Chapter 7

Logic of behavior: Sheaves, toposes, and internal languages

7.1 How can we prove our machine is safe?

Imagine you are trying to design a system of interacting components. You wouldn't be doing this if you didn't have a goal in mind: you want the system to do something, to behave in a certain way. In other words, you want to restrict its possibilities to a smaller set: you want the car to remain on the road, you want the temperature to remain in a particular range, you want the bridge to be safe for trucks to pass. Out of all the possibilities, your system should only permit some.

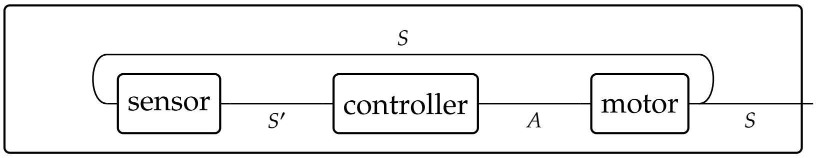
Since your system is made of components that interact in specified ways, the possible behavior of the whole—in any environment—is determined by the possible behaviors of each of its components in their local environments, together with the precise way in which they interact.[1] In this chapter, we will discuss a logic wherein one can describe general types of behavior that occur over time, and prove properties of a larger-scale system from the properties and interaction patterns of its components.

For example, suppose we want an autonomous vehicle to maintain a distance of some $\mathtt{safe} \in \mathbb{R}$ from other objects. To do so, several components must interact: a sensor that approximates the real distance by an internal variable S', a controller that uses S' to decide what action A to take, and a motor that moves the vehicle with an

[1] The well-known concept of emergence is not about possibilities, it is about prediction. Predicting the behavior of a system given predictions of its components is notoriously hard. The behavior of a double pendulum is chaotic—meaning extremely sensitive to initial conditions—whereas those of the two component pendulums are not. However, the set of possibilities for the double pendulum is completely understood: it is the set of possible angular positions and velocities of both arms. When we speak of a machine's properties in this chapter, we always mean the guarantees on its behaviors, not the probabilities involved, though the latter would certainly be an interesting thing to contemplate.

acceleration based on A. This in turn affects the real distance S, so there is a feedback loop.

Consider the following model diagram:

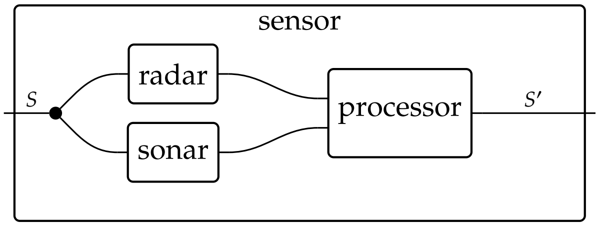


(7.1)

In the diagram shown, the distance S is exposed by the exterior interface. This just means we imagine S as being a variable that other components of a larger system may want to interact with. We could have exposed no variables (making it a closed system) or we could have exposed A and/or S' as well.

In order for the system to ensure $S \geq \texttt{safe}$, we need each of the components to ensure a property of its own. But what are these components, 'sensor, controller, motor', and what do they do?

One way to think about any of the components is to open it up and see how it is put together; with a detailed study we may be able to say what it will do. For example, just as S was exposed in the diagram above, one could imagine opening up the 'sensor' component box in Eq. (7.1) and seeing an interaction between subcomponents



This ability to zoom in and see a single unit as being composed of others is important for design. But at the end of the day, you eventually need to stop diving down and simply use the properties of the components in front of you to prove properties of the composed system. Have no fear: everything we do in this chapter will be fully compositional, i.e. compatible with opening up lower-level subsystems and using the fractal-like nature of composition. However at a given time, your job is to design the system at a given level, taking the component properties of lower-level systems as given.

We will think of each component in terms of the relationship it maintains (through time) between the changing values on its ports. "Whenever I see a flash, I will increase pressure on the button": this is a relationship I maintain through time between the changing values on my eye port and my finger port. We will make this more precise soon, but fleshing out the situation in Eq. (7.1) should help. The sensor maintains a relationship between S and S', e.g. that the real distance S and its internal representation S' differ by no more than 5cm. The controller maintains a relationship between S' and the action signal A, e.g. that if at any time $S < \texttt{safe}$, then within one second it will emit the signal $A = \mathtt{go}$. The motor maintains a relationship between A and S, e.g. that A dictates the second derivative of S by the formula

$$\big((A = \mathtt{go}) \Rightarrow \ddot{S} > 1\big) \wedge \big((A = \mathtt{stop}) \Rightarrow \ddot{S} = 0\big)\,. \tag{7.2}$$

If we want to prove properties of the whole interacting system, then the relationships maintained by each component need to be written in a formal logical language, something like what we saw in Eq. (7.2). From that basis, we can use standard proof techniques to combine properties of subsystems into properties of the whole. This is our objective in the present chapter.

We have said how component systems, wired together in some arrangement, create larger-scale systems. We have also said that, given the wiring arrangement, the behavioral properties of the component systems dictate the behavioral properties of the whole. But what exactly are behavioral properties?

In this chapter, we want to give a formal language and semantics for a very general notion of behavior. Mathematics is itself a formal language; the usual style of mathematical modeling is to use any piece of this vast language at any time and for any reason. One uses "human understanding" to ensure that the different models are fitting together in an appropriate way when different systems are combined. The present work differs in that we want to find a domain-specific language for modeling behavior, any sort of behavior, and nothing but behavior. Unlike in the wide world of math, we want a setting where the only things that can be discussed are behaviors.

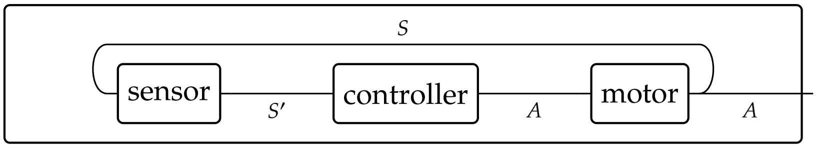
For this, we will construct what is called a *topos*, which is a special kind of category. Our topos, let's call it **BT**, will have behavior types—roughly speaking, sets whose elements can change through time—as its objects. An amazing fact about toposes[2] is that they come with an *internal language* that looks very much like the usual formal language of mathematics itself. Thus one can define graphs, groups, topological spaces, etc. in any topos. But in **BT**, what we call graphs will actually be graphs that change through time, and similarly what we call groups and spaces will actually be groups and spaces that change through time.

The topos **BT** not only has an internal language, but also a mathematical semantics using the notion of sheaves. Technically, a sheaf is a certain sort of functor, but one can imagine it as a space of possibilities, varying in a controlled way; in our case it will be a space of possible behaviors varying in a certain notion of time. Every property we prove in our logic of behavior types will have meaning in this category of sheaves.

When discussing systems and components—such as sensors, controllers, motors, etc.—we mentioned behavior types; these will be the objects in the topos **BT**. Every wire in the picture below will stand for a behavior type, and every box X will stand for a behavioral property, a relation that X maintains between the changing values on its

[2] The plural of topos is often written *topoi*, rather than toposes. This seems a bit fancy for our taste. As Johnstone suggests in [Joh77], we might ask those who "persist in talking about topoi whether, when they go out for a ramble on a cold day, they carry supplies of hot tea with them in thermoi." It's all in good fun; either term is perfectly reasonable and well-accepted.

ports.



For example we could imagine that

- S (wire): The behavior of S over a time-interval $[a, b]$ is that of all continuous real-valued functions $[a, b] \to \mathbb{R}$.
- A (wire): The behavior of A over a time-interval $[a, b]$ is all piecewise constant functions, taking values in the finite set such as $\{\texttt{go}, \texttt{stop}\}$.
- controller (box): the relation $\{(S', A) \mid Eq.\ (7.2)\}$, i.e. all behavioral pairs (S', A) that conform to what we said our controller is supposed to do in Eq. (7.2).

7.2 The category Set as an exemplar topos

We want to think about a very abstract sort of thing, called a topos, because we will see that behavior types form a topos. To get started, we begin with one of the easiest toposes to think about, namely the topos **Set** of sets. In this section we will discuss commonalities between sets and every other topos. We will go into some details about the category of sets, so as to give intuition for other toposes. In particular, we'll pay careful attention to the logic of sets, because we eventually want to understand the logic of behaviors.

Indeed, logic and sets are closely related. For example, the logical statement—more formally known as a predicate—`likes_cats` defines a function from the set P of people to the set $\mathbb{B} = \{\texttt{false}, \texttt{true}\}$ of truth values, where `Brendan` $\in P$ maps to `true` because he likes cats whereas `Ursula` $\in P$ maps to `false` because she does not. Alternatively, `likes_cats` also defines a subset of P, consisting of exactly the people that do like cats

$$\{p \in P \mid \texttt{likes\_cats}(p)\}.$$

In terms of these subsets, logical operations correspond to set operations, e.g. AND corresponds to intersection: indeed, the set of people for (mapped to `true` by) the predicate `likes_cats_AND_likes_dogs` is equal to the intersection of the set for `likes_cats` and the set for `likes_dogs`.

We saw in Chapter 3 that such operations, which are examples of database queries, can be described in terms of limits and colimits in **Set**. Indeed, the category **Set** has many such structures and properties, which together make logic possible in that setting. In this section we want to identify these properties, and show how logical operations can be defined using them.

Why would we want to abstractly find such structures and properties? In the next section, we'll start our search for other categories that also have them. Such categories, called toposes, will be **Set**-like enough to do logic, but have much more complex and interesting semantics. Indeed, we will discuss one whose logic allows us to reason not about properties of sets, but about behavioral properties of very general machines.

7.2.1 Set-like properties enjoyed by any topos

Although we will not prove it in this book, toposes are categories that are similar to **Set** in many ways. Here are some facts that are true of any topos $\mathcal{E}$:

1. $\mathcal{E}$ has all limits,
2. $\mathcal{E}$ has all colimits,
3. $\mathcal{E}$ is cartesian closed,
4. $\mathcal{E}$ has epi-mono factorizations,
5. $\mathcal{E}$ has a subobject classifier $1 \xrightarrow{\texttt{true}} \Omega$.

In particular, since **Set** is a topos, all of the above facts are true for $\mathcal{E} = \textbf{Set}$. Our first goal is to briefly review these concepts, focusing most on the subobject classifier.

Limits and colimits. We discussed limits and colimits briefly in Section 3.4.2, but the basic idea is that one can make new objects from old by taking products, using equations to define subobjects, forming disjoint unions, and taking quotients.object 0. One of the most important types of limit (resp. colimit) is that of pullbacks (resp. pushouts); see Example 3.99 and Definition 6.19. For our work below, we'll need to know a touch more about pullbacks than we have discussed so far, so let's begin there.

Suppose that $\mathcal{C}$ is a category and consider the diagrams below:

$$\begin{tikzcd}
A \arrow[r] \arrow[d] & B \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & C \arrow[d] \\
D \arrow[r] & E \arrow[r] & F
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
A \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & B \arrow[r] \arrow[d] & C \arrow[d] \\
D \arrow[r] & E \arrow[r] & F
\end{tikzcd}$$

In the left-hand square, the corner symbol $\lrcorner$ unambiguously means that the square (B, C, E, F) is a pullback. But in the right-hand square, does the corner symbol mean that (A, B, D, E) is a pullback or that (A, C, D, F) is a pullback? It's ambiguous, but as we next show, it becomes unambiguous if the right-hand square is a pullback.

