1 Generative Effects: Orders and Galois Connections

In this book, we explore a wide variety of situations – in the world of science, engineering, and commerce – where we see something we might call *compositionality*. These are cases in which systems or relationships can be combined to form new systems or relationships. In each case we find category-theoretic constructs – developed for their use in pure math – which beautifully describe the compositionality of the situation.

This chapter, being the first of the book, must serve this goal in two capacities. First, it must provide motivating examples of compositionality, as well as the relevant categorical formulations. Second, it must provide the mathematical foundation for the rest of the book. Since we are starting with minimal assumptions about the reader's background, we must begin slowly and build up throughout the book. As a result, examples in the early chapters are necessarily simplified. However, we hope the reader will already begin to see the sort of structural approach to modeling that category theory brings to the fore.

1.1 More Than the Sum of Their Parts

We motivate this first chapter by noticing that while many real-world structures are compositional, the results of observing them are often not. The reason is that observation is inherently "lossy": in order to extract information from something, one must drop the details. For example, one stores a real number by rounding it to some precision. But if the details are actually relevant in a given system operation, then the observed result of that operation will not be as expected. This is clear in the case of roundoff error, but it also shows up in non-numerical domains: observing a complex system is rarely enough to predict its behavior because the observation is lossy.

A central theme in category theory is the study of structures and structure-preserving maps. A map $f : X \to Y$ is a kind of observation of object X via a specified relationship it has with another object, Y. For example, think of X as the subject of an experiment and Y as a meter connected to X, which allows us to extract certain features of X by looking at the reaction of Y.

Asking which aspects of X one wants to preserve under the observation f becomes the question "what category are you working in?" As an example, there are many functions f from $\mathbb{R}$ to $\mathbb{R}$ (where $\mathbb{R}$ is the set of real numbers), and we can think of them as observations: rather than view x "directly," we only observe $f(x)$. Out of all the

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

functions $f : \mathbb{R} \to \mathbb{R}$, only some of them preserve the order of numbers, only some of them preserve the distance between numbers, only some of them preserve the sum of numbers, etc. Let's check in with an exercise; a solution can be found in the Appendix.

Exercise 1.1. Some terminology: a function $f : \mathbb{R} \to \mathbb{R}$ is said to be

(a) *order-preserving* if $x \leq y$ implies $f(x) \leq f(y)$, for all $x, y \in \mathbb{R}$;[1]
(b) *metric-preserving* if $|x - y| = |f(x) - f(y)|$;
(c) *addition-preserving* if $f(x + y) = f(x) + f(y)$.

For each of the three properties defined above – call it *foo* – find an f that is *foo*-preserving and an example of an f that is not *foo*-preserving. ◇

In category theory we want to keep control over which aspects of our systems are being preserved under various observations. As we said above, the less structure is preserved by our observation of a system, the more "surprises" occur when we observe its operations. One might call these surprises *generative effects*.

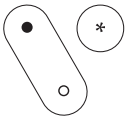
In using category theory to explore generative effects, we follow the basic ideas from work by Adam [Ada17]. He goes much more deeply into the issue than we can here; see Section 1.5. But as mentioned above, we must also use this chapter to give an order-theoretic warm-up for the full-fledged category theory to come.

1.1.1 A First Look at Generative Effects

To explore the notion of a generative effect we need a sort of system, a sort of observation, and a system-level operation that is not preserved by the observation. Let's start with a simple example.

A simple system

Consider three points; we'll call them $\bullet$, $\circ$, and $*$. In this example, a *system* will simply be a way of connecting these points together. We might think of our points as sites on a power grid, with a system describing connection by power lines, or as people susceptible to some disease, with a system describing interactions that can lead to contagion. As an abstract example of a system, there is a system where $\bullet$ and $\circ$ are connected, but neither is connected to $*$. We shall draw this like so:



[1] We are often taught to view functions $f : \mathbb{R} \to \mathbb{R}$ as plots in the (x, y)-coordinate system, where x is the domain (independent) variable and y is the codomain (dependent) variable. In this book, we do not adhere to that naming convention; e.g. in Example 1.1, both x and y are being "plugged in" as input to f. As an example consider the function $f(x) = x^2$. Then f being order-preserving would say that, for any $x, y \in \mathbb{R}$, if $x \leq y$ then $x^2 \leq y^2$; is that true?

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

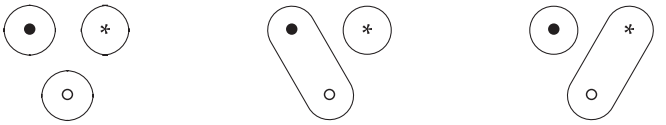
Connections are symmetric, so if a is connected to b, then b is connected to a. Connections are also transitive, meaning that if a is connected to b, and b is connected to c, then a is connected to c; that is, all a, b, and c are connected. Friendship is not transitive – my friend's friend is not necessarily my friend – but possible communication of a concept or a disease is.

Here we depict two more systems, one in which none of the points are connected, and one in which all three points are connected.

There are five systems in all, and we depict them below.

Now that we have defined the sort of system we want to discuss, suppose that Alice is observing this system. Her observation of interest, which we call Φ, extracts a single feature from a system, namely whether the point $\bullet$ is connected to the point $*$; this is what she wants to know. Her observation of the system will be an assignment of either `true` or `false`; she assigns `true` if $\bullet$ is connected to $*$, and `false` otherwise. So Φ assigns the value `true` to the following two systems:

and Φ assigns the value `false` to the three remaining systems:



(1.1)

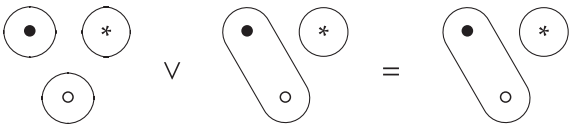
The last piece of setup is to give a sort of operation that Alice wants to perform on the systems themselves. It's a very common operation – one that will come up many times throughout the book – called *join*. If the reader has been following the story arc, the expectation here is that Alice's connectivity observation will not be compositional with respect to the operation of system joining; that is, there will be generative effects. Let's see what this means.

Joining our simple systems

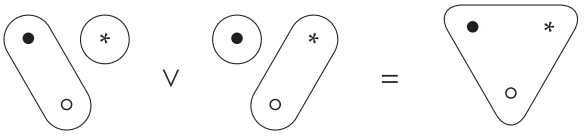
Joining two systems A and B is performed simply by combining their connections. That is, we shall say the *join* of systems A and B, denoted $A \vee B$, has a connection between points x and y if there are some points $z_1, \ldots, z_n$ such that each of the following is true in at least one of A or B: x is connected to z_1, z_i is connected to z_{i+1}, and z_n is connected to y. In a three-point system, the above definition is overkill, but we want to say something that works for systems with any number of elements. The high-level way

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

to say it is "take the transitive closure of the union of the connections in A and B." In our three-element system, it means for example that

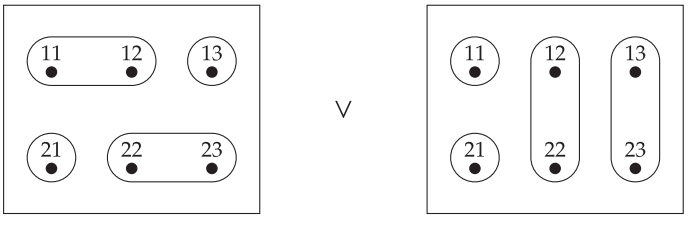


and



(1.2)

Exercise 1.2. What is the result of joining the following two systems?



◇

We are now ready to see the generative effect. We don't want to build it up too much – this example has been made as simple as possible – but we shall see that Alice's observation fails to preserve the join operation. We've been denoting her observation – measuring whether $\bullet$ and $*$ are connected – by the symbol Φ; it returns a boolean result, either `true` or `false`.

We see above in Eq. (1.1) that $\Phi($  $) = \Phi($  $) =$ `false`: in both cases $\bullet$ is not connected to $*$. On the other hand, when we join these two systems as in Eq. (1.2), we see that $\Phi($  $\vee$  $) = \Phi($ $) =$ `true`: in the joined system, $\bullet$ *is* connected to $*$. The question that Alice is interested in, that of Φ, is inherently lossy with respect to join, and there is no way to fix it without a more detailed observation, one that includes not only $*$ and $\bullet$ but also $\circ$.

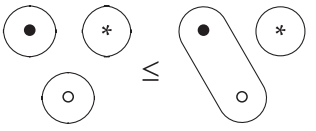
While this was a simple example, it should be noted that whether the potential for such effects exist – i.e. determining whether an observation is operation-preserving – can be incredibly important information to know. For example, Alice could be in charge of putting together the views of two local authorities regarding possible contagion between an infected person $\bullet$ and a vulnerable person $*$. Alice has noticed that if they separately extract information from their raw data and combine the results, it gives a different answer than if they combine their raw data and extract information from it.

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

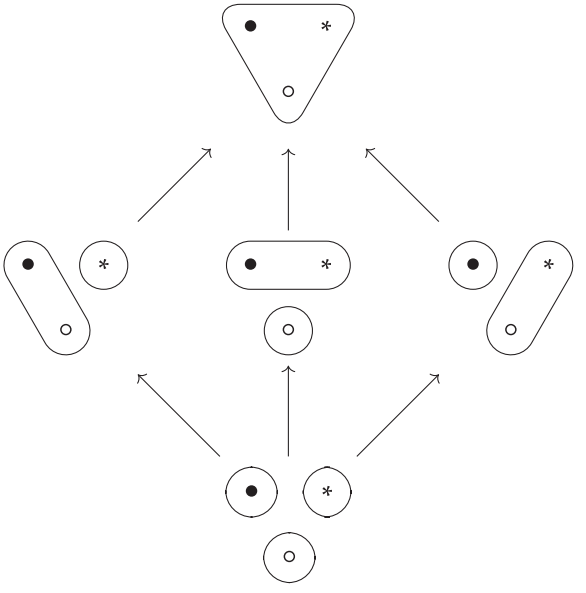
1.1.2 Ordering Systems

Category theory is all about organizing and layering structures. In this section we will explain how the operation of joining systems can be derived from a more basic structure: order. We shall see that while joining is not preserved by Alice's connectivity observation Φ, order is.

To begin, we note that the systems themselves are ordered in a hierarchy. Given systems A and B, we say that $A \leq B$ if, whenever x is connected to y in A, then x is connected to y in B. For example,



This notion of $\leq$ leads to the following diagram:



(1.3)

where an arrow from system A to system B means $A \leq B$. Such diagrams are known as *Hasse diagrams*.

As we were saying above, the notion of join is derived from this order. Indeed, for any two systems A and B in the Hasse diagram (1.3), the joined system $A \vee B$ is the smallest system that is bigger than both A and B. That is, $A \leq (A \vee B)$ and $B \leq (A \vee B)$, and for any C, if $A \leq C$ and $B \leq C$ then $(A \vee B) \leq C$. Let's walk through this with an exercise.

Exercise 1.3.

1. Write down all the partitions of a two-element set $\{\bullet, *\}$, order them as above, and draw the Hasse diagram.

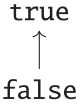
https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

2. Now do the same thing for a four-element set, say $\{1, 2, 3, 4\}$. There should be 15 partitions.

Choose any two systems in your 15-element Hasse diagram, call them A and B.

3. What is $A \vee B$, using the definition given in the paragraph above Eq. (1.2)?
4. Is it true that $A \leq (A \vee B)$ and $B \leq (A \vee B)$?
5. What are all the systems C for which both $A \leq C$ and $B \leq C$?
6. Is it true that in each case $(A \vee B) \leq C$? ◇

The set $\mathbb{B} = \{\texttt{true}, \texttt{false}\}$ of booleans also has an order, $\texttt{false} \leq \texttt{true}$:



Thus $\texttt{false} \leq \texttt{false}$, $\texttt{false} \leq \texttt{true}$, and $\texttt{true} \leq \texttt{true}$, but $\texttt{true} \not\leq \texttt{false}$. In other words, $A \leq B$ if A implies B.[2]

For any A, B in $\mathbb{B}$, we can again write $A \vee B$ to mean the least element that is greater than both A and B.

