Chapter 2

Resource theories: Monoidal preorders and enrichment

2.1 Getting from a to b

You can't make an omelette without breaking an egg. To obtain the things we want requires resources, and the process of transforming what we have into what we want is often an intricate one. In this chapter, we will discuss how monoidal preorders can help us think about this matter.

Consider the following three questions you might ask yourself:

- Given what I have, is it *possible* to get what I want?
- Given what I have, what is the *minimum cost* to get what I want?
- Given what I have, what is the *set of ways* to get what I want?

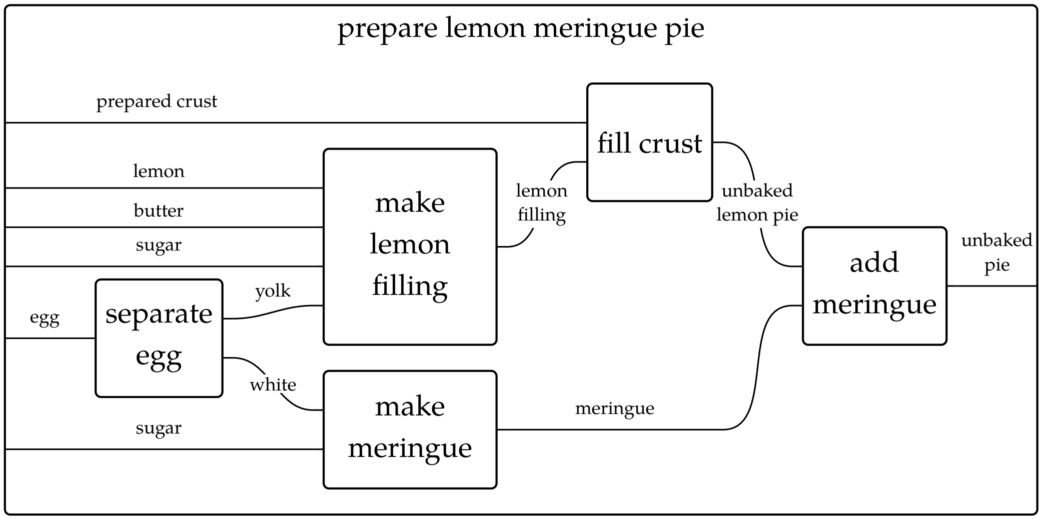
These questions are about resources—those you have and those you want—but perhaps more importantly, they are about moving from have to want: possibility of, cost of, and ways to.

Such questions come up not only in our lives, but also in science and industry. In chemistry, one asks whether a certain set of compounds can be transformed into another set, how much energy such a reaction will require, or what methods exist for making it happen. In manufacturing, one asks similar questions.

From an external point of view, both a chemist and an industrial firm might be regarded as store-houses of information on the above subjects. The chemist knows which compounds she can make given other ones, and how to do so; the firm has stored knowledge of the same sort. The research work of the chemist and the firm is to use what they know in order to derive—or discover—new knowledge.

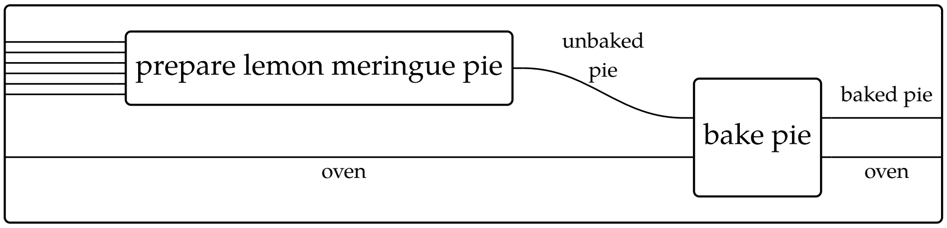
This is roughly the first goal of this chapter: to discuss a formalism for expressing recipes—methods for transforming one set of resources into another—and for deriving new recipes from old. The idea here is not complicated, neither in life nor in our mathematical formalism. The value added then is to simply see how it works, so we can build on it within the book, and so others can build on it in their own work.

We briefly discuss the categorical approach to this idea—namely that of *monoidal preorders*—for building new recipes from old. The following *wiring diagram* shows, assuming one knows how to implement each of the interior boxes, how to implement the preparation of a lemon meringue pie:



(2.1)

The wires show resources: we start with prepared crust, lemon, butter, sugar, and egg resources, and we end up with an unbaked pie resource. We could take this whole method and combine it with others, e.g. baking the pie:



In the above example we see that resources are not always consumed when they are used. For example, we use an oven to convert—or catalyze the transformation of—an unbaked pie into a baked pie, and we get the oven back after we are done. It's a nice feature of ovens! To use economic terms, the oven is a "means of production" for pies.

String diagrams are important mathematical objects that will come up repeatedly in this book. They were invented in the mathematical context—more specifically in the context of monoidal categories—by Joyal and Street [JS93], but they have been used less formally by engineers and scientists in various contexts for a long time.

As we said above, our first goal in this chapter is to use monoidal preorders, and the corresponding wiring diagrams, as a formal language for recipes from old. Our second goal is to discuss something called $\mathcal{V}$-categories for various monoidal preorders $\mathcal{V}$.

A $\mathcal{V}$-category is a set of objects, which one may think of as points on a map, where $\mathcal{V}$ somehow "structures the question" of getting from point a to point b. The examples of monoidal preorders $\mathcal{V}$ that we will be most interested in are called **Bool** and **Cost**. Roughly speaking, a **Bool**-category is a set of points where the question of getting from point a to point b has a `true / false` answer. A **Cost**-category is a set of points where the question of getting from a to b has an answer $d \in [0, \infty]$, a cost.

This story works in more generality than monoidal preorders. Indeed, in Chapter 4 we will discuss something called a monoidal category, a notion which generalizes monoidal preorders, and we will generalize the definition of $\mathcal{V}$-category accordingly. In this more general setting, $\mathcal{V}$-categories can also address our third question above, describing *methods* of getting between points. For example a **Set**-category is a set of points where the question of getting from point a to point b has a set of answers (elements of which might be called methods).

We will begin in Section 2.2 by defining symmetric monoidal preorders, giving a few preliminary examples, and discussing wiring diagrams. We then give many more examples of symmetric monoidal preorders, including both some real-world examples, in the form of resource theories, and some mathematical examples that will come up again throughout the book. In Section 2.3 we discuss enrichment and $\mathcal{V}$-categories—how a monoidal preorder $\mathcal{V}$ can "structure the question" of getting from a to b—and then give some important constructions on $\mathcal{V}$-categories (Section 2.4), and analyze them using a sort of matrix multiplication technique (Section 2.5).

2.2 Symmetric monoidal preorders

In Section 1.2.2 we introduced preorders. The notation for a preorder, namely $(X, \leq)$, refers to two pieces of structure: a set called X and a relation called $\leq$ that is reflexive and transitive.

We want to add to the concept of preorders a way of combining elements in X, an operation taking two elements and adding or multiplying them together. However, the operation does not have to literally be addition or multiplication; it only needs to satisfy some of the properties one expects from them.

2.2.1 Definition and first examples

We begin with a formal definition of symmetric monoidal preorders.

Definition 2.2. A *symmetric monoidal structure* on a preorder $(X, \leq)$ consists of two constituents:

(i) an element $I \in X$, called the *monoidal unit*, and

(ii) a function $\otimes \colon X \times X \to X$, called the *monoidal product*.

These constituents must satisfy the following properties, where we write $\otimes(x_1, x_2) = x_1 \otimes x_2$:

(a) for all $x_1, x_2, y_1, y_2 \in X$, if $x_1 \leq y_1$ and $x_2 \leq y_2$, then $x_1 \otimes x_2 \leq y_1 \otimes y_2$,
(b) for all $x \in X$, the equations $I \otimes x = x$ and $x \otimes I = x$ hold,
(c) for all $x, y, z \in X$, the equation $(x \otimes y) \otimes z = x \otimes (y \otimes z)$ holds, and
(d) for all $x, y \in X$, the equation $x \otimes y = y \otimes x$ holds.

We call these conditions *monotonicity*, *unitality*, *associativity*, and *symmetry* respectively. A preorder equipped with a symmetric monoidal structure, $(X, \leq, I, \otimes)$, is called a *symmetric monoidal preorder*.

Anyone can propose a set X, an order $\leq$ on X, an element I in X, and a binary operation $\otimes$ on X and ask whether $(X, \leq, I, \otimes)$ is a symmetric monoidal preorder. And it will indeed be one, as long as it satisfies rules a, b, c, and d of Definition 2.2.

Remark 2.3. It is often useful to replace = with ≅ throughout Definition 2.2. The result is a perfectly good notion, called a *weak monoidal structure*. The reason we chose equality is that it makes equations look simpler, which we hope aids first-time readers.

The notation for the monoidal unit and the monoidal product may vary: monoidal units we have seen include I (as in the definition), 0, 1, `true`, `false`, $\{*\}$, and more. Monoidal products we have seen include $\otimes$ (as in the definition), $+$, $*$, $\wedge$, $\vee$, and $\times$. The *preferred notation* in a given setting is whatever best helps our brains remember what we're trying to do; the names I and $\otimes$ are just defaults.

Example 2.4. There is a well-known preorder structure, denoted $\leq$, on the set $\mathbb{R}$ of real numbers; e.g. $-5 \leq \sqrt{2}$. We propose 0 as a monoidal unit and $+ \colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ as a monoidal product. Does $(\mathbb{R}, \leq, 0, +)$ satisfy the conditions of Definition 2.2?

If $x_1 \leq y_1$ and $x_2 \leq y_2$, it is true that $x_1 + x_2 \leq y_1 + y_2$. It is also true that $0 + x = x$ and $x + 0 = x$, that $(x + y) + z = x + (y + z)$, and that $x + y = y + x$. Thus $(\mathbb{R}, \leq, 0, +)$ satisfies the conditions of being a symmetric monoidal preorder.

Exercise 2.5. Consider again the preorder $(\mathbb{R}, \leq)$ from Example 2.4. Someone proposes 1 as a monoidal unit and $*$ (usual multiplication) as a monoidal product. But an expert walks by and says "that won't work." Figure out why, or prove the expert wrong! ◊

Example 2.6. A *monoid* consists of a set M, a function $* \colon M \times M \to M$ called the *monoid multiplication*, and an element $e \in M$ called the *monoid unit*, such that, when you write $*(m, n)$ as $m * n$, i.e. using infix notation, the equations

$$m * e = m, \qquad e * m = m, \qquad (m * n) * p = m * (n * p) \tag{2.7}$$

hold for all $m, n, p \in M$. It is called *commutative* if also $m * n = n * m$.

Every set S determines a discrete preorder $\mathbf{Disc}_S$ (where $m \leq n$ iff $m = n$; see Example 1.32), and it is easy to check that if $(M, e, *)$ is a commutative monoid then $(\mathbf{Disc}_M, =, e, *)$ is a symmetric monoidal preorder.

Exercise 2.8. We said it was easy to check that if $(M, *, e)$ is a commutative monoid then $(\mathbf{Disc}_M, =, *, e)$ is a symmetric monoidal preorder. Are we telling the truth? ◊

Example 2.9. Here is a non-example for people who know the game "standard poker." Let H be the set of all poker hands, where a hand means a choice of five cards from the standard 52-card deck. As an order, put $h \leq h'$ if h' beats or equals h in poker.

One could propose a monoidal product $\otimes \colon H \times H \to H$ by assigning $h_1 \otimes h_2$ to be "the best hand one can form out of the ten cards in h_1 and h_2." If some cards are in both h_1 and h_2, just throw the duplicates away. So for example $\{2\heartsuit, 3\heartsuit, 4\heartsuit, 6\spadesuit, 7\spadesuit\} \otimes \{2\heartsuit, 5\heartsuit, 6\heartsuit, 6\spadesuit, 7\spadesuit\} = \{2\heartsuit, 3\heartsuit, 4\heartsuit, 5\heartsuit, 6\heartsuit\}$, because the latter is the best hand you can make with the former two.

This proposal for a monoidal structure will fail the condition (a) of Definition 2.2: it could be the case that $h_1 \leq i_1$ and $h_2 \leq i_2$, and yet *not* be the case that $h_1 \otimes h_2 \leq i_1 \otimes i_2$. For example, consider this case:

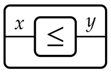
$$h_1 := \{2\heartsuit, 3\heartsuit, 10\spadesuit, \mathrm{J}\spadesuit, \mathrm{Q}\spadesuit\} \qquad i_1 := \{4\clubsuit, 4\spadesuit, 6\heartsuit, 6\diamondsuit, 10\diamondsuit\}$$
$$h_2 := \{2\diamondsuit, 3\diamondsuit, 4\diamondsuit, \mathrm{K}\spadesuit, \mathrm{A}\spadesuit\} \qquad i_2 := \{5\spadesuit, 5\heartsuit, 7\heartsuit, J\diamondsuit, Q\diamondsuit\}.$$

Here, $h_1 \leq i_1$ and $h_2 \leq i_2$, but $h_1 \otimes h_2 = \{10\spadesuit, \mathrm{J}\spadesuit, \mathrm{Q}\spadesuit, \mathrm{K}\spadesuit, \mathrm{A}\spadesuit\}$ is the best possible hand and beats $i_1 \otimes i_2 = \{5\spadesuit, 5\heartsuit, 6\heartsuit, 6\diamondsuit, Q\diamondsuit\}$.