Proposition 7.3. In the commutative diagram below, suppose that the (B, C, B', C') square is a pullback:

$$\begin{tikzcd}
A \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & B \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & C \arrow[d] \\
A' \arrow[r] & B' \arrow[r] & C'
\end{tikzcd}$$

Then the (A, B, A', B') square is a pullback iff the (A, C, A', C') rectangle is a pullback.

Exercise 7.4. Prove Proposition 7.3 using the definition of limit from Section 3.4.2. ◊

Epi-mono factorizations. The abbreviation ‘epi’ stands for *epimorphism*, and the abbreviation ‘mono’ stands for monomorphism. Epimorphisms are maps that act like surjections, and monomorphisms are maps that act like injections.[3] We can define them formally in terms of pushouts and pullbacks.

Definition 7.5. Let $\mathcal{C}$ be a category, and let $f\colon A \to B$ be a morphism. It is called a *monomorphism* (resp. *epimorphism*) if the square to the left is a pullback (resp. the square to the right is a pushout):

$$\begin{tikzcd}
A \arrow[r, "\mathrm{id}_A"] \arrow[d, "\mathrm{id}_A"'] \arrow[dr, phantom, "\lrcorner", very near start] & A \arrow[d, "f"] \\
A \arrow[r, "f"'] & B
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
A \arrow[r, "f"] \arrow[d, "f"'] & B \arrow[d, "\mathrm{id}_B"] \\
B \arrow[r, "\mathrm{id}_B"'] & B \arrow[ul, phantom, "\ulcorner", very near start]
\end{tikzcd}$$

Exercise 7.6. Show that in **Set**, monomorphisms are just injections:

1. Show that if f is a monomorphism then it is injective.
2. Show that if $f\colon A \to B$ is injective then it is a monomorphism. ◊

Exercise 7.7.

1. Show that the pullback of an isomorphism along any morphism is an isomorphism. That is, suppose that $i\colon B' \to B$ is an isomorphism and $f\colon A \to B$ is any morphism. Show that i' is an isomorphism, in the following diagram:

$$\begin{tikzcd}
A' \arrow[r, "f'"] \arrow[d, "i'"', "\cong"] \arrow[dr, phantom, "\lrcorner", very near start] & B' \arrow[d, "i", "\cong"'] \\
A \arrow[r, "f"'] & B
\end{tikzcd}$$

2. Show that for any map $f\colon A \to B$, the square shown is a pullback:

$$\begin{tikzcd}
A \arrow[r, "f"] \arrow[d, equal] \arrow[dr, phantom, "\lrcorner", very near start] & B \arrow[d, equal] \\
A \arrow[r, "f"'] & B
\end{tikzcd}$$

◊

Exercise 7.8. Suppose the following diagram is a pullback in a category $\mathcal{C}$:

$$\begin{tikzcd}
A' \arrow[r, "g"] \arrow[d, "f'"'] \arrow[dr, phantom, "\lrcorner", very near start] & A \arrow[d, tail, "f"] \\
B' \arrow[r, "h"'] & B
\end{tikzcd}$$

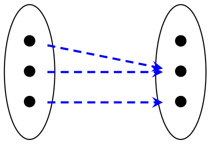
Use Proposition 7.3 and Exercise 7.7 to show that if f is a monomorphism, then so is f'. ◊

[3] Surjections are sometimes called ‘onto’ and injections are sometimes called ‘one-to-one’, hence the Greek prefixes *epi* and *mono*.

Now that we have defined epimorphisms and monomorphisms, we can say what epi-mono factorizations are. We say that a morphism $f\colon C \to D$ in $\mathcal{E}$ has an epi-mono factorization if it has an 'image'; that is, there is an object $\mathrm{im}(f)$, an epimorphism $C \twoheadrightarrow \mathrm{im}(f)$, and a monomorphism $\mathrm{im}(f) \rightarrowtail D$, whose composite is f.

In **Set**, epimorphisms are surjections and monomorphisms are injections. Every function $f\colon C \to D$ may be factored as a surjective function onto its image $\mathrm{im}(f) = \{f(c) \mid c \in C\}$, followed by the inclusion of this image into the codomain D. Moreover, this factorization is unique up to isomorphism.

Exercise 7.9. Factor the following function $f\colon \underline{3} \to \underline{3}$ as an epimorphism followed by a monomorphism.



◊

This is the case in any topos $\mathcal{E}$: for any morphism $f\colon c \to d$, there exists an epimorphism e and a monomorphism m such that $f = (e \mathbin{⨟} m)$ is their composite.

Cartesian closed. A category $\mathcal{C}$ being cartesian closed means that $\mathcal{C}$ has a symmetric monoidal structure given by products, and it is monoidal closed with respect to this. (We previously saw monoidal closure in Definition 2.79 (for preorders) and Proposition 4.60, as a corollary of compact closure.) Slightly more down-to-earth, cartesian closure means that for any two objects $C, D \in \mathcal{C}$, there is a 'hom-object' $D^C \in \mathcal{C}$ and a natural isomorphism for any $A \in \mathcal{C}$:

$$\mathcal{C}(A \times C, D) \cong \mathcal{C}(A, D^C) \tag{7.10}$$

Think of it this way. Suppose you're A and I'm C, and we're interacting through some game $f(-,-)\colon A \times C \to D$: for whatever action $a \in A$ that you take and action $c \in C$ that I take, $f(a,c)$ is some value in D. Since you're self-centered but loving, you think of this situation as though you're creating a game experience for me. When you do a, you make a game $f(a,-)\colon C \to D$ for me alone. In the formalism, D^C represents the set of games for me. So now you've transformed a two-player game, valued in D, into a one-player game, you're the player, valued in... one player games valued in D. This transformation is invertible—you can switch your point of view at will—and it's called *currying*. This is the content of Example 3.72.

Exercise 7.11. Let $\mathcal{V} = (V, \leq, I, \otimes)$ be a (unital, commutative) quantale—see Definition 2.90—and suppose it satisfies the following for all $v, w, x \in V$:

- $v \leq I$,
- $v \otimes w \leq v$ and $v \otimes w \leq w$, and
- if $x \leq v$ and $x \leq w$ then $x \leq v \otimes w$.

1. Show that $\mathcal{V}$ is a cartesian closed category, in fact a cartesian closed preorder.
2. Can every cartesian closed preorder be obtained in this way? ◊

Subobject classifier. The concept of a subobject classifier requires more attention, because its existence has huge consequences for a category $\mathcal{C}$. In particular, it creates the setting for a rich system of *higher order logic* to exist inside $\mathcal{C}$; it does so by providing some things called 'truth values'. The higher order logic manifests in its fully glory when $\mathcal{C}$ has finite limits and is cartesian closed, because these facts give rise to the logical operations on truth values.[4] In particular, the higher order logic exists in any topos.

We will explain subobject classifiers in as much detail as we can; in fact, it will be our subject for the rest of Section 7.2.

7.2.2 The subobject classifier

Before giving the definition of subobject classifiers, recall that monomorphisms in **Set** are injections, and any injection $X \rightarrowtail Y$ is isomorphic to a subset of Y. This gives a simple and useful way to conceptualize monomorphisms into Y when reading the following definition: it will do no harm to think of them as subobjects of Y.

Definition 7.12. Let $\mathcal{E}$ be a category with finite limits, i.e. with pullbacks and a terminal object 1. A *subobject classifier* in $\mathcal{E}$ consists of an object $\Omega \in \mathcal{E}$, together with a monomorphism $\mathtt{true}\colon 1 \to \Omega$, satisfying the following property: for any objects X and Y and monomorphism $m\colon X \rightarrowtail Y$ in $\mathcal{E}$, there is a unique morphism $\ulcorner m \urcorner\colon Y \to \Omega$ such that the diagram on the left of Eq. (7.13) is a pullback in $\mathcal{E}$:

$$\begin{tikzcd} X \arrow[r, "!"] \arrow[d, tail, "m"'] \arrow[dr, phantom, "\lrcorner", very near start] & 1 \arrow[d, "\mathtt{true}"] \\ Y \arrow[r, "\ulcorner m \urcorner"'] & \Omega \end{tikzcd} \qquad\qquad \begin{tikzcd} \{Y \mid p\} \arrow[r, "!"] \arrow[d, tail] \arrow[dr, phantom, "\lrcorner", very near start] & 1 \arrow[d, "\mathtt{true}"] \\ Y \arrow[r, "p"'] & \Omega \end{tikzcd} \tag{7.13}$$

We refer to $\ulcorner m \urcorner$ as the *characteristic map* of m, or we say that $\ulcorner m \urcorner$ *classifies* m. Conversely, given any map $p\colon Y \to \Omega$, we denote the pullback of $\mathtt{true}$ as on the right of Eq. (7.13).

A *predicate* on Y is a morphism $Y \to \Omega$.

Definition 7.12 is a bit difficult to get one's mind around, partly because it is hard to imagine its consequences. It is like a superdense nugget from outer space, and through scientific explorations in the latter half of the 20th century, we have found that it brings super powers to whichever categories possess it. We will explain some of the consequences below, but very quickly, the idea is the following.

When a category has a subobject classifier, it provides a translator, turning subobjects of any object Y into maps from that Y to the particular object Ω. Pullback of the

[4] A category that has finite limits, is cartesian closed, and has a subobject classifier is called an *elementary topos*. We will not discuss these further, but they are the most general notion of topos in ordinary category theory. When someone says topos, you might ask "Grothendieck topos or elementary topos?," because there does not seem to be widespread agreement on which is the default.

monomorphism $\texttt{true}\colon 1 \to \Omega$ provides a translator going back, turning maps $Y \to \Omega$ into subobjects of Y. We can replace our fantasy of the superdense nugget with a slightly more refined story: "any object Y understands itself—its parts and the logic of how they fit together—by asking questions of the oracle Ω, looking for what's true." Or to fully be precise but dry, "subobjects of Y are classified by predicates on Y."

Let's move from stories and slogans to concrete facts.

The subobject classifier in Set. Since **Set** is a topos, it has a subobject classifier. It will be a set with supposedly wonderful properties; what set is it?

The subobject classifier in **Set** is the set of booleans,

$$\Omega_{\mathbf{Set}} \coloneqq \mathbb{B} = \{\texttt{true}, \texttt{false}\}. \tag{7.14}$$

So in **Set**, the truth values are true and false.

By definition (Def. 7.12), the subobject classifier comes equipped with a morphism, generically called $\texttt{true}\colon 1 \to \Omega$; in the case of **Set** it is played by the function $1 \to \{\texttt{true}, \texttt{false}\}$ that sends 1 to $\texttt{true}$. In other words, the morphism $\texttt{true}$ is aptly named in this case.

For sets, monomorphism just means injection, as we mentioned above. So Definition 7.12 says that for any injective function $m\colon X \rightarrowtail Y$ between sets, we are supposed to be able to find a characteristic function $\ulcorner m \urcorner\colon Y \to \{\texttt{true}, \texttt{false}\}$ with some sort of pullback property. We propose the following definition of $\ulcorner m \urcorner$:

$$\ulcorner m \urcorner(y) \coloneqq \begin{cases} \texttt{true} & \text{if } m(x) = y \text{ for some } x \in X \\ \texttt{false} & \text{otherwise} \end{cases}$$

In other words, if we think of X as a subobject of Y, then we make $\ulcorner m \urcorner(y)$ equal to $\texttt{true}$ iff $y \in X$.