Exercise 1.4. Using the order $\texttt{false} \leq \texttt{true}$ on $\mathbb{B} = \{\texttt{true}, \texttt{false}\}$, what is:

1. $\texttt{true} \vee \texttt{false}$?
2. $\texttt{false} \vee \texttt{true}$?
3. $\texttt{true} \vee \texttt{true}$?
4. $\texttt{false} \vee \texttt{false}$? ◇

Let's return to our systems with $\bullet$, $\circ$, and $*$, and Alice's "$\bullet$ is connected to $*$" function, which we called Φ. It takes any such system and returns either $\texttt{true}$ or $\texttt{false}$. Note that the map Φ preserves the $\leq$ order: if $A \leq B$ and there is a connection between $\bullet$ and $*$ in A, then there is such a connection in B too. The possibility of a generative effect is captured in the inequality

$$\Phi(A) \vee \Phi(B) \leq \Phi(A \vee B). \tag{1.4}$$

We saw on page 4 that this can be a strict inequality: we showed two systems A and B with $\Phi(A) = \Phi(B) = \texttt{false}$, so $\Phi(A) \vee \Phi(B) = \texttt{false}$, but where $\Phi(A \vee B) = \texttt{true}$. In this case, a generative effect exists.

These ideas capture the most basic ideas in category theory. Most directly, we have seen that the map Φ preserves some structure but not others: it preserves order but not join. In fact, we have seen here hints of more complex notions from category theory, without making them explicit; these include the notions of category, functor, colimit, and adjunction. In this chapter we will explore these ideas in the elementary setting of ordered sets.

[2] In mathematical logic, $\texttt{false}$ implies $\texttt{true}$ but $\texttt{true}$ does not imply $\texttt{false}$. That is "P implies Q" means, "if P is true, then Q is true too, but if P is not true, I'm making no claims."

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

1.2 What is Order?

Above we informally spoke of two different ordered sets: the order on system connectivity and the order on booleans `false` $\leq$ `true`. Then we related these two ordered sets by means of Alice's observation Φ. Before continuing, we need to make such ideas more precise. We begin in Section 1.2.1 with a review of sets and relations. In Section 1.2.2 we will give the definition of a *preorder* – short for preordered set – and a good number of examples.

1.2.1 Review of Sets, Relations, and Functions

We will not give a definition of *set* here, but informally we will think of a set as a collection of things, known as elements. These things could be all the leaves on a certain tree, or the names of your favorite fruits, or simply some symbols a, b, c. For example, we write $A = \{h, 1\}$ to denote the set, called A, that contains exactly two elements, one called h and one called 1. The set $\{h, h, 1, h, 1\}$ is exactly the same as A because they both contain the same elements, h and 1, and repeating an element more than once in the notation doesn't change the set.[3] For an arbitrary set X, we write $x \in X$ if x is an element of X; so we have $h \in A$ and $1 \in A$, but $0 \notin A$.

Example 1.5. Here are some important sets from mathematics – and the notation we will use – that will appear again in this book.

- $\varnothing$ denotes the empty set; it has no elements.
- $\{1\}$ denotes a set with one element; it has one element, 1.
- $\mathbb{B}$ denotes the set of *booleans*; it has two elements, `true` and `false`.
- $\mathbb{N}$ denotes the set of *natural numbers*; it has elements $0, 1, 2, 3, \ldots, 90^{717}, \ldots$.
- $\underline{n}$, for any $n \in \mathbb{N}$, denotes the nth *ordinal*; it has n elements $1, 2, \ldots, n$. For example, $\underline{0} = \varnothing$, $\underline{1} = \{1\}$, and $\underline{5} = \{1, 2, 3, 4, 5\}$.
- $\mathbb{Z}$, the set of *integers*; it has elements $\ldots, -2, -1, 0, 1, 2, \ldots, 90^{717}, \ldots$.
- $\mathbb{R}$, the set of *real numbers*; it has elements like $\pi, 3.14, 5 * \sqrt{2}, e, e^2, -1457, 90^{717}$, etc.

Given sets X and Y, we say that X is a *subset* of Y, and write $X \subseteq Y$, if every element in X is also in Y. For example $\{h\} \subseteq A$. Note that the empty set $\varnothing \coloneqq \{\}$ is a subset of every other set.[4] Given a set Y and a property P that is either true or false for each element of Y, we write $\{y \in Y \mid P(y)\}$ to mean the subset of those y's that satisfy P.

Exercise 1.6.

1. Is it true that $\mathbb{N} = \{n \in \mathbb{Z} \mid n \geq 0\}$?

[3] If you want a notion where "h, 1" is different from "$h, h, 1, h, 1$," you can use something called *bags*, where the number of times an element is listed matters, or *lists*, where order also matters. All of these are important concepts in applied category theory, but sets will come up the most for us.

[4] When we write $Z \coloneqq$ `foo`, it means "assign the meaning `foo` to variable Z," whereas $Z =$ `foo` means simply that Z is equal to `foo`, perhaps as discovered via some calculation. In particular, $Z \coloneqq$ `foo` implies $Z =$ `foo` but not vice versa; indeed it *would not* be proper to write $3 + 2 \coloneqq 5$ or $\{\} \coloneqq \varnothing$.

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

2. Is it true that $\mathbb{N} = \{n \in \mathbb{Z} \mid n \geq 1\}$?
3. Is it true that $\varnothing = \{n \in \mathbb{Z} \mid 1 < n < 2\}$? ◇

If both X_1 and X_2 are subsets of Y, their *union*, denoted $X_1 \cup X_2$, is also a subset of Y, namely the one containing the elements in X_1 and the elements in X_2 but no more. For example if $Y = \{1, 2, 3, 4\}$ and $X_1 = \{1, 2\}$ and $X_2 = \{2, 4\}$, then $X_1 \cup X_2 = \{1, 2, 4\}$. Note that $\varnothing \cup X = X$ for any $X \subseteq Y$.

Similarly, if both X_1 and X_2 are subsets of Y, then their *intersection*, denoted $X_1 \cap X_2$, is also a subset of Y, namely the one containing all the elements of Y that are both in X_1 and in X_2, and no others. So $\{1, 2, 3\} \cap \{2, 5\} = \{2\}$.

What if we need to union together or intersect a lot of subsets? For example, consider the sets $X_0 = \varnothing$, $X_1 = \{1\}$, $X_2 = \{1, 2\}$, etc. as subsets of $\mathbb{N}$, and we want to know what the union of all of them is. This union is written $\bigcup_{n \in \mathbb{N}} X_n$, and it is the subset of $\mathbb{N}$ that contains every element of every X_n, but no others. Namely, $\bigcup_{n \in \mathbb{N}} X_n = \{n \in \mathbb{N} \mid n \geq 1\}$. Similarly one can write $\bigcap_{n \in \mathbb{N}} X_n$ for the intersection of all of them, which will be empty in the above case.

Given two sets X and Y, the *product* $X \times Y$ of X and Y is the set of pairs (x, y), where $x \in X$ and $y \in Y$.

Finally, we may want to take a *disjoint* union of two sets, even if they have elements in common. Given two sets X and Y, their *disjoint union* $X \sqcup Y$ is the set of pairs of the form $(x, 1)$ or $(y, 2)$, where $x \in X$ and $y \in Y$.

Exercise 1.7. Let $A := \{h, 1\}$ and $B := \{1, 2, 3\}$.

1. There are eight subsets of B; write them out.
2. Take any two nonempty subsets of B and write out their union.
3. There are six elements in $A \times B$; write them out.
4. There are five elements of $A \sqcup B$; write them out.
5. If we consider A and B as subsets of the set $\{h, 1, 2, 3\}$, there are four elements of $A \cup B$; write them out. ◇

Relationships between different sets – for example between the set of trees in your neighborhood and the set of your favorite fruits – are captured using subsets and product sets.

Definition 1.8. Let X and Y be sets. A *relation between X and Y* is a subset $R \subseteq X \times Y$. A *binary relation on X* is a relation between X and X, i.e. a subset $R \subseteq X \times X$.

It is convenient to use something called *infix notation* for binary relations $R \subseteq A \times A$. This means one picks a symbol, say $\star$, and writes $a \star b$ to mean $(a, b) \in R$.

Example 1.9. There is a binary relation on $\mathbb{R}$ with infix notation $\leq$. Rather than writing $(5, 6) \in R$, we write $5 \leq 6$.

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

Other examples of infix notation for relations are $=, \approx, <, >$. In number theory, we are interested in whether one number divides without remainder into another number; this relation is denoted with infix notation $|$, so $5|10$.

Partitions and equivalence relations

We can now define partitions more formally.

Definition 1.10. If A is a set, a *partition* of A consists of a set P and, for each $p \in P$, a nonempty subset $A_p \subseteq A$, such that

$$A = \bigcup_{p \in P} A_p \qquad \text{and} \qquad \text{if } p \neq q \text{ then } A_p \cap A_q = \varnothing. \tag{1.5}$$

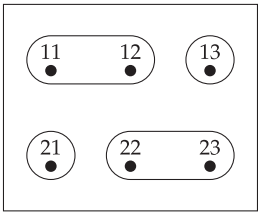
We may denote the partition by $\{A_p\}_{p \in P}$. We refer to P as the set of *part labels* and if $p \in P$ is a part label, we refer to A_p as the pth *part*. The condition (1.5) says that each element $a \in A$ is in exactly one part.

We consider two different partitions $\{A_p\}_{p \in P}$ and $\{A'_{p'}\}_{p' \in P'}$ of A to be the same if for each $p \in P$ there exists a $p' \in P'$ with $A_p = A'_{p'}$. In other words, if two ways to divide A into parts are exactly the same – the only change is in the labels – then we don't make a distinction between them.

Exercise 1.11. Suppose that A is a set and $\{A_p\}_{p \in P}$ and $\{A'_{p'}\}_{p' \in P'}$ are two partitions of A such that for each $p \in P$ there exists a $p' \in P'$ with $A_p = A'_{p'}$.

1. Show that for each $p \in P$ there is at most one $p' \in P'$ such that $A_p = A'_{p'}$.
2. Show that for each $p' \in P'$ there is a $p \in P$ such that $A_p = A'_{p'}$. ◇

Exercise 1.12. Consider the partition shown below:



For any two elements $a, b \in \{11, 12, 13, 21, 22, 23\}$, let's allow ourselves to write a twiddle (tilde) symbol $a \sim b$ between them if a and b are both in the same part. Write down every pair of elements (a, b) that are in the same part. There should be 10.[5] ◇

We shall see in Proposition 1.14 that there is a strong relationship between partitions and something called equivalence relations, which we define next.

[5] Hint: whenever someone speaks of "two elements a, b in a set A," the two elements may be the same!

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

Definition 1.13. Let A be a set. An *equivalence relation* on A is a binary relation, let's give it infix notation $\sim$, satisfying the following three properties:

(a) $a \sim a$, for all $a \in A$,
(b) $a \sim b$ iff[6] $b \sim a$, for all $a, b \in A$,
(c) if $a \sim b$ and $b \sim c$ then $a \sim c$, for all $a, b, c \in A$.

These properties are called *reflexivity*, *symmetry*, and *transitivity*, respectively.

Proposition 1.14. Let A be a set. There is a one-to-one correspondence between the ways to partition A and the equivalence relations on A.

Proof. We first show that every partition gives rise to an equivalence relation, and then that every equivalence relation gives rise to a partition. Our two constructions will be mutually inverse, proving the proposition.

Suppose we are given a partition $\{A_p\}_{p \in P}$; we define a relation $\sim$ and show it is an equivalence relation. Define $a \sim b$ to mean that a and b are in the same part: there is some $p \in P$ such that $a \in A_p$ and $b \in A_p$. It is obvious that a is in the same part as itself. Similarly, it is obvious that if a is in the same part as b then b is in the same part as a, and that if further b is in the same part as c then a is in the same part as c. Thus $\sim$ is an equivalence relation as defined in Definition 1.13.

