

Set Operations as Toy Examples in Category Theory

hora-algebra

English translation by Astra

Even the most elementary set operations are full of categorical structure. When learning category theory, there is nothing like having simple, familiar examples to hand. Starting by viewing sets and truth values as categories, this text offers toy examples of categories, functors, natural transformations, limits, colimits, adjunctions, monads, cartesian closed categories, Kan extensions, and their properties, all within the world of set operations.

Contents

1	Introduction	3
1.1	What this text might offer, and what it cannot	3
1.2	A note on the relation to posets	3
1.3	A few other notes	4
2	Categories, functors, and natural transformations	4
2.1	Categories from sets and truth values	4
2.2	Functors, maps, and predicates	4
2.3	Natural transformations and power sets	5
2.4	Precomposition between functor categories and inverse images	6
2.5	Postcomposition between functor categories and complements	7
3	Limits and colimits	8
3.1	Limits and colimits as intersections and unions	8
3.2	Limits and colimits in functor categories	8
4	Adjunctions	9
4.1	Adjoints of inverse image and subsets	10
4.2	Adjoints of inverse image and quantifiers	12
4.3	Adjoints of negation and exchanging negation with quantifiers	14
4.4	Mutually right adjoints and Galois connections from binary relations, I	15
4.5	Equivalences induced by adjunctions and the Galois correspondence	17
4.6	Reflective subcategories, the adjoint functor theorem, and generation, I	19
4.7	Factoring an idempotent adjunction	22
5	Monads	23
5.1	Monads, Eilenberg–Moore categories, and generation, II	24
5.2	Free monads and generation, III	25
5.3	Monads from adjunctions and closure operators from relations	30
5.4	Kleisli categories and transitivity of generation	31
5.5	Colimit preservation: orders, topologies, and finitary algebras	33
6	Cartesian Closed Category	37
6.1	Power sets, implication, and modus ponens	37
6.2	The dual picture and set difference	40
6.3	Mutually right adjoints	40
6.4	Cartesian closed functors, implication, and the projection formula	41
7	Kan extensions	42
7.1	The pointwise formula and power-set-valued Kan extensions	42
7.2	Truth-value-valued Kan extensions, inverse image, and quantifiers	44
7.3	Yoneda and power sets as free cocompletions	45
7.4	Universal adjunctions and Galois connections from relations, II	46

1 Introduction

1.1 What this text might offer, and what it cannot

This text describes set operations as toy examples of category theory. Here is what it might offer:

- Elementary, simple examples of categorical concepts for beginners.
- A way of organizing various topics about set operations around categorical concepts, for readers with some category theory under their belts.

On the other hand, please do not expect too much of the following:

- Proofs of nontrivial mathematical theorems, whether about category theory or set operations.
- Useful mathematical tools.

For instance, we will use category-theoretic theorems to prove completely obvious facts about set operations. If your aim is to learn those facts, this may look like nothing more than a detour through category theory. But the point of these proofs is not simply to establish that the statements about sets are true.

Here is the sort of example I have in mind for beginners. We use the adjoint functor theorem to show that “generation” can be described by intersections (Theorem 4.6.13). The statement itself is not really what I want to convey,¹ but the idea behind the proof. The general adjoint functor theorem constructs a left adjoint using limits. If that method felt unfamiliar, I hope the proof of Theorem 4.6.13 makes you think, “Oh, I have seen this before, in something this elementary!” Ideas in categorical definitions and proofs may seem strange and unapproachable; I want you to discover that ordinary, simple things you already know are examples of them.

For readers with some category theory, here is the sort of organization I mean. We describe the quantifier $\forall$ as a Kan extension. The general theory tells us that a functor may have two universal extensions. It lets us turn the intuition that quantifiers are two universal ways to remove a variable from a predicate into a mathematical fact in the language of categories.

There are also limits imposed by my own abilities. As I get better at category theory, I expect almost every part of this text could be improved, and more material added.

Still, some readers may find parts of the sections on adjunctions and monads interesting as discussions of set operations in their own right. They mention a few facts about sets that I only noticed by thinking about their categorical connections.

1.2 A note on the relation to posets

Most categories in this text are built by applying operations to sets and truth values. In particular, almost all are preorders—indeed, posets. A substantial proportion of the statements we prove already hold at the level of posets. I have singled out set operations to keep things familiar, but many statements really belong in the more general setting of posets.

I suspect that some statements should be treated using categories enriched in **2**, but I do not yet know enough to write that out explicitly.

¹Though if you have not seen it before, I certainly want you to know it!

1.3 A few other notes

Remarks contain material that is not itself a set-operation toy example: counterparts elsewhere in mathematics, informal observations, or statements that hold more generally for posets.

In the Japanese original, I have ended up mixing Japanese and English categorical terminology. I intend the Japanese terms to follow [2].

Each section begins with a summary. You can read all the summaries first.

Corrections and suggestions are always welcome, whether or not I explicitly ask for help at a particular point. Please contact me on Twitter at @hora_algebra.

2 Categories, functors, and natural transformations

Here we recast our set-theoretic ingredients in categorical language. In a sense, everything in this section is obvious.²

In Section 2.1, we regard sets and truth values as categories; in Section 2.2, we consider functors between them.

Section 2.3 views the power set, our main category, as a functor category. This is very easy to see, but important throughout the text. For instance, Section 3.2 examines pointwise limits in functor categories, and Section 7.3 treats the power set as a presheaf category and considers the Yoneda embedding.

Section 2.4 identifies inverse image with precomposition. This fact,³ means that Kan extensions provide left and right adjoints, leading to Sections 4.1, 4.2, and 7.2.

Section 2.4 describes negation as postcomposition. Since precomposition and postcomposition commute, these obvious reformulations give the commutation of negation and inverse image. This leads to Section 4.3.

2.1 Categories from sets and truth values

The categories we study are built from two extremely simple⁴ kinds of category.

Definition 2.1.1 (A set as a category). For a set X , write X also for the category whose objects are the elements of X and whose only morphisms are identities.

Definition 2.1.2 (Truth values as a category). Write **2** for the category with exactly two objects, true and false, and, besides their identities, just one morphism $\text{false} \rightarrow \text{true}$.

We will regard sets and truth values as categories in this way. So we will say things like “a functor from a set to the truth values.”

Recall that a poset naturally gives a category. Both categories above arise this way: a set has the trivial order, and truth values are ordered by implication $\Rightarrow$. Ordinals are posets too, hence categories; the truth-value category is the one corresponding to the ordinal 2, which explains our notation **2**. Almost every category we meet will be a poset.⁵

2.2 Functors, maps, and predicates

Functors out of a set are very simple. Choose freely where the objects go; since the only morphisms are identities, their images are then forced. Functoriality is automatic for the same reason. We mainly want sets and truth values as codomains, but let us state the observation generally first.

²I am not promising that later sections will contain anything non-obvious.

³Together with the bicompleteness of **2**.

⁴Mathematical terminology and ordinary language will sometimes collide. Unless the mathematical meaning is clearly intended, please read the words in their ordinary sense.

⁵My knowledge is too limited to be definite, but from the viewpoint of preorders as truth-value-enriched categories (<https://ncatlab.org/nlab/show/preorder#definition>), I suspect their closure under basic categorical operations should feel natural.

Proposition 2.2.1 (Functors out of a set). For a set X and a (small) category $\mathcal{C}$, functors $X \rightarrow \mathcal{C}$ correspond bijectively to maps $X \rightarrow \text{ob}(\mathcal{C})$.

Let us note that this correspondence is natural.

Remark 2.2.2 (Naturality of the correspondence). The correspondence above is natural in the sense of an adjunction: the functor $\text{Dis}(-) : \text{Set} \rightarrow \text{Cat}$ viewing a set as a discrete category is left adjoint to $\text{ob}(-) : \text{Cat} \rightarrow \text{Set}$.

We are interested in taking $\mathcal{C}$ to be a set or the truth values. The functors then become maps or predicates, respectively. A predicate on X is a map $X \rightarrow \{\text{true}, \text{false}\}$. For example, define a predicate ϕ on $\mathbb{Z}$ by $\phi(n) = \text{true}$ for even n and $\phi(n) = \text{false}$ for odd n . We think of it as expressing “ n is even.” You might fairly ask why I do not just call it the characteristic function of a subset.⁶ We will address that in the next subsection.

Corollary 2.2.3 (Maps and predicates as functors). For sets X, Y , maps $X \rightarrow Y$ correspond naturally and bijectively to functors $X \rightarrow \mathbf{2}$. For a set X , functors $X \rightarrow \mathbf{2}$ correspond bijectively to predicates on X .

2.3 Natural transformations and power sets

Once we have functors, we ought to look at natural transformations. Between maps of sets, there are only trivial natural transformations. Let us instead consider functors from a set to the truth values.

Take functors $F, G : X \rightarrow \mathbf{2}$. A natural transformation from F to G assigns to each $x \in X$ a morphism $F(x) \rightarrow G(x)$ in $\mathbf{2}$. Such a morphism is unique if it exists. Moreover, if a morphism $F(x) \rightarrow G(x)$ exists for every $x \in X$, naturality holds automatically. From the predicate viewpoint, a natural transformation is logical implication.

Proposition 2.3.1 (Logical implication as a natural transformation). For predicates (functors) $F, G : X \rightarrow \mathbf{2}$, a natural transformation $F \rightarrow G$ exists if and only if $F \Rightarrow G$: for every $x \in X$, there is a morphism $F(x) \rightarrow G(x)$ in $\mathbf{2}$. If it exists, it is unique.

Recall the bijection between predicates on X and subsets of X . To $\phi : X \rightarrow \{\text{true}, \text{false}\}$, we assign the inverse image of true, namely the set $\{x \in X \mid \phi(x)\}$ of elements where ϕ “holds.”⁷ To a subset $A \subset X$, we assign its characteristic function $\chi_A : X \rightarrow \{\text{true}, \text{false}\}$, equal to true on A and false outside it.

Remark 2.3.2 (Subobject classifiers). A brief digression: an object Ω giving a natural correspondence between (isomorphism classes of) subobjects of X and morphisms $X \rightarrow \Omega$ is called a subobject classifier. See [3], <https://ncatlab.org/nlab/show/subobject+classifier>, for details. Such objects occur in the class of categories called toposes, including Set —indeed, their existence is required by definition.⁸ My understanding⁹ is that predicates-as-functors correspond to subsets here because we are only looking at the set-like part of the situation, not because of Cat itself.

This gives a bijection between subsets of X and functors from X to the truth values. Let us see that it is an isomorphism of categories between the power set and the functor category. We view the power set as a category as follows.

⁶What I like about the predicate–subset correspondence is that it connects apparently different things: logical statements and parts of a set. That is why I treat predicates separately. There are also technical reasons, such as introducing quantifiers later.

⁷Since $\phi(x)$ is formally an element true or false, please read this notation informally.

⁸As of May 2021, I am studying topos theory.

⁹At least, this is my rather limited view as of January 2021.

Definition 2.3.3 (A power set as a category). For a set X , its power set $P(X)$ is a category via the inclusion order. We write this category as $P(X)$ too.

This is the main category of the text. The discussion can be summarized as follows: a power set is a category of functors to the truth values.

Theorem 2.3.4 (Power sets are functor categories). For a set X , there is an isomorphism of categories $P(X) \cong \mathbf{2}^X$ between its power set and the category of functors from X to the truth values.

We have now described the subset–predicate correspondence including its order structure. We are ready to use power sets as toy examples of facts and theorems about functor categories.

2.4 Precomposition between functor categories and inverse images

Given categories $\mathcal{C}, \mathcal{D}, \mathcal{E}$ and a functor $F : \mathcal{D} \rightarrow \mathcal{E}$, precomposition with F gives a functor $F^* : \mathcal{C}^{\mathcal{E}} \rightarrow \mathcal{C}^{\mathcal{D}}$, subject to size issues. Precomposition is important for several reasons. For example, asking for its left and right adjoints amounts to asking for Kan extensions—and hence includes asking for limits and colimits!¹⁰

What does this look like for set operations? For a map $f : X \rightarrow Y$, we have the following.

Proposition 2.4.1 (Inverse image as precomposition). For a map $f : X \rightarrow Y$, the precomposition functor $f^* : \mathbf{2}^Y \rightarrow \mathbf{2}^X$ corresponds, under Theorem 2.3.4, to the inverse-image map¹¹ $f^{-1} : P(Y) \rightarrow P(X)$.

More precisely, the diagram

$$\begin{array}{ccc} \mathbf{2}^Y & \xrightarrow{f^*} & \mathbf{2}^X \\ \downarrow & & \downarrow \\ P(Y) & \xrightarrow{f^{-1}} & P(X) \end{array}$$

commutes, where the vertical arrows are the isomorphisms from Theorem 2.3.4.

For later use, here are two analogous functors from ordinary mathematics.

Remark 2.4.2 (Restriction of representations). Take a group G , a subgroup¹² H , and a field $\mathbb{K}$. Restricting representations of G over $\mathbb{K}$ to H gives $\text{res}_H^G : \text{Vect}_{\mathbb{K}}^G \rightarrow \text{Vect}_{\mathbb{K}}^H$. Viewing the groups as one-object categories, this is precomposition with the inclusion $\iota : H \rightarrow G$.

Remark 2.4.3 (Including invertible discrete dynamical systems). A discrete dynamical system in a category $\mathcal{C}$ is a pair (X, f) consisting of an object X and an endomorphism $f : X \rightarrow X$. Call it invertible if f is an isomorphism. The category $\mathcal{C}$ can be anything: sets Set , topological spaces Top , manifolds Man , for instance. Let $\text{Dyn}_{\mathcal{C}}$ be the category of such systems and morphisms of $\mathcal{C}$ commuting with their endomorphisms; similarly, let $\text{Dyn}_{\mathcal{C}}^{\text{inv}}$ be the category of invertible systems. Regard the additive monoids $\mathbb{N}, \mathbb{Z}$ as one-object categories. Then $\text{Dyn}_{\mathcal{C}} \simeq \mathcal{C}^{\mathbb{N}}$ and $\text{Dyn}_{\mathcal{C}}^{\text{inv}} \simeq \mathcal{C}^{\mathbb{Z}}$. Under these equivalences, the inclusion $\text{Dyn}_{\mathcal{C}}^{\text{inv}} \rightarrow \text{Dyn}_{\mathcal{C}}$ is¹³ the precomposition functor $\mathcal{C}^{\mathbb{Z}} \rightarrow \mathcal{C}^{\mathbb{N}}$ induced by $\iota : \mathbb{N} \rightarrow \mathbb{Z}$.

¹⁰That is a slight overstatement: it amounts to asking for all Kan extensions of the same shape. See, for example, Proposition 6.1.5 of [1].

¹¹It is a functor, but a functor between posets is determined by its action on objects, so we sometimes think of it as a map between their sets of objects.

¹²It need not actually be a subgroup.

¹³At least up to natural isomorphism, with natural choices of the equivalences.

2.5 Postcomposition between functor categories and complements

We have seen examples of precomposition between functor categories. Dually, postcomposition gives functors too. Here we describe taking complements—or negation, from the predicate viewpoint—as postcomposition.

The basic observation is that $\mathbf{2}$ is contravariantly isomorphic to itself. Namely:

Definition 2.5.1 (The negation functor). Define the contravariant functor $\neg : \mathbf{2}^{\text{op}} \rightarrow \mathbf{2}$ by $\neg \text{true} = \text{false}$ and $\neg \text{false} = \text{true}$.

Its action on morphisms is forced.

Lemma 2.5.2 (Negation is a contravariant isomorphism). We have $\neg \circ \neg = \text{id}_{\mathbf{2}}$. In particular, negation is a contravariant automorphism of $\mathbf{2}$.

Now consider postcomposition with this functor.

Proposition 2.5.3 (Complementation is postcomposition). For a set X , the contravariant complement functor $(-)^c : P(X)^{\text{op}} \rightarrow P(X)$ corresponds, under Theorem 2.3.4, to postcomposition with $\neg$. In particular, taking complements is a contravariant automorphism of $P(X)$.

Proof. Under Theorem 2.3.4, taking complements is postcomposition with $\neg$, hence gives a contravariant functor. Here we use $(\mathbf{2}^{\text{op}})^X \cong (\mathbf{2}^{X^{\text{op}}})^{\text{op}} \cong (\mathbf{2}^X)^{\text{op}}$. Since $\neg \circ \neg = \text{id}_{\mathbf{2}}$, composing this contravariant functor with itself also gives the identity. Thus it is a contravariant automorphism. $\square$

One benefit of describing this as postcomposition is that its commutation with precomposition becomes immediate. In any category $\mathcal{C}$, morphisms $f : A \rightarrow B$ and $g : C \rightarrow D$ give a commutative diagram of hom-sets:

$$\begin{array}{ccc} \mathcal{C}(B, C) & \xrightarrow{g_*} & \mathcal{C}(B, D) \\ \downarrow f^* & & \downarrow f^* \\ \mathcal{C}(A, C) & \xrightarrow{g_*} & \mathcal{C}(A, D) \end{array}$$

This is just associativity in the definition of a category, and surely one of the most basic commutative diagrams. Applying it in Cat to $F : \mathcal{A} \rightarrow \mathcal{B}$ and $G : \mathcal{C} \rightarrow \mathcal{D}$ gives

$$\begin{array}{ccc} \mathcal{C}^{\mathcal{B}} & \xrightarrow{G_*} & \mathcal{D}^{\mathcal{B}} \\ \downarrow F^* & & \downarrow F^* \\ \mathcal{C}^{\mathcal{A}} & \xrightarrow{G_*} & \mathcal{D}^{\mathcal{A}} \end{array}$$

a diagram commuting on objects. It is also easy to check that it commutes on morphisms (natural transformations).

Remark 2.5.4 (The action on natural transformations). To describe the action on natural transformations generally, one can use, for example, the cartesian closed structure.¹⁴

Apply this observation to a map $f : X \rightarrow Y$ and the negation functor $\neg : \mathbf{2}^{\text{op}} \rightarrow \mathbf{2}$. We get:

$$\begin{array}{ccc} P(Y)^{\text{op}} & \xrightarrow{(-)^c} & P(Y) \\ \downarrow (f^{-1})^{\text{op}} & & \downarrow f^{-1} \\ P(X)^{\text{op}} & \xrightarrow{(-)^c} & P(X) \end{array}$$

¹⁴As of January 2021, I do not know enough to say for certain, but a natural generalization staying in the language of homs should probably use Cat -enriched categories, or strict 2-categories.

Proposition 2.5.5 (Inverse image commutes with complements). Precomposition and postcomposition commute. In particular, for $f : X \rightarrow Y$ and $U \subset Y$, we have $f^{-1}(U^c) = f^{-1}(U)^c$.

Proof. Inverse image is precomposition with f , and complementation is postcomposition with $\neg$. $\square$

Looking at adjoints turns this commutation into another form: the rules for exchanging negation and quantifiers. We will give examples from ordinary mathematics in the section on adjunctions.

3 Limits and colimits

This section describes limits and colimits in power sets.

Section 3.1 simply observes that limits and colimits in a power set are intersections and unions, while those in the truth values are conjunctions and disjunctions.

Section 3.2 treats the facts from Section 3.1 as simple toy examples of pointwise computation of (co)limits in a functor category.

3.1 Limits and colimits as intersections and unions

We call a category *bicomplete* if it is both complete and cocomplete.

Recall that limits and colimits in posets are infima and suprema. So we already understand them completely in $\mathbf{2}$, X , and $P(X)$. In $\mathbf{2}$, limits are conjunctions and colimits are disjunctions. In X , only the trivial limits and colimits exist. In $P(X)$, limits are intersections and colimits are unions. Let us record this.

Proposition 3.1.1 (Limits and colimits in $P(X)$ are intersections and unions). For a set X , limits in $P(X)$ are intersections, and colimits are unions.

