm{op}}_{\mathbf{C}} : \mathbf{C} \rightsquigarrow \mathbf{C}^{\mathrm{op}}$ which sends an object A to A^{op} and a morphism f to f^{op}. It is called the **opposite functor**.

[76] Note that the $-^{\mathrm{op}}$ notation here is just used to distinguish elements in $\mathbf{C}$ and $\mathbf{C}^{\mathrm{op}}$ but the class of objects and morphisms are the same.

With this definition, one can see contravariant functors as covariant functors. Formally, let $F : \mathbf{C} \rightsquigarrow \mathbf{D}$ be a contravariant functor, we can view F as covariant functor from $\mathbf{C}^{\mathrm{op}}$ to $\mathbf{D}$ or from $\mathbf{C}$ to $\mathbf{D}^{\mathrm{op}}$ via the compositions $F \circ (-)^{\mathrm{op}}_{\mathbf{C}^{\mathrm{op}}}$ and $(-)^{\mathrm{op}}_{D} \circ F$ respectively.

In the rest of this book, we choose to work with functors of type $\mathbf{C}^{\mathrm{op}} \to \mathbf{D}$ instead of contravariant functors.[77]

[77] We still had to introduce the notion because you might see contravariant functors in the wild.

Examples 98. 1. As hinted at before, the category corresponding to $(X, \geq)$ is the opposite category of $(X, \leq)$ and $(\mathbf{B}(G))^{\mathrm{op}}$ is the category corresponding to the opposite group of G, i.e.: $\mathbf{B}(G)^{\mathrm{op}} = \mathbf{B}(G^{\mathrm{op}})$.

2. We have seen that functors $\mathbf{B}(G) \rightsquigarrow \mathbf{Set}$ correspond to left actions of a group G. You can check that functors $\mathbf{B}(G)^{\mathrm{op}} \rightsquigarrow \mathbf{Set}$ correspond to right actions of G.

3. The two Hom functors defined in Example 96 are now written

$$\mathrm{Hom}_{\mathbf{C}}(A, -) : \mathbf{C} \rightsquigarrow \mathbf{Set} \text{ and } \mathrm{Hom}_{\mathbf{C}}(-, A) : \mathbf{C}^{\mathrm{op}} \rightsquigarrow \mathbf{Set}.$$

By Exercise 93, they can be combined into a functor $\mathrm{Hom}_{\mathbf{C}}(-,-)$ acting on objects as $(A, B) \mapsto \mathrm{Hom}_{\mathbf{C}}(A, B)$ and on morphisms as $(f, g) \mapsto (g \circ - \circ f)$. This will be called the Hom **bifunctor**.

Exercise 99 (NOW!)**.** Let $F : \mathbf{C} \rightsquigarrow \mathbf{D}$ be a functor, show that F^{op} defined by $A^{\mathrm{op}} \mapsto (FA)^{\mathrm{op}}$ on objects and $f^{\mathrm{op}} \mapsto (Ff)^{\mathrm{op}}$ on morphisms is a functor.

Remark 100. It is sometimes useful to compose the Hom bifunctor with other functors as follows. Given two functors $F, G : \mathbf{C} \rightsquigarrow \mathbf{D}$, there is a functor $\mathrm{Hom}_{\mathbf{D}}(F-, G-) : \mathbf{C}^{\mathrm{op}} \times \mathbf{C} \rightsquigarrow \mathbf{D}$ action on objects by $(X, Y) \mapsto \mathrm{Hom}_{\mathbf{D}}(FX, GY)$ and on morphisms by $(f, g) \mapsto Gg \circ (-) \circ Ff$.

One can check functoriality by showing

$$\mathrm{Hom}_{\mathbf{D}}(F-, G-) = \mathrm{Hom}_{\mathbf{D}}(-,-) \circ (F^{\mathrm{op}} \times G).$$

Duality in Action

Let us start illustrating how duality can be useful with some simple definitions and results.

Definition 101 (Monomorphism)**.** Let $\mathbf{C}$ be a category, a morphism $f \in \mathbf{C}_1$ is said to be **monic** (or a **monomorphism**) if for any $(f, g), (f, h) \in \mathbf{C}_2$ where g and h have the same source, $f \circ g = f \circ h$ implies $g = h$. Equivalently, f is monic if $g = h$ whenever the following diagram commutes.

$$\begin{tikzcd} \bullet \arrow[r, bend left, "g"] \arrow[r, bend right, "h"'] & \bullet \arrow[r, "f"] & \bullet \end{tikzcd} \tag{18}$$

Standard notation for a monomorphism is $\bullet \hookrightarrow \bullet$ (`\hookrightarrow`).

Proposition 102. *Let $\mathbf{C}$ be a category and $f : A \to B$ a morphism, if there exists $f' : B \to A$ such that $f' \circ f = \mathrm{id}_A$,*[78] *then f is a monomorphism.*

[78] We say that f' is a **left inverse** of f.

Proof. If $f \circ g = f \circ h$, then $f' \circ f \circ g = f' \circ f \circ h$ implying $g = h$. $\square$

A monomorphism with a left inverse is called a **split monomorphism**.

Proposition 103. *Let $\mathbf{C}$ be a category and $(f_1, f_2) \in \mathbf{C}_2$, if $f_1 \circ f_2$ is a monomorphism, then f_2 is a monomorphism.*

Proof. Let $g, h \in \mathbf{C}_1$ be such that $f_2 \circ g = f_2 \circ h$, we readily get that $(f_1 \circ f_2) \circ g = (f_1 \circ f_2) \circ h$. Since $f_1 \circ f_2$ is a monomorphism, this implies $g = h$. $\square$

The last two results make it obvious that monomorphisms are analogous to injective functions and we will see that they are exactly the same in the category **Set**, but first let us introduce the dual concept. Given a definition or statement in an arbitrary category $\mathbf{C}$, one could view this concept inside the category $\mathbf{C}^{\mathrm{op}}$ and obtain a similar definition or statement where all morphisms and the order of composition are reversed , this is called the **dual** concept. Dualizing the definition of a monomorphism yields an epimorphism.

Definition 104 (Epimorphism)**.** Let $\mathbf{C}$ be a category, a morphism $f \in \mathbf{C}_1$ is said to be **epic** (or an **epimorphism**) if for any two morphisms $(g, f), (h, f) \in \mathbf{C}_2$ where g and h have the same target, $g \circ f = h \circ f$ implies $g = h$. Equivalently, f is epic if $g = h$ whenever the following diagram commutes.[79]

$$\begin{tikzcd} \bullet \arrow[r, "f"] & \bullet \arrow[r, bend left, "g"] \arrow[r, bend right, "h"'] & \bullet \end{tikzcd} \tag{19}$$

[79] Seeing the diagrams make it clearer that the concepts are dual.

Standard notation for an epimorphism is $\bullet \twoheadrightarrow \bullet$ (`\twoheadrightarrow`).

The dual versions of Propositions 102 and 103 also hold. Although translating our previous proofs to the dual case is straightforward, we will do the two next proofs relying on duality to convey the general sketch that works anytime a dual result needs to be proven.

Proposition 105. *Let $\mathbf{C}$ be a category and $f : A \to B$ a morphism, if there exists $f' : B \to A$ such that $f \circ f' = \mathrm{id}_B$, then f is epic.*[80]

[80] We say that f' is a **right inverse** of f.

Proof. Observe that f is epic in $\mathbf{C}$ if and only if f^{op} is monic in $\mathbf{C}^{\mathrm{op}}$ (reverse the arrows in the definition).[81] Moreover, by definition,

$$f'^{\mathrm{op}} \circ f^{\mathrm{op}} = (f \circ f')^{\mathrm{op}} = \mathrm{id}_B{}^{\mathrm{op}} = \mathrm{id}_{B^{\mathrm{op}}},$$

so by the result for monomorphisms, f^{op} is monic and hence f is epic. □

[81] This is one other way to see that two concepts are dual.

An epimorphism with a right inverse is called a **split epimorphism**.

Proposition 106. *Let $\mathbf{C}$ be a category and $(f_1, f_2) \in \mathbf{C}_2$, if $f_1 \circ f_2$ is epic, then f_2 is epic.*

Proof. Since $f_2{}^{\mathrm{op}} \circ f_1{}^{\mathrm{op}} = (f_1 \circ f_2)^{\mathrm{op}}$ is monic, the result for monomorphisms implies $f_2{}^{\mathrm{op}}$ is monic and hence f_2 is epic. □

Example 107 (**Set**)**.**

- A function $f : A \to B$ is a monomorphism in **Set** if and only if it is injective:[82]

  ($\Leftarrow$) Since f is injective, it has a left inverse, so it is monic by Proposition 102.

  ($\Rightarrow$) Given $a \in A$, let $g_a : \mathbf{1} := \{*\} \to A$ be the function sending $*$ to a. For any $a_1 \neq a_2 \in A$, the functions g_{a_1} and g_{a_2} are different, hence $f \circ g_{a_1} \neq f \circ g_{a_2}$. Therefore, $f(a_1) \neq f(a_2)$ and since a_1 and a_2 were arbitrary, f is injective.

[82] As a consequence, since all injective functions have a left inverse, all the monomorphisms in **Set** are split monic.

- A function $f : A \to B$ is an epimorphism if and only if it is surjective:[83]

  ($\Leftarrow$) Since f is surjective, it has a right inverse, so it is epic by Proposition 105.

  ($\Rightarrow$) Let $h : B \to \{0,1\} =: \mathbf{2}$ be the constant function at 1 and $g : B \to \mathbf{2}$ be the indicator function of $\mathrm{Im}(f) \subseteq B$, namely,

$$g(x) = \begin{cases} 1 & \exists a \in A, x = f(a) \\ 0 & \text{o/w} \end{cases}.$$

  It is clear that $g \circ f = h \circ f \equiv 1$ and since f is epic, it implies $g = h$. Thus, any element of B is in the image of f, that is f is surjective.

[83] If you assume the axiom of choice, all surjective functions have a right inverse and thus all epimorphisms in **Set** are split epic.

Example 108 (Mon). Inside the category **Mon**, the monomorphisms correspond exactly to injective homomorphisms.

($\Rightarrow$) Let $f : M \to M'$ be an injective homomorphisms and $g_1, g_2 : N \to M$ be two parallel homomorphisms. Suppose that $f \circ g_1 = f \circ g_2$, then for all $x \in N$, $f(g_1(x)) = f(g_2(x))$, so by injectivity of f, $g_1(x) = g_2(x)$. Therefore $g_1 = g_2$ and since g_1 and g_2 were arbitrary, f is a monomorphism.

($\Leftarrow$) Let $f : M \to M'$ be a monomorphism. Let $x, y \in M$ and define $p_x : \mathbb{N} \to M$ by $k \mapsto x^k$ and similarly for p_y. It is easy to show that p_x and p_y are homomorphisms.[84] If $f(x) = f(y)$, then, by the homomorphism property, for all $k \in \mathbb{N}$

$$f(p_x(k)) = f(x^k) = f(x)^k = f(y)^k = f(y^k) = f(p_y(k)).$$

In other words, we get $f \circ p_x = f \circ p_y$, so $p_x = p_y$ and $x = y$. This direction follows.

[84] It follows from the definition of x^k which is $x \overset{k}{\cdots} x$.

Conversely, an epimorphism is not necessarily surjective. For example, the inclusion homomorphism $i : \mathbb{N} \to \mathbb{Z}$ is clearly not surjective but it is an epimorphism. Indeed, let $g, h : \mathbb{Z} \to M$ be two monoid homomorphisms satisfying $g \circ i = h \circ i$. In particular, $g(n) = h(n)$ for any $n \in \mathbb{N} \subset \mathbb{Z}$. It remains to show that also $g(-n) = h(-n)$: we have

$$h(n)g(-n) = g(n)g(-n) = g(n-n) = g(0) = 1_M = h(0) = h(-n+n) = h(-n)h(n),$$

but then $g(-n) = h(-n)h(n)g(-n) = h(-n)$.

Definition 109 (Isomorphism)**.** Let **C** be a category, a morphism $f : A \to B$ is said to be an **isomorphism** if there exists a morphism $f^{-1} : B \to A$ such that $f \circ f^{-1} = \mathrm{id}_B$ and $f^{-1} \circ f = \mathrm{id}_A$.[85]

[85] Then f^{-1} is called the **inverse** of f.

As you might expect from the terminology, in general, we will not distinguish between isomorphic objects in a category because all the properties we care about will hold for one if and only if it holds for the other.

See solution.

Exercise 110. Show that composing monic/epic/isomorphisms yields monic/epic/isomorphisms.

Remark 111. The results shown about monic and epic morphisms[86] imply that any isomorphism is monic and epic. However, the converse is not true as witnessed by the inclusion morphism i described in Example 108.[87] If there exists an isomorphism between two objects A and B, then they are **isomorphic**, denoted $A \cong B$. Isomorphic objects are also isomorphic in the opposite category,[88] that is, the concept of **isomorphism** is **self-dual**.

[86] Proposition 102 and 105.

[87] This is not akin to the situation in **Set** because, there, all monomorphisms and epimorphisms are split (assuming the axiom of choice).

[88] Because the left inverse becomes the right inverse and vice-versa.

Example 112 (**Set**)**.** A function $f : X \to Y$ in $\mathbf{Set}_1$ has an inverse f^{-1} if and only if f is bijective, thus isomorphisms in **Set** are bijections. As a consequence, we have $A \cong B$ if and only if $|A| = |B|$.[89]

[89] This is in fact the definition of cardinality.

Example 113 (**Cat**)**.** An isomorphism in **Cat** is a functor $F : \mathbf{C} \rightsquigarrow \mathbf{D}$ with an inverse $F^{-1} : \mathbf{D} \rightsquigarrow \mathbf{C}$. This implies that F_0 and F_1 are bijections[90] because F_0^{-1} is the inverse of F_0 and F_1^{-1} is the inverse of F_1.

[90] Note that F_1 being a bijection is equivalent to F being fully faithful.

Conversely, if $F : \mathbf{C} \rightsquigarrow \mathbf{D}$ is a functor whose components on objects and morphisms are bijective, we can check that defining $F^{-1} : \mathbf{D} \rightsquigarrow \mathbf{C}$ with $F_0^{-1} := (F_0)^{-1}$ and $F_1^{-1} = (F_1)^{-1}$ yields a functor. Therefore, isomorphisms are precisely the fully faithful functors which are bijective on objects.

Examples 114 (Concrete categories)**.**

1. It is a simple exercise in an algebra class to show that isomorphisms in the categories **Mon**, **Grp**, **Ring**, **Field** and $\mathbf{Vect}_k$ are the isomorphisms in their respective theory.
2. In **Poset**, isomorphisms are bijective order-preserving functions.
3. In **Top**, it is not enough to have a bijective continuous function, we need to require that it has a continuous inverse. Such functions are called *homeomorphisms*.

Definition 115 (Initial object)**.** Let $\mathbf{C}$ be a category, an object $A \in \mathbf{C}_0$ is said to be **initial** if for any $B \in \mathbf{C}_0$, $|\mathrm{Hom}_{\mathbf{C}}(A, B)| = 1$, namely there are no two parallel morphisms with source A and every object has a morphism coming from A. The[91] initial object of a category, if it exists, is denoted $\emptyset$ and the *unique* morphism from $\emptyset$ to $X \in \mathbf{C}_0$ is denoted $() : \emptyset \to X$.

[91] We will soon see why we can use *the* instead of *an*.

Definition 116 (Terminal object)**.** Let $\mathbf{C}$ be a category, an object $A \in \mathbf{C}_0$ is said to be **terminal** (or final) if for any $B \in \mathbf{C}_0$, $|\mathrm{Hom}_{\mathbf{C}}(B, A)| = 1$, namely there are no two parallel morphisms with target A and every object has a morphism going to A. The terminal object of a category, if it exists, is denoted $\mathbf{1}$ and the *unique* morphism from $X \in \mathbf{C}_0$ into $\mathbf{1}$ is denoted $[] : X \to \mathbf{1}$.

Remark 117 (Notation)*.* The motivation behind the notations $\emptyset$ and $\mathbf{1}$ is given shortly, but the notations for the morphisms will be explained in Chapter .

An object is initial in a category $\mathbf{C}$ if and only if it is terminal in $\mathbf{C}^{\mathrm{op}}$. Also, if an object is initial and terminal, we say it is a **zero** object and usually denote it $\mathbf{0}$.

Example 118 (**Set**)**.** Let X be a set, there is a unique function from the empty set into X, it is the empty function.[92] We infer that the emptyset is the initial object in **Set**, hence the notation $\emptyset$. For the terminal object, we observe that there is a unique function $X \to \{*\}$ sending all elements of X to $*$, thus $\{*\}$ is terminal in **Set**.

[92] Recall (or learn here) that a function $f : A \to B$ is defined via subset of $f \subseteq A \times B$ that satisfies $\forall a \in A, \exists! b \in B, (a, b) \in f$. When A is empty, $A \times B$ is empty and the unique subset of $\emptyset \subseteq A \times B$ satisfies the condition vacuously. In passing, when B is empty but A is not, the unique subset of $A \times B$ does not satisfy the condition.

In this example, we could have chosen any singleton to show it is terminal. However, that choice is irrelevant to a good category theorician since, as any two singletons are isomorphic (because they have the same cardinality), any two terminal objects are isomorphic.

Proposition 119. *Let $\mathbf{C}$ be a category and $A, B \in \mathbf{C}_0$ be initial, then $A \cong B$.*

Proof. Let f be the single element in $\mathrm{Hom}_{\mathbf{C}}(A,B)$ and f' be the single element in $\mathrm{Hom}_{\mathbf{C}}(B,A)$. Since the identity morphisms are the only elements of $\mathrm{Hom}_{\mathbf{C}}(A,A)$ and $\mathrm{Hom}_{\mathbf{C}}(B,B)$, $f' \circ f$ and $f \circ f'$, belonging to these sets, must be the identities. In other words f and f' are inverses, thus $A \cong B$. □

The dual result follows.

Proposition 120. *Let* $\mathbf{C}$ *be a category and* $A, B \in \mathbf{C}_0$ *be terminal, then* $A \cong B$.

Moreover, initial (resp. terminal) objects are unique up to *unique* isomorphisms.

Exercise 121. Show that in **Cat**, the initial object is the empty category (no objects and no morphisms) and the terminal object is $\mathbf{1}$ (hence the notation).[93]

[93] **Hint**: the unique functor $[] : \mathbf{C} \to \mathbf{1}$ is the constant functor at the object $\bullet \in \mathbf{1}_0$.

Example 122 (Grp). Similarly to **Set**, the trivial group with one element is terminal in **Grp**. Moreover, note that there are no empty group (because there is no identity element), but any group homomorphism from $\{1\}$ into a group G must send 1 to 1_G, which completely determines the homomorphism. Therefore, the trivial group is also initial in **Grp**, it is the zero object.

Examples 123. Here are more examples of categories where initial and terminal objects may or may not exist.

1. $\exists$ terminal, $\nexists$ initial: Consider the poset $(\mathbb{N}, \geq)$ represented by diagram (20). It is clear that 0 is terminal and no element can be initial because $0 \geq x$ implies $x = 0$.

$$\overset{0}{\bullet} \longleftarrow \overset{1}{\bullet} \longleftarrow \overset{2}{\bullet} \longleftarrow \cdots \qquad (20)$$

2. $\nexists$ terminal, $\exists$ initial:[94] The category **FinGrpInj** where the objects are finite groups and the morphisms are injective homomorphisms only contains an initial object $\{1\}$. Indeed, an injective homomorphism $G \hookrightarrow H$ can be seen as subgroup of H isomorphic to G. The trivial group $\{1\}$ can only be isomorphic to the subgroup $\{1_H\}$ as any other element has degree more than 1, so $\{1\}$ is initial. Moreover, a group G cannot be terminal as $G \times (\mathbb{Z}/2\mathbb{Z})$ cannot be isomorphic to any subgroup of G.

[94] Of course, you could take the opposite of $(\mathbb{N}, \geq)$, that is $(\mathbb{N}, \leq)$, but that is not fun.

3. $\nexists$ terminal, $\nexists$ initial: Let G be a non trivial group, the delooping of G has no terminal and no initial objects. The category $\mathbf{B}(G)$ has a single object $*$ with $\mathrm{Hom}_{\mathbf{B}(G)}(*,*) = G$, so $*$ cannot be initial nor terminal when $|G| > 1$.

   For a more interesting example, consider the category **Field**. Its underlying oriented graph is disconnected[95] because there are no field homomorphisms between fields of different characteristic. Therefore, **Field** has no initial nor terminal objects.

[95] There are objects with no morphisms between them.

4. $\exists$ terminal, $\exists$ initial: Let X be a non-empty topological space where τ is the collection of open sets.[96] The category of open sets $\mathcal{O}(X)$ satisfies

$$\mathrm{Hom}_{\mathcal{O}(X)}(U,V) = \begin{cases} \{i_{U,V}\} & U \subseteq V \\ \varnothing & U \nsubseteq V \end{cases}$$

[96] Recall that it must contain $\varnothing$ and X.

Since the empty set is contained in every open set, it is an initial object. Since the full set X contains every open set, it is a terminal object. No other set can be initial as it cannot be contained in $\emptyset$ nor be terminal as it cannot contain X. Moreover, note that the two objects are not isomorphic because $X \not\subseteq \emptyset$.

Example 124. For our last application of duality in this chapter,[97] let X be a set and consider the posetal category $(\mathcal{P}(X), \subseteq)$. We would like to define the union of two subsets of X in this category. The usual definition $A \cup B = \{x \in X \mid x \in A \text{ or } x \in B\}$ is not suitable because the data in the posetal category $\mathcal{P}(X)$ never refers to elements of X. In particular, the subsets $A, B \subseteq X$ are simply objects in the category and it is not clear to us how we can determine what elements are in A and B with our categorical tools (objects and morphisms).

We propose another characterization of the union of A and B. First, what is obvious, $A \cup B$ contains A and it contains B. Second, $A \cup B$ is the smallest subset of X containing A and B. Indeed, if $Y \subseteq X$ contains all element in A and B, then it also contains $A \cup B$. Using the order $\subseteq$ (or equivalently, the morphisms in the category $\mathcal{P}(X)$), we have $A, B \subseteq A \cup B$ and $\forall Y$ s.t. $A, B \subseteq Y$ then $A \cup B \subseteq Y$.[98] This yields a definition of $\cup$ within the category $\mathcal{P}(X)$, which means we can dualize it.

The dual of this property (reversing all inclusions) is as follows.[99]

$$A \square B \subseteq A, B \text{ and } \forall Y \text{ s.t. } Y \subseteq A, B \text{ then } Y \subseteq A \square B$$

Putting this in words, $A \square B$ is the largest subset of X which is contained in A and B. That is, of course, the intersection $A \cap B$. In this way, union and intersection are dual operations. If you search your memory for properties about union and intersection that you proved when you first learned about sets, you will find that they usually come in pairs; the first property being the dual of the second.

[97] Don't worry, we will have plenty of opportunities to use duality later.

[98] We leave it as an exercise to show that $A \cup B$ is the only subset of X satisfying this property.

[99] The symbol $\square$ is a placeholder for the operation which we will find to be dual to union.

More Vocabulary

Exercise 125. Let $\mathbf{C}$ be a category and $X \in \mathbf{C}_0$, we define the relation $\sim$ on monomorphisms $Y \hookrightarrow X$ by

See solution.

$$m \sim m' \Leftrightarrow \exists \text{ isomorphism } i, m = m' \circ i.$$

Show that $\sim$ is an equivalence relation.

Definition 126 (Subobject)**.** Let $\mathbf{C}$ be a category, a **subobject** of $X \in \mathbf{C}_0$ is a equivalence class of the relation $\sim$ defined above. We will often abusively refer to a subobject simply with a monomorphism $Y \hookrightarrow X$. The collection of subobjects of X is denoted $\mathrm{Sub}_{\mathbf{C}}(X)$. If for any $X \in \mathbf{C}_0$, $\mathrm{Sub}_{\mathbf{C}}(X)$ is a set, we say that $\mathbf{C}$ is **well-powered**.

Exercise 127. Let $\mathbf{C}$ be a category and $X \in \mathbf{C}_0$, we define the relation $\leq$ on $\mathrm{Sub}_{\mathbf{C}}(X)$ by

See solution.

$$[m] \leq [m'] \Leftrightarrow \exists \text{ morphism } k, m = m' \circ k.$$

Show that $\leq$ is a well-defined partial order.

Limits and Colimits

The unifying power of categorical abstraction is arguably its biggest benefit. Indeed, it is often the case that many mathematical objects or results from different fields fit under the same categorical definition or fact. In my opinion, category theory is at its peak of elegance when a complex idea becomes close to trivial when viewed categorically, and when this same view helps link together the intuitions behind many ideas throughout mathematics.

The next two chapters concern one particular instance of this power, that is, the use of universal properties to define mathematical constructions. This term is somewhat delicate to define, therefore, we postpone its definition to next chapter and for a while, we suggest the reader to try and recognize *universality* as the thing that all definitions of (co)limits have in common. This chapter will cover limits and colimits which are specific cases of universal constructions.

The first section presents several examples; each of its subsection is dedicated to one kind of limit or colimit of which a detailed example in **Set** is given along with a couple of interesting examples in other categories. The second section gives a formal framework to talk about all the examples previously explored as well as a few general results. In the sequel, **C** denotes a category.

Examples

Product

Given two sets S and T, the most common construction of the Cartesian product $S \times T$ is conceptually easy: you take all pairs of elements S and T, that is,

$$S \times T := \{(s,t) \mid s \in S, t \in T\}.$$

However, this does not have a nice categorical analog because it requires to pick out elements in S and T. If one hopes to generalize products to other categories, the construction must only involve objects and morphisms.

Question 128. *What are significant functions (morphisms in **Set**) to consider when studying $S \times T$?*

Answer. Projection maps. They are functions $\pi_1 : S \times T \to S$ and $\pi_2 : S \times T \to T$,[100] but that is not enough to define the product. Indeed, there are projection maps

[100] The projections are defined by $\pi_1(s,t) = s$ and $\pi_2(s,t) = t$ for all $(s,t) \in S \times T$.

$\pi'_1 : S \times T \times S \to S$ and $\pi'_2 : S \times T \times S \to T$, but $S \times T \times S$ is not always isomorphic to $S \times T$. $\square$

Question 129. *What is unique*[101] *about $S \times T$ with the projections π_1 and π_2?*

[101] Always up to isomorphism of course.

Answer. For one, π_1 and π_2 are surjective and while they are not injective, they have an invertible-like property. Namely, given $s \in S$ and $t \in T$, the pair (s,t) is completely determined from $\pi_1^{-1}(s) \cap \pi_2^{-1}(t)$. $\square$

Again, in order to discharge the references to specific elements, another point of view is needed. Let X be a set of *choices* of pairs, an element $x \in X$ chooses elements in S and T via functions $c_1 : X \to S$ and $c_2 : X \to T$ (similar to the projections). Now, the *quasi-inverse* defined above yields a function

$$! : X \to S \times T = x \mapsto \pi^{-1}(c_1(x)) \cap \pi^{-1}(c_2(x)).$$

This function maps $x \in X$ to an element in $S \times T$ that makes the same choice as x, and it is the only one that does so. Categorically, $!$ is the unique morphism in $\mathrm{Hom}_{\mathbf{C}}(X, S \times T)$ satisfying $\pi_i \circ ! = c_i$ for $i = 1, 2$. Later, we will see that this property completely determines $S \times T$. For now, enjoy the power we gain from generalizing this idea.

Definition 130 (Binary product)**.** Let $A, B \in \mathbf{C}_0$. A (categorical) **binary product** of A and B is an object, denoted $A \times B$, along with two morphisms $\pi_A : A \times B \to A$ and $\pi_B : A \times B \to B$ called **projections** that satisfy the following universal property[102]: for every object $X \in \mathbf{C}_0$ with morphisms $f_A : X \to A$ and $f_B : X \to B$, there is a unique morphism $! : X \to A \times B$ making diagram (21) commute.[103]

[102] Remember that the word universal is not yet defined, we are trying to give you an idea of what it means with these examples.

[103] It is common to denote $! = (f_A, f_B)$.

$$\begin{tikzcd} & X \arrow[dl, "f_A"'] \arrow[d, dashed, "!"] \arrow[dr, "f_B"] & \\ A & A \times B \arrow[l, "\pi_A"] \arrow[r, "\pi_B"'] & B \end{tikzcd} \tag{21}$$

Example 131 (**Set**)**.** Cleaning up the argument above, we show that the Cartesian product $A \times B$ with the usual projections is a binary product in **Set**. To show that it satisfies the universal property, let X, f_A and f_B be as in the definition. A function $! : X \to A \times B$ that makes (21) commute must satisfy

$$\forall x \in X, \pi_A(!(x)) = f_A(x) \text{ and } \pi_B(!(x)) = f_B(x).$$

Equivalently, $!(x) = (f_A(x), f_B(x))$. Since this uniquely determines $!$, $A \times B$ is indeed the binary product.

Examples 132. Most of the constructions throughout mathematics with the name product can also be realized with a binary product. Examples include the direct product of groups, rings or vector spaces, the product of topologies, etc. The fact that all these constructions are based on the Cartesian product of the underlying sets is a corollary of a deeper result about the forgetful functor that all these categories have in common.[104]

[104] cf.

In another flavour, let X be a topological space and $\mathcal{O}(X)$ be the category of opens. If $A, B \subseteq X$ are open, what is their product? Following Definition 130, the existence of π_A and π_B imply that $A \times B$[105] is included in both sets, or equivalently $A \times B \subseteq A \cap B$.

Moreover, for any open set X included in A and B (via f_A and f_B), X should be included in $A \times B$ (via !).[106] In particular, X can be $A \cap B$ (it is open by definition of a topology), thus $A \cap B \subseteq A \times B$. In conclusion, the product of two open sets is their intersection. In an arbitrary poset, the same argument is used to show the product is the greatest lower bound/infimum/meet.

[105] Recall that $\times$ denotes the categorical product, not the Cartesian product of sets.

[106] Notice that uniqueness of ! is already given in a posetal category.

Remark 133. Given two objects in an arbitrary category, their product does not necessarily exist. Nevertheless, when it exists, one can (and we will) show that it is unique up to unique isomorphism.[107] Thus, in the sequel, we will speak of *the* product of two objects and similarly for other constructions presented in this chapter.

[107] The uniqueness of the isomorphism is under the condition that it preserves the structure of the product. We will clear up this subtlety in Remark 170.

To generalize the categorical product to more than two objects, one can, for instance, define the product of a finite family of sets recursively with the binary product.[108] However, this implies having to show the associativity and commutativity of $\times$ for it to be well-defined.[109] In contrast, generalizing the universal property illustrated in (21) yields a simpler definition that works even for arbitrary families.

[108] For a family $\{X_1, \dots, X_n\} \subseteq \mathbf{C}_0$:

$$\prod_{i=1}^{n} X_i = \begin{cases} X_1 & n = 1 \\ \left(\prod_{i=1}^{n-1} X_i\right) \times X_n & \end{cases}$$

[109] These proofs are not very involved, they heavily rely on uniqueness, cf. Exercises.

Definition 134 (Product)**.** Let $\{X\}_{i \in I}$ be an I–indexed family of objects of $\mathbf{C}$. The **product** of this family is an object, denoted $\prod_{i \in I} X_i$ along with projections $\pi_j : \prod_{i \in I} X_i \to X_j$ for all $j \in I$ satisfying the following universal property: for any object X with morphisms $\{f_j : X \to X_j\}_{j \in I}$, there is a unique morphism $! : X \to \prod_{i \in I} X_i$ making (22) commute for all $j \in I$.[110]

$$\begin{tikzcd} X \arrow[d, dashed, "!"'] \arrow[dr, "f_j"] & \\ \prod_{i \in I} X_i \arrow[r, "\pi_j"'] & X_j \end{tikzcd} \tag{22}$$

In the case of the category of open subsets of a topological space, the arbitrary product is not always the intersection. This is because arbitrary intersections of open sets are not necessarily open. To resolve this problem, it suffices to take the interior of the intersection which is open by definition.

[110] Analogously to the binary case, we may write $! = (f_j)_{j \in I}$ or, in the finite case, $! = (f_1, \dots, f_n)$.

A family of objects in a category is also called a discrete diagram,[111] the product is then the limit of this diagram.

[111] The terminology comes from Definition 158.

Exercise 135. Show that the product of an arbitrary family of sets is still the Cartesian product of this family.

See solution.

Exercise 136 (NOW!)**.** Let $\{f_i : X_i \to Y_i\}_{i \in I}$ be a family of morphisms in $\mathbf{C}$, show that there is a unique morphism $\prod_{i \in I} f_i : \prod_{i \in I} X_i \to \prod_{i \in I} Y_i$ making the following square commute for all $j \in I$.

$$\begin{tikzcd} \prod_{i \in I} X_i \arrow[r, "\prod_{i \in I} f_i"] \arrow[d, "\pi_j"'] & \prod_{i \in I} Y_i \arrow[d, "\pi_j"] \\ X_j \arrow[r, "f_j"'] & Y_j \end{tikzcd} \tag{23}$$

In the finite case, we will write $f_1 \times \cdots \times f_n$.

The big takeaway from last chapter is that each time we read a new definition, it is worth to dualize. Thus we ask, what is the colimit of a discrete diagram?

Coproduct

Definition 137 (Coproduct)**.** Let $\{X\}_{i\in I}$ be an I–indexed family of objects in $\mathbf{C}$, its **coproduct** is an object, denoted $\coprod_{i\in I} X_i$ (or $X_1 + X_2$ in the binary case), along with morphisms $\kappa_j : X_j \to \coprod_{i\in I} X_i$ for all $j \in I$ called **coprojections** satisfying the following universal property: for any object X with morphisms $\{f_j : X_j \to X\}_{j\in I}$, there is a unique morphism $! : \coprod_{i\in I} X_i \to X$ making (24) commute for all $j \in I$.[112]

$$\begin{tikzcd} X_j \arrow[r, "\kappa_j"] \arrow[dr, "f_j"'] & \coprod_{i\in I} X_i \arrow[d, dashed, "!"] \\ & X \end{tikzcd} \tag{24}$$

[112] We may denote $! = [f_j]_{j\in I}$ or, in the finite case, $! = [f_1, \dots, f_n]$.

Let us find out what coproducts of sets are.

Example 138 (**Set**)**.** Let $\{X_i\}_{i\in I}$ be a family of sets, first note that if $X_j = \emptyset$ for $j \in I$, then there is only one morphism $X_j \to X$ for any X. In particular, (24) commutes no matter what $\coprod_{i\in I} X_i$ and X are. Therefore, removing X_j from this family does not change how the coproduct behaves, hence no generality is loss from assuming all X_is are non-empty.

Second, for any $j \in I$, let $X = X_j$, $f_j = \mathrm{id}_{X_j}$ and for any $j' \neq j$, let $f_{j'}$ be any morphism in $\mathrm{Hom}(X_{j'}, X_j)$.[113] Commutativity of (24) implies κ_j has a left inverse because $! \circ \kappa_j = f_j = \mathrm{id}_{X_j}$, so all coprojections are injective.

[113] One exists because X_j is non-empty.

Third, we claim that for any $j \neq j' \in I$, $\mathrm{Im}(\kappa_j) \cap \mathrm{Im}(\kappa_{j'}) = \emptyset$. Assume towards a contradiction that there exists $j \neq j' \in I$, $x \in X_j$ and $x' \in X_{j'}$ such that $\kappa_j(x) = \kappa_{j'}(x')$. Then, let $X = \{0,1\}$, $f_j \equiv 0$, $f_{j'} \equiv 1$ and the other morphisms be chosen arbitrarily. The universal property implies that $! \circ \kappa_j \equiv 0$ and $! \circ \kappa_{j'} \equiv 1$, but it contradicts $!(\kappa_j(x)) = !(\kappa_{j'}(x'))$.

Finally, the previous point says that $\coprod_{i\in I} X_i$ contains distinct copies of the images of all coprojections. Furthermore, the κ_js being injective, their image can be identified with the X_js to obtain[114]

$$\bigsqcup_{i\in I} X_i \subseteq \coprod_{i\in I} X_i.$$

[114] The symbol $\sqcup$ denotes the disjoint union of sets.

For the converse inclusion, in (24), let X be the disjoint union and the f_js be the inclusions. Assume there exists x in the R.H.S. that is not in the L.H.S., then we can define $!' : \coprod_{i\in I} X_i \to \bigsqcup_{i\in I} X_i$ that only differs from $!$ at x. Since x is not in the image of any of the κ_j, the diagrams still commute and this contradicts the uniqueness of $!$.

In conclusion, the coproduct in **Set** is the disjoint union and the coprojections are the inclusions.[115]

[115] We recover the intuition for why empty sets can be ignored. This is a general fact proved in Exercise

Remark 139. If this example looks more complicated than the product of sets, it is because we started knowing nothing concrete about coproducts of sets and gradually discovered what properties they had using specific objects and morphisms we know exist in **Set**. In contrast, we knew what products of sets were and we just had to show they satisfied the universal property.[116]

In general, the hard part is to find what construction satisfies a universal property, proving it does is easier.

Examples 140. In the category of open sets of (X, τ)**:** let $\{U_i\}_{i \in I}$ be a family of open sets and suppose $\coprod_i U_i$ exists. The coprojections yield inclusions $U_j \subseteq \coprod_i U_i$ for all $j \in I$, so $\coprod_i U_i$ must contain all U_js and thus $\cup_i U_i$. Moreover, in (24), letting f_j be the inclusion $U_j \hookrightarrow \cup_i U_i$ for all $j \in I$,[117] the existence of ! yields an inclusion $\coprod_i U_i \subseteq \cup_i U_i$. We conclude that the coproduct in this category is the union. In an arbitrary poset, the same argument is used to show the coproduct is the least upper bound/supremum/join.

In Vect$_k$: the coproduct, also called the direct sum, is defined by[118]

$$\coprod_{i \in I} V_i = \bigoplus_{i \in I} V_i := \left\{ v \in \prod_{i \in I} V_i \mid v(i) \neq 0 \text{ for finitely many } i\text{'s} \right\},$$

where $\kappa_j : V_j \hookrightarrow \coprod_i V_i$ sends v to $\bar{v} \in \prod_i V_i$ with $\bar{v}_j = v$ and $\bar{v}_{j'} = 0$ whenever $j \neq j'$. To verify this, let $\{f_j : V_j \to X\}_{j \in I}$ be a family of linear maps. We can construct ! by defining it on basis elements of the direct sum, which are just the basis elements of all V_js seen as elements of the sum (via the coprojections).[119] Indeed, if b is in the basis of V_j, we let $!(\bar{b}) = f_j(b)$. Extending linearly yields a linear map $! : \coprod_i V_i \to X$. Uniqueness is clear because if $h : \coprod_i V_i \to X$ differs from ! on one of the basis elements, it does not make (24) commute.

Exercise 141. Show that products are dual to coproducts, namely, if a product of a familiy $\{X_i\}_{i \in I}$ exists in $\mathbf{C}$, then this object is the coproduct of this family in $\mathbf{C}^{\mathrm{op}}$ and vice-versa. Conclude that you can define the **coproduct morphisms** dually to Exercise 136, we denote them $\coprod_{i \in I} f_i$ or $f_1 + \cdots + f_n$ in the finite case.

See solution.

Exercise 142. Let $\mathbf{C}$ be a category with a terminal object $\mathbf{1}$. Show that the assignment $X \mapsto X + \mathbf{1}$ is functorial, i.e.: define the action of $(- + \mathbf{1})$ on morphisms and show it satisfies the axioms of a functor.[120]

See solution.

In a very similar way to the product and coproduct, we will define various constructions in **Set** as limits or colimits.

Equalizer

Definition 143 (Fork)**.** A **fork** in $\mathbf{C}$ is a diagram of shape (25) or (26) that commutes.[121]

$$O \xrightarrow{o} A \underset{g}{\overset{f}{\rightrightarrows}} B \qquad (25) \qquad\qquad A \underset{g}{\overset{f}{\rightrightarrows}} B \xrightarrow{o} O \qquad (26)$$

Because these are dual notions, we will prefer to call (26) a **cofork**.