In particular, the subobject classifier property turns subsets $X \subseteq Y$ into functions $p\colon Y \to \mathbb{B}$, and vice versa. How it works is encoded in Definition 7.12, but the basic idea is that X will be the set of all things in Y that p sends to $\texttt{true}$:

$$X = \{y \in Y \mid p(y) = \texttt{true}\}. \tag{7.15}$$

This might help explain our abstract notation $\{Y \mid p\}$ in Eq. (7.13).

Exercise 7.16. Let $X = \mathbb{N} = \{0, 1, 2, \ldots\}$ and $Y = \mathbb{Z} = \{\ldots, -1, 0, 1, 2, \ldots\}$; we have $X \subseteq Y$, so consider it as a monomorphism $m\colon X \rightarrowtail Y$. It has a characteristic function $\ulcorner m \urcorner\colon Y \to \mathbb{B}$, as in Definition 7.12.

1. What is $\ulcorner m \urcorner(-5) \in \mathbb{B}$?
2. What is $\ulcorner m \urcorner(0) \in \mathbb{B}$? ◊

Exercise 7.17.

1. Consider the identity function $\mathrm{id}_{\mathbb{N}}\colon \mathbb{N} \to \mathbb{N}$. It is an injection, so it has a characteristic function $\ulcorner \mathrm{id}_{\mathbb{N}} \urcorner\colon \mathbb{N} \to \mathbb{B}$. Give a concrete description of $\ulcorner \mathrm{id}_{\mathbb{N}} \urcorner$, i.e. its exact value for each natural number $n \in \mathbb{N}$.

2. Consider the unique function $!_{\mathbb{N}}\colon \varnothing \to \mathbb{N}$ from the empty set. Give a concrete description of $\ulcorner !_{\mathbb{N}} \urcorner\colon \mathbb{N} \to \mathbb{B}$. ◊

7.2.3 Logic in the topos Set

As we said above, the subobject classifier of any topos $\mathcal{E}$ gives the setting in which to do logic. Before we explain a bit about how topos logic works in general, we continue to work concretely by focusing on logic in the topos **Set**.

Obtaining the AND operation. Consider the function $1 \to \mathbb{B} \times \mathbb{B}$ picking out the element $(\mathtt{true}, \mathtt{true})$. This is a monomorphism, so it defines a characteristic function $\ulcorner(\mathtt{true}, \mathtt{true})\urcorner\colon \mathbb{B} \times \mathbb{B} \to \mathbb{B}$. What function is it? By Eq. (7.15) the only element of $\mathbb{B} \times \mathbb{B}$ that can be sent to $\mathtt{true}$ is $(\mathtt{true}, \mathtt{true})$. Thus $\ulcorner(\mathtt{true}, \mathtt{true})\urcorner(P, Q) \in \mathbb{B}$ must be given by the following truth table

P	Q	$\ulcorner(\mathtt{true}, \mathtt{true})\urcorner(P, Q)$
true	true	true
true	false	false
false	true	false
false	false	false

This is exactly the truth table for the AND of P and Q, i.e. for $P \wedge Q$. In other words, $\ulcorner(\mathtt{true}, \mathtt{true})\urcorner = \wedge$. Note that this defines $\wedge$ as a function $\wedge\colon \mathbb{B} \times \mathbb{B} \to \mathbb{B}$, and we use the usual infix notation $x \wedge y \coloneqq \wedge(x, y)$.

Obtaining the OR operation. Let's go backwards this time. The truth table for the OR of P and Q, i.e. that of the function $\vee\colon \mathbb{B} \times \mathbb{B} \to \mathbb{B}$ defining OR, is:

P	Q	$P \vee Q$
true	true	true
true	false	true
false	true	true
false	false	false

(7.18)

If we wanted to obtain this function as the characteristic function $\ulcorner m \urcorner$ of some subset $m\colon X \subseteq \mathbb{B} \times \mathbb{B}$, what subset would X be? By Eq. (7.15), X should be the set of $y \in Y$ that are sent to $\mathtt{true}$. Thus m is the characteristic map for the three element subset

$$X = \{(\mathtt{true}, \mathtt{true}), (\mathtt{true}, \mathtt{false}), (\mathtt{false}, \mathtt{true})\} \subseteq \mathbb{B} \times \mathbb{B}.$$

To prepare for later generalization of this idea in any topos, we want a way of thinking of X only in terms of properties listed at the beginning of Section 7.2.1. In fact, one can think of X as the union of $\{\mathtt{true}\} \times \mathbb{B}$ and $\mathbb{B} \times \{\mathtt{true}\}$—a colimit of limits involving the subobject classifier and terminal object. This description will construct an analogous subobject of $\Omega \times \Omega$, and hence classify a map $\Omega \times \Omega \to \Omega$, in any topos $\mathcal{E}$.

Exercise 7.19. Every boolean has a negation, $\neg\mathtt{false} = \mathtt{true}$ and $\neg\mathtt{true} = \mathtt{false}$. The function $\neg\colon \mathbb{B} \to \mathbb{B}$ is the characteristic function of some thing, (*?*).

1. What sort of thing should (*?*) be? For example, should $\neg$ be the characteristic function of an object? A topos? A morphism? A subobject? A pullback diagram?
2. Now that you know the sort of thing (*?*) is, which thing of that sort is it? ◊

Exercise 7.20. Given two booleans P, Q, define $P \Rightarrow Q$ to mean $P = (P \wedge Q)$.

1. Write down the truth table for the statement $P = (P \wedge Q)$:

P	Q	$P \wedge Q$	$P = (P \wedge Q)$
true	true	?	?
true	false	?	?
false	true	?	?
false	false	?	?

2. If you already have an idea what $P \Rightarrow Q$ should mean, does it agree with the last column of table above?
3. What is the characteristic function $m\colon \mathbb{B} \times \mathbb{B} \to \mathbb{B}$ for $P \Rightarrow Q$?
4. What subobject does m classify? ◊

Exercise 7.21. Consider the sets $E \coloneqq \{n \in \mathbb{N} \mid n \text{ is even}\}$, $P \coloneqq \{n \in \mathbb{N} \mid n \text{ is prime}\}$, and $T \coloneqq \{n \in \mathbb{N} \mid n \geq 10\}$. Each is a subset of $\mathbb{N}$, so defines a function $\mathbb{N} \to \mathbb{B}$.

1. What is $\ulcorner E \urcorner(17)$?
2. What is $\ulcorner P \urcorner(17)$?
3. What is $\ulcorner T \urcorner(17)$?
4. Name the smallest three elements in the set classified by $(\ulcorner E \urcorner \wedge \ulcorner P \urcorner) \vee \ulcorner T \urcorner$. ◊

Review. Let's take stock of where we are and where we're going. In Section 7.1, we set out our goal of proving properties about behavior, and we said that topos theory is a good mathematical setting for doing that. We are now at the end of Section 7.2, which was about **Set** as an examplar topos. What happened?

In Section 7.2.1, we talked about properties of **Set** that are enjoyed by any topos: limits and colimits, cartesian closure, epi-mono factorizations, and subobject classifiers. Then in Section 7.2.2 we launched into thinking about the subobject classifier in general and in the specific topos **Set**, where it is the set $\mathbb{B}$ of booleans because any subset of Y is classified by a specific predicate $p\colon Y \to \mathbb{B}$. Finally, in Section 7.2.3 we discussed how to understand logic in terms of Ω: there are various maps $\wedge, \vee, \Rightarrow\colon \Omega \times \Omega \to \Omega$ and $\neg\colon \Omega \to \Omega$ etc., which serve as logical connectives. These are operations on truth values.

We have talked a lot about toposes, but we've only seen one so far: the category of sets. But we've actually seen more without knowing it: the category $\mathcal{C}$-**Inst** of instances on any database schema from Definition 3.60 is a topos. Such toposes are called *presheaf toposes* and are fundamental, but we will focus on *sheaf toposes*, because our topos of behavior types will be a sheaf topos.

Sheaves are fascinating, but highly abstract mathematical objects. They are not for the faint of mathematical heart (those who are faint of physical heart are welcome to proceed).

7.3 Sheaves

Sheaf theory began before category theory, e.g. in the form of something called "local coefficient systems for homology groups." However its modern formulation in terms of functors and sites is due to Grothendieck, who also invented toposes.

The basic idea is that rather than study spaces, we should study what happens *on* spaces. A space is merely the 'site' at which things happen. For example, if we think of the plane $\mathbb{R}^2$ as a space, we might examine only points and regions in it. But if we think of $\mathbb{R}^2$ as a site where things happen, then we might think of things like weather systems throughout the plane, or sand dunes, or trajectories and flows of material. There are many sorts of things that can happen on a space, and these are the sheaves: a sheaf on a space is roughly "a sort of thing that can happen on the space." If we want to think about points or regions from the sheaf perspective, we would consider them as different points of view on what's happening. That is, it's all about what happens on a space: the parts of the space are just perspectives from which to watch the show.

This is reminiscent of databases. The schema of a database is not the interesting part; the data is what's interesting. To be clear, the schema of a database is a site—it's acting like the space—and the category of all instances on it is a topos. In general, we can think of any small category $\mathcal{C}$ as a site; the corresponding topos is the category of functors $\mathcal{C}^{\text{op}} \to \textbf{Set}$.[5] Such functors are called *presheaves on* $\mathcal{C}$.

Did you notice that we just introduced a huge class of toposes? For any category $\mathcal{C}$, we said there is a topos of presheaves on it. So before we go on to sheaves, let's discuss this preliminary topic of presheaves. We will begin to develop some terminology and ways of thinking that will later generalize to sheaves.

7.3.1 Presheaves

Recall the definition of functor and natural transformation from Section 3.3. Presheaves are just functors, but they have special terminology that leads us to think about them in a certain geometric way.

Definition 7.22. Let $\mathcal{C}$ be a small category. A *presheaf* P on $\mathcal{C}$ is a functor $P\colon \mathcal{C}^{\text{op}} \to \textbf{Set}$. To each object $c \in \mathcal{C}$, we refer to the set $P(c)$ as *the set of sections of* P *over* c. To each morphism $f\colon c' \to c$, we refer to the function $P(f)\colon P(c) \to P(c')$ as the *restriction map along* f. For any section $s \in P(c)$, we may denote $P(f)(s) \in P(c')$, i.e. its restriction along f, by $s\big|_f$.

[5] The category of functors $\mathcal{C} \to \textbf{Set}$ is also a topos: use $\mathcal{C}^{\text{op}}$ as the defining site.