Subsections 2.2.3 and 2.2.4 are dedicated to examples of symmetric monoidal preorders. Some are aligned with the notion of resource theories, others come from pure math. When discussing the former, we will use wiring diagrams, so here is a quick primer.

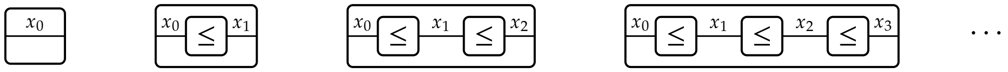
2.2.2 Introducing wiring diagrams

Wiring diagrams are visual representations for building new relationships from old. In a preorder without a monoidal structure, the only sort of relationship between objects is $\leq$, and the only way you build a new $\leq$ relationship from old ones is by chaining them together. We denote the relationship $x \leq y$ by



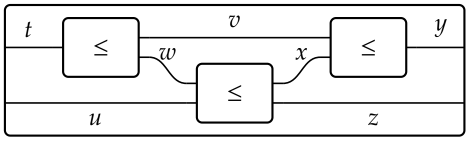
(2.10)

We can chain some number of these $\leq$-relationships—say 0, 1, 2, or 3 of them—together in series as shown here



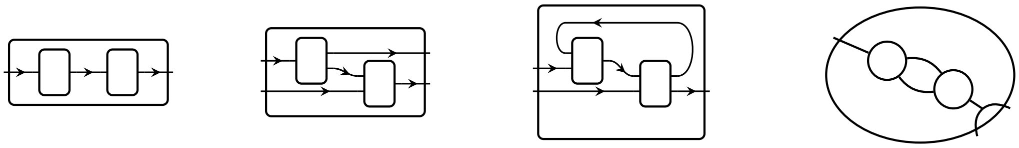
(2.11)

If we add a symmetric monoidal structure, we can combine relationships not only in series but also in parallel. Here is an example:



(2.12)

Different styles of wiring diagrams In fact, we will see later that there are many styles of wiring diagrams. When we are dealing with preorders, the sort of wiring diagram we can draw is that with single-input, single-output boxes connected in series. When we are dealing with symmetric monoidal preorders, we can have more complex boxes and more complex wiring diagrams, including parallel composition. Later we will see that for other sorts of categorical structures, there are other styles of wiring diagrams:

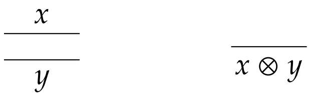


(2.13)

Wiring diagrams for symmetric monoidal preorders The style of wiring diagram that makes sense in any symmetric monoidal preorder is that shown in Eq. (2.12): boxes can have multiple inputs and outputs, and they may be arranged in series and parallel. Symmetric monoidal preorders and their wiring diagrams are tightly coupled with each other. How so?

The answer is that a monoidal preorder $(X, \leq, I, \otimes)$ has some notion of element $(x \in X)$, relationship $(\leq)$, and combination (using transitivity and $\otimes$), and so do wiring diagrams: the wires represent elements, the boxes represent relationships, and the wiring diagrams themselves show how relationships can be combined. We call boxes and wires *icons*; we will encounter several more icons in this chapter, and throughout the book.

To get a bit more rigorous about the connection, let's start with a monoidal preorder $(X, \leq, I, \otimes)$ as in Definition 2.2. Wiring diagrams have wires on the left and the right. Each element $x \in X$ can be made the label of a wire. Note that given two objects x, y, we can either draw two wires in parallel—one labeled x and one labeled y—or we can draw one wire labeled $x \otimes y$.

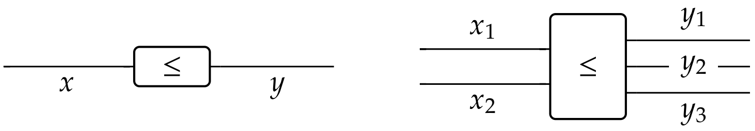


We consider wires in parallel to represent the monoidal product of their labels, so we consider both cases above to represent the element $x \otimes y$. Note also that a wire labeled

I or an absence of wires:

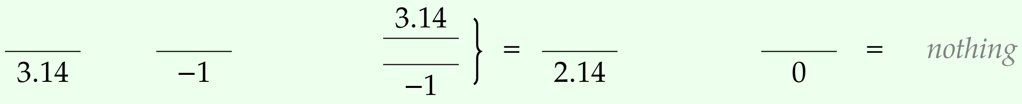
both represent the monoidal unit I; another way of thinking of this is that the unit is the empty monoidal product.

A wiring diagram runs between a set of parallel wires on the left and a set of parallel wires on the right. We say that a wiring diagram is *valid* if the monoidal product of the elements on the left is less than the monoidal product of those on the right. For example, if we have the inequality $x \leq y$, the the diagram that is a box with a wire labeled x on the left and a wire labeled y on the right is valid; see the first box below:

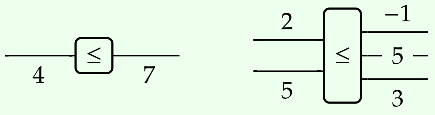


The validity of the second box corresponds to the inequality $x_1 \otimes x_2 \leq y_1 \otimes y_2 \otimes y_3$. Before going on to the properties from Definition 2.2, let us pause for an example of what we've discussed so far.

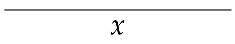
Example 2.14. Recall the symmetric monoidal preorder $(\mathbb{R}, \leq, 0, +)$ from Example 2.4. The wiring diagrams for it allow wires labeled by real numbers. Drawing wires in parallel corresponds to adding their labels, and the wire labeled 0 is equivalent to no wires at all.



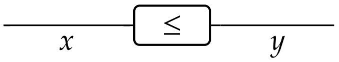
And here we express a couple facts about $(\mathbb{R}, \leq, 0, +)$ in this language: $4 \leq 7$ and $2 + 5 \leq -1 + 5 + 3$.



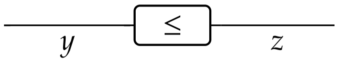
We now return to how the properties of symmetric monoidal preorders correspond to properties of this sort of wiring diagram. Let's first talk about the order structure: conditions (a)—reflexivity—and (b)—transitivity—from Definition 1.30. Reflexivity says that $x \leq x$, this means the diagram just consisting of a wire



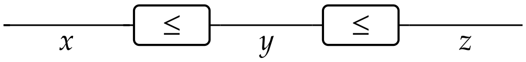
is always valid. Transitivity allows us to connect facts together: it says that if $x \leq y$ and $y \leq z$, then $x \leq z$. This means that if the diagrams



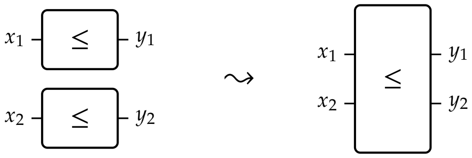
and



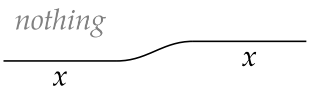
are valid, we can put them together and obtain the valid diagram



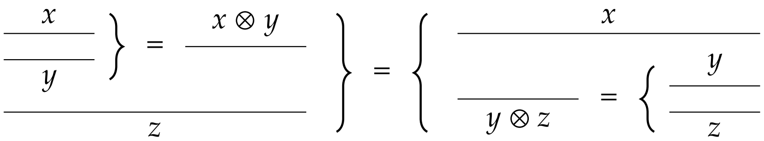
Next let's talk about the properties (a)–(d) from the definition of symmetric monoidal structure (Definition 2.2). Property (a) says that if $x_1 \leq y_1$ and $x_2 \leq y_2$ then $x_1 \otimes x_2 \leq y_1 \otimes y_2$. This corresponds to the idea that stacking any two valid boxes in parallel is still valid:



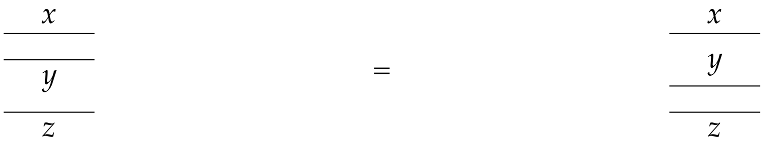
Condition (b), that $I \otimes x = x$ and $x \otimes I = x$, says we don't need to worry about I or blank space; in particular diagrams such as the following are valid:



Condition (c), that $(x \otimes y) \otimes z = x \otimes (y \otimes z)$ says that we don't have to worry about whether we build up diagrams from the top or from the bottom

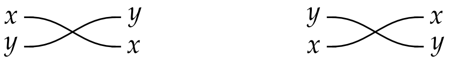


But this looks much harder than it is: the associative property should be thought of as saying that there is no distinction between the stuff on the very left above and the stuff on the very right, i.e.



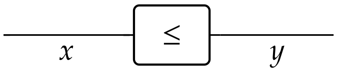
and indeed a diagram that moves from one to the other is valid.

Finally, the symmetry condition (d), that $x \otimes y = y \otimes x$, says that a diagram is valid even if its wires cross:

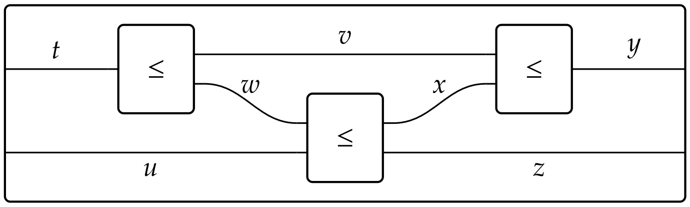


One may regard the pair of crossing wires as another icon in our iconography, in addition to the boxes and wires we already have.

Wiring diagrams as graphical proofs Given a monoidal preorder $\mathcal{X} = (X, \leq, I, \otimes)$, a wiring diagram is a graphical proof of something about $\mathcal{X}$. Each box in the diagram has a left side and a right side, say x and y, and represents the assertion that $x \leq y$.



A wiring diagram is a bunch of interior boxes connected together inside an exterior box. It represents a graphical proof that says: if all of the interior assertions are correct, then so is the exterior assertion.



(2.15)

The inner boxes in Eq. (2.15) translate into the assertions:

$$t \leq v + w \qquad w + u \leq x + z \qquad v + x \leq y \tag{2.16}$$

and the outer box translates into the assertion:

$$t + u \leq y + z. \tag{2.17}$$

The whole wiring diagram 2.15 says "if you know that the assertions in 2.16 are true, then I am a proof that the assertion in 2.17 is also true." What exactly is the proof that diagram 2.15 represents? It is the proof

$$t + u \;\leq\; v + w + u \;\leq\; v + x + z \;\leq\; y + z. \tag{2.18}$$

Indeed, each inequality here is a vertical slice of the diagram 2.15, and the transitivity of these inequalities is expressed by connecting these vertical slices together.

Example 2.19. Recall the lemon meringue pie wiring diagram from Eq. (2.1). It has five interior boxes, such as "separate egg" and "fill crust," and it has one exterior box called "prepare lemon meringue pie." Each box is the assertion that, given the collection of resources on the left, say an egg, you can transform it into the collection of resources on the right, say an egg white and an egg yolk. The whole string diagram is a proof that if each of the interior assertions is true—i.e. you really do know how to separate eggs, make lemon filling, make meringue, fill crust, and add meringue—then the exterior assertion is true: you can prepare a lemon meringue pie.

Exercise 2.20. The string of inequalities in Eq. (2.18) is not quite a proof, because technically there is no such thing as $v + w + u$, for example. Instead, there is $(v + w) + u$ and $v + (w + u)$, and so on.

1. Formally prove, using only the rules of symmetric monoidal preorders (Definition 2.2), that given the assertions in Eq. (2.16), the conclusion in Eq. (2.17) follows.
2. Reflexivity and transitivity should show up in your proof. Make sure you are explicit about where they do.
3. How can you look at the wiring diagram Eq. (2.12) and know that the symmetry axiom (Definition 2.2(d)) does not need to be invoked? ◊

We next discuss some examples of symmetric monoidal preorders. We begin in Section 2.2.3 with some more concrete examples, from science, commerce, and informatics. Then in Section 2.2.4 we discuss some examples arising from pure math, some of which will get a good deal of use later on, e.g. in Chapter 4.

2.2.3 Applied examples

Resource theories are studies of how resources are exchanged in a given arena. For example, in social resource theory one studies a marketplace where combinations of goods can be traded for—as well as converted into—other combinations of goods.

Whereas marketplaces are very dynamic, and an apple might be tradable for an orange on Sunday but not on Monday, what we mean by resource theory in this chapter is a static notion: deciding "what buys what," once and for all.[1] This sort of static notion of conversion might occur in chemistry: the chemical reactions that are possible one day will quite likely be possible on a different day as well. Manufacturing may be somewhere in between: the set of production techniques—whereby a company can convert one set of resources into another—do not change much from day to day.

We learned about resource theories from [CFS16; Fri17], who go much further than we will; see Section 2.6 for more information. In this section we will focus only on the main idea. While there are many beautiful mathematical examples of symmetric monoidal preorders, as we will see in Section 2.2.4, there are also ad hoc examples coming from life experience. In the next chapter, on databases, we will see the same theme: while there are some beautiful mathematical categories out there, database schemas are *ad hoc* organizational patterns of information. Describing something as "ad hoc" is often considered derogatory, but it just means "formed, arranged, or done for a particular purpose only." There is nothing wrong with doing things for a particular purpose; it's common outside of pure math and pure art. Let's get to it.