Definition 144 (Equalizer)**.** Let $A, B \in \mathbf{C}_0$ and $f, g : A \to B$ be parallel morphisms. The equalizer of f and g is an object E and a morphism $e : E \to A$ satisfying

[116] One might argue that coming up with this universal property was the hard part in that case.

[117] These morphisms are in $\mathcal{O}(X)$ because $\cup_i U_i$ is open.

[118] Here, the symbol $\prod$ denotes the Cartesian product of the V_is as sets. The categorical product of vector spaces is also the direct sum, where the projections are the usual ones.

[119] It is necessary to require finitely many non-zero entries, otherwise the basis of the coproduct would not be the union of all bases of the V_js.

[120] We call $(- + \mathbf{1})$ the **maybe functor**.

[121] Again, we make use of our convention that commutativity does not make parallel morphisms equal.

$f \circ e = g \circ e$ with the following universal property: for any object O with morphism $o : O \to A$ satisfying $f \circ o = g \circ o$, there is a unique $! : O \to E$ making (27) commute.

$$\begin{tikzcd} O \arrow[d, dashed, "!"'] \arrow[dr, "o"] & & \\ E \arrow[r, "e"'] & A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \end{tikzcd} \tag{27}$$

Example 145 (**Set**)**.** Let $f, g : A \to B$ be two functions and suppose $e : E \to A$ is their equalizer. By associativity, for any $h : O \to E$, the composite $e \circ h$ is a candidate for o in diagram (27) because $f \circ (e \circ h) = g \circ (e \circ h)$. What is more, if h' is such that $e \circ h = e \circ h'$, then $h = h'$ or it would contradict the uniqueness of !. In other words, e is monic/injective.[122]

This implies E can be identified with its image under e. Since e makes a fork with f and g, its image is contained in the subset $\{a \in A \mid f(a) = g(a)\}$. But, by the universal property, letting O be this set and o be the inclusion, there is an injection[123] $! : \{a \in A \mid f(a) = g(a)\} \hookrightarrow E$, thus both sets are equal. In conclusion, the equalizer of two parallel functions is the subset E in which they are equal and $e : E \hookrightarrow A$ is the inclusion.

[122] This argument was independent of the category, hence we can conclude that an equalizer of parallel morphisms is always monic.

[123] The fact that ! is an injection comes from the fact that the inclusion o is an injection and $e \circ ! = o$.

Examples 146. In a posetal category: hom–sets are singletons, so it must be the case that $f = g$ whenever f and g are parallel. Therefore, any $o : O \to A$ satisfies $f \circ o = g \circ o$. Written using the order notation, the universal property is then equivalent to the fact that $O \leq A$ implies $O \leq E$. In particular, if $O = A$, then $A \leq E$, so $A = E$ by antisymmetry.

In Ab, Ring or Vect$_k$: For the same reason that the Cartesian product of the underlying sets is the underlying set of the product,[124] the construction of equalizers is as in **Set**. Nevertheless, since each of these categories have a notion of additive inverse for morphisms, the equalizer of f and g has a cooler name, that is, $\ker(f - g)$.[125]

[124] We explain this later:

[125] The equalizer of f and g is the subset of A where f and g are equal, or equivalently, where $f - g$ is 0 (when $f - g$ and 0 are defined).

The equalizer of f and g is the limit of the diagram containing only the two parallel morphisms, we define its colimit in the next section.

Coequalizer

Definition 147 (Coequalizer)**.** Let $A, B \in \mathbf{C}_0$ and $f, g : A \to B$ be parallel morphisms. The **coequalizer** of f and g is an object D and a morphism $d : B \to D$ satisfying $d \circ f = d \circ g$ with the following universal property: for any object O with morphism $o : B \to O$ satisfying $o \circ f = o \circ g$, there is a unique $! : D \to O$ making (28) commute.

$$\begin{tikzcd} A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \arrow[r, "d"] \arrow[dr, "o"'] & D \arrow[d, dashed, "!"] \\ & & O \end{tikzcd} \tag{28}$$

Example 148 (**Set**)**.** Let $f, g : A \to B$ be two functions and suppose $d : B \to D$ is their coequalizer. Similarly to the dual case, one can show that d is epic/surjective.

Since $d \circ f = d \circ g$, for any $b, b' \in B$,

$$\big(\exists a \in A, f(a) = b \text{ and } g(a) = b'\big) \implies d(b) = d(b'). \tag{$*$}$$

Denoting $\sim$ to be the relation in the L.H.S. of $(*)$, the implication is $b \sim b' \implies d(b) = d(b')$. Note that $\sim$ is not an equivalence relation while $=$ is, thus, the converse implication does not always hold. For instance, when $b \sim b' \sim b''$, $d(b) = d(b'')$, but it might not be the case that $b \sim b''$.

Consequently, it makes sense to consider the equivalence relation generated by $\sim$,[126] denoted $\simeq$. As noted above, the forward implication $b \simeq b' \implies d(b) = d(b')$ still holds. For the converse, in (28), let $O := B/\simeq$ and $o : B \to B/\simeq$ be the quotient map, by post-composing with $!$, we have

[126] In this case, it is simply the transitive closure.

$$d(b) = d(b') \implies o(b) = o(b') \implies b \simeq b'.$$

In conclusion, $D = B/\simeq$ and $d : B \to D$ is the quotient map.

Examples 149. In a posetal category: an argument dual to the one for equalizers shows the coequalizer of $f, g : A \to B$ is B.

In Ab, Ring or Vect$_k$: Let $f, g : A \to B$ be homomorphisms and suppose $d : B \to D$ is their coequalizers. Consider the homomorphism $f - g$, since d makes a cofork with f and g, $d \circ (f - g) = d \circ f - d \circ g = 0$, or equivalently, $\operatorname{Im}(f - g) \subseteq \ker(d)$. Now, consider diagram (29) as a particular instance of (28), where q is the quotient map.[127]

[127] It is commutative because $q \circ (f - g) = 0$ by definition of q.

$$\begin{tikzcd} A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \arrow[r, "d"] \arrow[dr, "q"'] & D \arrow[d, dashed, "!"] \\ & & B/\operatorname{Im}(f-g) \end{tikzcd} \tag{29}$$

We claim that $!$ has an inverse, implying that $D \cong B/\operatorname{Im}(f - g)$. Indeed, for $[x] \in B/\operatorname{Im}(f - g)$, we must have

$$!^{-1}([x]) = !^{-1}(q(x)) = !^{-1}(!(d(x))) = d(x),$$

and it is only left to show $!^{-1}$ is well-defined because the inverse of a homomorphism is a homomorphism. This follows because if $[x] = [x']$, then there exists $y \in \operatorname{Im}(f - g)$ such that $x = x' + y$, so

$$!^{-1}(x) = d(x) = d(x' + y) = d(x') + d(y) = d(x') + 0 = !^{-1}(x').$$

In the special case that $g \equiv 0$, $B/\operatorname{Im}(f)$ is called the *cokernel* of f, denoted $\operatorname{coker}(f)$.

Monoid presentations: Let M be a monoid, recall that a set $A \subseteq M$ *generates* M, denoted $M = \langle A \rangle$, if any element of M is a finite product of elements of A. Namely, for any $m \in M$, there exists $a_1, \dots, a_n \in A$ such that $a_1 \cdots a_n = m$. If we consider the set of all finite products on A, call it $F(A)$, $M = \langle A \rangle$ yields a surjection $F(A) \to M$. However, the converse is not true because such a surjection does not necessarily behave well with the monoid operation.

However, there is a natural monoid operation on $F(A)$, that is concatenation:

$$(a_1 \cdots a_n) \cdot (a'_1 \cdots a'_m) = a_1 \cdots a_n a'_1 \cdots a'_m,$$

with the empty product as the identity.[128] Now, a surjective homomorphism $d : F(A) \twoheadrightarrow M$ does imply $M = \langle A \rangle$. Indeed, a product $a_1 \cdots a_n$ in the preimage of m has to equal m inside M or it would contradict the homomorphism property.

[128] Even if $1_M \in A$, the identity of $F(A)$ is still the empty product because $1_M a \neq a$ as elements of $F(A)$.

By the first isomorphism theorem, M is isomorphic to $F(A)/\ker(d)$. To realize d as a coequalizer, we will find a morphism f such that $\mathrm{coker}(f)$ is $M \cong F(A)/\ker(d)$, namely, we need to find $f : X \to F(A)$ with $\mathrm{Im}(f) = \ker(d)$.[129] This is similar to what we were doing at the start of this example. Indeed, let $R \subseteq F(A)$ be a set of generators of $\ker(d)$, then there is a homomorphism $f : F(R) \to F(A)$ satisfying $\mathrm{Im}(f) = \ker(d)$. In fact, we can take the morphism f that simply views products of products of A as products of A by concatenation. We have shown that (30) forms a fork and the argument used in **Ab** can be applied here to show this is a coequalizer.

[129] In this category, g is not 0 but 1 everywhere.

$$F(R) \underset{1}{\overset{f}{\rightrightarrows}} F(A) \xrightarrow{d} M \cong F(A)/\ker(d) \tag{30}$$

Thus, one can see M as generated by A subject to R that identify some products of A with the identity. Elements of R are called *relations* and the pair A and R is a *presentation* of M, denoted $M = \langle A \mid R \rangle$.

Pullback

Definition 150 (Cospan)**.** A **cospan** in **C** is comprised of three objects A, B, C and two morphisms f and g as in (31).

$$A \xrightarrow{f} C \xleftarrow{g} B \tag{31}$$

Definition 151 (Pullback)**.** Let $A \xrightarrow{f} C \xleftarrow{g} B$ be a cospan in **C**. Its **pullback** is an object, denoted $A \times_C B$, along with morphisms $p_A : A \times_C B \to A$ and $p_B : A \times_C B \to B$ such that $f \circ p_A = g \circ p_B$ and the following universal property holds: for any object X and morphisms $s : X \to A$ and $t : X \to B$ satisfying $f \circ s = g \circ t$, there is a unique morphism $! : X \to A \times_C B$ making (32) commute.[130]

[130] The $\lrcorner$ symbol is a standard convention to specify that the square is not only commutative, but also a pullback square.

$$\begin{tikzcd} X \arrow[dr, dashed, "!"] \arrow[drr, bend left, "t"] \arrow[ddr, bend right, "s"'] & & \\ & A \times_C B \arrow[r, "p_B"] \arrow[d, "p_A"'] \arrow[dr, phantom, "\lrcorner"] & B \arrow[d, "g"] \\ & A \arrow[r, "f"'] & C \end{tikzcd} \tag{32}$$

Example 152 (**Set**)**.** Let $A \xrightarrow{f} C \xleftarrow{g} B$ be a cospan in **Set** and suppose that its pullback is $A \xleftarrow{p_A} A \times_C B \xrightarrow{p_B} B$. Observe that p_A and p_B look like projections, and in fact, by the universality of the product $A \times B$, there is a map $h : A \times_C B \to A \times B$ such that $h(x) = (p_A(x), p_B(x))$ ((33) commutes). Consider the image of h, if $(a, b) \in \mathrm{Im}(h)$, then there exists $x \in A \times_C B$ such that $p_A(x) = a$ and $p_B(x) = b$. Moreover, the commutativity of the square in (33) implies $f(a) = g(b)$, hence

$$\begin{tikzcd} A \times_C B \arrow[rr, "p_B"] \arrow[dr, dashed, "h"] \arrow[dd, "p_A"'] & & B \arrow[dd, "g"] \\ & A \times B \arrow[ur, "\pi_B"'] \arrow[dl, "\pi_A"] & \\ A \arrow[rr, "f"'] & & C \end{tikzcd} \tag{33}$$

$$\mathrm{Im}(h) \subseteq \{(a, b) \in A \times B \mid f(a) = g(b)\} =: E.$$

Now, letting $X = E$, $s = \pi_A$ and $t = \pi_B$, by definition, $f \circ s = g \circ t$ hence, there is a unique $! : E \to A \times_C B$ satisfying $p_A \circ ! = \pi_A$ and $p_B \circ ! = \pi_B$. Viewing h as going in the opposite direction to $!$,[131] it is easy to see that for any $(a, b) \in E$,[132]

$$(h \circ !)(a, b) = (p_A(!(a, b)), p_B(a, b)) = (\pi_A(a, b), \pi_B(a, b)) = (a, b),$$

thus $!$ has a left inverse and is injective. Assume towards a contradiction that it is not surjective, then let $y \in A \times_C B$ not be in the image of $!$ and denote $x = !(p_A(y), p_B(y))$. Define $!'$ as acting exactly like $!$ except on $(p_A(y), p_B(y))$ where it goes to y instead of x. This ensure that $!'$ still makes the diagram commutes, but this contradicts the uniqueness of $!$.

As a particular case, if a cospan is comprised of two inclusions $A \hookrightarrow C \hookleftarrow B$, then its pullback is the intersection $A \cap B$ with p_A and p_B being the inclusions.

[131] We just saw that the image of h is contained in E, so we can see h as a function $h : A \times_C B \to A \times B$.

[132] We use the fact that $\pi_A \circ h \circ ! = p_A \circ !$ and similarly for B.

Examples 153. **In a posetal category**, the commutativity of the square in (32) does not depend on the morphisms, thus the universal property is equivalent to the property of being a product.

Exercise 154. Let $f : X \to Y$ be a morphism in $\mathbf{C}$. Show f is monic if and only if the square in (34) is a pullback.[133]

$$\begin{tikzcd} X \arrow[r, "\mathrm{id}_X"] \arrow[d, "\mathrm{id}_X"'] \arrow[dr, phantom, "\lrcorner", very near start] & X \arrow[d, "f"] \\ X \arrow[r, "f"'] & Y \end{tikzcd} \tag{34}$$

State and prove the dual statement.

See solution.

[133] This result and its dual will sometimes be used to treat monomorphisms (resp. epimorphisms) as limits (resp. colimits). In most of these cases, it will be crucial that this limit (resp. colimit) only involves the monomorphism (resp. epimorphism) and the identity morphism which is preserved by any functor.

Pushout

$\ulcorner$ **Definition 155** (Span)**.** A **span** in $\mathbf{C}$ is comprised of three objects A, B, C and two morphisms f and g as in (35).

$$A \xleftarrow{f} C \xrightarrow{g} B \tag{35}$$

$\ulcorner$ **Definition 156** (Pushout)**.** Let $A \xleftarrow{f} C \xrightarrow{g} B$ form a span in $\mathbf{C}$. Its **pushout** is an object, denoted $A +_C B$, along with morphisms $k_A : A \to A +_C B$ and $k_B : B \to A +_C B$ such that $k_A \circ f = k_B \circ g$ and the following universal property holds: for any object X and morphisms $s : A \to X$ and $t : B \to X$ satisfying $s \circ f = t \circ g$, there is a unique morphism $! : A +_C B \to X$ making (36) commute.[134]

$\ulcorner$ [134] The $\ulcorner$ symbol is a standard convention to specify that the square is not only commutative, but also a pushout square.

$$\begin{tikzcd}
C \arrow[r, "g"] \arrow[d, "f"'] & B \arrow[d, "k_B"] \arrow[rdd, bend left, "t"] & \\
A \arrow[r, "k_A"'] \arrow[rrd, bend right, "s"'] & A +_C B \arrow[ul, phantom, "\ulcorner", very near start] \arrow[rd, dashed, "!"] & \\
& & X
\end{tikzcd} \tag{36}$$

Example 157 (**Set**)**.** Let $A \xleftarrow{f} C \xrightarrow{g} B$ be a span in **Set** and suppose its pushout is $A \xrightarrow{k_A} A +_C B \xleftarrow{k_B} B$. Similarly to above, observe that k_A and k_B are like coprojections, so there is a unique map $! : A + B \to A +_C B$ such that $!(a) = k_A(a)$ and $!(b) = k_B(b)$. Furthermore, for any $c \in C$, $!(f(c)) =!(g(c))$, thus

$$\exists c \in C, f(c) = a \text{ and } g(c) = b \implies !(a) =!(b).$$

This is very similar to what happened for coequalizers and after working everything out, we obtain that $! : A + B \to A +_C B$ is the coequalizer of $\kappa_A \circ f$ and $\kappa_B \circ g$. This is a general fact that does not only apply in **Set** but in every category with binary coproducts and coequalizers.

As a particular case, if $C = A \cap B$ and f and g are simply inclusions, then $A +_C B = A \cup B$ (the *non-disjoint* union).

Generalization

In case you have not figured out the pattern, note that products, equalizers and pullbacks are examples of limits while coproducts, coequalizers and pushouts are examples of colimits. These six examples give quite a good idea of what it is to be a limit or colimit. Roughly, all of the definitions go as follows.

- Some shape is specified for a diagram D (i.e.: a discrete diagram, two parallel morphisms, a span, a cospan, etc.).
- The limit (resp. colimit) of D is an object L along with morphisms in $\mathrm{Hom}_{\mathbf{C}}(L, O)$ (resp. $\mathrm{Hom}_{\mathbf{C}}(O, L)$) for any object O in D such that combining D with these morphisms yields a commutative diagram.
- These morphisms satisfy a universal property. More specifically, for any object L' with morphisms in $\mathrm{Hom}_{\mathbf{C}}(L', O)$ (resp. $\mathrm{Hom}_{\mathbf{C}}(O, L')$) commuting with D, there is a unique $! : L' \to L$ (resp. $L \to L'$) such that combining all the morphisms with D yields a commutative diagram.

The first step towards a formal generalization is to formally define a diagram.

Definitions

Definition 158 (Diagram)**.** A **diagram** in $\mathbf{C}$ is a functor $F : \mathbf{D} \rightsquigarrow \mathbf{C}$ where $\mathbf{D}$ is usually a small or even finite category.

$$\begin{tikzcd} \cdot \arrow[r] \arrow[d] & \cdot \arrow[d] \\ \cdot \arrow[r] & \cdot \end{tikzcd} \tag{37}$$

Remark 159. Diagrams are usually represented by (partially) drawing the image of F. All the diagrams drawn up to this point define the domain of the functor implicitly. For instance, when considering a commutative square in $\mathbf{C}$, what is actually considered is the image from a functor with codomain $\mathbf{C}$ and domain the category $\mathbf{2} \times \mathbf{2}$ represented in (**??**). It follows trivially from this definition that functors preserve commutative diagrams.[135]

[135] If $F : \mathbf{D} \rightsquigarrow \mathbf{C}$ is a diagram of shape $\mathbf{D}$ in $\mathbf{C}$ and $G : \mathbf{C} \rightsquigarrow \mathbf{C}'$ is a functor, then $G \circ F$ is a diagram of shape $\mathbf{D}$ in $\mathbf{C}'$.

Next, notice that the morphisms given for L and L' have the same conditions, they form a cone or cocone.

Definition 160 (Cone)**.** Let $F : \mathbf{D} \rightsquigarrow \mathbf{C}$ be a diagram. A cone from X to F is an object $X \in \mathbf{C}_0$, called the **tip**, along with a family of morphisms $\{\psi_Y : X \to F(Y)\}$ indexed by objects $Y \in \mathbf{D}_0$ such that for any morphism $f : Y \to Z$ in $\mathbf{D}_1$, $F(f) \circ \psi_Y = \psi_Z$, i.e.: diagram (38) commutes.

$$\begin{tikzcd} & X \arrow[dl, "\psi_Y"'] \arrow[dr, "\psi_Z"] & \\ F(Y) \arrow[rr, "F(f)"'] & & F(Z) \end{tikzcd} \tag{38}$$

Often, the terminology cone over F is used.

Next, the fact that the morphism ! keeps everything commutative can be generalized.

Definition 161 (Morphism of cones)**.** Let $F : \mathbf{D} \rightsquigarrow \mathbf{C}$ be a diagram and $\{\psi_Y : A \to F(Y)\}_{Y \in \mathbf{D}_0}$ and $\{\phi_Y : B \to F(Y)\}_{Y \in \mathbf{D}_0}$ be two cones over F. A **morphism of cones** from A to B is a morphism $g : A \to B$ in $\mathbf{C}_1$ such that for any $Y \in \mathbf{D}_0$, $\phi_Y \circ g = \psi_Y$, i.e.: (39) commutes.

$$\begin{tikzcd} A \arrow[rr, "g"] \arrow[dr, "\psi_Y"'] & & B \arrow[dl, "\phi_Y"] \\ & F(Y) & \end{tikzcd} \tag{39}$$

After verifying that morphisms can be composed, the last two definitions give rise to the category of cones over a diagram F which we denote $\mathrm{Cone}(F)$. Finally, the universal property can be stated in terms of cones, thus giving the general definition of a limit. Indeed, the limit of a diagram D is a cone L over D such that for every cone L' over D, there is a unique cone morphism $! : L' \to L$. Equivalently, L is the terminal object of $\mathrm{Cone}(F)$.

Definition 162 (Limit)**.** Let $F : \mathbf{D} \rightsquigarrow \mathbf{C}$ be a diagram, the **limit** of F denoted $\lim F$ (or $\lim \mathbf{D}$), if it exists, is the terminal object of $\mathrm{Cone}(F)$.

Remark 163. Often, $\lim F$ also designates the tip of the cone as an object in $\mathbf{C}$ rather than the whole cone.

Examples 164. While you can play around with the three examples of limits we have already given and make them fit in this general definition, we add to this list a trivial example and a more complex one.

1. Consider an empty diagram in $\mathbf{C}$, that is, the only functor $\varnothing$ from the empty category to $\mathbf{C}$. A cone from X to $\varnothing$ is just an object $X \in \mathbf{C}_0$ as there are no objects in the diagram. Consequently, a morphism in $\mathrm{Cone}(\varnothing)$ is simply a morphism in $\mathbf{C}$, so $\mathrm{Cone}(\varnothing)$ is the same as the original category $\mathbf{C}$ and $\lim \varnothing$ is the terminal object of $\mathbf{C}$ if it exists.[136]

2. Let $X = \{x_1, \dots, x_n\}$ be a set of indeterminates (also called variables) and k be a field, $k[X]$ denotes the ring of polynomials over X.[137] We will construct $k[\![X]\!]$, the ring of formal power series over X, using limits.

   Let $I = \langle X \rangle$ be the ideal generated by X, the following three key properties are satisfied.

   a) For any $n < m \in \mathbb{N}$ and $p \in k[X]/I^m$, forgetting about all terms in p of degree at least n yields a ring homomorphism $\pi_{m,n} : k[X]/I^m \to k[X]/I^n$.

   b) For any $n \in \mathbb{N}$, we can do the same thing for power series to obtain a homomorphism $\pi_{\infty,n} : k[\![X]\!] \to k[X]/I^n$.

   c) Any composition of the homomorphisms above can be seen as a single homomorphism. Namely, $\forall n < m < l \in \mathbb{N} \cup \infty$,

   $$\pi_{m,n} \circ \pi_{l,m} = \pi_{l,n}.$$

   Consider the posetal category $(\mathbb{N}, \geq)$, a) and c) imply that $F(n) := k[X]/I^n$ and $F(m > n) := \pi_{m,n}$ defines a functor $F : (\mathbb{N}, \geq) \to \mathbf{Ring}$. This is represented in (40).

   $$\cdots \longrightarrow k[X]/I^n \xrightarrow{\pi_{n,n-1}} \cdots \longrightarrow k[X]/I^2 \xrightarrow{\pi_{2,1}} k[X]/I \xrightarrow{\pi_{1,0}} k[X] \tag{40}$$

   Now, using b) and c), we see that $k[\![X]\!]$ along with $\{\pi_{\infty,n}\}_{n\in\mathbb{N}}$ is a cone over the diagram F. It is in fact the terminal cone. Let $\{p_n : R \to k[X]/I^n\}$ be another cone over F and $! : R \to k[\![X]\!]$ a morphism of cones. By commutativity, the coefficients of $!(r)$ must agree with $p_n(r)$ on all monomials of degree at most n, thus,

   $$!(r) = p_0(r) + \sum_{n>0} p_n(r) - p_{n-1}(r).$$

   This completely determines $!$, so it is unique.[138]

   The construction of this diagram from quotienting different powers of the same ideal is used in different contexts, it is called the **completion** of $k[X]$ with respect to I. For instance, one can define the p–adic integers with base ring $\mathbb{Z}$ and the ideal generated by p for any prime p.

[136] Dually, $\mathrm{colim}\varnothing$ is the is the initial object of $\mathbf{C}$ if it exists (colim is defined in the next section).

[137] While, we will describe a nice categorical definition of $k[X]$ in Chapter , let us assume we know what it is.

[138] Existence follows from the same equation.

Codefinitions

Put simply, a colimit in $\mathbf{C}$ is a limit in $\mathbf{C}^{\mathrm{op}}$. We suggests you spend a bit of time trying to dualize all of the previous section on your own, but we have done it for completeness.

Definition 165 (Cocone)**.** Let $F : \mathbf{D} \rightsquigarrow \mathbf{C}$ be a diagram. A **cocone** from F to X is an object $X \in \mathbf{C}_0$ along with a family of morphisms $\{\psi_Y : F(Y) \to X\}$ indexed by objects of $Y \in \mathbf{D}_0$ such that for any morphism $f : Y \to Z$ in $\mathbf{D}$, $\psi_Z \circ F(f) = \psi_Y$, i.e.: (41) commutes.

$$\begin{tikzcd} F(Y) \arrow[rr, "F(f)"] \arrow[dr, "\psi_Y"'] & & F(Z) \arrow[dl, "\psi_Z"] \\ & X & \end{tikzcd} \tag{41}$$

Definition 166 (Morphism of cocones)**.** Let $F : \mathbf{D} \rightsquigarrow \mathbf{C}$ be a diagram and $\{\psi_Y : F(Y) \to A\}_{Y \in \mathbf{D}_0}$ and $\{\phi_Y : F(Y) \to B\}_{Y \in \mathbf{D}_0}$ be two cocones. A **morphism of cocones** from A to B is a morphism $g : A \to B$ in $\mathbf{C}$ such that for any $Y \in \mathbf{D}_0$, $g \circ \psi_Y = \phi_Y$, i.e.: (42) commutes.

$$\begin{tikzcd} & F(Y) \arrow[dl, "\psi_Y"'] \arrow[dr, "\phi_Y"] & \\ A \arrow[rr, "g"] & & B \end{tikzcd} \tag{42}$$

The category of cocones from F, sometimes called cones under F, is denoted $\mathrm{Cocone}(F)$.

Definition 167 (Colimit)**.** Let $F : \mathbf{D} \rightsquigarrow \mathbf{C}$ be a diagram, the colimit of F denoted $\mathrm{colim} F$, if it exists, is the initial object of $\mathrm{Cocone}(F)$.

Example 168. The colimit of the empty diagram is the initial object if it exists.

Result

Proposition 169 (Uniqueness)**.** *Let $F : \mathbf{D} \rightsquigarrow \mathbf{C}$ be a diagram, the limit (resp. colimit) of F, if it exists, is unique up to unique isomorphism.*

Proof. This follows from the uniqueness of terminal (resp. initial) objects. □

Remark 170*.* The isomorphism between two limits (also colimits) is unique when viewed as a morphism of cone. There might exists an isomorphism between the tips that is not a morphism of cone. For instance, let A, B and C be finite sets. One can check that both $A \times (B \times C)$ and $(A \times B) \times C$ are products of $\{A, B, C\}$ (with the usual projection maps). Thus, there is an isomorphism between them. One can check that, for it to be a morphism of cones, it must send $(a, (b, c))$ to $((a, b), c)$, but any other bijection between them is an isomorphism in **Set**.

For this reason, the limit really consists of the whole cone, and not just of the object at the tip! Unfortunately, this subtlety is not well cared for in the literature and it can and has led to errors.

Diagram chasing

We show four results in increasing order of complexity to demonstrate diagram chasing through examples.

Theorem 171. *Consider the pullback square in* (43)*.*

$$\begin{tikzcd} A \times_C B \arrow[r, "p_B"] \arrow[d, "p_A"'] \arrow[dr, phantom, "\lrcorner", very near start] & B \arrow[d, "g"] \\ A \arrow[r, "f"'] & C \end{tikzcd} \tag{43}$$

If g is monic, then p_A also is. Symmetrically, if f is monic, then p_B also is.[139]

[139] This is commonly stated simply as: "The pullback of a monomorphism is a monomorphism."

Proof. Let $h_1, h_2 : X \to A \times_C B$ be such that $p_A \circ h_1 = p_A \circ h_2$, we need to show that $h_1 = h_2$. First, observe that h_1 and h_2 yield two cones over the cospan $A \xrightarrow{f} C \xleftarrow{g} B$ as depicted in (44).

The two cones are

$$\begin{tikzcd} X \arrow[r, "p_B \circ h_1"] \arrow[d, "p_A \circ h_1"'] & B \\ A & \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} X \arrow[r, "p_B \circ h_2"] \arrow[d, "p_A \circ h_2"'] & B \\ A & \end{tikzcd}$$

They make the squares commute because the original pullback square commutes.

$$\begin{tikzcd} X \arrow[drr, bend left, "p_B \circ h_2"] \arrow[drr, bend left=15, "p_B \circ h_1"] \arrow[dr, shift left, "h_2"] \arrow[dr, shift right, "h_1"'] \arrow[ddr, bend right, "p_A \circ h_1 = p_A \circ h_2"'] & & \\ & A \times_C B \arrow[r, "p_B"] \arrow[d, "p_A"'] \arrow[dr, phantom, "\lrcorner", very near start] & B \arrow[d, "g"] \\ & A \arrow[r, "f"'] & C \end{tikzcd} \tag{44}$$

Furthermore, h_1 and h_2 are cone morphisms between X and $A \times_C B$ and since the pullback is the terminal cone over this cospan, they are unique. Now, we already have that the projections onto A is the same for both new cones, but we claim this is also true for the projections onto B. Indeed, because g is monic and the square commutes, we have the following implications.

$$\begin{aligned} p_A \circ h_1 = p_A \circ h_2 &\implies & f \circ p_A \circ h_1 = f \circ p_A \circ h_2 \\ &\implies & g \circ p_B \circ h_1 = g \circ p_B \circ h_2 \\ &\implies & p_B \circ h_1 = p_B \circ h_2 \end{aligned}$$

In other words, the two new cones are in fact the same cones, hence h_1 and h_2 are the same morphisms by uniqueness, which concludes our proof. □

Corollary 172. *The pushout of an epimorphism is an epimorphism.*

Theorem 173 (Pasting Lemma)**.** *Consider diagram* (45)*, where the right square is a pullback.*

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "\alpha"'] & B \arrow[r, "g"] \arrow[d, "\beta"'] \arrow[dr, phantom, "\lrcorner", very near start] & C \arrow[d, "\gamma"] \\ A' \arrow[r, "f'"'] & B' \arrow[r, "g'"'] & C' \end{tikzcd} \tag{45}$$

If (45) *commutes, the left square is a pullback if and only if the rectangle is.*

Proof. ($\Rightarrow$) Explicitly, we have to show that $\alpha : A' \leftarrow A \rightarrow C : g \circ f$ is the pullback of $g' \circ f' : A' \rightarrow C' \leftarrow C : \gamma$. The commutativity $g' \circ f' \circ \alpha = \gamma \circ g \circ f$ implies this is already a cone over the cospan we just described. Now, suppose there is another cone over this cospan, namely, there exist morphisms $p_{A'} : X \rightarrow A'$ and $p_C : X \rightarrow C$ satisfying $g' \circ f' \circ p_{A'} = \gamma \circ p_C$ as depicted in (46).

$$\begin{tikzcd}
X \arrow[drrr, bend left, "p_C"] \arrow[drr, dashed, "!_B"] \arrow[dr, dashed, "!_A"'] \arrow[ddr, bend right, "p_{A'}"'] & & & \\
& A \arrow[r, "f"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "\lrcorner"] & B \arrow[r, "g"] \arrow[d, "\beta"'] \arrow[dr, phantom, "\lrcorner"] & C \arrow[d, "\gamma"] \\
& A' \arrow[r, "f'"'] & B' \arrow[r, "g'"'] & C'
\end{tikzcd} \tag{46}$$

Notice that composing $p_{A'}$ with f', we obtain a cone over the cospan in the right square and by universality of B, this yields a unique morphism $!_B : X \rightarrow B$ satisfying $g \circ !_B = p_C$ and $\beta \circ !_B = f' \circ p_{A'}$. This second equality yields cone over the cospan in the left square, thus we get a unique morphism $!_A : X \rightarrow A$ satisfying $\alpha \circ !_A = p_{A'}$ and $f \circ !_A = !_B$. Composing the last equality with g, we get

$$g \circ f \circ !_A = g \circ !_B = p_C,$$

showing that $!_A$ is a morphism of cones over the rectangular cospan.

What is more, any other morphism $m : X \rightarrow A$ of cones over this cospan must satisfy

$$g \circ f \circ m = p_C \text{ and } \beta \circ f \circ m = f' \circ \alpha \circ m = f' \circ p_{A'},$$

and thus, $f \circ m$ is a morphism of cones over the cospan in the right rectangle. By uniqueness, $f \circ m = !_B$, so m is also a morphism of cones over the cospan in the left square, and by universality of A, $m = !_A$.

($\Leftarrow$) Explicitly, we have to show that $\alpha : A' \leftarrow A \rightarrow B : f$ is the pullback of $f' : A' \rightarrow B \leftarrow B : \beta$.

$$\begin{tikzcd}
X \arrow[drr, bend left, "p_B"] \arrow[dr, dashed, "!_A"'] \arrow[ddr, bend right, "p_{A'}"'] & & & \\
& A \arrow[r, "f"] \arrow[d, "\alpha"'] & B \arrow[r, "g"] \arrow[d, "\beta"'] \arrow[dr, phantom, "\lrcorner"] & C \arrow[d, "\gamma"] \\
& A' \arrow[r, "f'"'] & B' \arrow[r, "g'"'] & C'
\end{tikzcd} \tag{47}$$

Let $p_{A'} : A' \leftarrow X \rightarrow B : p_B$ be a cone over the cospan of the left square (i.e.: $\beta \circ p_B = f' \circ p_{A'}$). The commutativity of (45) implies $p_{A'} : A' \leftarrow X \rightarrow C : g \circ p_B$ is a cone over the rectangle cospan, then by universality of A, there exists a unique $!_A : X \rightarrow A$ such that $g \circ f \circ !_A = g \circ p_B$ and $\alpha \circ !_A = p_A$. Moreover, with the commutativity of the left square, we find that $f \circ !_A$ is a morphism of cones over the right cospan satisfying $\beta \circ f \circ !_A = f' \circ \alpha \circ !_A = f' \circ p_{A'} = \beta \circ p_B$ and $g \circ f \circ !_A = g \circ p_B$. But since our hypothesis on $p_{A'}$ and p_B implies p_B is a morphism of cones satisfying the same equations, by universality of B, $p_B = f \circ !_A$. Therefore, $!_A$ is a morphism of cone over the left cospan.

Finally, if $m : X \to A$ also satisfies $\alpha \circ m = p_{A'}$ and $f \circ m = p_B$. We find in particular that m is a morphism of cones over the rectangle cospan, hence by universality of A, $m = !_A$. □

Corollary 174. *In diagram (45) where the right square is not necessarily a pullback but the left square is a pushout, the right square is a pushout if and only if the rectangle is.*

Definition 175 ((Co)completeness)**.** A category is said to be **(co)complete** (resp. **finitely** (co)complete) if any small (resp. finite) diagram has a (co)limit.

Theorem 176. *Suppose that a category* $\mathbf{C}$ *has all products and equalizers then* $\mathbf{C}$ *has all limits, i.e.:* $\mathbf{C}$ *is complete.*

Proof. Let $F : J \rightsquigarrow \mathbf{C}$ be a diagram, we will show that the limit of F is obtained from the equalizer of two morphisms[140]

$$u_1, u_2 : \prod_{X \in J_0} F(X) \to \prod_{a \in J_1} F(t(a)),$$

which are defined below. The equalizer and the products it involves exist by hypothesis.

Recall that for any $X \in J_0$ and $a \in J_1$, we have two canonical projections

$$\pi_X : \prod_{X \in J_0} F(X) \to F(X) \quad \text{and} \quad \pi_a : \prod_{a \in J_1} F(t(a)) \to F(t(a)).$$

The first family of projections makes $\prod_{X \in J_0}$ into a cone over $\{F(t(a)) \mid a \in J_1\}$ with projections $\pi_{t(a)}$. Hence, there is a unique morphism $u_1 : \prod_{X \in J_0} F(X) \to \prod_{a \in J_1} F(t(a))$ that satisfies $\pi_a \circ u_1 = \pi_{t(a)}$. What is more, there is another way to project from $\prod_{X \in J_0}$ to $F(t(a))$, namely, via $F(a) \circ \pi_{s(a)}$, thus we get a unique morphism $u_2 : \prod_{X \in J_0} F(X) \to \prod_{a \in J_1} F(t(a))$ that satisfies $\pi_a \circ u_2 = F(a) \circ \pi_{s(a)}$. The situation is summarized in (48).

$$\begin{tikzcd}
\prod_{X \in J_0} F(X) \arrow[rr, equal] \arrow[dr, dashed, "u_1"] \arrow[ddr, bend right, "\pi_{t(a)}"'] & & \prod_{X \in J_0} F(X) \arrow[dl, dashed, "u_2"'] \arrow[ddl, bend left, "F(a) \circ \pi_{s(a)}"] \\
& \prod_{a \in J_1} F(t(a)) \arrow[d, "\pi_a"] & \\
& F(t(a)) &
\end{tikzcd} \tag{48}$$

Let $e : E \to \prod_{X \in J_0} F(X)$ be the equalizer of u_1 and u_2 and for any $X \in J_0$, let $\psi_X = \pi_X \circ e$. For any $f : Y \to Z$ in J, we have

$$\begin{aligned}
F(f) \circ \psi_Y &= F(f) \circ \pi_Y \circ e && \text{(def. of } \psi_Y\text{)} \\
&= \pi_f \circ u_2 \circ e && \text{(def. of } u_2\text{)} \\
&= \pi_f \circ u_1 \circ e && \text{(def. of } e\text{)}
\end{aligned}$$

[140] Recall that s and t denote the sources and targets of morphisms.

$$= \pi_Z \circ e = \psi_Z, \qquad \text{(def. of } u_1 \text{ and } \psi_Z\text{)}$$

so we indeed obtain a cone from E to F, depicted in (49).

$$\begin{tikzcd} & E \arrow[dl, "\pi_X \circ e"'] \arrow[dr, "\pi_Y \circ e"] & \\ F(X) \arrow[rr, "F(f)"'] & & F(Y) \end{tikzcd} \tag{49}$$

Next, any other cone $\{U_X : O \to F(X)\}_{X \in J_0}$ over F can also be viewed as a cone over the discrete diagram $\{F(t(a))\}_{a \in J_1}$ with projections $\{U_{t(a)}\}_{a \in J_1}$. Moreover, the universality of the product yields a unique morphism $p : O \to \prod_{X \in J_0} F(X)$ such that $\pi_X \circ p = U_X$. We claim that both $u_1 \circ p$ and $u_2 \circ p$ make (50) commute for all $a \in J_1$.

$$\begin{tikzcd} O \arrow[r, "p"] \arrow[drr, "U_{t(a)}"'] & \prod_{X \in J_0} F(X) \arrow[r, "u_i"] & \prod_{a \in J_1} F(t(a)) \arrow[d, "\pi_a"] \\ & & F(t(a)) \end{tikzcd} \tag{50}$$

This follows from two simple derivations.

$$\begin{aligned} \pi_a \circ u_1 \circ p &= \pi_{t(a)} \circ p \\ &= U_{t(a)} \end{aligned} \qquad\qquad \begin{aligned} \pi_a \circ u_2 \circ p &= F(a) \circ \pi_{s(a)} \circ p \\ &= F(a) \circ U_{s(a)} \\ &= U_{t(a)} \end{aligned}$$

Hence, $u_1 \circ p = u_2 \circ p$ as they are both morphisms of cone to the terminal cone $\prod_{a \in J_1} F(t(a))$. Now, by universality of the equalizer, we get a unique morphism $n : O \to E$ such that $e \circ n = p$. Furthermore, for any $X \in J_0$, we have

$$\psi_X \circ n = \pi_X \circ e \circ n = \pi_X \circ p = U_X,$$

so n is also a morphism of cones $(O, U_X) \to (E, \psi_X)$. Since any other morphism of cones m needs to satisfy $e \circ m = p$, we see that n is unique and conclude that E is $\lim F$.