Suppose we are given an equivalence relation $\sim$; we will form a partition on A by saying what the parts are. Say that a subset $X \subseteq A$ is $(\sim)$-closed if, for every $x \in X$ and $x' \sim x$, we have $x' \in X$. Say that a subset $X \subseteq A$ is $(\sim)$-connected if it is nonempty and $x \sim y$ for every $x, y \in X$. Then the parts corresponding to $\sim$ are exactly the $(\sim)$-closed, $(\sim)$-connected subsets. It is not hard to check that these indeed form a partition. □

Exercise 1.15. Let's complete the "it's not hard to check" part in the proof of Proposition 1.14. Suppose that $\sim$ is an equivalence relation on a set A, and let P be the set of $(\sim)$-closed and $(\sim)$-connected subsets $\{A_p\}_{p \in P}$.

1. Show that each part A_p is nonempty.
2. Show that if $p \neq q$, i.e. if A_p and A_q are not exactly the same set, then $A_p \cap A_q = \varnothing$.
3. Show that $A = \bigcup_{p \in P} A_p$. ◇

Definition 1.16. Given a set A and an equivalence relation $\sim$ on A, we say that the *quotient* $A/\sim$ of A under $\sim$ is the set of parts of the corresponding partition.

Functions

The most frequently used sort of relation between sets is that of functions.

[6] "Iff" is short for "if and only if."

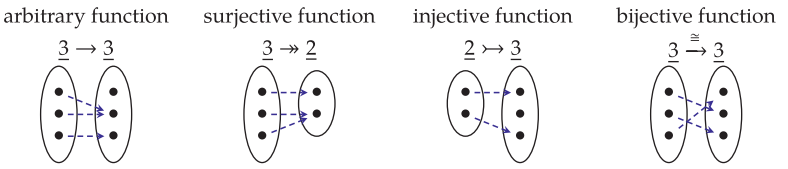
https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

Definition 1.17. Let S and T be sets. A *function from S to T* is a subset $F \subseteq S \times T$ such that for all $s \in S$, there exists a unique $t \in T$ with $(s, t) \in F$.

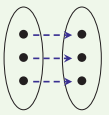
The function F is often denoted $F\colon S \to T$. From now on, we write $F(s) = t$, or sometimes $s \mapsto t$, to mean $(s, t) \in F$. For any $t \in T$, the *preimage of t along F* is the subset $f^{-1}(t) \coloneqq \{s \in S \mid F(s) = t\}$.

A function is called *surjective*, or a *surjection*, if for all $t \in T$, there exists $s \in S$ with $F(s) = t$. A function is called *injective*, or an *injection*, if for all $t \in T$ and $s_1, s_2 \in S$ with $F(s_1) = t$ and $F(s_2) = t$, we have $s_1 = s_2$. A function is called *bijective* if it is both surjective and injective.

We use various decorations on arrows, $\to$, $\twoheadrightarrow$, $\rightarrowtail$, $\xrightarrow{\cong}$ to denote these special sorts of functions. Here is a table with the name, arrow decoration, and an example of each sort of function:



Example 1.18. An important but very simple sort of function is the *identity function* on a set X, denoted id_X. It is the bijective function $\mathrm{id}_X(x) = x$.

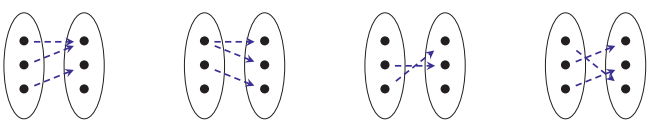


For notational consistency with Definition 1.17, the arrows in Example 1.18 might be drawn as $\mapsto$ rather than $\dashrightarrow$. The $\dashrightarrow$-style arrows were drawn because we thought it was prettier, i.e. easier on the eye. Beauty is important too; an imbalanced preference for strict correctness over beauty becomes *pedantry*. But, outside of pictures, we will be careful.

Exercise 1.19. In the following, do not use any examples already drawn above.

1. Find two sets A and B and a function $f\colon A \to B$ that is injective but not surjective.
2. Find two sets A and B and a function $f\colon A \to B$ that is surjective but not injective.

Now consider the four relations shown here:



https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

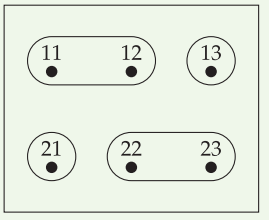
For each relation, answer the following two questions.

3. Is it a function?
4. If not, why not? If so, is it injective, surjective, both (i.e. bijective), or neither? ◇

Exercise 1.20. Suppose that A is a set and $f : A \to \varnothing$ is a function to the empty set. Show that A is empty. ◇

Example 1.21. A partition on a set A can also be understood in terms of surjective functions out of A. Given a surjective function $f : A \twoheadrightarrow P$, where P is any other set, the preimages $f^{-1}(p) \subseteq A$, one for each element $p \in P$, form a partition of A. Here is an example.

Consider the partition of $S := \{11, 12, 13, 21, 22, 23\}$ shown below:

$S :=$ 

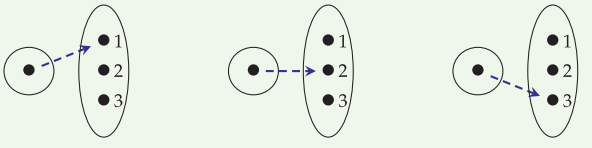
It has been partitioned into four parts, so let $P = \{a, b, c, d\}$ and let $f : S \twoheadrightarrow P$ be given by

$$f(11) = a, \quad f(12) = a, \quad f(13) = b, \quad f(21) = c, \quad f(22) = d, \quad f(23) = d.$$

Exercise 1.22. Write down a surjection corresponding to each of the five partitions in Eq. (1.3). ◇

Definition 1.23. If $F : X \to Y$ is a function and $G : Y \to Z$ is a function, their *composite* is the function $X \to Z$ defined to be $G(F(x))$ for any $x \in X$. It is often denoted $G \circ F$, but we prefer to denote it $F \fatsemi G$. It takes any element $x \in X$, evaluates F to get an element $F(x) \in Y$ and then evaluates G to get an element $G(F(x))$.

Example 1.24. If X is any set and $x \in X$ is any element, we can think of x as a function $\{1\} \to X$, namely the function sending 1 to x. For example, the three functions $\{1\} \to \{1, 2, 3\}$ shown below correspond to the three elements of $\{1, 2, 3\}$:



https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

Suppose we are given a function $F\colon X \to Y$ and an element of X, thought of as a function $x\colon \{1\} \to X$. Then evaluating F at x is given by the composite $F(x) = x \fatsemi F$.

1.2.2 Preorders

In Section 1.1, we often used the symbol $\leq$ to denote a sort of order. Here is a formal definition of what it means for a set to have an order.

Definition 1.25. A *preorder relation* on a set X is a binary relation on X, here denoted with infix notation $\leq$, such that

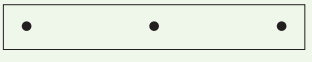
(a) $x \leq x$; and
(b) if $x \leq y$ and $y \leq z$, then $x \leq z$.

The first condition is called *reflexivity* and the second is called *transitivity*. If $x \leq y$ and $y \leq x$, we write $x \cong y$ and say x and y are *equivalent*. We call a pair $(X, \leq)$ consisting of a set equipped with a preorder relation a *preorder*.

Remark 1.26. Observe that reflexivity and transitivity are familiar from Definition 1.13: equivalence relations are preorders with an additional symmetry condition.

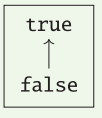
Example 1.27 (Discrete preorders)**.** Every set X can be considered as a discrete preorder $(X, =)$. This means that the only order relationships on X are of the form $x \leq x$; if $x \neq y$ then neither $x \leq y$ nor $y \leq x$ hold.

We depict discrete preorders as simply a collection of points:



Example 1.28 (Codiscrete preorders)**.** From every set we may also construct its codiscrete preorder $(X, \leq)$ by equipping it with the total binary relation $X \times X \subseteq X \times X$. This is a very trivial structure: it means that for *all* x and y in X we have $x \leq y$ (and hence also $y \leq x$).

Example 1.29 (Booleans)**.** The booleans $\mathbb{B} = \{\texttt{false}, \texttt{true}\}$ form a preorder with $\texttt{false} \leq \texttt{true}$.



https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

Remark 1.30 (Partial orders are skeletal preorders)**.** A preorder is a *partial order* if we additionally have that

(c) $x \cong y$ implies $x = y$.

In category theory terminology, the requirement that $x \cong y$ implies $x = y$ is known as *skeletality*, so partial orders are *skeletal preorders*. For short, we also use the term *poset*, a contraction of partially ordered set.

The difference between preorders and partial orders is rather minor. A partial order already is a preorder, and every preorder can be made into a partial order by equating any two elements x, y for which $x \cong y$, i.e. for which $x \leq y$ and $y \leq x$.

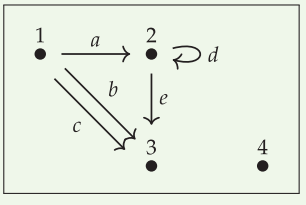
For example, any discrete preorder is already a partial order, while any codiscrete preorder simply becomes the unique partial order on a one-element set.

We have already introduced a few examples of preorders using Hasse diagrams. It will be convenient to continue to do this, so let us be a bit more formal about what we mean. First, we need to define a graph.

Definition 1.31. A *graph* $G = (V, A, s, t)$ consists of a set V whose elements are called *vertices*, a set A whose elements are called *arrows*, and two functions $s, t \colon A \to V$ known as the *source* and *target* functions respectively. Given $a \in A$ with $s(a) = v$ and $t(a) = w$, we say that a is an arrow from v to w.

By a *path* in G we mean any sequence of arrows such that the target of one arrow is the source of the next. This includes sequences of length 1, which are just arrows $a \in A$ in G, and sequences of length 0, which just start and end at the same vertex v, without traversing any arrows.

Example 1.32. Here is a picture of a graph:

$G =$ 

It has $V = \{1, 2, 3, 4\}$ and $A = \{a, b, c, d, e\}$. The source and target functions, $s, t \colon A \to V$ are given by the following partially filled-in tables (see Exercise 1.33):

Arrow a	**source** $s(a) \in V$	**target** $t(a) \in V$
a	1	?
b	1	3
c	?	?
d	?	?
e	?	?

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

There are no paths from 4 to 3, but there is one path from 4 to 4, namely the path of length 0. There are infinitely many paths $1 \to 2$ because one can loop and loop and loop through d as many times as one pleases.

Exercise 1.33. Copy and complete the table from Example 1.32. ◇

Remark 1.34. From every graph we can get a preorder. Indeed, a Hasse diagram is a graph $G = (V, A, s, t)$ that gives a *presentation* of a preorder $(P, \leq)$. The elements of P are the vertices V in G, and the order $\leq$ is given by $v \leq w$ iff there is a path $v \to w$. For any vertex v, there is always a path $v \to v$, and this translates into the reflexivity law from Definition 1.25. The fact that paths $u \to v$ and $v \to w$ can be concatenated to a path $u \to w$ translates into the transitivity law.

Exercise 1.35. What preorder relation $(P, \leq)$ is depicted by the graph G in Example 1.32? That is, write down the elements of P and write down every pair (p_1, p_2) for which $p_1 \leq p_2$. ◇

Exercise 1.36. Does a collection of points, like the one in Example 1.27, count as a Hasse diagram? ◇

Exercise 1.37. Let X be the set of partitions of $\{\bullet, \circ, *\}$; it has five elements and an order by coarseness, as shown in the Hasse diagram Eq. (1.3). Write down every pair (x, y) of elements in X such that $x \leq y$. There should be 12. ◇

Remark 1.38. In Example 1.30 we discussed partial orders – preorders with the property that whenever two elements are equivalent, they are the same – and then said that this property is fairly inconsequential: any preorder can be converted to a partial order that's "equivalent" category-theoretically. A partial order is like a preorder with a fancy haircut: some mathematicians might not even notice it.

However, there are other types of preorders that are more special and noticeable. For example, a *total order* has the following additional property:

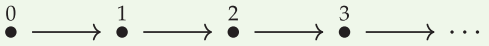
(d) for all x, y, either $x \leq y$ or $y \leq x$.