So we already know all about limits and colimits in the categories we care about. What is this section for, then? We want to see this phenomenon in the context of general category theory. That is, after all, the point of looking at set operations as toy examples!

In this section and several later ones, notice how $\mathbf{2}$ plays a role analogous to $\mathbf{Set}$ in ordinary mathematics.¹⁵

Proposition 3.1.2 ($\mathbf{2}$ is bicomplete). The truth-value category $\mathbf{2}$ is bicomplete.

Of course, its limits and colimits are extremely simple, as we just noted. For later use, let us record the corresponding property of $\mathbf{Set}$ too.

Remark 3.1.3 ($\mathbf{Set}$ is bicomplete). The category $\mathbf{Set}$ is bicomplete. Moreover, its limits and colimits can be computed explicitly, for example as (co)equalizers of (co)products (an informal claim).

3.2 Limits and colimits in functor categories

Limits and colimits in $\mathcal{C}^{\mathcal{D}}$ are determined by those in $\mathcal{C}$. Explicitly, the functor $\mathcal{C}^{\mathcal{D}} \rightarrow \mathcal{C}^{\mathbf{ob}(\mathcal{D})}$ that forgets the action on morphisms—precomposition with $\mathbf{ob}(\mathcal{D}) \rightarrow \mathcal{D}$ —strictly creates all (co)limits present in $\mathcal{C}^{\mathbf{ob}(\mathcal{D})}$.¹⁶ In particular, if $\mathcal{C}$ is bicomplete, so is $\mathcal{C}^{\mathcal{D}}$. Informally, creation also means that if we can compute (co)limits in $\mathcal{C}$, we can compute them in the functor category “object by object.” See, for example, Proposition 3.3.9 of [1].

Together with Theorem 2.3.4, this general theory gives the following fact, which we already knew.

¹⁵Perhaps this is where I ought to mention enriched categories, but I do not yet know enough about them.

¹⁶Watch out for size issues.

Proposition 3.2.1 ($P(X)$ is bicomplete). For a set X , $P(X)$ is bicomplete. Its limits are pointwise conjunctions (intersections), and its colimits are pointwise disjunctions (unions).

We already knew bicompleteness from Proposition 3.1.1. You may think this is overkill for something so simple—and you would be right! But the point is to view intersection as pointwise conjunction and union as pointwise disjunction. This is an utterly simple fact, apparent from $x \in A \cap B \iff x \in A \wedge x \in B$. Yet it is a toy example of the important theorem that limits and colimits in functor categories are computed objectwise. Let us look at analogous phenomena elsewhere in mathematics.

Remark 3.2.2 (Functor-category limits and colimits in ordinary mathematics). Since $\mathbf{Set}$ is bicomplete, so are categories of set-valued functors, with limits and colimits computed objectwise. Here are some examples.

- The category $G\text{-}\mathbf{Set}$ of sets with a left action of a group G is equivalent to $\mathbf{Set}^G$, viewing G as a one-object category. Thus it is bicomplete, with limits and colimits computed on underlying sets.
- For a topological space X , its presheaf category $\mathbf{PSh}_X$ is equivalent to $\mathbf{Set}^{\mathcal{O}(X)^{\text{op}}}$, where $\mathcal{O}(X)$ is the category of open subsets ordered by inclusion. Thus it is bicomplete, with limits and colimits computed set by open set.
- The category $\mathbf{DirGraph}$ of directed graphs is equivalent to $\mathbf{Set}^{\mathbf{Parallel}}$, where $\mathbf{Parallel}$ has objects $\{\text{Edge}, \text{Vertex}\}$ and, besides identities, two parallel arrows $\text{source}, \text{target} : E \rightarrow V$. Thus it is bicomplete, with limits and colimits computed separately on vertices and edges.

There are many more interesting examples: simplicial sets, discrete dynamical systems on sets (sets with an endomap), and so on.

Moving beyond $\mathbf{Set}$, the representation category $\mathbf{Vect}_{\mathbb{K}}^G$ of a group is another typical functor category. Its limits and colimits are computed on vector spaces. This explains why so many universal constructions on vector spaces lift directly to representations.

Knowing the limits and colimits of a functor category is useful, and we will use this fact again.

4 Adjunctions

Here we explore some adjunctions among set operations. Only some: Section 6 will discuss a few more simple ones.

Section 4 falls roughly into two halves. The first runs from Section 4.1 to Section 4.3; the second from Section 4.4 to Section 4.7.

The first half focuses on the two adjoints of inverse image. Every adjunction immediately gives us

- preservation of limits by the right adjoint,
- preservation of colimits by the left adjoint,
- a unit (an inequality, in our setting!),
- a counit.

Extracting several rules for set operations from one adjunction is very much in the spirit of this text.

Section 4.1 introduces the familiar adjoints of inverse image. If you know Kan extensions, their existence was already clear in Section 2.4. We return to this in Section 7.2.

Section 4.2 views these adjoints as the quantifiers $\forall, \exists$. Using the subset–predicate correspondence from Section 2.3, we move from operations on sets to operations on predicates. The right adjoint of inverse image may look unfamiliar at first, but it is just universal quantification. We also see that quantifiers of the same kind commute because they are adjoints of a completely obvious commutative diagram.

Section 4.3 returns to quantifiers as adjoints from Section 4.2. Continuing Section 4.2, Taking adjoints of the obvious commutative diagram from Section 2.5 shows why moving a quantifier past negation changes its kind. We also explain that “left adjoints preserve colimits” has formally the very same proof, assuming all colimits of the shape in question exist.

The second half treats classes of adjunctions rather than individual ones. Specifically, these are

- mutually right adjoints (contravariant adjunctions) between two power sets,
- reflective subcategories of a power set.

These classes are related; Section 4.7 explores the relationship.

Section 4.4 shows, by working with elements, that mutually right adjoints between $P(X)$ and $P(Y)$ correspond bijectively to binary relations between X and Y . There are as many examples as binary relations; we give a few. They immediately provide examples for Sections 4.5, 4.6, and 4.7. The categorical explanation and connections with other categorical phenomena come in the section on Kan extensions, especially Section 7.4; Remark 7.4.6 summarizes them.

Section 4.5 applies the fact that fixed points of an adjunction yield an equivalence to the mutually right adjoints constructed in Section 4.4. Hilbert’s Nullstellensatz and the Galois correspondence are typical examples. Naturally, the set operations alone do not prove either the Nullstellensatz or the fundamental theorem of Galois theory! I have tried to explain what we do get, what we do not, and why it is useful.

Section 4.6 identifies reflective subcategories of a functor category with “generation” of subsets.¹⁷ Reflective subcategories are closely related to monads, especially for posets, leading to Section 5, notably Sections 5.1, 5.2, and 5.3.

Section 4.7 applies the general factorization of an idempotent adjunction into reflective subcategories and an equivalence to the mutually right adjoints constructed in Section 4.4. This relates the two classes considered in the second half of Section 4 and leads to Section 5.3.

4.1 Adjoints of inverse image and subsets

By Proposition 2.4.1, precomposition with a functor (map) $f : X \rightarrow Y$ gives the inverse-image functor $f^* : \mathbf{2}^Y \rightarrow \mathbf{2}^X$. Let us find its left and right adjoints.¹⁸

For $U \subset X$ and $V \subset Y$,

$$U \subset f^{-1}(V) \iff f(U) \subset V$$

holds. Recalling the definition of an adjunction, this says that the direct-image map f_* is left adjoint to f^{-1} .

Remark 4.1.1 (Galois connection). An adjunction between posets is called a Galois connection. In the hom-set definition of an adjunction, naturality is automatic here because every hom-set has at most one element. To check a Galois connection, we need only check an equivalence between inequalities.

The right adjoint of f^{-1} is less familiar than direct image. We will call it the dual-image map.

Definition 4.1.2 (Dual image). For a map $f : X \rightarrow Y$, define $f_! : P(X) \rightarrow P(Y)$ by $f_!(U) = \{y \in Y \mid f^{-1}(y) \subset U\}$. We call this the dual-image map of f .¹⁹

¹⁷This overlaps with my earlier [6]. Together with the monad section, the text you are reading almost entirely supersedes [6], apart from details such as the somewhat wider range of examples there.

¹⁸Later, pointwise Kan extensions will give both their existence and their explicit form abstractly!

¹⁹This is the term used on the nLab: <https://ncatlab.org/nlab/show/internal+logic>.

Taking complements is a contravariant automorphism of $P(X)$, and commutes with inverse image. Consequently, dual image is the complement of the direct image of the complement. Later we will obtain the existence and nature of both adjoints from the fact that inverse image is precomposition between “nice functor categories.” For now, let us prove it by elementary set operations.

Proposition 4.1.3 (Adjoint of inverse image). For $f : X \rightarrow Y$, direct image f_* and dual image $f_!$ are, respectively, left and right adjoint to $f^{-1} : P(Y) \rightarrow P(X)$.

Proof. We prove only the dual-image assertion. For $U \subset X$ and $V \subset Y$,

$$f^{-1}(V) \subset U \iff [\forall x \in X, \forall y \in V (f(x) = y \Rightarrow x \in U)] \iff V \subset f_!(U)$$

as required. □

In a diagram,

$$\begin{array}{ccc} & f_* & \\ \swarrow & & \searrow \\ P(Y) & \xrightarrow{f^{-1} \perp} & P(X) \\ \nwarrow & & \nearrow \\ & f_! & \end{array}$$

as above.²⁰

The famous theorems say that left adjoints preserve colimits (LAPC) and right adjoints preserve limits (RAPL). They are so famous that I hesitate to give ordinary mathematical examples now. Perhaps, if the mood strikes, I will write dozens of pages of LAPC examples.²¹ Inverse image is both a left and a right adjoint, and direct image is a left adjoint. Proposition 3.1.1 therefore gives:

Corollary 4.1.4 (LAPC and RAPL for direct and inverse image). For $f : X \rightarrow Y$, direct image preserves unions, while inverse image preserves both unions and intersections.

$$\begin{aligned} f \left(\bigcup_{\lambda \in \Lambda} V_\lambda \right) &= \bigcup_{\lambda \in \Lambda} f(V_\lambda) \\ f^{-1} \left(\bigcup_{\lambda \in \Lambda} U_\lambda \right) &= \bigcup_{\lambda \in \Lambda} f^{-1}(U_\lambda) \\ f^{-1} \left(\bigcap_{\lambda \in \Lambda} U_\lambda \right) &= \bigcap_{\lambda \in \Lambda} f^{-1}(U_\lambda) \end{aligned}$$

Direct image need not preserve intersections. The fact that inverse image has adjoints on both sides is often offered as an answer to “Why does inverse image behave better than direct image?”

Adjunctions can also be defined using units and counits. Every adjunction comes with these natural transformations; since morphisms in posets are inequalities, we obtain:

Corollary 4.1.5 (The unit and counit for direct and inverse image). For $f : X \rightarrow Y$ and any $U \subset X$, $V \subset Y$,

$$\begin{aligned} U &\subset f^{-1}(f(U)) \\ f(f^{-1}(V)) &\subset V \end{aligned}$$

hold.

²⁰If you know a good way to typeset adjunctions in T_EX, please tell me on Twitter at @hora_algebra.

²¹Probably not.

Proof. The upper inclusion is a component of the unit, and the lower one a component of the counit. $\square$

Since units and counits give another definition of adjunction, these inequalities can also be used to define a Galois connection.

The map f is surjective exactly when the counit is an equality. We will give a slightly stronger result in the cartesian closed category section.²²

4.2 Adjoints of inverse image and quantifiers

So far, we have spoken about subsets. Recall $\mathbf{2}^X \cong P(X)$, and ask what happens on the predicate side. The somewhat mysterious dual image is universal quantification; direct image is existential quantification.

For sets X, Y , consider the adjunctions induced by the second projection $p : X \times Y \rightarrow Y$:

$$\begin{array}{ccc} & p_* & \\ & \curvearrowright & \\ 2^Y & \xrightarrow{p^*} & 2^{X \times Y} \\ & \curvearrowleft & \\ & p_! & \end{array}$$

as above.²³ Notice that I have changed only p^{-1} to p^* , just for the feel of it. The functor p^* introduces a dummy variable.

By the definition of direct image, for $U \subset X \times Y$ and $y \in Y$,

$$\begin{aligned} y \in p(U) &\iff \exists z \in U, y = p(z) \\ &\iff \exists x \in X, (x, y) \in U \end{aligned}$$

holds. In terms of predicates, for $\phi : X \times Y \rightarrow \mathbf{2}$ and $y \in Y$,

$$p_*(\phi)(y) = \text{true} \iff \exists x \in X, \phi(x, y) = \text{true}$$

holds. Dually, for $\phi : X \times Y \rightarrow \mathbf{2}$ and $y \in Y$,

$$p_!(\phi)(y) = \text{true} \iff \forall x \in X, \phi(x, y) = \text{true}$$

holds. Thus, on predicates, p_* acts as existential quantification and $p_!$ as universal quantification. Let us introduce notation for this.

Definition 4.2.1 (Quantifiers on predicates). For sets X, Y , write $\exists_X$ for the direct-image map of the second projection $p : X \times Y \rightarrow Y$, and $\forall_X$ for its dual-image map.²⁴

This view of quantifiers as adjoints is described on the nLab at https://ncatlab.org/nlab/show/quantification#lawvere_quantifiers_quantification_as_adjunction, and in Section I.9, “Quantifiers as Adjoint,” of [4].

What do the properties of adjunctions tell us about these quantifiers?

First, LAPC and RAPL become the following.

Proposition 4.2.2 (Quantifiers commute with disjunction or conjunction). For $\phi, \psi : X \times Y \rightarrow \mathbf{2}$, we have $\exists_X(\phi \vee \psi) = (\exists_X \phi) \vee (\exists_X \psi)$ and, dually, $\forall_X(\phi \wedge \psi) = (\forall_X \phi) \wedge (\forall_X \psi)$.

The unit and counit give these implications:

²²More generally, the unit or counit is an isomorphism exactly when the corresponding left or right adjoint is fully faithful. The inclusions in Corollary 4.1.5 become equalities under injectivity or surjectivity conditions. I intend to add a suitably general account if I find a good way to write it.

²³Now urgently seeking advice on how to typeset adjunctions nicely in T_EX!

²⁴Strictly speaking, we should indicate Y too, but we suppress it.

Proposition 4.2.3 (Units and counits of quantifiers). For sets X, Y and a predicate $\phi : X \times Y \rightarrow \mathbf{2}$,

$$\phi \Rightarrow \exists_X \phi$$

$$\forall_X \phi \Rightarrow \phi$$

hold. Once the quantifier has been applied, the X -variable is a dummy variable.

Proof. Remember that adjoining a dummy variable is p^* . The upper implication is the unit of the existential adjunction; the lower one is the counit of the universal adjunction. $\square$

If X is empty, these statements are vacuous. The counit for existential quantification and the unit for universal quantification first adjoin a dummy variable and then remove it with a quantifier.

Adjoints are unique up to natural isomorphism, so a commutative diagram of right adjoints gives a diagram of left adjoints commuting up to natural isomorphism. For our posets, it commutes strictly. This gives the following quantifier rules. We will comment on the argument in the next subsection.

Proposition 4.2.4 (Commuting quantifiers). For sets X, Y, Z and a predicate $\phi : X \times Y \times Z \rightarrow \mathbf{2}$,

$$\exists_X \exists_Y \phi = \exists_Y \exists_X \phi$$

$$\forall_X \forall_Y \phi = \forall_Y \forall_X \phi$$

hold.

Proof. The two ways of projecting

$$\begin{array}{ccc} X \times Y \times Z & \longrightarrow & X \times Z \\ \downarrow & & \downarrow \\ Y \times Z & \longrightarrow & Z \end{array}$$

commute. Therefore, their precomposition functors

$$\begin{array}{ccc} \mathbf{2}^{X \times Y \times Z} & \longleftarrow & \mathbf{2}^{X \times Z} \\ \uparrow & & \uparrow \\ \mathbf{2}^{Y \times Z} & \longleftarrow & \mathbf{2}^Z \end{array}$$

commute too. Hence their left adjoints

$$\begin{array}{ccc} \mathbf{2}^{X \times Y \times Z} & \xrightarrow{\exists_Y} & \mathbf{2}^{X \times Y} \\ \downarrow \exists_X & & \downarrow \exists_X \\ \mathbf{2}^{Y \times Z} & \xrightarrow{\exists_Y} & \mathbf{2}^Z \end{array}$$

and their right adjoints

$$\begin{array}{ccc} \mathbf{2}^{X \times Y \times Z} & \xrightarrow{\forall_Y} & \mathbf{2}^{X \times Y} \\ \downarrow \forall_X & & \downarrow \forall_X \\ \mathbf{2}^{Y \times Z} & \xrightarrow{\forall_Y} & \mathbf{2}^Z \end{array}$$

commute up to natural isomorphism. Since these are posets, they commute strictly. $\square$

4.3 Adjoints of negation and exchanging negation with quantifiers

Another familiar quantifier rule concerns negation. First consider its adjoints. Since negation is a contravariant automorphism, it is adjoint to itself. LAPC gives De Morgan's laws.

Proposition 4.3.1 (Adjoints of negation and De Morgan's laws). Negation, or complementation $(-)^c : P(X)^{\text{op}} \rightarrow P(X)$, is a contravariant automorphism, hence both left and right adjoint to itself. Thus $(U \cap V)^c = U^c \cup V^c$ and $(U \cup V)^c = U^c \cap V^c$. Equivalently, for predicates $\phi, \psi : X \rightarrow \mathbf{2}$, we have $\neg(\phi \wedge \psi) = \neg\psi \vee \neg\phi$ and $\neg(\phi \vee \psi) = \neg\psi \wedge \neg\phi$.

If all we wanted were the complement–intersection–union rules, a contravariant equivalence would already suffice. I emphasize adjunctions because we want to relate negation to quantifiers.

In general, if a diagram of functors commutes and every functor has a left adjoint, the left adjoints commute up to natural isomorphism. For posets, they commute strictly.

Proposition 4.3.2 (Exchanging quantifiers and negation). For sets X, Y and a predicate ϕ on $X \times Y$, we have $\neg\exists_X\phi = \forall_X\neg\phi$ and $\neg\forall_X\phi = \exists_X\neg\phi$.

Proof. Take the commutative diagram from Proposition 2.5.5 for the second projection $p : X \times Y \rightarrow Y$:

$$\begin{array}{ccc} (\mathbf{2}^X)^{\text{op}} & \xrightarrow{\neg_*} & \mathbf{2}^X \\ \downarrow (p^*)^{\text{op}} & & \downarrow p^* \\ (\mathbf{2}^{X \times Y})^{\text{op}} & \xrightarrow{\neg_*} & (\mathbf{2}^{X \times Y}) \end{array}$$

Passing to left adjoints gives

$$\begin{array}{ccc} (\mathbf{2}^X)^{\text{op}} & \xleftarrow{\neg_*} & \mathbf{2}^X \\ \uparrow (\forall_X)^{\text{op}} & & \uparrow \exists_X \\ (\mathbf{2}^{X \times Y})^{\text{op}} & \xleftarrow{\neg_*} & (\mathbf{2}^{X \times Y}) \end{array}$$

and passing to right adjoints gives

$$\begin{array}{ccc} (\mathbf{2}^X)^{\text{op}} & \xleftarrow{\neg_*} & \mathbf{2}^X \\ \uparrow (\exists_X)^{\text{op}} & & \uparrow \forall_X \\ (\mathbf{2}^{X \times Y})^{\text{op}} & \xleftarrow{\neg_*} & (\mathbf{2}^{X \times Y}) \end{array}$$

□

Remark 4.3.3 (Commuting free constructions). This is a common categorical phenomenon: to show that a diagram of functors commutes up to natural isomorphism, passing to adjoints may make the problem easier. Free constructions between categories of algebras are typical examples. Free groups, polynomial rings, free monoids, abelianization, extension of scalars, tensor algebras, and so on commute because their right adjoints are forgetful functors, whose commutation is obvious.