Chemistry In high school chemistry, we work with chemical equations, where material collections such as

$$H_2O, \quad NaCl, \quad 2NaOH, \quad CH_4 + 3O_2$$

[1] Using some sort of temporal theory, e.g. the one presented in Chapter 7, one could take the notion here and have it change in time.

are put together in the form of reaction equations, such as

$$2H_2O + 2Na \longrightarrow 2NaOH + H_2.$$

The collection on the left, $2H_2O + 2Na$ is called the *reactant*, and the collection on the right, $2NaOH + H_2$ is called the *product*.

We can consider reaction equations such as the one above as taking place inside a single symmetric monoidal preorder $(\mathrm{Mat}, \to, 0, +)$. Here Mat is the set of all collections of atoms and molecules, sometimes called *materials*. So we have $NaCl \in \mathrm{Mat}$ and $4H_2O + 6Ne \in \mathrm{Mat}$.

The set Mat has a preorder structure denoted by the $\to$ symbol, which is the preferred symbol in the setting of chemistry. To be clear, $\to$ is taking the place of the order relation $\leq$ from Definition 2.2. The $+$ symbol is the preferred notation for the monoidal product in the chemistry setting, taking the place of $\otimes$. While it does not come up in practice, we use 0 to denote the monoidal unit.

Exercise 2.21. Here is an exercise for people familiar with reaction equations: check that conditions (a), (b), (c), and (d) of Definition 2.2 hold. ◊

An important notion in chemistry is that of catalysis: one compound *catalyzes* a certain reaction. For example, one might have the following set of reactions:

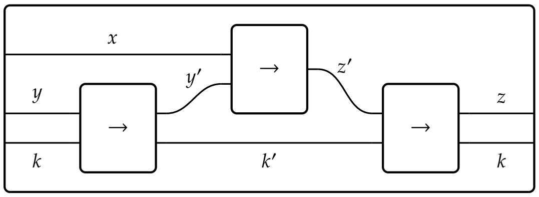
$$y + k \to y' + k' \qquad x + y' \to z' \qquad z' + k' \to z + k \tag{2.22}$$

Using the laws of monoidal preorders, we obtain the composed reaction

$$x + y + k \to x + y' + k' \to z' + k' \to z + k. \tag{2.23}$$

Here k is the catalyst because it is found both in the reactant and the product of the reaction. It is said to catalyze the reaction $x + y \to z$. The idea is that the reaction $x + y \to z$ cannot take place given the reactions in Eq. (2.22). But if k is present, meaning if we add k to both sides, the resulting reaction can take place.

The wiring diagram for the reaction in Eq. (2.23) is shown in Eq. (2.24). The three interior boxes correspond to the three reactions given in Eq. (2.22), and the exterior box corresponds to the composite reaction $x + y + k \to z + k$.



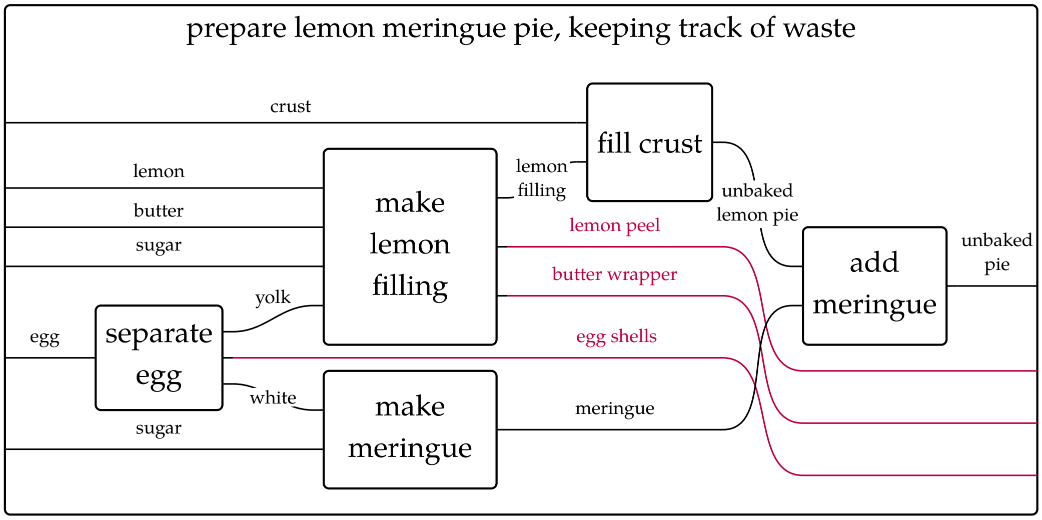
(2.24)

Manufacturing Whether we are talking about baking pies, building smart phones, or following pharmaceutical recipes, manufacturing firms need to store basic recipes, and build new recipes by combining simpler recipes in schemes like the one shown in Eq. (2.1) or Eq. (2.24).

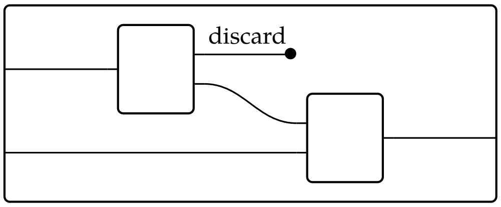
The basic idea in manufacturing is exactly the same as that for chemistry, except there is an important assumption we can make in manufacturing that does not hold for chemical reactions:

You can trash anything you want, and it disappears from view.

This simple assumption has caused the world some significant problems, but it is still in effect. In our meringue pie example, we can ask: "what happened to the egg shell, or the paper wrapping the stick of butter"? The answer is they were trashed, i.e. thrown in the garbage bin. It would certainly clutter our diagram and our thinking if we had to carry these resources through the diagram:



Instead, in our daily lives and in manufacturing, we do not have to hold on to something if we don't need it; we can just discard it. In terms of wiring diagrams, this can be shown using a new icon —●, as follows:



(2.25)

To model this concept of waste using monoidal categories, one just adds an additional axiom to (a), (b), (c), and (d) from Definition 2.2:

(e) $x \leq I$ for all $x \in X$. (discard axiom)

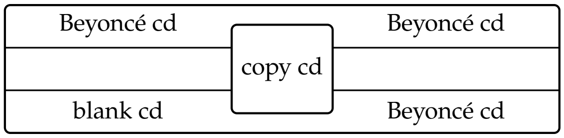
It says that every x can be converted into the monoidal unit I. In the notation of the chemistry section, we would write instead $x \to 0$: any x yields nothing. But this is certainly not accepted in the chemistry setting. For example,

$$H_2O + NaCl \to^? H_2O$$

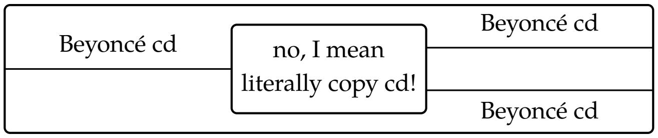
is certainly not a legal chemical equation. It is easy to throw things away in manufacturing, because we assume that we have access to—the ability to grab onto and directly manipulate—each item produced. In chemistry, when you have 10^{23} of substance A dissolved in something else, you cannot just simply discard A. So axiom (e) is valid in manufacturing but not in chemistry.

Recall that in Section 2.2.2 we said that there were many different styles of wiring diagrams. Now we're saying that adding the discard axiom changes the wiring diagram style, in that it adds this new discard icon that allows wires to terminate, as shown in Eq. (2.25). In informatics, we will change the wiring diagram style yet again.

Informatics A major difference between information and a physical object is that information can be copied. Whereas one cup of butter never becomes two, it is easy for a single email to be sent to two different people. It is much easier to copy a music file than it is to copy a CD. Here we do not mean "copy the information from one compact disc onto another"—of course that's easy—instead, we mean that it's quite difficult to copy the physical disc, thereby forming a second physical disc! In diagrams, the distinction is between the relation



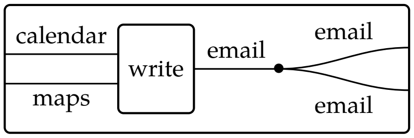
and the relation



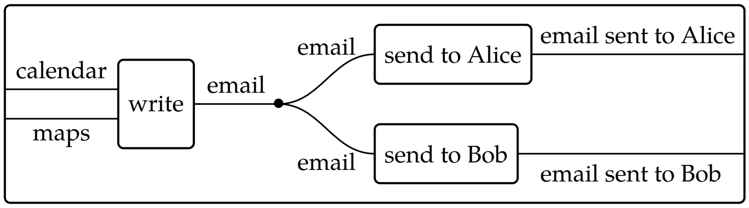
The former is possible, the latter is magic.

Of course material objects can sometimes be copied; cell mitosis is a case in point. But this is a remarkable biological process, certainly not something that is expected for ordinary material objects. In the physical world, we would make mitosis a box transforming one cell into two. But in (classical, not quantum) information, everything can be copied, so we add a new icon to our repertoire.

Namely, in wiring diagram notation, copying information appears as a new icon, -•<, allowing us to split wires:



Now with two copies of the email, we can send one to Alice and one to Bob.



(2.26)

Information can also be discarded, at least in the conventional way of thinking, so in addition to axioms (a) to (d) from Definition 2.2, we can keep axiom (e) from page 50 and add a new *copy axiom*:

(f) $x \leq x + x$ for all $x \in X$. (copy axiom)

allowing us to make mathematical sense of diagrams like Eq. (2.26).

Now that we have examples of monoidal preorders under our belts, let's discuss some nice mathematical examples.

2.2.4 Abstract examples

In this section we discuss several mathematical examples of symmetric monoidal structures on preorders.

The Booleans The simplest nontrivial preorder is the booleans: $\mathbb{B} = \{\texttt{true}, \texttt{false}\}$ with $\texttt{false} \leq \texttt{true}$. There are two different symmetric monoidal structures on it.

Example 2.27 (Booleans with AND). We can define a monoidal structure on $\mathbb{B}$ by letting the monoidal unit be `true` and the monoidal product be $\wedge$ (AND). If one thinks of $\texttt{false} = 0$ and $\texttt{true} = 1$, then $\wedge$ corresponds to the usual multiplication operation $*$. That is, with this correspondence, the two tables below match up:

∧	false	true
false	false	false
true	false	true

*	0	1
0	0	0
1	0	1

(2.28)

One can check that all the properties in Definition 2.2 hold, so we have a monoidal preorder which we denote $\textbf{Bool} := (\mathbb{B}, \leq, \texttt{true}, \wedge)$.

Bool will be important when we get to the notion of enrichment. Enriching in a monoidal preorder $\mathcal{V} = (V, \leq, I, \otimes)$ means "letting $\mathcal{V}$ structure the question of getting from a to b." All of the structures of a monoidal preorder—i.e. the set V, the ordering relation $\leq$, the monoidal unit I, and the monoidal product $\otimes$—play a role in how enrichment works.

For example, let's look at the case of **Bool** $= (\mathbb{B}, \leq, \texttt{true}, \wedge)$. The fact that its underlying set is $\mathbb{B} = \{\texttt{false}, \texttt{true}\}$ will translate into saying that "getting from a to b is a `true`/`false` question." The fact that `true` is the monoidal unit will translate into saying "you can always get from a to a." The fact that $\wedge$ is the monoidal product will translate into saying "if you can get from a to b AND you can get from b to c then you can get from a to c." Finally, the "if-then" form of the previous sentence is coming from the order relation $\leq$. We will make this more precise in Section 2.3.

We will be able to play the same game with other monoidal preorders, as we will see after we define a monoidal preorder called **Cost** in Example 2.37.

Some other monoidal preorders It is a bit imprecise to call **Bool** "the" boolean monoidal preorder, because there is another monoidal structure on $(\mathbb{B}, \leq)$, which we describe in Exercise 2.29. The first structure, however, seems to be more useful in practice than the second.

Exercise 2.29. Let $(\mathbb{B}, \leq)$ be as above, but now consider the monoidal product to be $\vee$ (OR).

$\vee$	`false`	`true`
`false`	`false`	`true`
`true`	`true`	`true`

max	0	1
0	0	1
1	1	1

What must the monoidal unit be in order to satisfy the conditions of Definition 2.2?

Does it satisfy the rest of the conditions? ◊

In Example 2.30 and Exercise 2.31 we give two different monoidal structures on the preorder $(\mathbb{N}, \leq)$ of natural numbers, where $\leq$ is the usual ordering ($0 \leq 1$ and $5 \leq 16$).

Example 2.30 (Natural numbers with addition). There is a monoidal structure on $(\mathbb{N}, \leq)$ where the monoidal unit is 0 and the monoidal product is $+$, i.e. $6 + 4 = 10$. It is easy to check that $x_1 \leq y_1$ and $x_2 \leq y_2$ implies $x_1 + x_2 \leq y_1 + y_2$, as well as all the other conditions of Definition 2.2.

Exercise 2.31. Show there is a monoidal structure on $(\mathbb{N}, \leq)$ where the monoidal product is $*$, i.e. $6 * 4 = 24$. What should the monoidal unit be? ◊

Example 2.32 (Divisibility and multiplication). Recall from Example 1.45 that there is a "divisibility" order on $\mathbb{N}$: we write $m|n$ to mean that m divides into n without remainder. So $1|m$ for all m and $4|12$.