We say two elements x, y of a preorder are *comparable* if either $x \leq y$ or $y \leq x$, so a total order is a preorder where *every* two elements are comparable.

Exercise 1.39. Is it correct to say that a discrete preorder is one where *no* two elements are comparable? ◇

Example 1.40 (Natural numbers)**.** The natural numbers $\mathbb{N} := \{0, 1, 2, 3, \ldots\}$ are a preorder with the order given by the usual size ordering, e.g. $0 \leq 1$ and $5 \leq 100$. This is a total order: either $m \leq n$ or $n \leq m$ for all m, n. One can see that its Hasse diagram looks like a line:

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press



What made Eq. (1.3) not look like a line is that there are non-comparable elements a and b – namely all those in the middle row – which satisfy neither $a \leq b$ nor $b \leq a$.

Note that, for any set S, there are many different ways of assigning an order to S. Indeed, for the set $\mathbb{N}$, we could also use the discrete ordering: only write $n \leq m$ if $n = m$. Another ordering is the reverse ordering, such as $5 \leq 3$ and $3 \leq 2$, like how golf is scored (5 is worse than 3).

Yet another ordering on $\mathbb{N}$ is given by division: we say that $n \leq m$ if n divides into m without remainder. In this ordering $2 \leq 4$, for example, but $2 \not\leq 3$, since there is a remainder when 2 is divided into 3.

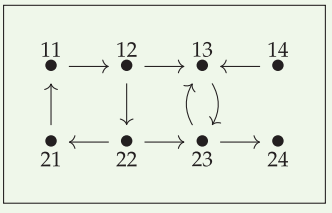
Exercise 1.41. Write down the numbers $1, 2, \ldots, 10$ and draw an arrow $a \to b$ if a divides perfectly into b. Is it a total order? ◇

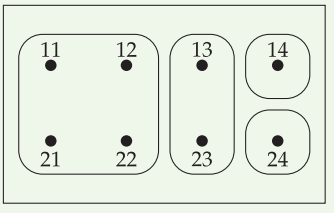
Example 1.42 (Real numbers)**.** The real numbers $\mathbb{R}$ also form a preorder with the "usual ordering," e.g. $-500 \leq -499 \leq 0 \leq \sqrt{2} \leq 100/3$.

Exercise 1.43. Is the usual $\leq$ ordering on the set $\mathbb{R}$ of real numbers a total order? ◇

Example 1.44 (Partition from preorder)**.** Given a preorder, i.e. a preordered set $(P, \leq)$, we defined the notion of equivalence of elements, denoted $x \cong y$, to mean $x \leq y$ and $y \leq x$. This is an equivalence relation, so it induces a partition on P. (The phrase "A induces B" means that we have an automatic way to turn an A into a B. In this case, we're saying that we have an automatic way to turn equivalence relations into partitions, which we do; see Proposition 1.14.)

For example, the preorder whose Hasse diagram is drawn on the left corresponds to the partition drawn on the right.

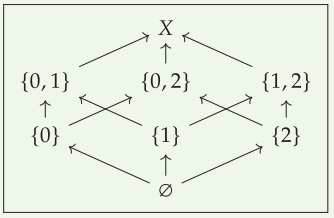




Example 1.45 (Power set)**.** Given a set X, the set of subsets of X is known as the *power set* of X; we denote it $\mathsf{P}(X)$. The power set can naturally be given an order by inclusion of subsets (and from now on, whenever we speak of the power set as an ordered set, this is the order we mean).

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

For example, taking $X = \{0, 1, 2\}$, we depict $\mathsf{P}(X)$ as



See the cube? The Hasse diagram for the power set of a finite set, say $\mathsf{P}\{1, 2, \ldots, n\}$,[7] always looks like a cube of dimension n.

Exercise 1.46. Draw the Hasse diagrams for $\mathsf{P}(\varnothing)$, $\mathsf{P}\{1\}$, and $\mathsf{P}\{1, 2\}$. ◇

Example 1.47 (Partitions)**.** We talked about getting a partition from a preorder; now let's think about how we might order the set $\mathsf{Prt}(A)$ of *all partitions* of A, for some set A. In fact, we have done this before in Eq. (1.3). Namely, we order partitions by fineness: a partition P is *finer* than a partition Q if, for every part $p \in P$, there is a part $q \in Q$ such that $A_p \subseteq A_q$. We could also say that Q is *coarser* than P.

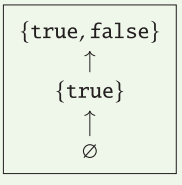
Recall from Example 1.21 that partitions on A can be thought of as surjective functions out of A. Then $f\colon A \twoheadrightarrow P$ is finer than $g\colon A \twoheadrightarrow Q$ if there is a function $h\colon P \to Q$ such that $f \fatsemi h = g$.

Exercise 1.48. For any set S there is a coarsest partition, having just one part. What surjective function does it correspond to?

There is also a finest partition, where everything is in its own part. What surjective function does it correspond to? ◇

Example 1.49 (Upper sets)**.** Given a preorder $(P, \leq)$, an *upper set* in P is a subset U of P satisfying the condition that if $p \in U$ and $p \leq q$, then $q \in U$. "If p is an element then so is anything bigger." Write $\mathsf{U}(P)$ for the set of upper sets in P. We can give the set U an order by letting $U \leq V$ if U is contained in V.

For example, if $(\mathbb{B}, \leq)$ is the booleans (Example 1.29), then its preorder of upper sets $\mathsf{U}(\mathbb{B})$ is



[7] Note that we omit the parentheses here, writing $\mathsf{P}X$ instead of $\mathsf{P}(X)$; throughout this book we will omit parentheses if we judge the presentation is cleaner and it is unlikely to cause confusion.

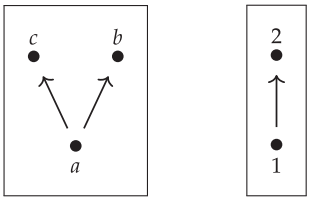
https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

The subset $\{\texttt{false}\} \subseteq \mathbb{B}$ is not an upper set, because $\texttt{false} \leq \texttt{true}$ and $\texttt{true} \notin \{\texttt{false}\}$.

Exercise 1.50. Prove that the preorder of upper sets on a discrete preorder (see Example 1.27) on a set X is simply the power set $\mathsf{P}(X)$. ◇

Example 1.51 (Product preorder)**.** Given preorders $(P, \leq)$ and $(Q, \leq)$, we may define a preorder structure on the product set $P \times Q$ by setting $(p, q) \leq (p', q')$ if and only if $p \leq p'$ and $q \leq q'$. We call this the *product preorder*. This is a basic example of a more general construction known as the product of categories.

Exercise 1.52. Draw the Hasse diagram for the product of the two preorders drawn below:



For bonus points, compute the upper set preorder on the result. ◇

Example 1.53 (Opposite preorder)**.** Given a preorder $(P, \leq)$, we may define the opposite preorder $(P, \leq^{\mathrm{op}})$ to have the same set of elements, but with $p \leq^{\mathrm{op}} q$ if and only if $q \leq p$.

1.2.3 Monotone Maps

We have said that the categorical perspective emphasizes relationships between things. For example, a preorder is a setting – or world – in which we have one sort of relationship, $\leq$, and any two objects may be, or may not be, so-related. Jumping up a level, the categorical perspective emphasizes that preorders themselves – each a miniature world composed of many relationships – can be related to one another.

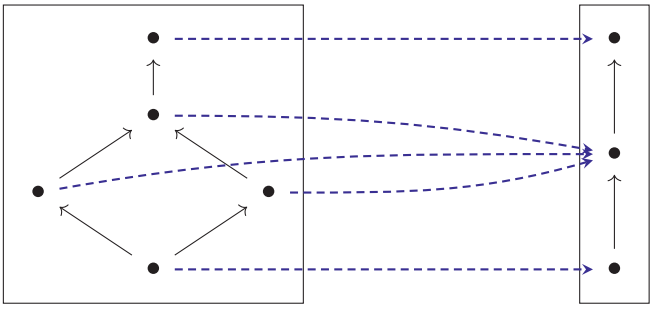
The most important sort of relationship between preorders is called a *monotone map*. These are functions that preserve preorder relations – in some sense mappings that respect $\leq$ – and are hence considered the right notion of *structure-preserving map* for preorders.

Definition 1.54. A *monotone map* between preorders $(A, \leq_A)$ and $(B, \leq_B)$ is a function $f : A \to B$ such that, for all elements $x, y \in A$, if $x \leq_A y$ then $f(x) \leq_B f(y)$.

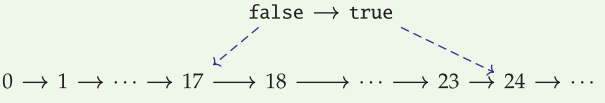
A monotone map $A \to B$ between two preorders associates to each element of preorder A an element of the preorder B. We depict this by drawing a dotted arrow from

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

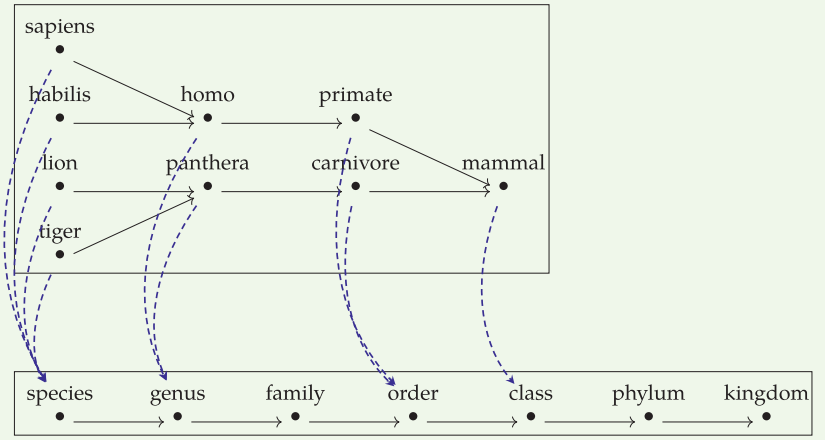
each element $x \in A$ to its image $f(x) \in B$. Note that the order must be preserved in order to count as a valid monotone map, so if element x is above element y in the left-hand preorder A, then the image $f(x)$ will be above the image $f(y)$ in the right-hand preorder.



Example 1.55. Let $\mathbb{B}$ and $\mathbb{N}$ be the preorders of booleans from Example 1.29 and $\mathbb{N}$ be the preorder of natural numbers from Example 1.40. The map $\mathbb{B} \to \mathbb{N}$ sending `false` to 17 and `true` to 24 is a monotone map, because it preserves order.



Example 1.56 (The tree of life)**.** Consider the set of all animal classifications, for example "tiger," "mammal," "sapiens," "carnivore," etc. These are ordered by specificity: since "tiger" is a type of "mammal," we write tiger $\leq$ mammal. The result is a preorder, which in fact forms a tree, often called the tree of life. At the top of the following diagram we see a small part of it:



https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

At the bottom we see the hierarchical structure as a preorder. The dashed arrows show a monotone map, call it F, from the classifications to the hierarchy. It is monotone because it preserves order: whenever there is a path $x \to y$ upstairs, there is a path $F(x) \to F(y)$ downstairs.

Example 1.57. Given a finite set X, recall the power set $\mathsf{P}(X)$ and its natural order relation from Example 1.45. The map $|\cdot| \colon \mathsf{P}(X) \to \mathbb{N}$ sending each subset S to its number of elements $|S|$, also called its *cardinality*, is a monotone map.

Exercise 1.58. Let $X = \{0, 1, 2\}$.

1. Draw the Hasse diagram for $\mathsf{P}(X)$.
2. Draw the Hasse diagram for the preorder $0 \leq 1 \leq 2 \leq 3$.
3. Draw the cardinality map $|\cdot|$ from Example 1.57 as dashed lines between them. ◇

Example 1.59. Recall the notion of upper set from Example 1.49. Given a preorder $(P, \leq)$, the map $\mathsf{U}(P) \to \mathsf{P}(P)$ sending each upper set of $(P, \leq)$ to itself – considered as a subset of P – is a monotone map.