Remark 4.3.4 (The analogy with LAPC). LAPC is even more directly related. To put it provocatively, it works by the same mechanism as exchanging negation and quantifiers. Assuming all colimits of the relevant shape exist, both statements follow by passing the commutation of precomposition and postcomposition across adjunctions.

Recall first that, when all (co)limits of a fixed shape exist, they are adjoint to the diagonal functor. For a small category J and a locally small category $\mathcal{C}$, consider precomposition $!^* : \mathcal{C}^{\mathbf{1}} \cong \mathcal{C} \rightarrow \mathcal{C}^J$ with the unique functor $! : J \rightarrow \mathbf{1}$. Thus the diagonal functor is precomposition. It has a left adjoint $\text{colim}_J : \mathcal{C}^J \rightarrow \mathcal{C}$ exactly when $\mathcal{C}$ has all J -indexed colimits.

Suppose $F : \mathcal{C} \rightarrow \mathcal{D}$ is left adjoint to $G : \mathcal{D} \rightarrow \mathcal{C}$, and both categories have J -indexed colimits. Associativity for G and $! : J \rightarrow \mathbf{1}$ gives the commutative diagram

$$\begin{array}{ccc} \mathcal{D}^{\mathbf{1}} & \xrightarrow{G_*} & \mathcal{C}^{\mathbf{1}} \\ \downarrow !^* & & \downarrow !^* \\ \mathcal{D}^J & \xrightarrow{G_*} & \mathcal{C}^J \end{array}$$

whose left adjoints form the diagram

$$\begin{array}{ccc} \mathcal{D}^{\mathbf{1}} & \xleftarrow{F_*} & \mathcal{C}^{\mathbf{1}} \\ \uparrow \text{colim}_J & & \uparrow \text{colim}_J \\ \mathcal{D}^J & \xleftarrow{F_*} & \mathcal{C}^J \end{array}$$

commuting up to natural isomorphism. This says that left adjoints preserve colimits. We used the fact that an adjunction induces an adjunction between functor categories. RAPL follows dually.

Remark 4.3.5 (The unit and counit for negation). The inequalities supplied by negation's unit and counit are $U \subset U^{cc}$ for subsets and $\phi \rightarrow \neg\neg\phi$ for predicates. Here they are isomorphisms, of course, but their direction has several interpretations. We will say more in the cartesian closed category section; for now, consider a Heyting algebra H . Negation gives mutually right adjoints from H to itself, whose unit is $\phi \rightarrow \neg\neg\phi$ for each $\phi \in H$. Double-negation elimination need not hold in a Heyting algebra, so this arrow need not be an isomorphism (here, an equality). We discuss the condition for it to be one in the Galois correspondence subsection.

4.4 Mutually right adjoints and Galois connections from binary relations, I

Mutually right adjoints between categories $\mathcal{C}, \mathcal{D}$ consist of contravariant functors $F : \mathcal{C}^{\text{op}} \rightarrow \mathcal{D}$, $G : \mathcal{D}^{\text{op}} \rightarrow \mathcal{C}$, with natural isomorphisms $\mathcal{C}(c, Gd) \cong \mathcal{D}(d, Fc)$. Naturality means the naturality for an adjunction between $\mathcal{C}^{\text{op}}$ and $\mathcal{D}$. See, for example, Definition 4.3.1 of [1].

For us, everything is a poset, so naturality is automatic.

Remark 4.4.1 (antitone Galois connection). Mutually right adjoints between posets are called an *antitone Galois connection*, and have a simple description: order-reversing maps $f : P \rightarrow Q$, $g : Q \rightarrow P$ with $p \leq g(q) \iff q \leq f(p)$ for all $p \in P$, $q \in Q$.

Here we study mutually right adjoints between power sets $P(X), P(Y)$. We will carry one typical example alongside the general discussion and give others later.

Example 4.4.2 (An antitone Galois connection in Galois theory). For a finite Galois extension $K \subset L$ with group $\text{Gal}(L/K)$, the order-reversing maps between $P(L)$ and $P(\text{Gal}(L/K))$

$$R^L : P(L)^{\text{op}} \rightarrow P(\text{Gal}(L/K)) : L \supset U \mapsto \{\sigma \in \text{Gal}(L/K) \mid \forall x \in U, \sigma(x) = x\} \subset \text{Gal}(L/K)$$

$$\text{Gal}(L/K) R : P(\text{Gal}(L/K))^{\text{op}} \rightarrow P(L) : \text{Gal}(L/K) \supset V \mapsto \{x \in L \mid \forall \sigma \in V, \sigma(x) = x\} \subset L$$

are mutually right adjoint.

This construction has a general form. In the section on Kan extensions, especially Theorem 7.4.5, we will view it as a universal adjunction. For now, let us write it out elementwise.

Proposition 4.4.3 (Mutually right adjoints from a binary relation). For sets X, Y and a binary relation $R : X \times Y \rightarrow \mathbf{2}$,²⁵ the maps

$$R^Y : P(Y)^{\text{op}} \rightarrow P(X) : Y \supset U \mapsto \{x \in X \mid \forall y \in U, R(x, y) = \text{true}\} \subset X$$

$${}^X R : P(X)^{\text{op}} \rightarrow P(Y) : X \supset V \mapsto \{y \in Y \mid \forall x \in V, R(x, y) = \text{true}\} \subset Y$$

are mutually right adjoint.

²⁵It makes no difference yet, but when we describe this as a universal adjunction, we will see that we should have viewed it as $R : X \times Y \rightarrow \mathbf{2}^{\text{op}}$.

Proof. For $V \subset X$ and $U \subset Y$,

$$\begin{aligned} V \subset R^Y(U) &\iff \forall (x, y) \in V \times UR(x, y) = \text{true} \\ &\iff U \subset {}^X R(V) \end{aligned}$$

as required. $\square$

In Galois theory, take $R(\sigma, x)$ to mean $\sigma(x) = x$. Here are some more examples.

Example 4.4.4 (Hilbert’s Nullstellensatz). For an algebraically closed field $\mathbb{K}$ and a positive integer n , take $X = \mathbb{K}[x_1, \dots, x_n]$, $Y = \mathbb{K}^n$, and define $R : X \times Y \rightarrow \mathbf{2}$ by $R(f, (a_1, \dots, a_n)) = \text{true} \iff f(a_1, \dots, a_n) = 0$. Then ${}^X R$ takes common zero sets, and R^Y takes all polynomials vanishing on a given set. We will connect this to the Nullstellensatz itself in the next subsection.

Two notions of orthogonality (<https://ncatlab.org/nlab/show/orthogonality>) give further examples.

Example 4.4.5 (Orthogonality in a vector space). For an inner product space V , the mutually right adjoints induced by orthogonality $\perp$ take orthogonal complements.

Example 4.4.6 (Orthogonality in a category). Morphisms $f : A \rightarrow B$ and $g : C \rightarrow D$ in a category $\mathcal{C}$ are orthogonal, written $f \perp g$, if every commutative square

$$\begin{array}{ccc} A & \longrightarrow & C \\ \downarrow f & & \downarrow g \\ B & \longrightarrow & D \end{array}$$

has a unique morphism $h : B \rightarrow C$

$$\begin{array}{ccc} A & \longrightarrow & C \\ \downarrow f & \exists! h \nearrow & \downarrow g \\ B & \longrightarrow & D \end{array}$$

making both triangles commute. For a (small) category $\mathcal{C}$, take $X = Y = \text{Mor}(\mathcal{C})$ and the relation $\perp$. The induced mutually right adjoints collect, respectively, the morphisms left orthogonal and right orthogonal to a given collection.

This notion appears in orthogonal factorization systems. In Set , or more generally any topos, every epimorphism e is orthogonal to every monomorphism m : $e \perp m$. Indeed, epis and monos form an orthogonal factorization system (<https://ncatlab.org/nlab/show/orthogonal+factorization+system>).

Finally, an example from logic. Our reference is [5]; it also appears as Example 4.3.3 of [1].

Example 4.4.7 (The Galois Connection between Syntax and Semantics). For a first-order language $\mathcal{L}$, consider satisfaction $\models : \text{Structure}_{\mathcal{L}} \times \text{Sentence}_{\mathcal{L}} \rightarrow \mathbf{2}$, between the **class**²⁶ of $\mathcal{L}$ -structures and the set of $\mathcal{L}$ -sentences. The mutually right adjoint **assignments** take an axiom system to **all its models**, and structures to all the sentences they satisfy.

Mutually right adjoints are adjoints, so LAPC applies.

Proposition 4.4.8 (LAPC for an antitone Galois connection). For sets X, Y , mutually right adjoints $f : P(X)^{\text{op}} \rightarrow P(Y)$ and $g : P(Y)^{\text{op}} \rightarrow P(X)$ send unions to intersections.

$$f \left(\bigcup_{\lambda \in \Lambda} V_{\lambda} \right) = \bigcap_{\lambda \in \Lambda} f(V_{\lambda})$$

²⁶Red text marks size issues whose resolution I do not understand as of January 2021. Section 3.6, “Posets as sets,” of [5] discusses a resolution.

$$g \left(\bigcup_{\lambda \in \Lambda} U_\lambda \right) = \bigcap_{\lambda \in \Lambda} g(U_\lambda)$$

Proposition 4.4.9 (Units of mutually right adjoints). For mutually right adjoints $f : P(X)^{\text{op}} \rightarrow P(Y)$, $g : P(Y)^{\text{op}} \rightarrow P(X)$ and any $V \subset X$, $U \subset Y$,

$$V \subset g(f(V))$$

$$U \subset f(g(U))$$

hold.

As for any adjunction, gf and fg come with natural multiplications making them induced monads. We will discuss this in the monad section.

So far we have obtained mutually right adjoints from binary relations. In fact, LAPC shows that every pair of mutually right adjoints between power sets comes from a binary relation. Here LAPC is a technical tool for the elementwise proof; a more conceptual explanation via universal adjunctions will come in Remark 7.4.6, after Kan extensions.

Theorem 4.4.10 (Binary relations correspond to mutually right adjoints). For sets X, Y , assigning to $R : X \times Y \rightarrow \mathbf{2}$ its mutually right adjoints ${}^X R, R^Y$ gives a bijection between binary relations on X, Y and mutually right adjoints between $P(X), P(Y)$.

Proof. Let us construct the inverse. Given mutually right adjoints $f : P(X)^{\text{op}} \rightarrow P(Y)$, $g : P(Y)^{\text{op}} \rightarrow P(X)$, define $R_{f,g} : X \times Y \rightarrow \mathbf{2}$ by

$$R_{f,g}(x, y) = \text{true} \iff x \in g(\{y\}) \iff y \in f(\{x\})$$

Proposition 4.4.8 shows that this is the inverse construction. □

In this sense, binary relations and mutually right adjoints between power sets are two sides of the same coin. What relation corresponds to an adjunction we already know?

Example 4.4.11 (The relation inducing the negation adjunction). Negation on a power set is contravariantly right adjoint to itself. Its corresponding relation is the negation of equality,²⁷ $\neq : X \times X \rightarrow \mathbf{2}$.

Let us now examine some structures obtained from mutually right adjoints.

4.5 Equivalences induced by adjunctions and the Galois correspondence

We have generalized the correspondence between subsets of a Galois group and subsets of a field using binary relations and mutually right adjoints. But what corresponds to the Galois correspondence itself, between subgroups and intermediate fields? It is a toy example of the equivalence induced by any adjunction.

An adjunction induces an equivalence between the full subcategories of objects where its unit and counit are isomorphisms: its fixed points! See <https://ncatlab.org/nlab/show/fixed+point+of+an+adjunction>, or Exercise 4.2.i of [1].

For an example elsewhere in mathematics, consider the equivalence of two definitions of a sheaf. We will return to this analogy later.

Remark 4.5.1 (Two definitions of a sheaf). For a topological space X , consider Top/X and the presheaf category $\text{Set}^{\mathcal{O}(X)^{\text{op}}}$. Taking sections gives $\Gamma : \text{Top}/X \rightarrow \text{Set}^{\mathcal{O}(X)^{\text{op}}}$, whose left adjoint $L : \text{Set}^{\mathcal{O}(X)^{\text{op}}} \rightarrow \text{Top}/X$ takes the étale space of a presheaf. Writing out the condition that the

²⁷The word “inequality” has already been claimed by another notion.

counit be an isomorphism gives the local-homeomorphism condition; doing the same for the unit gives the sheaf gluing condition.

Thus the two definitions of sheaf are “derived” by spelling out the fixed-point condition, and general theory guarantees the equivalence of the resulting categories. In the Kan extension section, we will explain how this adjunction, like those induced by binary relations, arises naturally in a general framework.²⁸

Another part of the analogy is that this adjunction is idempotent: <https://ncatlab.org/nlab/show/idempotent+adjunction>.

We want the fixed points of mutually right adjoints between power sets. They follow immediately from the general fact that a Galois connection is an idempotent adjunction (<https://ncatlab.org/nlab/show/idempotent+adjunction>), but let us work this out in our case. First, a definition that will apparently be generalized in the monad section.

Definition 4.5.2 (Closed subsets). For a relation $R : X \times Y \rightarrow \mathbf{2}$, a subset of X is *closed with respect to R* if it belongs to the image of $R^Y : P(Y)^{\text{op}} \rightarrow P(X)$. Similarly, a subset of Y is closed if it belongs to the image of ${}^X R : P(X)^{\text{op}} \rightarrow P(Y)$.²⁹ Write $\text{Closed}_{X,R}$ and $\text{Closed}_{R,Y}$ for the respective sets of closed subsets.

Example 4.5.3 (Closed subsets in Galois theory). For a finite Galois extension $K \subset L$, the closed subsets for its mutually right adjoints are the intermediate fields and the subgroups of $\text{Gal}(L/K)$, respectively.

We discuss facts valid for every relation R . Naturally, this does not give a complete method for identifying the closed subsets for a particular relation, as in this example. Some later theorems do give general techniques and constraints, though.

Lemma 4.5.4 (Closed subsets are fixed points). For a relation $R : X \times Y \rightarrow \mathbf{2}$, a subset $V \subset X$ is closed if and only if it is a fixed point:

$$R^Y({}^X R(V)) = V$$

Proof. A fixed point is closed immediately, since it is the image of ${}^X R(V)$. Conversely, if V is closed, write $V = R^Y(U)$ for some $U \subset Y$. By Proposition 4.4.9, we have $U \subset {}^X R R^Y(U)$ and $V \subset R^Y {}^X R(V)$. Apply the order-reversing map R^Y to the first inclusion to get $R^Y(U) \supset R^Y {}^X R R^Y(U)$, that is, $V \supset R^Y {}^X R(V)$. $\square$

This proof is using the triangle identities for the unit and counit. It generalizes easily to idempotent adjunctions, for example the adjunction between sheafification and sections.

Remark 4.5.5 (Fixed points and images for sheaves). A presheaf is a sheaf if and only if it lies in the essential image of the section functor Γ . A bundle is an étale bundle if and only if it lies in the essential image of the functor L taking étale spaces of presheaves.

Interpreted in a Heyting algebra, this phenomenon gives a necessary and sufficient condition for double-negation elimination in intuitionistic logic.

Remark 4.5.6 (Heyting algebra). For a Heyting algebra H , Remark 4.3.5 gives an arrow $\phi \rightarrow \neg\neg\phi$. Since fixed points are precisely the image, this is an isomorphism exactly when $\phi = \neg\psi$ for some ψ . Here the underlying fact is that triple negation reduces to single negation: $\neg\psi \leftrightarrow \neg\neg\neg\psi$.

Now for the main point. The equivalence between fixed points takes the following form. We follow <https://ncatlab.org/nlab/show/Galois+connection#Definition> in calling it a Galois correspondence.

²⁸Called nerve and realization (<https://ncatlab.org/nlab/show/nerve+and+realization>), or a universal adjunction.

²⁹The monad section will explain the word “closed.”

Theorem 4.5.7 (Galois correspondence). For a relation $R : X \times Y \rightarrow \mathbf{2}$, the maps ${}^X R$ and R^Y give an order-reversing isomorphism between $\text{Closed}_{X,R}$ and $\text{Closed}_{R,Y}$.

We have already mentioned the classical Galois correspondence as a typical example, and the two definitions of a sheaf as an analogous phenomenon in ordinary mathematics.

Example 4.5.8 (Hilbert’s Nullstellensatz). As before, take an algebraically closed field $\mathbb{K}$, a positive integer n , $X = \mathbb{K}[x_1, \dots, x_n]$, $Y = \mathbb{K}^n$, and the relation $R(f, (a_1, \dots, a_n)) = \text{true} \iff f(a_1, \dots, a_n) = 0$. The closed subsets are, respectively, radical ideals and algebraic sets.

There are as many further examples as binary relations, including complementation and orthogonal complements from our earlier adjunctions.

4.6 Reflective subcategories, the adjoint functor theorem, and generation, I

Much of this subsection overlaps with my earlier [6], “A General Theory of Generation and Reflective Subcategories” (https://hora-algebra.github.io/gen_ref.pdf). The categorical toy-example viewpoint is developed more fully here; [6] has more examples and more informal explanation of generation.

Let us identify the reflective subcategories of $P(X)$. The analogous objects elsewhere in mathematics are reflective subcategories of set-valued functor categories. As Section 2 showed, a functor category valued in a bicomplete category is bicomplete. A reflective subcategory of a bicomplete category is bicomplete too; see Proposition 4.5.15 of [1]. Its inclusion creates limits. Combined with objectwise limits in functor categories, this says that limits in such a reflective subcategory are objectwise. Colimits are obtained by applying the reflector to objectwise colimits. See Proposition 4.5.15 of [1] for these facts.

Example 4.6.1 (Sheaves). The category of sheaves on a topological space X^{30} is a reflective subcategory of $\text{Set}^{\mathcal{O}(X)^{\text{op}}}$, hence bicomplete. Its limits are computed open set by open set, and its colimits are the sheafifications of those computed open set by open set.

Here is another reflective subcategory of a functor category.

Example 4.6.2 (Categories). The category of categories is a reflective subcategory of simplicial sets, hence bicomplete.

Apparently, many other nice categories, such as categories of models of algebraic theories, have this form. I have not yet studied this as of February 2021, but here is a link: <https://ncatlab.org/nlab/show/locally+presentable+category#AsLocalizationsOfPresheafCategories>.

What does this mean for set operations? Full subcategories of $P(X)$ correspond to subsets of $P(X)$, that is, families of subsets of X . We want families Σ whose inclusion $\iota : \Sigma \rightarrow P(X)$ has a left adjoint. Such an adjoint turns an arbitrary subset into a member of the family, so we expect it to behave like generation.

$$\begin{array}{ccc} & \curvearrowright & \\ \Sigma & \xrightarrow[\iota]{} & P(X) \end{array}$$

A functor $U : \mathcal{C} \rightarrow \mathcal{D}$ has a left adjoint if and only if every comma category $d \downarrow U$ has an initial object. An adjunction is a universal arrow for each object! For the inclusion $\iota : \Sigma \rightarrow P(X)$ and $A \in P(X)$, the comma category $A \downarrow \iota$ is the poset of members of Σ containing A , ordered by inclusion. Its initial object is the least member of Σ containing A . This suggests the following definition.

Definition 4.6.3 (A family admitting generation). A family $\Sigma \subset P(X)$ *admits generation* (or is a Moore collection; see <https://ncatlab.org/nlab/show/Moore+closure#InTermsOfClosureCondition>) if, for every $A \subset X$, the set $\{S \in \Sigma \mid A \subset S\}$ has a least element.