There is a monoidal structure on $(\mathbb{N}, |\ )$, where the monoidal unit is 1 and the monoidal product is $*$, i.e. $6 * 4 = 24$. Then if $x_1|y_1$ and $x_2|y_2$, then $(x_1 * x_2)|(y_1 * y_2)$. Indeed, if there is some $p_1, p_2 \in \mathbb{N}$ such that $x_1 * p_1 = y_1$ and $x_2 * p_2 = y_2$, then $(p_1 * p_2) * (x_1 * x_2) = y_1 * y_2$.

Exercise 2.33. Again taking the divisibility order $(\mathbb{N}, |\,)$. Someone proposes 0 as the monoidal unit and + as the monoidal product. Does that proposal satisfy the conditions of Definition 2.2? Why or why not? ◊

Exercise 2.34. Consider the preorder $(P, \leq)$ with Hasse diagram $\boxed{\texttt{no} \to \texttt{maybe} \to \texttt{yes}}$. We propose a monoidal structure with `yes` as the monoidal unit and "min" as the monoidal product.

1. Make sense of "min" by filling in the multiplication table with elements of P.

min	no	maybe	yes
no	?	?	?
maybe	?	?	?
yes	?	?	?

2. Check the axioms of Definition 2.2 hold for $\mathbf{NMY} := (P, \leq, \texttt{yes}, \min)$, given your definition of min. If not, change your definition of min. ◊

Exercise 2.35. Let S be a set and let $\mathsf{P}(S)$ be its power set, the set of all subsets of S, including the empty subset, $\varnothing \subseteq S$, and the "everything" subset, $S \subseteq S$. We can give $\mathsf{P}(S)$ an order: $A \leq B$ is given by the subset relation $A \subseteq B$, as discussed in Example 1.50. We propose a symmetric monoidal structure on $\mathsf{P}(S)$ with monoidal unit S and monoidal product given by intersection $A \cap B$.

Does it satisfy the conditions of Definition 2.2? ◊

Exercise 2.36. Let $\texttt{Prop}^{\mathbb{N}}$ denote the set of all mathematical statements one can make about a natural number, where we consider two statements to be the same if one is true if and only if the other is true. For example "n is prime" is an element of $\texttt{Prop}^{\mathbb{N}}$, and so are "$n = 2$" and "$n \geq 11$." The statements "$n + 2 = 5$" and "$n$ is the least odd prime" are considered the same. Given $P, Q \in \texttt{Prop}^{\mathbb{N}}$, we say $P \leq Q$ if for all $n \in \mathbb{N}$, whenever $P(n)$ is true, so is $Q(n)$.

Define a monoidal unit and a monoidal product on $\texttt{Prop}^{\mathbb{N}}$ that satisfy the conditions of Definition 2.2. ◊

The monoidal preorder Cost As we said above, when we enrich in monoidal preorders we see them as different ways to structure the question of "getting from here to there." We will explain this in more detail in Section 2.3. The following monoidal preorder will eventually structure a notion of distance or cost for getting from here to there.

Example 2.37 (Lawvere's monoidal preorder, **Cost**). Let $[0, \infty]$ denote the set of nonnegative real numbers—such as 0, 1, $15.33\overline{3}$, and 2π—together with ∞. Consider the preorder $([0, \infty], \geq)$, with the usual notion of $\geq$, where of course $\infty \geq x$ for all $x \in [0, \infty]$.

There is a monoidal structure on this preorder, where the monoidal unit is 0 and the monoidal product is +. In particular, $x + \infty = \infty$ for any $x \in [0, \infty]$. Let's call this monoidal preorder

$$\mathbf{Cost} := ([0,\infty], \geq, 0, +),$$

because we can think of the elements of $[0,\infty]$ as costs. In terms of structuring "getting from here to there," **Cost** seems to say "getting from a to b is a question of cost." The monoidal unit being 0 will translate into saying that you can always get from a to a at no cost. The monoidal product being $+$ will translate into saying that the cost of getting from a to c is at most the cost of getting from a to b *plus* the cost of getting from b to c. Finally, the "at most" in the previous sentence is coming from the $\geq$.

The opposite of a monoidal preorder One can take the opposite of any preorder, just flip the order: $(X,\leq)^{\text{op}} := (X,\geq)$; see Example 1.58. Proposition 2.38 says that if the preorder had a symmetric monoidal structure, so does its opposite.

Proposition 2.38. Suppose $\mathcal{X} = (X,\leq)$ is a preorder and $\mathcal{X}^{\text{op}} = (X,\geq)$ is its opposite. If $(X,\leq,I,\otimes)$ is a symmetric monoidal preorder then so is its opposite, $(X,\geq,I,\otimes)$.

Proof. Let's first check monotonicity. Suppose $x_1 \geq y_1$ and $x_2 \geq y_2$ in $\mathcal{X}^{\text{op}}$; we need to show that $x_1 \otimes x_2 \geq y_1 \otimes y_2$. But by definition of opposite order, we have $y_1 \leq x_1$ and $y_2 \leq x_2$ in $\mathcal{X}$, and thus $y_1 \otimes y_2 \leq x_1 \otimes x_2$ in $\mathcal{X}$. Thus indeed $x_1 \otimes x_2 \geq y_1 \otimes y_2$ in $\mathcal{X}^{\text{op}}$. The other three conditions are even easier; see Exercise 2.39. □

Exercise 2.39. Complete the proof of Proposition 2.38 by proving that the three remaining conditions of Definition 2.2 are satisfied. ◊

Exercise 2.40. Since **Cost** is a symmetric monoidal preorder, Proposition 2.38 says that $\mathbf{Cost}^{\text{op}}$ is too.

1. What is $\mathbf{Cost}^{\text{op}}$ as a preorder?
2. What is its monoidal unit?
3. What is its monoidal product? ◊

2.2.5 Monoidal monotone maps

Recall from Example 1.49 that for any preorder $(X,\leq)$, there is an induced equivalence relation $\cong$ on X, where $x \cong x'$ iff both $x \leq x'$ and $x' \leq x$.

Definition 2.41. Let $\mathcal{P} = (P, \leq_P, I_P, \otimes_P)$ and $\mathcal{Q} = (Q, \leq_Q, I_Q, \otimes_Q)$ be monoidal preorders. A *monoidal monotone* from $\mathcal{P}$ to $\mathcal{Q}$ is a monotone map $f\colon (P,\leq_P) \to (Q,\leq_Q)$, satisfying two conditions:

(a) $I_Q \leq_Q f(I_P)$, and

(b) $f(p_1) \otimes_Q f(p_2) \leq_Q f(p_1 \otimes_P p_2)$

for all $p_1, p_2 \in P$.

There are strengthenings of these conditions that are also important. If f satisfies the following conditions, it is called a *strong monoidal monotone*:

(a′) $I_Q \cong f(I_P)$, and

(b′) $f(p_1) \otimes_Q f(p_2) \cong f(p_1 \otimes_P p_2)$;

and if it satisfies the following conditions it is called a *strict monoidal monotone*:

(a″) $I_Q = f(I_P)$, and

(b″) $f(p_1) \otimes_Q f(p_2) = f(p_1 \otimes_P p_2)$.

Monoidal monotones are examples of *monoidal functors*, which we will see various incarnations of throughout the book; see Definition 6.68. What we call monoidal monotones could also be called *lax monoidal monotones*, and there is a dual notion of *oplax monoidal monotones*, where the inequalities in (a) and (b) are reversed; we will not use oplaxity in this book.

Example 2.42. There is a monoidal monotone $i\colon (\mathbb{N}, \leq, 0, +) \to (\mathbb{R}, \leq, 0, +)$, where $i(n) = n$ for all $n \in \mathbb{N}$. It is clearly monotonic, $m \leq n$ implies $i(m) \leq i(n)$. It is even strict monoidal because $i(0) = 0$ and $i(m + n) = i(m) + i(n)$.

There is also a monoidal monotone $f\colon (\mathbb{R}, \leq, 0, +) \to (\mathbb{N}, \leq, 0, +)$ going the other way. Here $f(x) := \lfloor x \rfloor$ is the floor function, e.g. $f(3.14) = 3$. It is monotonic because $x \leq y$ implies $f(x) \leq f(y)$. Also $f(0) = 0$ and $f(x) + f(y) \leq f(x + y)$, so it is a monoidal monotone. But it is not strict or even strong because $f(0.5) + f(0.5) \neq f(0.5 + 0.5)$.

Recall **Bool** $= (\mathbb{B}, \leq, \texttt{true}, \wedge)$ from Example 2.27 and **Cost** $= ([0, \infty], \geq, 0, +)$ from Example 2.37. There is a monoidal monotone $g\colon \textbf{Bool} \to \textbf{Cost}$, given by $g(\texttt{false}) := \infty$ and $g(\texttt{true}) := 0$.

Exercise 2.43.

1. Check that the map $g\colon (\mathbb{B}, \leq, \texttt{true}, \wedge) \to ([0, \infty], \geq, 0, +)$ presented above indeed
   - is monotonic,
   - satisfies condition (a) of Definition 2.41, and
   - satisfies condition (b) of Definition 2.41.
2. Is g strict? ◊

Exercise 2.44. Let **Bool** and **Cost** be as above, and consider the following quasi-inverse functions $d, u\colon [0, \infty] \to \mathbb{B}$ defined as follows:

$$d(x) := \begin{cases} \texttt{false} & \text{if } x > 0 \\ \texttt{true} & \text{if } x = 0 \end{cases} \qquad u(x) := \begin{cases} \texttt{false} & \text{if } x = \infty \\ \texttt{true} & \text{if } x < \infty \end{cases}$$

1. Is d monotonic?
2. Does d satisfy conditions (a) and (b) of Definition 2.41?
3. Is d strict?
4. Is u monotonic?
5. Does u satisfy conditions (a) and (b) of Definition 2.41?
6. Is u strict? ◊

Exercise 2.45.

1. Is $(\mathbb{N}, \leq, 1, *)$ a monoidal preorder, where $*$ is the usual multiplication of natural numbers?
2. If not, why not? If so, does there exist a monoidal monotone $(\mathbb{N}, \leq, 0, +) \to (\mathbb{N}, \leq, 1, *)$? If not; why not? If so, find it.
3. Is $(\mathbb{Z}, \leq, *, 1)$ a monoidal preorder? ◊

2.3 Enrichment

In this section we will introduce $\mathcal{V}$-categories, where $\mathcal{V}$ is a symmetric monoidal preorder. We will see that **Bool**-categories are preorders, and that **Cost**-categories are a nice generalization of the notion of metric space.

2.3.1 $\mathcal{V}$-categories

While $\mathcal{V}$-categories can be defined even when $\mathcal{V}$ is not symmetric, i.e. just obeys conditions (a)–(c) of Definition 2.2, certain things don't work quite right. For example, we will see later in Exercise 2.75 that the symmetry condition is necessary in order for products of $\mathcal{V}$-categories to exist. Anyway, here's the definition.

Definition 2.46. Let $\mathcal{V} = (V, \leq, I, \otimes)$ be a symmetric monoidal preorder. A *$\mathcal{V}$-category* $\mathcal{X}$ consists of two constituents, satisfying two properties. To specify $\mathcal{X}$,

(i) one specifies a set $\mathrm{Ob}(\mathcal{X})$, elements of which are called *objects*;

(ii) for every two objects x, y, one specifies an element $\mathcal{X}(x, y) \in V$, called the *hom-object*.[2]

The above constituents are required to satisfy two properties:

(a) for every object $x \in \mathrm{Ob}(\mathcal{X})$ we have $I \leq \mathcal{X}(x, x)$, and

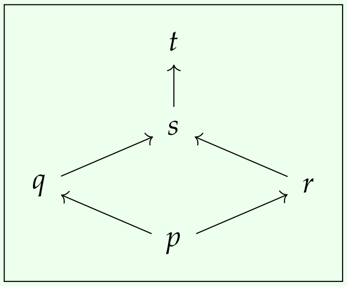
(b) for every three objects $x, y, z \in \mathrm{Ob}(\mathcal{X})$, we have $\mathcal{X}(x, y) \otimes \mathcal{X}(y, z) \leq \mathcal{X}(x, z)$.

We call $\mathcal{V}$ the *base of the enrichment* for $\mathcal{X}$ or say that $\mathcal{X}$ is *enriched* in $\mathcal{V}$.

Example 2.47. As we shall see in the next subsection, from every preorder we can construct a **Bool**-category, and vice versa. So, to get a feel for $\mathcal{V}$-categories, let us

[2] The word "hom" is short for *homomorphism* and reflects the origins of this subject. A more descriptive name for $\mathcal{X}(x, y)$ might be *mapping object*, but we use "hom" mainly because it is an important jargon word to know in the field.

consider the preorder generated by the Hasse diagram:



(2.48)

How does this correspond to a **Bool**-category $\mathcal{X}$? Well, the objects of $\mathcal{X}$ are simply the elements of the preorder, i.e. $\text{Ob}(\mathcal{X}) = \{p, q, r, s, t\}$. Next, for every pair of objects (x, y) we need an element of $\mathbb{B} = \{\texttt{false}, \texttt{true}\}$: simply take `true` if $x \leq y$, and `false` if otherwise. So for example, since $s \leq t$ and $t \not\leq s$, we have $\mathcal{X}(s, t) = \texttt{true}$ and $\mathcal{X}(t, s) = \texttt{false}$. Recalling from Example 2.27 that the monoidal unit I of **Bool** is `true`, it's straightforward to check that this obeys both (a) and (b), so we have a **Bool**-category.