Exercise 1.60. Consider the preorder $\mathbb{B}$. The Hasse diagram for $\mathsf{U}(\mathbb{B})$ was drawn in Example 1.49, and you drew the Hasse diagram for $\mathsf{P}(\mathbb{B})$ in Exercise 1.46. Now draw the monotone map between them, as described in Example 1.59. ◇

Exercise 1.61. Let $(P, \leq)$ be a preorder, and recall the notion of opposite preorder from Example 1.53.

1. Show that the set $\uparrow p \coloneqq \{p' \in P \mid p \leq p'\}$ is an upper set, for any $p \in P$.
2. Show that this construction defines a monotone map $\uparrow \colon P^{\mathrm{op}} \to \mathsf{U}(P)$.
3. Show that $p \leq p'$ in P if and only if $\uparrow(p') \subseteq \uparrow(p)$.
4. Draw a picture of the map $\uparrow$ in the case where P is the preorder $(b \geq a \leq c)$ from Exercise 1.52.

This is known as the *Yoneda lemma* for preorders. The if and only if condition proved in part 3 implies that, up to equivalence, knowing an element p is the same as knowing its upper set P– that is, knowing its web of relationships with the other elements of the preorder. The general Yoneda lemma is a powerful tool in category theory, and a fascinating philosophical idea besides. ◇

Exercise 1.62. As you know, a monotone map $f \colon (P, \leq_P) \to (Q, \leq_Q)$ consists of a function $f \colon P \to Q$ that satisfies a "monotonicity" property. Show that when $(P, \leq_P)$ is a discrete preorder, then *every* function $P \to Q$ satisfies the monotonicity property, regardless of the order $\leq_Q$. ◇

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

Example 1.63. Recall from Example 1.47 that given a set X we define $\mathsf{Prt}(X)$ to be the set of partitions on X, and that a partition may be defined using a surjective function $s\colon X \twoheadrightarrow P$ for some set P.

Any surjective function $f\colon X \twoheadrightarrow Y$ induces a monotone map $f^*\colon \mathsf{Prt}(Y) \to \mathsf{Prt}(X)$, going "backwards." It is defined by sending a partition $s\colon Y \twoheadrightarrow P$ to the composite $f \fatsemi s\colon X \twoheadrightarrow P$.[8]

Exercise 1.64. Choose two sets X and Y with at least three elements each and choose a surjective, non-identity function $f\colon X \twoheadrightarrow Y$ between them. Write down two different partitions P and Q of Y, and then find $f^*(P)$ and $f^*(Q)$. ◇

The following proposition, Proposition 1.65, is straightforward to check. Recall the definition of the identity function from Example 1.18 and the definition of composition from Definition 1.23.

Proposition 1.65. For any preorder $(P, \leq_P)$, the identity function is monotone.

If $(Q, \leq_Q)$ and $(R, \leq_R)$ are preorders and $f\colon P \to Q$ and $g\colon Q \to R$ are monotone, then $(f \fatsemi g)\colon P \to R$ is also monotone.

Exercise 1.66. Check the two claims made in Proposition 1.65. ◇

Example 1.67. Recall again the definition of opposite preorder from Example 1.53. The identity function $\mathrm{id}_P\colon P \to P$ is a monotone map $(P, \leq) \to (P, \leq^{\mathrm{op}})$ if and only if for all $p, q \in P$ we have $q \leq p$ whenever $p \leq q$. For historical reasons connected to linear algebra, when this is true, we call $(P, \leq)$ a *dagger preorder*.

But in fact, we have seen dagger preorders before in another guise. Indeed, if $(P, \leq)$ is a dagger preorder, then the relation $\leq$ is symmetric: $p \leq q$ if and only if $q \leq p$, and it is also reflexive and transitive by definition of preorder. So in fact $\leq$ is an equivalence relation (Definition 1.13).

Exercise 1.68. Recall the notion of skeletal preorders (Remark 1.30) and discrete preorders (Example 1.27). Show that a skeletal dagger preorder is just a discrete preorder, and hence can be identified with a set. ◇

Remark 1.69. We say that an A "can be identified with" a B when any A gives us a unique B and any B gives us a unique A, and both round-trips – from an A to a B and back to an A, or from a B to an A and back to a B – return us where we started. For example, any discrete preorder $(P, \leq)$ has an underlying set P, and any set P can be made into a discrete preorder ($p_1 \leq p_2$ iff $p_1 = p_2$), and the round-trips return us where we started. So what's the difference? It's like the notion of *object-permanence*

[8] We shall later see that any function $f\colon X \to Y$, not necessarily surjective, induces a monotone map $f^*\colon \mathsf{Prt}(Y) \to \mathsf{Prt}(X)$, but it involves an extra step. See Section 1.4.2.

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

from child development jargon: we can recognize "the same chair, just moved from one room to another." A chair in the room of sets can be moved to a chair in the room of preorders. The lighting is different but the chair is the same.

Eventually, we will be able to understand this notion in terms of *equivalence of categories*, which are related to isomorphisms, which we will explore next in Definition 1.70.

Definition 1.70. Let $(P, \leq_P)$ and $(Q, \leq_Q)$ be preorders. A monotone function $f\colon P \to Q$ is called an *isomorphism* if there exists a monotone function $g\colon Q \to P$ such that $f \fatsemi g = \mathrm{id}_P$ and $g \fatsemi f = \mathrm{id}_Q$. This means that, for any $p \in P$ and $q \in Q$, we have

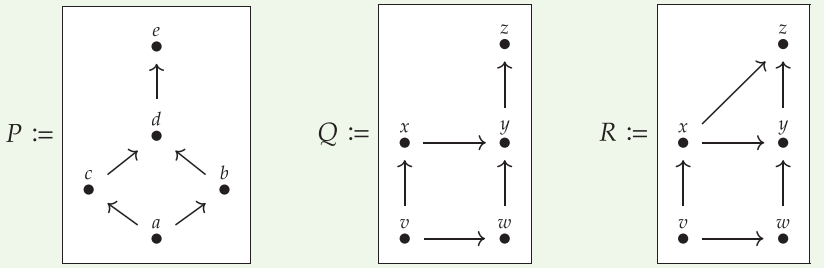
$$p = g(f(p)) \qquad \text{and} \qquad q = f(g(q)).$$

We refer to g as the *inverse* of f, and vice versa: f is the inverse of g.

If there is an isomorphism $P \to Q$, we say that P and Q are *isomorphic*.

An isomorphism between preorders is basically just a relabeling of the elements.

Example 1.71. Here are the Hasse diagrams for three preorders P, Q, and R, all of which are isomorphic:



The map $f\colon P \to Q$ given by $f(a) = v$, $f(b) = w$, $f(c) = x$, $f(d) = y$, and $f(e) = z$ has an inverse.

In fact Q and R are the same preorder. One may be confused by the fact that there is an arrow $x \to z$ in the Hasse diagram for R and not one in Q, but in fact this arrow is superfluous. By the transitivity property of preorders (Definition 1.25), since $x \leq y$ and $y \leq z$, we must have $x \leq z$, whether it is drawn or not. Similarly, we could have drawn an arrow $v \to y$ in either Q or R and it would not have changed the preorder.

Recall the preorder $\mathbb{B} = \{\texttt{false}, \texttt{true}\}$, where $\texttt{false} \leq \texttt{true}$. As simple as this preorder is, it is also one of the most important.

Exercise 1.72. Show that the map Φ from Section 1.1.1, which was roughly given by "Is $\bullet$ connected to $*$?" is a monotone map $\mathsf{Prt}(\{*, \bullet, \circ\}) \to \mathbb{B}$; see also Eq. (1.3). ◇

Proposition 1.73. Let P be a preorder. Monotone maps $P \to \mathbb{B}$ are in one-to-one correspondence with upper sets of P.

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

Proof. Let $f\colon P \to \mathbb{B}$ be a monotone map. We will show that the subset $f^{-1}(\texttt{true}) \subseteq P$ is an upper set. Suppose $p \in f^{-1}(\texttt{true})$ and $p \leq q$; then $\texttt{true} = f(p) \leq f(q)$. But in $\mathbb{B}$, if $\texttt{true} \leq f(q)$ then $\texttt{true} = f(q)$. This implies $q \in f^{-1}(\texttt{true})$ and thus shows that $f^{-1}(\texttt{true})$ is an upper set.

Conversely, if U is an upper set in P, define $f_U\colon P \to \mathbb{B}$ such that $f_U(p) = \texttt{true}$ when $p \in U$, and $f_U(p) = \texttt{false}$ when $p \notin U$. This is a monotone map, because if $p \leq q$, then either $p \in U$, so $q \in U$ and $f_U(p) = \texttt{true} = f(q)$, or $p \notin U$, so $f_U(p) = \texttt{false} \leq f_U(q)$.

These two constructions are mutually inverse, and hence prove the proposition. □

Exercise 1.74 (Pullback map)**.** Let P and Q be preorders, and $f\colon P \to Q$ be a monotone map. Then we can define a monotone map $f^*\colon \mathtt{U}(Q) \to \mathtt{U}(P)$ sending an upper set $U \subseteq Q$ to the upper set $f^{-1}(U) \subseteq P$. We call this the *pullback along* f.

Viewing upper sets as monotone maps to $\mathbb{B}$ as in Proposition 1.73, the pullback can be understood in terms of composition. Indeed, show that f^* is defined by taking $u\colon Q \to \mathbb{B}$ to $(f \fatsemi u)\colon P \to \mathbb{B}$. ◇

1.3 Meets and Joins

As we have said, a preorder is a set P endowed with an order $\leq$ relating the elements. With respect to this order, certain elements of P may have distinctive characterizations, either absolutely or in relation to other elements. We have discussed joins before, but we discuss them again now that we have built up some formalism.

1.3.1 Definition and Basic Examples

Consider the preorder $(\mathbb{R}, \leq)$ of real numbers ordered in the usual way. The subset $\mathbb{N} \subseteq \mathbb{R}$ has many lower bounds, namely -1.5 is a lower bound: every element of $\mathbb{N}$ is bigger than -1.5. But within all lower bounds for $\mathbb{N} \subseteq \mathbb{R}$, one is distinctive: a *greatest lower bound* – also called a *meet* – namely 0. It is a lower bound, and there is no lower bound for $\mathbb{N}$ that is above it. However, the set $\mathbb{N} \subseteq \mathbb{R}$ has no upper bound, and certainly no least upper bound – which would be called a *join*. On the other hand, the set

$$\left\{ \frac{1}{n+1} \;\middle|\; n \in \mathbb{N} \right\} = \left\{ 1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \dots \right\} \subseteq \mathbb{R}$$

has both a greatest lower bound (meet), namely 0, and a least upper bound (join), namely 1.

These notions will have correlates in category theory, called limits and colimits, which we will discuss in Chapter 3. More generally, we say these distinctive characterizations are *universal properties*, since, for example, a greatest lower bound is greatest among *all* lower bounds. For now, however, we simply want to make the definition of greatest lower bounds and least upper bounds, called meets and joins, precise.

Exercise 1.75.

1. Why is 0 a lower bound for $\{\frac{1}{n+1} \mid n \in \mathbb{N}\} \subseteq \mathbb{R}$?
2. Why is 0 a *greatest* lower bound (meet)? ◇

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

Definition 1.76. Let $(P, \leq)$ be a preorder, and let $A \subseteq P$ be a subset. We say that an element $p \in P$ is a *meet* of A if

(a) for all $a \in A$, we have $p \leq a$,
(b) for all q such that $q \leq a$ for all $a \in A$, we have that $q \leq p$.

We write $p = \bigwedge A$, $p = \bigwedge_{a \in A} a$, or, if the dummy variable a is clear from context, just $p = \bigwedge_A a$. If A just consists of two elements, say $A = \{a, b\}$, we can denote $\bigwedge A$ simply by $a \wedge b$.

Similarly, we say that p is a *join* of A if

(a) for all $a \in A$ we have $a \leq p$,
(b) for all q such that $a \leq q$ for all $a \in A$, we have that $p \leq q$.