³⁰Or, more generally, a Grothendieck topos.

This is exactly the naive idea of generation that we use all the time. It is the condition under which a definition such as “the thingamajig generated by A is the smallest thingamajig containing A ” makes sense.

Algebraic substructures are typical examples.

Example 4.6.4 (Subgroups). For a group X , the family Σ of subgroups admits generation.

Example 4.6.5 (Normal subgroups). For a group X , the family Σ of normal subgroups admits generation.

Example 4.6.6 (Subrings). For a ring X , the family Σ of subrings admits generation.

Example 4.6.7 (Ideals). For a commutative ring X , the family Σ of ideals admits generation.

And so on: algebra alone supplies endless examples. Recalling topological closure gives another.

Example 4.6.8 (Closed subsets). For a topological space X , the family Σ of closed subsets admits generation.

Generating binary relations also fits this framework.

Example 4.6.9 (Equivalence relations). For a set Y , take $X = Y^2$. The family Σ of equivalence relations admits generation.

Example 4.6.10 (Preorders). For a set Y , take $X = Y^2$. The family Σ of preorders admits generation.

We can also generate families of sets.

Example 4.6.11 (σ -algebras). For a set Y , take $X = 2^Y$. The family Σ of σ -algebras on Y admits generation.

Logic gives another example:

Example 4.6.12 (Deductively closed set). For the set $X = \text{Sentence}_{\mathcal{L}}$ of sentences in a first-order language $\mathcal{L}$, the family Σ of deductively closed subsets admits generation.

Any other notion defined as “the smallest such-and-such containing so-and-so” has this form too. See [6] for further examples.

Often, we show that a family is closed under intersections, then generate from A by intersecting all members containing A . This works for every family admitting generation.³¹ Combining it with our definition, chosen to match reflectivity, gives:

Theorem 4.6.13 (Generation, reflectivity, and closure under intersections). For a family Σ of subsets of X , the following are equivalent.

1. Σ admits generation.
2. Σ is a reflective subcategory of $P(X)$: with both viewed as inclusion-ordered categories, the inclusion $\iota : \Sigma \rightarrow P(X)$ has a left adjoint.
3. Σ is closed under intersections: for every $\Sigma' \subset \Sigma$, we have $\bigcap_{S \in \Sigma'} S \in \Sigma$. Taking $\Sigma' = \emptyset$ in particular says $X \in \Sigma$.

Proof. (1) $\iff$ (2) was proved just before the definition. Admitting generation spells out the existence of initial objects in comma categories, that is, universal arrows.

(2) $\Rightarrow$ (3) follows because the inclusion of a reflective subcategory creates limits.

(3) $\Rightarrow$ (2): the inclusion $\iota : \Sigma \rightarrow P(X)$ is then a continuous functor from a complete category. Its domain is small, so the solution set condition is automatic. The general adjoint functor theorem supplies a left adjoint. $\square$

³¹Moreover, this construction is an example of a codensity monad.

This gives a very concise, familiar toy example of basic adjunction theory. First, it illustrates the existence of adjoints: both the criterion using initial objects of comma categories and the adjoint functor theorem itself. I am tempted to say that much of the technical difficulty of the adjoint functor theorem concerns the solution set condition arising from nonsmallness (my personal opinion as of February 2021). Since our categories are small, this is unfortunately not a useful example for getting comfortable with that condition. What it does show is that the familiar method of generating from A by intersecting everything containing A is a simple instance of constructing a left adjoint using limits.

It also illustrates how reflective subcategories of functor categories inherit good properties. Just as for sheaves, a family admitting generation has all small limits and colimits, and we even know their form. Bicompleteness passes from $\mathbf{Set}$ to presheaves $\mathbf{Set}^{\mathcal{O}(X)^{\mathrm{op}}}$ and then to sheaves. In exactly the same way, it passes from the very simple category $\mathbf{2}$ to $P(X)$ and then to a family admitting generation.

Corollary 4.6.14 (Families admitting generation are complete lattices). A family Σ of subsets of X admitting generation is a complete lattice under inclusion. Its limits are intersections, and its colimits are the members of Σ generated by unions.

Proof. This follows from being a reflective subcategory of the bicomplete category $P(X)$. See Proposition 4.5.15 of [1], for example. $\square$

Let us record the converse too.

Remark 4.6.15 (Every complete lattice is a family admitting generation). Every complete lattice is order-isomorphic to a family admitting generation. Thus a poset is a complete lattice if and only if it can be realized as a reflective subcategory of a power set. Indeed, for a complete lattice L with underlying set $|L|$, the map

$$L \rightarrow P(|L|) : l \mapsto \{k \in L \mid k \leq l\}$$

preserves infima, so the adjoint functor theorem gives a left adjoint. It is also easy to show that this embedding is fully faithful. That proves the claim. A more natural perspective might use the continuity of the “Yoneda embedding”³² $L \rightarrow \mathbf{2}^L$, where $\mathbf{2}^L$ is the poset of order-preserving maps $L \rightarrow \mathbf{2}$, not a power set. The analogue of the Yoneda embedding being fully faithful and continuous holds here too.³³ Restricting ourselves to sets rather than general posets is getting in the way. To force the discussion back to set operations, we can pass to $P(X)$, that is, $\mathbf{2}^{|L|}$: check that $\mathbf{2}^L \rightarrow \mathbf{2}^{|L|}$ is fully faithful and continuous. Since $\mathbf{2}^L$ can be viewed as the inclusion-ordered poset of downward-closed subsets of L , the next remark, for instance, immediately shows that this inclusion is reflective.

Remark 4.6.16 (A practical test for admitting generation). Theorem 4.6.13 gave two equivalent conditions. Section 5.2 will add another: informally, the family can be defined by rules saying “if these elements are present, then that element is present too.” Precisely, $\Sigma \subset P(X)$ admits generation if and only if there is a map $d : P(X) \rightarrow P(X)$ such that, for every $U \subset X$,

$$U \in \Sigma \iff \forall S \subset U, d(S) \subset U$$

holds. This is handy. For example, if normal subgroups are defined as subsets closed under the identity, products, inverses, and conjugation, we simply translate the definition into

$$d(S) = \{e\} \cup \{xy \mid x, y \in S\} \cup \{x^{-1} \mid x \in S\} \cup \{gxg^{-1} \mid x \in S, g \in X\}$$

to obtain d . Section 5.2 identifies the construction of the family, or more precisely its closure operator, from d as the construction of a free monad.

³²I hope that learning truth-value enrichment will let me remove those quotation marks.

³³I have not yet learned truth-value enrichment as of February 2021.

4.7 Factoring an idempotent adjunction

Mutually right adjoints from a relation, their Galois correspondence, and reflective subcategories fit together as follows.

Theorem 4.7.1 (Galois connections from relations and reflective subcategories). For a relation $R : X \times Y \rightarrow \mathbf{2}$, the induced mutually right adjoints

$$\begin{array}{ccc} & R^Y & \\ & \curvearrowright & \\ P(X)^{\text{op}} & \perp & P(Y) \\ & \curvearrowleft & \\ & {}^X R & \end{array}$$

are the composite of three adjunctions:

- a reflective subcategory of $P(X)$,

$$\begin{array}{ccc} P(X)^{\text{op}} & \xleftarrow{\quad \iota \quad} & \text{Closed}_{X,R}^{\text{op}} \\ & \searrow \quad \perp \quad \nearrow & \\ & R^Y \circ {}^X R & \end{array}$$

- the Galois correspondence (an isomorphism of categories),

$$\text{Closed}_{X,R}^{\text{op}} \xleftarrow{\quad \cong \quad} \text{Closed}_{R,Y}$$

- a reflective subcategory of $P(Y)$.

$$\begin{array}{ccc} & {}^X R \circ R^Y & \\ & \curvearrowright & \\ \text{Closed}_{R,Y} & \perp & P(Y) \\ & \xrightarrow{\quad \iota \quad} & \end{array}$$

Their composite is

$$\begin{array}{ccccccc} & & & & {}^X R \circ R^Y & & \\ & & & & \curvearrowright & & \\ P(X)^{\text{op}} & \xleftarrow{\quad \iota \quad} & \text{Closed}_{X,R}^{\text{op}} & \xleftarrow{\quad \cong \quad} & \text{Closed}_{R,Y} & \xrightarrow{\quad \iota \quad} & P(Y) \\ & \searrow \quad \perp \quad \nearrow & & & \searrow \quad \perp \quad \nearrow & & \\ & R^Y \circ {}^X R & & & {}^X R \circ R^Y & & \end{array}$$

Proof. Since images are fixed points, ${}^X R \circ R^Y \circ {}^X R = {}^X R$ and $R^Y \circ {}^X R \circ R^Y = R^Y$, giving equality of the functors. We will prove reflectivity of the two categories Closed in the monad section: they are the Eilenberg–Moore categories of the induced monads. $\square$

This phenomenon holds for any idempotent adjunction. See <https://ncatlab.org/nlab/show/idempotent+adjunction#definition>.

Recall that reflective inclusions create limits. Thus infima in $\text{Closed}_{X,R}$ are intersections. Suprema, by the general description of colimits in a reflective subcategory, are generated by unions. Alternatively, the Galois correspondence turns them into infima in the other Closed . The supremum of a family of Galois subgroups is both the subgroup generated by their union and the subgroup corresponding to the intersection of their intermediate fields. Here is another example where factoring an idempotent adjunction tells us about limits and colimits.

Remark 4.7.2 (Limits and colimits of sheaves). The sheaf category Sh_X is a reflective subcategory of the presheaf category $\text{Set}^{\mathcal{O}(X)^{\text{op}}}$. Its limits are therefore presheaf limits, computed in Set open

set by open set; its colimits are sheafifications of presheaf colimits. So far, reflectivity alone explains everything.

idempotent adjunction

$$\begin{array}{ccc} & L & \\ \text{Top}/X & \xleftarrow{\quad} & \text{Set}^{\mathcal{O}(X)^{\text{op}}} \\ & \xrightarrow{\quad \Gamma & \end{array}$$

$\perp$

Now consider its factorization

$$\begin{array}{ccccc} \text{Top}/X & \xleftarrow{\quad \iota & \text{Etale}_X & \xleftarrow{\quad \cong & \text{Sh}_X & \xrightarrow{\quad \iota & \text{Set}^{\mathcal{O}(X)^{\text{op}}} \\ & \searrow \perp & \nearrow L \circ \Gamma & & & \nearrow \Gamma \circ L & \\ & & & & & \perp & \end{array}$$

Here Etale_X is the coreflective subcategory of Top/X consisting of local homeomorphisms. Colimits of sheaves become easier when viewed there: as with any coreflective subcategory, colimits in Etale_X are created from Top/X . As with any slice category, those in Top/X are created from Top . Finally, colimits in Top are set-theoretic colimits equipped with the finest topology making the cocone continuous, so we have computed them. Thus the reflective embedding into presheaves and the coreflective embedding into Top/X tell us many properties of sheaves.

Remark 4.7.3 (Complete Boolean algebras induced by posets). For a symmetric relation $\perp$ on a set X , we have $\text{Closed}_{X,\perp} = \text{Closed}_{\perp,X}$. Write this complete lattice as L . The Galois correspondence gives a contravariant isomorphism $L^{\text{op}} \cong L$.

If X is a poset and $\perp$ means incompatibility (having no common lower bound), this isomorphism serves as negation and makes L a complete Boolean algebra. The closed subsets for this relation are those $S \subset X$ satisfying

- S : Downward Closed
- $\forall x \in X, S:\text{dense below } x \Rightarrow x \in S$

This complete Boolean algebra appears in set-theoretic forcing. For a forcing poset $\mathbb{P}$, the forcing relation gives a Boolean algebra homomorphism into this complete Boolean algebra, from the Boolean algebra of sentences in the forcing language modulo an appropriate equivalence relation.

We have seen that a poset and its incompatibility relation give a Boolean algebra. But whether we obtain a Boolean algebra depends only on the relation $\perp$, not on the order itself. As of April 2021, I do not yet know a simple characterization, purely in terms of a symmetric relation, of when it induces a Boolean algebra.

5 Monads

Section 5.1 revisits generation from Section 4.6 through its operator, called a closure operator. We check that these are precisely monads on power sets, also called Moore closures. The monad–Eilenberg–Moore-category correspondence becomes the correspondence between closure operators and families admitting generation.

Section 5.2 makes precise the intuition that a family admitting generation consists of subsets “closed under certain operations,” and shows how useful this is. We derive it from free monads generated by self-maps, encountering some special adjunctions as toy examples along the way.

Section 5.3 connects with the second half of Section 4, as a toy example of monads induced by adjunctions. Mutually right adjoints from a binary relation induce closure operators.

Section 5.4 looks at the Kleisli category of a closure operator: composition of Kleisli morphisms becomes a kind of transitivity of generation. We also discuss the universal properties of Kleisli and Eilenberg–Moore categories for mutually right adjoints between power sets.

Section 5.5 starts by asking which colimits a monad preserves. It shows that preorders, topologies, and algebras with finitary operations correspond to monads on power sets preserving certain classes of colimits. No other subsection directly depends on it. Its remarks also wander a little beyond our set-operation toy-example theme.

5.1 Monads, Eilenberg–Moore categories, and generation, II

Recall that a monad on $\mathcal{C}$ is a triple (T, η, μ) consisting of an endofunctor $T : \mathcal{C} \rightarrow \mathcal{C}$, a unit $\eta : \text{id}_{\mathcal{C}} \rightarrow T$, and a multiplication $\mu : T^2 \rightarrow T$, satisfying the unit and associativity diagrams. See Definition 5.1.1 of [1].

One ordinary mathematical use of monads is to study monads on Set or other categories and their algebras (Eilenberg–Moore categories), obtaining results in universal algebra. See Chapter 5 of [1].³⁴

Here we consider monads on $P(X)$. Since this is a poset, natural transformations become inequalities and all commutativity conditions are automatic.

Definition 5.1.1 (Closure operator). A *closure operator*, or Moore closure, on X is a monad on $P(X)$: a map $\text{cl} : P(X) \rightarrow P(X)$ satisfying

Functoriality $V \subset U \Rightarrow \text{cl}(V) \subset \text{cl}(U)$ for all $V, U \subset X$.

Unit $V \subset \text{cl}(V)$ for all $V \subset X$.

Multiplication $\text{cl}(\text{cl}(V)) \subset \text{cl}(V)$ for all $V \subset X$.

Unit and multiplication together give $\text{cl}(\text{cl}(V)) = \text{cl}(V)$ for every $V \subset X$. Thus every monad on $P(X)$, as on any poset, is idempotent.

Let us define closed subsets for a closure operator.

Definition 5.1.2 (Closed subset). A subset $U \subset X$ is *closed* for a closure operator cl if it is a fixed point: $\text{cl}(U) = U$. Write $\text{Closed}(\text{cl})$ for the set of all closed subsets.

Closed subsets are actually algebras for the monad.

Theorem 5.1.3 (Closed subsets are algebras for the closure operator). For a closure operator cl on X , its monad algebras are precisely its closed subsets.³⁵ Thus $\text{Closed}(\text{cl})$ is its Eilenberg–Moore category. Consequently, the inclusion has a left adjoint, and $\text{Closed}(\text{cl})$ admits generation.

Proof. An algebra for (T, η, μ) is a pair (c, α) with $\alpha : Tc \rightarrow c$ satisfying unit and associativity conditions. In a power set, as in any poset, these conditions are automatic and α is an inequality. Thus a cl -algebra is a subset V with $\text{cl}(V) \subset V$. The reverse inclusion holds by definition, so these are exactly the closed subsets. Algebra morphisms are underlying morphisms satisfying a commutativity condition, again automatic here. The algebra category is therefore $\text{Closed}(\text{cl})$ ordered by inclusion. The inclusion is fully faithful, so this is a reflective subcategory and, as we saw earlier, a family admitting generation. $\square$

Conversely, since any adjunction induces a monad, a reflective subcategory gives a closure operator. The correspondence between reflective subcategories and idempotent monads is general. For power sets, it takes this form.

³⁴As of February 2021, I think that chapter is really fascinating.

³⁵More precisely, the inclusion $\iota : \text{Closed}(\text{cl}) \rightarrow P(X)$ is monadic and induces the monad cl .

Corollary 5.1.4 (Families admitting generation correspond to closure operators). In a power set $P(X)$ too, idempotent monads correspond to reflective subcategories. Thus families of subsets of X admitting generation correspond bijectively to closure operators on X via

family admitting generation $\mapsto$ induced monad,

closure operator $\mapsto$ Eilenberg–Moore category.

Generally one must pass to isomorphism classes, but here a monad has no naturally isomorphic monad other than itself, and the families give a natural complete set of representatives.

So there are as many examples as families admitting generation. For example, the family of subgroups corresponds to the operator taking a subset to the subgroup it generates.

Remark 5.1.5 (Posets other than power sets). We often restrict statements valid for arbitrary posets to power sets. This subsection is another instance.

For generation outside power sets, consider completion of measure spaces. Let $\text{Meas}(X)$ be the set of measure-space structures on X , ordered by extension. Complete measure-space structures and the completion operation correspond in the same sense as in Corollary 5.1.4.

5.2 Free monads and generation, III

As we saw earlier, families admitting generation are reflective subcategories of a power set. Remark 4.6.16 promised another test.

Proposition 5.2.1 (A practical test for admitting generation). A family $\Sigma \subset P(X)$ admits generation if and only if there is a map $d : P(X) \rightarrow P(X)$ such that, for every $U \subset X$,

$$U \in \Sigma \iff \forall S \subset U, d(S) \subset U$$

holds.

One can prove this by direct calculation, but we will derive it from free monads at the end of the subsection. It is genuinely useful: for many familiar families, simply rewriting the definition as a map d immediately shows that they admit generation.

Example 5.2.2 (Subgroups). A subgroup of G is a subset closed under the identity, products, and inverses, so take

$$d(S) = \{e\} \cup \{xy \mid x, y \in S\} \cup \{x^{-1} \mid x \in S\}$$

As in this example, d describes operations, and Proposition 5.2.1 says to be closed under them. Admitting generation is therefore equivalent to being definable by conditions saying that, whenever certain elements belong, the elements obtained by applying operations to them belong too.³⁶

Example 5.2.3 (Equivalence relations). A relation $R \subset X \times X$ is an equivalence relation if it is reflexive, symmetric, and transitive, so take

$$d(R) = \{(x, x) \mid x \in X\} \cup \{(y, x) \mid (x, y) \in R\} \cup \{(x, z) \mid (x, y), (y, z) \in R\}$$

Thus Proposition 5.2.1 says that families defined by “if you contain these things, you contain those things” admit generation.³⁷ For more details and examples, see [6].

Let us give one more example.

³⁶Even closed subsets have such a description: take the closure operator itself as d . That is a rather boring example here, though.

³⁷It may help to compare partial orders, which do not admit generation, with preorders, which do.

Example 5.2.4 (Equivalence relations compatible with operations). The proposition also shows that, for an algebraic structure, equivalence relations compatible with its operations admit generation. These are equivalence relations for which the operations descend to the quotient independently of representatives.³⁸ Just write down that applying an operation to related elements gives related elements.

For example, given a monoid M and $x, y \in M$, this immediately gives a universal quotient monoid in which $[x] = [y]$.

Remark 5.2.5 (Generating families that admit generation). Among the many consequences of Proposition 5.2.1, let us spell out one: the collection of families of subsets of X admitting generation itself admits generation, as a family of subsets of $P(X)$. This follows immediately from Proposition 5.2.1 and condition 3 of Theorem 4.6.13.

This is useful too. Theorem 4.6.13 says that an intersection of families admitting generation again admits generation. For instance, do closed subgroups of a topological group admit generation? Yes, immediately, because subgroups and closed subsets each do.