Just for fun, here is what the whole diagram would look like if it were drawn at once (on the board or on paper).

$$\begin{tikzcd} & E \arrow[dl, "e"'] \arrow[dr, "e"] \arrow[rrr, dashed, "n"] & & & O \arrow[dllll, dashed, "p"'] \arrow[dll, dashed, "p"'] \arrow[dl, "U_X"] \arrow[dddlll, bend left, "U_{t(a)}"] \\ \prod_{X \in J_0} F(X) \arrow[rr, equal] \arrow[dr, dashed, "u_1"] \arrow[ddr, bend right, "\pi_{t(a)}"'] & & \prod_{X \in J_0} F(X) \arrow[r, "\pi_X"] \arrow[dl, dashed, "u_2"'] \arrow[ddl, bend left, "F(a) \circ \pi_{s(a)}"] & F(X) & \\ & \prod_{a \in J_1} F(t(a)) \arrow[d, "\pi_a"] & & & \\ & F(t(a)) & & & \end{tikzcd}$$

□

Remark 177. The same proof yields a more general statement: For any cardinal κ, if a category $\mathbf{C}$ has all products of size less than κ and equalizers, then it has limits of any diagram with less than κ objects and morphisms.

Definition 178. A functor $\mathbf{C} \rightsquigarrow \mathbf{D}$ is said to be (**finitely**) **(co)continuous** if it preserves all (finite) (co)limit.

Exercise 179. Show that a category with all pullbacks and a terminal object is finitely complete.

See solution.

Universal Properties

Free Monoid

The construction of a *free* object is common to different fields of mathematics and the example we will carry out in **Mon** can be carried out in many other categories like **Grp**, **Ab**, **Ring**, $\mathbf{Mod}_R$ (we will do this one in the next section). In fact, one way to view this construction comes from the forgetful functor to **Set** that all these categories have in common. In Chapter , we will cover adjoints and recover the free constructions from U.

We choose **Mon** because the concrete characterization of a free monoid is the simplest.

Definition 180 (Classical)**.** A monoid M is said to be **free** if it can be presented by a set of generators without any relations, i.e. $M = \langle A \mid \varnothing \rangle$. In this case, M is called the **free monoid on** A and denoted A^*.

It is easy to check that A^* is the set of finite words with symbols in A with the operation being concatenation and identity being the empty word (denoted **"**). In order to give a categorical characterization, we need to look at homomorphisms from or into the free monoid. Notice that any homomorphism $h^* : A^* \to M$ is completely determined by where h^* sends elements of A. Indeed, in order to satisfy the homomorphism property, we must have for any $a_1, a_2 \in A$,

$$h^*(a_1 a_2) = h^*(a_1) \cdot h^*(a_2) \text{ and } h^*(\varepsilon) = 1_M.$$

In general, the unique homomorphism sending $a \in A$ to $h(a)$ can be defined recursively:

$$h^*(w) = \begin{cases} h(a) \cdot h^*(w') & a \in A, w \in A^*, w = aw' \\ 1_M & w = \varepsilon \end{cases}.$$

Now, suppose that a monoid N contains A and satisfies the same property, that is for any (set-theoretic) function $h : A \to M$, there is a unique homomorphism $h^* : N \to M$ with $h^*(a) = h(a)$.

If we take $M = A^*$, and $h : A \to A^* = a \mapsto a$, then we get a homomorphism $h^*_N : N \to A^*$. Moreover, taking $M = N$ and $i : A \hookrightarrow N$ be the inclusion, the property of A^* means there is a unique homomorphism $i^* : A^* \to N$. Note that $h^*_N \circ i^* : A^* \to A^*$ is a homomorphism satisfying $a \mapsto a$, so it must be the identity by uniqueness. We conclude that N and A^* are isomorphic.

Definition 181 (Categorical)**.** The free monoid of a set A is an object A^* in **Mon** along with a *canonical inclusion* $i : A \to U(A^*)$ that satisfies the following universal property: for any monoid M and function $h : A \to U(M)$, there exists a unique homomorphism $h^* : A^* \to M$ such that $U(h^*) \circ i = h$, namely, $h^*(i(a)) = h(a)$. This is summarized in (51), where we omit the U as the underlying set of a monoid is often denoted with the same symbol as the monoid.

$$\begin{array}{ccc} \text{in } \mathbf{Set} & & \text{in } \mathbf{Mon} \\ \begin{tikzcd} A \arrow[r, "i"] \arrow[dr, "h"'] & A^* \arrow[d, dashed, "h^*"] \\ & M \end{tikzcd} & \leftsquigarrow & \begin{tikzcd} A^* \arrow[d, dashed, "h^*"] \\ M \end{tikzcd} \end{array} \tag{51}$$

Abelianization

Definition 182 (Classical)**.** Let G be a group, the **abelianization** of G, denoted G^{ab}, is the quotient of G with $G' := \{xyx^{-1}y^{-1} \mid x, y \in G\} \leq G$, called the *commutator subgroup*, that is $G^{\mathrm{ab}} := G/G'$.

Let us get insight into this definition. The abelianization is supposed to be the *biggest* abelian quotient of G. To see why, note that if A is an abelian group, any homomorphism $h : G \to A$ must satisfy $h(xyx^{-1}y^{-1}) = 1_A$ for any $x, y \in G$. Hence, G' is contained in the kernel of h. This yields a factorization $h = G \xrightarrow{\pi} G/G' \xrightarrow{h^*} A$ with h^* unique, where π is the canonical quotient map.

Moreover, since **Ab** is a full subcategory of **Grp**, h^* is also unique as a morphism in **Ab**. Using the fact that G/G' is abelian, we conclude the following categorical definition of G^{ab}.

Definition 183 (Categorical)**.** Let G be a group, the abelianization of G is an abelian group G^{ab} with a map $\pi : G \to G^{\mathrm{ab}}$ satisfying the following universal property: for any homomorphism $h : G \to A$ where A is abelian, there is a unique homomorphism $h^* : G^{\mathrm{ab}} \to A$ such that $h^* \circ \pi = h$. This is summarized in (52).

$$\begin{array}{ccc} \text{in } \mathbf{Grp} & & \text{in } \mathbf{Ab} \\ \begin{tikzcd} G \arrow[r, "\pi"] \arrow[dr, "h"'] & G^{\mathrm{ab}} \arrow[d, dashed, "h^*"] \\ & A \end{tikzcd} & \leftsquigarrow & \begin{tikzcd} G^{\mathrm{ab}} \arrow[d, dashed, "h^*"] \\ A \end{tikzcd} \end{array} \tag{52}$$

Vector Space Basis

Definition 184 (Classical)**.** Let V be a vector space over a field k, a **basis** for V is a subset $S \subseteq V$ that is linearly independent and generates V, namely, any $v \in V$ can be expressed as a linear combination of elements in S and any $s \in S$ cannot be expressed as a linear combination of elements in $S \setminus \{s\}$.

Once again, we would like to get rid of the content of this definition talking about elements, so we focus on what this means for linear maps coming out of V. Let S be a basis of V, W be another vector space over k and $T : V \to W$ be a linear map. By linearity, T is completely determined by where it sends the elements of S. Indeed, for any $v \in V$, write v as a linear combination $\sum_{s \in S} \lambda_s s$ with $\lambda_s \in k$ (only finitely many of the coefficients are non-zero), then $T(v) = \sum_{s \in S} \lambda_s T(s)$. We conclude that any (set-theoretic) function $t : S \to W$ extends to a unique linear map $T : V \to W$.

We claim that this property completely characterizes bases of V. Indeed, let $S \subseteq V$ be such that for any $t : S \to W$, there is a unique linear map $T : V \to W$ extending t. We will show that S is generating and linearly independent.

1. Assume towards a contradiction that S is not generating, that is, there exists $v \in V$ that is not a linear combination of vectors in S. Equivalently, if U is the subspace generated by S, then V/U is not 0. Now, let $t : S \to V/U$ be the 0 map, both the quotient map $\pi : V \to V/U$ and the 0 map $0 : V \to V/U$ extend t, and since V/U is not trivial, they are different maps.

2. Assume towards a contradiction that S is not linearly dependent, that is, there exists $v \in S$ is such that $v = \sum_{s \in S - v} \lambda_s s$. Consider the function

$$t : S \to V \oplus V = \begin{cases} (s, 0) & s \neq v \\ (0, v) & s = v \end{cases}.$$

There cannot exist a linear map $T : V \to V \oplus V$ extending t because by linearity, we can show

$$(0, v) = t(v) = T(v) = T(\sum_{s \in S - v} \lambda_s s) = \sum_{s \in S - v} \lambda_s T(s) = \sum_{s \in S - v} \lambda_s (s, 0),$$

which is absurd.

In conclusion, we have the following alternate definition of a vector space basis.

Definition 185 (Categorical)**.** Let V be a vector space, a basis of V is a set S along with an inclusion $i : S \to V$ satisfying the following universal property: for any function $t : S \to W$ where W is a vector space, there is a unique linear map $T : V \to W$ such that $T \circ i = t$. This is summarized in (53).

$$\begin{tikzcd} & \text{in } \mathbf{Set} & & \text{in } \mathbf{Vect}_k \\ S \arrow[r, "i"] \arrow[dr, "t"'] & V \arrow[d, dashed, "T"] & & V \arrow[d, dashed, "T"] \arrow[ll, squiggly, shift left=0, phantom] \\ & W & & W \end{tikzcd} \tag{53}$$

Exponential Objects

Let A and X be sets and denote A^X the set of functions $X \to A$. In hope to generalize this construction to other categories, let us study morphisms into A^X. Given a set B and a morphism $f : B \to A^X$, there is a natural operation called **uncurrying** that takes f to $\lambda^{-1} f : B \times X \to A$ which basically evaluates both f and its output at the same time. Namely, $\lambda^{-1} f(b, x) = f(b)(x)$.

As a particular case, we consider the identity function $A^X \to A^X$. Uncurrying yields the **evaluation** function $\mathsf{ev} : A^X \times X \to A$ that evaluates the function in the first coordinate at the second coordinate: $\mathsf{ev}(f, x) = f(x)$.

Now, as the name suggests, uncurrying has an inverse operation called **currying** which takes $g : B \times X \to A$ to $\lambda g : B \to A^X$. Morally, λg delays the evaluation of g to later.[141] Moreover, notice that if we are given $b \in B$ and $x \in X$, then we obtain an element of $\mathsf{ev}(\lambda g(b), x) = g(b, x) \in A$. This along with the fact that currying and uncurrying are bijective operations leads to a universal property that ev satisfies. It is summarized in (54).

[141] For computer scientists, this is also related to the concept of *continuations*.

$$\text{in } \mathbf{Set} \qquad\qquad\qquad\qquad \text{in } \mathbf{Set}$$
$$\begin{tikzcd} A & A^X \times X \arrow[l, "\mathsf{ev}"'] & & A^X \arrow[dl, squiggly] \\ & B \times X \arrow[ul, "g"] \arrow[u, dashed, "\lambda g \times \mathrm{id}_X"'] & & B \arrow[u, dashed, "\lambda g"'] \end{tikzcd} \tag{54}$$

This is entirely categorical, so we can define an *exponential object* in an arbitrary category $\mathbf{C}$ (with binary products) as an object A^X along with a morphism $\mathsf{ev} : A^X \times X \to A$ such that for all $g : B \times X \to A$, there is a unique $\lambda g : B \to A^X$ making (54) commute.

Generalization

Definition 186 (Comma category)**.** Given two functors $\mathbf{D} \overset{F}{\rightsquigarrow} \mathbf{C} \overset{G}{\leftsquigarrow} \mathbf{E}$, there is a category $F{\downarrow}G$, called the **comma category**, whose objects are triples (X, Y, α) with $X \in \mathbf{D}_0$, $Y \in \mathbf{E}_0$ and $\alpha : F(X) \to G(Y)$ (in $\mathbf{C}_1$) and morphisms between (X_1, Y_1, α) and (X_2, Y_2, β) are pairs of morphisms $(f, g) \in \mathrm{Hom}_{\mathbf{D}}(X_1, X_2) \times \mathrm{Hom}_{\mathbf{E}}(Y_1, Y_2)$ yielding a commutative square as in (55).

$$\begin{tikzcd} F(X_1) \arrow[r, "F(f)"] \arrow[d, "\alpha"'] & F(X_2) \arrow[d, "\beta"] \\ G(Y_1) \arrow[r, "G(g)"'] & G(Y_2) \end{tikzcd} \tag{55}$$

Definition 187 (Arrow category)**.** In the setting of Definition 186, if $F = G = \mathrm{id}_{\mathbf{C}}$, then $\mathrm{id}_{\mathbf{C}}{\downarrow}\mathrm{id}_{\mathbf{C}}$ is called the **arrow category** of $\mathbf{C}$ and denoted $\mathbf{C}^{\to}$. Its objects are morphisms in $\mathbf{C}$ and its morphisms are commutative squares in $\mathbf{C}$.

Definition 188 (Slice category)**.** In the setting of Definition 186, if $F = \mathrm{id}_{\mathbf{C}}$ and $G = X : \mathbf{1} \rightsquigarrow \mathbf{C}$ is a constant functor selecting one object $G(\bullet) = X \in \mathbf{C}_0$, then $\mathrm{id}_{\mathbf{C}}{\downarrow}X$ is called the **slice category** over X and denoted $\mathbf{C}/X$.[142] Its objects are morphisms in $\mathbf{C}$ with target X and its morphisms are commutative triangles with X as a tip as in (57).

[142] Some authors call this category $\mathbf{C}$ over X.

$$\begin{tikzcd} A \arrow[rr] \arrow[dr] & & B \arrow[dl] \\ & X & \end{tikzcd} \tag{56}$$

Definition 189 (Coslice category)**.** In the setting of Definition 186, if $G = \mathrm{id}_{\mathbf{C}}$ and $F = X : \mathbf{1} \rightsquigarrow \mathbf{C}$ is a constant functor selecting one object $F(\bullet) = X \in \mathbf{C}_0$, then $X{\downarrow}\mathrm{id}_{\mathbf{C}}$ is called the **coslice category** under X and denoted $X/\mathbf{C}$.[143] Its objects are morphisms in $\mathbf{C}$ with source X and its morphisms are commutative triangles with X as a tip as in (57).

$$\begin{tikzcd} & X \arrow[dl] \arrow[dr] & \\ A \arrow[rr] & & B \end{tikzcd} \tag{57}$$

[143] Some authors call this category $\mathbf{C}$ under X.

Exercise 190. Show that for any category $\mathbf{C}$ and object $X \in \mathbf{C}_0$, the slice category $\mathbf{C}/X$ has a terminal object. State and prove the dual statement.

See solution.

Back to universal properties.

Definition 191 (Universal morphism)**.** If $F : \mathbf{D} \rightsquigarrow \mathbf{C}$ is a functor and $X \in \mathbf{C}_0$. A **universal morphism** from X to F is an initial object in $X{\downarrow}F$. Namely, it is a morphism $a : X \to F(A)$ such that for any other morphism $b : X \to F(B)$, there is unique commutative triangle as in (58).

$$\begin{tikzcd} & X \arrow[dl, "a"'] \arrow[dr, "b"] & \\ F(A) \arrow[rr, dashed, "F(f)"'] & & F(B) \end{tikzcd} \tag{58}$$

Notice that equivalently, one could say that for any $b : X \to F(B)$, there is a unique morphism $f : A \to B$ in $\mathbf{D}$ such that $F(f) \circ a = b$, which is summarized in (59).

$$\begin{tikzcd} & \text{in } \mathbf{C} & & \text{in } \mathbf{D} \\ X \arrow[r, "a"] \arrow[dr, "b"'] & F(A) \arrow[d, dashed, "F(f)"] & & A \arrow[d, dashed, "f"] \arrow[ll, squiggly, shorten=1em] \\ & F(B) & & B \end{tikzcd} \tag{59}$$

The dual notion is a universal morphism from F to X, it is a terminal object in $F{\downarrow}X$. The dual of (59) is depicted below.

$$\begin{tikzcd} & \text{in } \mathbf{C} & & \text{in } \mathbf{D} \\ X & F(A) \arrow[l, "a"'] & & A \arrow[ll, squiggly, shorten=1em] \\ & F(B) \arrow[ul, "b"] \arrow[u, dashed, "F(f)"] & & B \arrow[u, dashed, "f"'] \end{tikzcd} \tag{60}$$

Definition 192 (Universal property)**.** A **universal property** is the property of being a universal morphism.

We will not bother applying this general definition anymore because the formalism is not crucial to the study of universal properties. Recall that we claimed that limits satisfied some universal properties, and indeed, you can show this very formally, but notice that our definition of universal property also uses a special case of limits, that is, initial and terminal objects. What is more, in the following chapters, we will introduce a couple more concepts which often coincide[144] with the concepts of limits and universal properties.

[144] By *coincide*, we mean that one is a special case of the other or vice-versa or both directions.

Natural Transformations

Natural Transformations

Natural transformations are admittedly what made mathematicians want to study category theory in the first place. In short, they are morphisms between functors, i.e.: transformations that preserve the structure of functors.

The abstract structure of a category is very familiar because it resembles what is found in algebraic structures such as groups, rings or vector spaces. That is to say, it consists of the data of one or more sets with one or more operations satisfying one or more properties. In contrast, the definition of a functor is more opaque and by itself, the structure of a functor is not obvious. A functor is effectively a morphism between categories, hence a natural transformation will be a *morphism between morphisms*. Before moving on, one might find it enlightening to look for a satisfying definition of morphism between two group homomorphisms $f, g : G \to H$ and then observe its meaning when f and g are seen as functors $\mathbf{B}(G) \rightsquigarrow \mathbf{B}(H)$.

For the general case, let $F, G : \mathbf{C} \rightsquigarrow \mathbf{D}$ be functors. Morally, the structure of F and G is encapsulated in the following diagrams for every arrow, $f \in \mathrm{Hom}_{\mathbf{C}}(A, B)$.

$$\begin{tikzcd} A \arrow[r, "F_0"] \arrow[d, "f"'] & F(A) \arrow[d, "F_1(f)"] \\ B \arrow[r, "F_0"'] & F(B) \end{tikzcd} \quad (61) \qquad \begin{tikzcd} A \arrow[r, "G_0"] \arrow[d, "f"'] & G(A) \arrow[d, "G_1(f)"] \\ B \arrow[r, "G_0"'] & G(B) \end{tikzcd} \quad (62)$$

Thus, a morphism between F and G should fit in this picture by sending diagram (61) to diagram (62) in a commutative way.

Definition 193 (Natural transformation)**.** Let $F, G : \mathbf{C} \rightsquigarrow \mathbf{D}$ be two (covariant) functors, a natural transformation $\phi : F \Rightarrow G$ is a map $\phi : \mathbf{C}_0 \to \mathbf{D}_1$ that satisfies $\phi(A) \in \mathrm{Hom}_{\mathbf{D}}(F(A), G(A))$ for all $A \in \mathbf{C}_0$ and makes diagram (63) commute for any $f \in \mathrm{Hom}_{\mathbf{C}}(A, B)$:

$$\begin{tikzcd} F(A) \arrow[r, "\phi(A)"] \arrow[d, "F(f)"'] & G(A) \arrow[d, "G(f)"] \\ F(B) \arrow[r, "\phi(B)"'] & G(B) \end{tikzcd} \qquad (63)$$

Each $\phi(A)$ will be called a **component** of ϕ and may also be denoted ϕ_A.

As usual, there are trivial examples of natural transformations such as the **identity transformation** $\mathbb{1}_F : F \Rightarrow F$ that sends every object A to the identity map $\mathrm{id}_{F(A)}$, but let us go back to the group case. Although very specific to single object categories, it is simple enough to quickly digest.

Example 194. Let $f, g : \mathbf{B}(G) \rightsquigarrow \mathbf{B}(H)$ be functors (i.e.: group homomorphisms), both send the unique object $*$ in $\mathbf{B}(G)$ to $*$ in $\mathbf{B}(H)$. Thus, a natural transformation $\phi : f \Rightarrow g$ has a single component $\phi(*) : * \to *$ in H, which is simply an element $\phi \in H$. The commutativity condition is then exhibited by diagram (64) (which lives in $\mathbf{B}(H)$) for any $x \in G$.

$$\begin{tikzcd} * \arrow[r, "\phi"] \arrow[d, "f(x)"'] & * \arrow[d, "g(x)"] \\ * \arrow[r, "\phi"'] & * \end{tikzcd} \tag{64}$$

Recall that composition in $\mathbf{B}(H)$ is just multiplication in H, so naturality of ϕ says that for any $x \in G$, $\phi \cdot f(x) = g(x) \cdot \phi$. Equivalently, $\phi f(x) \phi^{-1} = g(x)$. Therefore, $g = c_\phi \circ f$ where c_ϕ denotes conjugation by ϕ.[145] In short, natural transformations between group homomorphisms correspond to factorizations through conjugations.

[145] In a group $(H, \cdot)$, **conjugation** by an element $h \in H$ is the homomorphism c_h defined $x \mapsto hxh^{-1}$.

Next, an example closer to the general idea of a natural transformation.

Example 195. Fix some $n \in \mathbb{N}$ and define the functor $\mathrm{GL}_n : \mathbf{CRing} \rightsquigarrow \mathbf{Grp}$ by[146]

$$R \mapsto \mathrm{GL}_n(R) \text{ for any commutative ring } R \text{ and}$$
$$f \mapsto \mathrm{GL}_n(f) \text{ for any ring homomorphism } f.$$

[146] The map $\mathrm{GL}_n(f)$ is just the extension of f on $\mathrm{GL}_n(R)$ by applying f to every element of the matrices.

The second functor is $(-)^\times : \mathbf{CRing} \rightsquigarrow \mathbf{Grp}$ which sends a commutative ring R to its group of units $R^\times$ and a ring homomorphism f to $f^\times$, its restriction on $R^\times$. Checking these mappings define two covariant functors is left as an (simple) exercise, but one might expect these to be functors as they play nicely with the structure of the objects involved.

A natural transformation between these two functors is $\det : \mathrm{GL}_n \Rightarrow (-)^\times$ which maps a commutative ring R to $\det_R$, the function calculating the determinant of a matrix in $\mathrm{GL}_n(R)$. The first thing to check is that $\det_R \in \mathrm{Hom}_{\mathbf{Grp}}(\mathrm{GL}_n(R), R^\times)$ which is clear because the determinant of an invertible matrix is always a unit, $\det_R(I_n) = 1$ and $\det_R$ is a multiplicative map.[147] The second thing is to verify that diagram (65) commutes for any $f \in \mathrm{Hom}_{\mathbf{CRing}}(R, S)$:

[147] i.e.: $\det_R(AB) = \det_R(A)\det_R(B)$.

$$\begin{tikzcd} \mathrm{GL}_n(R) \arrow[r, "\det_R"] \arrow[d, "\mathrm{GL}_n(f)"'] & R^\times \arrow[d, "f^\times = f|_{R^\times}"] \\ \mathrm{GL}_n(S) \arrow[r, "\det_S"'] & S^\times \end{tikzcd} \tag{65}$$

We will check the claim for $n = 2$, but the general proof should only involve more notation to write the bigger expressions, no novel idea. Let $a, b, c, d \in R$, we have

$$(\det_S \circ \mathrm{GL}_2(f))\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = \det_S\left(\begin{bmatrix} f(a) & f(b) \\ f(c) & f(d) \end{bmatrix}\right)$$

$$\begin{aligned}
&= f(a)f(d) - f(b)f(c) \\
&= f(ad - bc) \\
&= f^{\times}(ad - bc) \\
&= (f^{\times} \circ \det_R)\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right).
\end{aligned}$$

We conclude that the diagram commutes and that det is indeed a natural transformation.[148]

[148] Modulo the cases $n > 2$. See solution.

Exercise 196. Let $F, G : \mathbf{C} \times \mathbf{C}' \rightsquigarrow \mathbf{D}$ be two functors. Show that a family

$$\{\phi_{X,Y} : F(X,Y) \to G(X,Y) \mid X \in \mathbf{C}_0, Y \in \mathbf{C}'_0\}$$

is a natural transformation if and only if for any $X \in \mathbf{C}_0$ and $Y \in \mathbf{C}'_0$, both

$$\phi_{X,-} : F(X,-) \Rightarrow G(X,-) \text{ and } \phi_{-,Y} : F(-,Y) \Rightarrow G(-,Y)$$

are natural.

Now, in order to talk about a category of functors, it remains to describe the composition of natural transformations.

Definition 197 (Vertical composition)**.** Let $F, G, H : \mathbf{C} \rightsquigarrow \mathbf{D}$ be parallel functors and $\phi : F \Rightarrow G$ and $\eta : G \Rightarrow H$ be two natural transformations. Then, the **vertical composition** of ϕ and η, denoted $\eta \cdot \phi : F \Rightarrow H$ is defined by $(\eta \cdot \phi)(A) = \eta(A) \circ \phi(A)$ for all $A \in \mathbf{C}_0$. If $f : A \to B$ is a morphism in $\mathbf{C}$, then diagram (66) commutes by naturality of ϕ and η, showing that $\eta \cdot \phi$ is a natural transformation from F to H.

The notation $\cdot$ is not widespread, most authors use $\circ$ because vertical composition is the composition in a functor category. We believe the distinction is helpful as you learn this material.

$$\begin{tikzcd}
F(A) \arrow[r, "\phi(A)"] \arrow[d, "F(f)"'] & G(A) \arrow[r, "\eta(A)"] \arrow[d, "G(f)"'] & H(A) \arrow[d, "H(f)"'] \\
F(B) \arrow[r, "\phi(B)"'] & G(B) \arrow[r, "\eta(B)"'] & H(B)
\end{tikzcd} \tag{66}$$

The meaning of *vertical* will come to light when horizontal composition is introduced in a bit.

Definition 198 (Functor categories)**.** For any two categories $\mathbf{C}$ and $\mathbf{D}$, there is a functor category denoted $[\mathbf{C}, \mathbf{D}]$.[149] Its objects are functors from $\mathbf{C}$ to $\mathbf{D}$, its morphisms are natural transformations between such functors and the composition is the vertical composition defined above. One can check that associativity of $\cdot$ follows from associativity of composition in $\mathbf{D}$ and that the identity morphism for a functor F is $\mathbb{1}_F$.

[149] Some authors denote it $\mathbf{D}^{\mathbf{C}}$, analogously to the exponential of sets.

Example 199. Recall that a left action of a group G on a set S is just a functor $\mathbf{B}(G) \rightsquigarrow \mathbf{Set}$. Now, between two such functors $F, F' \in [\mathbf{B}(G), \mathbf{Set}]$, a natural transformation is a single map $\sigma : F(*) \to F'(*)$ such that $\sigma \circ F(g) = F'(g) \circ \sigma$ for any $g \in G$. In other words, denoting $\cdot$ for both group actions on $F(*)$ and on $F'(*)$, σ

satisfies $\sigma(g \cdot x) = g \cdot (\sigma(x))$ for any $g \in G$ and $x \in F(*)$. In group theory, such a map is called G–**equivariant**.

Therefore, the category $[\mathbf{B}(G), \mathbf{Set}]$ can be identified as the category of G–sets (sets equipped with an action of G) with A–equivariant maps as the morphisms.

Exercise 200 (NOW!)**.** Isomorphisms in a functor category are called **natural isomorphisms**. Show that they are precisely the natural transformations whose components are all isomorphisms.

See solution.

Functors that are naturally isomorphic are essentially the same functor; they send the same object to isomorphic objects and the same morphism to morphisms that are well-behaved under composition with isomorphisms between the source and targets.

Examples 201. We can recover constructions we have seen before by studying categories of functors with a simple domain.

1. The terminal category $\mathbf{1}$ has a single object $\bullet$ and no morphism other than the identity. Notice that for any category $\mathbf{C}$, a functor $F : \mathbf{1} \rightsquigarrow \mathbf{C}$ is a simply a choice of object $F(\bullet) \in \mathbf{C}_0$ because $F(\mathrm{id}_\bullet) = \mathrm{id}_{F(\bullet)}$. If $F, G \in [\mathbf{1}, \mathbf{C}]$, then a natural transformation $\phi : F \Rightarrow G$ is simply a choice of morphism $\phi : F(\bullet) \rightsquigarrow G(\bullet)$ because naturality square (67) for the only morphism $\mathrm{id}_\bullet$ is trivially commutative. We conclude that $[\mathbf{1}, \mathbf{C}]$ can be identified with the category $\mathbf{C}$ itself.

$$\begin{tikzcd} F(\bullet) \arrow[r, "F(\mathrm{id}_\bullet)"] \arrow[d, "\phi"'] & F(\bullet) \arrow[d, "\phi"] \\ G(\bullet) \arrow[r, "G(\mathrm{id}_\bullet)"'] & G(\bullet) \end{tikzcd} \tag{67}$$

2. Similarly, we can see a functor $F : \mathbf{1} + \mathbf{1} \rightsquigarrow \mathbf{C}$[150] as a choice of two objects $F(\bullet_1)$ and $F(\bullet_2)$ and a natural transformation $\phi : F \Rightarrow G$ between two such functors as a choice of two morphisms $\phi_1 : F(\bullet_1) \to G(\bullet_1)$ and $\phi_2 : F(\bullet_2) \to G(\bullet_2)$. Therefore, we infer that $[\mathbf{1} + \mathbf{1}, \mathbf{C}]$ can be identified with $\mathbf{C} \times \mathbf{C}$.

[150] Recall $\mathbf{1} + \mathbf{1}$ is the category depicted in (4).

3. Let us go one level harder. A functor $F : \mathbf{2} \rightsquigarrow \mathbf{C}$[151] is a choice of two objects FA and FB as well as a morphism $Ff : FA \to FB$. It can also be seen as a single choice of morphism Ff because FA and FB are determined to be the source and target of Ff respectively. A natural transformation $\phi : F \Rightarrow G$ between two such functors is *not* simply a choice of two morphisms $\phi_A : FA \to GA$ and $\phi_B : FB \to GB$ because, while the naturality squares for id_A and id_B trivially commute, the naturality square (68) for f is an additional constraint on ϕ. Namely, it says (ϕ_A, ϕ_B) makes a commutative square with Ff and Gf, hence we can identify $[\mathbf{2}, \mathbf{C}]$ with the arrow category $\mathbf{C}^{\to}$.

[151] Recall $\mathbf{2}$ is the category depicted in (5).

$$\begin{tikzcd} FA \arrow[r, "Ff"] \arrow[d, "\phi_A"'] & FB \arrow[d, "\phi_B"] \\ GA \arrow[r, "Gf"'] & GB \end{tikzcd} \tag{68}$$

It is now time to build intuition for the horizontal composition of natural transformations which will ultimately lead to the notion of a 2–category.

Definition 202 (The left action of functors)**.** Let $F, F' : \mathbf{C} \rightsquigarrow \mathbf{D}$, $G : \mathbf{D} \rightsquigarrow \mathbf{D}'$ be functors and $\phi : F \Rightarrow F'$ a natural transformation as summarized in (69).[152]

$$\begin{tikzcd} \mathbf{C} \arrow[r, bend left, squiggly, "F"{name=U}] \arrow[r, bend right, squiggly, "F'"'{name=D}] & \mathbf{D} \arrow[r, "G"] & \mathbf{D}' \arrow[Rightarrow, from=U, to=D, "\phi"] \end{tikzcd} \tag{69}$$

[152] Using squiggly arrows for functors in diagrams is very non-standard, but I believe it helps remember what kind of objects we are dealing with. Moreover, since these diagrams are not commutative, it makes a good contrast with the plain arrow notation which was mostly used for commutative diagrams.

The functor G acts on ϕ by sending it to $G\phi := A \mapsto G(\phi(A)) : \mathbf{C}_0 \to \mathbf{D}'_1$. Showing that (70) commutes for any $f \in \mathrm{Hom}_{\mathbf{C}}(A, B)$ will imply that $G\phi$ is a natural transformation from $G \circ F$ to $G \circ F'$.

$$\begin{tikzcd}
(G \circ F)(A) \arrow[r, "G\phi(A)"] \arrow[d, "(G\circ F)(f)"'] & (G \circ F')(A) \arrow[d, "(G\circ F')(f)"] \\
(G \circ F)(B) \arrow[r, "G\phi(B)"'] & (G \circ F')(B)
\end{tikzcd} \tag{70}$$

Consider this diagram after removing all applications of G, by naturality of ϕ, it is commutative. Since functors preserve commutativity, the diagram still commutes after applying G, hence $G\phi : G \circ F \Rightarrow G \circ F'$ is indeed natural.

We leave you to check this constitutes a left action, namely, for any $G : \mathbf{D} \rightsquigarrow \mathbf{D}'$, $G' : \mathbf{D}' \rightsquigarrow \mathbf{D}''$ and $\phi : F \Rightarrow F'$,

$$\mathrm{id}_{\mathbf{D}}\phi = \phi \text{ and } G'(G\phi) = (G' \circ G)\phi.$$

Definition 203 (The right action of functors)**.** Let $F, F' : \mathbf{C} \rightsquigarrow \mathbf{D}$, $H : \mathbf{C}' \rightsquigarrow \mathbf{C}$ be functors and $\phi : F \Rightarrow F'$ a natural transformation as summarized in (71).

$$\begin{tikzcd}
\mathbf{C}' \arrow[r, squiggly, "H"] & \mathbf{C} \arrow[r, squiggly, bend left, "F"] \arrow[r, squiggly, bend right, "F'"'] \arrow[r, phantom, "\Downarrow\phi"] & \mathbf{D}
\end{tikzcd} \tag{71}$$

The functor H acts on ϕ by sending it to $\phi H := A \mapsto \phi(H(A)) : \mathbf{C}'_0 \to \mathbf{D}_1$. Showing that (72) commutes for any $f \in \mathrm{Hom}_{\mathbf{C}'}(A, B)$ will imply that ϕH is a natural transformation from $F \circ H$ to $F' \circ H$.

$$\begin{tikzcd}
(F \circ H)(A) \arrow[r, "\phi H(A)"] \arrow[d, "(F\circ H)(f)"'] & (F' \circ H)(A) \arrow[d, "(F'\circ H)(f)"] \\
(F \circ H)(B) \arrow[r, "\phi H(B)"'] & (F' \circ H)(B)
\end{tikzcd} \tag{72}$$

Commutativity of (72) follows by naturality of ϕ: change f in diagram (63) with the morphism $H(f) : H(A) \to H(B)$.

We leave you to check this constitutes a right action, namely, for any $H : \mathbf{C}' \rightsquigarrow \mathbf{C}$, $H' : \mathbf{C}'' \rightsquigarrow \mathbf{C}'$ and $\phi : F \Rightarrow F'$,

$$\phi\mathrm{id}_{\mathbf{C}} = \phi \text{ and } (\phi H)H' = \phi(H \circ H').$$

Proposition 204. *The two actions commute, i.e.: in the setting of (73), $G(\phi H) = (G\phi)H$.*[153]

$$\begin{tikzcd}
\mathbf{C}' \arrow[r, squiggly, "H"] & \mathbf{C} \arrow[r, squiggly, bend left, "F"] \arrow[r, squiggly, bend right, "F'"'] \arrow[r, phantom, "\Downarrow\phi"] & \mathbf{D} \arrow[r, "G"] & \mathbf{D}'
\end{tikzcd} \tag{73}$$

Proof. In both the L.H.S. and the R.H.S., an object $A \in \mathbf{C}_0$ is sent to $G(\phi(H(A)))$. □

[153] For this reason and the associativity of the two actions, we will drop all the parentheses from such expressions. We will also drop the $\circ$ for composition of functors. All in all, expect to find expressions like $G'G\phi HH'$ and infer the natural transformation $A \mapsto G'(G(\phi(H(H'(A)))))$.

We will refer to these two actions as the **biaction** of functors on natural transformations and they will motivate the definition of another way to compose natural transformations.

Let $\mathbf{C}$, $\mathbf{D}$ and $\mathbf{E}$ be categories, $H, H' : \mathbf{C} \rightsquigarrow \mathbf{D}$ and $G, G' : \mathbf{D} \rightsquigarrow \mathbf{E}$ be functors and $\phi : H \Rightarrow H'$ and $\eta : G \Rightarrow G'$ be natural transformations. This is summarized in (74).

$$\begin{tikzcd} \mathbf{C} \arrow[r, bend left, squiggly, "H"] \arrow[r, bend right, squiggly, "H'"'] \arrow[r, phantom, "\Downarrow\phi"] & \mathbf{D} \arrow[r, bend left, squiggly, "G"] \arrow[r, bend right, squiggly, "G'"'] \arrow[r, phantom, "\Downarrow\eta"] & \mathbf{E} \end{tikzcd} \tag{74}$$

The ultimate goal is to obtain a new composition of ϕ and η that is a natural transformation $G \circ H \Rightarrow G' \circ H'$. Note that the biaction defined above yields four other natural transformations:

$$\begin{aligned} G\phi &: G \circ H \Rightarrow G \circ H' & \eta H &: G \circ H \Rightarrow G' \circ H \\ G'\phi &: G' \circ H \Rightarrow G' \circ H' & \eta H' &: G \circ H' \Rightarrow G' \circ H'. \end{aligned}$$

All of the functors involved go from $\mathbf{C}$ to $\mathbf{E}$, so all four natural transformations fit in diagram (75) that lives in the functor category $[\mathbf{C}, \mathbf{E}]$.

$$\begin{tikzcd} G \circ H \arrow[r, "G\phi"] \arrow[d, "\eta H"'] & G \circ H' \arrow[d, "\eta H'"] \\ G' \circ H \arrow[r, "G'\phi"'] & G' \circ H' \end{tikzcd} \tag{75}$$

At first glance, this suggests two different definitions for the horizontal composition, that is, the composition of the top path $(\eta H' \cdot G\phi)$ or the composition of the bottom path $(G'\phi \cdot \eta H)$. Surprisingly, both definitions coincide as shown in the next result.

Lemma 205. *Diagram* (75) *commutes, i.e.:* $\eta H' \cdot G\phi = G'\phi \cdot \eta H$.

Proof. Fix an object $A \in \mathbf{C}_0$. Under $\eta H' \cdot G\phi$, it is sent to $\eta(H'(A)) \circ G(\phi(A))$ and under $G'\phi \cdot \eta H$, it is sent to $G'(\phi(A)) \circ \eta(H(A))$. Thus, the proposition is equivalent to saying diagram (76) is commutative (in $\mathbf{E}$) for all $A \in \mathbf{C}_0$.

$$\begin{tikzcd} (G \circ H)(A) \arrow[r, "G(\phi(A))"] \arrow[d, "\eta(H(A))"'] & (G \circ H')(A) \arrow[d, "\eta(H'(A))"] \\ (G' \circ H)(A) \arrow[r, "G'(\phi(A))"'] & (G' \circ H')(A) \end{tikzcd} \tag{76}$$

This follows from the naturality of η.[154] $\square$

Definition 206 (Horizontal composition)**.** In the setting described in (74), we define the **horizontal composition** of η and ϕ by $\eta \diamond \phi = \eta H' \cdot G\phi = G'\phi \cdot \eta H$.[155]

The most important part we expect from a notion of composition is associativity, so let us check $\diamond$ is associative.

154 In (63), replace A with $H(A)$, B with $H'(A)$, f with $\phi(A)$, F with G and G with G'.

155 The $\diamond$ notation is not standard but there are no widespread symbol denoting horizontal composition. I have mostly seen $*$ or plain juxtaposition. Hopefully, you will encounter papers/books clear enough that you can typecheck to find what composition is being used.