In general, it's sometimes convenient to represent a $\mathcal{V}$-category $\mathcal{X}$ with a square matrix. The rows and columns of the matrix correspond to the objects of $\mathcal{X}$, and the (x, y)th entry is simply the hom-object $\mathcal{X}(x, y)$. So, for example, the above preorder in Eq. (2.48) can be represented by the matrix

↗	p	q	r	s	t
p	true	true	true	true	true
q	false	true	false	true	true
r	false	false	true	true	true
s	false	false	false	true	true
t	false	false	false	false	true

2.3.2 Preorders as Bool-categories

Our colleague Peter Gates has called category theory "a primordial ooze," because so much of it can be defined in terms of other parts of it. There is nowhere to rightly call the beginning, because that beginning can be defined in terms of something else. So be it; this is part of the fun.

Theorem 2.49. There is a one-to-one correspondence between preorders and **Bool**-categories.

Here we find ourselves in the ooze, because we are saying that preorders are the same as **Bool**-categories, whereas **Bool** is itself a preorder. "So then **Bool** is like... enriched in itself?" Yes, every preorder, including **Bool**, is enriched in **Bool**, as we will now see.

Proof of Theorem 2.49. Let's check that we can construct a preorder from any **Bool**-category. Since $\mathbb{B} = \{\texttt{false}, \texttt{true}\}$, Definition 2.46 says a **Bool**-category consists of two things:

(i) a set $\mathrm{Ob}(\mathcal{X})$, and

(ii) for every $x, y \in \mathrm{Ob}(\mathcal{X})$ an element $\mathcal{X}(x, y) \in \mathbb{B}$, i.e. either $\mathcal{X}(x, y) = \texttt{true}$ or $\mathcal{X}(x, y) = \texttt{false}$.

We will use these two things to begin forming a preorder whose elements are the objects of $\mathcal{X}$. So let's call the preorder $(X, \leq)$, and let $X \coloneqq \mathrm{Ob}(\mathcal{X})$. For the $\leq$ relation, let's declare $x \leq y$ iff $\mathcal{X}(x, y) = \texttt{true}$. We have the makings of a preorder, but for it to work, the $\leq$ relation must be reflexive and transitive. Let's see if we get these from the properties guaranteed by Definition 2.46:

(a) for every element $x \in X$ we have $\texttt{true} \leq \mathcal{X}(x, x)$,

(b) for every three elements $x, y, z \in X$ we have $\mathcal{X}(x, y) \wedge \mathcal{X}(y, z) \leq \mathcal{X}(x, z)$.

For $b \in$ **Bool**, if $\texttt{true} \leq b$ then $b = \texttt{true}$, so the first statement says $\mathcal{X}(x, x) = \texttt{true}$, which means $x \leq x$. For the second statement, one can consult Eq. (2.28). Since $\texttt{false} \leq b$ for all $b \in \mathbb{B}$, the only way statement (b) has any force is if $\mathcal{X}(x, y) = \texttt{true}$ and $\mathcal{X}(y, z) = \texttt{true}$, in which case it forces $\mathcal{X}(x, z) = \texttt{true}$. This condition exactly translates as saying that $x \leq y$ and $y \leq z$ implies $x \leq z$. Thus we have obtained reflexivity and transitivity from the two axioms of **Bool**-categories.

In Example 2.47, we constructed a **Bool**-category from a preorder. We leave it to the reader to generalize this example and show that the two constructions are inverses; see Exercise 2.50. □

Exercise 2.50.

1. Start with a preorder $(P, \leq)$, and use it to define a **Bool**-category as we did in Example 2.47. In the proof of Theorem 2.49 we showed how to turn that **Bool**-category back into a preorder. Show that doing so, you get the preorder you started with.
2. Similarly, show that if you turn a **Bool**-category into a preorder using the above proof, and then turn the preorder back into a **Bool**-category using your method, you get the **Bool**-category you started with. ♢

We now discuss a beautiful application of the notion of enriched categories: metric spaces.

2.3.3 Lawvere metric spaces

Metric spaces offer a precise way to describe spaces of points, each pair of which is separated by some distance. Here is the usual definition:

Definition 2.51. A *metric space* (X, d) consists of:

(i) a set X, elements of which are called *points*, and

(ii) a function $d\colon X \times X \to \mathbb{R}_{\geq 0}$, where $d(x, y)$ is called the *distance between x and y*.

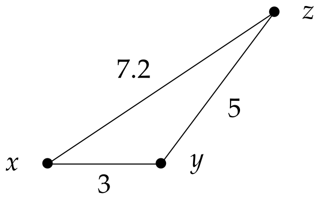
These constituents must satisfy four properties:

(a) for every $x \in X$, we have $d(x, x) = 0$,
(b) for every $x, y \in X$, if $d(x, y) = 0$ then $x = y$,
(c) for every $x, y \in X$, we have $d(x, y) = d(y, x)$, and
(d) for every $x, y, z \in X$, we have $d(x, y) + d(y, z) \geq d(x, z)$.

The fourth property is called the *triangle inequality*.

If we ask instead in (ii) for a function $d \colon X \times X \to [0, \infty] = \mathbb{R}_{\geq 0} \cup \{\infty\}$, we call (X, d) an *extended* metric space.

The triangle inequality says that when plotting a route from x to z, the distance is always at most what you get by choosing an intermediate point y and going $x \to y \to z$.



It can be invoked three different ways in the above picture: $3 + 5 \geq 7.2$, but also $5 + 7.2 \geq 3$ and $3 + 7.2 \geq 5$. Oh yeah, and $5 + 3 \geq 7.2$, $7.2 + 5 \geq 3$ and $7.2 + 3 \geq 5$.

The triangle inequality wonderfully captures something about distance, as does the fact that $d(x, x) = 0$ for any x. However, the other two conditions are not quite as general as we would like. Indeed, there are many examples of things that "should" be metric spaces, but which do not satisfy conditions (b) or (c) of Definition 2.51.

For example, what if we take X to be places in your neighborhood, but instead of measuring distance, you want $d(x, y)$ to measure *effort* to get from x to y. Then if there are any hills, the symmetry axiom, $d(x, y) \overset{?}{=} d(y, x)$, fails: it's easier to get from x downhill to y then to go from y uphill to x.

Another way to find a model that breaks the symmetry axiom is to imagine that the elements of X are not points, but whole regions such as the US, Spain, and Boston. Say that the distance from region A to region B is understood using the setup "I will put you in an arbitrary part of A and you just have to get anywhere in B; what is the distance in the worst-case scenario?" So $d(\text{US}, \text{Spain})$ is the distance from somewhere in the western US to the western tip of Spain: you just have to get into Spain, but you start in the worst possible part of the US for doing so.

Exercise 2.52. Which distance is bigger under the above description, $d(\text{Spain}, \text{US})$ or $d(\text{US}, \text{Spain})$? ◊

This notion of distance, which is strongly related to something called *Hausdorff distance*,[3] will again satisfy the triangle inequality, but it violates the symmetry condition. It also violates another condition, because $d(\text{Boston}, \text{US}) = 0$. No matter where you

[3] The Hausdorff distance gives a metric on the set of all subsets $U \subseteq X$ of a given metric space (X, d).

are in Boston, the distance to the nearest point of the US is 0. On the other hand, $d(\text{US}, \text{Boston}) \neq 0$.

Finally, one can imagine a use for distances that are not finite. In terms of my effort, the distance from here to Pluto is ∞, and it would not be any better if Pluto was still a planet. Similarly, in terms of Hausdorff distance, discussed above, the distance between two regions is often infinite, e.g. the distance between $\{r \in \mathbb{R} \mid r < 0\}$ and $\{0\}$ as subsets of $(\mathbb{R}, d)$ is infinite.

When we drop conditions (b) and (c) and allow for infinite distances, we get the following relaxed notion of metric space, first proposed by Lawvere. Recall the symmetric monoidal preorder **Cost** $= ([0, \infty], \geq, 0, +)$ from Example 2.37.

Definition 2.53. A *Lawvere metric space* is a **Cost**-category.

This is a very compact definition, but it packs a punch. Let's work out what it means, by relating it to the usual definition of metric space. By Definition 2.46, a **Cost**-category $\mathcal{X}$ consists of:

(i) a set $\text{Ob}(\mathcal{X})$,
(ii) for every $x, y \in \text{Ob}(\mathcal{X})$ an element $\mathcal{X}(x, y) \in [0, \infty]$.

Here the set $\text{Ob}(\mathcal{X})$ is playing the role of the set of points, and $\mathcal{X}(x, y) \in [0, \infty]$ is playing the role of distance, so let's write a little translator:

$$X \coloneqq \text{Ob}(\mathcal{X}) \qquad d(x, y) \coloneqq \mathcal{X}(x, y).$$

The properties of a category enriched in **Cost** are:

(a) $0 \geq d(x, x)$ for all $x \in X$, and
(b) $d(x, y) + d(y, z) \geq d(x, z)$ for all $x, y, z \in X$.

Since $d(x, x) \in [0, \infty]$, if $0 \geq d(x, x)$ then $d(x, x) = 0$. So the first condition is equivalent to the first condition from Definition 2.51, namely $d(x, x) = 0$. The second condition is the triangle inequality.

Example 2.54. The set $\mathbb{R}$ of real numbers can be given a metric space structure, and hence a Lawvere metric space structure. Namely $d(x, y) \coloneqq |y - x|$, the absolute value of the difference. So $d(3, 7) = 4$.

Exercise 2.55. Consider the symmetric monoidal preorder $(\mathbb{R}_{\geq 0}, \geq, 0, +)$, which is almost the same as **Cost**, except it does not include ∞. How would you characterize the difference between a Lawvere metric space and a $(\mathbb{R}_{\geq 0}, \geq, 0, +)$-category in the sense of Definition 2.46? ◊

One first defines

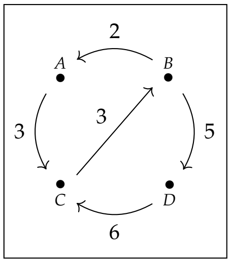
$$d_L(U, V) \coloneqq \sup_{u \in U} \inf_{v \in V} d(u, v),$$

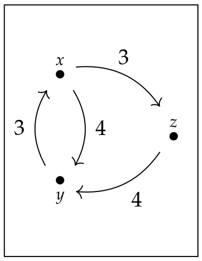
and this is exactly the formula we intend above; the result will be a Lawvere metric space. However, if one wants the Hausdorff distance to define a (symmetric) metric, as in Definition 2.51, one must take the above formula and symmetrize it: $d(U, V) \coloneqq \max(d_L(U, V), d_L(V, U))$. We happen to see the unsymmetrized notion as more interesting.

Presenting metric spaces with weighted graphs Just as one can convert a Hasse diagram into a preorder, one can convert any weighted graph—a graph whose edges are labeled with numbers $w \geq 0$—into a Lawvere metric space. In fact, we shall consider these as graphs labelled with elements of $[0, \infty]$, and more precisely call them **Cost**-weighted graphs.[4]

One might think of a **Cost**-weighted graph as describing a city with some one-way roads (a two-way road is modeled as two one-way roads), each having some effort-to-traverse, which for simplicity we just call length. For example, consider the following weighted graphs:

$X :=$





$=: Y$ (2.56)

Given a weighted graph, one forms a metric d_X on its set X of vertices by setting $d(p, q)$ to be the length of the shortest path from p to q. For example, here is the the table of distances for Y

$d(\nearrow)$	x	y	z
x	0	4	3
y	3	0	6
z	7	4	0

(2.57)

Exercise 2.58. Fill out the following table of distances in the weighted graph X from Eq. (2.56)

$d(\nearrow)$	A	B	C	D
A	0	?	?	?
B	2	?	5	?
C	?	?	?	?
D	?	?	?	?

♢

Above we converted a weighted graph G, e.g. as shown in Eq. (2.56), into a table of distances, but this takes a bit of thinking. There is a more direct construction for taking G and getting a square matrix M_G, whose rows and columns are indexed by the vertices of G. To do so, set M_G to be 0 along the diagonal, to be ∞ wherever an edge is missing, and to be the edge weight if there is an edge.

[4] This generalizes Hasse diagrams, which we could call **Bool**-weighted graphs—the edges of a Hasse diagram are thought of as weighted with `true`; we simply ignore any edges that are weighted with `false`, and neglect to even draw them!

For example, the matrix associated to Y in Eq. (2.56) would be

$$M_Y := \begin{array}{c|ccc} \nearrow & x & y & z \\ \hline x & 0 & 4 & 3 \\ y & 3 & 0 & \infty \\ z & \infty & 4 & 0 \end{array} \tag{2.59}$$

As soon as you see how we did this, you'll understand that it takes no thinking to turn a weighted graph G into a matrix M_G in this way. We will see later in Section 2.5.3 that the more difficult "distance matrices" d_Y, such as Eq. (2.57), can be obtained from the easy graph matrices M_Y, such as Eq. (2.59), by repeating a certain sort of "matrix multiplication."