Now let us express all this using free monads. As the nLab page <https://ncatlab.org/nlab/show/free+monad> explains, there are several conventions for free monads. We begin with the relevant definitions.

The category of monads on $\mathcal{C}$ has monads as objects and natural transformations commuting with unit and multiplication as morphisms. For posets, those commutativity conditions are automatic. Thus $\text{Monad}(P(X))$ is simply the poset of closure operators ordered pointwise: a full subcategory of $P(X)^{\text{ob}(P(X))}$, itself a poset. Similarly, define the categories of endofunctors and pointed endofunctors (endofunctors with a natural transformation from $\text{id}_{P(X)}$) as full subcategories of that same poset.³⁹

Definition 5.2.6 (The category of monads: the poset of closure operators). For a set X , define the following full subcategories of $P(X)^{\text{ob}(P(X))}$:

Monads $\text{Monad}(P(X))$, consisting of all closure operators on X ;

Pointed endofunctors $\text{PtEnd}(P(X))$, consisting of all endofunctors $F : P(X) \rightarrow P(X)$ with $V \subset F(V)$ for every $V \subset X$;

Endofunctors $\text{End}(P(X)) = P(X)^{P(X)}$, consisting of all endofunctors of $P(X)$.

The free monad we want is the left adjoint of the forgetful functor from $\text{Monad}(P(X))$,

$$\text{Monad}(P(X)) \rightarrow P(X)^{\text{ob}(P(X))}$$

We factor this forgetful functor into three stages,

$$\text{Monad}(P(X)) \hookrightarrow \text{PtEnd}(P(X)) \hookrightarrow \text{End}(P(X)) \hookrightarrow P(X)^{\text{ob}(P(X))}$$

and find their left adjoints separately. The other conventions mentioned above use the left adjoints of the forgetful functors to the intermediate categories $\text{PtEnd}(P(X))$ and $\text{End}(P(X))$. Here all right adjoints are fully faithful, so the three versions agree.⁴⁰

³⁸For groups or rings, the equivalence class of a distinguished element, the identity or zero, identifies such relations with normal subgroups or two-sided ideals. In general, we need the equivalence relation itself.

³⁹Again, all the commutativity conditions are automatic for posets, so these agree with the usual definitions.

⁴⁰For example, the free monad on a pointed endofunctor F is the same as the free monad on its underlying map $F : \text{ob}(P(X)) \rightarrow P(X)$.

Remark 5.2.7 (Factoring morphisms). It is generally useful to keep a morphism factored. First, we lose no information: we can compose whenever we like. Second, knowing that a morphism is built from simpler, familiar ones makes it easier to handle and understand. Here we want a somewhat complicated left adjoint, so we forget structure in stages and construct its left adjoint in stages too. Starting with a map $\text{ob}(P(X)) \rightarrow P(X)$, we make it (1) order-preserving, hence an endofunctor; (2) at least the identity, giving the unit or pointing; and (3) idempotent, giving multiplication. We realize these in separate adjunctions, then compose. Composites of left adjoints are left adjoints.

For an analogous example, try forgetting algebraic structure in stages and factoring the construction of a free algebra accordingly.

First consider $\text{End}(P(X)) \rightarrow P(X)^{\text{ob}(P(X))}$.

Lemma 5.2.8 (Making a map functorial). For a set X and a map $d : \text{ob}(P(X)) \rightarrow P(X)$, define $\tilde{d} : P(X) \rightarrow P(X)$ by

$$\tilde{d}(V) = \bigcup_{S \subset V} d(S)$$

Then $\widetilde{(-)}$ is left adjoint to the forgetful functor $P(X)^{P(X)} \rightarrow P(X)^{\text{ob}(P(X))}$.

Proof. This is easy to check directly, but also follows immediately from general facts about left Kan extensions. We will explain it in that section. $\square$

Next consider the adjoint of $\text{Monad}(P(X)) \rightarrow P(X)^{P(X)}$. The notation $\widetilde{(-)}$ below means something different from the one above.⁴¹

Lemma 5.2.9 (Making an endofunctor pointed). For a set X and an endofunctor $F : P(X) \rightarrow P(X)$, define $\tilde{F} : P(X) \rightarrow P(X)$ by

$$\tilde{F}(V) = V \cup F(V)$$

Then $\widetilde{(-)}$ is left adjoint to $\text{PtEnd}(P(X)) \rightarrow \text{End}(P(X))$.

Proof. Again this is easy to check directly, but it also follows immediately from general facts about slice categories, as the next remark explains. $\square$

Remark 5.2.10 (Forgetting from an undercategory and its left adjoint). In a category $\mathcal{C}$ with binary coproducts, for any $X \in \mathcal{C}$, the projection

$$X/\mathcal{C} \rightarrow \mathcal{C}$$

has a left adjoint, given on objects by $c \mapsto (X \rightarrow X \amalg c)$. See Exercise 11 in Chapter IV of [2], for example.

Since $\text{PtEnd}(P(X)) \cong \text{id}_{P(X)}/\text{End}(P(X))$, we need only set $\tilde{F} = \text{sup}(\text{id}_{P(X)}, F)$. Colimits in the functor category $\text{End}(P(X))$ are objectwise, giving the formula above.

For ordinary mathematical examples, consider the forgetful functors from pointed objects: $\text{Set}_* \rightarrow \text{Set}$, $\text{Top}_* \rightarrow \text{Top}$, and $\text{Group}_* \rightarrow \text{Group}$. Their left adjoints have this form. A less trivial example is extension of scalars for commutative rings. For $f : A \rightarrow B$,⁴² the induced functor $f^* : B\text{-alg} \rightarrow A\text{-alg}$ has left adjoint $B \otimes_A -$. Here these are the categories of commutative B - and A -algebras.

Of course, the dual statement holds too. For example, the universal way to make a topological space X into a bundle⁴³ over a space Y is $X \times Y \rightarrow Y$.

⁴¹I was too lazy to invent another symbol.

⁴²Think of $\mathbb{R} \rightarrow \mathbb{C}$, for example. Taking $A = \mathbb{Z}$ gives the category of commutative rings.

⁴³Here this just means a continuous map to Y .

Finally, we turn a pointed endofunctor into a monad. We send each subset to the least fixed point containing it. As of April 2021, I do not know a general categorical adjunction of which this is an instance.⁴⁴

Lemma 5.2.11 (Making a monad). For a set X and a pointed endofunctor $T : P(X) \rightarrow P(X)$, define $\tilde{T} : P(X) \rightarrow P(X)$ by

$$\tilde{T}(V) = \min\{U \subset X \mid V \subset U, T(U) = U\}$$

Then $(-)$ is left adjoint to $\text{Monad}(P(X)) \rightarrow \text{PtEnd}(P(X))$.

Proof. (Existence of $\tilde{T}$, and why it is a monad.) We show that intersections of fixed points are fixed points. For a family $\{V_\lambda\}_{\lambda \in \Lambda}$ of fixed points,

$$T\left(\bigcap_{\lambda \in \Lambda} V_\lambda\right) \leq \bigcap_{\lambda \in \Lambda} T(V_\lambda) = \bigcap_{\lambda \in \Lambda} V_\lambda \leq T\left(\bigcap_{\lambda \in \Lambda} V_\lambda\right)$$

so $\bigcap_{\lambda \in \Lambda} V_\lambda$ is a fixed point too. The first inclusion uses the property of intersections,⁴⁵ the equality uses that each V_λ is fixed, and the last inclusion uses that T is pointed.

By Theorem 4.6.13, the fixed points admit generation, and $\tilde{T}$ is their corresponding monad, as in the previous subsection.

(Why $T \leq \tilde{T}$.) For any $V \subset X$,

$$T(V) \leq T(\tilde{T}(V)) = \tilde{T}(V)$$

as required. The first inclusion uses $V \leq \tilde{T}(V)$, since $\tilde{T}$ is pointed, and monotonicity of T . The equality uses that $\tilde{T}(V)$ is a fixed point of T .

(Why $\tilde{T}$ is the least monad T' with $T \leq T'$.) Take any monad T' with $T \leq T'$. It suffices to show that $T'(V)$ is a T -fixed point containing V . Containment follows because T' is pointed. The fixed-point condition follows from

$$T'(V) \leq T(T'(V)) \leq T'(T'(V)) = T'(V)$$

where the first inclusion uses the pointing of T , the second uses $T \leq T'$, and the equality uses idempotence of the monad T' . $\square$

Let us gather the results into one proposition.

Proposition 5.2.12 (Free monads and closure operators generated by maps). For a set X , each functor in

$$\text{Monad}(P(X)) \hookrightarrow \text{PtEnd}(P(X)) \hookrightarrow \text{End}(P(X)) \hookrightarrow P(X)^{\text{ob}(P(X))}$$

has a left adjoint.

$$\text{Monad}(P(X)) \xleftarrow{\perp} \text{PtEnd}(P(X)) \xleftarrow{\perp} \text{End}(P(X)) \xleftarrow{\perp} P(X)^{\text{ob}(P(X))}$$

In particular, $\text{Monad}(P(X))$ is a reflective subcategory of $P(X)^{\text{ob}(P(X))}$.

⁴⁴Please tell me if you do.

⁴⁵This is the comparison arrow that always arises when a functor is applied to a limit cone.

We can now construct a free monad from a map. Spelling out its family of closed subsets proves Proposition 5.2.1, as promised.

Corollary 5.2.13 (Closed subsets for a free closure operator). For $d : P(X) \rightarrow P(X)$, write cl_d for its free monad. Then

$$\text{Closed}(\text{cl}_d) = \{U \subset X \mid \forall S \subset U, d(S) \subset U\}$$

holds. In particular, Proposition 5.2.1 is true.

Proof. We work backwards through pointed endofunctors, endofunctors, and maps, writing out the condition of being closed at each stage.

For a pointed endofunctor T , a subset U is closed for its free monad exactly when it is a fixed point of T :

$$T(U) = U$$

Since T is pointed, this is equivalent simply to

$$T(U) \subset U$$

Hence, for an endofunctor F , the necessary and sufficient condition is

$$U \cup F(U) \subset U$$

or simply

$$F(U) \subset U$$

Finally, for a map $d : \text{ob}(P(X)) \rightarrow P(X)$, the necessary and sufficient condition is

$$\bigcup_{S \subset U} d(S) \subset U$$

which is equivalent to

$$\forall S \subset U, d(S) \subset U$$

For the second assertion, Proposition 5.2.1 is an equivalence. We have just proved the harder direction. The easier one follows because $\text{Monad}(P(X))$ is reflective in $P(X)^{\text{ob}(P(X))}$: every closure operator is the free monad on some map $\text{ob}(P(X)) \rightarrow P(X)$, for example on itself. $\square$

Remark 5.2.14 (Free monoids in monoidal categories and recursive closure). I am interested in a general framework for recursive constructions of closure operators. Here is what I mean. For a subset S of a group G , define subsets S_n for $n \in \mathbb{N}$ by

$$\begin{aligned} S_0 &= S \\ S_{n+1} &= \{\text{inverses or products of elements of } S_n, \text{ or the identity}\}. \end{aligned}$$

Then $\bigcup_{n \in \mathbb{N}} S_n$ is the subgroup generated by S . Such a construction does not always work. It does work, at least, when $d : \text{ob}(P(X)) \rightarrow P(X)$ is empty on infinite subsets.⁴⁶ Such maps relate to finitary closure operators, which we revisit in Section 5.5. Write f for the free pointed endofunctor generated by d . The step from S_n to S_{n+1} in the subgroup example is application of f . Then $\bigcup_{n \in \mathbb{N}} f^n(S)$ is the closure of S under the closure operator generated by d . Without substituting a particular S , we can write

$$\sup_{n \in \mathbb{N}} f^n$$

⁴⁶For a contrasting example, define $d : P(\mathbb{R}) \rightarrow P(\mathbb{R})$ by $d(\{x\}) = \{x-1, x+1\}$ for integers x , $d(\mathbb{Z}) = \mathbb{R}$, and the empty set otherwise. Consider the closure of $\{0\}$. If we generalize to recursion along ordinals, regular cardinals become relevant for the same reason as in Lemma I.13.21 of [13].

for the free monad on f , and hence on d .

Free monoids and tensor algebras have analogous constructions. The free monoid on a set A (finite strings with concatenation) is

$$\coprod_{n \in \mathbb{N}} A^n$$

and the tensor algebra on a vector space V is

$$\bigoplus_{n \in \mathbb{N}} V^{\otimes n}$$

Recall that a monad is a monoid object in an endofunctor category. This gives a common setting: the left adjoint of $\text{Monoid}_B \rightarrow B$, for a monoidal category $(B, \otimes, \mathbb{I})$. If B has countable coproducts and both $b \otimes -$ and $- \otimes b$ preserve them, the free monad on b is known to be

$$\coprod_{n \in \mathbb{N}} b^{\otimes n}$$

See Theorem 2 of Chapter VII, Section 3 of [2]. A cocomplete monoidal closed category suffices; this explains free monoids on sets and tensor algebras.

Since suprema are coproducts and the monoidal product on endofunctors is composition, recursive closure is exactly the construction $\coprod_{n \in \mathbb{N}} b^{\otimes n}$. But closure operators need not satisfy the assumptions of Theorem 2 in Chapter VII, Section 3 of [2], so recursion does not work for every d . There seems to be interesting mathematics here, but as of April 2021, I do not know enough to go further. I suspect <https://ncatlab.org/nlab/show/transfinite+construction+of+free+algebras> is relevant.

5.3 Monads from adjunctions and closure operators from relations

An adjunction induces a monad on the codomain of its right adjoint. For more detail, see, for example, Lemma 5.1.3 of [1]. Applying this to the mutually right adjoints of Proposition 4.4.3 gives:

Proposition 5.3.1 (Closure operators induced by binary relations). For a relation $R : X \times Y \rightarrow \mathbf{2}$, the composite $R^Y \circ {}^X R$ is a closure operator on X , and ${}^X R \circ R^Y$ is one on Y . Moreover, $\text{Closed}_{X,R} = \text{Closed}(R^Y \circ {}^X R)$ and $\text{Closed}_{R,Y} = \text{Closed}({}^X R \circ R^Y)$.

This supplies the promised proof of reflectivity in Theorem 4.7.1.

Remark 5.3.2. In fact, every closure operator has this form. Every complete lattice is a reflective subcategory of a power set. Given cl on $P(X)$, choose $P(Y)$ containing $Q = (\text{Closed}_{\text{cl}})^{\text{op}}$ reflectively. Joining the two reflective adjunctions gives mutually right adjoints between power sets, hence an adjunction induced by a relation. Its fixed points are still $\text{Closed}_{\text{cl}}$, so every closure operator is induced by a binary relation.

There are as many examples as binary relations. Here are a few from Section 4.

Example 5.3.3 (Closure operators in Galois theory). For a finite Galois extension $K \subset L$, the relation $\sigma(x) = x$ induces the operators taking the intermediate field generated over K and the subgroup generated in $\text{Gal}(L/K)$.

Example 5.3.4 (Closure operators in the Nullstellensatz). For an algebraically closed field $\mathbb{K}$ and positive integer n , the relation $f(a_1, \dots, a_n) = 0$ between $\mathbb{K}[x_1, \dots, x_n]$ and $\mathbb{K}^n$ induces two closure operators: taking the least radical ideal containing $S \subset \mathbb{K}[x_1, \dots, x_n]$, and taking closure in the Zariski topology in the ordinary sense.

Example 5.3.5 (Closure from orthogonality in a Hilbert space). For a Hilbert space $\mathcal{H}$, orthogonality induces closed subsets that are exactly the closed linear subspaces. Its closure operator sends $S \subset \mathcal{H}$ to the smallest closed linear subspace containing it.

Example 5.3.6 (Closure in the **Galois connection between syntax and semantics**). For a first-order language $\mathcal{L}$, consider satisfaction $\models: \text{Structure}_{\mathcal{L}} \times \text{Sentence}_{\mathcal{L}} \rightarrow \mathbf{2}$, between the **class**⁴⁷ of structures and the set of sentences. The induced closure operator on sentences is deductive closure.⁴⁸

Thus important relations yield important closure operators by an abstract procedure. It is often important to turn the resulting operator into an explicit, workable description. For instance, closed subsets arising merely from polynomial evaluation can be identified elementwise as radical ideals: subsets containing 0 and closed under addition, scalar multiplication, and taking roots of powers. Likewise, the closure operator from syntax and semantics is defined semantically; Gödel’s completeness theorem can be viewed as saying that, with suitable logical axioms, this operator has a syntactic description.

Remark 5.3.7 (The sheaf analogue). idempotent adjunction

$$\begin{array}{ccc} & L & \\ \text{Top}/X & \xleftarrow{\quad} & \text{Set}^{\mathcal{O}(X)^{\text{op}}} \\ & \Gamma & \end{array} \quad \perp$$

The induced monad on $\text{Set}^{\mathcal{O}(X)^{\text{op}}}$ is sheafification $: \text{Set}^{\mathcal{O}(X)^{\text{op}}} \rightarrow \text{Set}^{\mathcal{O}(X)^{\text{op}}}$. There are several explicit descriptions of sheafification; we will mention one using the pointwise Kan extension formula later.

5.4 Kleisli categories and transitivity of generation

Certain containment relations involving generation are transitive. For example, write $\langle A \rangle$ for the ideal generated by $A \subset R$ in a commutative ring. If $A \subset \langle B \rangle$ and $B \subset \langle C \rangle$, then $A \subset \langle C \rangle$. Thus $A \rightsquigarrow B$ defined by $A \subset \langle B \rangle$ is a preorder on $P(R)$.⁴⁹ We have constructed $(P(R), \rightsquigarrow)$ from $(P(R), \langle - \rangle)$. This uses not just transitivity of inclusion but the closure-operator axioms. It is an instance of the Kleisli construction, available for any set with a closure operator, or more generally any category with a monad.

Since morphisms in posets are inequalities, the definition of Kleisli composition gives the following.

Proposition 5.4.1 (The Kleisli category of a closure operator). For a closure operator cl on X , define $A \rightsquigarrow B \iff A \subset \text{cl}(B)$ on $P(X)$. This relation is reflexive and transitive, hence a preorder and therefore a category.

Proof. Reflexivity is the unit axiom. For transitivity, suppose $A \subset \text{cl}(B)$ and $B \subset \text{cl}(C)$. Writing inclusions as arrows to emphasize the general Kleisli construction, we have

$$\begin{array}{ccccc} & & \text{cl}(B) & & \text{cl}(C) \\ & \nearrow f & \uparrow \eta & \nearrow g & \uparrow \eta \\ A & & B & & C \end{array}$$

⁴⁷**Red text** marks size issues whose resolution I do not understand as of January 2021. See Section 3.6, “Posets as sets,” of [5].

⁴⁸Yasuda (@yasuda_614_set) pointed out that, if we only want deductive closure, we can resolve the size issue as follows. Put $\kappa = \max(\aleph_0, |\mathcal{L}|)$. Structures whose underlying sets are subsets of κ form a set $\text{Structure}_{\mathcal{L}, \kappa}$. Satisfaction on this set induces the same deductive closure on $\text{Sentence}_{\mathcal{L}}$: by the Löwenheim–Skolem theorem, every model of cardinality at least κ has an elementarily equivalent model of cardinality κ .

⁴⁹Preorders are more natural than posets from several perspectives, including truth-value enrichment. They stand to posets as categories stand to skeletal categories.

The inclusions labelled η and μ are the unit and multiplication of the closure operator. Then

$$\begin{array}{ccccc}
 & & & \text{cl}(\text{cl}(C)) & \\
 & & \nearrow \text{cl}(g) & \uparrow \mu & \\
 & \text{cl}(B) & & \text{cl}(C) & \\
 \nearrow f & \uparrow \eta & \nearrow g & \uparrow \eta & \\
 A & B & & C &
 \end{array}$$

proves the claim. $\square$

The proof is somewhat wordy, but emphasizes the general Kleisli argument. Let us name this preorder.