We write $p = \bigvee A$ or $p = \bigvee_{a \in A} a$, or when $A = \{a, b\}$ we may simply write $p = a \vee b$.

Remark 1.77. In Definition 1.76, we committed a seemingly egregious abuse of notation. We shall see next in Example 1.79 that there could be two different meets of $A \subseteq P$, say $p = \bigwedge A$ and $q = \bigwedge A$ with $p \neq q$, which does not make sense if $p \neq q$!

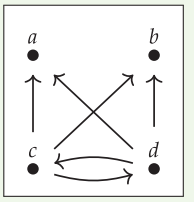
But in fact, as we use the symbol $\bigwedge A$, this abuse won't matter because any two meets p, q are automatically isomorphic: the very definition of meet forces both $p \leq q$ and $q \leq p$, and thus we have $p \cong q$. So, for any $x \in P$, we have $p \leq x$ iff $q \leq x$ and $x \leq p$ iff $x \leq q$. Thus as long as we are only interested in elements of P based on their relationships to other elements (and in category theory, this is the case: we should only care about things based on how they interact with other things, rather than on some sort of "internal essence"), the distinction between p and q will never matter.

This foreshadows a major theme of – as well as standard abuse of notation in – category theory, where any two things defined by the same universal property are automatically equivalent in a way known as "unique up to unique isomorphism"; this means that we generally do not run into trouble if we pretend they are equal. We'll pick up this theme of "the" vs. "a" again in Remark 3.70.

Example 1.78 (Meets or joins may not exist)**.** Note that, in an arbitrary preorder $(P, \leq)$, a subset A need not have a meet or a join. Consider the three-element set $P = \{p, q, r\}$ with the discrete ordering. The set $A = \{p, q\}$ does not have a join in P because if x was a join, we would need $p \leq x$ and $q \leq x$, and there is no such element x.

Example 1.79 (Multiple meets or joins may exist)**.** It may also be the case that a subset A has more than one meet or join. Here is an example.

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press



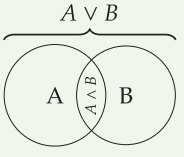
Let A be the subset $\{a, b\}$ in the preorder specified by this Hasse diagram. Then both c and d are meets of A: any element less than both a and b is also less than c, and also less than d. Note that, as in Remark 1.77, $c \leq d$ and $d \leq c$, so $c \cong d$. Such will always the case when there is more than one meet: any two meets of the same subset will be isomorphic.

Exercise 1.80. Let $(P, \leq)$ be a preorder and $p \in P$ an element. Consider the set $A = \{p\}$ with one element.

1. Show that $\bigwedge A \cong p$.
2. Show that if P is in fact a partial order, then $\bigwedge A = p$.
3. Are the analogous facts true when $\bigwedge$ is replaced by $\bigvee$? ◇

Example 1.81. In any partial order P, we have $p \vee p = p \wedge p = p$. The reason is that our notation says $p \vee p$ means $\bigvee\{p, p\}$. But $\{p, p\} = \{p\}$ (see Section 1.2.1), so $p \vee p = p$ by Exercise 1.80.

Example 1.82. In a power set $\mathsf{P}(X)$, the meet of a collection of subsets, say $A, B \subseteq X$, is their intersection $A \wedge B = A \cap B$, while the join is their union, $A \vee B = A \cup B$.



Perhaps this justifies the terminology: the joining of two sets is their union, the meeting of two sets is their intersection.

Example 1.83. In the booleans $\mathbb{B} = \{\texttt{false}, \texttt{true}\}$ (Example 1.29), the meet of any two elements is given by AND and the join of any two elements is given by OR (recall Exercise 1.4).

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

Example 1.84. In a total order, the meet of a set is its infimum, while the join of a set is its supremum. Note that $\mathbb{B}$ is a total order, and this generalizes Example 1.83.

Exercise 1.85. Recall the division ordering on $\mathbb{N}$ from Example 1.40: we write $n|m$ if n divides perfectly into m. The meet of any two numbers in this preorder has a common name, which you may have learned when you were around 10 years old; what is it? Similarly the join of any two numbers has a common name; what is it? ◇

Proposition 1.86. Suppose $(P, \leq)$ is a preorder and $A \subseteq B \subseteq P$ are subsets that have meets. Then $\bigwedge B \leq \bigwedge A$.

Similarly, if A and B have joins, then $\bigvee A \leq \bigvee B$.

Proof. Let $m = \bigwedge A$ and $n = \bigwedge B$. Then for any $a \in A$ we also have $a \in B$, so $n \leq a$ because n is a lower bound for B. Thus n is also a lower bound for A and hence $n \leq m$, because m is A's greatest lower bound. The second claim is proved similarly. □

1.3.2 Back to Observations and Generative Effects

In his thesis [Ada17], Adam thinks of monotone maps as observations. A monotone map $\Phi\colon P \to Q$ is a phenomenon (we might say "feature") of P as observed by Q. Adam defines the *generative effect* of such a map Φ to be its failure to preserve joins (or more generally, for categories, its failure to preserve colimits).

Definition 1.87. We say that a monotone map $f\colon P \to Q$ *preserves meets* if $f(a \wedge b) \cong f(a) \wedge f(b)$ for all $a, b \in P$. We similarly say f *preserves joins* if $f(a \vee b) \cong f(a) \vee f(b)$ for all $a, b \in P$.

Definition 1.88. We say that a monotone map $f\colon P \to Q$ *has a generative effect* if there exist elements $a, b \in P$ such that

$$f(a) \vee f(b) \ncong f(a \vee b).$$

In Definition 1.88, if we think of Φ as an observation or measurement of the systems a and b, then the left-hand side $f(a) \vee f(b)$ may be interpreted as the combination of the observation of a with the observation of b. On the other hand, the right-hand side $f(a \vee b)$ is the observation of the combined system $a \vee b$. The inequality implies that we see something when we observe the combined system that we could not expect by merely combining our observations of the pieces. That is, that there are generative effects from the interconnection of the two systems.

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

Exercise 1.89. In Definition 1.88, we defined generativity of f as the inequality $f(a \vee b) \neq f(a) \vee f(b)$, but in the subsequent text we seemed to imply there would be not just a difference, but *more stuff* in $f(a \vee b)$ than in $f(a) \vee f(b)$.

Prove that for any monotone map $f\colon P \to Q$, if $a, b \in P$ have a join and $f(a), f(b) \in Q$ have a join, then indeed $f(a) \vee f(b) \leq f(a \vee b)$. ◇

In his work on generative effects, Adam restricts his attention to generative maps that preserve meets (but do not preserve joins). The preservation of meets implies that the map Φ behaves well when restricting to subsystems, even though it can throw up surprises when joining systems.

This discussion naturally leads into Galois connections, which are pairs of monotone maps between preorders, one of which preserves all joins and the other of which preserves all meets.

1.4 Galois Connections

The preservation of meets and joins, and in particular issues concerning generative effects, is tightly related to the theory of *Galois connections*, which is a special case of a more general theory we will discuss later, namely that of *adjunctions*. We will use some adjunction terminology when describing Galois connections.

1.4.1 Definition and Examples of Galois Connections

Galois connections between preorders were first considered by Évariste Galois – who didn't call them by that name – in the context of a connection he found between "field extensions" and "automorphism groups." We will not discuss this further, but the idea is that, given two preorders P and Q, a Galois connection is a pair of maps back and forth – from P to Q and from Q to P – with certain properties, which make it like a relaxed version of isomorphisms. To be a bit more precise, preorder isomorphisms are examples of Galois connections, but Galois connections need not be preorder isomorphisms.

Definition 1.90. A *Galois connection* between preorders P and Q is a pair of monotone maps $f\colon P \to Q$ and $g\colon Q \to P$ such that

$$f(p) \leq q \quad \text{if and only if} \quad p \leq g(q). \tag{1.6}$$

We say that f is the *left adjoint* and g is the *right adjoint* of the Galois connection.

Example 1.91. Consider the map $(3 \times -)\colon \mathbb{Z} \to \mathbb{R}$ which sends $x \in \mathbb{Z}$ to $3x$, which we can consider as a real number $3x \in \mathbb{Z} \subseteq \mathbb{R}$. Let's find a left adjoint for the map $(3 \times -)$.

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

Write $\lceil z \rceil$ for the smallest natural number above $z \in \mathbb{R}$, and write $\lfloor z \rfloor$ for the largest integer below $z \in \mathbb{R}$, e.g. $\lceil 3.14 \rceil = 4$ and $\lfloor 3.14 \rfloor = 3$.[9] As the left adjoint $\mathbb{R} \to \mathbb{Z}$, let's see if $\lceil -/3 \rceil$ works.

It is easily checked that

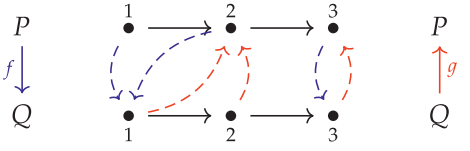
$$\lceil x/3 \rceil \leq y \text{ if and only if } x \leq 3y.$$

Success! Thus we have a Galois connection between $\lceil -/3 \rceil$ and $(3 \times -)$.

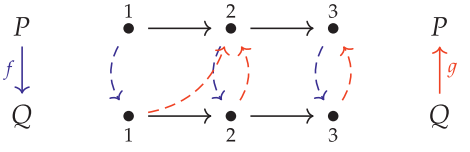
Exercise 1.92. In Example 1.91 we found a left adjoint for the monotone map $(3 \times -) \colon \mathbb{Z} \to \mathbb{R}$. Now find a right adjoint for the same map, and show it is correct. ◇

Exercise 1.93. Consider the preorder $P = Q = \underline{3}$.

1. Let f, g be the monotone maps shown below:



   Is it the case that f is left adjoint to g? Check that for each $1 \leq p, q \leq 3$, one has $f(p) \leq q$ iff $p \leq g(q)$.
2. Let f, g be the monotone maps shown below:



   Is it the case that f is left adjoint to g? ◇

Remark 1.94. The diagrams in Exercise 1.93 suggest the following idea. If P and Q are total orders and $f \colon P \to Q$ and $g \colon Q \to P$ are drawn with arrows bending counterclockwise, then f is left adjoint to g iff the arrows do not cross. With a little bit of thought, this can be formalized. We think this is a pretty neat way of visualizing Galois connections between total orders!

Exercise 1.95.

1. Does $\lceil -/3 \rceil$ have a left adjoint $L \colon \mathbb{Z} \to \mathbb{R}$?
2. If not, why? If so, does its left adjoint have a left adjoint? ◇

[9] By "above" and "below," we mean *greater than or equal to* or *less than or equal to*; the latter being a mouthful. Anyway, $\lfloor 3 \rfloor = 3 = \lceil 3 \rceil$.

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

1.4.2 Back to Partitions

Recall from Example 1.47 that we can understand the set $\mathsf{Prt}(S)$ of partitions on a set S in terms of surjective functions out of S.

Suppose we are given any function $g\colon S \to T$. We will show that this function g induces a Galois connection $g_!\colon \mathsf{Prt}(S) \leftrightarrows \mathsf{Prt}(T) : g^*$ between the preorder of S-partitions and the preorder of T-partitions. The way you might explain it to a seasoned category theorist is:

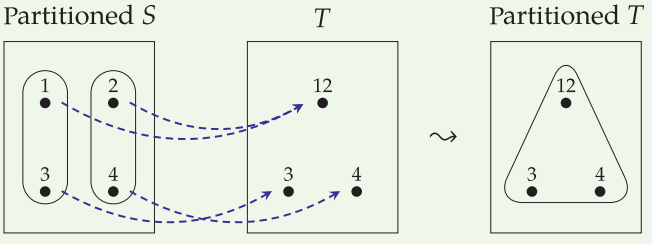
> The left adjoint is given by taking any surjection out of S and pushing out along g to get a surjection out of T. The right adjoint is given by taking any surjection out of T, composing with g to get a function out of S, and then taking the epi-mono factorization to get a surjection out of S.

By the end of this book, the reader will understand pushouts and epi-mono factorizations, so he or she will be able to make sense of the above statement. But for now we will explain the process in more down-to-earth terms.