Exercise 2.60. Fill out the matrix M_X associated to the graph X in Eq. (2.56):

$$M_X = \begin{array}{c|cccc} \nearrow & A & B & C & D \\ \hline A & 0 & ? & ? & ? \\ B & 2 & 0 & \infty & ? \\ C & ? & ? & ? & ? \\ D & ? & ? & ? & ? \end{array}$$ ◊

2.3.4 $\mathcal{V}$-variations on preorders and metric spaces

We have told the story of **Bool** and **Cost**. But in Section 2.2.4 we gave examples of many other monoidal preorders, and each one serves as the base of enrichment for a kind of enriched category. Which of them are useful? Something only becomes useful when someone finds a use for it. We will find uses for some and not others, though we encourage readers to think about what it would mean to enrich in the various monoidal categories discussed above; maybe they can find a use we have not explored.

Exercise 2.61. Recall the monoidal preorder **NMY** $:= (P, \leq, \texttt{yes}, \min)$ from Exercise 2.34. Interpret what a **NMY**-category is. ◊

In the next two exercises, we use $\mathcal{V}$-weighted graphs to construct $\mathcal{V}$-categories. This is possible because we will use preorders that, like **Bool** and **Cost**, have joins.

Exercise 2.62. Let M be a set and let $\mathcal{M} := (\mathsf{P}(M), \subseteq, M, \cap)$ be the monoidal preorder whose elements are subsets of M.

Someone gives the following interpretation, "for any set M, imagine it as the set of modes of transportation (e.g. car, boat, foot). Then an $\mathcal{M}$-category $\mathcal{X}$ tells you all the modes that will get you from a all the way to b, for any two points $a, b \in \mathrm{Ob}(\mathcal{X})$."

1. Draw a graph with four vertices and four or five edges, each labeled with a subset of $M = \{\text{car}, \text{boat}, \text{foot}\}$.
2. From this graph is it possible to construct an $\mathcal{M}$-category, where the hom-object from x to y is computed as follows: for each path p from x to y, take the intersection of the sets labelling the edges in p. Then, take the union of the these

sets over all paths p from x to y. Write out the corresponding four-by-four matrix of hom-objects, and convince yourself that this is indeed an $\mathcal{M}$-category.

3. Does the person's interpretation look right, or is it subtly mistaken somehow? ◊

Exercise 2.63. Consider the monoidal preorder $\mathcal{W} := (\mathbb{N} \cup \{\infty\}, \leq, \infty, \min)$.

1. Draw a small graph labeled by elements of $\mathbb{N} \cup \{\infty\}$.
2. Write out the matrix whose rows and columns are indexed by the nodes in the graph, and whose (x, y)th entry is given by the *maximum* over all paths p from x to y of the *minimum* edge label in p.
3. Prove that this matrix is the matrix of hom-objects for a $\mathcal{W}$-category. This will give you a feel for how $\mathcal{W}$ works.
4. Make up an interpretation, like that in Exercise 2.62, for how to imagine enrichment in $\mathcal{W}$. ◊

2.4 Constructions on $\mathcal{V}$-categories

Now that we have a good intuition for what $\mathcal{V}$-categories are, we give three examples of what can be done with $\mathcal{V}$-categories. The first (Section 2.4.1) is known as change of base. This allows us to use a monoidal monotone $f : \mathcal{V} \to \mathcal{W}$ to construct $\mathcal{W}$-categories from $\mathcal{V}$-categories. The second construction (Section 2.4.2), that of $\mathcal{V}$-functors, allows us to complete the analogy: a preorder is to a **Bool**-category as a monotone map is to what? The third construction (Section 2.4.2) is known as a $\mathcal{V}$-product, and gives us a way of combining two $\mathcal{V}$-categories.

2.4.1 Changing the base of enrichment

Any monoidal monotone $\mathcal{V} \to \mathcal{W}$ between symmetric monoidal preorders lets us convert $\mathcal{V}$-categories into $\mathcal{W}$-categories.

Construction 2.64. Let $f : \mathcal{V} \to \mathcal{W}$ be a monoidal monotone. Given a $\mathcal{V}$-category $\mathcal{C}$, one forms the associated $\mathcal{W}$-category, say $\mathcal{C}_f$ as follows.

(i) We take the same objects: $\mathrm{Ob}(\mathcal{C}_f) := \mathrm{Ob}(\mathcal{C})$.

(ii) For any $c, d \in \mathrm{Ob}(\mathcal{C})$, put $\mathcal{C}_f(c, d) := f(\mathcal{C}(c, d))$.

This construction $\mathcal{C}_f$ does indeed obey the definition of a $\mathcal{W}$-category, as can be seen by applying Definition 2.41 (of monoidal monotone) and Definition 2.46 (of $\mathcal{V}$-category):

(a) for every $c \in \mathcal{C}$, we have

$$\begin{aligned} I_W &\leq f(I_V) && (f \text{ is monoidal monotone}) \\ &\leq f(\mathcal{C}(c, c)) && (\mathcal{C} \text{ is } \mathcal{V}\text{-category}) \\ &= \mathcal{C}_f(c, c) && (\text{definition of } \mathcal{C}_f) \end{aligned}$$

(b) for every $c, d, e \in \mathrm{Ob}(\mathcal{C})$ we have

$$\begin{aligned}
\mathcal{C}_f(c,d) \otimes_W \mathcal{C}_f(d,e) &= f(\mathcal{C}(c,d)) \otimes_W f(\mathcal{C}(d,e)) && \text{(definition of } \mathcal{C}_f\text{)}\\
&\leq f\big(\mathcal{C}(c,d) \otimes_V \mathcal{C}(d,e)\big) && \text{(} f \text{ is monoidal monotone)}\\
&\leq f(\mathcal{C}(c,e)) && \text{(} \mathcal{C} \text{ is } \mathcal{V}\text{-category)}\\
&= \mathcal{C}_f(c,e) && \text{(definition of } \mathcal{C}_f\text{)}
\end{aligned}$$

Example 2.65. As an example, consider the function $f\colon [0,\infty] \to \{\mathtt{true}, \mathtt{false}\}$ given by

$$f(x) := \begin{cases} \mathtt{true} & \text{if } x = 0 \\ \mathtt{false} & \text{if } x > 0 \end{cases} \tag{2.66}$$

It is easy to check that f is monotonic and that f preserves the monoidal product and monoidal unit; that is, it's easy to show that f is a monoidal monotone. (Recall Exercise 2.44.)

Thus f lets us convert Lawvere metric spaces into preorders.

Exercise 2.67. Recall the "regions of the world" Lawvere metric space from Exercise 2.52 and the text above it. We just learned that, using the monoidal monotone f in Eq. (2.66), we can convert it to a preorder. Draw the Hasse diagram for the preorder corresponding to the regions: US, Spain, and Boston. How could you interpret this preorder relation? ◊

Exercise 2.68.

1. Find another monoidal monotone $g\colon$ **Cost** $\to$ **Bool** different from the one defined in Eq. (2.66).
2. Using Construction 2.64, both your monoidal monotone g and the monoidal monotone f in Eq. (2.66) can be used to convert a Lawvere metric space into a preorder. Find a Lawvere metric space $\mathcal{X}$ on which they give different answers, $\mathcal{X}_f \neq \mathcal{X}_g$. ◊

2.4.2 Enriched functors

The notion of functor provides the most important type of relationship between categories.

Definition 2.69. Let $\mathcal{X}$ and $\mathcal{Y}$ be $\mathcal{V}$-categories. A *$\mathcal{V}$-functor from $\mathcal{X}$ to $\mathcal{Y}$*, denoted $F\colon \mathcal{X} \to \mathcal{Y}$, consists of one constituent:

(i) a function $F\colon \mathrm{Ob}(\mathcal{X}) \to \mathrm{Ob}(\mathcal{Y})$

subject to one constraint

(a) for all $x_1, x_2 \in \mathrm{Ob}(\mathcal{X})$, one has $\mathcal{X}(x_1, x_2) \leq \mathcal{Y}(F(x_1), F(x_2))$.

Example 2.70. For example, we have said several times—e.g. in Theorem 2.49—that preorders are **Bool**-categories, where $\mathcal{X}(x_1, x_2) = \texttt{true}$ is denoted $x_1 \leq x_2$. One would hope that monotone maps between preorders would correspond exactly to **Bool**-functors, and that's true. A monotone map $(X, \leq_X) \to (Y, \leq_Y)$ is a function $F \colon X \to Y$ such that for every $x_1, x_2 \in X$, if $x_1 \leq_X x_2$ then $F(x_1) \leq_Y F(x_2)$. In other words, we have

$$\mathcal{X}(x_1, x_2) \leq \mathcal{Y}(F(x_1), F(x_2)),$$

where the above $\leq$ takes place in the enriching category $\mathcal{V} = \mathbf{Bool}$; this is exactly the condition from Definition 2.69.

Remark 2.71. In fact, we have what is called an *equivalence* of categories between the category of preorders and the category of **Bool**-categories. In the next chapter we will develop the ideas necessary to state what this means precisely (Remark 3.59).

Example 2.72. Lawvere metric spaces are **Cost**-categories. The definition of **Cost**-functor should hopefully return a nice notion—a "friend"—from the theory of metric spaces, and it does: it recovers the notion of Lipschitz function. A Lipschitz (or more precisely, 1-Lipschitz) function is one under which the distance between any pair of points does not increase. That is, given Lawvere metric spaces (X, d_X) and (Y, d_Y), a **Cost**-functor between them is a function $F \colon X \to Y$ such that for every $x_1, x_2 \in X$ we have $d_X(x_1, x_2) \geq d_Y(F(x_1), F(x_2))$.

Exercise 2.73. The concepts of opposite, dagger, and skeleton (see Examples 1.58 and 1.72 and Remark 1.35) extend from preorders to $\mathcal{V}$-categories. The *opposite* of a $\mathcal{V}$-category $\mathcal{X}$ is denoted $\mathcal{X}^{\mathrm{op}}$ and is defined by

(i) $\mathrm{Ob}(\mathcal{X}^{\mathrm{op}}) := \mathrm{Ob}(\mathcal{X})$, and

(ii) for all $x, y \in \mathcal{X}$, we have $\mathcal{X}^{\mathrm{op}}(x, y) := \mathcal{X}(y, x)$.

A $\mathcal{V}$-category $\mathcal{X}$ is a *dagger* $\mathcal{V}$-category if the identity function is a $\mathcal{V}$-functor $\dagger \colon \mathcal{X} \to \mathcal{X}^{\mathrm{op}}$. And a *skeletal* $\mathcal{V}$-category is one in which if $I \leq \mathcal{X}(x, y)$ and $I \leq \mathcal{X}(y, x)$, then $x = y$.

Recall that an extended metric space (X, d) is a Lawvere metric space with two extra properties; see properties (b) and (c) in Definition 2.51.

1. Show that a skeletal dagger **Cost**-category is an extended metric space.
2. Use Exercise 1.73 to make sense of the following analogy: "preorders are to sets as Lawvere metric spaces are to extended metric spaces." ◊

2.4.3 Product $\mathcal{V}$-categories

If $\mathcal{V} = (V, \leq, I, \otimes)$ is a symmetric monoidal preorder and $\mathcal{X}$ and $\mathcal{Y}$ are $\mathcal{V}$-categories, then we can define their $\mathcal{V}$-product, which is a new $\mathcal{V}$-category.

Definition 2.74. Let $\mathcal{X}$ and $\mathcal{Y}$ be $\mathcal{V}$-categories. Define their *$\mathcal{V}$-product*, or simply *product*, to be the $\mathcal{V}$-category $\mathcal{X} \times \mathcal{Y}$ with

(i) $\mathrm{Ob}(\mathcal{X} \times \mathcal{Y}) := \mathrm{Ob}(\mathcal{X}) \times \mathrm{Ob}(\mathcal{Y})$,

(ii) $(\mathcal{X} \times \mathcal{Y})\big((x, y), (x', y')\big) := \mathcal{X}(x, x') \otimes \mathcal{Y}(y, y')$,

for two objects (x, y) and (x', y') in $\mathrm{Ob}(\mathcal{X} \times \mathcal{Y})$.

Product $\mathcal{V}$-categories are indeed $\mathcal{V}$-categories (Definition 2.46); see Exercise 2.75.

Exercise 2.75. Let $\mathcal{X} \times \mathcal{Y}$ be the $\mathcal{V}$-product of $\mathcal{V}$-categories as in Definition 2.74.