Definition 5.4.2 (Kleisli). For a closure operator cl on X , write $\text{Kleisli}(\text{cl})$ for the preorder $(P(X), \rightsquigarrow)$ just defined.

This sort of transitivity is useful and often used. The Kleisli preorder simply compares closures by inclusion. Here are examples beyond the generation of subalgebras with which we began.

Example 5.4.3 (The stone problem). First, an example unlikely to interest most people, but very personally familiar to me. The argument in Claim 5.2 of my earlier “The Stone Problem” [7] is a Kleisli-category example. A free monoid acts on a standard simplex, and from this action we construct a closure operator on the simplex.

Example 5.4.4 (Transitivity of provability). For a first-order language $\mathcal{L}$, let cl be deductive closure on $\text{Sentence}_{\mathcal{L}}$. Its Kleisli category says that if $\Sigma \subset \text{cl}(\Sigma')$ and $\Sigma' \subset \text{cl}(\Sigma'')$, then $\Sigma \subset \text{cl}(\Sigma'')$. In terms of provability: if $\Sigma' \vdash_{\mathcal{L}} \phi$ for every $\phi \in \Sigma$, and $\Sigma'' \vdash_{\mathcal{L}} \psi$ for every $\psi \in \Sigma'$, then $\Sigma'' \vdash_{\mathcal{L}} \psi$ for every $\psi \in \Sigma$.

Two subsets Σ, Σ' of $\text{Sentence}_{\mathcal{L}}$ are isomorphic in $\text{Kleisli}(\text{cl})$ exactly when their deductive closures agree, that is, when they axiomatize the same theory. ⁵⁰

Kleisli composition of arrows that cannot be composed in the ordinary sense appears all over mathematics: composition of partial maps, stochastic maps between measurable spaces, and more. See, for example, Example 5.2.10 of [1].

Notice that the **class**-of-models assignment is functorial not just for inclusions of axiom systems, but for Kleisli morphisms: it reverses inclusion of deductive closures too. Likewise, the fixed field of a subset S of a Galois group is the fixed field of the subgroup it generates. In Example 5.3.4, a set of polynomials has the same common zero set as its generated ideal or radical ideal. These are instances of the universal property of the Kleisli category. Let us state this, together with the Eilenberg–Moore universal property, for mutually right adjoints arising from a relation.

Proposition 5.4.5 (Universal properties of Kleisli and Eilenberg–Moore categories). For a relation $R : X \times Y \rightarrow \mathbf{2}$, there is a unique order-preserving map in each case making the following diagram commute.

$$\begin{array}{ccccc}
 \text{Kleisli}(R^Y \circ {}^X R) & \xrightarrow{\exists!} & P(Y)^{\text{op}} & \xrightarrow{\exists!} & \text{Closed}_{X,R} \\
 & \nwarrow & \uparrow R^Y & \nearrow {}^X R & \\
 & & P(X) & &
 \end{array}$$

Their actions on elements agree with R^Y and ${}^X R$, respectively.

Thus Kleisli is the natural domain making R^Y order-preserving, and Closed is the natural codomain for ${}^X R$.

⁵⁰Here this means exactly that their deductive closures agree!

5.5 Colimit preservation: orders, topologies, and finitary algebras

Which colimits a monad preserves is important information. Indeed, when a monad and its square preserve certain colimits, the associated monadic functor creates those colimits.⁵¹ For more detail, see, for example, Theorem 5.6.5. of [1]. Here are examples from ordinary mathematics.

First, a reason why module colimits are computed as abelian groups.

Remark 5.5.1 (Colimits of modules). For a ring R , left R -modules form the Eilenberg–Moore category of $R \otimes_{\mathbb{Z}} -$ on $\mathbf{Ab}$. This monad is cocontinuous, since it has a right adjoint. Thus $R\text{-Mod} \rightarrow \mathbf{Ab}$ creates all colimits.

Algebras for monads on $\mathbf{Set}$ include ordinary-looking algebras and less obviously algebraic things such as $\mathbf{cHaus}$. Preservation of filtered colimits is one way to distinguish the ordinary-looking ones.⁵²

Remark 5.5.2 (Filtered colimits of groups and algebras). Forgetful functors to $\mathbf{Set}$ from categories of models of algebraic theories,⁵³ such as $\mathbf{Group}$, $\mathbf{Monoid}$, $\mathbf{Ab}$, $\mathbf{Ring}$, $\mathbf{Rng}$, $\mathbf{CRing}$, $\mathbf{Vect}_{\mathbb{K}}$, $\mathbf{Lattice}$, and $R\text{-Mod}$, preserve filtered colimits.

For example, a free algebra on an infinite set is a union of free algebras on finite subsets. Consider polynomials over $\mathbb{C}$ in infinitely many variables. The forgetful functor $U : \mathbb{C}\text{-alg} \rightarrow \mathbf{Set}$ creates filtered colimits for the same reason. To construct the free algebra on X , first write X as the colimit of its finite subsets and their inclusions, a filtered diagram in $\mathbf{Set}$. Apply the free-algebra functor $F : \mathbf{Set} \rightarrow \mathbb{C}\text{-alg}$. Since it is a left adjoint, it preserves this colimit. Since U creates filtered colimits, the resulting free algebra is the union, along the inclusions, of the ordinary polynomial rings in finitely many variables, equipped with the unique compatible commutative $\mathbb{C}$ -algebra structure.

To see why $\mathbf{cHaus} \rightarrow \mathbf{Set}$ does not create filtered colimits, consider its ultrafilter monad. If it preserved filtered colimits, the unit, which is an isomorphism on finite sets, would also be one on infinite sets.⁵⁴

What about $P(X)$? We distinguish three classes by the colimits they preserve. Following standard terminology, call functors preserving all colimits cocontinuous, those preserving finite colimits right exact, and those preserving filtered colimits finitary. We use the same terms for closure operators. Here colimits are suprema, or unions in power sets, so the arrows in a diagram do not affect preservation. In particular, right exactness here is equivalent to preservation of finite coproducts.

Definition 5.5.3 (Colimit preservation by closure operators). A closure operator cl on X is

- *right exact* if it preserves finite colimits;
- *cocontinuous* if it preserves all colimits;
- *finitary* if it preserves filtered colimits.

Explicitly, right exactness means preserving the empty set and binary unions:

$$\text{cl}(\emptyset) = \emptyset$$

$$\text{cl}(U \cup V) = \text{cl}(U) \cup \text{cl}(V)$$

⁵¹Taking for granted the already remarkable fact that it creates all existing limits.

⁵²See <https://ncatlab.org/nlab/show/finitary+monad>.

⁵³In the sense of Definition 5.5.5 of [1]. Looking at the definition, this assertion alone is almost tautological.

⁵⁴This uses the existence of a nonprincipal ultrafilter on some infinite set. I do not know the set-theoretic details, but here is a discussion of their existence in ZF: <https://mathoverflow.net/questions/59157/reference-request-independence-of-the-ultrafilter-lemma-from-zf>.

as above.⁵⁵

Cocontinuity means that, for every family $\{U_\lambda\}_{\lambda \in \Lambda}$ of subsets of X ,

$$\text{cl} \left(\bigcup_{\lambda \in \Lambda} U_\lambda \right) = \bigcup_{\lambda \in \Lambda} \text{cl}(U_\lambda).$$

Finitariness means that whenever a family $\{U_\lambda\}_{\lambda \in \Lambda}$ is filtered—for every finite subset $\Lambda_0 \subset \Lambda$ there is a $\mu \in \Lambda$ with $\bigcup_{\lambda \in \Lambda_0} U_\lambda \subset U_\mu$ —⁵⁶ we have

$$\text{cl} \left(\bigcup_{\lambda \in \Lambda} U_\lambda \right) = \bigcup_{\lambda \in \Lambda} \text{cl}(U_\lambda).$$

In fact, increasing unions suffice to test finitariness. We will not use this later, and it goes beyond our categorical toy-example purpose, so I put it in a remark.

Remark 5.5.4 (A concise criterion for finitariness). A closure operator is finitary if and only if increasing unions of closed subsets are closed. More precisely, for every nonempty totally ordered subset $\Sigma \subset \text{Closed}(\text{cl})$, we require

$$\bigcup_{S \in \Sigma} S \in \text{Closed}(\text{cl})$$

that its union be closed.

Proof. The forward implication is easy; the converse takes a little work. By Proposition 5.5.11, it suffices to show, for every $U \subset X$,

$$\text{cl}(U) = \bigcup_{\substack{V \subset U \\ V \text{ finite}}} \text{cl}(V).$$

Use transfinite induction on the cardinality of U . The finite case is easy. For infinite U , put $\kappa = |U|$ and enumerate its elements as

$$U = \{u_\xi \mid \xi \in \kappa\}$$

indexed by κ , formally choosing a bijection $\kappa \rightarrow U$. For every $\alpha \in \kappa$, the initial segment $U_\alpha := \{u_\xi \mid \xi \in \alpha\}$ has cardinality strictly less than κ . Hence induction gives

$$\text{cl}(U_\alpha) = \bigcup_{\substack{V \subset U_\alpha \\ V \text{ finite}}} \text{cl}(V).$$

By our assumption on increasing unions of closed subsets,

$$\begin{aligned} \text{cl}(U) &= \bigcup_{\alpha \in \kappa} \text{cl}(U_\alpha) \\ &= \bigcup_{\alpha \in \kappa} \left(\bigcup_{\substack{V \subset U_\alpha \\ V \text{ finite}}} \text{cl}(V) \right) \\ &= \bigcup_{\substack{V \subset U \\ V \text{ finite}}} \text{cl}(V), \end{aligned}$$

as required. □

⁵⁵This is the familiar trick of reducing finiteness to a few finite cases, rather like reducing general associativity for monoids to arity 0 (the unit) and arity 3 (associativity).

⁵⁶Splitting finiteness into 0 and 2, it is enough to consider empty and two-element Λ_0 .

Our three classes correspond to topologies, preorders, and closure under finitary operations, respectively.

First, right exact closure operators correspond to topologies. See also <https://ncatlab.org/nlab/show/topological+space#AlternateDefinitions>.

Proposition 5.5.5 (Right exact monads and topological spaces). Topologies on X correspond bijectively to right exact closure operators on X , by

topology $\mapsto$ the usual closure operator,

right exact closure operator $\mapsto$ the topology with its fixed points as closed sets.

The usual closure returns the least closed set containing a subset; fixed points here are closed sets in the sense of Definition 5.1.2.

Proof. This is easy to check directly, but also follows immediately from the fact that if a monad and its square preserve colimits, the monadic functor creates them. $\square$

Remark 5.5.6 (Defining topologies by closure operators). The closure-operator definition of a topology is often technically useful. For instance, the proof of Corollary 5.5.6 of [1], that $\mathbf{cHaus} \rightarrow \mathbf{Set}$ is monadic, uses it.⁵⁷ Pontrjagin also defines topological spaces by closure operators in *Topological Groups* [8].⁵⁸

Ionads, introduced in [14] as categorified topological spaces, also use the dual of this viewpoint: <https://ncatlab.org/nlab/show/ionad>.⁵⁹

Since a relation determines its closure operator, it should be possible to characterize relations inducing right exact closure using only the relation itself. As of April 2021, I have no concise description, so here are examples instead. Every closure operator comes from a relation, so every topology does too, whether or not the description is natural. Two famous examples follow.

Example 5.5.7 (The Zariski topology). For a commutative ring R , write $\mathbf{Spec}(R)$ for its set of prime ideals. The relation $\in: R \times \mathbf{Spec}(R) \rightarrow \mathbf{2}$ induces a right exact closure operator on $\mathbf{Spec}(R)$. This is the Zariski topology.

Example 5.5.8 (Stone spaces). For a Boolean algebra B , let $\mathbf{Bool}(B, \mathbf{2})$ be its set of Boolean algebra homomorphisms to $\mathbf{2}$. The relation “evaluation gives 0” induces a right exact closure operator on this set. This is the Stone space of B .

Next, cocontinuous closure operators correspond to preorders. The following is an immediate corollary of Proposition 7.3.4.

Proposition 5.5.9 (Cocontinuous monads and preorders). Preorders on X correspond bijectively to cocontinuous closure operators on X , by

preorder $\mapsto$ downward closure,

$\text{cl} \mapsto (x \leq y \iff x \in \text{cl}(\{y\})).$

Proof. A cocontinuous functor $F: P(X) \rightarrow P(X)$ is determined by its values on singletons. See Proposition 7.3.4. Thus it is determined by the relation $x \in F(\{y\})$.⁶⁰ Translating the monad axioms into conditions on this relation, the unit demands reflexivity and multiplication demands transitivity. $\square$

⁵⁷Continuity can be expressed through closure. When all maps under consideration are closed, as in $\mathbf{cHaus}$, continuity amounts to commuting with closure, which simplifies matters.

⁵⁸With the extra T_1 condition that the closure of a singleton is itself.

⁵⁹As of April 2021, I have no idea how fundamental ionads are.

⁶⁰For the same reason as in Theorem 4.4.10.

Cocontinuity implies right exactness, so a preorder should give a topology. Indeed, declaring downward-closed subsets to be closed makes a preorder into a topological space. This is the Alexandrov topology .

Remark 5.5.10 (κ -cocontinuous closure operators). More generally, for a cardinal κ , we can consider κ -cocontinuous closure operators: those preserving colimits of diagrams of size less than κ . For $\kappa = \aleph_0$, these are topologies; for κ sufficiently large relative to X , say $\kappa > 2^{|X|}$, they are preorders. Thus topologies and preorders sit in a cardinal-indexed sequence of classes of closure operators.⁶¹

Finally, let us characterize finitariness. Proposition 5.2.1 describes a family admitting generation as being closed under some operation d . The next proposition says that finitary closure is equivalent to being describable by closure under a possibly infinite collection of finitary operations.

Proposition 5.5.11 (Finitary monads and finitary operations). A closure operator $\text{cl} : P(X) \rightarrow P(X)$ is finitary if and only if there is a map $d : \text{ob}(P(X)) \rightarrow P(X)$ such that

- $d(U) = \emptyset$ for every infinite $U \subset X$;
- cl is the free monad generated by d .⁶²

Proof. If cl is finitary, take d to agree with cl on finite subsets and to be empty on infinite ones. For every $U \subset X$,

$$\begin{aligned} \text{cl}(U) &= \bigcup_{\substack{V \subset U \\ V \text{ finite}}} \text{cl}(V) \\ &= \bigcup_{V \subset U} d(V) \end{aligned}$$

holds. The first equality uses preservation of the filtered colimit, here the union, of all finite subsets of U . Thus cl is the functorialization of d from Lemma 5.2.8. Since this functorialization is already a monad, it is the free monad generated by d .⁶³

Conversely, suppose such a map d exists. Take a filtered family of closed subsets $\{S_\lambda\}_{\lambda \in \Lambda}$. It suffices to show that $S = \bigcup_{\lambda \in \Lambda} S_\lambda$ is closed. Every finite subset of S lies in some S_λ , so Corollary 5.2.13 finishes the proof. $\square$

Example 5.5.12 (Subalgebras). Subgroups, ideals, subrings, submodules, submonoids, and other subalgebras have finitary closure operators: just encode their operations in d . The same immediately shows finitariness for equivalence relations, preorders, and anything defined by “if these finitely many things belong, that thing belongs.”

Increasing unions are filtered colimits, so an increasing union of closed subsets, such as ideals, is closed again.^{64,65}

Remark 5.5.13 (Subalgebras are the only examples). In the following sense, subalgebras are the only examples of finitary closure operators. Call a set equipped with a possibly infinite collection of finitary operations a finitary algebra. Formally, it is a tuple $(X, O, \alpha, \{*_o\}_{o \in O})$, consisting of an underlying set X , operation symbols O , an arity map $\alpha : O \rightarrow \mathbb{N}$, and maps $*_o : X^{\alpha(o)} \rightarrow X$ for $o \in O$.

⁶¹As of April 2021, I have not thought of a good story to tell about this.

⁶²In the sense of Proposition 5.2.12.

⁶³Here we use reflectivity of the free-monad adjunction.

⁶⁴As with our other claims, this is easy to prove directly.

⁶⁵This is a typical setting for Zorn’s lemma. If some singleton generates the whole set, as for ideals or deductive closure, one can obtain maximal proper closed subsets: maximal ideals or maximal consistent sets, respectively.

A subalgebra is a subset closed under every operation.⁶⁶

By Proposition 5.2.1, subalgebras form a family admitting generation. The preceding proposition also shows that its closure is finitary. Conversely, every finitary closure operator has this form. Given cl on X , we can equip X with a finitary algebra structure whose subalgebra closure is cl . The construction need not be natural.⁶⁷

For instance, convex subsets of $\mathbb{R}^n$ admit generation and are closed under filtered unions. They should therefore be subalgebras of some finitary algebra structure. Take, for each $t \in [0, 1]$, the binary operation selecting the point dividing a line segment in the ratio $t : 1 - t$.

Remark 5.5.14 (How the classes of closure operators overlap). We have considered cocontinuous, right exact, and finitary closure operators. For our closure operators, it is easy to see that

$$\text{cocontinuous} \iff \text{right exact and finitary}.$$

Here are the possibilities.

Cocontinuous Downward-closed subsets of a suitable poset.

Right exact, not finitary Any topology not arising from downward-closed subsets, such as the usual closure on $\mathbb{R}$.

Finitary, not right exact Any finitary algebra whose subalgebras do not form the closed sets of a topology. For example, subgroup closure in the additive group $\mathbb{Z}$.

Neither Everything else. For example, take the closure operator on the disjoint union $\mathbb{R} \amalg \mathbb{Z}$ of the topological space $\mathbb{R}$ and the additive group $\mathbb{Z}$.⁶⁸

6 Cartesian Closed Category

Here we discuss power sets as cartesian closed categories.

Section 6.1 shows that a power set is cartesian closed: taking a product has a right adjoint. On predicates, that adjoint is implication. An adjunction gives four rules through LAPC, RAPL, unit, and counit; here they include distributivity of intersection and modus ponens.

Section 6.2 gives the dual results, using that the opposite of a power set category is again a power set. The resulting adjunction involves set difference.

Section 6.3 considers mutually right adjoints from the cartesian closed structure. These give implication in the opposite direction from Section 6.1.

Section 6.4 applies a criterion for a right adjoint between cartesian closed categories to be cartesian closed to inverse image, giving the projection formula for sets. As of April 2021, I am not familiar enough with this to explain its relation to the many other projection formulas, so I have only linked to the nLab.

6.1 Power sets, implication, and modus ponens

Recall that a cartesian closed category (CCC) has finite products and a right adjoint to $A \times (-)$ for each object A . For more detail, see, for example, Definition 4.3.9. of [1]. Power sets are cartesian closed.

⁶⁶If we include only the binary group operation, subalgebras are merely subsemigroups. We also need the nullary identity and unary inverse operations. This is a definitional extension of the binary-operation definition, allowing groups to be described by an equational theory. See Section II.15 of [12].

⁶⁷For example, for an operation not defined everywhere, we can fill in undefined values by arbitrarily returning one of its inputs.

⁶⁸Take closure separately in the two parts.

Proposition 6.1.1 (Power sets are cartesian closed). For a set X and $A \subset X$, the functor

$$A \cap (-) : P(X) \rightarrow P(X)$$

has right adjoint

$$(-) \cup A^c : P(X) \rightarrow P(X).$$

That is, for all $U, V \subset X$,

$$A \cap U \subset V \iff U \subset V \cup A^c$$

holds. Thus $P(X)$ is cartesian closed.