Start with $g\colon S \to T$; we first want to understand $g_!\colon \mathsf{Prt}(S) \to \mathsf{Prt}(T)$. So start with a partition $\sim_S$ of S. To begin the process of obtaining a partition $\sim_T$ on T, say that two elements $t_1, t_2 \in T$ are in the same part, $t_1 \sim_T t_2$, if there exist $s_1, s_2 \in S$ such that $s_1 \sim_S s_2$ and $g(s_1) = t_1$ and $g(s_2) = t_2$. However, the result of doing so will not necessarily be transitive – you may get $t_1 \sim_T t_2$ and $t_2 \sim_T t_3$ without $t_1 \overset{?}{\sim}_T t_3$ – and partitions must be transitive. So complete the process by just adding in the missing pieces (take the transitive closure). The result is $g_!(\sim_S) \coloneqq \sim_T$.

Again starting with g, we want to get the right adjoint $g^*\colon \mathsf{Prt}(T) \to \mathsf{Prt}(S)$. So start with a partition $\sim_T$ of T. Get a partition $\sim_S$ on S by saying that $s_1 \sim_S s_2$ iff $g(s_1) \sim_T g(s_2)$. The result is $g^*(\sim_T) \coloneqq \sim_S$.

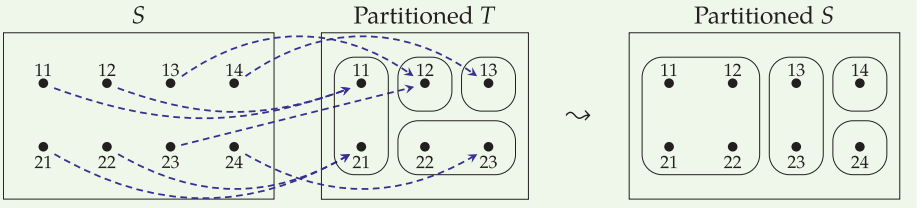
Example 1.96. Let $S = \{1, 2, 3, 4\}$, $T = \{12, 3, 4\}$, and define $g\colon S \to T$ by $g(1) \coloneqq g(2) \coloneqq 12$, $g(3) \coloneqq 3$, and $g(4) \coloneqq 4$. The partition shown left below is translated by $g_!$ to the partition shown on the right.



https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

Exercise 1.97. There are 15 different partitions of a set with four elements. Choose four different ones and, for each one, call it $c\colon S \twoheadrightarrow P$, find $g_!(c)$, where S, T, and $g\colon S \to T$ are the same as they were in Example 1.96. ◇

Example 1.98. Let S, T be as below, and let $g\colon S \to T$ be the function shown in blue. Here is a picture of how g^* takes a partition on T and "pulls it back" to a partition on S:



Exercise 1.99. There are five partitions possible on a set with three elements, say $T = \{12, 3, 4\}$. Using the same S and $g\colon S \to T$ as in Example 1.96, determine the partition $g^*(c)$ on S for each of the five partitions $c\colon T \twoheadrightarrow P$. ◇

To check that, for any function $g\colon S \to T$, the monotone map $g_!\colon \mathsf{Prt}(S) \to \mathsf{Prt}(T)$ really is left adjoint to $g^*\colon \mathsf{Prt}(T) \to \mathsf{Prt}(S)$ would take too much time for this sketch. But the following exercise gives some evidence.

Exercise 1.100. Let S, T, and $g\colon S \to T$ be as in Example 1.96.

1. Choose a nontrivial partition $c\colon S \twoheadrightarrow P$ and let $g_!(c)$ be its push forward partition on T.
2. Choose any coarser partition $d\colon T \twoheadrightarrow P'$, i.e. where $g_!(c) \leq d$.
3. Choose any non-coarser partition $e\colon T \twoheadrightarrow Q$, i.e. where $g_!(c) \not\leq e$. (If you can't do this, revise your answer for #1.)
4. Find $g^*(d)$ and $g^*(e)$.
5. The adjunction formula Eq. (1.6) in this case says that since $g_!(c) \leq d$ and $g_!(c) \not\leq e$, we should have $c \leq g^*(d)$ and $c \not\leq g^*(e)$. Show that this is true. ◇

1.4.3 Basic Theory of Galois Connections

Proposition 1.101. Suppose that $f\colon P \to Q$ and $g\colon Q \to P$ are monotone maps. The following are equivalent:

(a) f and g form a Galois connection where f is left adjoint to g,
(b) for every $p \in P$ and $q \in Q$ we have

$$p \leq g(f(p)) \qquad \text{and} \qquad f(g(q)) \leq q. \tag{1.7}$$

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

Proof. Suppose f is left adjoint to g. Take any $p \in P$, and let $q \coloneqq f(p)$. By reflexivity, we have $f(p) \leq q$, so by Definition 1.90 of Galois connection we have $p \leq g(q)$, but this means $p \leq g(f(p))$. The proof that $f(g(q)) \leq q$ is similar.

Now suppose that Eq. (1.7) holds for all $p \in P$ and $q \in Q$. We want to show that $f(p) \leq q$ iff $p \leq g(q)$. Suppose $f(p) \leq q$; then since g is monotonic, $g(f(p)) \leq g(q)$, but $p \leq g(f(p))$ so $p \leq g(q)$. The proof that $p \leq g(q)$ implies $f(p) \leq q$ is similar. □

Exercise 1.102. Complete the proof of Proposition 1.101 by showing that

1. if f is left adjoint to g then for any $q \in Q$, we have $f(g(q)) \leq q$,
2. if Eq. (1.7) holds, then $p \leq g(q)$ iff $f(p) \leq q$ holds, for all $p \in P$ and $q \in Q$. ◇

If we replace $\leq$ with $=$ in Eq. (1.7), we get back the definition of isomorphism (Definition 1.70); this is why we said at the beginning of Section 1.4.1 that Galois connections are a kind of relaxed version of isomorphisms.

Exercise 1.103.

1. Show that if $f\colon P \to Q$ has a right adjoint g, then it is unique up to isomorphism. That means, for any other right adjoint g', we have $g(q) \cong g'(q)$ for all $q \in Q$.
2. Is the same true for left adjoints? That is, if $h\colon P \to Q$ has a left adjoint, is it necessarily unique up to isomorphism? ◇

Proposition 1.104 (Right adjoints preserve meets)**.** Let $f\colon P \to Q$ be left adjoint to $g\colon Q \to P$. Suppose that $A \subseteq Q$ is any subset, and let $g(A) \coloneqq \{g(a) \mid a \in A\}$ be its image. Then if A has a meet $\bigwedge A \in Q$ then $g(A)$ has a meet $\bigwedge g(A)$ in P, and we have

$$g\left(\bigwedge A\right) \cong \bigwedge g(A).$$

That is, right adjoints preserve meets. Similarly, left adjoints preserve joins: if $A \subseteq P$ is any subset that has a join $\bigvee A \in P$, then $f(A)$ has a join $\bigvee f(A)$ in Q, and we have

$$f\left(\bigvee A\right) \cong \bigvee f(A).$$

Proof. Let $f\colon P \to Q$ and $g\colon Q \to P$ be adjoint monotone maps, with g right adjoint to f. Let $A \subseteq Q$ be any subset and let $m \coloneqq \bigwedge A$ be its meet. Then since g is monotone $g(m) \leq g(a)$ for all $a \in A$, so $g(m)$ is a lower bound for the set $g(A)$. We will be done if we can show $g(m)$ is a greatest lower bound.

So take any other lower bound b for $g(A)$; that is, suppose that for all $a \in A$ we have $b \leq g(a)$ and we want to show that $b \leq g(m)$. Then by definition of g being a right adjoint (Definition 1.90), we also have $f(b) \leq a$. This means that $f(b)$ is a lower bound for A in Q. Since the meet m is the greatest lower bound, we have $f(b) \leq m$.

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

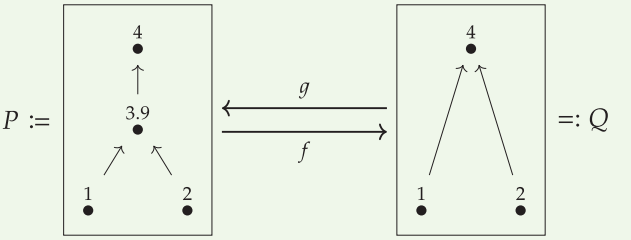
Once again using the Galois connection, $b \leq g(m)$, proving that $g(m)$ is indeed the greatest lower bound for $g(A)$, as desired.

The second claim is proved similarly; see Exercise 1.105. □

Exercise 1.105. Complete the proof of Proposition 1.104 by showing that left adjoints preserve joins. ◇

Since left adjoints preserve joins, we know that they cannot have generative effects. In fact, we shall see in Theorem 1.108 that a monotone map does not have generative effects – i.e. it preserves joins – if and only if it is a left adjoint to some other monotone.

Example 1.106. Right adjoints need not preserve joins. Here is an example:



Let g be the map that preserves labels, and let f be the map that preserves labels as far as possible but with $f(3.9) \coloneqq 4$. Both f and g are monotonic, and one can check that g is right adjoint to f (see Exercise 1.107). But g does not preserve joins because $1 \vee 2 = 4$ holds in Q, whereas $g(1) \vee g(2) = 1 \vee 2 = 3.9 \neq 4 = g(4)$ in P.

Exercise 1.107. To be sure that g really is right adjoint to f in Example 1.106, there are twelve tiny things to check; do so. That is, for every $p \in P$ and $q \in Q$, check that $f(p) \leq q$ iff $p \leq g(q)$. ◇

Theorem 1.108 (Adjoint functor theorem for preorders)**.** Suppose Q is a preorder that has all meets, and let P be any preorder. A monotone map $g\colon Q \to P$ preserves meets if and only if it is a right adjoint.

Similarly, if P has all joins and Q is any preorder, a monotone map $f\colon P \to Q$ preserves joins if and only if it is a left adjoint.

Proof. We will prove only the claim about meets; the claim about joins follows similarly.

We proved one direction in Proposition 1.104, namely that right adjoints preserve meets. For the other, suppose that g is a monotone map that preserves meets; we will construct a left adjoint f. We define our candidate $f\colon P \to Q$ on any $p \in P$ by

$$f(p) \coloneqq \bigwedge \{q \in Q \mid p \leq g(q)\}; \tag{1.8}$$

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

this meet is well defined because Q has all meets, but for f to really be a candidate, we need to show it is monotone. So suppose that $p \leq p'$. Then $\{q' \in Q \mid p' \leq g(q')\} \subseteq \{q \in Q \mid p \leq g(q)\}$. By Proposition 1.86, this implies $f(p) \leq f(p')$. Thus f is monotone.

By Proposition 1.104, it suffices to show that $p_0 \leq g(f(p_0))$ and that $f(g(q_0)) \leq q_0$ for all $p_0 \in P$ and $q_0 \in Q$. For the first, we have

$$p_0 \leq \bigwedge \{g(q) \in P \mid p_0 \leq g(q)\} \cong g\left(\bigwedge \{q \in Q \mid p_0 \leq g(q)\}\right) = g(f(p_0)),$$

where the first inequality follows from the fact that if p_0 is below every element of a set, then it is below their meet, and the isomorphism is by definition of g preserving meets. For the second, we have

$$f(g(q_0)) = \bigwedge \{q \in Q \mid g(q_0) \leq g(q)\} \leq \bigwedge \{q_0\} = q_0,$$

where the first inequality follows from Proposition 1.86 since $\{q_0\} \subseteq \{q \in Q \mid g(q_0) \leq g(q)\}$, and the fact that $\bigwedge \{q_0\} = q_0$. □

Example 1.109. Let $f\colon A \to B$ be a function between sets. We can imagine A as a set of apples, B as a set of buckets, and f as putting each apple in a bucket.

Then we have the monotone map $f^*\colon \mathsf{P}(B) \to \mathsf{P}(A)$ that category theorists call "pull-back along f." This map takes a subset $B' \subseteq B$ to its preimage $f^{-1}(B') \subseteq A$: that is, it takes a collection B' of buckets, and tells you all the apples that they contain in total. This operation is monotonic (more buckets means more apples) and it has both a left and a right adjoint. (There are two different adjunctions here involving f^*.)