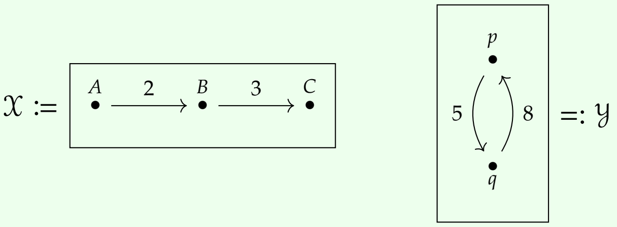
1. Check that for every object $(x, y) \in \mathrm{Ob}(\mathcal{X} \times \mathcal{Y})$ we have $I \leq (\mathcal{X} \times \mathcal{Y})\big((x, y), (x, y)\big)$.
2. Check that for every three objects (x_1, y_1), (x_2, y_2), and (x_3, y_3), we have

$$(\mathcal{X} \times \mathcal{Y})\big((x_1, y_1), (x_2, y_2)\big) \otimes (\mathcal{X} \times \mathcal{Y})\big((x_2, y_2), (x_3, y_3)\big) \leq (\mathcal{X} \times \mathcal{Y})\big((x_1, y_1), (x_3, y_3)\big).$$

3. We said at the start of Section 2.3.1 that the symmetry of $\mathcal{V}$ (condition (d) of Definition 2.2) would be required here. Point out exactly where that condition is used. ◊

When taking the product of two preorders $(P, \leq_P) \times (Q, \leq_Q)$, as first described in Example 1.56, we say that $(p_1, q_1) \leq (p_2, q_2)$ iff both $p_1 \leq p_2$ AND $q_1 \leq q_2$; the AND is the monoidal product $\otimes$ from of **Bool**. Thus the product of preorders is an example of a **Bool**-product.

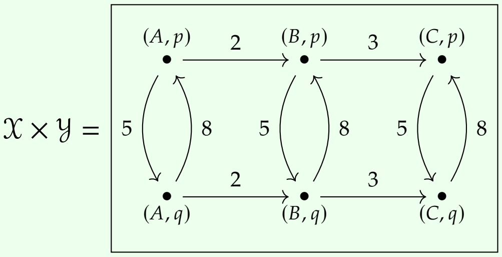
Example 2.76. Let $\mathcal{X}$ and $\mathcal{Y}$ be the Lawvere metric spaces (i.e. **Cost**-categories) defined by the following weighted graphs:



(2.77)

Their product is defined by taking the product of their sets of objects, so there are six objects in $\mathcal{X} \times \mathcal{Y}$. And the distance $d_{X \times Y}((x, y), (x', y'))$ between any two points is given by the sum $d_X(x, x') + d_Y(y, y')$.

Examine the following graph, and make sure you understand how easy it is to derive from the weighted graphs for $\mathcal{X}$ and $\mathcal{Y}$ in Eq. (2.77):



Exercise 2.78. Consider $\mathbb{R}$ as a Lawvere metric space, i.e. as a **Cost**-category (see Example 2.54). Form the **Cost**-product $\mathbb{R}\times\mathbb{R}$. What is the distance from $(5,6)$ to $(-1,4)$? Hint: apply Definition 2.74; the answer is not $\sqrt{40}$. ◊

In terms of matrices, $\mathcal{V}$-products are also quite straightforward. They generalize what is known as the Kronecker product of matrices. The matrices for $\mathcal{X}$ and $\mathcal{Y}$ in Eq. (2.77) are shown below

$\mathcal{X}$	A	B	C
A	0	2	5
B	∞	0	3
C	∞	∞	0

$\mathcal{Y}$	p	q
p	0	5
q	8	0

and their product is as follows:

$\mathcal{X}\times\mathcal{Y}$	(A,p)	(B,p)	(C,p)	(A,q)	(B,q)	(C,q)
(A,p)	0	2	5	5	7	10
(B,p)	∞	0	3	∞	5	8
(C,p)	∞	∞	0	∞	∞	5
(A,q)	8	10	13	0	2	5
(B,q)	∞	8	11	∞	0	3
(C,q)	∞	∞	8	∞	∞	0

We have drawn the product matrix as a block matrix, where there is one block—shaped like $\mathcal{X}$—for every entry of $\mathcal{Y}$. Make sure you can see each block as the $\mathcal{X}$-matrix shifted by an entry in $\mathcal{Y}$. This comes directly from the formula from Definition 2.74 and the fact that the monoidal product in **Cost** is $+$.

2.5 Computing presented $\mathcal{V}$-categories with matrix multiplication

In Section 2.3.3 we promised a straightforward way to construct the matrix representation of a **Cost**-category from a **Cost**-weighted graph. To do this, we use a generalized matrix multiplication. We shall show that this works, not just for **Cost**, but also for **Bool**, and many other monoidal preorders. The property required of the preorder is that of being a unital, commutative quantale. These are preorders with all joins, plus one additional ingredient, being *monoidal closed*, which we define next, in Section 2.5.1. The definition of a quantale will be given in Section 2.5.2.

2.5.1 Monoidal closed preorders

The definition of $\mathcal{V}$-category makes sense for any symmetric monoidal preorder $\mathcal{V}$. But that does not mean that any base of enrichment $\mathcal{V}$ is as useful as any other. In this section we define closed monoidal categories, which in particular enrich themselves! "Before you can really enrich others, you should really enrich yourself."

Definition 2.79. A symmetric monoidal preorder $\mathcal{V} = (V, \leq, I, \otimes)$ is called *symmetric monoidal closed* (or just *closed*) if, for every two elements $v, w \in V$, there is an element $v \multimap w$ in $\mathcal{V}$, called the *hom-element*, with the property

$$(a \otimes v) \leq w \quad \text{iff} \quad a \leq (v \multimap w). \tag{2.80}$$

for all $a, v, w \in V$.

Remark 2.81. The term 'closed' refers to the fact that a hom-element can be constructed for any two elements, so the preorder can be seen as closed under the operation of "taking homs." In later chapters we'll meet the closely-related concepts of compact closed categories (Definition 4.58) and cartesian closed categories (Section 7.2.1) that make this idea more precise. See especially Exercise 7.11.

One can consider the hom-element $v \multimap w$ as a kind of "single-use v-to-w converter." So Eq. (2.80) says that a and v are enough to get w if and only if a is enough to get a single-use v-to-w converter.

Exercise 2.82. Condition Eq. (2.80) says precisely that there is a Galois connection in the sense of Definition 1.95. Let's prove this fact. In particular, we'll prove that a monoidal preorder is monoidal closed iff, given any $v \in V$, the map $(- \otimes v) \colon V \to V$ given by multiplying with v has a right adjoint. We write this right adjoint $(v \multimap -) \colon V \to V$.

1. Using Definition 2.2, show that $(- \otimes v)$ is monotone.
2. Supposing that $\mathcal{V}$ is closed, show that for all $v, w \in V$ we have $\big((v \multimap w) \otimes v\big) \leq w$.
3. Using 2., show that $(v \multimap -)$ is monotone.
4. Conclude that a symmetric monoidal preorder is closed if and only if the monotone map $(- \otimes v)$ has a right adjoint. ◊

Example 2.83. The monoidal preorder **Cost** $= ([0, \infty], \geq, 0, +)$ is monoidal closed. Indeed, for any $x, y \in [0, \infty]$, define $x \multimap y := \max(0, y - x)$. Then, for any $a, x, y \in [0, \infty]$, we have

$$a + x \geq y \quad \text{iff} \quad a \geq y - x \quad \text{iff} \quad \max(0, a) \geq \max(0, y - x) \quad \text{iff} \quad a \geq (x \multimap y)$$

so $\multimap$ satisfies the condition of Eq. (2.80).

Note that we have not considered subtraction in **Cost** before; we can in fact use monoidal closure to *define* subtraction in terms of the order and monoidal structure!

Exercise 2.84. Show that **Bool** $= (\mathbb{B}, \leq, \texttt{true}, \wedge)$ is monoidal closed. ◊

Example 2.85. A non-example is $(\mathbb{B}, \leq, \texttt{false}, \vee)$. Indeed, suppose we had a $\multimap$ operator as in Definition 2.79. Note that $\texttt{false} \leq p \multimap q$, for any p, q no matter what $\multimap$ is, because `false` is less than everything. But using $a = \texttt{false}$, $p = \texttt{true}$, and $q = \texttt{false}$, we then get a contradiction: $(a \vee p) \not\leq q$ and yet $a \leq (p \multimap q)$.

Example 2.86. We started this chapter talking about resource theories. What does the closed structure look like from that perspective? For example, in chemistry it would say that for every two material collections c, d one can form a material collection $c \multimap d$ with the property that for any a, one has

$$a + c \to d \quad \text{if and only if} \quad a \to (c \multimap d).$$

Or more down to earth, since we have the reaction $2H_2O + 2Na \to 2NaOH + H_2$, we must also have

$$2H_2O \to (2Na \multimap (2NaOH + H_2))$$

So from just two molecules of water, you can form a certain substance, and not many substances fit the bill—our preorder Mat of chemical materials is not closed.

But it is not so far-fetched: this hypothetical new substance $(2Na \multimap (2NaOH + H_2))$ is not really a substance, but a potential reaction: namely that of converting a sodium to sodium-hydroxide-plus-hydrogen. Two molecules of water unlock that potential.

Proposition 2.87. Suppose $\mathcal{V} = (V, \leq, I, \otimes, \multimap)$ is a symmetric monoidal preorder that is closed. Then

(a) For every $v \in V$, the monotone map $- \otimes v \colon (V, \leq) \to (V, \leq)$ is left adjoint to $v \multimap - \colon (V, \leq) \to (V, \leq)$.

(b) For any element $v \in V$ and set of elements $A \subseteq V$, if the join $\bigvee_{a \in A} a$ exists then so does $\bigvee_{a \in A} v \otimes a$ and we have

$$\left(v \otimes \bigvee_{a \in A} a\right) \cong \bigvee_{a \in A} (v \otimes a). \tag{2.88}$$

(c) For any $v, w \in V$, we have $v \otimes (v \multimap w) \leq w$.

(d) For any $v \in V$, we have $v \cong (I \multimap v)$.

(e) For any $u, v, w \in V$, we have $(u \multimap v) \otimes (v \multimap w) \leq (u \multimap w)$.

Proof. We go through the claims in order.

(a) The definition of $(- \otimes v)$ being left adjoint to $(v \multimap -)$ is exactly the condition Eq. (2.80); see Definition 1.95 and Exercise 2.82.

(b) This follows from (a), using the fact that left adjoints preserve joins (Proposition 1.111).

(c) This follows from (a), using the equivalent characterisation of Galois connection in Proposition 1.107. More concretely, from reflexivity $(v \multimap w) \leq (v \multimap w)$, we obtain $(v \multimap w) \otimes v \leq w$ Eq. (2.80), and we are done by symmetry, which says $v \otimes (v \multimap w) = (v \multimap w) \otimes v$.

(d) Since $v \otimes I = v \leq v$, Eq. (2.80) says $v \leq (I \multimap v)$. For the other direction, we have $(I \multimap v) = I \otimes (I \multimap v) \leq v$ by (c).

(e) To obtain this inequality, we just need $u \otimes (u \multimap v) \otimes (v \multimap w) \leq w$. But this follows by two applications of (c). □

One might read (c) as saying "if I have a v and a single-use v-to-w converter, I can have a w." One might read (d) as saying "having a v is the same as having a single-use nothing-to-v converter." And one might read (e) as saying "if I have a single-use u-to-v converter and a single-use v-to-w converter, I can get a single-use u-to-w converter.

Remark 2.89. We can consider $\mathcal{V}$ to be enriched in itself. That is, for every $v, w \in \mathrm{Ob}(\mathcal{V})$, we can define $\mathcal{V}(v, w) := (v \multimap w) \in \mathcal{V}$. For this to really be an enrichment, we just need to check the two conditions of Definition 2.46. The first condition $I \leq \mathcal{X}(x, x) = (x \multimap x)$ is satisfied because $I \otimes x \leq x$. The second condition is satisfied by Proposition 2.87(e).

2.5.2 Quantales

To perform matrix multiplication over a monoidal preorder, we need one more thing: joins. These were first defined in Definition 1.81.

Definition 2.90. A *unital commutative quantale* is a symmetric monoidal closed preorder $\mathcal{V} = (V, \leq, I, \otimes, \multimap)$ that has all joins: $\bigvee A$ exists for every $A \subseteq V$. In particular, we often denote the empty join by $0 := \bigvee \varnothing$.

Whenever we speak of quantales in this book, we mean unital commutative quantales. We will try to remind the reader of that. There are also very interesting applications of noncommutative quantales; see Section 2.6.

Example 2.91. In Example 2.83, we saw that **Cost** is monoidal closed. To check whether **Cost** is a quantale, we take an arbitrary set of elements $A \subseteq [0, \infty]$ and ask if it has a join $\bigvee A$. To be a join, it needs to satisfy two properties:

a. $a \geq \bigvee A$ for all $a \in A$, and
b. if $b \in [0, \infty]$ is any element such that $a \geq b$ for all $a \in A$, then $\bigvee A \geq b$.

In fact we can define such a join: it is typically called the *infimum*, or greatest lower bound, of A.[5] For example, if $A = \{2, 3\}$ then $\bigvee A = 2$. We have joins for infinite sets too: if $B = \{2.5, 2.05, 2.005, \ldots\}$, its infimum is 2. Finally, in order to say that $([0, \infty], \geq)$ has all joins, we need a join to exist for the empty set $A = \varnothing$ too. The first condition becomes vacuous—there are no a's in A—but the second condition says that for any $b \in [0, \infty]$ we have $\bigvee \varnothing \geq b$; this means $\bigvee \varnothing = \infty$.

Thus indeed $([0, \infty], \geq)$ has all joins, so **Cost** is a quantale.

Exercise 2.92.