Proposition 207. *In the setting of* (77), $\psi \diamond (\eta \diamond \phi) = (\psi \diamond \eta) \diamond \phi$.

$$\begin{tikzcd} \mathbf{C} \arrow[r, squiggly, bend left, "H"{name=H}] \arrow[r, squiggly, bend right, "H'"'{name=Hp}] \arrow[r, phantom, "\Downarrow\phi"] & \mathbf{D} \arrow[r, squiggly, bend left, "G"] \arrow[r, squiggly, bend right, "G'"'] \arrow[r, phantom, "\Downarrow\eta"] & \mathbf{E} \arrow[r, squiggly, bend left, "K"] \arrow[r, squiggly, bend right, "K'"'] \arrow[r, phantom, "\Downarrow\psi"] & \mathbf{F} \end{tikzcd} \tag{77}$$

Proof. Similarly to how we constructed diagram (75) in $[\mathbf{C}, \mathbf{E}]$ previously, we can use the biaction of functors and composition of functors to obtain the following diagram in $[\mathbf{C}, \mathbf{E}]$.[156]

$$\begin{tikzcd} & K'GH \arrow[rr, "K'G\phi"] \arrow[dd, "K'\eta H"] & & K'GH' \arrow[dd, "K'\eta H'"] \\ KGH \arrow[ur, "\psi GH"] \arrow[rr, "KG\phi"] \arrow[dd, "K\eta H"'] & & KGH' \arrow[ur, "\psi GH'"'] \arrow[dd, "K\eta H'"] & \\ & K'G'H \arrow[rr, "K'G'\phi"] & & K'G'H' \\ KG'H \arrow[ur, "\psi G'H"] \arrow[rr, "KG'\phi"'] & & KG'H' \arrow[ur, "\psi G'H'"] & \end{tikzcd} \tag{78}$$

This commutes because each face of the cube corresponds to a variant of diagram (75) (with some substitutions and application of a functor) and combining commutative diagrams yields commutative diagrams.[157] Then, it follows easily that $\diamond$ is associative. □

[156] All $\circ$'s are left out for simplicity.

[157] Here is the rundown: Call Nat(phi,f,A,B) then use it.

Top face:

Bottom face:

Left face:

Right face:

Front face:

Back face:

There is one last thing to conclude that **Cat** is a 2–category, namely, that the vertical and horizontal compositions interact nicely.

Proposition 208 (Interchange identity)**.** *In the setting of* (80), *the* ***interchange identity*** *holds:*

$$(\eta' \cdot \eta) \diamond (\phi' \cdot \phi) = (\eta' \diamond \phi') \cdot (\eta \diamond \phi). \tag{79}$$

It is in the drawing of (80) that the intuition behind vertical and horizontal is taken.

$$\begin{tikzcd}[row sep=large] \mathbf{C} \arrow[r, squiggly, bend left=50, "H"] \arrow[r, squiggly, "H'" near end] \arrow[r, squiggly, bend right=50, "H''"'] \arrow[r, phantom, shift left=3, "\Downarrow\phi"] \arrow[r, phantom, shift right=3, "\Downarrow\phi'"] & \mathbf{D} \arrow[r, squiggly, bend left=50, "G"] \arrow[r, squiggly, "G'" near end] \arrow[r, squiggly, bend right=50, "G''"'] \arrow[r, phantom, shift left=3, "\Downarrow\eta"] \arrow[r, phantom, shift right=3, "\Downarrow\eta'"] & \mathbf{E} \end{tikzcd} \tag{80}$$

Proof. Akin to the other proofs, this is a matter of combining the right diagrams. After combining the diagrams in $[\mathbf{C}, \mathbf{E}]$ corresponding to $\eta \diamond \phi$ and $\eta' \diamond \phi'$, it is easy to see that the R.H.S. of (79) is the morphism going from $G \circ H$ to $G'' \circ H''$ (see (81)).

$$\begin{tikzcd}
G \circ H \arrow[r, "G\phi"] \arrow[d, "\eta H"'] & G \circ H' \arrow[d, "\eta H'"] & \\
G' \circ H \arrow[r, "G'\phi"'] & G' \circ H' \arrow[r, "G'\phi'"] \arrow[d, "\eta' H'"'] & G' \circ H'' \arrow[d, "\eta' H''"] \\
& G'' \circ H' \arrow[r, "G''\phi'"'] & G'' \circ H''
\end{tikzcd} \tag{81}$$

Moreover, observe that the diagram corresponding to the L.H.S. can be factored with the following equations.

$$(\eta' \cdot \eta)H = \eta' H \cdot \eta H \qquad (\eta' \cdot \eta)H'' = \eta' H'' \cdot \eta H''$$
$$G(\phi' \cdot \phi) = G\phi' \cdot G\phi \qquad G''(\phi' \cdot \phi) = G''\phi' \cdot G''\phi$$

Combining the factored diagram with (81), we obtain (82) from which the interchange identity readily follows.

$$\begin{tikzcd}
G \circ H \arrow[r, "G\phi"] \arrow[d, "\eta H"'] & G \circ H' \arrow[r, "G\phi'"] \arrow[d, "\eta H'"] & G \circ H'' \arrow[d, "\eta H''"] \\
G' \circ H \arrow[r, "G'\phi"'] \arrow[d, "\eta' H"'] & G' \circ H' \arrow[r, "G'\phi'"] \arrow[d, "\eta' H'"'] & G' \circ H'' \arrow[d, "\eta' H''"] \\
G'' \circ H \arrow[r, "G''\phi"'] & G'' \circ H' \arrow[r, "G''\phi'"'] & G'' \circ H''
\end{tikzcd} \tag{82}$$

□

Definition 209 (Strict 2–cateory)**.** A **strict 2–category** consists of

- a category $\mathbf{C}$,
- for every $A, B \in \mathbf{C}_0$ a category $\mathbf{C}(A, B)$ with $\mathrm{Hom}_{\mathbf{C}}(A, B)$ as its objects (composition is denoted $\cdot$ and identities $\mathbb{1}$) and morphisms are called 2–**morphism**,
- a category with $\mathbf{C}_0$ as its objects, where the morphisms are pairs of parallel morphisms of $\mathbf{C}$ along with a 2–morphism between them[158] and the identity map sends $A \in \mathbf{C}_0$ to the pair $(\mathrm{id}_A, \mathrm{id}_A)$ and the 2–morphism $\mathbb{1}_{\mathrm{id}_A}$ (composition of 2–cells is denoted $\diamond$),

such that the interchange identity (79) holds.

We will not cover it in this book, but there are notions of morphisms between 2–categories (called 2–functors), between 3–categories as well as between n–categories for any n (even $n = \infty$!), these objects are more deeply studied in higher category theory.[159]

A very useful corollary of Proposition 208 is shown in the next exercise (it also follows from the properties of the biaction of functors). In particular, it means that for any commutative diagram in $[\mathbf{C}, \mathbf{D}]$, we can pre-compose and post-compose

[158] A morphism in this category is also called a 2–**cell**.

[159] Most of higher category theory drops the *strict* part of our definition of 2–category because this condition is too strong. Very briefly, they allow the properties of composition, namely associativity and identities, to hold up to natural isomorphisms.

with any functors and still obtain a commutative diagram. For instance, if (83) commutes in $[\mathbf{C},\mathbf{D}]$, then for any functors $F : \mathbf{C}' \rightsquigarrow \mathbf{C}$ and $G : \mathbf{D} \rightsquigarrow \mathbf{D}'$, then (84) commutes.[160]

$$\begin{tikzcd} X \arrow[r, "\eta"] \arrow[d, "\phi"'] & Y \arrow[d, "\phi'"] \\ X' \arrow[r, "\eta'"'] & Y' \end{tikzcd} \qquad (83) \qquad \begin{tikzcd} F \circ X \circ G \arrow[r, "F\eta G"] \arrow[d, "F\phi G"'] & F \circ Y \circ G \arrow[d, "F\phi' G"] \\ F \circ X' \circ G \arrow[r, "F\eta' G"'] & F \circ Y' \circ G \end{tikzcd} \qquad (84)$$

[160] We will often use this property by writing things like "apply $F(-)G$ to (83)" to use the commutativity of (84) in a proof.

Exercise 210 (NOW!)**.** Show that there is a functor $[\mathbf{D},\mathbf{E}] \times [\mathbf{C},\mathbf{D}] \rightsquigarrow [\mathbf{C},\mathbf{E}]$ whose action on objects is $(F, G) \mapsto F \circ G$.

See solution.

Equivalences

As is expected, an isomorphism of categories is an isomorphism in the category **Cat**, namely, a functor $F : \mathbf{C} \rightsquigarrow \mathbf{D}$ with an inverse $G : \mathbf{D} \rightsquigarrow \mathbf{C}$ such that $F \circ G = \mathrm{id}_{\mathbf{D}}$ and $G \circ F = \mathrm{id}_{\mathbf{C}}$. As is typical in mathematics, one cannot distinguish between isomorphic categories as they only differ in notations and terminology.

Examples 211.

1. It was already shown in Example 199 (the details were implicit) that for a group G, the category $[\mathbf{Set}, \mathbf{B}(G)]$ is isomorphic to the category of G–sets with G–equivariant maps as morphisms.

Another example for readers who know a bit of advanced algebra. Let k be a field and G a finite group, the categories of $k[G]$–modules ($k[G]$ is the group ring of k over G) and of k–linear representations of G are isomorphic.

2. In Example 201, three other isomorphisms were implicitly given:

$$[\mathbf{1},\mathbf{C}] \cong \mathbf{C} \qquad [\mathbf{1}+\mathbf{1},\mathbf{C}] \cong \mathbf{C} \times \mathbf{C} \qquad [\mathbf{2},\mathbf{C}] \cong \mathbf{C}^{\rightarrow}.$$

3. The category **Rel** of sets with relations is isomorphic to $\mathbf{Rel}^{\mathrm{op}}$.[161] The functor $\mathbf{Rel} \rightsquigarrow \mathbf{Rel}^{\mathrm{op}}$ is the identity on objects and sends a relation $R \subseteq X \times Y$ to the opposite relation $\reflectbox{$R$} \subseteq Y \times X$ (which is a morphism $X \to Y$ in $\mathbf{Rel}^{\mathrm{op}}$) defined by $(y, x) \in \reflectbox{$R$} \Leftrightarrow (x, y) \in R$. The inverse is defined similarly.

[161] An arbitrary category **C** is not always isomorphic to its opposite. While the opposite functors $(-)^{\mathrm{op}}_{\mathbf{C}} : \mathbf{C} \rightsquigarrow \mathbf{C}^{\mathrm{op}}$ and $(-)^{\mathrm{op}}_{\mathbf{C}^{\mathrm{op}}} : \mathbf{C}^{\mathrm{op}} \rightsquigarrow \mathbf{C}$ are inverses of each other, they are contravariant functors.

4. Given three categories **C**, **D** and **E**, there is an isomorphism[162]

$$[\mathbf{C} \times \mathbf{D}, \mathbf{E}] \cong [\mathbf{C}, [\mathbf{D}, \mathbf{E}]].$$

Let $F : \mathbf{C} \times \mathbf{D} \rightsquigarrow \mathbf{E}$, the currying of F is $\lambda F : \mathbf{C} \rightsquigarrow [\mathbf{D}, \mathbf{E}]$ defined as follows. For $X \in \mathbf{C}_0$, the

[162] You might recognize a similarity with exponentials which rely on an isomorphism $\mathrm{Hom}_{\mathbf{C}}(B \times X, A) \cong \mathrm{Hom}_{\mathbf{C}}(B, A^X)$. The example here is more than an instance of exponentials of categories because the isomorphism is not only as sets but as categories.

Although there are other interesting instances of isomorphic categories, natural transformations lead to a more nuanced (and often more useful) equality between two categories, that is, equivalence.

Definition 212 (Equivalence)**.** A functor $F : \mathbf{C} \rightsquigarrow \mathbf{D}$ is an **equivalence** of categories if there exists a functor $G : \mathbf{D} \rightsquigarrow \mathbf{C}$ such that $F \circ G \cong \mathrm{id}_{\mathbf{D}}$ and $G \circ F \cong \mathrm{id}_{\mathbf{C}}$.[163] This is clearly symmetric, so we say two categories **C** and **D** are **equivalent**, denoted $\mathbf{C} \simeq \mathbf{D}$, if there is an equivalence between them. Moreover, we say that G is a **quasi-inverse** of F and vice-versa.

[163] Recall that $\cong$ between functors stands for natural isomorphisms.

In order to gain more intuition on how equivalences equate two categories, let us observe what properties this forces on the functor F. For any morphism $f \in \mathrm{Hom}_{\mathbf{C}}(A, B)$, the following square commutes where $\phi(A)$ and $\phi(B)$ are isomorphisms.[164]

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, shift left, "\phi(A)"] & B \arrow[d, shift right, "\phi(B)"'] \\ GF(A) \arrow[r, "GF(f)"'] \arrow[u, shift left, "\phi(A)^{-1}"] & GF(B) \arrow[u, shift right, "\phi(B)^{-1}"'] \end{tikzcd} \tag{85}$$

This implies that the map $f \mapsto GF(f) : \mathrm{Hom}_{\mathbf{C}}(A, B) \to \mathrm{Hom}_{\mathbf{C}}(GF(A), GF(B))$ is a bijection. Indeed, pre-composition by $\phi(A)^{-1}$ and post-composition by $\phi(B)$ are both bijections,[165] so

$$f \mapsto \phi(B) \circ f \circ \phi(A)^{-1} = GF(f)$$

is a bijection. Since A and B are arbitrary, $G \circ F$ is a fully faithful functor and a symmetric argument shows $F \circ G$ is also fully faithful. Then, it is easy to conclude that F and G must be fully faithful as well.

What is more, the existence of an isomorphism $\eta(A) : A \to FG(A)$ for any object A implies F (symmetrically G) has the following property.

Definition 213 (Essentially surjective)**.** A functor $F : \mathbf{C} \rightsquigarrow \mathbf{D}$ is **essentially surjective** if for any $X \in \mathbf{D}_0$, there exists $Y \in \mathbf{C}_0$ such that $X \cong F(Y)$.

We will show that these two properties (full faithfulness and essential surjectivity are necessary and sufficient for F to be an equivalence.

Theorem 214. *A functor $F : \mathbf{C} \rightsquigarrow \mathbf{D}$ is an equivalence of categories if and only if F is fully faithful and essentially surjective.*

Proof. ($\Rightarrow$) Shown above.

($\Leftarrow$) We construct a functor $G : \mathbf{D} \rightsquigarrow \mathbf{C}$ such that $G \circ F \cong \mathrm{id}_{\mathbf{C}}$ and $F \circ G \cong \mathrm{id}_{\mathbf{D}}$. Since F is essentially surjective, for any $A \in \mathbf{D}_0$, there exists an object $G(A) \in \mathbf{C}_0$ and an isomorphism $\phi(A) : F(G(A)) \cong A$. Hence, $A \mapsto G(A)$ is a good candidate to describe the action of G on objects.

Next, similarly to the converse direction, note that for any $A, B \in \mathbf{D}_0$, the map

$$f \mapsto \phi(B) \circ f \circ \phi(A)^{-1}$$

is a bijection from $\mathrm{Hom}_{\mathbf{D}}(A, B)$ to $\mathrm{Hom}_{\mathbf{D}}(FG(A), FG(B))$. Moreover, since the functor F is fully faithful, it induces a bijection

$$F_{A,B} : \mathrm{Hom}_{\mathbf{C}}(G(A), G(B)) \to \mathrm{Hom}_{\mathbf{D}}(FG(A), FG(B))$$

which in turns yields a bijection

$$G_{A,B} : \mathrm{Hom}_{\mathbf{D}}(A, B) \to \mathrm{Hom}_{\mathbf{C}}(G(A), G(B)) = f \mapsto F_1^{-1}(\phi(B) \circ f \circ \phi(A)^{-1}).$$

This is the action of G on morphisms. Observe that the construction of G ensures that $F \circ G \cong \mathrm{id}_{\mathbf{D}}$ through the natural transformation ϕ. It remains to show that G is indeed a functor and find a natural isomorphism $\eta : G \circ F \cong \mathrm{id}_{\mathbf{C}}$.

[164] Naturality of ϕ only gives us $GF(f) \circ \phi(A) = \phi(B) \circ f$, but by composing with $\phi(A)^{-1}$ or $\phi(B)^{-1}$, we obtain the commutativity of all of (85). In particular, we have $GF(f) = \phi(B) \circ f \circ \phi(A)^{-1}$.

[165] Recall the definitions of monomorphisms and epimorphisms and the fact that isomorphisms are monic and epic.

For any composable morphisms (f, g), it is easy to verify that

$$F(G(f) \circ G(g)) = FG(f) \circ FG(g) = FG(f \circ g),$$

so functoriality of G follows after applying F_1^{-1}. To find η, recall that the definition of G yields commutativity of (86) for any $f \in \mathrm{Hom}_{\mathbf{C}}(A, B)$.

$$\begin{tikzcd} F(A) \arrow[r, "F(f)"] \arrow[d, leftrightarrow, "\phi(F(A))"'] & F(B) \arrow[d, leftrightarrow, "\phi(F(B))"] \\ FGF(A) \arrow[r, "FGF(f)"'] & FGF(B) \end{tikzcd} \tag{86}$$

Then, because F is fully faithful, the following square also commutes in $\mathbf{C}$ where $\eta = X \mapsto F_1^{-1}(\phi(F(X)))$ and we conclude that η is a natural isomorphism $\mathrm{id}_{\mathbf{C}} \cong G \circ F$.

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, leftrightarrow, "\eta(A)"'] & B \arrow[d, leftrightarrow, "\eta(B)"] \\ GF(A) \arrow[r, "GF(f)"'] & GF(B) \end{tikzcd} \tag{87}$$

□

The insight to extract from this argument is that two categories are equivalent if they describe the same objects and morphisms with the only relaxation that isomorphic objects can appear any number of times in either category. In contrast, categories can only be isomorphic if they have exactly the same objects and morphisms.

Remark 215. We used the axiom of choice to construct the quasi-inverse of F.

We will detail a couple of *easy* examples of equivalences and briefly metion a few *harder* ones.

Examples 216 (Easy)**.** 1. Consider the full subcategory of **FinSet** consisting only of the sets $\emptyset, \{1\}, \{1,2\}, \ldots, \{1, \ldots, n\}, \ldots$, denote it **FinOrd**.[166] The inclusion functor is, by definition, already fully faithful and we claim it is essentially surjective. Indeed, any set $X \in \mathbf{FinSet}_0$ has a finite cardinality n, so $X \cong \{1, \ldots, n\} \in \mathbf{FinOrd}_0$.

[166] The name **FinOrd** is an abbreviation of finite ordinals, because we can also define **FinOrd** as the category of finite ordinals and functions between them.

2. In a very similar fashion, an early result in linear algebra says that any finite dimensional vector space over a field k is isomorphic to k^n for some $n \in \mathbb{N}$. Thus, the category whose objects are k^n for all $n \in \mathbb{N}$ and morphisms are $m \times n$ matrices with entries in k,[167] which we denote $\mathbf{Mat}(k)$, is equivalent to the category of finite dimensional vector spaces.

[167] After making a choice of basis for all k^n, an $m \times n$ matrix with entries in k corresponds to a linear map $k^n \to k^m$.

3. A **partial** function $f : X \rightharpoonup Y$ is a function that may not be defined on all of X.[168] There is category **Par** of sets and partial functions where identity morphism and composition are defined straightforwardly.[169] We can view a partial function $f : X \rightharpoonup Y$ as a total function $f' : X \to Y + \mathbf{1}$ which assigns to every x where $f(x)$

[168] In this context, a *normal* function defined on all of X is called **total**.

[169] You can view **Par** as the subcategory of **Rel** where you only take the relations $R \subseteq X \times Y$ satisfying for any $x \in X$ (cf. Remark 78),

$$|\{y \in Y \mid (x, y) \in R\}| \leq 1.$$

is undefined the value $* \in \mathbf{1}$. Further extending f' to $[f', \mathrm{id}_{\mathbf{1}}] : X + \mathbf{1} \to Y + \mathbf{1}$, we can see any partial function as a function between pointed sets where the distinguished element corresponds to being undefined.

We claim that this yields a fully faithful functor $\mathbf{Par} \rightsquigarrow \mathbf{Set}_*$ sending X to $(X + \mathbf{1}, *)$ and $f : X \rightharpoonup Y$ to $[f', \mathrm{id}_{\mathbf{1}}]$.

The first two examples and many other simple examples of equivalences are examples of skeletons. They are morally a subcategory where all the isomorphic copies are removed.

Definition 217 (Skeleton)**.** A category is called **skeletal** if there it contains no two isomorphic objects. A **skeleton** of a category is an equivalent skeletal category.

Examples 218. We have shown that $\mathbf{FinOrd} \simeq \mathbf{FinSet}$ and $\mathbf{Mat}(k) \simeq \mathbf{FDVect}_k$ and we leave to you the easy task to check that these are examples of skeletons.[170]

[170] Namely, you should show that no two sets in **FinOrd** are isomorphic and no two spaces in $\mathbf{Mat}(k)$ are isomorphic.

A category always has a skeleton if you assume the axiom of choice and the next result justifies say *the* skeleton of a category.

Exercise 219. Show that all skeletons of a category are isomorphic.

See solution.

Here are other more interesting examples of equivalent categories.

Examples 220 (Hard)**.** Examples of significant equivalences are all over the place in higher mathematics. However, they require a bit of work to describe them, thus let us only say a few words on them.

1. The equivalence between the category of affine schemes and the opposite of the category of commutative rings is a seminal result scheme theory, a huge part of modern algebraic geometry.

2. The equivalence between Boolean lattices and Stone spaces is again seminal in the theory of Stone-type dualities. These can lead to deep connections between topology and logic. One application in particular is the study of the behavior of computer programs through formal semantics.

Exercise 221. Show that equivalence of categories is an equivalence relation.

See solution.

Exercise 222. Let $\mathbf{C} \simeq \mathbf{C}'$ and $\mathbf{D} \simeq \mathbf{D}'$ be equivalent categories, then $[\mathbf{C}, \mathbf{D}] \simeq [\mathbf{C}', \mathbf{D}']$.

See solution.

Yoneda Lemma

Representable Functors

Throughout, let $\mathbf{C}$ be a locally small category. Recall that for an object $A \in \mathbf{C}_0$, there are two Hom functors from $\mathbf{C}$ to $\mathbf{Set}$. The covariant one, $\mathrm{Hom}_{\mathbf{C}}(A, -)$, sends an object $B \in \mathbf{C}_0$ to $\mathrm{Hom}_{\mathbf{C}}(A, B)$ and a morphism $f : B \to B'$ to $f \circ (-)$. The contravariant one, $\mathrm{Hom}_{\mathbf{C}}(-, A)$, sends an object $B \in \mathbf{C}_0$ to $\mathrm{Hom}_{\mathbf{C}}(B, A)$ and a morphism $f : B \to B'$ to $(-) \circ f$. In order to lighten the notation, we denote these functors H^A and H_A respectively.[171]

Although these functors are sometimes interesting on their own, their full power is unleashed when they are related to other functors through natural transformations. For instance, some of these Hom functors can be described in simpler terms.

[171] It might seem like this contradicts the notation used so far because H^A is covariant and H_A contravariant. However, this is not their *variance* in the parameter A, and we will show that in fact, the *variance* in A are opposites.

Examples 223.

1. Let $\mathbf{1} = \{*\}$ be the terminal object in $\mathbf{Set}$, then what is the action of $H^{\mathbf{1}}$? For any object B,
$$H^{\mathbf{1}}(B) = \mathrm{Hom}_{\mathbf{Set}}(\mathbf{1}, B)$$
is easy to describe because for any element $b \in B$, there is a unique function $f : \mathbf{1} \to B = * \mapsto b$. Hence, there is an isomorphism from $H^{\mathbf{1}}(B)$ to B for any $B \in \mathbf{C}_0$, it sends f to $f(*)$ and its inverse sends $b \in B$ to the map $* \mapsto b$. Moreover, these isomorphisms are natural in B because (88) clearly commutes for any $f : B \to B'$, yielding a natural isomorphism $H^{\mathbf{1}} \cong \mathrm{id}_{\mathbf{C}}$.
$$\begin{tikzcd} H^{\mathbf{1}}(B) \arrow[r, "f\circ(-)"] \arrow[d, leftrightarrow] & H^{\mathbf{1}}(B') \arrow[d, leftrightarrow] \\ B \arrow[r, "f"'] & B' \end{tikzcd} \tag{88}$$

2. Consider again the terminal object in the category $\mathbf{Grp}$, namely, the group $\mathbf{1}$ only containing an identity. Then, for any group G, the set $H^{\mathbf{1}}(G)$ is a singleton because any homomorphism $f : \mathbf{1} \to G$ must send the identity to the identity and no other choice can be made. Therefore, unlike in $\mathbf{Set}$, $H^{\mathbf{1}}$ is very uninteresting and acts like the constant functor $\mathbf{1} : \mathbf{Grp} \rightsquigarrow \mathbf{Set}$.

3. A better choice of object to mimic the behavior of $\mathrm{id}_{\mathbf{Grp}}$ is the additive group $\mathbb{Z}$. Indeed, for any $g \in G$, there is a unique homomorphism $f : \mathbb{Z} \to G$ sending 0

to the identity and 1 to g.[172] A very similar argument as above yields a natural isomorphism $H^{\mathbb{Z}} \cong \mathrm{id}_{\mathbf{Grp}}$.

[172] Note that f is completely determined by $f(1)$ because $f(n) = f(1) + \overset{n}{\cdots} + f(1)$ and $f(0)$ must be the identity.

4. The terminal object in **Cat** is the category $\mathbf{1}$ with a single object $\bullet$ and no morphism other than the identity. Observe that for any category $\mathbf{C}$, a functor $\mathbf{1} \rightsquigarrow \mathbf{C}$ is just a choice of object. Therefore, the same argument will show that $H^{\mathbf{1}} \cong (-)_0$, where $(-)_0$ sends a category to its set[173] of objects and a functor to its action restricted on objects.

   In order to obtain a similar way to extract morphisms, consider the category $\mathbf{2}$ with two objects and a single morphism between them. One obtains a natural isomorphism $H^{\mathbf{2}} \cong (-)_1$.

[173] Recall that **Cat** only contains small categories.

These examples suggest that functors that are naturally isomorphic to Hom functors have nice properties, they are said to be representable.

Definition 224 (Representable functor)**.** A covariant functor $F : \mathbf{C} \rightsquigarrow \mathbf{Set}$ is **representable** if there is an object $X \in \mathbf{C}_0$ such that F is naturally isomorphic to $\mathrm{Hom}_{\mathbf{C}}(X, -)$. If F is contravariant, then it is representable if it is naturally isomorphic to $\mathrm{Hom}_{\mathbf{C}}(-, X)$.

Examples 225. Let us give examples of the contravariant kind.

1. The contravariant powerset functor $\widehat{\mathcal{P}} : \mathbf{Set} \rightsquigarrow \mathbf{Set}$ sends a set X to its powerset $\mathcal{P}(X)$ and a function $f : X \to Y$ to the inverse image $f^{-1} : \mathcal{P}(Y) \to \mathcal{P}(X)$. It is common to identify subsets of a given set with functions from this set into $2 = \{0, 1\}$. Formally, this is an isomorphism $\widehat{\mathcal{P}}(X) \cong H_2(X) = 2^X$ for any X, it maps $S \subseteq X$ to the characteristic function χ_S.[174] In the reverse direction, it sends a function $g : X \to \{0, 1\}$ to $g^{-1}(1)$. It is easy to check that for any $f : X \to Y$, the isomorphisms make (89) commute, so $\widehat{\mathcal{P}} \cong H_2$.

$$\begin{tikzcd} H_2(X) \arrow[r, "f \circ (-)"] \arrow[d, leftrightarrow] & H_2(Y) \arrow[d, leftrightarrow] \\ \widehat{\mathcal{P}}(X) \arrow[r, "f^{-1}"'] & \widehat{\mathcal{P}}(Y) \end{tikzcd} \tag{89}$$

[174] It sends $x \in X$ to 1 if $x \in S$ and to 0 otherwise.

2. In functional programming, it is often useful to transform a function taking multiple arguments so that it ends up taking a single argument but outputs another function. For instance, the multiplication function $\mathsf{mult} : \mathsf{int} \times \mathsf{int} \to \mathsf{int}$ that takes two numbers as inputs and outputs their product can be rewritten as $\mathsf{multc} : \mathsf{int} \to (\mathsf{int} \to \mathsf{int})$. The function multc takes a number as input and outputs a function that outputs the product of its input and the initial input of multc. For example $\mathsf{multc}(3)$ is a function that outputs $3 \cdot n$ when n is the input. This new function multc is said to be the curried version of mult in honor of Haskell Curry. This leads to a more general argument in **Set**.

   Fix two sets A and B. The functor $\mathrm{Hom}(- \times A, B)$ maps a set X to $\mathrm{Hom}(X \times A, B)$ and a function $f : X \to Y$ to the function $(-) \circ (f \times \mathrm{id}_A)$.[175] As suggested

[175] You can see it as the composition $H_B \circ (- \times A)$.

by the currying process for mult, for any set X, there is a bijection $\mathrm{Hom}(X \times A, B) \cong \mathrm{Hom}(X, B^A)$. The image of $f : X \times A \to B$ is denoted λf and it satisfies $f(x,a) = \lambda f(x)(a)$ for any $x \in X$ and $a \in A$. It is easy to check that this is a bijection and also that it is natural in X because the (90) commutes for any $f : X \to Y$, so $\mathrm{Hom}(- \times A, B) \cong \mathrm{Hom}(-, B^A)$.

$$\begin{tikzcd} \mathrm{Hom}(X \times A, B) \arrow[r, "(-)\circ(f\times \mathrm{id}_A)"] \arrow[d, leftrightarrow] & \mathrm{Hom}(Y \times A, B) \arrow[d, leftrightarrow] \\ \mathrm{Hom}(X, B^A) \arrow[r, "(-)\circ f"'] & \mathrm{Hom}(Y, B^A) \end{tikzcd} \tag{90}$$

In the first item of Examples 223 and 225, we made an arbitrary choice of set. That is, we could have taken any singleton in the first case and any set with two elements in the second. More generally, one can show that if $A \cong B$, then $H_A \cong H_B$ and $H^A \cong H^B$.

Exercise 226. Let $A, B \in \mathbf{C}_0$ be isomorphic objects. Show that $H^A \cong H^B$. Dually, show that $H_A \cong H_B$.

Surprisingly, the converse is also true and it will follow from the Yoneda lemma, but we prove it on its own first as a warm-up for the proof of the lemma.

Proposition 227. *Let $A, B \in \mathbf{C}_0$ be such that $H^A \cong H^B$, then $A \cong B$.*

Proof. The natural isomorphism gives two natural transformations $\phi : H^A \Rightarrow H^B$ and $\eta : H^B \Rightarrow H^A$ such that for any object $X \in \mathbf{C}_0$,

$$\eta_X \circ \phi_X : H^A(X) \to H^A(X) \quad \text{and} \quad \phi_X \circ \eta_X : H^B(X) \to H^B(X)$$

are identities. In order to show $A \cong B$, we will find two morphisms $f : B \to A$ and $g : A \to B$ such that $f \circ g = \mathrm{id}_A$ and $g \circ f = \mathrm{id}_B$.

First, note that putting X equal to A, we get $\eta_A(\phi_A(\mathrm{id}_A)) = \mathrm{id}_A$ and we claim that

$$\eta_A(\phi_A(\mathrm{id}_A)) = \phi_A(\mathrm{id}_A) \circ \eta_B(\mathrm{id}_B).$$

Since $\phi_A(\mathrm{id}_A)$ is a morphism from B to A, (91) commutes by naturality of η. The equality then follows, starting with $\mathrm{id}_B \in H_B(B)$.

$$\begin{tikzcd} H_B(A) \arrow[r, "\eta_A"] & H_A(A) \\ H_B(B) \arrow[u, "\phi_A(\mathrm{id}_A)\circ(-)"] \arrow[r, "\eta_B"'] & H_A(B) \arrow[u, "\phi_A(\mathrm{id}_A)\circ(-)"'] \end{tikzcd} \tag{91}$$

A dual argument shows that

$$\mathrm{id}_B = \phi_B(\eta_B(\mathrm{id}_B)) = \eta_B(\mathrm{id}_B) \circ \phi_A(\mathrm{id}_A),$$

so we can conclude, letting $f = \phi_A(\mathrm{id}_A)$ and $g = \eta_B(\mathrm{id}_B)$, that $A \cong B$. $\square$

For every $A \in \mathbf{C}_0$, there are two functors H^A and H_A, they are objects of $[\mathbf{C}, \mathbf{Set}]$ and $[\mathbf{C}^{\text{op}}, \mathbf{Set}]$ respectively. It is then reasonable to expect that the assignments $A \mapsto H^A$ and $A \mapsto H_A$ are functorial.

Definition 228 (Yoneda embeddings). The contravariant embedding $H^{(-)} : \mathbf{C}^{\text{op}} \rightsquigarrow [\mathbf{C}, \mathbf{Set}]$ sends $A \in \mathbf{C}_0$ to the Hom functor H^A and a morphism $f : A' \to A$ to the natural transformation $H^f : H^A \Rightarrow H^{A'}$ defined by $H^f_B := \text{Hom}_{\mathbf{C}}(f, B) = (-) \circ f$ for every $B \in \mathbf{C}_0$. The naturality of H^f follows because (92) commutes (by associativity) for any $g : B \to B'$.

$$\begin{tikzcd} H^A(B) \arrow[r, "(-)\circ f"] \arrow[d, "g\circ(-)"'] & H^{A'}(B) \arrow[d, "g\circ(-)"] \\ H^A(B') \arrow[r, "(-)\circ f"'] & H^{A'}(B') \end{tikzcd} \tag{92}$$

The covariant embedding $H_{(-)} : C \rightsquigarrow [\mathbf{C}^{\text{op}}, \mathbf{Set}]$ sends $B \in \mathbf{C}_0$ to the Hom functor H_B and a morphism $f : B \to B'$ to the natural transformation $H_f : H_B \to H_{B'}$ defined by $H^A_f = \text{Hom}_{\mathbf{C}}(A, f) = f \circ (-)$ for any $A \in \mathbf{C}_0$. Naturality follows from a similar argument.

Functoriality is left for the reader to check. The embeddings are called like that because both functors are fully faithful as will follow from the Yoneda lemma.

Yoneda Lemma

We have understood how an object $A \in \mathbf{C}_0$ sees the category $\mathbf{C}$ through representables, but since a representable is an object of another category, it is daring to study what representables see and how it relates to the object it represents. More formally, what is the functor $\text{Hom}_{[\mathbf{C},\mathbf{Set}]}(H^A, -)$ describing. For simplicity, we denote it $\text{Nat}(H^A, -)$ because, for a functor $F : \mathbf{C} \rightsquigarrow \mathbf{Set}$, $\text{Nat}(H^A, F)$ is the collection[176] of natural transformations from H^A to F.

The surprising relation that the Yoneda lemma describes is that $\text{Nat}(H^A, F)$ is isomorphic to $F(A)$ naturally in F and A. We first show the isomorphism and then explain the naturality.

[176] Even if $\mathbf{C}$ is locally small, there is no guarantee that $[\mathbf{C}, \mathbf{Set}]$ is locally small. Nevertheless, one consequence of the Yoneda lemma is that $\text{Nat}(F, G)$ is a set whenever F is representable.

Lemma 229 (Yoneda lemma I)**.** *For any $A \in \mathbf{C}_0$ and $F : C \rightsquigarrow \mathbf{Set}$,*

$$\text{Nat}(H^A, F) \cong F(A).$$

Proof. Fix A and F, let $\phi_{A,F} : \text{Nat}(H^A, F) \to F(A)$ be defined by $\alpha \mapsto \alpha_A(\text{id}_A)$ (check that the types match). Let $\eta_{A,F} : F(A) \to \text{Nat}(H^A, F)$ send an element $a \in F(A)$ to the natural transformation that has components $\eta_{A,F}(a)_B : f \mapsto F(f)(a) : \text{Hom}_{\mathbf{C}}(A, B) \to F(B)$ for any $B \in \mathbf{C}_0$. Checking (93) commutes for any $g : B \to B'$ shows that $\eta_{A,F}(a)$ is a natural transformation.

$$\begin{tikzcd} H^A(B) \arrow[r, "F(-)(a)"] \arrow[d, "g\circ(-)"'] & F(B) \arrow[d, "F(g)"] \\ H^A(B') \arrow[r, "F(-)(a)"'] & F(B') \end{tikzcd} \tag{93}$$

We now check that $\phi_{A,F}$ and $\eta_{A,F}$ are inverses. First, $(\eta \circ \phi)_{A,F}$ sends $\alpha \in \mathsf{Nat}(H^A, F)$ to $\eta_{A,F}(\alpha_A(\mathrm{id}_A))$, and at any $B \in \mathbf{C}_0$, we have

$$\begin{aligned} \eta_{A,F}(\alpha_A(\mathrm{id}_A))_B(f) &= F(f)(\alpha_A(\mathrm{id}_A)) && \text{def of } \eta \\ &= \alpha_B(f \circ \mathrm{id}_A) && \text{naturality of } \alpha \\ &= \alpha_B(f), \end{aligned}$$

thus $\alpha = (\eta \circ \phi)_{A,F}(\alpha)$.

Conversely, $(\phi \circ \eta)_{A,F}$ sends $a \in F(A)$ to $\eta_{A,F}(a)_A(\mathrm{id}_A) = F(\mathrm{id}_A)(a) = a$.

We conclude that $\eta_{A,F}$ and $\phi_{A,F}$ are inverses. □

What this results first tells us is that $\mathsf{Nat}(H^A, F)$ is a set (because it is isomorphic to $F(A)$ which is a set). This lets us define two new functors to understand the second part of the Yoneda lemma.

The assignment $(A, F) \mapsto \mathsf{Nat}(H^A, F)$ is a functor $\mathbf{C} \times [\mathbf{C}, \mathbf{Set}] \rightsquigarrow \mathbf{Set}$. We denote it $\mathsf{Nat}(H^{(-)}, -)$, it sends a morphism $(g, \mu) : (A, F) \to (A', F')$ to $\mu \cdot (-) \cdot H^g : \mathsf{Nat}(H^A, F) \to \mathsf{Nat}(H^{A'}, F')$.[177]

The assignment $(A, F) \mapsto F(A)$ is another functor of the same type. We denote it Ev (for evaluation), it sends a morphism $(g, \mu) : (A, F) \to (A', F')$ to $F'(g) \circ \mu_A : F(A) \to F'(A')$.

[177] As $\mathsf{Nat}(-,-)$ is the Hom bifunctor of $[\mathbf{C}, \mathbf{Set}]$, we can see $\mathsf{Nat}(H^{(-)}, -)$ as the composition

$$\mathsf{Nat}(-,-) \circ (H^{(-)} \times \mathrm{id}_{[\mathbf{C},\mathbf{Set}]}).$$

Lemma 230 (Yoneda lemma II)**.** *There is a natural isomorphism* $\mathsf{Nat}(H^{(-)}, -) \cong \mathsf{Ev}$.

Proof. The components of this isomorphism are the ones described in the first part of the result. It remains to show that ϕ is natural in (A, F). For any $(g, \mu) : (A, F) \to (A', F')$, we need to show the following square commutes.

$$\begin{tikzcd} \mathsf{Nat}(H^A, F) \arrow[r, "\phi_{A,F}"] \arrow[d, "\mu \cdot (-) \cdot H^g"'] & F(A) \arrow[d, "F'(g) \circ \mu_A"] \\ \mathsf{Nat}(H^{A'}, F') \arrow[r, "\phi_{A',F'}"'] & F'(A') \end{tikzcd} \tag{94}$$

Starting with a natural transformation $\alpha \in \mathsf{Nat}(H^A, F)$ the lower path sends it to $(\mu \cdot \alpha \cdot H^g)_{A'}(\mathrm{id}_{A'})$ and the upper path sends it to $(F'(g) \circ \mu_A)(\alpha_A(\mathrm{id}_A))$. The following derivation shows they are equal.

$$\begin{aligned} (\mu \cdot \alpha \cdot H^g)_{A'}(\mathrm{id}_{A'}) &= (\mu_{A'} \circ \alpha_{A'})(H^g_{A'}(\mathrm{id}_{A'})) && \text{def of composition} \\ &= (\mu_{A'} \circ \alpha_{A'})(g) && \text{def of } H^g_{A'} \\ &= (\mu_{A'} \circ \alpha_{A'})(H^A_g(\mathrm{id}_A)) && \text{def of } H^A_g \\ &= (\mu_{A'} \circ \alpha_{A'} \circ H^A_g)(\mathrm{id}_A) \\ &= (\mu_{A'} \circ F(g) \circ \alpha_A)(\mathrm{id}_A) && \text{naturality of } \alpha \\ &= (F'(g) \circ \mu_A)(\alpha_A(\mathrm{id}_A)) && \text{naturality of } \mu \end{aligned}$$

□

Corollary 231. *The Yoneda embeddings* $H^{(-)}$ *and* $H_{(-)}$ *are fully faithful.*

Proof. Left as an exercise. □

Example 232 (Cayley's theorem with the Yoneda lemma)**.** Cayley's theorem states that any group is isomorphic to the subgroup of a permutation group. We will use the Yoneda lemma to show that.