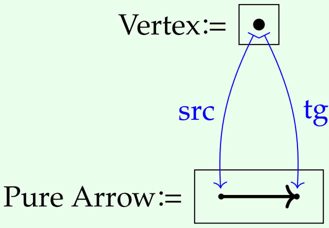
If P and Q are presheaves, a *morphism* $\alpha\colon P \to Q$ between them is a natural transformation of functors

$$\begin{tikzcd}
\mathcal{C}^{\mathrm{op}} \arrow[r, bend left, "P"{name=U}] \arrow[r, bend right, "Q"'{name=D}] & \mathbf{Set}. \arrow[from=U, to=D, Rightarrow, "\alpha"]
\end{tikzcd}$$

Example 7.23. Let **ArShp** be the category shown below:

$$\mathbf{ArShp} := \boxed{\begin{tikzcd}
\overset{\text{Vertex}}{\bullet} \arrow[r, shift left, "\text{src}"] \arrow[r, shift right, "\text{tgt}"'] & \overset{\text{Pure Arrow}}{\bullet}
\end{tikzcd}}$$

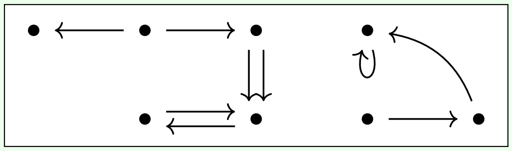
The reason we call our category **ArShp** is that we can imagine of it as an 'arrow shape.'



(7.24)

A presheaf on **ArShp** is a functor $I\colon \mathbf{ArShp}^{\mathrm{op}} \to \mathbf{Set}$, which is a database instance on $\mathbf{ArShp}^{\mathrm{op}}$. Note that $\mathbf{ArShp}^{\mathrm{op}}$ is what we called **Gr** in Section 3.3.5; there we showed that database instances on **Gr**—i.e. presheaves on **ArShp**— are just directed graphs, e.g.

$P :=$



$: \mathbf{ArShp}^{\mathrm{op}} \to \mathbf{Set}$

Thinking of presheaves on any category $\mathcal{C}$, it often makes sense to imagine the objects of $\mathcal{C}$ as shapes of some sort, and the morphisms of $\mathcal{C}$ as continuous maps between shapes, just like we did for the arrow shape in Eq. (7.24). In that context, one can think of a presheaf P as a kind of lego construction: P is built out of the shapes in $\mathcal{C}$, connected together using the morphisms in $\mathcal{C}$. In the case where $\mathcal{C}$ is the arrow shape, a presheaf is a graph. So this would say that a graph is a sort of lego construction, built out of vertices and arrows connected together using the inclusion of a vertex as the source or target of an arrow. Can you see it?

This statement can be made pretty precise; though we cannot go through it here, the above lego idea is summarized by the formal statement that "the category of presheaves on $\mathcal{C}$ is the free colimit completion of $\mathcal{C}$." Ask a friendly neighborhood category theorist for details.

However one thinks of presheaves—in terms of lego assemblies or database instances—they're relatively straightforward. The difference between presheaves and sheaves is that sheaves take into account some sort of ‘covering information.’ The trivial notion of covering is to say that every object covers itself and nothing more; if one uses this trivial covering, presheaves and sheaves are the same thing. In our behavioral context we will need a non-trivial notion of covering, so sheaves and presheaves will be slightly different. Our next goal is to understand sheaves on a topological space.

7.3.2 Topological spaces

We said in Section 7.3 that, rather than study spaces, we consider spaces as mere ‘sites’ on which things happen. We also said the things that can happen on a space are called sheaves, and always form a type of category called a topos. To define a topos of sheaves, we must start with the site on which they exist.

Sites are very abstract mathematical objects, and we will not make them precise in this book. However, one of the easiest sorts of sites to think about are those coming from topological spaces: every topological space naturally has the structure of a site. We’ve talked about spaces for a while without making them precise; let’s do so now.

Definition 7.25. Let X be a set, and let $\mathsf{P}(X) = \{U \subseteq X\}$ denote its set of subsets. A *topology* on X is a subset $\mathbf{Op} \subseteq \mathsf{P}(X)$, elements of which we call *open sets*,[6] satisfying the following conditions:

(a) Whole set: the subset $X \subseteq X$ is open, i.e. $X \in \mathbf{Op}$.

(b) Binary intersections: if $U, V \in \mathbf{Op}$ then $(U \cap V) \in \mathbf{Op}$.

(c) Arbitrary unions: if I is a set and if we are given an open set $U_i \in \mathbf{Op}$ for each i, then their union is also open, $\left(\bigcup_{i\in I} U_i\right) \in \mathbf{Op}$. We interpret the particular case where $I = \varnothing$ to mean that the empty set is open: $\varnothing \in \mathbf{Op}$.

If $U = \bigcup_{i\in I} U_i$, we say that $(U_i)_{i\in I}$ *covers* U.

A pair $(X, \mathbf{Op})$, where X is a set and $\mathbf{Op}$ is a topology on X, is called a *topological space*.

A *continuous function* between topological spaces $(X, \mathbf{Op}_X)$ and $(Y, \mathbf{Op}_Y)$ is a function $f\colon X \to Y$ such that for every $U \in \mathbf{Op}_Y$, the preimage $f^{-1}(U)$ is in $\mathbf{Op}_X$.

At the very end of Section 7.3.1 we mentioned how sheaves differ from presheaves in that they take into account ‘covering information.’ The notion of covering an open set by a union of other open sets was defined in Definition 7.25, and it will come into play when we define sheaves in Definition 7.35.

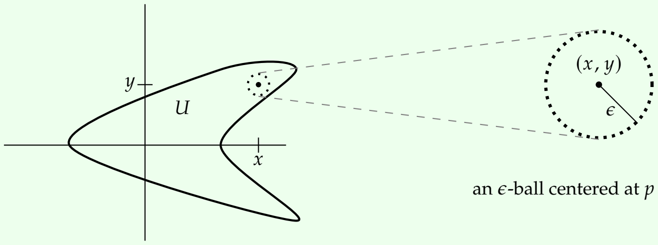
Example 7.26. The usual topology $\mathbf{Op}$ on $\mathbb{R}^2$ is based on ‘ϵ-balls.’ For any $\epsilon \in \mathbb{R}$ with $\epsilon > 0$, and any point $p = (x, y) \in \mathbb{R}^2$, define *the ϵ-ball centered at p* to be:

$$B(p;\epsilon) := \{p' \in \mathbb{R}^2 \mid d(p, p') < \epsilon\}^{7}$$

[6] In other words, we refer to a subset $U \subseteq X$ as *open* if $U \in \mathbf{Op}$.

In other words, $B(x, y; \epsilon)$ is the set of all points within ϵ of (x, y).

For an arbitrary subset $U \subseteq \mathbb{R}^2$, we call it open and put it in $\mathbf{Op}$ if, for every $(x, y) \in U$ there exists a (small enough) $\epsilon > 0$ such that $B(x, y; \epsilon) \subseteq U$.



an open set $U \subseteq \mathbb{R}^2$, a point $p = (x, y) \in U$, and an ϵ-ball $B(x, y; \epsilon) \subseteq U$.

an ϵ-ball centered at $p = (x, y)$

The same idea works if we replace $\mathbb{R}^2$ with any other metric space X (Definition 2.51): it can be considered as a topological space where the open sets are subsets U such that for any $p \in U$ there is an ϵ-ball centered at p and contained in U. So every metric space can be considered as a topological space.

Exercise 7.27. Consider the set $\mathbb{R}$. It is a metric space with $d(x_1, x_2) := |x_1 - x_2|$.

1. What is the 1-dimensional analogue of ϵ-balls as found in Example 7.26? That is, for each $x \in \mathbb{R}$, define $B(x, \epsilon)$.
2. When is an arbitrary subset $U \subseteq \mathbb{R}$ called open, in analogy with Example 7.26?
3. Find three open sets U_1, U_2, and U in $\mathbb{R}$, such that $(U_i)_{i \in \{1,2\}}$ covers U.
4. Find an open set U and a collection $(U_i)_{i \in I}$ of opens sets where I is infinite, such that $(U_i)_{i \in I}$ covers U. ◊

Example 7.28. For any set X, there is a 'coarsest' topology, having as few open sets as possible: $\mathbf{Op}_{\text{crse}} = (\varnothing, X)$. There is also a 'finest' topology, having as many open sets as possible: $\mathbf{Op}_{\text{fine}} = \mathsf{P}(X)$. The latter, $(X, \mathsf{P}(X))$ is called the *discrete space on the set* X.

Exercise 7.29.

1. Verify that for any set X, what we called $\mathbf{Op}_{\text{crse}}$ in Example 7.28 really is a topology, i.e. satisfies the conditions of Definition 7.25.
2. Verify also that $\mathbf{Op}_{\text{fine}}$ really is a topology.
3. Show that if $(X, \mathsf{P}(X))$ is discrete and $(Y, \mathbf{Op}_Y)$ is any topological space, then every function $X \to Y$ is continuous. ◊

Example 7.30. There are four topologies possible on $X = \{1, 2\}$. Two are $\mathbf{Op}_{\text{crse}}$ and

[7] Here, $d((x, y), (x', y')) := \sqrt{(x - x')^2 + (y - y')^2}$ is the usual 'Euclidean distance' between two points. One can generalize d to any metric.

$\mathbf{Op}_{\text{fine}}$ from Example 7.28. The other two are:

$$\mathbf{Op}_1 \coloneqq \{\varnothing, \{1\}, X\} \qquad \text{and} \qquad \mathbf{Op}_2 \coloneqq \{\varnothing, \{2\}, X\}$$

The two topological spaces $(\{1,2\}, \mathbf{Op}_1)$ and $(\{1,2\}, \mathbf{Op}_2)$ are isomorphic; either one can be called *the Sierpinski space*.

The open sets of a topological space form a preorder. Given a topological space $(X, \mathbf{Op})$, the set $\mathbf{Op}$ has the structure of a preorder using the subset relation, $(\mathbf{Op}, \subseteq)$. It is reflexive because $U \subseteq U$ for any $U \in \mathbf{Op}$, and it is transitive because if $U \subseteq V$ and $V \subseteq W$ then $U \subseteq W$.

Recall from Section 3.2.3 that we can regard any preorder, and hence $\mathbf{Op}$, as a category: its objects are the open sets U and for any U, V the set of morphisms $\mathbf{Op}(U, V)$ is empty if $U \not\subseteq V$ and it has one element if $U \subseteq V$.

Exercise 7.31. Recall the Sierpinski space, say $(X, \mathbf{Op}_1)$ from Example 7.30.

1. Write down the Hasse diagram for its preorder of opens.
2. Write down all the covers. ◊

Exercise 7.32. Given any topological space $(X, \mathbf{Op})$, any subset $Y \subseteq X$ can be given the *subspace topology*, call it $\mathbf{Op}_{?\cap Y}$. This topology defines any $A \subseteq Y$ to be open, $A \in \mathbf{Op}_{?\cap Y}$, if there is an open set $B \in \mathbf{Op}$ such that $A = B \cap Y$.

1. Find a $B \in \mathbf{Op}$ that shows that the whole set Y is open, i.e. $Y \in \mathbf{Op}_{?\cap Y}$.
2. Show that $\mathbf{Op}_{?\cap Y}$ is a topology in the sense of Definition 7.25.[8]
3. Show that the inclusion function $Y \hookrightarrow X$ is a continuous function. ◊

Remark 7.33. Suppose $(X, \mathbf{Op})$ is a topological space, and consider the preorder $(\mathbf{Op}, \subseteq)$ of open sets. It turns out that $(\mathbf{Op}, \subseteq, X, \cap)$ is always a quantale in the sense of Definition 2.79. We will not need this fact, but we invite the reader to think about it a bit in Exercise 7.34.

Exercise 7.34. In Sections 2.3.2 and 2.3.3 we discussed how **Bool**-categories are preorders and **Cost**-categories are Lawvere metric spaces, and in Section 2.3.4 we imagined interpretations of $\mathcal{V}$-categories for other quantales $\mathcal{V}$.