The left adjoint $f_!(A)$ is given by the direct image: it maps a subset $A' \subseteq A$ to

$$f_!(A') \coloneqq \{b \in B \mid \text{there exists } a \in A' \text{ such that } f(a) = b\}.$$

This map takes a set A' of apples, and tells you all the buckets that contain at least one of those apples.

The right adjoint f_* maps a subset $A' \subseteq A$ to

$$f_*(A') \coloneqq \{b \in B \mid \text{for all } a \text{ such that } f(a) = b, \text{ we have } a \in A'\}.$$

This map takes a set A' of apples, and tells you all the buckets b that are all-A': all the apples in b are from the chosen subset A'. Note that if a bucket doesn't contain any apples at all, then vacuously all its apples are from A', so empty buckets count as far as f_* is concerned.

Notice that all three of these operations turn out to be interesting: start with a set B' of buckets and return all the apples in them, or start with a set A' of apples and find either the buckets that contain at least one apple from A', or the buckets whose only apples are from A'. But we did not invent these mappings f^*, $f_!$, and f_*: they were *induced* by the function f. They were automatic. It is one of the pleasures of category theory that adjoints so often turn out to have interesting semantic interpretations.

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

Exercise 1.110. Choose sets X and Y with between two and four elements each, and choose a function $f : X \to Y$.

1. Choose two different subsets $B_1, B_2 \subseteq Y$ and find $f^*(B_1)$ and $f^*(B_2)$.
2. Choose two different subsets $A_1, A_2 \subseteq X$ and find $f_!(A_1)$ and $f_!(A_2)$.
3. With the same $A_1, A_2 \subseteq X$, find $f_*(A_1)$ and $f_*(A_2)$. ◇

1.4.4 Closure Operators

Given a Galois connection with $f : P \to Q$ left adjoint to $g : Q \to P$, we may compose f and g to arrive at a monotone map $f \fatsemi g : P \to P$ from preorder P to itself. This monotone map has the property that $p \leq (f \fatsemi g)(p)$, and that $(f \fatsemi g \fatsemi f \fatsemi g)(p) \cong (f \fatsemi g)(p)$ for any $p \in P$. This is an example of a *closure operator*.[10]

Exercise 1.111. Suppose that f is left adjoint to g. Use Proposition 1.101 to show the following.

1. $p \leq (f \fatsemi g)(p)$.
2. $(f \fatsemi g \fatsemi f \fatsemi g)(p) \cong (f \fatsemi g)(p)$. To prove this, show inequalities in both directions, $\leq$ and $\geq$. ◇

Definition 1.112. A *closure operator* $j : P \to P$ on a preorder P is a monotone map such that for all $p \in P$ we have

(a) $p \leq j(p)$,
(b) $j(j(p)) \cong j(p)$.

Example 1.113. Here is an example of closure operators from computation, very roughly presented. Imagine computation as a process of rewriting input expressions to output expressions. For example, a computer can rewrite the expression `7 + 2 + 3` as the expression `12`. The set of arithmetic expressions has a partial order according to whether one expression can be rewritten as another.

We might think of a computer program, then, as a method of taking an expression and reducing it to another expression. So it is a map $j : \texttt{exp} \to \texttt{exp}$. It furthermore is desirable to require that this computer program is a closure operator. Monotonicity means that if an expression x can be rewritten into expression y, then the reduction $j(x)$ can be rewritten into $j(y)$. Moreover, the requirement $x \leq j(x)$ implies that j can only turn one expression into another if doing so is a permissible rewrite. The requirement $j(j(x)) = j(x)$ implies that if you try to reduce an expression that has already been reduced, the computer program leaves it as is. These properties provide useful structure in the analysis of program semantics.

[10] The other composite $g \fatsemi f$ satisfies the dual properties: $(g \fatsemi f)(q) \leq q$ and $(g \fatsemi f \fatsemi g \fatsemi f)(q) \cong (g \fatsemi f)(q)$ for all $q \in Q$. This is called an *interior operator*, though we will not discuss this concept further.

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

Example 1.114 (Adjunctions from closure operators)**.** Just as every adjunction gives rise to a closure operator, from every closure operator we may construct an adjunction.

Let P be a preorder and let $j\colon P \to P$ be a closure operator. We can define a preorder fix_j to have elements the fixed points of j; that is,

$$\mathrm{fix}_j \coloneqq \{p \in P \mid j(p) \cong p\}.$$

This is a subset of P and inherits an order as a result; hence fix_j is a sub-preorder of P. Note that $j(p)$ is a fixed point for all $p \in P$, since $j(j(p)) \cong j(p)$.

We define an adjunction with left adjoint $j\colon P \to \mathrm{fix}_j$ sending p to $j(p)$, and right adjoint $g\colon \mathrm{fix}_j \to P$ simply the inclusion of the sub-preorder. To see it's really an adjunction, we need to see that for any $p \in P$ and $q \in \mathrm{fix}_j$, we have $j(p) \leq q$ if and only if $p \leq q$. Let's check it. Since $p \leq j(p)$, we have that $j(p) \leq q$ implies $p \leq q$ by transitivity. Conversely, since q is a fixed point, $p \leq q$ implies $j(p) \leq j(q) \cong q$.

Example 1.115. Another example of closure operators comes from logic. This will be discussed in the final chapter of the book, in particular Section 7.4.5, but we will give a quick overview here. In essence, logic is the study of when one formal statement – or proposition – implies another. For example, if n is prime then n is not a multiple of 6, or if it is raining then the ground is getting wetter. Here "n is prime," "n is not a multiple of 6," "it is raining," and "the ground is getting wetter" are propositions, and we gave two implications.

Take the set of all propositions and order them by $p \leq q$ iff p implies q, denoted $p \Rightarrow q$. Since $p \Rightarrow p$ and since whenever $p \Rightarrow q$ and $q \Rightarrow r$, we also have $p \Rightarrow r$, this is indeed a preorder.

A closure operator on it is often called a *modal operator*. It is a function j from propositions to propositions, for which $p \Rightarrow j(p)$ and $j(j(p)) = j(p)$. An example of a j is "assuming Bob is in San Diego…." Think of this as a proposition B; so "assuming Bob is in San Diego, p" means $B \Rightarrow p$. Let's see why $B \Rightarrow -$ is a closure operator.

If p is true then "assuming Bob is in San Diego, p" is still true. Suppose that "assuming Bob is in San Diego it is the case that, 'assuming Bob is in San Diego, p' is true." It follows that "assuming Bob is in San Diego, p" is true. So we have seen, at least informally, that "assuming Bob is in San Diego …" is a closure operator.

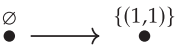
1.4.5 Level Shifting

The last thing we want to discuss in this chapter is a phenomenon that happens often in category theory, something we might informally call "level shifting." It is easier to give an example of this than to explain it directly.

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

Given any set S, there is a set $\mathbf{Rel}(S)$ of binary relations on S. An element $R \in \mathbf{Rel}(S)$ is formally a subset $R \subseteq S \times S$. The set $\mathbf{Rel}(S)$ can be given an order via the subset relation, $R \subseteq R'$, i.e. if whenever $R(s_1, s_2)$ holds then so does $R'(s_1, s_2)$.

For example, the Hasse diagram for $\mathbf{Rel}(\{1\})$ is:



Exercise 1.116. Draw the Hasse diagram for the preorder $\mathbf{Rel}(\{1, 2\})$ of all binary relations on the set $\{1, 2\}$. ◇

For any set S, there is also a set $\mathbf{Pos}(S)$, consisting of all the preorder relations on S. In fact there is a preorder structure $\sqsubseteq$ on $\mathbf{Pos}(S)$, again given by inclusion: $\leq$ is below $\leq'$ (we'll write $\leq \sqsubseteq \leq'$) if $a \leq b$ implies $a \leq' b$ for every $a, b \in S$. A preorder of preorder structures? That's what we mean by a level shift.

Every preorder relation is – in particular – a relation, so we have an inclusion $\mathbf{Pos}(S) \to \mathbf{Rel}(S)$. This is the right adjoint of a Galois connection. Its left adjoint is a monotone map $\mathrm{Cl}\colon \mathbf{Rel}(S) \to \mathbf{Pos}(S)$ given by taking any relation R, writing it in infix notation using $\leq$, and taking the reflexive and transitive closure, i.e. adding $s \leq s$ for every s and adding $s \leq u$ whenever $s \leq t$ and $t \leq u$.

Exercise 1.117. Let $S = \{1, 2, 3\}$. Let's try to understand the adjunction discussed above.

1. Come up with any preorder relation $\leq$ on S, and define $U(\leq)$ to be the subset $U(\leq) := \{(s_1, s_2) \mid s_1 \leq s_2\} \subseteq S \times S$, i.e. $U(\leq)$ is the image of $\leq$ under the inclusion $\mathbf{Pos}(S) \to \mathbf{Rel}(S)$, the relation "underlying" the preorder.
2. Come up with any two binary relations $Q \subseteq S \times S$ and $Q' \subseteq S \times S$ such that $Q \subseteq U(\leq)$ but $Q' \not\subseteq U(\leq)$. Note that your choice of Q, Q' do not have to come from preorders.

We now want to check that, in this case, the closure operation $\mathtt{Cl}$ is really left adjoint to the "underlying relation" map U.

3. Concretely (without using the assertion that there is some sort of adjunction), show that $\mathtt{Cl}(Q) \sqsubseteq \leq$, where $\sqsubseteq$ is the order on $\mathbf{Pos}(S)$, defined immediately above this exercise.
4. Concretely show that $\mathtt{Cl}(Q') \not\sqsubseteq \leq$. ◇

1.5 Summary and Further Reading

In this first chapter, we set the stage for category theory by introducing one of the simplest interesting sorts of example: preorders. From this seemingly simple structure, a bunch of further structure emerges: monotone maps, meets, joins, and more. In terms of modeling real-world phenomena, we thought of preorders as the states of a system,

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press

and monotone maps as describing a way to use one system to observe another. From this point of view, generative effects occur when observations of the whole cannot be deduced by combining observations of the parts.

In the final section we introduced Galois connections. A Galois connection, or adjunction, is a pair of maps that are like inverses, but allowed to be more "relaxed" by getting the orders involved. Perhaps surprisingly, it turns out adjunctions are closely related to joins and meets: if a preorder P has all joins, then a monotone map out of P is a left adjoint if and only if it preserves joins; similarly for meets and right adjoints.

The next two chapters build significantly on this material, but in two different directions. Chapter 2 adds a new operation on the underlying set: it introduces the idea of a monoidal structure on preorders. This allows us to construct an element $a \otimes b$ of a preorder P from any elements $a, b \in P$, in a way that respects the order. On the other hand, Chapter 3 adds new structure on the order itself: it introduces the idea of a morphism, which describes not only whether $a \leq b$, but gives a name f for how a relates to b. This structure is known as a category. These generalizations are both fundamental to the story of compositionality, and in Chapter 4 we'll see them meet in the concept of a monoidal category. The lessons we have learned in this chapter will illuminate the more highly structured generalizations in the chapters to come. Indeed, it is a useful principle in studying category theory to try to understand concepts first in the setting of preorders – where often much of the complexity is stripped away and one can develop some intuition – before considering the general case.

But perhaps you might be interested in exploring some ideas in this chapter in other directions. While we won't return to them in this book, we learned about generative effects from Elie Adam's thesis [Ada17], and a much richer treatment of generative effect can be found there. In particular, he discusses abelian categories and cohomology, providing a way to detect generative effects in quite a general setting.

Another important application of preorders, monotone maps, and Galois connections is found in the analysis of programming languages. In this setting, preorders describe the possible states of a computer, and monotone maps describe the action of programs, or relationships between different ways of modeling computation states. Galois connections are useful for showing how different models may be closely related, and for transporting program analysis from one framework to another. For more detail on this, see Chapter 4 of the textbook [NNH99].

https://doi.org/10.1017/9781108668804.002 Published online by Cambridge University Press