Recall the first part of the Yoneda lemma which states that for a category $\mathbf{C}$, a functor $F : \mathbf{C} \rightsquigarrow \mathbf{Set}$ and an object $A \in \mathbf{C}_0$, we have

$$\mathrm{Nat}(\mathrm{Hom}(A, -), F) \cong F(A).$$

Moreover, we know the explicit maps, namely, a natural transformation ϕ in the L.H.S. is mapped to $\phi_A(\mathrm{id}_A)$ and an element $u \in F(A)$ is mapped to the natural transformation $\{\phi_B = f \mapsto F(f)(u) \mid B \in \mathbf{C}_0\}$.

Let us apply this to $\mathbf{C}$ being the delooping of G. Recall that any functor $F : \mathbf{B}(G) \rightsquigarrow \mathbf{Set}$ sends $*$ to a set S and any $g \in G$ to a permutation of S, it corresponds to an action of G on S.

To use the Yoneda lemma, our only choice of object for A is $*$ and we will choose for F the functor it represents, i.e.: $F = \mathrm{Hom}(*, -)$. The Yoneda lemma yields

$$\mathrm{Nat}(\mathrm{Hom}(*, -), \mathrm{Hom}(*, -)) \cong \mathrm{Hom}(*, *).$$

We already know what the R.H.S. is G,[178] but we have to do a bit of work to understand the L.H.S. First, observe that a natural transformation $\phi : \mathrm{Hom}(*, -) \Rightarrow \mathrm{Hom}(*, -)$ is just one morphism $\phi_* : \mathrm{Hom}(*, *) \to \mathrm{Hom}(*, *)$. Namely, it is a map from G to G. Second, recalling that $\mathrm{Hom}(*, g) = g \circ (-)$ and that $*$ is the only object in $\mathbf{C}_0$, we get that ϕ_* must only make (95) commute.

[178] By definition of $\mathbf{B}(G)$.

$$\begin{tikzcd} G \arrow[r, "\phi_*"] \arrow[d, "g\circ(-)"'] & G \arrow[d, "g\circ(-)"] \\ G \arrow[r, "\phi_*"'] & G \end{tikzcd} \tag{95}$$

This is equivalent to $\phi_*(g \cdot h) = g \cdot \phi_*(h)$, and we get that each ϕ_* is a G–equivariant map. Denote the set of G–equivariant maps $\mathrm{Hom}_G(G, G)$. We obtain that, as sets,

$$\mathrm{Hom}_G(G, G) \cong G.$$

Now, we can check that $\mathrm{Hom}_G(G, G)$ is a subgroup of Σ_G (the group of permutations of the set G) and that the bijection is in fact an group isomorphism. Cayley's theorem follows.

To check that $\mathrm{Hom}_G(G, G) < \Sigma_G$, we have to show that id_G is G–equivariant, that G–equivariant maps are bijective and that they are stable under composition and taking inverse. First, we have $\mathrm{id}_G(g \cdot h) = g \cdot h = g \cdot \mathrm{id}_G(h)$, so $\mathrm{id}_G \in \mathrm{Hom}_G(G, G)$. Second, let f be a G–equivariant map. For any $g \in G$, we have $f(g) = f(g \cdot 1) = g \cdot f(1)$. Thus, f is determined only by where it sends the identity. Additionally, sice for any choice of $f(1)$, $g \cdot f(1)$ ranges over G when g ranges over G, f is bijective. Therefore, if f and f' are both G–equivariant map, then

$$(f \circ f')(g \cdot h) = f(f'(g \cdot h)) = f(g \cdot f'(h)) = g \cdot (f \circ f')(h),$$

hence $f \circ f'$ is G–equivariant. Finally, f^{-1} is the G–equivariant map sending 1 to $f(1)^{-1}$ and we conclude that $\mathrm{Hom}_G(G,G)$ is a subgroup of Σ_G.

The final check is that the Yoneda bijection $G \to \mathrm{Hom}_G(G,G)$ sending g to $(-)\cdot g$ is a group homomorphism.[179] It is clear that it sends the identity to the identity and for any $g,h \in G$

$$(-)\cdot gh = ((-)\cdot g)\cdot h = ((-)\cdot h)\circ((-)\cdot g),$$

so this is a group homomorphism.

[179] isomorphism follows because it is a bijection.

Universality as Representability

It turns out the main concepts of the previous chapter and this one are closely related. In this section, we will explore how any universal property is equivalent to representability of some functor.

Definition 233 (Generalized diagonal functor)**.** Let $\mathbf{J}$ and $\mathbf{C}$ be categories the **generalized diagonal functor** $\Delta^{\mathbf{J}}_{\mathbf{C}} : \mathbf{C} \rightsquigarrow [\mathbf{J},\mathbf{C}]$ sends an object $X \in \mathbf{C}_0$ to the constant functor at X and a morphism $f : X \to Y \in \mathbf{C}_1$ to the natural transformation whose components are all $f : X \to Y$.

We have $\Delta^{\mathbf{J}}_{\mathbf{C}}(f) : X \Rightarrow Y$ because for any $j \in \mathbf{J}_1$, we have

$$\begin{tikzcd} X \arrow[r, "X(j)=\mathrm{id}_X"] \arrow[d, "f"'] & X \arrow[d, "f"] \\ Y \arrow[r, "Y(j)=\mathrm{id}_Y"'] & Y \end{tikzcd}$$

Remark 234. This is a generalization of the diagonal functor $\Delta_{\mathbf{C}} : \mathbf{C} \rightsquigarrow \mathbf{C} \times \mathbf{C}$ because with the isomorphism $[\mathbf{1}+\mathbf{1},\mathbf{C}] \cong \mathbf{C}\times\mathbf{C}$ described in Example 211, we can identify $\Delta_{\mathbf{C}}$ with $\Delta^{\mathbf{1}+\mathbf{1}}_{\mathbf{C}}$.

Proposition 235. *Let $F : \mathbf{J} \rightsquigarrow \mathbf{C}$ be a diagram. The limit of F exists if and only if there is an object $L \in \mathbf{C}_0$ such that $\mathrm{Nat}(\Delta^{\mathbf{J}}_{\mathbf{C}}(-),F) \cong \mathrm{Hom}_{\mathbf{C}}(-,L)$.*[180]

[180] Observe that

$$\mathrm{Nat}(\Delta^{\mathbf{J}}_{\mathbf{C}}(-),F) = \mathrm{Nat}(-,F)\circ\Delta^{\mathbf{J}}_{\mathbf{C}}.$$

Proof. First, we note that for any $X \in \mathbf{C}_0$, a natural transformation $\psi : \Delta^{\mathbf{J}}_{\mathbf{C}}(X) \Rightarrow F$ is a cone over F with tip X. Indeed, for any $j : A \to B \in \mathbf{J}_1$, the naturality square in (96) is commutative.

$$\begin{tikzcd} X \arrow[r, "X(j)=\mathrm{id}_X"] \arrow[d, "\psi_A"'] & X \arrow[d, "\psi_B"] \\ FA \arrow[r, "F(j)"'] & FB \end{tikzcd} \tag{96}$$

This is equivalent to $\{\psi_A : X \to FA\}_{A\in\mathbf{J}_0}$ being a cone over F. Furthermore, a morphism of cones $\phi \to \psi$ is a morphism f between the tips such that $\forall A \in \mathbf{J}_0, \phi_A = \psi_A \circ f$. By looking at (97), we see this condition is equivalent to $\phi = \psi \circ \Delta^{\mathbf{J}}_{\mathbf{C}}(f)$.

$$\begin{tikzcd} & Y \arrow[rr, "\mathrm{id}_Y"] \arrow[dl, "f"'] \arrow[dd, "\phi_A"] & & Y \arrow[dl, "f"'] \arrow[dd, "\phi_B"] \\ X \arrow[rr, "\mathrm{id}_X"] \arrow[dr, "\psi_A"'] & & X \arrow[dr, "\psi_B"'] & \\ & FA \arrow[rr, "F(j)"'] & & FB \end{tikzcd} \tag{97}$$

$(\Rightarrow)$ Let $\{\psi_A : L \to FA\}_{A\in\mathbf{J}_0}$ be the terminal cone over F and see it as a natural transformation $\psi : \Delta^{\mathbf{J}}_{\mathbf{C}}(L) \Rightarrow F$. We need to define a natural isomorphism $\mathrm{Nat}(\Delta^{\mathbf{J}}_{\mathbf{C}}(-),F) \cong \mathrm{Hom}_{\mathbf{C}}(-,L)$. Similarly to the proofs of the previous section, we will see that we only need to see where id_L is sent to and the rest of the natural transformation will *construct itself*. Our only choice for the cone corresponding to id_L is ψ (it is the only cone we know exists).

Indeed, for any $f : X \to L$ the naturality square in (98) means the cone corresponding to $f : X \to L$ is $\{\psi_A \circ f : X \to FA\}_{A\in\mathbf{J}_0}$ by starting with id_L in the top

$$\begin{tikzcd} \mathrm{Nat}(\Delta^{\mathbf{J}}_{\mathbf{C}}(L),F) \arrow[r, leftrightarrow] \arrow[d, "-\circ\Delta^{\mathbf{J}}_{\mathbf{C}}(f)"'] & \mathrm{Hom}_{\mathbf{C}}(L,L) \arrow[d, "-\circ f"] \\ \mathrm{Nat}(\Delta^{\mathbf{J}}_{\mathbf{C}}(X),F) \arrow[r, leftrightarrow] & \mathrm{Hom}_{\mathbf{C}}(X,L) \end{tikzcd} \tag{98}$$

right. Now, since ψ is the terminal cone, for any cone $\{\phi_A : X \to FA\}_{A\in \mathbf{J}_0}$, there is a unique morphism of cones $f : X \to L$ which satisfies $\forall A \in \mathbf{J}_0, \psi_A \circ f = \phi_A$. We conclude that $f \mapsto \psi \circ \Delta^{\mathbf{J}}_{\mathbf{C}}(f)$ is a natural isomorphism.

($\Leftarrow$) Let $\psi : \Delta^{\mathbf{J}}_{\mathbf{C}}(L) \Rightarrow F$ be the cone corresponding to $\mathrm{id}_L \in \mathrm{Hom}_{\mathbf{C}}(L, L)$ under the natural isomorphism, we will show it is terminal. By the commutativity of (98) and bijectivity of the horizontal arrows, for any cone $\phi : \Delta^{\mathbf{J}}_{\mathbf{C}}(X) \Rightarrow F$, there is a unique morphism $f : X \to L$ such that $\phi = \psi \circ \Delta^{\mathbf{J}}_{\mathbf{C}}(f)$. By the first paragraph of the proof, this is the unique morphism of cones showing ψ is terminal. $\square$

Adjunctions

Definition 236 (Adjunction)**.** Two functors $L : \mathbf{C} \rightsquigarrow \mathbf{D}$ and $R : \mathbf{D} \rightsquigarrow \mathbf{C}$ are **adjoint** if there exists two natural transformations $\eta : \mathrm{id}_{\mathbf{C}} \Rightarrow RL$ and $\varepsilon : LR \Rightarrow \mathrm{id}_{\mathbf{D}}$ called the **unit** and **counit** satisfying the **triangle identities** shown in (99) and (100).

$$\begin{tikzcd} L \arrow[r, "L\eta"] \arrow[dr, "\mathbb{1}_L"'] & LRL \arrow[d, "\varepsilon L"] \\ & L \end{tikzcd} \qquad (99) \qquad\qquad \begin{tikzcd} RLR \arrow[d, "R\varepsilon"'] & R \arrow[l, "\eta R"'] \arrow[dl, "\mathbb{1}_L"] \\ L & \end{tikzcd} \qquad (100)$$

Example 237 (Boring)**.** The identity functor on any category is self-adjoint, i.e.: $\mathrm{id}_{\mathbf{C}} \dashv \mathrm{id}_{\mathbf{C}}$. Both the unit and counit are $\mathbb{1}_{\mathrm{id}_{\mathbf{C}}}$. This adjunction follows from the next result as $\mathrm{id}_{\mathbf{C}}$ is its own inverse.

Proposition 238. *Let $L : \mathbf{C} \rightsquigarrow \mathbf{D}$ and $R : \mathbf{D} \rightsquigarrow \mathbf{C}$ be quasi-inverses, then $L \dashv R$ and $R \dashv L$.*

Proof. It is enough to show $L \dashv R$ as the definition of quasi-inverses is symmetric. □

Example 239. Recall from Exercise 142 the maybe functor $- + \mathbf{1}$. Denote $\mathbf{1} = \{*\}$ for the terminal object of **Set**. We consider a very similar functor $- + \mathbf{1} : \mathbf{Set} \rightsquigarrow \mathbf{Set}_*$ sending a set X to $(X + \mathbf{1}, *)$ and $f : X \to Y$ to $f + \mathrm{id}_{\mathbf{1}} : X + \mathbf{1} \to Y + \mathbf{1}$. In the other direction, we have the forgetful functor $U : \mathbf{Set}_* \rightsquigarrow \mathbf{Set}$ that forgets about the distinguished element of a pointed set. We claim that $- + \mathbf{1} \dashv U$.

First, for every set X, we need to define $\eta_X : X \to U((X + \mathbf{1}, *)) = X + \mathbf{1}$. The only obvious choice is to let η_X be the inclusion of X in $X + \mathbf{1}$ and one can check it makes η into a natural transformation $\mathrm{id}_{\mathbf{Set}} \Rightarrow U(- + \mathbf{1})$.

Second, for every pointed set (X, x), we need to define $\varepsilon_{(X,x)} : (X + \mathbf{1}, *) \to (X, x)$. Again, there is one clear choice, i.e.: acting like the identity on X and sending $*$ to x, we will denote $\varepsilon_{(X,x)} = [\mathrm{id}_X, * \mapsto x]$.

Finally, we need to check the triangle identities which we instantiate below.[181]

$$\begin{tikzcd} (X+\mathbf{1}, *) \arrow[r, "\eta_X + \mathrm{id}_{\mathbf{1}}"] \arrow[dr, "\mathrm{id}_{X+\mathbf{1}}"'] & ((X+\mathbf{1})+\mathbf{1}, \star) \arrow[d, "{[\mathrm{id}_{X+\mathbf{1}}, \star \mapsto *]}"] \\ & (X+\mathbf{1}, *) \end{tikzcd} \qquad (101) \qquad \begin{tikzcd} X \arrow[r, "\eta_X"] \arrow[dr, "\mathrm{id}_X"'] & X+\mathbf{1} \arrow[d, "{[\mathrm{id}_X, * \mapsto x]}"] \\ & X \end{tikzcd} \qquad (102)$$

Check η and ε are natural:

$$\begin{tikzcd} X \arrow[r, "\eta_X"] \arrow[d, "f"'] & X+\mathbf{1} \arrow[d, "f+\mathrm{id}_{\mathbf{1}}"] \\ Y \arrow[r, "\eta_Y"'] & Y+\mathbf{1} \end{tikzcd} \qquad \begin{tikzcd} (X,x) \arrow[d, "f"'] & (X+\mathbf{1}, *) \arrow[l, "\varepsilon_{(X,x)}"'] \arrow[d, "f+\mathrm{id}_{\mathbf{1}}"] \\ (Y,y) & (Y+\mathbf{1}, *) \arrow[l, "\varepsilon_{(Y,y)}"] \end{tikzcd}$$

181 When dealing with a set $(X + \mathbf{1}) + \mathbf{1}$, we will denote $*$ for the element of the inner $\mathbf{1}$ and $\star$ for the outer one.
In (102), $X = U(X, x)$.

We conclude that $-+\mathbf{1} \dashv U$. A good exercise in categorical thinking is to generalize this example to an arbitrary category $\mathbf{C}$ with binary coproducts and a terminal object.[182]

[182] See ... for a solution.

Definition 240 (Adjunction)**.** Two functors $L : \mathbf{C} \rightsquigarrow \mathbf{D}$ and $R : \mathbf{D} \rightsquigarrow \mathbf{C}$ are adjoint if there is a natural isomorphism[183]

$$\mathrm{Hom}_{\mathbf{C}}(-, R-) \cong \mathrm{Hom}_{\mathbf{D}}(L-, -).$$

[183] We use Remark 100 to define

$$\mathrm{Hom}_{\mathbf{C}}(-, R-) := \mathrm{Hom}_{\mathbf{C}}(-,-) \circ (\mathrm{id}_{\mathbf{C}^{\mathrm{op}}} \times R)$$
$$\mathrm{Hom}_{\mathbf{D}}(L-, -) := \mathrm{Hom}_{\mathbf{D}}(-,-) \circ (L^{\mathrm{op}} \times \mathrm{id}_{\mathbf{D}})$$

Less concisely, for any $X \in \mathbf{C}_0$ and $Y \in \mathbf{D}_0$, there is an isomorphism $\Phi_{X,Y} : \mathrm{Hom}_{\mathbf{C}}(X, RY) \cong \mathrm{Hom}_{\mathbf{D}}(LX, Y)$ such that for any $f : X \to X' \in \mathbf{C}_1$ and $g : Y \to Y' \in \mathbf{D}_1$ the following commutes. We split the naturality in two squares because we will often use one square on its own.[184]

[184] This is possible by Exercise 196.

$$\begin{tikzcd}
\mathrm{Hom}_{\mathbf{C}}(X', RY) \arrow[r, "-\circ f"] \arrow[d, leftrightarrow, "\Phi_{X',Y}"'] & \mathrm{Hom}_{\mathbf{C}}(X, RY) \arrow[r, "Rg\circ -"] \arrow[d, leftrightarrow, "\Phi_{X,Y}"'] & \mathrm{Hom}_{\mathbf{C}}(X, RY') \arrow[d, leftrightarrow, "\Phi_{X,Y'}"] \\
\mathrm{Hom}_{\mathbf{D}}(LX', Y) \arrow[r, "-\circ Lf"'] & \mathrm{Hom}_{\mathbf{D}}(LX, Y) \arrow[r, "g\circ -"'] & \mathrm{Hom}_{\mathbf{D}}(LX, Y')
\end{tikzcd} \tag{103}$$

Proposition 241. *Let $\mathbf{C} : L \dashv R : \mathbf{D}$ be adjoint functors and $X, Y \in \mathbf{D}_0$, if $X \times Y$ exists, then $R(X \times Y)$ with the projections $R(\pi_X)$ and $R(\pi_Y)$ is the product $R(X) \times R(Y)$. In other words, right adjoints preserve binary products.*[185]

[185] Dually, if $A, B \in \mathbf{C}_0$ and $A + B$ exists, then $L(A+B)$ with the coprojections $L(\kappa_A)$ and $L(\kappa_B)$ is the coproduct $L(A) \times L(B)$. In other words, left adjoints preserve binary coproducts.

Proof. Let $p_X : A \to RX$ and $p_Y : A \to RY$ be such that (104) commutes.

$$\begin{tikzcd}
 & A \arrow[dl, "p_X"'] \arrow[dr, "p_Y"] & \\
RX & R(X \times Y) \arrow[l, "R\pi_X"] \arrow[r, "R\pi_Y"'] & RY
\end{tikzcd} \tag{104}$$

We need to show there is a unique mediating morphism $A \to R(X \times Y)$. First, we will get rid of the applications of R at the bottom, in order to use the universal property of the product $X \times Y$. To do this, we apply L to (104) and use the counit $\varepsilon : LR \Rightarrow \mathrm{id}_{\mathbf{D}}$ to obtain (105).

$$\begin{tikzcd}
 & LA \arrow[dl, "Lp_X"'] \arrow[dr, "Lp_Y"] & \\
LRX \arrow[d, "\varepsilon_X"'] & LR(X \times Y) \arrow[l, "LR\pi_X"] \arrow[r, "LR\pi_Y"'] \arrow[d, "\varepsilon_{X\times Y}"'] & LRY \arrow[d, "\varepsilon_Y"'] \\
X & X \times Y \arrow[l, "\pi_X"] \arrow[r, "\pi_Y"'] & Y
\end{tikzcd} \tag{105}$$

The universal property of $X \times Y$ tells us there is a unique $! : LA \to X \times Y$ such that $\pi_X \circ ! = \varepsilon_X \circ Lp_X$ and $\pi_Y \circ ! = \varepsilon_Y \circ Lp_Y$. We claim that $!^{\mathrm{t}}$ is the mediating morphism of (104), i.e.: $R\pi_X \circ !^{\mathrm{t}} = p_X$ and $R\pi_Y \circ !^{\mathrm{t}} = p_Y$. Using the adjunction $L \dashv R$, we obtain the following commutative square.

$$\begin{tikzcd}
 & LA \arrow[dl, "Lp_X"'] \arrow[dr, "Lp_Y"] \arrow[dd, dashed, "!"] & \\
LRX \arrow[d, "\varepsilon_X"'] & & LRY \arrow[d, "\varepsilon_Y"'] \\
X & X \times Y \arrow[l, "\pi_X"] \arrow[r, "\pi_Y"'] & Y
\end{tikzcd}$$

$$\begin{tikzcd}
\mathrm{Hom}_{\mathbf{D}}(LA, X \times Y) \arrow[r, leftrightarrow] \arrow[d, "\pi_X\circ -"'] & \mathrm{Hom}_{\mathbf{C}}(A, R(X \times Y)) \arrow[d, "R\pi_X\circ -"] \\
\mathrm{Hom}_{\mathbf{D}}(LA, X) \arrow[r, leftrightarrow] & \mathrm{Hom}_{\mathbf{C}}(A, RX)
\end{tikzcd} \tag{106}$$

Now, starting with ! on the top left corner, we obtain the following derivation.

$$\begin{aligned} p_X &= {p_X}^{\mathsf{t}\mathsf{t}} \\ &= (\varepsilon_X \circ Lp_X)^{\mathsf{t}} \\ &= \pi_X \circ !^{\mathsf{t}} && \text{definition of } ! \\ &= R\pi_X \circ !^{\mathsf{t}} && \text{commutativity of (106)} \end{aligned}$$

$$\begin{tikzcd} & A \arrow[dl, "p_X"'] \arrow[d, "!^{\mathsf{t}}"] \arrow[dr, "p_Y"] & \\ RX & R(X \times Y) \arrow[l, "R\pi_X"] \arrow[r, "R\pi_Y"'] & RY \end{tikzcd} \tag{107}$$

Replacing X with Y in the previous argument shows $!^{\mathsf{t}}$ makes (107) commute. For the uniqueness, note that if $m : A \to R(X \times Y)$ can replace $!^{\mathsf{t}}$, then (108) commutes which implies by uniqueness of ! that $m^{\mathsf{t}} = \varepsilon_{X \times Y} \circ Lm = !$. Transposing yields $!^{\mathsf{t}} = m$.

$$\begin{tikzcd} & LA \arrow[dl, "Lp_X"'] \arrow[d, "Lm"] \arrow[dr, "Lp_Y"] & \\ LRX \arrow[d, "\varepsilon_X"'] & LR(X \times Y) \arrow[l, "LR\pi_X"] \arrow[r, "LR\pi_Y"'] \arrow[d, "\varepsilon_{X \times Y}"'] & LRY \arrow[d, "\varepsilon_Y"'] \\ X & X \times Y \arrow[l, "\pi_X"] \arrow[r, "\pi_Y"'] & Y \end{tikzcd} \tag{108}$$

□

Proposition 242. *Let $\mathbf{C} : L \dashv R : \mathbf{D}$ be adjoint functors and $g : X \to Y \in \mathbf{D}_1$ be an monomorphism, then $R(g)$ is monic. In other words, right adjoints preserve monomorphisms.*[186]

[186] Dually, if $f : A \to B \in \mathbf{C}_1$ is epic, then $L(f)$ is epic. In other words, left adjoints preserve epimorphisms.

Proof. Let $h_1, h_2 : Z \to R(X)$ be such that $R(g) \circ h_1 = R(g) \circ h_2$, we need to show that $h_1 = h_2$. Since $L \dashv R$, we have the following commutative square.

$$\begin{tikzcd} \mathrm{Hom}_{\mathbf{C}}(Z, RX) \arrow[r, leftrightarrow] \arrow[d, "Rg\circ-"'] & \mathrm{Hom}_{\mathbf{D}}(FZ, X) \arrow[d, "g\circ-"] \\ \mathrm{Hom}_{\mathbf{C}}(Z, RY) \arrow[r, leftrightarrow] & \mathrm{Hom}_{\mathbf{D}}(FZ, Y) \end{tikzcd} \tag{109}$$

Starting with h_1 and h_2 in the top left corner, we find that[187]

$$g \circ {h_1}^{\mathsf{t}} = Rg \circ h_1 = Rg \circ h_2 = g \circ {h_2}^{\mathsf{t}},$$

[187] The first and last equality follow from commutativity of (109) and the middle equality is a hypothesis.

which, by monicity of g implies ${h_1}^{\mathsf{t}} = {h_2}^{\mathsf{t}}$. This in turn means that $h_1 = h_2$ because $(-)^{\mathsf{t}}$ is a bijection. □

Theorem 243. *If $\mathbf{C} : L \dashv R : \mathbf{D}$ and $\mathbf{D} : L' \dashv R' : \mathbf{E}$ are two adjunctions, then $\mathbf{C} : L'L \dashv RR' : \mathbf{E}$ is an adjunction.*

Proof. Let η and ε be the unit and counit of the first adjunction and η' and ε' be the unit and counit of the second one. We define the following unit and counit for the composite adjunction:

$$\begin{aligned} \widehat{\eta} &= R\eta' L \cdot \eta : \mathrm{id}_{\mathbf{C}} \Rightarrow RR'L'L \\ \widehat{\varepsilon} &= \varepsilon' \cdot L'\varepsilon R' : L'LRR' \Rightarrow \mathrm{id}_{\mathbf{E}}. \end{aligned}$$

The following diagrams show the triangle identities.

$$
\begin{tikzcd}
L'L \arrow[r, "L'L\eta"] \arrow[rr, bend left, "L'L\hat{\eta}"] \arrow[dr, "\mathbb{1}_{L'L}"'] \arrow[ddrr, bend right, "\mathbb{1}_{L'L}"'] & L'LRL \arrow[r, "L'LR\eta' L"] \arrow[d, "L'\varepsilon L"] \arrow[d, phantom, "\text{(a)}", shift right=4ex] \arrow[dr, phantom, "\text{(b)}"] & L'LRR'L'L \arrow[d, "L'\varepsilon R'L'L"] \arrow[dd, bend left, "\hat{\varepsilon} L'L"] \\
& L'L \arrow[r, "L'\eta' L"] \arrow[dr, "\mathbb{1}_{L'L}"] \arrow[dr, phantom, "\text{(c)}", shift left=3ex] & L'R'L'L \arrow[d, "\varepsilon' L'L"] \\
& & L'L
\end{tikzcd}
\tag{110}
$$

Showing (110) commutes:

(a) Apply $L'(-)$ to the left triangle identity of η and ε.

(b) This is the commutative square in the definition of $L'(\varepsilon \diamond \eta')L$.

(c) Apply $(-)L$ to the left triangle identity of η' and ε'.

$$
\begin{tikzcd}
RR'L'LRR' \arrow[d, "RR'L'\varepsilon R'"] \arrow[dd, bend right, "RR'\hat{\varepsilon}"'] \arrow[dr, phantom, "\text{(a)}"] & RLRR' \arrow[l, "R\eta' LRR'"'] \arrow[d, "R\varepsilon R'"] & RR' \arrow[l, "\eta RR'"'] \arrow[ll, bend right, "\hat{\eta} RR'"'] \arrow[dl, "\mathbb{1}_{RR'}"'] \arrow[dl, phantom, "\text{(b)}", shift right=3ex] \arrow[ddll, bend left, "\mathbb{1}_{RR'}"] \\
RR'L'R' \arrow[d, "RR'\varepsilon'"] \arrow[dr, phantom, "\text{(c)}", shift left=3ex] & RR' \arrow[l, "R\eta' R'"'] \arrow[dl, "\mathbb{1}_{RR'}"] & \\
RR' & &
\end{tikzcd}
\tag{111}
$$

Showing (111) commutes:

(a) This is the commutative square in the definition of $R(\eta' \diamond \varepsilon)R'$.

(b) Apply $(-)R'$ to the right triangle identity of η and ε.

(c) Apply $R(-)$ to the right triangle identity of η' and ε'.

□

Monads and Algebras

POV: Category Theory

We will start from the concept of an adjunction which, as we hope was made clear in the previous chapter, is ubiquitous and powerful throughout mathematics. However, we will start with a great oversimplification; we will assume the categories concerned are posetal.

An adjunction between posets is more commonly called a Galois connection.

Definition 244 (Galois connection)**.** Let $(P, \leq)$ and $(Q, \sqsubseteq)$ be posets. A **Galois connection** between them is a pair of order-preserving functions $L : P \to Q$ and $R : Q \to P$ satisfying for any $p \in P$ and $q \in Q$, $L(p) \sqsubseteq q \iff p \leq R(q)$.

We are now interested in the composition $R \circ L$. It is also a monotonic function but the results about adjoints we have seen yield a couple of interesting properties. First, the existence of the unit $\eta : \mathrm{id}_P \Rightarrow RL$ means that for any $p \in P$, there is $\eta_p : p \to RL(p)$, so RL is extensive.[188] Second, the existence of the counit $\varepsilon : RL \Rightarrow \mathrm{id}_P$ means that for any $p \in P$, there is $R(\varepsilon_{L(p)}) : RLRL(p) \to RL(p)$ and $RL(\eta_p) : RL(p) \to RLRL(p)$, so RL is idempotent (i.e.: $\forall p \in P, RL(p) = RLRL(p)$). We say that RL is a closure operator.

[188] i.e.: $\forall p \in P, p \leq RL(p)$.

Definition 245 (Closure operator)**.** Let $(P, \leq)$ be a poset, a **closure operator** on P is a monotone extensive and idempotent function $c : P \to P$.

Example 246. We can give a very simple example hinting at the origins of the terminology. Consider the real numbers $\mathbb{R}$ with the standard topology.[189] We know that $(\mathcal{P}(\mathbb{R}), \subseteq)$ is a poset and we can define $c : \mathcal{P}(\mathbb{R}) \to \mathcal{P}(\mathbb{R})$ sending $U \subseteq \mathbb{R}$ to its closure $c(U)$ in the topological sense as in $c(U)$ is the set of limit points of U. Then, you probably have seen that for any $U, V \subseteq \mathbb{R}$, $U \subseteq V \implies c(U) \subseteq c(V)$, $U \subseteq c(U)$ and $c(U) = c(c(U))$, thus operation of closure is a closure operator.

[189] This is the topology induced by the standard metric $d(x, y) := |x - y|$.

We will generalize this discussion to arbitrary categories now. Let $L : \mathbf{C} \rightleftarrows \mathbf{D} : R$ be an adjoint pair of functors, we have two natural transformations $\eta : \mathrm{id}_{\mathbf{C}} \Rightarrow RL$ and $R\varepsilon L : RLRL \Rightarrow RL$. Recall these may interact well together via the zig-zag identities that we reformulate in (112) and (113) and we add to those the commutativity diagram (114) in the definition of $\varepsilon \diamond \varepsilon$.

$$\begin{tikzcd} L \arrow[r, Rightarrow, "L\eta"] \arrow[dr, Rightarrow, "\mathbb{1}_L"'] & LRL \arrow[d, Rightarrow, "\varepsilon L"] \\ & L \end{tikzcd} \quad (112) \qquad \begin{tikzcd} RLR \arrow[d, Rightarrow, "R\varepsilon"'] & R \arrow[l, Rightarrow, "\eta R"'] \arrow[dl, Rightarrow, "\mathbb{1}_L"] \\ L & \end{tikzcd} \quad (113) \qquad \begin{tikzcd} LRLR \arrow[r, Rightarrow, "\varepsilon LR"] \arrow[d, Rightarrow, "LR\varepsilon"'] & LR \arrow[d, Rightarrow, "\varepsilon"] \\ LR \arrow[r, Rightarrow, "\varepsilon"'] & \mathrm{id}_D \end{tikzcd} \quad (114)$$

With a bit of tinkering, we can make these diagrams all about the functor we are interested in, namely RL. Indeed, if we denote $M = RL$ and $\mu = R\varepsilon L$ before acting by R on the right of (112), L on the left of (113) and $R(\cdot)L$ on (114), we obtain the definition of a monad.

Definition 247 (Monad)**.** A **monad** is a triple comprised of an endofunctor $M : \mathbf{C} \rightsquigarrow \mathbf{C}$ and two natural transformations $\eta : \mathrm{id}_{\mathbf{C}} \Rightarrow M$ and $\mu : M^2 \Rightarrow M$ called the **unit** and **multiplication** respectively that make (115) and (116) commute in $[\mathbf{C}, \mathbf{C}]$.

$$\begin{tikzcd} M \arrow[r, Rightarrow, "M\eta"] \arrow[dr, Rightarrow, "\mathbb{1}_M"'] & M^2 \arrow[d, Rightarrow, "\mu"] & M \arrow[l, Rightarrow, "\eta M"'] \arrow[dl, Rightarrow, "\mathbb{1}_M"] \\ & M & \end{tikzcd} \quad (115) \qquad \begin{tikzcd} M^3 \arrow[r, Rightarrow, "M\mu"] \arrow[d, Rightarrow, "\mu M"'] & M^2 \arrow[d, Rightarrow, "\mu"] \\ M^2 \arrow[r, Rightarrow, "\mu"'] & M \end{tikzcd} \quad (116)$$

Examples 248. Our discussion above tells us that any adjoint pair $L \dashv R$ corresponds to a monad $(RL, \eta, R\varepsilon L)$, so all the examples of adjunctions you have seen correspond to suitable examples of monads. For instance, all closure operators are monads. Here, we describe three simple yet very useful examples and let you ponder on the adjunctions they might or might not originate from.

1. Suppose $\mathbf{C}$ has (binary) coproducts and a terminal object $\mathbf{1}$, then $(\cdot + \mathbf{1}) : \mathbf{C} \rightsquigarrow \mathbf{C}$ is a monad. We write inl^{X+Y} (resp. inr^{X+Y}) for the coprojection of X (resp. Y) into $X + Y$. First, note that for a morphism $f : X \to Y$,

$$f + \mathbf{1} = [\mathsf{inl}^{Y+\mathbf{1}} \circ f, \mathsf{inr}^{Y+\mathbf{1}}] : X + \mathbf{1} \to Y + \mathbf{1}.$$

The components of the unit are given by the coprojections, i.e.: $\eta_X = \mathsf{inl}^{X+\mathbf{1}} : X \to X + \mathbf{1}$, and the components of the multiplication are

$$\mu_X = [\mathsf{inl}^{X+\mathbf{1}}, \mathsf{inr}^{X+\mathbf{1}}, \mathsf{inr}^{X+\mathbf{1}}] : X + \mathbf{1} + \mathbf{1} \to X + \mathbf{1}.$$

Checking that (115) commutes, we have for any $X \in \mathbf{C}$:

$$\begin{aligned} \mu_X \circ (\eta_X + \mathbf{1}) &= [\mu_X \circ \mathsf{inl}^{(X+\mathbf{1})+\mathbf{1}} \circ \eta_X, \mu_X \circ \mathsf{inr}^{(X+\mathbf{1})+\mathbf{1}}] \\ &= [[\mathsf{inl}^{X+\mathbf{1}}, \mathsf{inr}^{X+\mathbf{1}}] \circ \mathsf{inl}^{X+\mathbf{1}}, \mathsf{inr}^{X+\mathbf{1}}] \\ &= [\mathsf{inl}^{X+\mathbf{1}}, \mathsf{inr}^{X+\mathbf{1}}] \\ &= \mathrm{id}_{X+\mathbf{1}} \\ &= [\mathsf{inl}^{X+\mathbf{1}}, \mathsf{inr}^{X+\mathbf{1}}] \\ &= \mu_X \circ \mathsf{inl}^{(X+\mathbf{1})+\mathbf{1}} \\ &= \mu_X \circ \eta_{X+\mathbf{1}} \end{aligned}$$

For (116), we have for any $X \in \mathbf{C}$:

$$\mu_X \circ (\mu_X + \mathbf{1}) = [\mu_X \circ \mathsf{inl}^{(X+\mathbf{1})+\mathbf{1}} \circ \mu_X, \mu_X \circ \mathsf{inr}^{(X+\mathbf{1})+\mathbf{1}}]$$

$$
\begin{aligned}
&= [[\mathsf{inl}^{X+\mathbf{1}}, \mathsf{inr}^{X+\mathbf{1}}] \circ \mu_X, \mathsf{inr}^{X+\mathbf{1}}] \\
&= [[\mathsf{inl}^{X+\mathbf{1}}, \mathsf{inr}^{X+\mathbf{1}}, \mathsf{inr}^{X+\mathbf{1}}], \mathsf{inr}^{X+\mathbf{1}}] \\
&= [\mu_X, \mathsf{inr}^{X+\mathbf{1}}] \\
&= [[\mathsf{inl}^{X+\mathbf{1}}, \mathsf{inr}^{X+\mathbf{1}}], \mathsf{inr}^{X+\mathbf{1}}, \mathsf{inr}^{X+\mathbf{1}}] \\
&= [\mu_X \circ \mathsf{inl}^{(X+\mathbf{1})+\mathbf{1}}, \mu_X \circ \mathsf{inr}^{(X+\mathbf{1})+\mathbf{1}}, \mu_X \circ \mathsf{inr}^{(X+\mathbf{1})+\mathbf{1}}] \\
&= \mu_X \circ \mu_{X+\mathbf{1}}
\end{aligned}
$$

2. The covariant powerset functor $\mathcal{P} : \mathbf{Set} \rightsquigarrow \mathbf{Set}$ is a monad with the following unit and multiplication:

$$\eta_X : X \to \mathcal{P}(X) = x \mapsto \{x\} \text{ and } \mu_X : \mathcal{P}(\mathcal{P}(X)) \to \mathcal{P}(X) = F \mapsto \bigcup_{s \in F} s.$$

Checking that (115), we have for any $S \subseteq \mathcal{P}(X)$,

$$
\begin{aligned}
\mu_X(\mathcal{P}(\eta_X)(S)) &= \mu_X\left(\{\{x\} \mid x \in S\}\right) \\
&= \bigcup_{x \in S} \{x\} \\
&= S \\
&= \bigcup \{S\} \\
&= \mu_X(\{S\}) \\
&= \mu_X(\eta_{\mathcal{P}(X)}(S))
\end{aligned}
$$

Checking that (116) commutes, we have for any $\mathcal{F} \in \mathcal{P}(\mathcal{P}(\mathcal{P}(X))$,

$$
\begin{aligned}
\mu_X(\mu_{\mathcal{P}(X)}(\mathcal{F})) &= \mu_X\left(\bigcup_{F \in \mathcal{F}} F\right) \\
&= \bigcup_{\substack{s \in \mathcal{P}(X) \\ \exists F \in \mathcal{F}, s \in F}} s \\
&= \{x \in X \mid \exists s \in \mathcal{P}(X), x \in s \text{ and } \exists F \in \mathcal{F}, s \in F\} \\
&= \bigcup_{F \in \mathcal{F}} \bigcup_{s \in F} s \\
&= \mu_X\left(\left\{\bigcup_{s \in F} s \mid F \in \mathcal{F}\right\}\right) \\
&= \mu_X(\mathcal{P}(\mu_X)(\mathcal{F}))
\end{aligned}
$$

3. The functor $\mathcal{D} : \mathbf{Set} \to \mathbf{Set}$ sends a set X to the set of finitely supported distributions on X, i.e.:

$$\mathcal{D}(X) := \{\varphi \in [0,1]^X \mid \sum_{x \in X} \varphi(x) = 1 \text{ and } \varphi(x) \neq 0 \text{ for finitely many } x\text{'s}\}.$$

It sends a function $f : X \to Y$ to the function between distributions

$$\lambda \varphi^{\mathcal{D}(X)}.\lambda y^Y.\varphi(f^{-1}(y)).$$

More verbosely, the weight of $\mathcal{D}(f)(\varphi)$ at point y is equal to the total weight of φ on the preimage of y under f. It is a monad with unit $\eta_X = x \mapsto \delta_x$, where δ_x is the Dirac distribution at x (all the weight is at x), and multiplication

$$\mu_X = \Phi \mapsto \lambda x^X. \sum_{\phi \in \mathrm{supp}(\Phi)} \Phi(\phi) \cdot \phi(x),$$

where $\mathrm{supp}(\Phi)$ is the support of Φ, i.e.: $\mathrm{supp}(\Phi) := \{\varphi \mid \Phi(\varphi) \neq 0\}$.