If $(X, \mathbf{Op})$ is a topological space and $\mathcal{V}$ the corresponding quantale as in Remark 7.33, how might we imagine a $\mathcal{V}$-category? ◊

7.3.3 Sheaves on topological spaces

To summarize where we are, a topological space $(X, \mathbf{Op})$ is a set X together with a bunch of subsets we call 'open'; these open subsets form a preorder—and hence category—denoted $\mathbf{Op}$. Sheaves on X will be presheaves on $\mathbf{Op}$ with a special property, aptly named the 'sheaf condition.'

[8] Hint 1: for any set I, collection of sets $(U_i)_{i\in I}$ with $U_i \subseteq X$, and set $V \subseteq X$, one has $(\bigcup_{i\in I} U_i) \cap V = \bigcup_{i\in I}(U_i \cap V)$. Hint 2: for any $U, V, W \subseteq X$, one has $(U \cap W) \cap (V \cap W) = (U \cap V) \cap W$.

Recall the terminology and notation for presheaves: a presheaf on **Op** is a functor $P\colon \mathbf{Op}^{\mathrm{op}} \to \mathbf{Set}$. Thus to every open set $U \in \mathbf{Op}$ we have a set $P(U)$, called the set of *sections over* U, and to every inclusion of open sets $V \subseteq U$ we have a function $P(U) \to P(V)$ called the *restriction*. If $s \in P(U)$ is a section over U, we may denote its restriction to V by $s\big|_V$. Recall that we say a collection of open sets $(U_i)_{i\in I}$ *covers* an open set U if $U = \bigcup_{i\in I} U_i$.

We are now ready to give the following definition, which comes in several waves: we first define matching families, then gluing, then sheaf condition, then sheaf, and finally the category of sheaves.

Definition 7.35. Let $(X, \mathbf{Op})$ be a topological space, and let $P\colon \mathbf{Op}^{\mathrm{op}} \to \mathbf{Set}$ be a presheaf on **Op**.

Let $(U_i)_{i\in I}$ be a collection of open sets $U_i \in \mathbf{Op}$ covering U. A *matching family* $(s_i)_{i\in I}$ *of P-sections over* $(U_i)_{i\in I}$ consists of a section $s_i \in P(U_i)$ for each $i \in I$, such that for every $i, j \in I$ we have

$$s_i\big|_{U_i\cap U_j} = s_j\big|_{U_i\cap U_j}.$$

Given a matching family $(s_i)_{i\in I}$ for the cover $U = \bigcup_{i\in I} U_i$, we say that $s \in P(U)$ is a *gluing*, or *glued section*, of the matching family if $s\big|_{U_i} = s_i$ holds for all $i \in I$.

If there exists a unique gluing $s \in P(U)$ for every matching family $(s_i)_{i\in I}$, we say that *P satisfies the sheaf condition for the cover $U = \bigcup_{i\in I} U_i$*. If P satisfies the sheaf condition for every cover, we say that P is a *sheaf* on $(X, \mathbf{Op})$.

Thus a sheaf is just a presheaf satisfying the sheaf condition for every open cover. If P and Q are sheaves, then a *morphism* $f\colon P \to Q$ between these sheaves is just a morphism—that is, a natural transformation—between their underlying presheaves. We denote by $\mathbf{Shv}(X, \mathbf{Op})$ the category of sheaves on X.

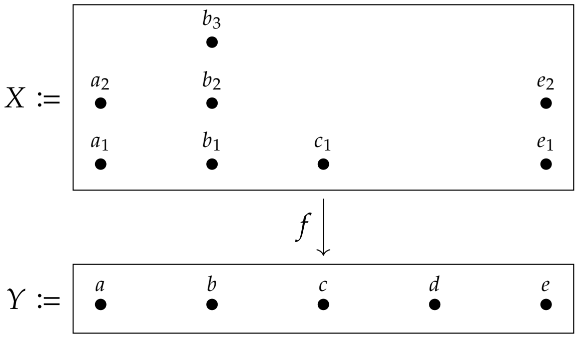
The category of sheaves on X is a topos, but we'll get to that.

Example 7.36. Here is a funny—but very important—special case to which the notion of matching family applies. We do not give this example for intuition, but because (to emphasize) it's an important and easy-to-miss case. Just like the sum of no numbers is 0 and the product of no numbers is 1, the union of no sets is the empty set. Thus if we take $U = \varnothing \subseteq X$ and $I = \varnothing$, then the empty collection of subsets (one for each $i \in I$, of which there are none) covers U. In this case the empty tuple () counts a matching family of sections, and it is the only matching family for the empty cover of the empty set.

In other words, in order for a presheaf $P\colon \mathbf{Op}^{\mathrm{op}} \to \mathbf{Set}$ to be a sheaf, a necessary (but rarely sufficient) condition is that $P(\varnothing) \cong \{()\}$, i.e. $P(\varnothing)$ must be a set with one element.

Extended example: sections of a function. This example is for intuition, and gives a case where the 'section' and 'restriction' terminology are easy to visualize.

Consider the function $f\colon X \to Y$ shown below, where each element of X is sent to the element of Y immediately below it. For example, $f(a_1) = f(a_2) = a$, $f(b_1) = b$, and so on.



(7.37)

For each point $y \in Y$, the preimage set $f^{-1}(y) \subseteq X$ above it is often called the *fiber over* y. Note that different f's would arrange the eight elements of X differently over Y: elements of Y would have different fibers.

Exercise 7.38. Consider the function $f\colon X \to Y$ shown in Eq. (7.37).

1. What is the fiber of f over a?
2. What is the fiber of f over c?
3. What is the fiber of f over d?
4. Gave an example of a function $f'\colon X \to Y$ for which every fiber has either one or two elements. ◊

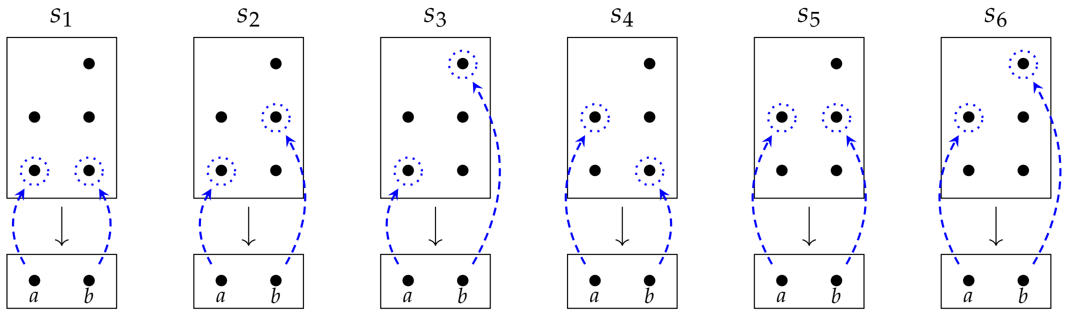
Let's consider X and Y as discrete topological spaces, so every subset is open, and f is automatically continuous (see Exercise 7.29). We will think of f as an arrangement of X over Y, in terms of fibers as above, and use it to build a sheaf on Y. To do this, we begin by building a presheaf—i.e. a functor $\mathsf{Sec}_f\colon \mathbf{Op}(Y)^{\mathrm{op}} \to \mathbf{Set}$—and then we'll prove it's a sheaf.

Define the presheaf Sec_f on an arbitrary subset $U \subseteq Y$ by:

$$\mathsf{Sec}_f(U) := \{s\colon U \to X \mid (s \mathbin{⨟} f)(u) = u \text{ for all } u \in U\}.$$

One might describe $\mathsf{Sec}_f(U)$ as the set of all ways to pick a 'cross-section' of the f arrangement over U. That is, an element $s \in \mathsf{Sec}_f(U)$ is a choice of one element per fiber over U.

As an example, let's say $U = \{a, b\}$. How many such s's are there in $\mathsf{Sec}_f(U)$? To answer this, let's clip the picture (7.37) and look only at the relevant part:



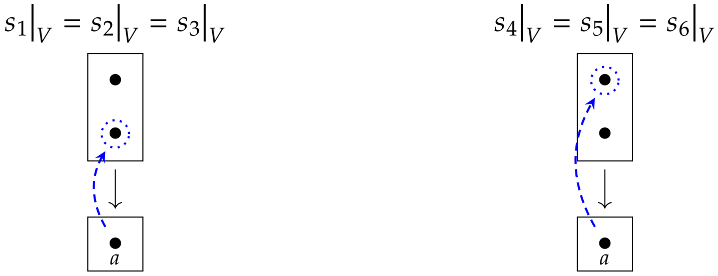
(7.39)

Looking at the picture (7.39), do you see how we get all cross-sections of f over U?

Exercise 7.40. Refer to Eq. (7.37).

1. Let $V_1 = \{a, b, c\}$. Draw all the sections over it, i.e. all elements of $\mathsf{Sec}_f(V_1)$, as we did in Eq. (7.39).
2. Let $V_2 = \{a, b, c, d\}$. Again draw all the sections, $\mathsf{Sec}_f(V_2)$.
3. Let $V_3 = \{a, b, d, e\}$. How many sections (elements of $\mathsf{Sec}_f(V_3)$) are there? ◊

By now you should understand the sections of $\mathsf{Sec}_f(U)$ for various $U \subseteq X$. This is Sec_f on objects, so you are half way to understanding Sec_f as a presheaf. That is, as a presheaf, Sec_f also includes a restriction maps for every subset $V \subseteq U$. Luckily, the restriction maps are easy: if $V \subseteq U$, say $V = \{a\}$ and $U = \{a, b\}$, then given a section s as in Eq. (7.39), we get a section over V by 'restricting' our attention to what s does on $\{a\}$.



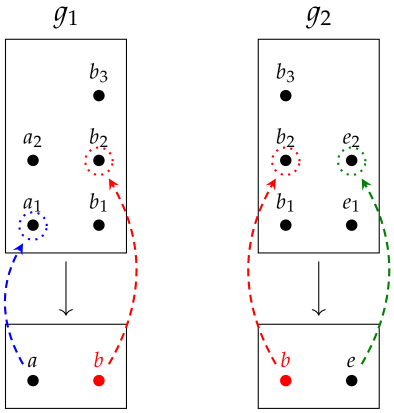
(7.41)

Exercise 7.42.

1. Write out the sets of sections $\mathsf{Sec}_f(\{a, b, c\})$ and $\mathsf{Sec}_f(\{a, c\})$.
2. Draw lines from the first to the second to indicate the restriction map. ◊

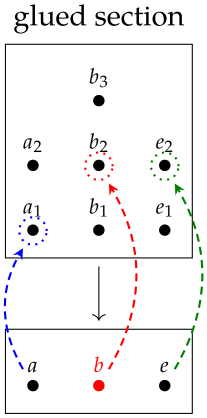
Now we have understood Sec_f as a presheaf; we next explain how to see that it is a sheaf, i.e. that it satisfies the sheaf condition for every cover. To understand the sheaf condition, consider the set $U_1 = \{a, b\}$ and $U_2 = \{b, e\}$. These cover the set $U = \{a, b, e\} = U_1 \cup U_2$. By Definition 7.35, a matching family for this cover consists of a section over U_1 and a section over U_2 that agree on the overlap set, $U_1 \cap U_2 = \{b\}$.