In terms of predicates, this right adjoint is familiar. From now on, distinguish $\rightarrow$ from $\Rightarrow$. Write $\Rightarrow$ for the order on 2^X . When there is a risk of confusion, we write “iff” for equivalence. For predicates ϕ, ψ , define a new predicate $\phi \rightarrow \psi$ by $(\phi \rightarrow \psi)(x) = \text{true}$ iff $\phi(x) = \text{false}$ or $\psi(x) = \text{true}$.

Proposition 6.1.2 (Cartesian closure in terms of predicates). For a predicate $\phi : X \rightarrow \mathbf{2}$, the functor

$$\phi \wedge (-) : \mathbf{2}^X \rightarrow \mathbf{2}^X$$

has right adjoint

$$\phi \rightarrow (-) : \mathbf{2}^X \rightarrow \mathbf{2}^X.$$

Thus, for predicates ψ, τ ,

$$\phi \wedge \psi \Rightarrow \tau \text{ iff } \psi \Rightarrow \phi \rightarrow \tau$$

holds.

Remark 6.1.3 (Multivariable adjunctions). Such “adjunctions with a parameter” are usually⁶⁹ described by multivariable adjunctions. Here we have the two-variable adjunction involving $\wedge, \rightarrow, \rightarrow$. This includes conjunction–implication, currying in $\mathbf{Set}$, and tensor–hom adjunctions in $\mathbf{Ab}$ or modules, all within the setting of symmetric closed monoidal categories. See https://ncatlab.org/nlab/show/closed+monoidal+category#symmetric_closed_monoidal_category.

Let us record the immediate consequences of the adjunction.

Corollary 6.1.4 (LAPC and distributivity). For subsets A, B, C of a set X ,

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

holds. Similarly, for predicates ϕ, ψ, τ on X ,

$$\phi \wedge (\psi \vee \tau) = (\phi \wedge \psi) \vee (\phi \wedge \tau)$$

holds.

The opposite of a CCC need not be cartesian closed, so for now we refrain from writing

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

as a dual claim. Of course, since every subset is the complement of another, RAPL does give this identity.

Let us look at ordinary mathematics—or should I say arithmetic!?

⁶⁹I am tempted to say always: if a multivariable functor has an adjoint for each parameter, these extend uniquely to a multivariable adjunction. For more detail, see, for example, Proposition 4.3.6. of [1].

Remark 6.1.5 (Arithmetic refresher, I: distributivity). The category $\mathbf{Fin}$ of finite sets is a typical CCC. The right adjoint of $A \times (-)$ is exponentiation $(-)^A$. Writing $A + B$ for disjoint union, the corresponding fact is

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

Passing to isomorphism classes, that is, sizes, gives the distributive law $l(m + n) = lm + ln$ for natural numbers.⁷⁰

Remark 6.1.6 (Distributivity of cardinal arithmetic). Finiteness was unnecessary: $\mathbf{Set}$ is also cartesian closed, and its isomorphism classes are cardinals. For cardinals κ, λ, θ , we get

$$\kappa \cdot (\lambda + \theta) \cong \kappa \cdot \lambda + \kappa \cdot \theta$$

For RAPL, let us give only the predicate version.

Corollary 6.1.7 (RAPL and distributivity of implication, I). For predicates ϕ, ψ, τ on X ,

$$\phi \rightarrow (\psi \wedge \tau) = (\phi \rightarrow \psi) \wedge (\phi \rightarrow \tau)$$

holds.

Remark 6.1.8 (Arithmetic refresher, II: an exponent law). The corresponding isomorphism in $\mathbf{Fin}$ is, for finite sets A, B, C ,

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

Passing to sizes gives $(mn)^l = m^l n^l$.

Remark 6.1.9 (An exponent law for cardinals, I). The corresponding isomorphism in $\mathbf{Set}$ is, for sets A, B, C ,

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

Passing to cardinalities gives $(\kappa \cdot \lambda)^\theta = \kappa^\theta \cdot \lambda^\theta$.

The unit and counit are further immediate consequences of an adjunction. Let us spell out just the predicate interpretation of the counit.

Corollary 6.1.10 (Modus ponens as a counit). For any predicates ϕ, ψ on X ,

$$\phi \wedge (\phi \rightarrow \psi) \Rightarrow \psi$$

holds.

Modus ponens corresponds to evaluation.

Remark 6.1.11 (The corresponding counit in $\mathbf{Set}$). For sets A, B , the corresponding map in the CCC $\mathbf{Set}$ is evaluation:

$$\text{ev}_{A,B} : A \times B^A \rightarrow B$$

Since $P(X)$ is contravariantly isomorphic to itself, there should be a dual version replacing intersection by union. Let us start a new subsection.

⁷⁰When we learned arithmetic, we did not call it addition and multiplication in $\mathbb{N}$: we could still see it as rules for finite collections of things!

6.2 The dual picture and set difference

We obtained implication as an adjoint of conjunction. The dual adjunction characterizes set difference.

First, the definition.

Definition 6.2.1 (Set difference). For subsets U, V of X , define

$$U \setminus V = \{x \in U \mid x \notin V\}$$

Set difference is characterized by the following relation.

Proposition 6.2.2 (Set difference as a left adjoint). For a set X and a subset V , the functor

$$A \cup (-) : P(X) \rightarrow P(X)$$

has

$$(-) \setminus A : P(X) \rightarrow P(X)$$

as its left adjoint. That is, for all subsets $U, V \subset X$,

$$U \setminus A \subset V \iff U \subset A \cup V$$

holds.

Let us collect the consequences of the adjunction too.

Corollary 6.2.3 (Consequences of the union–difference adjunction). For subsets A, U, V of X ,

unit $U \subset A \cup (U \setminus A)$

counit $(A \cup U) \setminus A \subset U$

LAPC $(U \cap V) \setminus A = (U \setminus A) \cap (V \setminus A)$

RAPL $A \cup (U \cap V) = (A \cup U) \cap (A \cup V)$

hold.

6.3 Mutually right adjoints

Besides $\phi \wedge (-) \dashv \phi \rightarrow (-)$, the two-variable adjunction for $\wedge, \rightarrow, \rightarrow$ contains a mutually right adjunction of $(-) \rightarrow \phi$ with itself. This follows from Proposition 6.1.1.⁷¹

Proposition 6.3.1 (Mutually right adjoints from implication). For a predicate ϕ on X , the functor

$$(-) \rightarrow \phi : \mathbf{2}^X \rightarrow \mathbf{2}^X$$

is mutually right adjoint to itself.

Remark 6.3.2 (The corresponding binary relation). Recall that mutually right adjoints between power sets correspond to binary relations. For $(-) \rightarrow \phi$, the relation is inequality with an exception for elements satisfying ϕ : $x \sim y \iff \phi(x) \text{ or } \phi(y) \text{ or } x \neq y$. If ϕ is identically false, this reduces to negation.

Since the two adjoints are the same, let us record RAPL.

⁷¹Use the unique extension of a parameterized adjunction to a multivariable adjunction.

Corollary 6.3.3 (RAPL and distributivity of implication, II). For predicates ϕ, ψ, τ on X ,

$$(\psi \vee \tau) \rightarrow \phi = (\psi \rightarrow \phi) \wedge (\tau \rightarrow \phi)$$

holds.

Implication corresponds to exponentiation, so the arithmetic counterpart is again an exponent law.

Remark 6.3.4 (Arithmetic refresher, III: another exponent law). For finite sets A, B, C , the corresponding isomorphism in $\mathbf{Fin}$ is

$$(A)^{B+C} \cong A^B \times A^C$$

and passing to sizes gives $m^{n+l} = m^n m^l$.

Remark 6.3.5 (An exponent law for cardinals, II). For sets A, B, C , the corresponding isomorphism in $\mathbf{Set}$ is

$$(A)^{B+C} \cong A^B \times A^C$$

and passing to cardinalities gives $\kappa^{\lambda+\theta} = \kappa^\lambda \cdot \kappa^\theta$.

Remark 6.3.6 (unit). The unit is

$$\phi \Rightarrow (\phi \rightarrow \psi) \rightarrow \psi$$

In particular, if ψ is identically false, this becomes

$$\phi \Rightarrow \neg \neg \psi.$$

The same phenomenon occurs in any CCC, or more generally any symmetric closed monoidal category. For example, in $\mathbf{Vect}_{\mathbb{K}}$ with tensor product, the functor $\mathbf{Vect}_{\mathbb{K}}(-, \mathbb{K})$ is dualization $(-)^*$. The unit is the canonical linear map $V \rightarrow V^{**}$ into the bidual. There are also maps both ways between the single and triple dual.⁷² For posets such as Heyting algebras, arrows both ways must be isomorphisms.

6.4 Cartesian closed functors, implication, and the projection formula

As of February 2021, I am not yet comfortable with this material. A functor $F : \mathcal{C} \rightarrow \mathcal{D}$ between CCCs is cartesian closed if it preserves products and the canonical map $F(A^B) \rightarrow F(A)^{F(B)}$ is an isomorphism for every $A, B \in \mathcal{C}$. See <https://ncatlab.org/nlab/show/cartesian+closed+functor>.

Proposition 6.4.1 (Inverse image is a cartesian functor). For any map $f : X \rightarrow Y$, inverse image $f^{-1} : P(X) \rightarrow P(Y)$ is a cartesian closed functor.

Proof. We saw product preservation in the adjunction section. Now use $A^B = A \cup B^c$ and preservation of unions and complements by inverse image. $\square$

For a right adjoint $R : \mathcal{C} \rightarrow \mathcal{D}$ between CCCs, with left adjoint L , cartesian closure is equivalent to the canonical map $L(R(A) \times B) \rightarrow A \times L(B)$ being an isomorphism for every $A \in \mathbf{ob}(\mathcal{C})$ and $B \in \mathbf{ob}(\mathcal{D})$. See <https://ncatlab.org/nlab/show/Frobenius+reciprocity#InCategoryTheory> and its reference [11].

Since direct image is left adjoint to inverse image, we obtain:

⁷²Dualize before or after taking the unit.

Corollary 6.4.2 (The projection formula for set operations). For $f : X \rightarrow Y$ and any $V \subset X$, $U \subset Y$,

$$f(f^{-1}(U) \cap V) = U \cap f(V)$$

holds.

The special case $V = X$ includes

$$f(f^{-1}(U)) = U \cap f(X)$$

Remark 6.4.3 (Other projection formulas). As of February 2021, I do not know enough to explain the relation to formulas such as the representation-theoretic projection formula

$$\text{Ind}(\text{Res}(W) \otimes V) \cong W \otimes \text{Ind}(V)$$

in detail. See <https://ncatlab.org/nlab/show/Frobenius+reciprocity> and <https://ncatlab.org/nlab/show/Wirthm%C3%BCller+context>.

7 Kan extensions

Here we use Kan extensions to give intuitive explanations of several phenomena encountered in this text.

Section 7.1 spells out the pointwise Kan extension formula for power-set-valued functors. It computes the “functorialization” from Section 5.2.

Section 7.2 treats truth-value-valued Kan extensions. These are very simple, but give the adjunctions from Sections 4.1 and 4.2. In particular, quantifiers emerge as two universal ways to remove variables from predicates.

Section 7.3 relates power sets to presheaf categories. Just as the Yoneda embedding gives free cocompletion, the power set freely cocompletes a set as a poset.

Section 7.4 explains the relation–adjunction correspondence from Section 4.4 through universal adjunctions, or nerve and realization. It also reveals why I kept choosing sheaf adjunctions as examples.

7.1 The pointwise formula and power-set-valued Kan extensions

For a small category $\mathcal{C}$, a locally small category $\mathcal{D}$, a complete category $\mathcal{E}$, and functors $K : \mathcal{C} \rightarrow \mathcal{D}$, $F : \mathcal{C} \rightarrow \mathcal{E}$, the right Kan extension of F along K

$$\begin{array}{ccc} \mathcal{C} & \xrightarrow{F} & \mathcal{E} \\ & \searrow K \quad \uparrow \quad \nearrow \text{Ran}_K F & \\ & \mathcal{D} & \end{array}$$

exists and is pointwise: it is given by

$$(\text{Ran}_K F)(d) = \lim(d \downarrow K \rightarrow \mathcal{C} \rightarrow \mathcal{E})$$

this formula.⁷³ The dual statement holds as well. For more detail, see, for example, Theorem 6.2.1, Corollary 6.2.6, and Theorem 6.3.7 of [1].

Truth values $\mathbf{2}$ and power sets $P(X)$ are bicomplete by Propositions 3.1.2 and 3.2.1. General theory therefore immediately gives the existence and construction of their pointwise Kan extensions. We will use this repeatedly, so let us record it. Below, $\bigcap \mathcal{F}$ and $\bigcup \mathcal{F}$ mean $\bigcap_{A \in \mathcal{F}} A$ and $\bigcup_{A \in \mathcal{F}} A$.

⁷³Each choice of limits gives a right Kan extension; different choices can differ by a canonical isomorphism.

Lemma 7.1.1 (Kan extensions into a power set). For a set X , small $\mathcal{C}$, locally small $\mathcal{D}$, and functors $F : \mathcal{C} \rightarrow P(X)$, $K : \mathcal{C} \rightarrow \mathcal{D}$, the pointwise right Kan extension $\text{Ran}_K F$

$$\begin{array}{ccc} \mathcal{C} & \xrightarrow{F} & P(X) \\ & \searrow K & \uparrow \parallel \\ & & \mathcal{D} \end{array} \quad \begin{array}{c} \nearrow \text{Ran}_K F \end{array}$$

and the pointwise left Kan extension $\text{Lan}_K F$

$$\begin{array}{ccc} \mathcal{C} & \xrightarrow{F} & P(X) \\ & \searrow K & \downarrow \parallel \\ & & \mathcal{D} \end{array} \quad \begin{array}{c} \nearrow \text{Lan}_K F \end{array}$$

both exist. At each $d \in \text{ob}(\mathcal{D})$, they are given by

$$\begin{aligned} \text{Ran}_K F(d) &= \bigcap \{Fc \mid c \in \text{ob}(\mathcal{C}), \exists f : d \rightarrow Kc\} \\ \text{Lan}_K F(d) &= \bigcup \{Fc \mid c \in \text{ob}(\mathcal{C}), \exists f : Kc \rightarrow d\} \end{aligned}$$

these formulas.

Proof. Existence was explained above. The formulas follow because limits and colimits in a power set are intersections and unions (Proposition 3.1.1). $\square$

Remark 7.1.2 (The action on morphisms and the natural transformation). The lemma does not mention the action on morphisms. But between two objects of a poset there is at most one morphism, so there is no choice. A Kan extension also includes a natural transformation; with a poset as codomain, this is just an objectwise inequality, again with no choice. Thus the lemma gives all the data that need specifying.

This does not mean that functoriality and natural transformations are irrelevant and should be ignored. The lemma guarantees functoriality (order preservation when $\mathcal{D}$ is a poset) and the existence of the natural transformation, facts we will certainly use.

Remark 7.1.3 (The role of the morphisms of $\mathcal{C}$). The role of morphisms in $\mathcal{C}$ is limited. The formulas for power-set-valued Kan extensions do not use them. They appear only as constraints on K and F being functorial.

In a poset, every diagram with matching endpoints commutes. Thus limits and colimits use only suprema or infima of objects, not the arrows of the indexing diagram. Naturally, this also appears in Kan extension computations, since those use limits and colimits.

Example 7.1.4 (Proof of Lemma 5.2.8). The left adjoint of precomposition is left Kan extension, so to prove Lemma 5.2.8, compute

$$\begin{array}{ccc} \text{ob}(P(X)) & \xrightarrow{d} & P(X) \\ & \searrow \iota & \downarrow \parallel \\ & & P(X) \end{array} \quad \begin{array}{c} \nearrow \text{Lan}_\iota d \end{array}$$

this left Kan extension. Substituting in Lemma 7.1.1 gives the result immediately.

Remark 7.1.5 (A toy example of the pointwise formula itself). Everyone may have their own intuition for the pointwise formula, but the lemma offers a good one. More precisely, the intuition applies to posets $\mathcal{C}, \mathcal{D}, \mathcal{E}$ with $\mathcal{E}$ complete.⁷⁴ To turn an object of $\mathcal{D}$ into one of $\mathcal{E}$, first approximate it from above using objects of $\mathcal{C}$ via K , then take an infimum in $\mathcal{E}$.

As a slight digression, this viewpoint of one-sided approximation followed by a limit is particularly clear when extending $2^x : \mathbb{Q} \rightarrow \mathbb{R}$ to $2^x : \mathbb{R} \rightarrow \mathbb{R}$ via Kan extensions. For more detail, see, for example, Example 6.2.7. of [1].

⁷⁴The explicit formula looks just like the lemma.

7.2 Truth-value-valued Kan extensions, inverse image, and quantifiers

The direct-image–inverse-image adjunctions were the two adjoints of precomposition. Such adjoints define global Kan extensions. For more detail, see, for example, Proposition 6.1.5 and the sentence following it of [1]. Thus direct and dual images can be written as Kan extensions.

Lemma 7.1.1 also applies to truth-value-valued Kan extensions, since the truth values are the power set of a singleton. Replace intersections and unions by conjunctions and disjunctions (Proposition 3.2.1).

The direct-image–inverse-image adjunctions are the case where K in Lemma 7.1.1 is simply a map of sets.

Proposition 7.2.1 (Direct and dual images as Kan extensions). For a map $f : X \rightarrow Y$ and $V \subset X$, the characteristic function of the dual image $f_!(V)$ is the right Kan extension of the characteristic function of V along f ,

$$\begin{array}{ccc} X & \xrightarrow{V} & \mathbf{2} \\ & \searrow f & \uparrow \parallel \\ & & Y \end{array} \quad \text{Ran}_f V = f_!(V)$$

and the characteristic function of the direct image $f_!(V)$ is the left Kan extension along f ,

$$\begin{array}{ccc} X & \xrightarrow{V} & \mathbf{2} \\ & \searrow f & \downarrow \parallel \\ & & Y \end{array} \quad \text{Lan}_f V = f_!(V)$$

Of course, this also gives explicit formulas. Suppose you did not know $f_!$ and wanted the dual of direct image, or the right adjoint of inverse image. Kan extensions give a quick answer: not just existence, but the construction itself.

Quantifiers are a special case. Left and right Kan extensions are two dual universal ways to change the domain of a functor along another functor. From this perspective, the following says that quantifiers are two dual universal ways to reduce the number of variables of a predicate.

Corollary 7.2.2 (Quantifiers as Kan extensions). For sets X, Y and a predicate $\phi : X \times Y \rightarrow \mathbf{2}$, $\forall_X \phi$ is the right Kan extension of ϕ along the projection $p : X \times Y \rightarrow Y$,

$$\begin{array}{ccc} X \times Y & \xrightarrow{\phi} & \mathbf{2} \\ & \searrow p & \uparrow \parallel \\ & & Y \end{array} \quad \text{Ran}_p \phi = \forall_X \phi$$

and $\exists_X \phi$ is its left Kan extension along the same projection,

$$\begin{array}{ccc} X \times Y & \xrightarrow{\phi} & \mathbf{2} \\ & \searrow p & \downarrow \parallel \\ & & Y \end{array} \quad \text{Lan}_p \phi = \exists_X \phi$$

Remark 7.2.3 (Analogues elsewhere in mathematics). Recall the precomposition functors from the section on categories, functors, and natural transformations. For restriction of representations $\text{res}_H^G : \text{Vect}_{\mathbb{K}}^G \rightarrow \text{Vect}_{\mathbb{K}}^H$, the pointwise formulas give two dual universal ways to turn a representation of a subgroup into one of the whole group, and tell us exactly how to construct them.⁷⁵ These constructions have names such as induced representations.

⁷⁵We do need to know limits and colimits of vector spaces. This is where basic categorical theorems let us replace them by limits and colimits in better-understood categories.