After looking long enough for adjunctions giving rise to the monads in Example 248, two questions dare to be asked. Does every monad arise from an adjunction? If yes, is that adjunction unique?

The second question might not be as natural to novices in category theory but it is almost as important as the first one. Indeed, uniqueness is a very strong property and if every monad had a unique corresponding adjunction, one might expect it to be fairly easy to find. This is part of the beauty of category theory. We are working with very little data M, η and μ so if it completely determined an adjunction L, R and $\mathrm{Hom}(L(-), -) \cong \mathrm{Hom}(-, R(-))$, it could not do so in a very convoluted way merely because there is not that many ways to manipulated the original data.

In any case, we will respectively give a positive and negative answer to these questions. Fortunately, while we might not benefit from the power of uniqueness, there are two special adjunctions arising from a monad whose descriptions are fairly straightforward. In the order we present them, the first is due to Kleisli and the second to Eilenberg and Moore. In the rest of this section, (M, η, μ) will be a monad on a category $\mathbf{C}$.

Kleisli Category $\mathbf{C}_M$

An intuitive way to think about monads is through *generalized elements*. Given an object $A \in \mathbf{C}_0$, we can view MA as extending A with more general or structured elements built from A.

In this picture, the morphisms $\eta_A : A \to MA$ give a way to understand anything inside A trivially as a general element of A. The morphisms $\mu_A : M^2A \to MA$ imply that higher order structures can collapsed so that generalized elements over generalized elements of A are generalized elements of A. The functoriality of M implies that the new structures in MA are somewhat independent of A. Indeed, for every morphisms $f : A \to B$, there is a morphism $Mf : MA \to MB$ which, by naturality of η ($Mf(\eta_A(-)) = \eta_B(f(-))$), acts just like f on the trivial generalization of elements in A. Commutativity of (115) says that the trivial generalization (there are two ways to do it) of a generalized element is indeed trivial as after collapsing via μ, we end up with what we started with. Finally, the associativity of μ (i.e.: commutativity of (116)) corresponds to the fact that in higher order of generalizations, one can collapse the structure at every level in any order and end up with the same thing.

Now, we can also consider *generalized morphisms*. Let us say we were given an ill-defined morphism $f : A \to B$ that sends some of the stuff in A outside of B. One way to fix this might be to consider general elements of B and see f as a morphism $A \to MB$. We will call such morphisms **Kleisli morphisms** and write $f : A \to MB$ or $f : A \nrightarrow B$.

With an arbitrary functor F, you might have a hard time to come up with a way to compose them Kleisli morphisms $A \to FB$ and $B \to FC$ or define the identity Kleisli morphism $A \to FA$, but the data of a monad lets you do just that. We end up with the category $\mathbf{C}_M$.

Definition 249 ($\mathbf{C}_M$)**.** Let $\mathbf{C}$ be a category and (M, η, μ) a monad on $\mathbf{C}$. The **Kleisli category** of M, denoted $\mathbf{C}_M$ has objects $\mathbf{C}_0$ and morphisms $f : A \to MB \in \mathbf{C}_1$. The identity for A is $\eta_A : A \to MA$ and composition is given by

$$g \circ_M f = \mu_C \circ Mg \circ f : A \nrightarrow C,$$

where $f : A \nrightarrow B$ and $g : B \nrightarrow C$.

Checking that this is indeed a category is left as an exercise and you should remark that it requires using all the properties M, η and μ satisfy.

As we have constructed $\mathbf{C}_M$ with objects and morphisms in $\mathbf{C}$, we would like to describe a forgetful functor $U_M : \mathbf{C}_M \rightsquigarrow \mathbf{C}$. A first guess to send objects A to themselves will fail when it is time to defining the action of U_M on morphisms for $f : A \nrightarrow B$ can send some stuff into generalized elements of B, and we have no way to get this back into B. Instead, we need to generalize everything before going into $\mathbf{C}$, namely, $U_M(A) = MA$ and $U_M f = \mu_B \circ Mf : MA \to MB$.

We can check that for any A, $U_M(\eta_A) = \mu_A \circ M(\eta_A) \overset{(115)}{=} \mathrm{id}_A$ and for any for any $f : A \nrightarrow B$ and $g : B \nrightarrow C$,

$$\begin{aligned}
U_M(g \circ_M f) &= U_M(\mu_C \circ Mg \circ f) \\
&= \mu_C \circ M(\mu_C \circ Mg \circ f) \\
&= \mu_C \circ M(\mu_C) \circ MMg \circ Mf \\
&= \mu_C \circ \mu_{MC} \circ MMg \circ Mf && \text{by (116)} \\
&= \mu_C \circ Mg \circ \mu_B \circ Mf && \text{by naturality of } \mu \\
&= U_M(g) \circ U_M(f).
\end{aligned}$$

We conclude that U_M is a functor.

In the opposite direction, we want a functor $F_M : \mathbf{C} \rightsquigarrow \mathbf{C}_M$. This time, we send A to itself because we can make any morphism $f : A \to B$ generalized by post-composing with η_B, namely $F_M(A) = A$ and $F_M(f) = \eta_B \circ f$ for any $f : A \to B$. For functoriality, $F_M(\mathrm{id}_A)$ is trivial as η_A is the identity on A in $\mathbf{C}_M$ and

$$\begin{aligned}
F_M(g \circ f) &= \eta_C \circ g \circ f \\
&= Mg \circ \eta_B \circ f && \text{by naturality of } \eta \\
&= Mg \circ \mu_B \circ M(\eta_B) \circ \eta_B \circ f && \text{by (115)} \\
&= \mu_C \circ MMg \circ M(\eta_B) \circ \eta_B \circ f && \text{by naturality of } \mu \\
&= \mu_C \circ M(\eta_C) \circ Mg \circ \eta_B \circ f && \text{by naturality of } \eta
\end{aligned}$$

$$= F_M(g) \circ_M F_M(f)$$

Before checking that this is indeed an adjunction, we need to make sure it composes to M, that is $U_M F_M = M$. On objects, it is clear. On a morphism $f : A \to B$, we have

$$U_M(F_M(f)) = U_M(\eta_B \circ f) = \mu_B \circ M(\eta_B) \circ Mf \overset{(115)}{=} Mf.$$

Let us now verify that $F_M \dashv U_M$. Let $A, B \in \mathbf{C}_0$ (we view B as an object of $\mathbf{C}_M$), we need to exhibit a natural isomorphism $\mathrm{Hom}_{\mathbf{C}_M}(F_M A, B) \cong \mathrm{Hom}_{\mathbf{C}}(A, U_M B)$. The isomorphism is clear as $U_M B = MB$, $F_M A = A$ and a morphism $A \to MB$ is precisely a Kleisli morphism $A \twoheadrightarrow B$, we will denote it id. For the naturality, we need to show the following square commutes for any $f : A' \to A$ and $g : B \twoheadrightarrow B'$.

$$\begin{tikzcd}
\mathrm{Hom}_{\mathbf{C}_M}(A,B) \arrow[r, "\mathrm{id}"] \arrow[d, "g \circ_M (-) \circ_M F_M f"'] & \mathrm{Hom}_{\mathbf{C}}(A, MB) \arrow[d, "U_M g \circ (-) \circ f"] \\
\mathrm{Hom}_{\mathbf{C}_M}(A',B') \arrow[r, "\mathrm{id}"'] & \mathrm{Hom}_{\mathbf{C}}(A', MB')
\end{tikzcd} \tag{117}$$

It follows from the following derivation starting with a morphism $k : A \to MB$.

$$\begin{aligned}
g \circ_M k \circ_M F_M f &= \mu_{B'} \circ M(g) \circ \mu_B \circ M(k) \circ \eta_A \circ f \\
&= \mu_{B'} \circ M(g) \circ \mu_B \circ \eta_{MB} \circ k \circ f && \text{by naturality of } \eta \\
&= \mu_{B'} \circ M(g) \circ \mathrm{id}_{MB} \circ k \circ f && \text{by (115)} \\
&= \mu_{B'} \circ M(g) \circ k \circ f \\
&= U_M g \circ k \circ f
\end{aligned}$$

Recall that we claimed (F_M, U_M) was special in some way and that this was the (informal) reason why it was relatively easy to find, the next proposition will make this precise.

Definition 250 (Adj_M)**.** Let $\mathbf{C}$ be a category and (M, η, μ) a monad on $\mathbf{C}$. The category of adjunctions inducing M is denoted Adj_M. Its objects are adjoint pairs (L, R) with $R \circ L = M$ whose unit is η and whose counit ε satisfies $R\varepsilon L = \mu$. Its morphisms $(L, R) \to (L', R')$ are functors K satisfying $K \circ L = L'$ and $R' \circ K = R$ as in (118).

$$\begin{tikzcd}
\mathbf{D} \arrow[rr, "K"] \arrow[dr, shift left, "R"] & & \mathbf{D}' \arrow[dl, shift left, "R'"] \\
& \mathbf{C} \arrow[ul, shift left, "L"] \arrow[ur, shift left, "L'"] \arrow[ul, phantom, "\top"] \arrow[ur, phantom, "\perp"] &
\end{tikzcd} \tag{118}$$

Let us first check that (F_M, U_M) induces (M, η, μ), we already know that $U_M F_M = M$, but it remains to find the unit and counit of this adjunction. We will need to use the correspondence between the definitions of adjoint pairs seen in the last lecture. Recall that the natural isomorphism $\mathrm{Hom}_{\mathbf{C}_M}(F_M -, -) \cong \mathrm{Hom}_{\mathbf{C}}(-, U_M -)$ is simply the identity function between these two sets. Hence, the unit of the adjunction is $\mathbf{C}_0 \ni A \mapsto \mathrm{id}(u_{\mathbf{C}_M}(F_M A)) = \eta_A$ and the counit is $(\mathbf{C}_M)_0 \ni A \mapsto \mathrm{id}(u_{\mathbf{C}}(U_M A)) =: \varepsilon_A$. Note that ε_A has type $MA \twoheadrightarrow A$. We check that $U_M(\varepsilon_{F_M A}) = \mu_A$:

$$U_M(\varepsilon_{F_M A}) = \mu_A \circ \varepsilon_{F_M A} = \mu_A \circ u_{\mathbf{C}}(MA) = \mu_A.$$

Proposition 251. *The adjunction (F_M, U_M) is initial in* Adj_M.

Proof. Let $\mathbf{C} : L \dashv R : \mathbf{D} \in \mathrm{Adj}_M$ with unit η and counit ε, we claim there is a unique functor $K : \mathbf{C}_M \rightsquigarrow \mathbf{D}$ satisfying $K \circ F_M = L$ and $R \circ K = U_M$ as in (119).

$$\begin{tikzcd} \mathbf{C}_M \arrow[rr, dashed, "K"] \arrow[dr, shift left, "U_M"] \arrow[dr, phantom, "\top"] & & \mathbf{D} \arrow[dl, shift left, "R"] \\ & \mathbf{C} \arrow[ul, shift left, "F_M"] \arrow[ur, shift left, "L"] \arrow[ur, phantom, "\dashv"] & \end{tikzcd} \qquad (119)$$

On objects, K is determined by $KA = KF_MA = LA$. To a morphism $f : A \nrightarrow B$, we need to assign a morphism in $Kf \in \mathrm{Hom}_{\mathbf{D}}(LA, LB)$. Denote $\Phi_{A,B} : \mathrm{Hom}_{\mathbf{C}_M}(F_MA, B) \cong \mathrm{Hom}_{\mathbf{C}}(A, U_MB)$ and $\Psi_{A,B} : \mathrm{Hom}_{\mathbf{D}}(LA, B) \cong \mathrm{Hom}_{\mathbf{C}}(A, RB)$ be given by the adjunctions $F_M \dashv U_M$ and $L \dashv R$ respectively. Using the fact that $RK = U_M$, we obtain the following commutative diagram.

$$\begin{tikzcd} \mathrm{Hom}_{\mathbf{C}_M}(F_MA, A) \arrow[r, "\Phi_{A,A}"] \arrow[d, mapsto, "f \circ_M (-)"] & \mathrm{Hom}_{\mathbf{C}}(A, U_MA) \arrow[r, equal] \arrow[d, mapsto, "U_Mf \circ (-)"] & \mathrm{Hom}_{\mathbf{C}}(A, RLA) \arrow[d, mapsto, "RKf \circ (-)"] & \mathrm{Hom}_{\mathbf{D}}(LA, LA) \arrow[l, "\Psi_{A,LA}"'] \arrow[d, mapsto, "Kf \circ (-)"] \\ \mathrm{Hom}_{\mathbf{C}_M}(F_MA, B) \arrow[r, "\Phi_{A,B}"'] & \mathrm{Hom}_{\mathbf{C}}(A, U_MB) \arrow[r, equal] & \mathrm{Hom}_{\mathbf{C}}(A, RLB) & \mathrm{Hom}_{\mathbf{D}}(LA, LB) \arrow[l, "\Psi_{A,LB}"] \end{tikzcd}$$

Because these adjunctions both have the same unit, we have $\Phi_{A,A}(u_{\mathbf{C}_M}(F_MA)) = \eta_A = \Psi_{A,A}(u_{\mathbf{D}}(LA))$. Therefore, if we start with $\eta_A \in \mathrm{Hom}_{\mathbf{C}_M}(F_MA, A)$ and follow the diagram, we infer that $Kf = \Psi^{-1}_{A,LB}(\Phi_{A,B}(f))$. Using the fact that $\Phi_{A,B}(f) = f$ and $\Psi^{-1}_{A,LB}(f) = \varepsilon_{LB} \circ Lf$ (follow the proof of the Yoneda lemma), we end up with $Kf = \varepsilon_{LB} \circ Lf$ and we can verify it is indeed functorial.

$$K(u_{\mathbf{C}_M}(A)) = K(\eta_A) = \varepsilon_{LB} \circ L(\eta_A) \overset{\text{zig-zag}}{=} \mathrm{id}_A$$

$$\begin{aligned} K(g \circ_M f) &= K(\mu_C \circ RLg \circ f) \\ &= \varepsilon_{LC} \circ L(\mu_C) \circ LRLg \circ Lf \\ &= \varepsilon_{LC} \circ LR\varepsilon_{LC} \circ LRLg \circ Lf && \text{by hypothesis on } \varepsilon \\ &= \varepsilon_{LC} \circ \varepsilon_{LRLC} \circ LRLg \circ Lf && \text{by naturality of } \varepsilon \\ &= \varepsilon_{LC} \circ Lg \circ \varepsilon_{LB} \circ Lf && \text{by naturality of } \varepsilon \\ &= Kg \circ Kf \end{aligned}$$

$\square$

Eilenberg-Moore Category $\mathbf{C}^M$

For the second solution to the problem of finding an adjunction inducing a given monad, we look at the more structural side of monads.

Definition 252 (M–algebra)**.** Let (M, η, μ) be a monad, an **Eilenberg-Moore algebra** for M or simply M**–algebra** is a pair (A, α) consisting of an object $A \in \mathbf{C}_0$ and morphism $\alpha : MA \to A$ such that (120) and (121) commute.

$$\begin{tikzcd} A \arrow[r, "\eta_A"] \arrow[dr, "\mathrm{id}_A"'] & MA \arrow[d, "\alpha"] \\ & A \end{tikzcd} \quad (120) \qquad \begin{tikzcd} M^2A \arrow[r, "\mu_A"] \arrow[d, "M(\alpha)"'] & MA \arrow[d, "\alpha"] \\ MA \arrow[r, "\alpha"'] & A \end{tikzcd} \quad (121)$$

We will often denote an M–algebra using only its underlying set or morphism.

Definition 253 (M–algebra homomorphism)**.** Given two M–algebras (A, α) and (B, β), an M–algebra homomorphism $f : (A, \alpha) \to (B, \beta)$ is a morphism $h : A \to B$ making (134) commute.

$$\begin{tikzcd} MA \arrow[r, "Mh"] \arrow[d, "\alpha"'] & MB \arrow[d, "\beta"] \\ A \arrow[r, "h"'] & B \end{tikzcd} \quad (122)$$

For a monad M, the category of M–algebras is called the **Eilenberg-Moore** category of M and denoted $\mathbf{C}^M$, composition and identities are induced by the composition and identities in $\mathbf{C}$. Again, since $\mathbf{C}^M$ was built from objects and morphisms in $\mathbf{C}$, there is an obvious candidate for a forgetful functor $U^M : \mathbf{C}^M \rightsquigarrow \mathbf{C}$ sending an M–algebra (A, α) to its underlying object A and a homomorphism to its underlying morphism. Next, we need a left adjoint to U^M, $F^M : \mathbf{C} \rightsquigarrow \mathbf{C}^M$. Since we want $U^M F^M = M$ to hold, F^M must send $A \in \mathbf{C}_0$ to an M–algebra over MA and $h \in \mathbf{C}_1$ to Mh. There is only one choice given to us by the data of M, that is, $F^M A = (MA, \mu_A)$ and it turns out naturality of μ_A yields

$$\begin{tikzcd} M^2A \arrow[r, "M^2h"] \arrow[d, "\mu_A"'] & M^2B \arrow[d, "\mu_B"] \\ MA \arrow[r, "Mh"'] & MB \end{tikzcd}, \quad (123)$$

which implies Mh is indeed an M–algbera homomorphism. Let us now show that $F^M \dashv U^M$ with unit η and counit ε satisfying $U^M \varepsilon F^M = \mu$.

We want to exhibit an isomorphism $\mathrm{Hom}_{\mathbf{C}^M}(\mu_A, \beta) \cong \mathrm{Hom}_{\mathbf{C}}(A, B)$ natural in A and $\beta : MB \to B$. In the forward direction, we send $h : MA \to B$ to $h \circ \eta_A : A \to B$ which ensures the unit of this adjunction is η. In the backwards direction, we send $h : A \to B$ to $\beta \circ Mh$ which is a homomorphism by the following diagram where (a) commutes by naturality of μ and (b) by β being an M–algebra.

$$\begin{tikzcd} M^2A \arrow[r, "M^2h"] \arrow[d, "\mu_A"'] \arrow[dr, phantom, "\text{(a)}"] & M^2B \arrow[r, "M\beta"] \arrow[d, "\mu_B"] \arrow[dr, phantom, "\text{(b)}"] & MB \arrow[d, "\beta"] \\ MA \arrow[r, "Mh"'] & MB \arrow[r, "\beta"] & B \end{tikzcd} \quad (124)$$

One can easily check that these operations are inverses of each other. Next, we turn to naturality. Let $f : A' \to A \in \mathbf{C}_1$ and $g : (B, \beta) \to (B', \beta') \in \mathbf{C}_1^M$, we claim that

(125) commutes.

$$\begin{tikzcd}
\mathrm{Hom}_{\mathbf{C}^M}(\mu_A, \beta) \arrow[r, "(-)\circ\eta_A"] \arrow[d, "g\circ(-)\circ Mf"'] & \mathrm{Hom}_{\mathbf{C}}(A, B) \arrow[d, "g\circ(-)\circ f"] \\
\mathrm{Hom}_{\mathbf{C}^M}(\mu_{A'}, \beta') \arrow[r, "(-)\circ\eta_{A'}"'] & \mathrm{Hom}_{\mathbf{C}}(A', B')
\end{tikzcd} \tag{125}$$

Starting with a homomorphism $h : (MA, \mu_A) \to (B, \beta)$ in the top-left object, we need to show $g \circ h \circ Mf \circ \eta_{A'} = g \circ h \circ \eta_A \circ f$ which holds by naturality of η.

We find the counit of this adjunction is $\varepsilon_\alpha = \alpha : (MA, \mu_A) \to (A, \alpha)$ which is a homomorphism because (121) commutes and we verify

$$U^M(\varepsilon_{F^M A}) = U^M(\varepsilon_{\mu_A}) = U^M(\mu_A) = \mu_A.$$

Dually to Proposition 251, we show that this adjunction is special in a precise way.

Proposition 254. *The adjunction* (F^M, U^M) *is terminal in* Adj_M.

Proof. Let $\mathbf{C} : L \dashv R : \mathbf{D} \in \mathrm{Adj}_M$ with unit η and counit ε, we claim there is a unique functor $K : \mathbf{D} \rightsquigarrow \mathbf{C}^M$ satisfying $K \circ L = F^M$ and $U^M \circ K = R$ as in (126).

$$\begin{tikzcd}
\mathbf{D} \arrow[rr, dashed, "K"] \arrow[dr, shift left, "R"] \arrow[dr, phantom, "\top"] & & \mathbf{C}^M \arrow[dl, shift left, "U^M"] \arrow[dl, phantom, "\perp"] \\
& \mathbf{C} \arrow[ul, shift left, "L"] \arrow[ur, shift left, "F^M"] &
\end{tikzcd} \tag{126}$$

As before, we can determine K by the equation $U^M K = R$ which means it sends $A \in \mathbf{D}_0$ to an algebra on RA and $f : A \to B \in \mathbf{D}_1$ to an algebra homomorphism $Rf : KA \to KB$. The only missing piece of this puzzle is the algebra structure on KA. The only clue we have is the fact that Rf is a homomorphism so is $KA = \alpha$ and $KB = \beta$, then (127) commutes.

$$\begin{tikzcd}
MRA \arrow[r, "MRf"] \arrow[d, "\alpha"'] & MRB \arrow[d, "\beta"] \\
RA \arrow[r, "Rf"'] & RB
\end{tikzcd} \tag{127}$$

Replacing M with RL, we recognize this as the naturality diagram for ε acted on by $R(\cdot)$. Hence, we find $KA = (RA, R\varepsilon_A)$ is a suitable candidate after showing it is indeed an M–algebra. For the unit diagram, we have $R\varepsilon_A \circ \eta_A = \mathrm{id}_A$ by a zig-zag identity. For the multiplication diagram, we have

$$R\varepsilon_A \circ \mu_A = R\varepsilon_A \circ R\varepsilon_{LA} = R(\varepsilon_A \circ \varepsilon_{LA}) = R(\varepsilon_A \circ LR(\varepsilon_A)) = R\varepsilon_A \circ MR\varepsilon_A.$$

As K acts like R on morphisms, it is obviously functorial. We leave the proof that K is unique as an exercise as the argument should be somewhat similar to what we have done in the dual proposition. $\square$

In summary, you can keep in mind the following diagram.

$$
\begin{tikzcd}
& \mathbf{D} \arrow[d, shift left=2, "R"] \arrow[d, phantom, "\dashv"] \arrow[dr, dashed] & \\
\mathbf{C}_M \arrow[ur, dashed] \arrow[r, shift left=2, "U_M"] \arrow[r, phantom, "\top"] & \mathbf{C} \arrow[l, shift left=2, "F_M"] \arrow[u, shift left=2, "L"] \arrow[r, shift left=2, "F^M"] \arrow[r, phantom, "\perp"] & \mathbf{C}^M \arrow[l, shift left=2, "U^M"]
\end{tikzcd}
\tag{128}
$$

One last thing we want to mention is the embedding of the Kleisli category inside the Eilenberg-Moore category. After a bit more work, we can infer from the discussion above that the unique morphism of adjunctions $K : \mathbf{C}_M \rightsquigarrow \mathbf{C}^M$ is a fully faithful functor sending an object A in $\mathbf{C}_M$ to the algebra (MA, μ_A), called the **free algebra** on A, and a Kleisli morphism $f : A \twoheadrightarrow B$ to the homomorphism $\mu_B \circ Mf : (A, \mu_A) \to (B, \mu_B)$.

POV: Universal Algebra

In this section, we will highlight the link between algebraic structures as you have encountered them in other classes with the Eilenberg-Moore algebras discussed above. We will only work over the category **Set**. We start by developing an example.

Example 255 ($\mathcal{P}_{\mathrm{ne}}$)**.** Consider the non-empty finite powerset functor $\mathcal{P}_{\mathrm{ne}}$ sending X to $\{S \in \mathcal{P}(X) \mid S \text{ is finite and non-empty}\}$. The same unit and multiplication defined for $\mathcal{P}$ make $\mathcal{P}_{\mathrm{ne}}$ into a monad. A $\mathcal{P}_{\mathrm{ne}}$–algebra is a function $\alpha : \mathcal{P}_{\mathrm{ne}}(A) \to A$ satisfying the equations $\alpha\{a\} = a$ and $\alpha(\mathcal{P}_{\mathrm{ne}}(\alpha)(S)) = \alpha(\bigcup S)$. From this, we can extract a binary operation $\oplus_\alpha : A \times A \to A$ by defining $x \oplus_\alpha y = \alpha\{x, y\}$. This operation is clearly commutative and idempotent, i.e.: $x \oplus_\alpha y = y \oplus_\alpha y$ and $x \oplus_\alpha x = x$, but it is also associative by the following derivation.

$$
\begin{aligned}
(x \oplus_\alpha y) \oplus_\alpha z &= \alpha\{x, y\} \oplus_\alpha z \\
&= \alpha\{\alpha\{x, y\}, z\} \\
&= \alpha\{\alpha\{x, y\}, \alpha\{z\}\} \\
&= \alpha\{\mathcal{P}_{\mathrm{ne}}\alpha\{\{x, y\}, \{z\}\}\} \\
&= \alpha\{\mu_A\{\{x, y\}, \{z\}\}\} \\
&= \alpha\{x, y, z\}.
\end{aligned}
$$

Since a $\mathcal{P}_{\mathrm{ne}}$–algebra homomorphism $h : (A, \alpha) \to (B, \beta)$ commutes with α and β it also commutes with $\oplus_\alpha$ and $\oplus_\beta$.

Conversely, if $(A, \oplus)$ is an idempotent, associative and commutative binary operation on A, we can define $\alpha_\oplus$ on non-empty finite sets of A by iterating $\oplus$. Namely,

$$
\alpha_\oplus\{x\} = x \oplus x \quad \text{and} \quad \alpha_\oplus\{x_1, \dots, x_n\} = x_1 \oplus x_2 \oplus \cdots \oplus x_n.
$$

It is well-defined by associativity and commutativity and we can check that it is the inverse of the operation described in the previous paragraph. That is to say, we can check that $\alpha_{\oplus_\alpha} = \alpha$ and $\oplus_{\alpha_\oplus} = \oplus$. For the former, it is clear for singleton sets and for any $n > 1$, we have the following derivation.

$$\begin{aligned}
\alpha_{\oplus_\alpha}\{x_1, \dots, x_n\} &= x_1 \oplus_\alpha \cdots \oplus_\alpha x_n \\
&= \alpha\{x_1, x_2 \oplus_\alpha \cdots \oplus_\alpha x_n\} \\
&= \vdots \\
&= \alpha\{x_1, \alpha\{x_2, \alpha\{\cdots, \alpha\{x_n\}\}\}\} \\
\text{using } \alpha \circ \mathcal{P}_{\mathrm{ne}}(\alpha) = \alpha \circ \mu_A &= \alpha\{x_1, x_2, \alpha\{\cdots, \alpha\{x_n\}\}\} \\
&= \vdots \\
&= \alpha\{x_1, \dots, x_n\}
\end{aligned}$$

For the latter, we have

$$x \oplus_{\alpha_\oplus} y = \alpha_\oplus\{x, y\} = x \oplus y.$$

A set equipped with an idempotent, commutative and associative binary operation is called a **(join/meet)-semilattice** and we have shown above that $\mathcal{P}_{\mathrm{ne}}$–algebras are in correspondence with semilattices. Through the introduction of basic notions in universal algebra, we will explain how this correspondence is functorial and generalize the core idea behind it.

Definition 256 (Algebraic theory)**.** An **algebraic signature** is a set Σ of operation symbols along with arities in $\mathbb{N}$, we denote $f : n \in \Sigma$ for an n–ary operation f in Σ. Given a set X, one constructs the set of **Σ–terms** with variables in X, denoted $T_\Sigma(X)$ by iterating operations symbols:

$$\forall x \in X, x \in T_\Sigma(X)$$
$$\forall t_1, \dots, t_n \in T_\Sigma(X), f : n \in \Sigma, f(t_1, \dots, t_n) \in T_\Sigma(X).$$

An **equation** (sometimes **axiom**) E over Σ is a pair of Σ–terms over a set of dummy variables. We will call the tuple (Σ, E) an **algebraic theory**.

Definition 257 ((Σ, E)–algebras)**.** Given a signature Σ and a set of equations E over this signature, a **(Σ, E)–algebra** is a set A along with operations $f^A : A^n \to A$ for all $f : n \in \Sigma$ such that the pair of terms in E are always equal when the operation symbols and dummy variables are instantiated in A. We usually denote Σ^A for the set operations f^A.

Examples 258. As is suggested by the terminology, the common algebraic structures can be define with simple algebraic theories.

1. We can define a monoid as an algebra for the signature $\{\cdot : 2, 1 : 0\}$ and the equations $x \cdot (y \cdot z) = (x \cdot y) \cdot z$, $1 \cdot x = x$, $x \cdot 1 = x$.
2. Adding the unary operation $(-)^{-1}$ and the equations $x \cdot x^{-1} = 1$ and $x^{-1} \cdot x = 1$, we obtain the theory of groups.
3. Adding the equation $x \cdot y = y \cdot x$ yields the theory of abelian groups.

4. With the signature $\{+ : 2, \cdot : 2, 1 : 0, 0 : 0\}$, we can add the abelian group equations for the operation $+$ (identity is 0), the monoid equations for $\cdot$ (identity is 1) and the distributivity equation $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$ and thus obtain the theory of rings.

We also have homomorphisms.

Definition 259 ((Σ, E)–algebra homomorphisms)**.** Given two (Σ, E)–algebras A and B, a **homomorphism** between them is a map $h : A \to B$ commuting with all operations in Σ, that is $\forall f : n \in \Sigma, h \circ f^A = f^B \circ h^n$.

The category of (Σ, E)–algebras and their homomorphisms (with the obvious composition and identities) is denoted $\mathrm{Alg}(\Sigma, E)$.

Example 260 ($\Sigma_{\mathrm{S}}, E_{\mathrm{S}}$)**.** Recall in Example 255 that $\mathcal{P}_{\mathrm{ne}}$–algebras correspond to semilattices. Formally, the theory of semilattices has a single binary operation $\Sigma_{\mathrm{S}} = \{\oplus : 2\}$ satisfying the following equations in E_{S}:

$$
\begin{aligned}
x \oplus x &= x && I \\
x \oplus y &= y \oplus x && C \\
(x \oplus y) \oplus z &= x \oplus (y \oplus z). && A
\end{aligned}
$$

Up to a couple of missing functoriality arguments, we have shown that the categories $\mathbf{Set}^{\mathcal{P}_{\mathrm{ne}}}$ and $\mathrm{Alg}(\Sigma_{\mathrm{S}}, E_{\mathrm{S}})$ are isomorphic. We say that $(\Sigma_{\mathrm{S}}, E_{\mathrm{S}})$ is an **algebraic presentation** of the monad $\mathcal{P}_{\mathrm{ne}}$.

It turns out all algebraic theories present at least one monad.

Definition 261 (Term monad)**.** Let (Σ, E) be an algebraic theory, one can assign to any set X, the set $T_{\Sigma,E}(X)$ of terms in $T_\Sigma(X)$ modulo the equations in E. This can be extended to functions $f : X \to Y$, by variable substitution, i.e.: $T_\Sigma(f)$ acts on a term t by replacing all occurrences of $x \in X$ by $f(x) \in Y$ and $T_{\Sigma,E}(f)$ acts on equivalence classes by $[t] \mapsto [T_\Sigma(f)(t)]$. We obtain a functor $T_{\Sigma,E}$ on which we can put a monad structure.

The unit map is obvious because any element of X is a Σ–term, thus $\eta_X : X \to T_{\Sigma,E}(X)$ maps x to the equivalence class containing the term x. The multiplication map is derived from the fact that applying operations in Σ to Σ–terms yields Σ–terms. More explicitly, μ_X is a flattening operation defined recursively by

$$
\begin{gathered}
\forall t \in T_\Sigma(X), \mu_X([[t]]) = [t] \\
\forall f : n \in \Sigma, t_1, \dots, t_n \in T_\Sigma T_{\Sigma,E}(X), \mu_X([f(t_1, \dots, t_n)]) = [f(\mu_X([t_1]), \dots, \mu_X([t_n]))]
\end{gathered}
$$

One can show that $\mathbf{Set}^{T_{\Sigma,E}}$ is the category of (Σ, E) algebras.

Unfortunately, the terms monads are not very simple to work with and it is often desirable to find other simpler monads which are presented by the same theory or conversely to find an algebraic presentation for a given monad.

Examples 262. 1. The algebraic theory for presenting $\mathcal{D}$ is called the theory of **convex algebras** and is denoted $(\Sigma_{\mathsf{CA}}, E_{\mathsf{CA}})$, it consists of a binary operation $+_p : 2$ for any $p \in (0,1)$ which is meant to represent a choice between the two terms in the operation, the left one being chosen with probability p and the second one with probability $1 - p$. There are three equations in the theory that morally ensure that terms representing the same choice are equal.

$$x +_p x = x \qquad I_p\text{: idempotence}$$
$$x +_p y = y +_{\overline{p}} x \qquad C_p\text{: skew-commutativity}$$
$$(x +_q y) +_p z = x +_{pq} (y +_{\frac{p\bar{q}}{\overline{pq}}} z) \qquad A_p\text{: skew-associativity}$$

These equations are necessary for every distribution in $\mathcal{D}X$ to correspond uniquely to an equivalence class in $T_{\Sigma_{\mathsf{CA}}, E_{\mathsf{CA}}}(X)$.

2. The monad $(\cdot + \mathbf{1})$ is particular because it is really simple and combines very well with other monads.

   Proposition 263. *For any monad M, there is a monad structure on the composition $M(\cdot + \mathbf{1})$. Moreover, if M is presented by (Σ, E) the monad $M(\cdot + \mathbf{1})$ is presented by $(\Sigma \cup \{* : 0\}, E)$, that is, the new theory only has an additional constant which is neutral with respect to the equations.*

   We often qualify theories with an added constant as **pointed**. For instance, the theories presented by $\mathcal{P}_{\text{ne}}(\cdot + \mathbf{1})$ and $\mathcal{D}(\cdot + \mathbf{1})$ are those of **pointed semilattices** (PS) and **pointed convex algebras** (PCA) respectively.

Remark 264 (Lawvere's way). There is another way to do universal algebra *more categorically* still very much linked to monads: Lawvere theories. Algebras over a Lawvere theory are defined more abstractly using the categorical language and, on this account, they enjoy straightforward generalization through enrichment or lifting to higher order categories.

POV: Computer Programs

In this section, we will develop on an original idea by Eugenio Moggi that monads are suitable models for a general notion of *computation*. In the sequel, we will use the terms *type* and *set* interchangeably.

Moggi gave a justification for using monads in computer science (particularly in programming semantics) via the informal intuition of *computational types*. For a type A, the computational type of A should contain all computations which return a value of type A. It is intended for the interpretation of *computation* to be made explicit by an instance of a monad. In most cases, it can be thought of as a piece of code which returns some value, but for now, we start by building the intuition in an abstract sense.

Let MA denote the computational type of A and MMA the computational type of MA, that is computations returning values which are themselves computations of type A. The following items should coincide with our intuition of computation.

1. For any $x \in A$, there is a trivial computation $\mathsf{return}\ x \in MA$.

2. For any $C \in MMA$, we can reduce C to $\mathsf{flatten}(C) \in MA$ which executes C and the computation returned by C to obtain a final return value of type A.

3. If $C \in MA$, then $\mathsf{flatten}(\mathsf{return}\ C) = C$.

4. If $C \in MA$ and $C' \in MMA$ does the same computation as C but instead of returning a value x, it returns the computation $\mathsf{return}\ x$, then $\mathsf{flatten}(C') = C$.

5. If $MMMA$ is the computational type of MMA and $C \in MMMA$, then there are two ways to flatten C. First, there is the computation C_1 which executes C and executes the returned computation (of type MMA) to obtain a final value of type MA, hence $C_1 \in MMA$ and $\mathsf{flatten}(C_1) \in MA$. Second, C_2 executes C and flattens the returned computation to obtain a final value of type MA, C_2 is also of type MMA and $\mathsf{flatten}(C_2) \in MA$. These two operations should yield the same result.

Now, a monad M is a description of computational types that is general, namely, for any type A, the monad M gives a type MA behaving as expected. You can check that $x \mapsto \mathsf{return}\ x$ is the unit of this monad and $\mathsf{flatten}$ is the multiplication.

Examples 265. Here, we list more examples commonly used in computer science.

List monad: For any set X, let $L(X)$ denote the set of all finite lists whose elements are chosen in X. This is a functor that sends a function $f : X \to Y$ to its extension on lists $L(f) : L(X) \to L(Y)$ which applies f to all elements on the list (in lots of programming languages, one writes $L(f) := \mathsf{map}(f, -)$).Then, we can put a monad structure on L. The unit maps send an element $x \in X$ to the list containing only that element: $\eta_X = x \mapsto [x]$. The multiplication maps concatenate all the lists in a lists of lists: $\mu_X = [\ell_1, \dots, \ell_n] \mapsto \ell_1\ell_2 \cdots \ell_n$. It is easy to check diagrams (115) to (116) commute.

Termination: In order to model computations that might terminate with no output, the monad $(\cdot + \mathbf{1})$ is often used. For any type X, the type $X + \mathbf{1}$ has all the values of type X and an additional termination value denoted $*$. The behavior of the unit and multiplication of the monad can be interpreted as the fact that the stage of the computation that leads to a termination is irrelevant. This monad is also known as the Maybe monad.

Non-deterministic choice: The model for nondeterministic choice is given by the monad $\mathcal{P}_{\mathrm{ne}}$. The elements of $S \in \mathcal{P}_{\mathrm{ne}}(X)$ are seen as the possible outcomes of a nondeterministic choice. The unit is basically viewing a deterministic choice as a nondeterministic choice. The multiplication reduces the number of choices without changing the behavior. For instance, consider a process that nondeterministically chooses between two boxes containing two coins each and then chooses a coin in the box. By simply observing the final choice, we would not be able to distinguish it from a process that nondeterministically chooses between the four coins from the start.

Probabilistic choice: In the same vein, probabilistic choice can be interpreted with the monad $\mathcal{D}$ of finitely supported distributions.

Exceptions: As a generalization of termination, we can put a monad structure on the functor $(\cdot + E)$ where E is a set of exceptions that the computation can raise.

This view sheds light on one important features of monads we have not yet explored. If M and $\widehat{M}$ are monads describing computational effects, it is natural to ask for a way to combine them. Indeed, it does not seem too ambitious to have a model for programs which, for instance, make nondeterministic choices and also might terminate with no output. It turns out there is a very useful tool to deal with this at the level of monads.