So consider $s_1 \in \mathsf{Sec}_f(U_1)$ and $s_2 \in \mathsf{Sec}_f(U_2)$ shown below.



(7.43)

Since sections g_1 and g_2 agree on the overlap—they both send b to b_2—the two sections shown in Eq. (7.43) can be glued to form a single section over $U = \{a, b, e\}$:



Exercise 7.44. Again let $U_1 = \{a, b\}$ and $U_2 = \{b, e\}$, so the overlap is $U_1 \cap U_2 = \{b\}$.

1. Find a section $s_1 \in \mathsf{Sec}_f(U_1)$ and a section $s_2 \in \mathsf{Sec}_f(U_2)$ that *do not* agree on the overlap.
2. For your answer (s_1, s_2) in part 1, can you find a section $s \in \mathsf{Sec}_f(U_1 \cup U_2)$ such that $s|_{U_1} = s_1$ and $s|_{U_2} = s_2$?
3. Find a section $h_1 \in \mathsf{Sec}_f(U_1)$ and a section $h_2 \in \mathsf{Sec}_f(U_2)$ that *do* agree on the overlap, but which are different than our choice in Eq. (7.43).
4. Can you find a section $h \in \mathsf{Sec}_f(U_1 \cup U_2)$ such that $h|_{U_1} = h_1$ and $h|_{U_2} = h_2$? ◊

Other examples of sheaves. The extended example above generalizes to any continuous function $f\colon X \to Y$ between topological spaces.

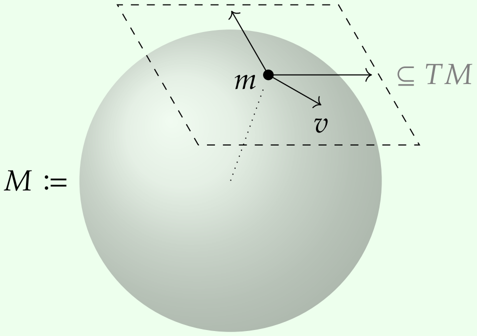
Example 7.45. Let $f\colon (X, \mathbf{Op}_X) \to (Y, \mathbf{Op}_Y)$ be a continuous function. Consider the functor $\mathsf{Sec}_f\colon \mathbf{Op}_Y^{\mathrm{op}} \to \mathbf{Set}$ given by

$$\mathsf{Sec}_f(U) := \{g\colon U \to X \mid g \text{ is continuous and } (g \mathbin{⨾} f)(u) = u \text{ for all } u \in U\},$$

The morphisms of $\mathbf{Op}_Y$ are inclusions $V \subseteq U$. Given $g: U \to X$ and $V \subseteq U$, what we call the restriction of g to V is the usual thing we mean by restriction, the same as it was in Eq. (7.41). One can again check that Sec_f is a sheaf.

Example 7.46. A nice example of a sheaf on a space M is that of vector fields on M. If you calculate the wind velocity at every point on Earth, you will have what's called a vector field on Earth. If you know the wind velocity at every point in Afghanistan and I know the wind velocity at every point in Pakistan, and our calculations agree around the border, then we can glue our information together to get the wind velocity over the union of the two countries. All possible wind velocity fields over all possible open sets of the Earth's surface together form the sheaf of vector fields.

Let's say this a bit more formally. A manifold M—you can just imagine a sphere such as the Earth's surface—always has something called a tangent bundle. It is a space TM whose points are pairs (m, v), where $m \in M$ is a point in the manifold and v is a tangent vector emanating from it. Here's a picture of one tangent plane—all the tangent vectors emanating from some fixed point—on a sphere:



The tangent bundle TM includes the whole tangent plane shown above—including the three vectors drawn on it—as well as the tangent plane at every other point on the sphere.

The tangent bundle TM on a manifold M comes with a continuous map $\pi: TM \to M$ back down to the manifold, sending $(m, v) \mapsto m$. One might say that π "forgets the tangent vector and just remembers the point it emanated from." By Example 7.45, π defines a sheaf Sec_π. It could be called the sheaf of 'tangent vector sections on M', but its usual name is the sheaf of *vector fields on M*. This is what we were describing when we spoke of the sheaf of wind velocities on Earth, above. Given an open subset $U \subseteq M$, an element $v \in \mathsf{Sec}_\pi(U)$ is called a vector field over U because it continuously assigns a tangent vector $v(u)$ to each point $u \in U$. The tangent vector at u tells us the velocity of the wind at that point.

Here's a fun digression: in the case of a spherical manifold M like the Earth, it's possible to prove that for every open set U, as long as $U \neq M$, there is a vector field

$v \in \mathsf{Sec}_\pi(U)$ that is never 0: the wind could be blowing throughout U. However, a theorem of Poincaré says that if you look at the whole sphere, there is guaranteed to be a point $m \in M$ at which the wind is not blowing at all. It's like the eye of a hurricane or perhaps a cowlick. A cowlick in someone's hair occurs when the hair has no direction to go, so it sticks up! Hair sticking up would not count as a tangent vector: tangent vectors must start out lying flat along the head. Poincaré proved that if your head was covered completely with inch-long hair, there would be at least one cowlick. This difference between local sections (over arbitrary $U \subseteq X$) and global sections (over X)—namely that hair can be well combed whenever $U \neq X$ but cannot be well combed when $U = X$—can be thought of as a generative effect, and can be measured by cohomology (see Section 1.5).

Exercise 7.47. If M is a sphere as in Example 7.46, we know from Definition 7.35 that we can consider the category $\mathbf{Shv}(M)$ of sheaves on M; in fact, such categories are toposes and these are what we're getting to.

But are the sheaves on M the vector fields? That is, is there a one-to-one correspondence between sheaves on M and vector fields on M? If so, why? If not, how are sheaves on M and vector fields on M related? ◊

Example 7.48. For every topological space $(X, \mathbf{Op})$, we have the topos of sheaves on it. The topos of sets, which one can regard as the story of set theory, is the category of sheaves on the one-point space $\{*\}$. In topos theory, we see the category of sets—an huge, amazing, and rich category—as corresponding to a single point. Imagine how much more complex arbitrary toposes are, when they can take place on much more interesting topological spaces (and in fact even more general 'sites').

Exercise 7.49. Consider the Sierpinski space $(\{1,2\}, \mathbf{Op}_1)$ from Example 7.30.

1. What is the category $\mathbf{Op}$ for this space? (You may have already figured this out in Exercise 7.31; if not, do so now.)
2. What does a presheaf on $\mathbf{Op}$ consist of?
3. What is the sheaf condition for $\mathbf{Op}$?
4. How do we identify a sheaf on $\mathbf{Op}$ with a function? ◊

7.4 Toposes

A *topos* is defined to be a category of sheaves.[9] So for any topological space $(X, \mathbf{Op})$, the category $\mathbf{Shv}(X, \mathbf{Op})$ defined in Definition 7.35 is a topos. In particular, taking the one-point space $X = \mathbf{1}$ with its unique topology, we find that the category $\mathbf{Set}$ is a topos, as we've been saying all along and saw again explicitly in Example 7.48. And for any

[9] This is sometimes called a *sheaf topos* or a *Grothendieck topos*. There is a more general sort of topos called an *elementary topos* due to Lawvere.

database schema—i.e. finitely presented category—$\mathcal{C}$, the category $\mathcal{C}$-**Inst** of database instances on $\mathcal{C}$ is also a topos.[10] Toposes encompass both of these sources of examples, and many more.

Toposes are incredibly nice structures, for a variety of seemingly disparate reasons. In this sketch, the reason in focus is that every topos has many of the same structural properties that the category **Set** has. Indeed, we discussed in Section 7.2.1 that every topos has limits and colimits, is cartesian closed, has epi-mono factorizations, and has a subobject classifier (see Section 7.2.2). Using these properties, one can do logic with semantics in the topos $\mathcal{E}$. We explained this for sets, but now imagine it for sheaves on a topological space. There, the same logical symbols $\wedge, \vee, \neg, \Rightarrow, \exists, \forall$ become operations that mean something about sub-sheaves—e.g. vector fields, sections of continuous functions, etc.—not just subsets.

To understand this more deeply, we should say what the subobject classifier $\mathtt{true}\colon 1 \to \Omega$ is in more generality. We said that, in the topos **Set**, the subobject classifier is the set of booleans $\Omega = \mathbb{B}$. In a sheaf topos $\mathcal{E} = \mathbf{Shv}(X, \mathbf{Op})$, the object $\Omega \in \mathcal{E}$ is a sheaf, not just a set. What sheaf is it?

7.4.1 The subobject classifier Ω in a sheaf topos

In this subsection we aim to understand the subobject classifier Ω, i.e. the object of truth values, in the sheaf topos $\mathbf{Shv}(X, \mathbf{Op})$. Since Ω is a sheaf, let's understand it by going through the definition of sheaf (Definition 7.35) slowly in this case. A sheaf Ω is a presheaf that satisfies the sheaf condition. As a presheaf it is just a functor $\Omega\colon \mathbf{Op}^{\mathrm{op}} \to \mathbf{Set}$; it assigns a set $\Omega(U)$ to each open $U \subseteq X$ and comes with a restriction map $\Omega(U) \to \Omega(V)$ whenever $V \subseteq U$. So in our quest to understand Ω, we first ask the question: what presheaf is it?

The answer to our question is that Ω is the presheaf that assigns to $U \in \mathbf{Op}$ the set of open subsets of U:

$$\Omega(U) \coloneqq \{U' \in \mathbf{Op} \mid U' \subseteq U\}. \tag{7.50}$$

That was easy, right? And given the restriction map for $V \subseteq U$ is given by

$$\begin{aligned} \Omega(U) &\to \Omega(V) \\ U' &\mapsto U' \cap V. \end{aligned} \tag{7.51}$$

One can check that this is functorial—see Exercise 7.53—and after doing so we will still need to see that it satisfies the sheaf condition. But at least we don't have to struggle to understand Ω: it's a lot like **Op** itself.

[10] We said that a topos is a category of sheaves, yet database instances are presheaves; so how is $\mathcal{C}$-**Inst** a topos? Well, presheaves in fact count as sheaves. We apologize that this couldn't be clearer. All of this could be made formal if we were to introduce *sites*. Unfortunately, that concept is simply too abstract for the scope of this chapter.

Exercise 7.52. Let $X = \{1\}$ be the one point space. We said above that its subobject classifier is the set $\mathbb{B}$ of booleans, but how does that align with the definition of Ω given in Eq. (7.50)? ◊

Exercise 7.53.