1. What is $\bigvee \varnothing$, which we generally denote 0, in the case
   a. $\mathcal{V} = \textbf{Bool} = (\mathbb{B}, \leq, \texttt{true}, \wedge)$?
   b. $\mathcal{V} = \textbf{Cost} = ([0, \infty], \geq, 0, +)$?
2. What is the join $x \vee y$ in the case
   a. $\mathcal{V} = \textbf{Bool}$, and $x, y \in \mathbb{B}$ are booleans?
   b. $\mathcal{V} = \textbf{Cost}$, and $x, y \in [0, \infty]$ are distances? ◊

Exercise 2.93. Show that $\textbf{Bool} = (\mathbb{B}, \leq, \texttt{true}, \wedge)$ is a quantale. ◊

Exercise 2.94. Let S be a set and recall the power set monoidal preorder $(\mathsf{P}(S), \subseteq, S, \cap)$ from Exercise 2.35. Is it a quantale? ◊

Remark 2.95. One can personify the notion of unital, commutative quantale as a kind of navigator. A navigator is someone who understands "getting from one place to another." Different navigators may care about or understand different aspects—whether one can get from A to B, how much time it will take, what modes of travel will work, etc.—but they certainly have some commonalities. Most importantly, a navigator needs to be able to read a map: given routes A to B and B to C, they understand how to get a route A to C. And they know how to search over the space of way-points to get from A to C. These will correspond to the monoidal product and the join operations, respectively.

Proposition 2.96. Let $\mathcal{P} = (P, \leq)$ be a preorder. It has all joins iff it has all meets.

Proof. The joins (resp. meets) in $\mathcal{P}$ are the meets (resp. joins) in $\mathcal{P}^{\mathrm{op}}$, so the two claims are dual: it suffices to show that if $\mathcal{P}$ has all joins then it has all meets.

Suppose $\mathcal{P}$ has all joins and suppose that $A \subseteq \mathcal{P}$ is a subset for which we want the meet. Consider the set $M_A \coloneqq \{p \in P \mid p \leq a \text{ for all } a \in A\}$ of elements below everything in A. Let $m_A \coloneqq \bigvee_{p \in M_A} p$ be their join. We claim that m_A is a meet for A.

We first need to know that for any $a \in A$ we have $m_A \leq a$, but this is by definition of join: since all $p \in M_A$ satisfy $p \leq a$, so does their join $m_A \leq a$. We second need to know that for any $m' \in P$ with $m' \leq a$ for all $a \in A$, we have $m' \leq m$. But every such m' is actually an element of M_A and m is their join, so $m' \leq m$. This completes the proof. □

[5] Here, by the infimum of a subset $A \subseteq [0, \infty]$, we mean infimum in the usual order on $[0, \infty]$: the largest number that is $\leq$ everything in A. For example, the infimum of $\{3.1, 3.01, 3.001, \ldots\}$ is 3. But note that this is the *supremum* in the reversed, $\geq$, order of **Cost**.

In particular, a quantale has all meets and all joins, even though we only define it to have all joins.

Remark 2.97. The notion of Hausdorff distance can be generalized, allowing the role of **Cost** to be taken by any quantale $\mathcal{V}$. If $\mathcal{X}$ is a $\mathcal{V}$-category with objects X, and $U \subseteq X$ and $V \subseteq X$, we can generalize the usual Hausdorff distance, on the left below, to the formula on the right:

$$d(U, V) \coloneqq \sup_{u \in U} \inf_{v \in V} d(u, v) \qquad\qquad \mathcal{X}(U, V) \coloneqq \bigwedge_{u \in U} \bigvee_{v \in V} \mathcal{X}(u, v).$$

For example, if $\mathcal{V} = \textbf{Bool}$, the Hausdorff distance between sub-preorders U and V answers the question "can I get into V from every $u \in U$," i.e. $\forall_{u \in U} . \exists_{v \in V} . u \leq v$. Or for another example, use $\mathcal{V} = \mathsf{P}(M)$ with its interpretation as modes of transportation, as in Exercise 2.62. Then the Hausdorff distance $d(U, V) \in \mathsf{P}(M)$ tells us those modes of transportation that will get us into V from every point in U.

Proposition 2.98. Suppose $\mathcal{V} = (V, \leq, I, \otimes)$ is any symmetric monoidal preorder that has all joins. Then $\mathcal{V}$ is closed—i.e. it has a $\multimap$ operation and hence is a quantale—if and only if $\otimes$ distributes over joins; i.e. if Eq. (2.88) holds for all $v \in V$ and $A \subseteq V$.

Proof. We showed one direction in Proposition 2.87(b): if $\mathcal{V}$ is monoidal closed then Eq. (2.88) holds. We need to show that Eq. (2.88) holds then $- \otimes v \colon V \to V$ has a right adjoint $v \multimap -$. This is just the adjoint functor theorem, Theorem 1.115. It says we can define $v \multimap w$ to be

$$v \multimap w \coloneqq \bigvee_{\{a \in V \mid a \otimes v \leq w\}} a. \qquad \square$$

2.5.3 Matrix multiplication in a quantale

A quantale $\mathcal{V} = (V, \leq, I, \otimes, \multimap)$, as defined in Definition 2.79, provides what is necessary to perform matrix multiplication.[6] The usual formula for matrix multiplication is:

$$(M * N)(i, k) = \sum_j M(i, j) * N(j, k). \tag{2.99}$$

We will get a formula where joins stand in for the sum operation $\sum$, and $\otimes$ stands in for the product operation $*$. Recall our convention of writing $0 \coloneqq \bigvee \varnothing$.

Definition 2.100. Let $\mathcal{V} = (V, \leq, \otimes, I)$ be a quantale. Given sets X and Y, a *matrix with entries in* $\mathcal{V}$, or simply a $\mathcal{V}$-*matrix*, is a function $M \colon X \times Y \to V$. For any $x \in X$ and $y \in Y$, we call $M(x, y)$ the (x, y)-*entry*.

[6] This works for noncommutative quantales as well.

Here is how you multiply $\mathcal{V}$-matrices $M\colon X \times Y \to V$ and $N\colon Y \times Z \to V$. Their product is defined to be the matrix $(M * N)\colon X \times Z \to V$, whose entries are given by the formula

$$(M * N)(x, z) \coloneqq \bigvee_{y \in Y} M(x, y) \otimes N(y, z). \tag{2.101}$$

Note how similar this is to Eq. (2.99).

Example 2.102. Let $\mathcal{V} = \mathbf{Bool}$. Here is an example of matrix multiplication $M * N$. Here $X = \{1, 2, 3\}$, $Y = \{1, 2\}$, and $Z = \{1, 2, 3\}$, matrices $M\colon X \times Y \to \mathbb{B}$ and $N\colon Y \times Z \to \mathbb{B}$ are shown to the left below, and their product is shown to the right:

$$\begin{pmatrix} \texttt{false} & \texttt{false} \\ \texttt{false} & \texttt{true} \\ \texttt{true} & \texttt{true} \end{pmatrix} * \begin{pmatrix} \texttt{true} & \texttt{true} & \texttt{false} \\ \texttt{true} & \texttt{false} & \texttt{true} \end{pmatrix} = \begin{pmatrix} \texttt{false} & \texttt{false} & \texttt{false} \\ \texttt{true} & \texttt{false} & \texttt{true} \\ \texttt{true} & \texttt{true} & \texttt{true} \end{pmatrix}$$

The identity V-matrix on a set X is $I_X\colon X \times X \to V$ given by

$$I_X(x, y) \coloneqq \begin{cases} I & \text{if } x = y \\ 0 & \text{if } x \neq y. \end{cases}$$

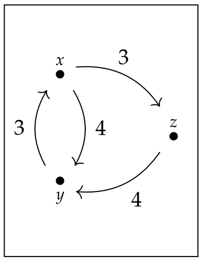
Exercise 2.103. Write down the 2×2-identity matrix for each of the quantales $(\mathbb{N}, \leq, 1, *)$, $\mathbf{Bool} = (\mathbb{B}, \leq, \texttt{true}, \wedge)$, and $\mathbf{Cost} = ([0, \infty], \geq, 0, +)$. ◊

Exercise 2.104. Let $\mathcal{V} = (V, \leq, I, \otimes, \multimap)$ be a quantale. Use Eq. (2.101) and Proposition 2.87 to prove the following.

1. Prove the *identity law*: for any sets X and Y and V-matrix $M\colon X \times Y \to V$, one has $I_X * M = M$.
2. Prove the *associative law*: for any matrices $M\colon W \times X \to V$, $N\colon X \times Y \to V$, and $P\colon Y \times Z \to V$, one has $(M * N) * P = M * (N * P)$. ◊

Recall the weighted graph Y from Eq. (2.56). One can read off the associated matrix M_Y, and one can calculate the associated metric d_Y:

$Y \coloneqq$



M_Y	x	y	z
x	0	4	3
y	3	0	∞
z	∞	4	0

d_Y	x	y	z
x	0	4	3
y	3	0	6
z	7	4	0

Here we fully explain how to compute d_Y using only M_Y.

The matrix M_Y can be thought of as recording the length of paths that traverse either 0 or 1 edges: the diagonals being 0 mean we can get from x to x without traversing any edges. When we can get from x to y in one edge we record its length in M_Y, otherwise we use ∞.

When we multiply M_Y by itself using the formula Eq. (2.101), the result M_Y^2 tells us the length of the shortest path traversing 2 edges or fewer. Similarly M_Y^3 tells us about the shortest path traversing 3 edges or fewer:

$$M_Y^2 = \begin{array}{c|ccc} \nearrow & x & y & z \\ \hline x & 0 & 4 & 3 \\ y & 3 & 0 & 6 \\ z & 7 & 4 & 0 \end{array} \qquad\qquad M_Y^3 = \begin{array}{c|ccc} \nearrow & x & y & z \\ \hline x & 0 & 4 & 3 \\ y & 3 & 0 & 6 \\ z & 7 & 4 & 0 \end{array}$$

One sees that the powers stabilize: $M_Y^2 = M_Y^3$; as soon as that happens one has the matrix of distances, d_Y. Indeed M_Y^n records the lengths of the shortest path traverse n edges or fewer, and the powers will always stabilize if the set of vertices is finite, since the shortest path from one vertex to another will never visit a given vertex more than once.[7]

Exercise 2.105. Recall from Exercise 2.60 the matrix M_X, for X as in Eq. (2.56). Calculate M_X^2, M_X^3, and M_X^4. Check that M_X^4 is what you got for the distance matrix in Exercise 2.58. ◊

This procedure gives an algorithm for computing the $\mathcal{V}$-category presented by any $\mathcal{V}$-weighted graph using matrix multiplication.

2.6 Summary and further reading

In this chapter we thought of elements of preorders as describing resources, with the order detailing whether one resource could be obtained from another. This naturally led to the question of how to describe what could be built from a pair of resources, which led us to consider monoid structures on preorders. More abstractly, these monoidal preorders were seen to be examples of enriched categories, or $\mathcal{V}$-categories, over the symmetric monoidal preorder **Bool**. Changing **Bool** to the symmetric monoidal preorder **Cost**, we arrived upon Lawvere metric spaces, a slight generalization of the usual notion of metric space. In terms of resources, **Cost**-categories tell us the cost of obtaining one resource from another.

At this point, we sought to get a better feel for $\mathcal{V}$-categories in two ways. First, we introduced various important constructions: base change, functors, products. Second, we looked at how to present $\mathcal{V}$-categories using labelled graphs; here, perhaps surprisingly, we saw that matrix multiplication gives an algorithm to compute the hom-objects from a labelled graph.

Resource theories are discussed in much more detail in [CFS16; Fri17]. The authors provide many more examples of resource theories in science, including in thermody-

[7] The method works even in the infinite case: one takes the infimum of all powers M_Y^n. The result always defines a Lawvere metric space.

namics, Shannon's theory of communication channels, and quantum entanglement. They also discuss more of the numerical theory than we did, including calculating the asymptotic rate of conversion from one resource into another.

Enrichment is a fundamental notion in category theory, and we will we return to it in Chapter 4, generalizing the definition so that categories, rather than mere preorders, can serve as bases of enrichment. In this more general setting we can still perform the constructions we introduced in Section 2.4—base change, functors, products—and many others; the authoratitive, but by no means easy, reference on this is the book by Kelly [Kel05].

While preorders were familiar before category theory came along, Lawvere metric spaces are a beautiful generalization of the previous notion of (symmetric) metric space, that is due to, well, Lawvere. A deeper exploration than the taste we gave here can be found in his classic paper [Law73], where he also discusses ideas like Cauchy completeness in category-theoretic terms, and which hence generalize to other categorical settings.

We observed that while any symmetric monoidal preorder can serve as a base for enrichment, certain preorders—quantales—are better than others. Quantales are well known for links to other parts of mathematics too. The word quantale is in fact a portmanteau of 'quantum locale', where quantum refers to quantum physics, and locale is a fundamental structure in topology. For a book-length introduction of quantales and their applications, one might check [Ros90]. The notion of cartesian closed categories, later generalized to monoidal closed categories, is due to Ronnie Brown [Bro61].

Note that while we have only considered commutative quantales, the noncommutative variety also arise naturally. For example, the power set of any monoid forms a quantale that is commutative iff the monoid is. Another example is the set of all binary relations on a set X, where multiplication is relational composition; this is non-commutative. Such noncommutative quantales have application to concurrency theory, and in particular process semantics and automata; see [AV93] for details.

18.S097 Applied Category Theory
January IAP 2019