Definition 266 (Monad distributive law)**.** Let (M, η, μ) and $(\widehat{M}, \widehat{\eta}, \widehat{\mu})$ be two monads on $\mathbf{C}$, a natural transformation $\lambda : M\widehat{M} \Rightarrow \widehat{M}M$ is called a **monad distributive law of M over $\widehat{M}$** if it makes (129), (130) commute.

$$\begin{tikzcd} M \arrow[r, Rightarrow, "M\widehat{\eta}"] \arrow[dr, Rightarrow, "\widehat{\eta}M"'] & M\widehat{M} \arrow[d, Rightarrow, "\lambda"] & \widehat{M} \arrow[l, Rightarrow, "\eta\widehat{M}"'] \arrow[dl, Rightarrow, "\widehat{M}\eta"] \\ & \widehat{M}M & \end{tikzcd} \tag{129}$$

$$\begin{tikzcd} MM\widehat{M} \arrow[rr, Rightarrow, "\mu\widehat{M}"] \arrow[d, Rightarrow, "M\lambda"'] & & M\widehat{M} \arrow[d, Rightarrow, "\lambda"] & & M\widehat{M}\widehat{M} \arrow[ll, Rightarrow, "M\widehat{\mu}"'] \arrow[d, Rightarrow, "\lambda\widehat{M}"] \\ M\widehat{M}M \arrow[r, Rightarrow, "\lambda M"'] & \widehat{M}MM \arrow[r, Rightarrow, "\widehat{M}\mu"'] & \widehat{M}M & \widehat{M}\widehat{M}M \arrow[l, Rightarrow, "\widehat{\mu}M"] & \widehat{M}M\widehat{M} \arrow[l, Rightarrow, "\widehat{M}\lambda"] \end{tikzcd} \tag{130}$$

Proposition 267. *If $\lambda : M\widehat{M} \Rightarrow \widehat{M}M$ is a monad distributive law, then the composite $\overline{M} = \widehat{M}M$ is a monad with unit $\overline{\eta} = \widehat{\eta} \diamond \eta$ and multiplication $\overline{\mu} = (\widehat{\mu} \diamond \mu) \cdot \widehat{M}\lambda M$.*

Proof. We have to show that the following instances of (115) and (116) commute.

$$\begin{tikzcd} \overline{M} \arrow[r, Rightarrow, "\overline{M}(\widehat{\eta}\diamond\eta)"] \arrow[ddr, Rightarrow, "1_{\overline{M}}"'] & \overline{M}^2 \arrow[d, Rightarrow, "\widehat{M}\lambda M"] & \overline{M} \arrow[l, Rightarrow, "(\widehat{\eta}\diamond\eta)\overline{M}"'] \arrow[ddl, Rightarrow, "1_{\overline{M}}"] \\ & \widehat{M}^2M^2 \arrow[d, Rightarrow, "\widehat{\mu}\diamond\mu"] & \\ & \overline{M} & \end{tikzcd} \tag{131}$$

$$\begin{tikzcd} \overline{M}^3 \arrow[r, Rightarrow, "\overline{M}\widehat{M}\lambda M"] \arrow[d, Rightarrow, "\widehat{M}\lambda M\overline{M}"'] & \overline{M}\widehat{M}^2M^2 \arrow[r, Rightarrow, "\overline{M}(\widehat{\mu}\diamond\mu)"] & \overline{M}^2 \arrow[d, Rightarrow, "\widehat{M}\lambda M"] \\ \widehat{M}^2M^2\overline{M} \arrow[d, Rightarrow, "(\widehat{\mu}\diamond\mu)\overline{M}"'] & & \widehat{M}^2M^2 \arrow[d, Rightarrow, "\widehat{\mu}\diamond\mu"] \\ \overline{M}^2 \arrow[r, Rightarrow, "\widehat{M}\lambda M"'] & \widehat{M}^2M^2 \arrow[r, Rightarrow, "\widehat{\mu}\diamond\mu"'] & \overline{M} \end{tikzcd} \tag{132}$$

For the left part of (131), we have the following diagram, the justifications of each part is given below with what diagram has to be considered and what functors should be applied to it (recall that acting on the diagrams does not affect commutativity). The notation (115).L (resp. .R) means only the left (resp. right) part of the diagram is considered.

$$
\begin{tikzcd}
\overline{M} \arrow[r, "\overline{M}\widehat{\eta}"] \arrow[rrr, Rightarrow, bend left, "\overline{M}(\widehat{\eta}\diamond\eta)"] \arrow[ddr, "\overline{M}\widehat{\eta}"'] \arrow[dddrrr, bend right, "\widehat{M}\widehat{\eta}M"] \arrow[ddddrrr, Rightarrow, bend right, "\mathbb{1}_{\overline{M}}=\mathbb{1}_{\widehat{M}}M"'] \arrow[dr, phantom, "\text{(a)}"] & \overline{M}\widehat{M} \arrow[rr, "\overline{M}\widehat{M}\eta"] \arrow[dr, "\overline{M}\eta\widehat{M}"] \arrow[dd, "\widehat{M}\mathbb{1}_M\widehat{M}"'] \arrow[d, phantom, "\text{(b)}"] & & \overline{M}^2 \arrow[dd, Rightarrow, "\widehat{M}\lambda M"] \arrow[dl, phantom, "\text{(c)}"] \\
& & \overline{M}M\widehat{M} \arrow[ur, "\overline{M}\lambda"'] \arrow[dl, "\widehat{M}\mu\widehat{M}"] \arrow[dr, phantom, "\text{(d)}"] & \\
& \overline{M}\widehat{M} \arrow[drr, "\widehat{M}\lambda"] \arrow[d, phantom, "\text{(e)}"] & & \widehat{M}^2M^2 \arrow[d, "\widehat{M}^2\mu"'] \arrow[dd, Rightarrow, bend left, "\widehat{\mu}\diamond\mu"] \\
& \text{(f)} & & \widehat{M}^2M \arrow[d, "\widehat{\mu}M"'] \\
& & & \overline{M}
\end{tikzcd}
\tag{133}
$$

(a) $\widehat{M}\mathbb{1}_M\widehat{M}$ is the identity transformation.

(b) Act on (115).L with $\widehat{M}$ on the left and right.

(c) Act on (129).R with $\overline{M}$ on the left.

(d) Act on (130).L with $\widehat{M}$ on the left.

(e) Act on (129).L with $\widehat{M}$ on the left.

(f) Act on (115) with M on the right.

Without a diagram, the derivation is this (we use ; to denote the opposite of $\circ$, i.e.: composition in the order read on the diagram):

$$
\begin{aligned}
\overline{M}(\widehat{\eta}\diamond\eta);\widehat{M}\lambda M;\widehat{\mu}\diamond\mu &= \overline{M}\widehat{\eta};\overline{M}\widehat{M}\eta;\widehat{M}\lambda M;\widehat{M}^2\mu;\widehat{\mu}M && \text{def of } \diamond \\
&= \overline{M}\widehat{\eta};\overline{M}\eta\widehat{M};\overline{M}\lambda;\widehat{M}\lambda M;\widehat{M}\widehat{M}\mu;\widehat{\mu}M && \overline{M}(129).\mathrm{R} \\
&= \overline{M}\widehat{\eta};\overline{M}\eta\widehat{M};\widehat{M}\mu\widehat{M};\widehat{M}\lambda;\widehat{\mu}M && \widehat{M}(130).\mathrm{L} \\
&= \overline{M}\widehat{\eta};\widehat{M}\mathbb{1}_M\widehat{M};\widehat{M}\lambda;\widehat{\mu}M && \widehat{M}(115).\mathrm{L}\widehat{M} \\
&= \overline{M}\widehat{\eta};\widehat{M}\lambda;\widehat{\mu}M && \\
&= \widehat{M}\widehat{\eta}M;\widehat{\mu}M && \widehat{M}(129).\mathrm{R} \\
&= \mathbb{1}_{\widehat{M}}M = \mathbb{1}_{\overline{M}} && (115).\mathrm{L}M
\end{aligned}
$$

For the right part of (131), the derivation is very similar.

For (132), we do the same thing.

$$
\begin{tikzcd}
\overline{M}^3 \arrow[rr, Rightarrow, "\overline{M}\widehat{M}\lambda M"] \arrow[d, Rightarrow, "\widehat{M}\lambda M\overline{M}"'] \arrow[drr, phantom, "(a)"] & & \overline{M}\widehat{M}^2M^2 \arrow[rr, "\overline{M}\widehat{M}^2\mu"] \arrow[d, "\widehat{M}\lambda\widehat{M}M^2"'] \arrow[rrrr, Rightarrow, bend left, "\overline{M}(\widehat{\mu}\diamond\mu)"] \arrow[drr, phantom, "(b)"] & & \overline{M}\widehat{M}^2M \arrow[rr, "\overline{M}\widehat{\mu}M"] \arrow[d, "\widehat{M}\lambda\widehat{M}M"] \arrow[ddrr, phantom, "(c)"] & & \overline{M}^2 \arrow[dd, Rightarrow, "\widehat{M}\lambda M"] \\
\widehat{M}^2M^2\overline{M} \arrow[rr, "\widehat{M}^2M\lambda M"] \arrow[dd, "\widehat{\mu}M^2\overline{M}"'] \arrow[dddd, Rightarrow, bend right, "(\widehat{\mu}\diamond\mu)\overline{M}"'] \arrow[dr, phantom, "(d)"] & & \widehat{M}^2M\widehat{M}M^2 \arrow[rr, "\widehat{M}^2M\widehat{M}\mu"] \arrow[dl, "\widehat{\mu}M\widehat{M}M^2"'] \arrow[dr, "\widehat{M}^2\lambda M^2"] \arrow[dd, phantom, "(e)"] \arrow[drr, phantom, "(f)"] & & \widehat{M}^2M\overline{M} \arrow[d, "\widehat{M}^2\lambda M"] & & \\
& \overline{M}\widehat{M}M^2 \arrow[dr, "\widehat{M}\lambda M^2"'] & & \widehat{M}^3M^3 \arrow[r, "\widehat{M}^3M\mu"] \arrow[dl, "\widehat{\mu}\widehat{M}M^3"'] \arrow[dd, "\widehat{M}^3\mu M"] \arrow[ddr, phantom, "(j)"] & \widehat{M}^3M^2 \arrow[rr, "\widehat{M}\widehat{\mu}M^2"] \arrow[dr, "\widehat{\mu}\widehat{M}M^2"] \arrow[dd, "\widehat{M}^3\mu"'] \arrow[drr, phantom, "(g)"] \arrow[ddr, phantom, "(k)"] & & \widehat{M}^2M^2 \arrow[d, "\widehat{\mu}M^2"] \arrow[ddd, Rightarrow, bend left, "\widehat{\mu}\diamond\mu"] \\
\widehat{M}M^2\overline{M} \arrow[ur, "\widehat{M}M\lambda M"'] \arrow[dd, "\widehat{M}\mu\overline{M}"'] \arrow[ddrr, phantom, "(h)"] & & \widehat{M}^2M^3 \arrow[dd, "\widehat{M}^2\mu M"'] \arrow[dr, phantom, "(i)"] & & & \widehat{M}^2M^2 \arrow[r, "\widehat{\mu}M^2"'] \arrow[dd, "\widehat{M}^2\mu"'] \arrow[ddr, phantom, "(m)"] & \widehat{M}M^2 \arrow[dd, "\widehat{M}\mu"] \\
& & & \widehat{M}^3M^2 \arrow[r, "\widehat{M}^3\mu"] \arrow[dl, "\widehat{\mu}\widehat{M}M^2"] \arrow[d, phantom, "(l)"] & \widehat{M}^3M \arrow[dr, "\widehat{\mu}\widehat{M}M"'] & & \\
\overline{M}^2 \arrow[rr, Rightarrow, "\widehat{M}\lambda M"'] & & \widehat{M}^2M^2 \arrow[rrr, "\widehat{M}^2\mu"'] \arrow[rrrr, Rightarrow, bend right, "\widehat{\mu}\diamond\mu"'] & & & \widehat{M}^2M \arrow[r, "\widehat{\mu}M"'] & \overline{M}
\end{tikzcd}
$$

(a) Def of $\widehat{M}\lambda \diamond \lambda M$.

(b) Def of $\widehat{M}\lambda\widehat{M} \diamond \mu$.

(c) Apply $\widehat{M}(\cdot)M$ to (130).R.

(d) Def of $\widehat{\mu} \diamond M\lambda M$.

(e) Def of $\widehat{\mu} \diamond \lambda M^2$.

(f) Def of $\widehat{M}^2\lambda \diamond \mu$.

(g) Apply $(\cdot)M^2$ to associativity of $\widehat{\mu}$ (116).

(h) Apply $\widehat{M}(\cdot)M$ to (130).L.

(i) Def of $\widehat{\mu}\widehat{M} \diamond \mu M$.

(j) Apply $\widehat{M}^3$ to associativity of μ (116).

(k) Def of $\widehat{\mu}\widehat{M} \diamond \mu$.

(l) Same as (k): Def of $\widehat{\mu}\widehat{M} \diamond \mu$.

(m) Def of $\widehat{\mu} \diamond \mu$.

□

Corollary 268. *If* **C** *has (binary) coproducts and a terminal object* **1** *and* M *is a monad, then* $M(\cdot + \mathbf{1})$ *is also monad.*

Proof. We will exhibit a monad distributive law of M over $(\cdot + \mathbf{1})$. We claim

$$\iota_X : MX + \mathbf{1} \to M(X+1) = [M(\mathsf{inl}^{X+\mathbf{1}}), \eta_{X+\mathbf{1}} \circ \mathsf{inr}^{X+\mathbf{1}}]$$

is a monad distributive law $\iota : (\cdot + \mathbf{1})M \Rightarrow M(\cdot + \mathbf{1})$. Then, it follows by Proposition 267. □

Example 269 (Rings)**.** Consider the term monads for the theory of monoids and abelian groups $T_{\mathbf{Mon}}$ and $T_{\mathbf{Ab}}$. You can check that they are the monads induced by the free-forgetful adjunctions between **Mon** and **Set** and **Ab** and **Set**. Also, $T_{\mathbf{Mon}}$ is the same thing as the list monad. Call the binary operation of $T_{\mathbf{Mon}}$ and $T_{\mathbf{Ab}}$ the product and sum respectively.

Then, by identifying products of sums (elements of $T_{\mathbf{Mon}}T_{\mathbf{Ab}}X$) with sums of products (elements of $T_{\mathbf{Ab}}T_{\mathbf{Mon}}X$) by *distributing* the product over the sum as we are used to do with, say, real numbers, we obtain a monad distributive law of $T_{\mathbf{Mon}}$ over $T_{\mathbf{Ab}}$. The resulting composite monad $T_{\mathbf{Ab}}T_{\mathbf{Mon}}$ is the term monad for the theory of rings. The term distributive law comes from this example.

Remark 270. It is not always possible to combine monads in such a natural way. For instance, it was shown that no distributive law exist between $\mathcal{P}_{\mathrm{ne}}$ and $\mathcal{D}$ and even that no monad structure can exist on $\mathcal{P}_{\mathrm{ne}}\mathcal{D}$ or $\mathcal{D}\mathcal{P}_{\mathrm{ne}}$. Thus, modelling combined probabilistic and nondeterministic effects has been quite a hard endeavor and is still an active area of research I discovered in an internship with Matteo Mio and Valeria Vignudelli at ENS de Lyon last summer.

If you are looking for more applications of this perspective on monads and especially if you enjoyed the assignment on Brzozowski's algorithm, I suggest you look into the paper *Generalizing Determinization From Automata to Coalgebras* available at `https://arxiv.org/abs/1302.1046`.

Exercises

1. Show that the triple $(\mathcal{D}, \eta, \mu)$ described in Example 248.3 is a monad.
2. Show that the Kleisli category of the powerset monad is the category **Rel** of relations.
3. Show that ι defined in the proof of Corollary 268 is a monad distributive law.
4. Show Proposition 263 with the monad structure on $M(\cdot + \mathbf{1})$ given in Corollary 268.

Bonus: (Co)algebras and (Co)induction

(Co)algebras and (Co)induction

In this section, we will introduce (co)algebras and (co)induction and ask you to fill out some gaps. While the definitions are given for an arbitrary category $\mathbf{C}$, all of the examples and exercises will be done on **Set**.

Algebras

The term *algebra* has a few different meanings and here we will more precisely consider F–algebras for some endofuntor $F : \mathbf{C} \rightsquigarrow \mathbf{C}$. Nevertheless, all the objects that are referred to as algebras have a common motto: *algebras only care about structure*.

For instance, in a first year algebra course, groups are studied up to isomorphisms (maps that preserve the structure) because all the useful properties of a group are determined completely by how the operation acts on the underlying set. As a concrete example, the groups $\mathbb{Z}_2 \times \mathbb{Z}_3$ and $\mathbb{Z}_6$ are the same group even if their elements have different names. It is a similar situation for rings, vector spaces and a lot more mathematical objects.

Before giving the general definition of an F–algebra, we categorify the definition of a group.

Example 271. Usually, a group is defined as a set G along with an operation $\cdot : G \times G \to G$ satisfying some conditions, namely, associativity, existence of an identity and existence of an inverse for each element. It is then a formal consequence that the identity and inverses are unique.

Therefore, it is equivalent to define a group as a set G with a binary operation $\cdot$, an identity $1 \in G$ and an inverse g^{-1} for all $g \in G$ that satisfy some properties. In order to abide to the categorical mindset, it is better to view the identity as a morphism $1 : \mathbf{1} \to G$ ($\mathbf{1}$ is the final object, i.e.: a singleton) and describe inverses with a morphism $(-)^{-1} : G \to G$. A few additional diagrams have to commute for G to satisfy all axioms of a group, but we leave their construction as an exercise. We conclude that a group can be seen as a morphism

$$[1, (-)^{-1}, \cdot] : \mathbf{1} + G + (G \times G) \to G.$$

This is our first example of an F–algebra, here $F : \mathbf{Set} \rightsquigarrow \mathbf{Set}$ sends a set G to $\mathbf{1} + G + (G \times G)$ and a morphism f to $[\mathrm{id}_{\mathbf{1}}, f, (f, f)]$.

Note that since we have not used the fact that G is a set, this definition gives rise to groups in other categories than **Set** provided they have a final object, products and coproducts.

Exercise 272 (1pt)**.** Draw the additional diagrams that G, $\cdot$, 1 and $(-)^{-1}$ should satisfy to obtain a group.

Definition 273 (F–algebra)**.** Let $F : \mathbf{C} \rightsquigarrow \mathbf{C}$ be a functor, an F**–algebra** is an object $A \in \mathbf{C}_0$ along with a morphism $\alpha : F(A) \to A \in \mathbf{C}_1$ called the **structure map**.

Examples 274.

1. Since a monoid M only has a binary operation and an identity, it can be represented as an algebra $[1, \cdot] : \mathbf{1} + (M \times M) \to M$. Similarly, one can construct algebras that represent rings and vector spaces, but not fields (why?).

2. We will see later that the induction principle we know comes from the algebra $[0, \mathsf{succ}] : \mathbf{1} + \mathbb{N} \to \mathbb{N}$, where $0(*) = 0$ and $\mathsf{succ}(n) = n + 1$.

3. Although we will not use them often, there are algebras in different categories than **Set**. In computer science, we often use induction to reason about lists, we will see that this is because lists are algebras. More precisely, for a type A, the type A^* of lists with elements of type A is has an algebra structure $[\mathsf{nil}, \mathsf{cons}] : 1 + (A \times A^*) \to A^*$ given by $\mathsf{nil}(*) = \varepsilon \in A^*$ (the empty list) and $\mathsf{cons}(a, w) = w \cdot a \in A^*$ (concatenation). The category in which this algebra lives depends on the programming language and types considered.

Remark 275*.* The components of the structure map (i.e.: 0 and succ in the second example) are often called the **constructors** because they define rules to construct elements of the algebra using other elements as building blocks.

As you might expect F–algebras form a category, denoted $\mathrm{Alg}(F)$, with the following notion of morphism.

Definition 276 (F–algebra homomorphism)**.** Let $F : \mathbf{C} \rightsquigarrow \mathbf{C}$ be a functor and $\alpha : F(A) \to A$ and $\beta : F(B) \to B$ be F–algebras. An F**–algebra homomorphism** from the former to the latter is a morphism $f : A \to B$ that makes this square commute.

$$\begin{tikzcd} F(A) \arrow[r, "F(f)"] \arrow[d, "\alpha"'] & F(B) \arrow[d, "\beta"] \\ A \arrow[r, "f"'] & B \end{tikzcd} \tag{134}$$

This definition also clarifies why we require F to be a functor.

Example 277. Let $F = X \mapsto \mathbf{1} + X + (X \times X)$ be the functor discussed Example 271. An F–algebra homomorphism is represented by the following square.

$$\begin{tikzcd} \mathbf{1} + G + (G \times G) \arrow[r, "{[\mathrm{id}_{\mathbf{1}}, f, (f,f)]}"] \arrow[d, "{[1_G, (-)^{-1}, \cdot]}"'] & \mathbf{1} + H + (H \times H) \arrow[d, "{[1_H, (-)^{-1}, \cdot]}"] \\ G \arrow[r, "f"'] & H \end{tikzcd} \tag{135}$$

Unwrapped, this says that $f(1_G) = 1_H$, $f(g^{-1}) = f(g)^{-1}$ and $f(g \cdot g') = f(g) \cdot f(g')$ for all $g, g' \in G$, i.e.: if both algebras represent groups as seen in Example 271, it is a group homomorphism.

Coalgebras

Now that we have a categorical notion of algebra, we can look at its dual.

Definition 278 (F–coalgebra)**.** Let $F : \mathbf{C} \rightsquigarrow \mathbf{C}$ be a functor, an F**–coalgebra** is an object $A \in \mathbf{C}_0$ called the **carrier** along with a morphism $\omega : A \to F(A)$ called the **behavior map**. We will refer to a coalgebra with A, ω or the pair (A, ω).

Examples 279. 1. If F is the identity on **Set**, then an F–coalgebra is just an endomorphism $\omega : A \to A$ and it is sometimes called a **dynamical system**. You can think of the elements of A as states and ω as the transition map for the system.

2. Let $\mathsf{Str}_{\mathbb{N}} : \mathbf{Set} \rightsquigarrow \mathbf{Set} = \mathbb{N} \times (-)$ be the functor sending a set X to $\mathbb{N} \times X$ and a function $f : X \to Y$ to $\mathrm{id}_{\mathbb{N}} \times f : \mathbb{N} \times X \to \mathbb{N} \times Y$. An example of a $\mathsf{Str}_{\mathbb{N}}$–coalgebra is the set $\mathbb{N}^{\mathbb{N}}$ of all infinite sequences (also called **streams**) of natural numbers with the structure map $(\mathsf{head}, \mathsf{tail}) : \mathbb{N}^{\mathbb{N}} \to \mathbb{N} \times \mathbb{N}^{\mathbb{N}}$ given by

$$\mathsf{head} : \mathbb{N}^{\mathbb{N}} \to \mathbb{N} = \sigma \mapsto \sigma(0) \quad \text{and} \quad \mathsf{tail} : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}} = \sigma \mapsto \sigma \circ \mathsf{succ}.$$

Unsurprisingly, we call $\mathsf{head}(\sigma)$ and $\mathsf{tail}(\sigma)$ the **head** and **tail** of the stream σ respectively.

Exercise 280 (1pt)**.** Denoting $2 = \{0, 1\}$, let $F = 2 \times (-)^A$ send a set X to $2 \times X^A$ and a function $f : X \to Y$ to $\mathrm{id}_2 \times (f \circ -) : 2 \times X^A \to 2 \times Y^A$. When A is finite, show that there is a correspondence between F–coalgebras with a finite carrier and DFAs over the alphabet A without initial states.

Remark 281*.* The components of the behavior map (i.e.: h and t in the second example) are often called **destructors** or **observers** because they decompose elements of the coalgebra.

We define morphisms of F–coalgebras in order to obtain a category $\mathrm{Coalg}(F)$.

Definition 282 (F–coalgebra homomorphism)**.** Let $F : \mathbf{C} \rightsquigarrow \mathbf{C}$ be a functor and $\alpha : A \to F(A)$ and $\beta : B \to F(B)$ be F–coalgebras. An F**–coalgebra homomorphism** from the former to the latter is a morphism $f : A \to B \in \mathbf{C}_1$ that makes (136) commute.

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "\alpha"'] & B \arrow[d, "\beta"] \\ F(A) \arrow[r, "F(f)"'] & F(B) \end{tikzcd} \tag{136}$$

Induction

Induction is a very well known and prevalent proof principle. In it most common form, it says that for any predicate P on $\mathbb{N}$, if $P(0)$ is true and $P(n) \implies P(n+1)$

is true for any $n \in \mathbb{N}$, then so is $P(n)$ for any $n \in \mathbb{N}$. In this section, we use the power of algebras to generalize this proof principle and give a few examples.

Definition 283 (Initial algebra)**.** Let $F : \mathbf{C} \rightsquigarrow \mathbf{C}$ be a functor, an **initial algebra** is an initial object in the category of F–algebras. Namely, it is an algbera (A, α) such that for any other algebra (B, β), there is a unique $f : A \to B$ making the following square commute.

$$\begin{tikzcd} F(A) \arrow[r, dashed, "F(f)"] \arrow[d, "\alpha"'] & F(B) \arrow[d, "\beta"] \\ A \arrow[r, dashed, "f"'] & B \end{tikzcd} \tag{137}$$

Example 284. The algebra $(\mathbb{N}, [0, \mathsf{succ}])$ is initial for the functor $\mathbf{1} + (-)$. Indeed, let $[z, s] : \mathbf{1} + X \to X$ be another algebra for this functor, then a map $f : \mathbb{N} \to X$ that makes the following diagram commute must necessarily satisfy $f(0) = z(*)$ and $f(n) = s^n(z(*))$.

$$\begin{tikzcd} \mathbf{1} + \mathbb{N} \arrow[r, "{[\mathrm{id}_{\mathbf{1}}, f]}"] \arrow[d, "{[0,\mathsf{succ}]}"'] & \mathbf{1} + X \arrow[d, "{[z,s]}"] \\ \mathbb{N} \arrow[r, "f"'] & X \end{tikzcd} \tag{138}$$

This completely determines f and moreover, defining f like this for any $(\mathbf{1} + (-))$-algebra $(X, [z, s])$ yields an algebra homomorphism.

Exercise 285 (2pts)**.** Show that the algebra for lists $[\mathsf{nil}, \mathsf{cons}] : 1 + A \times A^* \to A^*$ is initial for the functor $1 + A \times (-) : \mathbf{Set} \rightsquigarrow \mathbf{Set}$ (you know its action on sets, on morphisms it sends $f : X \to Y$ to $[\mathrm{id}_{\mathbf{1}}, (\mathrm{id}_A, f)]$).

We already know that initial objects are unique up to unique isomorphisms, but Lambek also showed furthermore that initial F–algebras are fixed points of F.

Proposition 286 (Lambek)**.** *Let $F : \mathbf{C} \rightsquigarrow \mathbf{C}$, if (A, α) is an initial F–algebra, then $\alpha : F(A) \to A$ is an isomorphism.*

Exercise 287 (1.5pts)**.** Prove Proposition 286. **Hint:** Consider the algebra $F(\alpha) : F^2(A) \to F(A)$.

Initial algebras generalize the inductive reasoning we use with the natural numbers to much more settings. We distinguish two cases where induction is used: inductive definitions and the induction proof principle.

For the former, the general idea is that, given an initial F–algebra (A, α), we can easily define a function $f : A \to B$ by looking at how it acts on constructors. Indeed, with only this data, we can construct an F–algebra structure on B such that the unique homomorphism $! : A \to B$ acts exactly like f.

Example 288 (Inductive definition)**.** Recall that $(A^*, [\mathsf{nil}, \mathsf{cons}])$ is the initial $(1 + A \times (-))$-algebra. We would like to define the function $\mathsf{len} : A^* \to \mathbb{N}$ that computes the length of a list. Intuitively, it satisfies the equations

$$\mathsf{len}(\mathsf{nil}) = 0 \qquad \mathsf{len}(\mathsf{cons}(a, l)) = 1 + \mathsf{len}(l).$$

Then, if we construct the $(1 + A \times (-))$-algebra $[z, s] : 1 + A \times \mathbb{N} \to \mathbb{N}$ defined by $z(*) = 0$ and $s(a, n) = 1 + n$, we can verify that the unique algebra homomorphism $! : A^* \to \mathbb{N}$ is the function len because both make the following diagram commute.

$$\begin{tikzcd} 1 + A \times A^* \arrow[r, "1+\mathrm{id}_A \times !"] \arrow[d, "{[\mathsf{nil},\mathsf{cons}]}"'] & 1 + A \times \mathbb{N} \arrow[d, "{[z,s]}"] \\ A^* \arrow[r, "!=\mathsf{len}"'] & \mathbb{N} \end{tikzcd} \tag{139}$$

Exercise 289 (1pt)**.** Use the initiality of $\mathbb{N}$ for the functor $1 + (-)$ to define the function $n \mapsto 2^n$.

Generalizing proofs by induction in this context is more involved and we will need the definition of F–congruences. While F–algebra homomorphisms are maps between algebras that preserve the structure, an F–congruence is a relation between two algebras that preserves the structure.

Definition 290. Let $F : \mathbf{C} \rightsquigarrow \mathbf{C}$ be a functor and (A, α) and (B, β) be F–algebras, a relation $R \subseteq A \times B$ is an F**-congruence** if there is a structure map $\gamma : F(R) \to R$ such that the projections $\pi_1 : R \to A$ and $\pi_2 : R \to B$ are algebra homomorphisms making this diagram commute.

$$\begin{tikzcd} F(A) \arrow[d, "\alpha"'] & F(R) \arrow[l, "F\pi_1"'] \arrow[r, "F\pi_2"] \arrow[d, "\gamma"] & F(B) \arrow[d, "\beta"] \\ A & R \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & B \end{tikzcd} \tag{140}$$

Example 291. If F is the identity functor, then for any algebras (A, α) and (B, β) and any relation $R \subseteq A \times B$, $\gamma = (\alpha \circ \pi_1, \beta \circ \pi_2)$ is a structure map making R into an F–congruence.

Exercise 292 (2pts)**.** Let $F = \mathbf{1} + (-)$, we have already seen that $\mathbb{N}$ is an initial F–algebra.

(a) Give a necessary and sufficient condition for $R \subseteq \mathbb{N} \times \mathbb{N}$ to be an F–congruence.

(b) Conclude that for any F–congruence $R \subseteq \mathbb{N} \times \mathbb{N}$, $\forall n \in \mathbb{N}, (n, n) \in \mathbb{R}$.

The next theorem generalizes the previous exercise.

Theorem 293 (General induction)**.** *Let $F : \mathbf{C} \rightsquigarrow \mathbf{C}$ be a functor and (A, α) be an initial F–algebra, if $R \subseteq A \times A$ is an F–congruence, then it is reflexive, that is $(a, a) \in R$ for all $a \in A$.*

Exercise 294 (1.5pts)**.** Prove Theorem 293.

Example 295 (Induction in $\mathbb{N}$)**.** We will see how the induction principle in Theorem 293 implies the usual induction principle. Let P be a predicate on $\mathbb{N}$ that satisfies $0 \in P$ and $n \in P \implies n + 1 \in P$. One can show that $P \times P \subseteq \mathbb{N} \times \mathbb{N}$ is an F–congruence and by general induction, $(n, n) \in P \times P$ for all $n \in \mathbb{N}$, i.e.: $\forall n \in \mathbb{N}, n \in P$.

Although going through all these abstractions and definitions seems like a really convoluted way to prove the induction principle, it lead us to two new concepts. First, we can now use inductive reasoning on all sorts of algebras even if they are in no way similar to $\mathbb{N}$. Second, we obtained an easy access to the dual of induction which we present in the following section.

Coinduction

Definition 296 (Final coalgebra)**.** Let $F : \mathbf{C} \rightsquigarrow \mathbf{C}$ be a functor, a **final coalgebra** is a final object in the category of F–coalgebras. Namely, it is a coalgbera (A, ω) such that for any other coalgebra (B, ψ), there is a unique morphism $f : B \to A$ making the following square commute.

$$\begin{tikzcd} B \arrow[r, dashed, "f"] \arrow[d, "\psi"'] & A \arrow[d, "\omega"] \\ F(B) \arrow[r, dashed, "F(f)"'] & F(A) \end{tikzcd} \tag{141}$$

Since final coalgebras are unique up to unique homomorphism, we will refer to **the** final coalgebra.

Example 297. The $\mathsf{Str}_{\mathbb{N}}$–coalgebra $(\mathsf{head}, \mathsf{tail}) : \mathbb{N}^{\mathbb{N}} \to \mathbb{N} \times \mathbb{N}^{\mathbb{N}}$ is final. That is, for any $\mathsf{Str}_{\mathbb{N}}$–coalgebra, $(h, t) : X \to \mathbb{N} \times X$, there is unique morphism $! : X \to \mathbb{N}^{\mathbb{N}}$ making (142) commute.

$$\begin{tikzcd} X \arrow[r, dotted, "!"] \arrow[d, "{(h,t)}"'] & \mathbb{N}^{\mathbb{N}} \arrow[d, "{(\mathsf{head},\mathsf{tail})}"] \\ \mathbb{N} \times X \arrow[r, dotted, "\mathrm{id}_N \times !"'] & \mathbb{N} \times \mathbb{N}^{\mathbb{N}} \end{tikzcd} \tag{142}$$

The equation corresponding to (142) is $(\mathsf{head}, \mathsf{tail}) \circ ! = (\mathrm{id}_{\mathbb{N}} \times !) \circ (h, t)$. We can decompose it into $\mathsf{head} \circ ! = h$ and $\mathsf{tail} \circ ! = ! \circ t$. The first equation tells us that $!(x)$ starts with the number $h(x)$ and the second equation tells us that the tail of $!(x)$ is the stream corresponding to $t(x)$ (via !). In short, we have $!(x) = h(x) \cdot !(t(x))$. If we further decompose the tail, we obtain

$$!(x) = h(x) \cdot h(t(x)) \cdot h(t(t(x))) \cdots h(t^n(x)) \cdots .$$

This should convince you that the only suitable choice for ! is $x \mapsto (n \mapsto h(t^n(x)))$.

Exercise 298 (3pts)**.** Let $F = 2 \times (-)^A$ with A finite and consider the F–coalgebra

$$(\varepsilon?, \omega) : 2^{A^*} \to 2 \times (2^{A^*})^A,$$

where for a language $L \subseteq A^*$ (ε denotes the empty string),

$$\varepsilon?(L) = \begin{cases} 1 & \varepsilon \in L \\ 0 & \text{o/w} \end{cases}, \qquad \omega(L) = a \mapsto L_a = \{w \in A^* \mid a \cdot w \in L\}.$$

The language $L_a = \omega(L)(a)$ is sometimes called the left a–derivative of L. Given a DFA M corresponding to the coalgebra $[f,\delta] : Q \to 2 \times Q^A$, show that the function

$$o : Q \to 2^{A^*} = q \mapsto \{w \in A^* \mid M \text{ accepts } w \text{ when starting in state } q\}$$

is the only map making (143) commute.

$$\begin{tikzcd} & 2 \\ X \arrow[ur, "f"] \arrow[r, dashed, "o"] \arrow[d, "\delta"'] & 2^{A^*} \arrow[u, "\varepsilon?"'] \arrow[d, "\omega"] \\ X^A \arrow[r, dashed, "o^A"'] & (2^{A^*})^A \end{tikzcd} \tag{143}$$

Remark 299. Extending your proof to X not necessarily finite, you could obtain the fact that $(2^{A^*}, (\varepsilon?, \omega))$ is the final $2 \times (-)^A$ coalgebra.

Proposition 300. *If $\omega : A \to F(A)$ is a final F–coalgebra, then ω is an isomorphism.*

Exercise 301 (0.5pts)**.** Prove Proposition 300.

Final coalgebras lets us use **coinductive definitions**. You will see that they are quite similar to inductive definitions you are used to. In fact, as the name suggests, they are dual to each other, but we will not make this formal here.

Examples 302 (Coinductive definitions)**.** Fix some set A and consider the functor $\mathsf{Str}_A = A \times (-)$, similarly to $\mathsf{Str}_{\mathbb{N}}$, the set $A^{\mathbb{N}}$ of streams in A is the final Str_A–coalgebra with the behavior map $(\mathsf{head}, \mathsf{tail})$ as defined in Example 279. We will define three different maps using the finality of $A^{\mathbb{N}}$.

1. The function $\mathsf{even} : A^{\mathbb{N}} \to A^{\mathbb{N}}$ takes a stream $\sigma = (\sigma(0), \sigma(1), \dots)$ and maps it to the stream of elements of σ at even positions, namely $\mathsf{even}(\sigma) = (\sigma(0), \sigma(2), \dots)$. To define it coinductively, we need to describe how destructors act on it. It is easy to verify that

$$\mathsf{head}(\mathsf{even}(\sigma)) = \mathsf{head}(\sigma) \quad \text{and} \quad \mathsf{tail}(\mathsf{even}(\sigma)) = \mathsf{even}(\mathsf{tail}(\mathsf{tail}(\sigma))).$$

   Hence, if we define a new Str_A–coalgebra on $A^{\mathbb{N}}$ by $(h,t) = (\mathsf{head}, \mathsf{tail}^2 = \mathsf{tail} \circ \mathsf{tail})$, then we conclude by finality and commutativity of the following diagram that $! : A^{\mathbb{N}} \to A^{\mathbb{N}}$ is the function even.

$$\begin{tikzcd} A^{\mathbb{N}} \arrow[r, dashed, "!"] \arrow[d, "(\mathsf{head},\mathsf{tail}^2)"'] & A^{\mathbb{N}} \arrow[d, "(\mathsf{head},\mathsf{tail})"] \\ A \times A^{\mathbb{N}} \arrow[r, dashed, "\mathrm{id}_A \times !"'] & A \times A^{\mathbb{N}} \end{tikzcd} \tag{144}$$

2. The operation of merging two streams is described by the function $\mathsf{merge} : A^{\mathbb{N}} \times A^{\mathbb{N}} \to A^{\mathbb{N}}$ mapping (σ, τ) to $(\sigma(0), \tau(0), \sigma(1), \tau(1), \dots)$. Observe that destructors act as follows:

$$\mathsf{head}(\mathsf{merge}(\sigma,\tau)) = \mathsf{head}(\sigma) \quad \text{and} \quad \mathsf{tail}(\mathsf{merge}(\sigma,\tau)) = \mathsf{merge}(\tau, \mathsf{tail}(\sigma)).$$

The existence of merge is then proven with finality of $A^{\mathbb{N}}$ and the following coalgebra behavior map (where π_1 and π_2 are the projections):

$$(\mathsf{head} \circ \pi_1, \pi_2, \mathsf{tail} \circ \pi_1) : A^{\mathbb{N}} \times A^{\mathbb{N}} \to A \times A^{\mathbb{N}} \times A^{\mathbb{N}}.$$

Exercise 303 (1pts)**.** Similarly to the first item, show that the function $\mathsf{odd} : A^{\mathbb{N}} \to A^{\mathbb{N}}$ mapping $\sigma = (\sigma(0), \sigma(1), \dots)$ to $\mathsf{odd}(\sigma) = (\sigma(1), \sigma(3), \dots)$ can be defined coinductively.

There is also a dual to the induction proof principle for which we need to define bisimulation.

Definition 304 (F–bisimulation)**.** Let $F : \mathbf{C} \rightsquigarrow \mathbf{C}$ be a functor and (A, ω) and (B, ψ) be F–coalgebras, a relation $R \subseteq A \times B$ is an F**–bisimulation** if there is a behavior map $\gamma : R \to F(R)$ such that the projections $\pi_1 : R \to A$ and $\pi_2 : R \to B$ are coalgebra homomorphisms making this diagram commute.

$$\begin{tikzcd} A \arrow[d, "\omega"'] & R \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[d, "\gamma"] & B \arrow[d, "\psi"] \\ F(A) & F(R) \arrow[l, "F\pi_1"] \arrow[r, "F\pi_2"'] & F(B) \end{tikzcd} \tag{145}$$

Theorem 305 (Coinductive proof principle)**.** *Let $F : \mathbf{C} \rightsquigarrow \mathbf{C}$ be a functor and (A, ω) be the final F–coalgebra, if $R \subseteq A \times A$ is an F–bisimulation, then it is contained in the diagonal relation, that is $(a, a') \in R$ implies $a = a'$.*

Exercise 306 (0.5pts)**.** Prove Theorem 305.