Similarly, for $\text{Dyn}_{\mathcal{C}}^{\text{inv}} \rightarrow \text{Dyn}_{\mathcal{C}}$, if $\mathcal{C}$ is bicomplete, the formulas give two dual universal ways to turn a possibly noninvertible dynamical system into an invertible one, together with explicit constructions.

The global definitions of limits and colimits⁷⁶ have the same form too: they are adjoints to precomposition with the unique functor to the terminal category.

7.3 Yoneda and power sets as free cocompletions

Here we describe a poset analogue of Yoneda.⁷⁷ The main statements stay with set operations; remarks discuss general posets.

The counterpart of the Yoneda embedding simply sends each element to its singleton, embedding a set in its power set.

Definition 7.3.1 (Yoneda). For a set X , write y_X for the map

$$X \rightarrow P(X) : x \mapsto \{x\}$$

Remark 7.3.2 (The definition for posets, and why it makes sense). The Yoneda embedding is the currying of $\text{Hom} : \mathcal{C}^{\text{op}} \times \mathcal{C} \rightarrow \mathbf{Set}$ to $\mathcal{C} \rightarrow \mathbf{Set}^{\mathcal{C}^{\text{op}}}$. Replace $\mathbf{Set}$ by $\mathbf{2}$. For a preorder Q , call the currying $Q \rightarrow \mathbf{2}^{Q^{\text{op}}}$ of $\leq : Q^{\text{op}} \times Q \rightarrow \mathbf{2}$ its Yoneda embedding, denoted y_Q . The presheaf counterpart $\mathbf{2}^{Q^{\text{op}}}$ is naturally isomorphic to the poset of downward-closed subsets of Q under inclusion. The embedding sends q to $\{p \in Q \mid p \leq q\}$. For a set with the trivial order, this agrees with the preceding definition.

Remark 7.3.3 (The Yoneda lemma for posets). For a poset Q , its Yoneda lemma says that, for $q \in Q$ and a downward-closed subset D , remembering that morphisms are inequalities,

$$\{q\} \subset D \iff q \in D$$

holds.⁷⁸

The Yoneda embedding is also characterized as the universal way to make a category cocomplete. A functor $F : \mathcal{C} \rightarrow \mathcal{D}$ from a small category to a cocomplete one factors essentially uniquely, up to natural isomorphism, through $y_{\mathcal{C}}$ followed by a cocontinuous functor $G : \mathbf{Set}^{\mathcal{C}^{\text{op}}} \rightarrow \mathcal{D}$. Moreover, G is the left Kan extension of F along Yoneda.

$$\begin{array}{ccc} \mathcal{C} & \xrightarrow{F} & \mathcal{D} \\ & \searrow y_{\mathcal{C}} & \downarrow \parallel \\ & & \mathbf{Set}^{\mathcal{C}^{\text{op}}} \\ & & \nearrow G = \text{Lan}_{y_{\mathcal{C}}} F \end{array}$$

See <https://ncatlab.org/nlab/show/free+cocompletion>.

The analogous fact for y_X characterizes the power set as a poset.

Proposition 7.3.4 (The power set as free cocompletion). For a set X , a complete lattice⁷⁹ Q , and a map $f : X \rightarrow Q$, there is a unique cocontinuous functor (supremum-preserving map) $g : P(X) \rightarrow Q$ making

$$\begin{array}{ccc} X & \xrightarrow{f} & Q \\ & \searrow y_X & \nearrow g \\ & & P(X) \end{array}$$

⁷⁶Here the explicit formula offers no immediate benefit, at least as far as I can see—I hesitate to be categorical about it!—because it only rewrites a limit as the very same limit. Expressing universal constructions through limits or colimits becomes useful when basic categorical theorems have made those limits or colimits understandable.

⁷⁷As of March 2021, I hope truth-value enrichment puts this in literally the same framework. In that case, I suspect we should really use preorders.

⁷⁸Viewed as a functor to $\mathbf{2}$, the “value” of D at q is the condition $q \in D$.

⁷⁹It suffices to assume all suprema, or colimits: all infima then follow automatically.

commute.⁸⁰ Moreover, $g = \text{Lan}_{y_X} f$, and hence, for every $V \subset X$,

$$g(V) = \sup_{x \in V} f(x)$$

holds.

Proof. Uniqueness follows because every element of $P(X)$ is a colimit of elements of X . It is also easy to check directly that $\text{Lan}_{y_X} f$ has the required properties.⁸¹ The last formula is the pointwise left Kan extension formula. $\square$

We discuss the related universal adjunction next.

Remark 7.3.5 (Free cocompletion of a poset). This holds for general posets R , not just trivially ordered sets. Every order-preserving map $f : R \rightarrow Q$ to a complete lattice factors uniquely through $y_R : R \rightarrow \mathbf{2}^R$ followed by a cocontinuous order-preserving map.

Remark 7.3.6 (density theorem). Recall the density theorem from category theory. A set-valued functor is canonically a colimit of representables. For more detail, see, for example, Theorems 6.5.7 and 6.5.8 of [1].

Its counterpart for set operations is obvious: for every $V \subset X$,

$$V = \bigcup_{x \in V} \{x\}$$

holds. Notice that the category of elements corresponds to V .

Of course this equality is obvious, but that is precisely why it is a useful toy example. In the density theorem, we reconstruct a functor by collecting its representable “constituents.” For sets, we literally collect the elements of a subset to reconstruct that subset.

We will mention analogues elsewhere in mathematics in the next subsection.

7.4 Universal adjunctions and Galois connections from relations, II

Let us realize the mutually right adjoints induced by a binary relation as a universal adjunction, or nerve and realization. Earlier, we constructed them explicitly using elements, “naturally extending” a relation to power sets. Now we will see where that idea comes from more generally.

Proposition 7.4.1 (The universal adjunction for a power set). For a set X , a complete lattice Q , and a map $f : X \rightarrow Q$, the map $g = \text{Lan}_{y_X} f$ from Proposition 7.3.4 has

$$Q(f(-), -) : Q \rightarrow P(X) : q \mapsto \{x \in X \mid f(x) \leq q\}$$

as a right adjoint.

Proof. For any $q \in Q$ and $V \in P(X)$,

$$\begin{aligned} g(V) \leq q &\iff \sup_{x \in V} f(x) \leq q \\ &\iff \forall x \in V, f(x) \leq q \\ &\iff V \subset Q(f(-), q) \end{aligned}$$

as required. $\square$

⁸⁰We restrict to complete lattices, rather than cocomplete preorders—perhaps not ideally—so uniqueness is strict here.

⁸¹As in the general case, there is an abstract argument: full faithfulness of Yoneda and pointwiseness give commutativity, while the universal adjunction in the next subsection gives a right adjoint, hence cocontinuity.

This proof runs exactly parallel to the proof of the categorical universal adjunction. Compare the chain of isomorphisms in Remark 6.5.9 of [1].⁸²

Remark 7.4.2 (For posets). The proposition holds for general posets, not just trivially ordered sets, with the same proof.

Typical universal adjunctions include our familiar sheaf adjunction and geometric realization of simplicial sets. See <https://ncatlab.org/nlab/show/nerve+and+realization>.

Remark 7.4.3 (The sheaf adjunction). For a topological space X , consider the poset $\mathcal{O}(X)$ of open subsets. Let $\iota_X : \mathcal{O}(X) \rightarrow \mathbf{Top}/X$ send U to its inclusion $U \rightarrow X$. Its universal adjunction is

$$\begin{array}{ccc} & L & \\ \mathbf{Top}/X & \perp & \mathbf{Set}^{\mathcal{O}(X)^{\mathrm{op}}} \\ & \Gamma & \end{array}$$

The presheaf category is the free cocompletion of $\mathcal{O}(X)$. Canonically extending the functor viewing open sets as bundles over X to this free cocompletion gives sheafification. A free cocompletion consists, so to speak, of formal colimits: “blueprints” specifying which open sets to arrange and how to glue them.⁸³ The right adjoint takes sections over open sets. Again this is general: the nerve functor is the Yoneda embedding with some information forgotten. Ignoring size issues, a functor $F : \mathcal{C} \rightarrow \mathcal{D}$ gives $\mathcal{D} \xrightarrow{y_{\mathcal{D}}} \mathbf{Set}^{\mathcal{D}^{\mathrm{op}}} \xrightarrow{F^*} \mathbf{Set}^{\mathcal{C}^{\mathrm{op}}}$, the right adjoint in nerve and realization. Yoneda considers all relations to objects of $\mathcal{D}$; here we consider only relations to objects of $\mathcal{C}$ through F .

Remark 7.4.4 (Geometric realization). Similarly, consider the adjunction between topological spaces and simplicial sets. Let Δ be the category of nonempty finite ordered sets and order-preserving maps, and $\iota : \Delta \rightarrow \mathbf{Top}$ the functor sending n to the standard n -simplex. Its universal adjunction has right adjoint $\mathbf{Top}(\iota(-), -) : \mathbf{Top} \rightarrow \mathbf{Set}^{\Delta^{\mathrm{op}}}$, taking a space to its simplicial set of singular simplices.⁸⁴ The left adjoint, as always, realizes a blueprint: view a simplicial set as a formal⁸⁵ colimit and glue standard simplices accordingly. This is geometric realization.

Thus a universal adjunction extends a functor $\mathcal{C} \rightarrow \mathcal{D}$ to the free cocompletion $\mathbf{Set}^{\mathcal{C}^{\mathrm{op}}}$, obtained by “naturally adjoining all small colimits” to $\mathcal{C}$. Our recurring mutually right adjoints are what happens when this method extends the functor given by a binary relation.

So far we viewed a relation as $X \times Y \rightarrow \mathbf{2}$; now view it as $X \times Y \rightarrow \mathbf{2}^{\mathrm{op}}$.⁸⁶ Currying gives $X \rightarrow (\mathbf{2}^{\mathrm{op}})^Y$, or $X \rightarrow P(Y)^{\mathrm{op}}$. Its universal adjunction in the sense of Proposition 7.4.1 is exactly the mutually right adjunction induced by the relation.

Theorem 7.4.5 (The adjunction from a relation is a universal adjunction). For a relation $X \times Y \rightarrow \mathbf{2}^{\mathrm{op}}$, the universal adjunction of its currying $X \rightarrow P(Y)^{\mathrm{op}}$, as in Proposition 7.4.1, agrees with the mutually right adjunction of Proposition 4.4.3.

Proof. It suffices to check one side of the adjunction. Either the left adjoint, using the pointwise formula, or the right adjoint is an almost direct restatement of the definitions in Proposition 4.4.3. Remember that the order is reversed in $P(Y)^{\mathrm{op}}$. $\square$

⁸²I have omitted one corresponding line, because the step seems obvious for set operations.

⁸³This is one general viewpoint on nerve and realization.

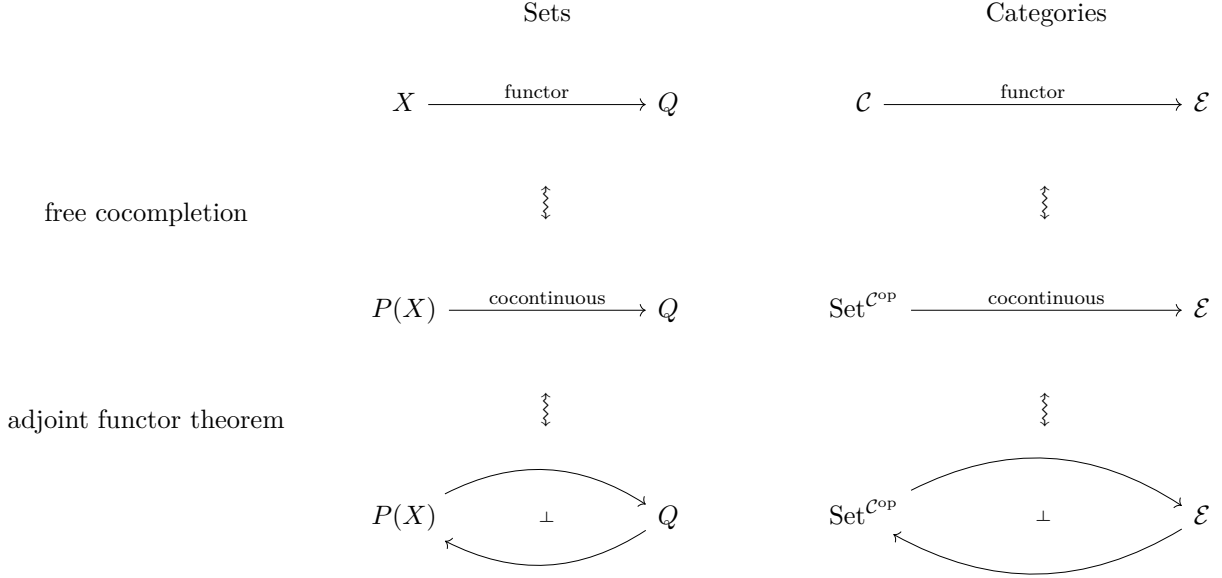
⁸⁴Singular homology factors through this functor, up to natural isomorphism: it is defined on $\mathbf{Set}^{\Delta^{\mathrm{op}}}$, and spaces acquire their singular homology by first being viewed as simplicial sets.

⁸⁵It really is a colimit of representables; I am only using the language of blueprints.

⁸⁶Partly because I want the resulting adjunction to look symmetric in X and Y .

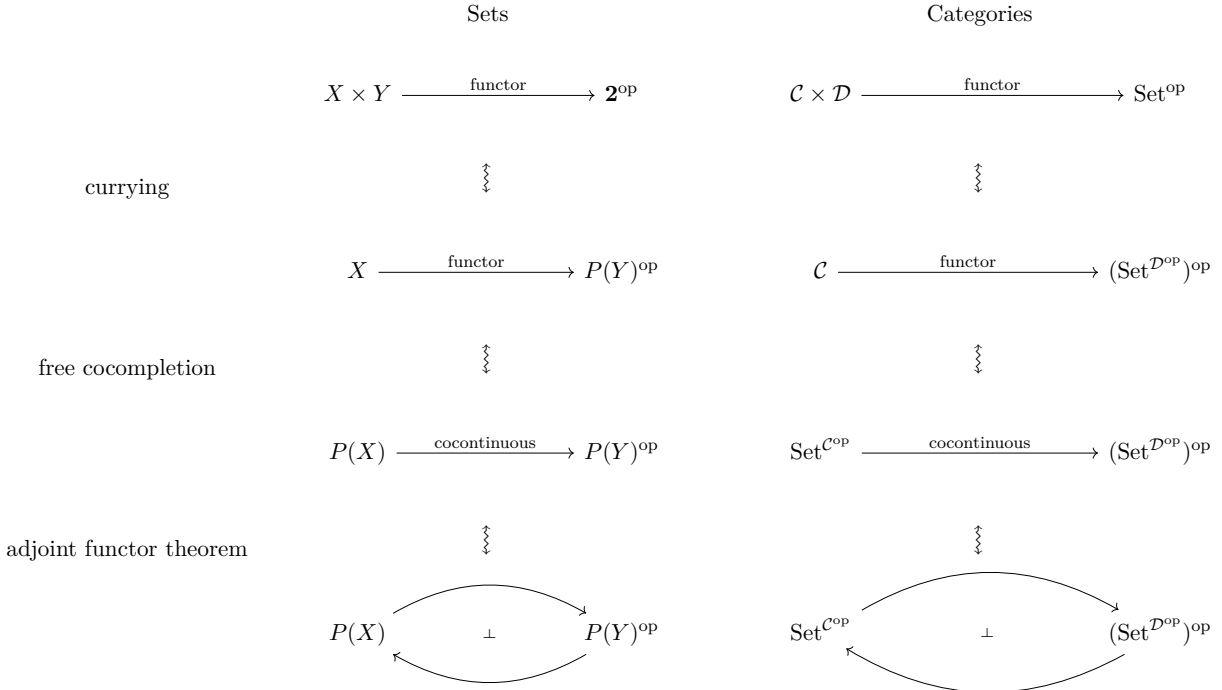
Remark 7.4.6 (The relation–adjunction correspondence and its categorical counterpart). As this subsection and Section 7.3 show, maps $X \rightarrow Q$, cocontinuous functors $P(X) \rightarrow Q$, and left adjoints $P(X) \rightarrow Q$ correspond bijectively when Q is a complete lattice.⁸⁷ In parallel, for small $\mathcal{C}$ and cocomplete $\mathcal{E}$, functors $\mathcal{C} \rightarrow \mathcal{E}$, cocontinuous functors $\text{Set}^{\mathcal{C}^{\text{op}}} \rightarrow \mathcal{E}$, and left adjoints with that source and target correspond up to natural isomorphism.⁸⁸

Here is a summary of the correspondences.



Our remarks that sheaf adjunctions and geometric realization are determined by relatively little data, a functor from a small category, illustrate the right-hand side.

For the relation–adjunction correspondence, take Q to be a power set. Including currying gives the following correspondences.



⁸⁷Restriction or the adjoint functor theorem gives the reverse correspondences.

⁸⁸The diagram below says “adjoint functor theorem,” but honestly I proved this using nerve and realization. To use an adjoint functor theorem, the nLab suggests the theorem for total categories: <https://ncatlab.org/nlab/show/adjoint+functor+theorem#statement>.

We now see that the correspondence between relations and mutually right adjoints between power sets is built from basic categorical correspondences.

You might wonder whether the categorical side has interesting applications. At an elementary level, it determines self-dualities, meaning mutually right adjoints to themselves, of presheaf categories: directed graphs, discrete dynamical systems on sets, or even $\mathbf{Set}$ itself. The adjunction corresponding to the dual $\mathcal{C} \times \mathcal{C}^{\text{op}} \rightarrow \mathbf{Set}^{\text{op}}$ of the hom-functor $\mathcal{C}(-, -) : \mathcal{C}^{\text{op}} \times \mathcal{C} \rightarrow \mathbf{Set}$ is Isbell duality: <https://ncatlab.org/nlab/show/Isbell+duality>.

8 Things I want to write about but have not yet

- Cartesian functors and filters, ideals, and ultrafilters.
- What holds in a topos.
- The **2**-enriched viewpoint.
- A table of toy examples.

References

- [1] Emily Riehl, *CATEGORY THEORY IN CONTEXT*, DOVER, 2016.
- [2] Saunders Mac Lane, *Categories for the Working Mathematician*, Japanese translation, Maruzen, 1998; original edition, 1971.
- [3] nLab <https://ncatlab.org/nlab/show/HomePage>
- [4] Saunders Mac Lane, Ieke Moerdijk, *Sheaves in Geometry and Logic A first Introduction to Topos Theory*, Springer, 1994.
- [5] Peter Smith, The Galois Connection between Syntax and Semantics, <https://www.logicmatters.net/resources/pdfs/Galois.pdf>.
- [6] *A General Theory of Generation and Reflective Subcategories* (in Japanese), https://hora-algebra.github.io/gen_ref.pdf.
- [7] *The Stone Problem* (in Japanese), <https://hora-algebra.github.io/%E7%9F%B3%E3%81%AE%E5%95%8F%E9%A1%8C.pdf>.
- [8] Lev Pontryagin, *Topological Groups*, Princeton University Press, 1939.
- [9] M.F. Atiyah and I.G. MacDonald, *Introduction to Commutative Algebra*, 1969.
- [10] MATTHEW DIRKS, *THE STONE REPRESENTATION THEOREM FOR BOOLEAN ALGEBRAS*, 2011.
- [11] Peter T. Johnstone, *Sketches of an Elephant A Topos Theory Compendium*, Oxford University Press, 2002.
- [12] Kenneth Kunen, *The Foundations of Mathematics*, Japanese translation, Nippon Hyoron Sha, 2016.
- [13] Kenneth Kunen, *Set Theory*, College Publications, 2011.
- [14] Richard Garner, *Ionads*, <https://arxiv.org/abs/0912.1415>, 2009.