1. Show that the definition of Ω given above in Eqs. (7.50) and (7.51) is functorial, i.e., that whenever $W \subseteq V \subseteq U$, the restriction map $\Omega(U) \to \Omega(V)$ followed by the restriction map $\Omega(V) \to \Omega(W)$ is the same as the restriction map $\Omega(U) \to \Omega(W)$.
2. Is that all that's necessary to conclude that Ω is a presheaf? ◊

To see that Ω as defined in Eq. (7.50) satisfies the sheaf condition (see Definition 7.35), suppose that we have a cover $U = \bigcup_{i \in I} U_i$, and suppose given an element $V_i \in \Omega(U_i)$, i.e. an open set $V_i \subseteq U_i$, for each $i \in I$. Suppose further that for all $i, j \in I$, it is the case that $V_i \cap U_j = V_j \cap U_i$, i.e. that the elements form a matching family. Define $V \coloneqq \bigcup_{i \in I} V_i$; it is an open subset of U, so we can consider V as an element of $\Omega(U)$. The following verifies that V is indeed a gluing for the $(V_i)_{i \in I}$:

$$V \cap U_j = \left(\bigcup_{i \in I} V_i\right) \cap U_j = \bigcup_{i \in I} (V_i \cap U_j) = \bigcup_{i \in I} (V_j \cap U_i) = \left(\bigcup_{i \in I} U_i\right) \cap V_j = V_j$$

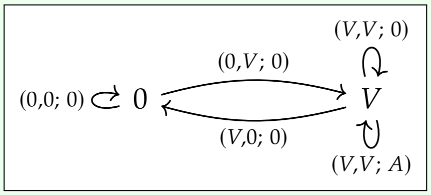
In other words $V \cap U_j = V_j$ for any $j \in I$. So our Ω has been upgraded from presheaf to sheaf!

The eagle-eyed reader will have noticed that we haven't yet given all the data needed to define a subobject classifier. To turn the object Ω into a subobject classifier in good standing, we also need to give a sheaf morphism $\texttt{true}\colon \{1\} \to \Omega$. Here $\{1\}\colon \mathbf{Op}^{\mathrm{op}} \to \mathbf{Set}$ is the terminal sheaf; it maps every open set to the terminal, one element set $\{1\}$. The correct morphism $\texttt{true}\colon \{1\} \to \Omega$ for the subobject classifier is the sheaf morphism that assigns, for every $U \in \mathbf{Op}$ the function $\{1\} = \{1\}(U) \to \Omega(U)$ sending $1 \mapsto U$, the largest open set $U \subseteq U$. From now on we denote $\{1\}$ simply as 1.

Upshot: Truth values are open sets. The point is that the truth values in the topos of sheaves on a space $(X, \mathbf{Op})$ are the open sets of that space. When someone says "is property P true?," the answer is not yes or no, but "it is true on the open subset U." If this U is everything, $U = X$, then P is really true; if U is nothing, $U = \varnothing$, then P is really false. But in general, it's just true some places and not others.

Example 7.54. The category **Grph** of graphs is a presheaf topos, and one can also think of it as the category of instances for a database schema, as we saw in Example 7.23. The subobject classifier Ω in the topos **Gr** is thus a graph, so we can draw it. Here's what it looks like:

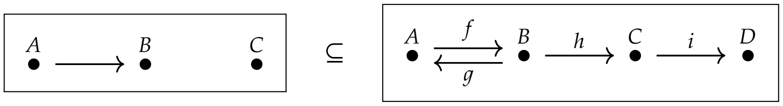
$\Omega_{\mathbf{Grph}} =$



Finding Ω for oneself is easiest using something called the Yoneda Lemma, but we have not introduced it. For a nice, easy introduction to the topos of graphs, see [Vig03]. The terminal graph is a single vertex with a single loop, and the graph homomorphism `true`: $1 \to \Omega$ sends that loop to $(V, V;\ A)$.

Given any graph G and subgraph $i\colon H \subseteq G$, we need to construct a graph homomorphism $\ulcorner H \urcorner\colon G \to \Omega$ classifying H. The idea is that for each part of G, we decide "how much of it is in H. A vertex in v in G is either in H or not; if so we send it to V and if not we send it to 0. But arrows a are more complicated. If a is in H, we send it $(V, V; A)$. But if it is not in H, the mathematics requires us to ask more questions: is its source in H? is its target in G"? both? neither? Based on the answers to these questions we send a to $(V, 0;\ 0)$, $(0, V;\ 0)$, $(V, V;\ 0)$, or $(0, 0;\ 0)$, respectively.

Exercise 7.55. Consider the subgraph $H \subseteq G$ shown here:



Find the graph homomorphism $\ulcorner H \urcorner\colon G \to \Omega$ classifying it. See Example 7.54. ◊

7.4.2 Logic in a sheaf topos

Let's consider the logical connectives, AND, OR, IMPLIES, and NOT. Suppose we have a topological space $X \in \mathbf{Op}$. Given two open sets U, V, considered as truth values $U, V \in \Omega(X)$, then their conjunction 'U AND V' is their intersection, and their disjunction 'U OR V' is their union;

$$(U \wedge V) := U \cap V \qquad \text{and} \qquad (U \vee V) := U \cup V. \tag{7.56}$$

These formulas are easy to remember, because $\wedge$ looks like $\cap$ and $\vee$ looks like $\cup$. The implication $U \Rightarrow V$ is the largest open set R such that $R \cap U \subseteq V$, i.e.

$$(U \Rightarrow V) := \bigcup_{\{R \in \mathbf{Op} \mid R \cap U \subseteq V\}} R. \tag{7.57}$$

In general, it is not easy to reduce Eq. (7.57) further, so implication is the hardest logical connective to think about topologically.

Finally, the negation of U is given by $\neg U := (U \Rightarrow \texttt{false})$, and this turns out to be relatively simple. By the formula in Eq. (7.57), it is the union of all R such that $R \cap U = \varnothing$, i.e. the union of all open sets in the complement of U. If you know topology, you might recognize that $\neg U$ is the 'interior of the complement of U.'

Example 7.58. Consider the real line $X = \mathbb{R}$ as a topological space (see Exercise 7.27). Let $U, V \in \Omega(X)$ be the open sets $U = \{x \in \mathbb{R} \mid x < 3\}$ and $V = \{x \in \mathbb{R} \mid -4 < x < 4\}$. Using interval notation, $U = (-\infty, 3)$ and $V = (-4, 4)$. Then

- $U \wedge V = (-4, 3)$.
- $U \vee V = (-\infty, 4)$.
- $\neg U = (3, \infty)$.
- $\neg V = (-\infty, -4) \cup (4, \infty)$.
- $(U \Rightarrow V) = (-4, \infty)$
- $(V \Rightarrow U) = U$

Exercise 7.59. Consider the real line $\mathbb{R}$ as a topological space, and consider the open subset $U = \mathbb{R} - \{0\}$.

1. What open subset is $\neg U$?
2. What open subset is $\neg\neg U$?
3. Is it true that $U \subseteq \neg\neg U$?
4. Is it true that $\neg\neg U \subseteq U$? ◊

Above we explained operations on open sets, one corresponding to each logical connective; there are also open sets corresponding to the the symbols `true` and `false`. We explore this in an exercise.

Exercise 7.60. Let $(X, \mathbf{Op})$ be a topological space.

1. Suppose the symbol `true` corresponds to an open set such that for any open set $U \in \mathbf{Op}$, we have $(\texttt{true} \wedge U) = U$. Which open set is it?
2. Other things we should expect from `true` include $(\texttt{true} \vee U) = \texttt{true}$ and $(U \Rightarrow \texttt{true}) = \texttt{true}$ and $(\texttt{true} \Rightarrow U) = U$. Do these hold for your answer to 1?
3. The symbol `false` corresponds to an open set $U \in \mathbf{Op}$ such that for any open set $U \in \mathbf{Op}$, we have $(\texttt{false} \vee U) = U$. Which open set is it?
4. Other things we should expect from `false` include $(\texttt{false} \wedge U) = \texttt{false}$ and $(\texttt{false} \Rightarrow U) = \texttt{true}$. Do these hold for your answer to 1? ◊

Example 7.61. For a vector bundle $\pi\colon E \to X$ over a space X, the corresponding sheaf is Sec_π corresponding to its sections: to each open set $i_U\colon U \subseteq X$, we associate the set of functions $s\colon U \to E$ for which $s \mathbin{⨾} \pi = i_U$. For example, in the case of the tangent bundle $\pi\colon TM \to M$ (see Example 7.46), the corresponding sheaf, call it `VF`, associates to each U the set $\texttt{VF}(U)$ of vector fields on U.

The internal logic of the topos can then be used to consider properties of vector fields. For example, one could have a predicate $\texttt{Grad}\colon \texttt{VF} \to \Omega$ that asks for the largest subspace $\texttt{Grad}(v)$ on which a given vector field v comes from the gradient of some scalar function. One could also have a predicate that asks for the largest open set on which a vector field is non-zero. Logical operations like $\wedge$ and $\vee$ could then be applied to hone in on precise submanifolds throughout which various desired properties hold, and to reason logically about what other properties are forced to hold there.

7.4.3 Predicates

In English, a predicate is the part of the sentence that comes after the subject. For example "…is even" or "…likes the weather" are predicates. Not every subject makes sense for a given predicate; e.g. the sentence "7 is even" may be false, but it makes sense. In contrast, the sentence "2.7 is even" does not really make sense, and "2.7 likes the weather" certainly doesn't. In computer science, they might say "The expression '2.7 likes the weather' does not type check."

The point is that each predicate is associated to a type, namely the type of subject that makes sense for that predicate. When we apply a predicate to a subject of the appropriate type, the result has a truth value: "7 is even" is either true or false. Perhaps "Bob likes the weather" is true some days and false on others. In fact, this truth value might change by the year (bad weather this year), by the season, by the hour, etc. In English, we expect truth values of sentences to change over time, which is exactly the motivation for this chapter. We're working toward a logic where truth values change over time.

In a topos $\mathcal{E} = \mathbf{Shv}(X, \mathbf{Op})$, a predicate is a sheaf morphism $p\colon S \to \Omega$ where $S \in \mathcal{E}$ is a sheaf and $\Omega \in \mathcal{E}$ is the subobject classifier, the sheaf of truth values. By Definition 7.35 we get a function $p(U)\colon S(U) \to \Omega(U)$ for any open set $U \subseteq X$. In the above example—which we will discuss more carefully in Section 7.5—if S is the sheaf of people (people come and go over time), and Bob $\in S(U)$ is a person existing over a time U, and p is the predicate "likes the weather," then p(Bob) is the set of times during which Bob likes the weather. So the answer to "Bob likes the weather" is something like "in summers yes, and also in April 2018 and May 2019 yes, but in all other times no." That's p(Bob), the temporal truth value obtained by applying the predicate p to the subject Bob.

Exercise 7.62. Just now we described how a predicate $p\colon S \to \Omega$, such as "…likes the weather," acts on sections $s \in S(U)$, say $s =$ Bob. But by Definition 7.12, any predicate $p\colon S \to \Omega$ also defines a subobject of $\{S \mid p\} \subseteq S$. Describe the sections of this subsheaf. ◊

The poset of subobjects. For a topos $\mathcal{E} = \mathbf{Shv}(X, \mathbf{Op})$ and object (sheaf) $S \in \mathcal{E}$, the set of S-predicates $|\Omega^E| = \mathcal{E}(S, \Omega)$ is naturally given the structure of a poset, which we denote

$$(|\Omega^S|, \leq^S) \tag{7.63}$$

Given two predicates $p, q\colon S \to \Omega$, we say that $p \leq^S q$ if the first implies the second. More precisely, for any $U \in \mathbf{Op}$ and section $s \in S(U)$ we obtain two open subsets $p(s) \subseteq U$ and $q(s) \subseteq U$. We say that $p \leq^S q$ if $p(s) \subseteq q(s)$ for all $U \in \mathbf{Op}$ and $s \in S(U)$. We often drop the superscript from $\leq^S$ and simply write $\leq$. In formal logic notation, one might write $p \leq^S q$ using the $\vdash$ symbol, e.g. in one of the following ways:

$$s : S \mid p(s) \vdash q(s) \qquad \text{or} \qquad p(s) \vdash_{s:S} q(s).$$

In particular, if $S = 1$ is the terminal object, we denote $|\Omega^S|$ by $|\Omega|$, and refer to elements $p \in |\Omega|$ as *propositions*. They are just morphisms $p \colon 1 \to \Omega$.