Example 307. We will use coinduction to prove that $\mathsf{odd}(\mathsf{merge}(\sigma, \tau)) = \tau$. By the previous theorem, it is enough to show that $\mathcal{R} = \{(\mathsf{odd}(\mathsf{merge}(\sigma, \tau)), \tau) \mid \sigma, \tau \in A^{\mathbb{N}}\}$ is an F–bisimulation. We claim that $\gamma = (x, y) \mapsto (\mathsf{head}(x), (\mathsf{tail}(x), \mathsf{tail}(y)))$ makes the following diagram commute.

$$\begin{tikzcd} A^{\mathbb{N}} \arrow[d, "(\mathsf{head},\mathsf{tail})"'] & \mathcal{R} \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[d, "\gamma"] & A^{\mathbb{N}} \arrow[d, "(\mathsf{head},\mathsf{tail})"] \\ A \times A^{\mathbb{N}} & A \times \mathcal{R} \arrow[l, "\mathrm{id}_A \times \pi_1"] \arrow[r, "\mathrm{id}_A \times \pi_2"'] & A \times A^{\mathbb{N}} \end{tikzcd} \tag{146}$$

To prove our claim, first note that

$$\begin{aligned} \mathsf{head}(\mathsf{tail}(\mathsf{merge}(\sigma, \tau))) &= \mathsf{head}(\mathsf{merge}(\tau, \mathsf{tail}(\sigma))) \\ &= \mathsf{head}(\tau), \end{aligned}$$

so if we can show that $(\mathsf{tail}(x), \mathsf{tail}(y)) \in \mathcal{R}$ for any $(x, y) \in \mathcal{R}$, then we would conclude that the diagram commutes. This last part follows from the derivation

$$\begin{aligned} \mathsf{tail}(\mathsf{odd}(\mathsf{merge}(\sigma, \tau))) &= \mathsf{odd}(\mathsf{tail}(\mathsf{tail}(\mathsf{merge}(\sigma, \tau)))) \\ &= \mathsf{odd}(\mathsf{tail}(\mathsf{merge}(\tau, \mathsf{tail}(\sigma)))) \end{aligned}$$

$$= \mathsf{odd}(\mathsf{merge}(\mathsf{tail}(\sigma), \mathsf{tail}(\tau))).$$

Indeed, we obtain

$$(\mathsf{tail}(\mathsf{odd}(\mathsf{merge}(\sigma, \tau), \mathsf{tail}(\tau)) = (\mathsf{odd}(\mathsf{merge}(\mathsf{tail}(\sigma), \mathsf{tail}(\tau))), \mathsf{tail}(\tau)) \in \mathcal{R}.$$

Exercise 308 (3pts)**.** (a) Given $f : A \to A$ coinductively define $\mathsf{map}_f : A^{\mathbb{N}} \to A^{\mathbb{N}}$ such that $\mathsf{map}_f(\sigma) = (f(\sigma(0)), f(\sigma(1)), \dots)$.

(b) Show by coinduction that for any $\sigma \in A^{\mathbb{N}}$, $\mathsf{even}(\mathsf{map}_f(\sigma)) = \mathsf{map}_f(\mathsf{even}(\sigma))$.

Preliminaries on Automata

There are no questions in this section, only stuff we believe you learned in a first course in theoretical CS. Give it a quick read in order to at least to identify my notation, the most important part is the definition of Brzozowski's algorithm in Section .

Deterministic Finite Automata

Definition 309 (DFA)**.** A **deterministic finite automaton** (DFA) M is composed of a **finite** alphabet Σ (a set of symbols), a finite set of **states** Q with a starting state identified by q_0, a **transition function** $\delta : Q \times \Sigma \to Q$ and a subset $F \subseteq Q$ of **accepting** states.

An input for M is a finite word (concatenation of finitely many elements) of Σ, we will denote it $w \in \Sigma^*$. The automaton reads its input letter by letter, starting in state q_0, and changes its state according to δ: $\delta(q, a) = q'$ means that if M is in state q and reads the symbol a, then M ends up in state q'. After reading all of its input M is in some state q and outputs "Accept" if $q \in F$ and "Reject" otherwise.

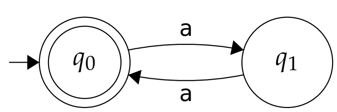
The global behavior of M is described by the subset $L \subseteq \Sigma^*$ of words that M accepts. We denote this subset $L(M)$ and say that $L(M)$ is the **language recognized** by M.

We will denote δ^* the extension of δ to Σ^*, it is defined inductively by

$$\delta^*(x, \varepsilon) = x \quad \text{and} \quad \delta^*(x, a \cdot w) = \delta^*(\delta(x, a), w).$$

Examples 310. Typically, it is more readable to describe DFAs by drawing a graphical representation than by defining each component.

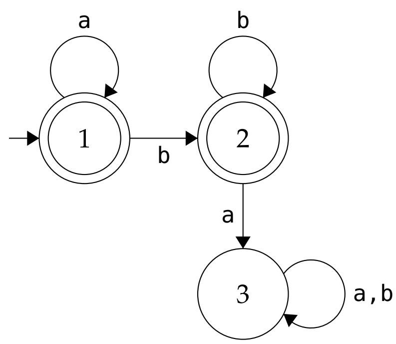
1. Consider the DFA described by $\Sigma = \{\mathsf{a}\}$, $Q = \{q_0, q_1\}$, $\delta = (q_0, \mathsf{a}) \mapsto q_1, (q_1, \mathsf{a}) \mapsto q_0$, $F = \{q_0\}$. It is represented by the following diagram.



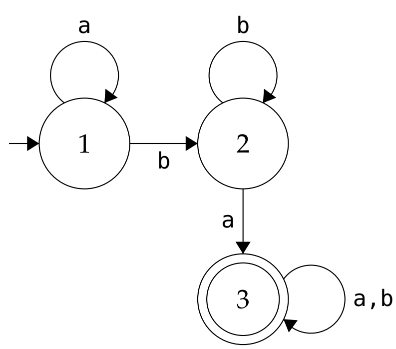
The circles represent the states and accepting states are denoted with a second inner circle. The arrows represent transitions and the labels are the symbols that need to be read for this transition to occur. The smaller arrow with no label designates the starting state.

It is easy to see that the language recognized by this DFA consists of all the words with an even number of $\mathtt{a}$'s (i.e.: $L(M) = \{\mathtt{a}^{2n} : n \in \mathbb{N}\}$).

2. The DFA recognizing the language $\{\mathtt{a}^n\mathtt{b}^m \mid n, m \in \mathbb{N}\}$ can be described as follows.



3. For any DFA M on an alphabet Σ, it is very easy to describe a DFA M' that recognizes the complement of $L(M)$, i.e.: such that $L(M') = \Sigma^* \setminus L(M)$. We just invert the roles of accepting and non-accepting states. Here is the DFA recognizing the complement of $\{\mathtt{a}^n\mathtt{b}^m \mid n, m \in \mathbb{N}\}$.



Remark 311. The term **deterministic** means that the transitions are completely determined by the input, that is, you can run the DFA on the same input many many times and it will always end in the same state and output the same thing. Mathematically, determinism comes from the fact that $\delta(q, a)$ takes the value of only one state for any $q \in Q$ and $a \in \Sigma$. If we allow δ to be nondeterministic, we get the definition of an NFA.

Nondeterminisitic Finite Automata

Definition 312. An **nondeterministic automaton** (NFA) M consists of a finite alphabet Σ, a finite set of states Q with a starting state identified by q_0, a transition function $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$ and a subset $F \subseteq Q$ of accepting states.

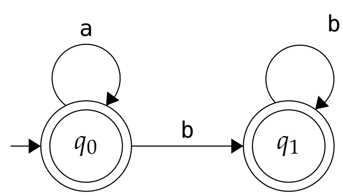
There are two main differences with a DFA. First, an NFA can sometimes make a transition without reading a symbol from the input but rather by reading ε (denoting an empty string). Second, the image of δ is a set of possible states for the transition to end in. Let us see how this affects the behavior of M.

The automaton still reads its input letter by letter starting in state q_0, but now instead of making the only possible transition, it makes all of them at the same time and continues the computation on multiple branches. The output of M is "Accept" if in at least one branch, all the input was read and M is in an accepting state, otherwise M outputs "Reject". The language recognized by M is defined as for DFAs.

Another way to view nondeterminism is to consider that M has access to an all-knowing oracle that will choose which transition to make (out of the possible ones). This oracle always make the choices that lead to accepting the input if possible, hence, M accepts $w \in \Sigma^*$ if and only if there is a sequence of choices that lead to an accepting state after reading all the input.

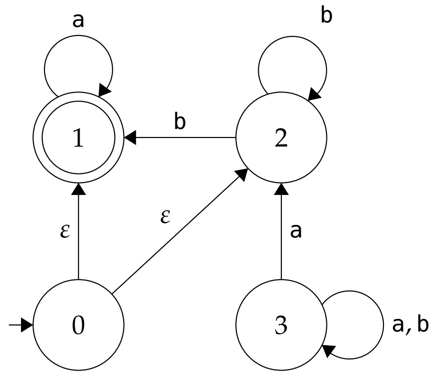
Examples 313. The representation of NFAs is really similar to that of DFAs but now, arrows can be labeled with an ε, multiple arrows coming out of the same state can have the same label, and states can have no arrows coming out with a specific label $a \in \Sigma$ (corresponding to the fact that $\delta(q, \mathsf{a}) = \emptyset$).

1. The NFA recognizing $\{\mathsf{a}^n\mathsf{b}^m \mid n, m \in \mathbb{N}\}$ is simpler than the DFA we drew above.

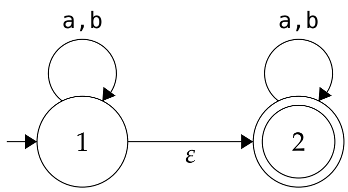


2. For any DFA M, we can easily construct an NFA M' that recognizes the reverse of $L(M)$, i.e.: $L(M') = \{w^{\mathcal{R}} \mid w \in L(M)\}$, where $^{\mathcal{R}}$ denotes the reverse of a word. Indeed, from the representation of a DFA, we can reverse all arrows, make the initial state an accepting state and add a pseudo initial state with ε–transitions to all the old accepting states as shown by the illustration below.

   The reverse of $\{\mathsf{a}^n\mathsf{b}^m \mid n, m \in \mathbb{N}\}$ is $\{\mathsf{b}^n\mathsf{a}^m \mid n, m \in \mathbb{N}\}$ and it is recognized by the following NFA.



3. Finding the complement of an NFA is not as easy as for DFAs. For instance, consider the following NFA on the alphabet $\{\mathtt{a}, \mathtt{b}\}$.



It recognizes all words generated by the alphabet, but if you swap the accepting and non-accepting states, the new NFA will also accept all words.

Remark 314. Since an NFA can have ε–transitions, the extension of δ to Σ^* is defined differently, $\delta^*(x, \varepsilon)$ is the ε–**closure** of x and $\delta^*(x, w)$ is defined inductively. More formally,

$$\delta^*(x, \varepsilon) = \{q \in Q : \exists\{x = x_0, x_2, \ldots, x_n = q\}, \forall 1 \leq i \leq n, x_i \in \delta(x_{i-1}, \varepsilon)\}$$
$$\delta^*(x, a \cdot w) = \bigcup_{q \in \delta(x,a)} \bigcup_{q' \in \delta^*(q,w)} \delta^*(q', \varepsilon).$$

Intuitively, $\delta^*(x, w)$ is the set of words that x can reach while reading the input w, taking into account that the machine can take any ε–transition.

One question that quickly arises is whether there are languages that can be recognized by an NFA, but not by any DFA. The converse is clearly false because any DFA can be written as an NFA where the image of the transition function only contains singletons, i.e.: there is only one possible state for each transition.

Surprisingly, the original question also has a negative answer. In other words, any NFA has a DFA recognizing the same language. To prove this, we describe the powerset construction transforming an NFA into an equivalent DFA.

Proof sketch. Let $M = (\Sigma, Q, q_0, \delta, F)$ be an NFA, we construct an equivalent DFA $M' = (\Sigma, Q', q'_0, \delta', F')$ as follows.

- The states of M' are sets of states of M, that is $Q' = \mathcal{P}(Q)$.

- The initial state q_0' is the set of states that can be reached by reading no symbol when running M. Formally, we have

$$q_0' = \delta^*(q_0, \varepsilon).$$

- The transition function will simulate all the possible choices by transitioning between sets of states. For any $S \subseteq Q$ and $a \in \Sigma$,

$$\delta'(S, a) = \bigcup_{q \in S} \delta^*(q, a)$$

- A set of states is accepting if and only if it contains an accepting state.

It is left to show that $L(M) = L(M')$. $\square$

The transformation we just described implies that NFAs are not necessary and we could always work with DFAs. However, observe that the size of the automaton increases exponentially in this procedure, so it is not practical to work with the resulting DFA. Consequently, one could ask whether we could further transform the DFA into a smaller (maybe smallest) but equivalent DFA. This is called DFA minimization and there are several polynomial time algorithms that solve this problem (Hopcroft's and Moore's). Another algorithm by Brzozowski is conceptually much simpler although it runs in exponential time in the worst case.

Brzozowski's Algorithm

Definition 315. Let M be a DFA, **Brzozowski's algorithm** is the following procedure.

1. Reverse M to obtain an NFA $M^{\mathcal{R}}$ which recognizes the reverse of $L(M)$.

2. Use the powerset construction to obtain a DFA $D(M^{\mathcal{R}})$, still recognizing the reverse of $L(M)$.

3. Discard all unreachable states of $D(M^{\mathcal{R}})$, denote the automaton obtained with N.

4. Apply the same procedure (steps 1 to 3) to N, i.e.: reverse N, determinize the result and discard unreachable states to obtain O.

Proposition 316. *The final automata O is the automaton with the least number of states satisfying $L(O) = L(M)$. Another way to say this: if* $\textsf{rev}$ *denotes the reversing operation,* $\textsf{det}$ *denotes the determinization and* $\textsf{reach}$ *denotes the operation of removing unreachable states, then*

$$\textsf{reach}(\textsf{det}(\textsf{rev}(\textsf{reach}(\textsf{det}(\textsf{rev}(M))))))$$

is the minimal automaton equivalent to M.

Remark 317. This result should be very surprising. Brzozowski's algorithm simply reverses the automaton twice to obtain a minimal form. Moreover, during the procedure, there are two steps inducing an exponential blow-up of the number of states. Indeed, determinization via the powerset construction leads to an automaton with $2^{|Q|}$ states. That means that in the worst cases, $\mathsf{det}(\mathsf{rev}(\mathsf{reach}(\mathsf{det}(\mathsf{rev}(M)))))$ could have $2^{2^{|Q|}}$ states. This is the reason why Brzozowski's algorithm can take exponential time. Nonetheless, in the end, you still end up with a minimal automaton. Another surprising thing, this algorithm often performs better than the worst case scenario.

This algorithm was first proven correct using complex combinatorial arguments in 1963, but in this assignment, we will give a much simpler proof using coalgebras. The next section is dedicated to introducing coalgebras and coinduction via their duals, algebras and induction.

Brzozowski's Algorithm Coalgebraically

In this section we follow the proof given in this paper (answers are in there).

Reachability and Observability

Let A be an alphabet, and (X, i, δ, F) be deterministic automaton, where X is the set of states, $\delta : X \times A \to X$ is the transition function, $i \in X$ is the initial state and F is the set of accepting states. For our purposes, we will view i as a function $i : \mathbf{1} \to X$, F as a function $f : X \to 2 = \{0, 1\}$ and δ will also denote the curried version $\delta : X \to X^A$. This leads to the following simple representation of the automaton.

$$\begin{tikzcd} \mathbf{1} \arrow[dr, "i"] & & 2 \\ & X \arrow[ur, "f"] \arrow[d, "\delta"] & \\ & X^A & \end{tikzcd} \tag{147}$$

In the light of the previous sections, we decompose this automaton into an algebra and a coalgebra.

First, we have the algebra $[i, \delta] : \mathbf{1} + (A \times X) \to X$ for the functor $\mathbf{1} + A \times (-)$. Recall from Exercise 285 that $(A^*, [\mathsf{nil} := \varepsilon, \mathsf{cons}])$ is initial for the functor $1 + (A \times -)$, so after currying cons, we obtain a unique morphism $r : A^* \to X$ that makes the following diagram commute.

$$\begin{tikzcd} \mathbf{1} \arrow[d, "\varepsilon"'] \arrow[dr, "i"] & & 2 \\ A^* \arrow[r, dashed, "r"] \arrow[d, "\mathsf{cons}"'] & X \arrow[ur, "f"] \arrow[d, "\delta"] & \\ (A^*)^A \arrow[r, dashed, "r^A"'] & X^A & \end{tikzcd} \tag{148}$$

Explicitly, the commutativity of the left half of diagram 148 amounts to

$$r(\varepsilon) = i \quad \text{and} \quad r(\mathsf{cons}(w, a)) = \delta(r(w), (a)), \forall a \in A$$

We infer that for any word $w \in A^*$, $r(w)$ is the state reached by the automaton after starting in state i and reading input w.

Second, we have the coalgebra $(f, \delta) : X \to 2 \times X^A$. Recall from diagram 143 that $(2^{A^*}, (\varepsilon?, \omega))$ is final for the functor $2 \times (-)^A$. Thus, we obtain a unique morphism $o : X \to 2^{A^*}$ that completes diagram 148.

$$\begin{tikzcd}
\mathbf{1} \arrow[d, "\mathsf{nil}"'] \arrow[dr, "i"] & & 2 \\
A^* \arrow[r, dashed, "r"] \arrow[d, "\mathsf{cons}"'] & X \arrow[ur, "f"] \arrow[r, dashed, "o"] \arrow[d, "\delta"] & 2^{A^*} \arrow[u, "\varepsilon?"'] \arrow[d, "\omega"] \\
(A^*)^A \arrow[r, dashed, "r^A"'] & X^A \arrow[r, dashed, "o^A"'] & (2^{A^*})^A
\end{tikzcd} \tag{149}$$

We recall from Exercise 298 that for any $x \in X$, $o(x)$ is the language in accepted by the automaton if started on state x.

Definition 318. In the setting above, the automaton (X, i, δ, f) is said to be

1. **reachable** if r is surjective,
2. **observable** if o is injective, and
3. **minimal** if it is both reachable and observable.

Exercise 319 (1pts). (a) Describe, in automata theoretic terms, what reachability and observability mean.

(b) Informally explain why an automaton has a minimal number of states if and only if it is both reachable and observable.

Reversing an Automaton

We will use our new method of representing automata to give the construction of an automata which recognizes the reverse language. Morally, our procedure does the same thing as the original reversing algorithm and the powerset construction at the same time.

In the sequel, let $2^{(-)}$ denote the contravariant powerset functor, namely, it sends X to 2^X and $f : X \to Y$ to $2^f : 2^Y \to 2^X = S \mapsto \{x \in X : f(x) \in S\}$.

The reversed powerset construction goes like this. Given a transition function $\delta : X \to X^A$, we can uncurry it to get $\delta : X \times A \to X$, then apply $2^{(-)}$ to obtain $2^\delta : 2^X \to 2^{X \times A}$ and finally curry the elements in the codomain to obtain $2^\delta : 2^X \to (2^X)^A$. This is now a transition function for an automata whose states are set of states of the original automata.

Exercise 320 (3pts). (a) Describe the action of 2^δ.

(b) Continue applying $2^{(-)}$ to the L.H.S. of diagram (149) to obtain (150). Briefly describe what each depicted morphism does.

$$\begin{tikzcd} & 2 \\ 2^X \arrow[ur, "2^i"] \arrow[r, "2^r"] \arrow[d, "2^\delta"'] & 2^{A^*} \arrow[u, "2^{\mathsf{nil}}"'] \arrow[d, "2^{\mathsf{cons}}"] \\ (2^X)^A \arrow[r, "2^{r^A}"'] & (2^{A^*})^A \end{tikzcd} \tag{150}$$

Warning: For cons, you will need to use the same trick as for δ.

By currying $f : X \to 2$ to $f : \mathbf{1} \to 2^X$, we obtain an $\mathbf{1} + A \times (-)$-algebra structure on 2^X which, by initiality of A^*, gives the following diagram.

$$\begin{tikzcd} \mathbf{1} \arrow[d, "\mathsf{nil}"'] \arrow[dr, "f"] & & 2 \\ A^* \arrow[r, dashed, "R"] \arrow[d, "\mathsf{cons}"'] & 2^X \arrow[ur, "2^i"] \arrow[r, "2^r"] \arrow[d, "2^\delta"'] & 2^{A^*} \arrow[u, "2^{\mathsf{nil}}"'] \arrow[d, "2^{\mathsf{cons}}"] \\ (A^*)^A \arrow[r, dashed, "R^A"'] & (2^X)^A \arrow[r, "2^{r^A}"'] & (2^{A^*})^A \end{tikzcd} \tag{151}$$

This is very close to what we are looking for:

- We have an automaton whose states are 2^X,
- its initial state is the set of states which contain at least one final state of the original automaton,
- its final states are the set of states which contain the initial state of the original automaton, and
- the morphism R tells us which states are reachable.

However, the R.H.S. of (151) is not excatly what we need to talk about observability of this automaton. You can see that $2^{\mathsf{nil}} = \varepsilon?$, but 2^{cons} is a kind of reversed version of ω, the concatenation is done in the opposite way.

Exercise 321 (1pts)**.** Describe the unique way to complete (151) into (152).

$$\begin{tikzcd} \mathbf{1} \arrow[d, "\mathsf{nil}"'] \arrow[dr, "f"] & & 2 \\ A^* \arrow[r, dashed, "R"] \arrow[d, "\mathsf{cons}"'] & 2^X \arrow[ur, "2^i"] \arrow[r, dashed, "O"] \arrow[d, "2^\delta"'] & 2^{A^*} \arrow[u, "\varepsilon?"'] \arrow[d, "\omega"] \\ (A^*)^A \arrow[r, dashed, "R^A"'] & (2^X)^A \arrow[r, dashed, "O^A"'] & (2^{A^*})^A \end{tikzcd} \tag{152}$$

In conclusion, we have an automaton $(2^X, f, 2^\delta, 2^i)$ which is reachable if R is surjective and observable if O is injective. Moreover, we can show two imporant properties.

Exercise 322 (4pts)**.** (a) If (X, i, δ, f) recognizes the language L, then $(2^X, f, 2^\delta, 2^i)$ recognizes the reverse of L.

(b) If (X, i, δ, f) is reachable, then $(2^X, f, 2^\delta, 2^i)$ is observable.

Correctness of the Algorithm

Exercise 323 (1pt)**.** Show that Brzozowski's algorithm is correct. Namely, if (X, i, δ, f) recognizes a language L, then

- applying the (new) reverse construction,
- keeping only the reachable states,
- applying the reverse construction again, and
- keeping only the reachable states

yields an automaton which is minimal and recognizes the language L.

Solutions to Exercises

Solutions to Chapter

Solution to Exercise 73. Take any monoid M with an idempotent element $x \neq 1_M$ (it satisfies $x \cdot x = x$). Letting $\mathbf{C}$ be $\mathbf{B}(M)$ and $\mathbf{C}'$ contain the object $*$ and only the morphism x yields a suitable example because the identity in $\mathbf{C}'$ is x. $\square$

Solution to Exercise 90. On morphisms, we define $\Delta_{\mathbf{C}}(f) = (f, f)$. The functoriality properties hold because everything in $\mathbf{C} \times \mathbf{C}$ is done componentwise.

i. For $f : X \to Y$, we have $(f, f) : (X, X) \to (Y, Y)$.

ii. For $f : X \to Y$ and $g : Y \to Z$, we have $(g, g) \circ (f, f) = (g \circ f, g \circ f)$.

iii. For any $X \in \mathbf{C}_0$, we have $\Delta_{\mathbf{C}}(\mathrm{id}_X) = (\mathrm{id}_X, \mathrm{id}_X) = \mathrm{id}_{(X,X)}$.

$\square$

Solution to Exercise 92. A quick way to show $F(X, -)$ is a functor is to recognize it as the composition of F with $X \times \mathrm{id}_{\mathbf{C}'}$, where X is the constant functor at X. Similarly, $F(-, Y) := F \circ (\mathrm{id}_{\mathbf{C}} \times Y)$. $\square$

Solution to Exercise 93. Let us show the three properties of functoriality.

i. For any $(f, g) : (X, Y) \to (X', Y')$

$\square$

Solutions to Chapter

Solution to Exercise 110. Let us have two morphisms $f : X \to Y$ and $g : Y \to Z$.

- Suppose f and g are monic. For any $h_1, h_2 : Z \to Z'$ satisfying $h_1 \circ g \circ f = h_2 \circ g \circ f$, monicity of f implies $h_1 \circ g = h_2 \circ g$ which in turn, by monicity of g imply $h_1 = h_2$. Thus, $g \circ f$ is monic.
- We apply duality. Suppose f and g are epic, then f^{op} and g^{op} are monic so $g \circ f^{\mathrm{op}} = f^{\mathrm{op}} \circ g^{\mathrm{op}}$ is monic, thus $g \circ f$ is epic.
- If f and g are isomorphisms, then it is easy to check that $f^{-1} \circ g^{-1}$ is the inverse of $g \circ f$, implying $g \circ f$ is an isomorphism.

□

Solution to Exercise 125. Let us have three monomorphisms $m : Y \hookrightarrow X$, $n : Z \hookrightarrow X$ and $o : W \hookrightarrow X$.

Reflexivity: We have $m \circ \mathrm{id}_Y = m$ thus $m \sim m$.

Symmetry: Suppose that $m \sim n$, namely, there is an isomorphism $i : Y \to X$ such that $m = n \circ i$. Then, pre-composing with the isomorphism i^{-1} yields $m \circ i^{-1} = n$ which implies $n \sim m$.

Transitivity: If $m \sim n$ and $n \sim o$, then there exist isomorphisms $i : Y \to Z$ and $i' : W \to Z$ satisfying $m = n \circ i$ and $n = o \circ i'$. Therefore, we have $m = o \circ i' \circ i$ which implies $m \sim o$.[190] □

[190] Recall that the composition of two isomorphisms is an isomorphism.

Solution to Exercise 127. Let us have five monomorphisms $m : Y \hookrightarrow X$, $m : Y' \hookrightarrow X$, $n : Z \hookrightarrow X$, $n' : Z' \hookrightarrow X$ and $o : W \hookrightarrow X$.[191]

[191] Recall that we often use m to refer to $[m]$.

Well-defined: Suppose that $m \leq n$, $m' \sim m$ and $n \sim n'$, namely, there is a morphism $k : Y \to Z$ and isomorphisms $i : Y \circ Y'$ and $i' : Z' \to Z$ such that $m = n \circ k$, $m' = m \circ i$ and $n = n' \circ i'$. Combining these equalities yields $m' = n' \circ i' \circ k \circ i$ which witnesses $m' \leq n'$.

Reflexivity: We have $m \circ \mathrm{id}_Y = m$ thus $m \leq m$.

Antisymmetry: If $m \leq n$ and $n \leq m$, then there exist morphisms $k : Y \to Z$ and $k' : Z \to Y$ satisfying $m = n \circ k$ and $n = m \circ k'$. Combining these two equalities yield $m = m \circ k' \circ k$ and $n = n \circ k \circ k'$. Therefore, since m and n are monic, we infer that $k' \circ k = \mathrm{id}_Y$ and $k \circ k' = \mathrm{id}_Z$. This means k is an isomorphism and $m \sim n$ (so $[m] = [n]$).

Transitivity: If $m \leq n$ and $n \leq o$, then there exist morphisms $k : Y \to Z$ and $k' : W \to Z$ satisfying $m = n \circ k$ and $n = o \circ k'$. Therefore, we have $m = o \circ k' \circ k$ which implies $m \leq o$. □

Solutions to Chapter

Solution to Exercise 136. The existence and uniqueness of $\prod_{i \in I} f_i$ is given by the universal property of the product $\prod_{i \in I} Y_i$ with for each $j \in I$, the morphism $f_j \circ \pi_j : \prod_{i \in I} X_i \to Y_j$. □

Solution to Exercise 154. ($\Rightarrow$) Suppose $f : X \to Y$ is monic, commutativity of (34) is trivial. For any $X \xleftarrow{g} Z \xrightarrow{h} X$ satisfying $f \circ g = f \circ h$, we have $g = h$. Thus $g = h$ is the mediating morphism ! of (153), it is unique because $\mathrm{id}_X \circ m = g$ implies $m = g$.

($\Leftarrow$) For any $g, h : Z \to X$ satisfying $f \circ g = f \circ h$, the universal property of the pullback tells us there is a unique $! : Z \to X$ making (153) commute. Since ! satisfies $g = \mathrm{id}_X \circ ! = h$, we conclude $g = ! = h$, thus f is a monomorphism.

$$\begin{tikzcd} Z \arrow[drr, bend left, "h"] \arrow[dr, dashed, "!"] \arrow[ddr, bend right, "g"'] & & \\ & X \arrow[r, "\mathrm{id}_X"] \arrow[d, "\mathrm{id}_X"'] \arrow[dr, phantom, "\lrcorner", very near start] & X \arrow[d, "f"] \\ & X \arrow[r, "f"'] & Y \end{tikzcd} \quad (153)$$

The dual statement is that $f : X \to Y$ is epic if and only if (154) is a pushout. We leave the proof to you. □

$$\begin{tikzcd} X \arrow[r, "f"] \arrow[d, "f"'] & Y \arrow[d, "\mathrm{id}_X"] \\ Y \arrow[r, "\mathrm{id}_X"'] \arrow[ur, phantom, "\ulcorner", very near end] & Y \end{tikzcd} \quad (154)$$

Solution to Exercise 179. We will show that if **C** has all pullbacks and a terminal object, then it has all finite products and equalizers. This implies, using Remark 177, that **C** is finitely complete.

For finite products, recall that it is enough to show that $\mathbf{C}$ has all binary products as it already has the empty product (the terminal object). We claim that the pullback of $A \xrightarrow{[]} \mathbf{1} \xleftarrow{[]} B$ is the binary product $A \times B$.

$$\begin{tikzcd} A \times_{\mathbf{1}} B \arrow[r, "\pi_B"] \arrow[d, "\pi_A"'] \arrow[dr, phantom, "\lrcorner", very near start] & B \arrow[d, "{[]}"] \\ A \arrow[r, "{[]}"'] & \mathbf{1} \end{tikzcd} \tag{155}$$

Indeed, for any $A \xleftarrow{p_A} X \xrightarrow{p_B} B$, we have $[] \circ p_A = [] \circ p_B$, thus, there is a unique morphism $! : X \to A \times_{\mathbf{1}} B$ making (156) commute. Since the commutativity of the squares always hold, this is equivalent to the unviersal property of the binary product. Hence $A \times B \cong A \times_{\mathbf{1}} B$.

$$\begin{tikzcd} X \arrow[dr, dashed, "!"] \arrow[drr, bend left, "p_B"] \arrow[ddr, bend right, "p_A"'] & & \\ & A \times_{\mathbf{1}} B \arrow[r, "\pi_B"] \arrow[d, "\pi_A"'] \arrow[dr, phantom, "\lrcorner", very near start] & B \arrow[d, "{[]}"] \\ & A \arrow[r, "{[]}"'] & \mathbf{1} \end{tikzcd} \tag{156}$$

$\square$

Solutions to Chapter

Solution to Exercise 190. The terminal object of $\mathbf{C}/X$ is the identity morphism $\mathrm{id}_X : X \to X$. For any object of the slice category $f : A \to X$, we have the commutative triangle (157) with $! = f$. Uniqueness of $!$ follows from $\mathrm{id}_X \circ ! = f \implies ! = f$.

$$\begin{tikzcd} A \arrow[rr, dashed, "f"] \arrow[dr, "f"'] & & X \arrow[dl, "\mathrm{id}_X"] \\ & X & \end{tikzcd} \tag{157}$$

The dual statement is that id_X is the initial object of $X/\mathbf{C}$. $\square$

Solutions to Chapter

Solution to Exercise 196. ($\Rightarrow$) For any $g : Y \to Y'$, the naturality of ϕ yields this commutative square.

$$\begin{tikzcd} F(X,Y) \arrow[r, "\phi_{X,Y}"] \arrow[d, "{F(X,g)=F(\mathrm{id}_X,g)}"'] & G(X,Y) \arrow[d, "{G(\mathrm{id}_X,g)=G(X,g)}"] \\ F(X,Y') \arrow[r, "\phi_{X,Y'}"'] & G(X,Y') \end{tikzcd} \tag{158}$$

We conclude that $\phi_{X,-}$ is a natural transformation $F(X,-)$. A symmetric argument works for $\phi_{-,Y}$ (see (159)).

$$\begin{tikzcd} F(X,Y) \arrow[r, "\phi_{X,Y}"] \arrow[d, "{F(f,\mathrm{id}_Y)}"'] & G(X,Y) \arrow[d, "{G(f,\mathrm{id}_Y)}"] \\ F(X',Y) \arrow[r, "\phi_{X',Y}"'] & G(X',Y) \end{tikzcd} \tag{159}$$

($\Leftarrow$) For any $(f,g) : (X,Y) \to (X',Y')$, we note that, by functoriality, $F(f,g) = F(f, \mathrm{id}_{Y'}) \circ F(\mathrm{id}_X, g)$ and similarly for G. Thus, we can combine the naturality of

$\phi_{X,-}$ and $\phi_{-,Y}$ to obtain the commutativity of $\phi_{X,Y}$ as shown in (160).

$$\begin{tikzcd}
F(X,Y) \arrow[r, "\phi_{X,Y}"] \arrow[d, "F(\mathrm{id}_X, g)"] \arrow[dd, bend right=60, "F(f,g)"'] & G(X,Y) \arrow[d, "G(\mathrm{id}_X, g)"'] \arrow[dd, bend left=60, "G(f,g)"] \\
F(X,Y') \arrow[r, "\phi_{X,Y'}"] \arrow[d, "F(f, \mathrm{id}_{Y'})"] & G(X,Y') \arrow[d, "G(f, \mathrm{id}_{Y'})"'] \\
F(X',Y') \arrow[r, "\phi_{X',Y'}"'] & G(X',Y')
\end{tikzcd} \tag{160}$$

□

Solution to Exercise 200. Let $F, G : \mathbf{C} \rightsquigarrow \mathbf{D}$ be functors.

($\Rightarrow$) If $\phi : F \Rightarrow G$ is a natural isomorphism, then it has an inverse $\phi^{-1} : G \Rightarrow F$ which satisfies $\phi \cdot \phi^{-1} = \mathbb{1}_G$ and $\phi^{-1} \cdot \phi = \mathbb{1}_F$. Looking at each components, we find $\phi_X \circ (\phi^{-1})_X = \mathrm{id}_X$ and $(\phi^{-1})_X \circ \phi_X = \mathrm{id}_X$, hence they are isomorphisms.

($\Leftarrow$) Let $\phi : F \Rightarrow G$ be a natural transformation such that ϕ_X is an isomorphism for each $X \in \mathbf{C}_0$. We claim that the family ϕ_X^{-1} is the inverse of ϕ. After we show that this family is a natural transformation $G \Rightarrow F$, the construction implies it is the inverse of ϕ. For any $f : X \to Y \in \mathbf{C}_1$, the naturality of ϕ implies $\phi_Y \circ F(f) = G(f) \circ \phi_X$. Pre-composing with ϕ_X^{-1}, we have $G(f) = \phi_Y \circ F(f) \circ \phi_X^{-1}$ and therefore

$$\phi_Y^{-1} \circ G(f) = \phi_Y^{-1} \circ \phi_Y \circ F(f) \circ \phi_X^{-1} = F(f) \circ \phi_X^{-1}$$

yields the naturality of ϕ^{-1}. □

Solution to Exercise 210. On morphisms, this functor must send a pair of natural transformations $\eta : F \Rightarrow F'$ and $\phi : G \Rightarrow G'$ to a natural transformation $FG \Rightarrow F'G'$. This is exactly what horizontal composition does.

To see that horizontal composition is functorial, first note that $\mathbb{1}_F \diamond \mathbb{1}_G = \mathbb{1}_{FG}$. Next, the fact that horizontal composition commutes with composition of functors is exactly the interchange identity. □

Solution to Exercise 221. We need to show that $\simeq$ is reflexive, symmetric and transitive. Symmetry is trivial because the definition of $\mathbf{C} \simeq \mathbf{D}$ is symmetric. Reflexivity follows from the fact that the identity functor on any category is fully faithful and essentially surjective.

For transitivity, given the categories and functors represented in (161) with natural isomorphisms $\phi : FG \Rightarrow \mathrm{id}_{\mathbf{D}}$, $\psi : GF \Rightarrow \mathrm{id}_{\mathbf{C}}$, $\phi' : F'G' \Rightarrow \mathrm{id}_{\mathbf{E}}$ and $\psi' : G'F' \Rightarrow \mathrm{id}_{\mathbf{D}}$, we claim that the composition $G \circ G'$ is the quasi-inverse of $F' \circ F$.

$$\begin{tikzcd}
\mathbf{C} \arrow[r, squiggly, bend left, "F"] & \mathbf{D} \arrow[l, squiggly, bend left, "G"] \arrow[r, squiggly, bend left, "F'"] & \mathbf{E} \arrow[l, squiggly, bend left, "G'"]
\end{tikzcd} \tag{161}$$

Since the biaction of functors preserves natural isomorphisms,[192] we have two natural isomorphisms

$$\phi' \cdot (F'\phi G') : F'FGG' \Rightarrow \mathrm{id}_{\mathbf{E}} \text{ and } \psi \cdot (G\psi' F) : GG'F'F \Rightarrow \mathrm{id}_{\mathbf{C}},$$

which shows $\mathbf{C} \simeq \mathbf{E}$. □

[192] This holds because acting on the left or right with a functor is a functor, part of this is shown in the next solution and it also follows from the previous exercise.

Solution to Exercise 222. We will show the following two implications

$$\mathbf{C} \simeq \mathbf{C}' \implies [\mathbf{C}, \mathbf{D}] \simeq [\mathbf{C}', \mathbf{D}]$$
$$\mathbf{D} \simeq \mathbf{D}' \implies [\mathbf{C}, \mathbf{D}] \simeq [\mathbf{C}, \mathbf{D}']$$

and infer that $\mathbf{C} \simeq \mathbf{C}'$ and $\mathbf{D} \simeq \mathbf{D}'$ implies

$$[\mathbf{C}, \mathbf{D}] \simeq [\mathbf{C}', \mathbf{D}] \simeq [\mathbf{C}', \mathbf{D}'].$$

For the first implication, let $F : \mathbf{C} \rightsquigarrow \mathbf{C}'$ and $G : \mathbf{C}' \rightsquigarrow \mathbf{C}$ be quasi-inverses. We define the functor $(-)F : [\mathbf{C}', \mathbf{D}] \rightsquigarrow [\mathbf{C}, \mathbf{D}]$ that acts on functors by pre-composition and on natural transformations by the right action in Definition 203.[193] Similarly, we define the functor $(-)G : [\mathbf{C}, \mathbf{D}] \rightsquigarrow [\mathbf{C}', \mathbf{D}]$. We claim that $(-)F$ and $(-)G$ are quasi-inverses.

Let $\Phi : GF \Rightarrow \mathrm{id}_{\mathbf{C}}$ be a natural isomorphism witnessing F and G being quasi-inverses, then $(-)\Phi$ is a natural isomorphism from $(-)GF$ to $\mathrm{id}_{[\mathbf{C},\mathbf{D}]}$. Indeed, for any $\phi : H \Rightarrow H' \in [\mathbf{C}, \mathbf{D}]_1$, (162) commutes as the top path and bottom path are both equal to $\phi \diamond \Phi$ and $H\Phi$ is an isomorphism because Φ is and functors preserve isomorphisms.

$$\begin{tikzcd} HGF \arrow[r, "H\Phi"] \arrow[d, "\phi GF"'] & H \arrow[d, "\phi"] \\ H'GF \arrow[r, "H'\Phi"'] & H' \end{tikzcd} \tag{162}$$

We leave to you the symmetric argument showing $(-)FG \cong \mathrm{id}_{[\mathbf{C}',\mathbf{D}]}$ and the similar argument for the second implication. □

[193] i.e.: $H : \mathbf{C} \rightsquigarrow \mathbf{D}$ is mapped to $HF = H \circ F$ and $\phi : H \Rightarrow H'$ is mapped to ϕF. Functoriality follows from the properties of the right action.
Another way to show functoriality is to recall that $\phi F = \phi \diamond \mathbb{1}_F$ and hence $(-)F$ is the composition of the functor

$$\mathrm{id}_{[\mathbf{C}',\mathbf{D}]} \times F : [\mathbf{C}', \mathbf{D}] \times \mathbf{1} \rightsquigarrow [\mathbf{C}', \mathbf{D}] \times [\mathbf{C}, \mathbf{C}']$$

with the horizontal composition functor defined in Exercise 210.

Solutions to Chapter

Solutions to Chapter

Solutions to Chapter

Solutions to Chapter