This preorder is partially ordered—a poset—meaning that if $p \leq q$ and $q \leq p$ then $p = q$. The reason is that for any subsets $U, V \subseteq X$, if $U \subseteq V$ and $V \subseteq U$ then $U = V$.

Exercise 7.64. Give an example of a space X, a sheaf $S \in \mathbf{Shv}(X)$, and two predicates $p, q \colon S \to \Omega$ for which $p(s) \vdash_{s:S} q(s)$ holds. You do not have to be formal. ◊

All of the logical symbols ($\mathtt{true}, \mathtt{false}, \wedge, \vee, \Rightarrow, \neg$) from Section 7.4.2 make sense in any such poset $|\Omega^S|$. For any two predicates $p, q \colon S \to \Omega$, we define $(p \wedge q) \colon S \to \Omega$ by $(p \wedge q)(s) \coloneqq p(s) \wedge q(s)$, and similarly for $\vee$. Thus one says that these operations are *computed pointwise* on S. With these definitions, the $\wedge$ symbol is the meet and the $\vee$ symbol is the join—in the sense of Definition 1.81—for the poset $|\Omega^S|$.

With all of the logical structure we've defined so far, the poset $|\Omega^S|$ of predicates on S forms what's called a Heyting algebra. We will not define it here, but more information can be found in Section 7.6. We now move on to quantification.

7.4.4 Quantification

Quantification comes in two flavors: universal and existential, or 'for all' and 'there exists.' Each takes in a predicate of $n+1$ variables and returns a predicate of n variables.

Example 7.65. Suppose we have two sheaves $S, T \in \mathbf{Shv}(X, \mathbf{Op})$ and a predicate $p \colon S \times T \to \Omega$. Let's say T represents what's considered newsworthy and S is again the set of people. So for a subset of time U, a section $t \in T(U)$ is something that's considered newsworthy throughout the whole of U, and a section $s \in S(U)$ is a person that lasts throughout the whole of U. Let's imagine the predicate p as "s is worried about t." Now recall from Section 7.4.3 that a predicate p does not simply return true or false; given a person s and a news-item t, it returns a truth value corresponding to the subset of times on which $p(s, t)$ is true.

"For all t in T, ... is worried about t" is itself a predicate on just one variable, S, which we denote

$$\forall(t : T).\, p(s, t).$$

Applying this predicate to a person s returns the times when that person is worried about everything in the news. Similarly, "there exists t in T such that s is worried about t" is also a predicate on S, which we denote $\exists(t : T).\, p(s, t)$. If we apply this predicate to a person s, we get the times when person s is worried about at least one thing in the news.

Exercise 7.66. In the topos **Set**, where $\Omega = \mathbb{B}$, consider the predicate $p\colon \mathbb{N} \times \mathbb{Z} \to \mathbb{B}$ given by

$$p(n,z) = \begin{cases} \texttt{true} & \text{if } n \le |z| \\ \texttt{false} & \text{if } n > |z|. \end{cases}$$

1. What is the set of $n \in \mathbb{N}$ for which the predicate $\forall(z : \mathbb{Z}).\, p(n,z)$ holds?
2. What is the set of $n \in \mathbb{N}$ for which the predicate $\exists(z : \mathbb{Z}).\, p(n,z)$ holds?
3. What is the set of $z \in \mathbb{Z}$ for which the predicate $\forall(n : \mathbb{N}).\, p(n,z)$ holds?
4. What is the set of $z \in \mathbb{Z}$ for which the predicate $\exists(n : \mathbb{N}).\, p(n,z)$ holds? ◊

So given p, we have a universally- and an existentially-quantified predicate $\forall(t : T).\, p(s,t)$ and $\exists(t : T).\, p(s,t)$ on S. How do we formally understand them as sheaf morphisms $S \to \Omega$ or, equivalently, as subsheaves of S?

Universal quantification. Given a predicate $p\colon S \times T \to \Omega$, the universally-quantified predicate $\forall(t : T).\, p(s,t)$ takes a section $s \in S(U)$, for any open set U, and returns a certain open set $V \in \Omega(U)$. Namely, it returns the largest open set $V \subseteq U$ for which $p(s|_V, t) = V$ holds for all $t \in T(V)$.

Exercise 7.67. Suppose s is a person alive throughout the interval U. Apply the above definition to the example $p(s,t) =$ "person s is worried about news t" from Example 7.65. Here, $T(V)$ is the set of items that are in the news throughout the interval V.

1. What open subset of U is $\forall(t : T).\, p(s,t)$ for a person s?
2. Does it have the semantic meaning you'd expect, given the less formal description in Section 7.4.4? ◊

Abstractly speaking, the universally-quantified predicate corresponds to the subsheaf given by the following pullback:

$$\begin{tikzcd}
\forall_t p \arrow[r] \arrow[d, tail] \arrow[dr, phantom, "\lrcorner", very near start] & 1 \arrow[d, tail, "\texttt{true}^T"] \\
S \arrow[r, "p'"'] & \Omega^T
\end{tikzcd}$$

where $p'\colon S \to \Omega^T$ is the currying of $S \times T \to \Omega$ and $\texttt{true}^T$ is the currying of the composite $1 \times T \xrightarrow{!} 1 \xrightarrow{\texttt{true}} \Omega$. See Eq. (7.10).

Existential quantification. Given a predicate $p\colon S \times T \to \Omega$, the existentially quantified predicate $\exists(t : T).\, p(s,t)$ takes a section $s \in S(U)$, for any open set U, and returns a certain open set $V \in \Omega(U)$, namely the union $V = \bigcup_i V_i$ of all the open sets V_i for which there exists some $t_i \in T(V_i)$ satisfying $p(s|_{V_i}, t_i) = V_i$. If the result is U itself, you might be tempted to think "ah, so there exists some $t \in T(U)$ satisfying $p(t)$," but that is not necessarily so. There is just a cover of $U = \bigcup U_i$ and local sections $t_i \in T(U_i)$, each satisfying p, as explained above. Thus the existential quantifier is doing a lot of work "under the hood," taking coverings into account without displaying that fact in the notation.

Exercise 7.68. Apply the above definition to the "person s is worried about news t" predicate from Example 7.65.

1. What open set is $\exists(t : T).\, p(s,t)$ for a person s?
2. Does it have the semantic meaning you'd expect? ◊

Abstractly speaking, the existentially-quantified predicate is given as follows. Start with the subobject classified by p, namely $\{(s,t) \in S \times T \mid p(s,t)\} \subseteq S \times T$, compose with the projection $\pi_S \colon S \times T \to S$ as on the upper right; then take the epi-mono factorization of the composite as on the lower left:

$$\begin{tikzcd}
\{S \times T \mid p\} \arrow[r, tail] \arrow[d, two heads] & S \times T \arrow[d, "\pi_S"] \\
\exists_t p \arrow[r, tail] & S
\end{tikzcd}$$

Then the bottom map is the desired subsheaf of S.

7.4.5 Modalities

Back in Example 1.123 we discussed modal operators—also known as modalities—saying they are closure operators on preorders which arise in logic. The preorders we were referring to are the ones discussed in Eq. (7.63): for any object $S \in \mathcal{E}$ there is the poset $(|\Omega^S|, \leq^S)$ of predicates on S, where $|\Omega^S| = \mathcal{E}(S, \Omega)$ is just the set of morphisms $S \to \Omega$ in the category $\mathcal{E}$.

Definition 7.69. A *modality* in $\mathbf{Shv}(X)$ is a sheaf morphism $j \colon \Omega \to \Omega$ satisfying three properties for all $U \subseteq X$ and $p, q \in \Omega(U)$:

(a) $p \leq j(p)$;
(b) $(j \mathbin{⨟} j)(p) \leq j(p)$; and
(c) $j(p \wedge q) = j(p) \wedge j(q)$.

Exercise 7.70. Suppose $j \colon \Omega \to \Omega$ is a morphism of sheaves on X, such that $p \leq j(p)$ holds for all $U \subseteq X$ and $p \in \Omega(U)$. Show that for all $q \in \Omega(U)$ we have $j(j(q)) \leq j(q)$ iff $j(j(q)) = j(q)$. ◊

In Example 1.123 we informally said that for any proposition p, e.g. "Bob is in San Diego," there is a modal operator "assuming p, ..." Now we are in a position to make that formal.

Proposition 7.71. Fix a proposition $p \in |\Omega|$. Then

(a) the sheaf morphism $\Omega \to \Omega$ given by sending q to $p \Rightarrow q$ is a modality.
(b) the sheaf morphism $\Omega \to \Omega$ given by sending q to $p \vee q$ is a modality.
(c) the sheaf morphism $\Omega \to \Omega$ given by sending q to $(q \Rightarrow p) \Rightarrow p$ is a modality.

We cannot prove Proposition 7.71 here, but we give references in Section 7.6.

Exercise 7.72. Let S be the sheaf of people as in Section 7.4.3, and let $j\colon \Omega \to \Omega$ be "assuming Bob is in San Diego..."

1. Name any predicate $p\colon S \to \Omega$, such as "likes the weather."
2. Choose a time interval U. For an arbitrary person $s \in S(U)$, what sort of thing is $p(s)$, and what does it mean?
3. What sort of thing is $j(p(s))$ and what does it mean?
4. Is it true that $p(s) \leq j(p(s))$? Explain briefly.
5. Is it true that $j(j(p(s))) = j(p(s))$? Explain briefly.
6. Choose another predicate $q\colon S \to \Omega$. Is it true that $j(p \wedge q) = j(p) \wedge j(q)$? Explain briefly. ◊

7.4.6 Type theories and semantics

We have been talking about the logic of a topos in terms of open sets, but this is actually a conflation of two ideas that are really better left unconflated. The first is logic, or formal language, and the second is semantics, or meaning. The formal language looks like this:

$$\forall(t : T).\, \exists(s : S).\, f(s) = t \tag{7.73}$$

and semantic statements are like "the sheaf morphism $f\colon S \to T$ is an epimorphism." In the former, logical world, all statements are linguistic expressions formed according to strict rules and all proofs are deductions that also follow strict rules. In the latter, semantic world, statements and proofs are about the sheaves themselves, as mathematical objects. We admit these are rough statements; again, our aim here is only to give a taste, an invitation to further reading.

To *provide semantics* for a logical system means to provide a compiler that converts each logical statement in the formal language into a mathematical statement about particular sheaves and their relationships. A computer can carry out logical deductions without knowing what any of them "mean" about sheaves. We say that semantics is *sound* if every formal proof is converted into a true fact about the relevant sheaves.

Every topos can be assigned a formal language, often called its *internal language*, in which to carry out constructions and formal proofs. This language has a sound semantics—a sort of logic-to-sheaf compiler—which goes under the name *categorical semantics* or *Kripke-Joyal semantics*. We gave the basic ideas in Section 7.4; we give references to the literature in Section 7.6.

Example 7.74. In every topos $\mathcal{E}$, and for every $f\colon S \to T$ in $\mathcal{E}$, the morphism f is an epimorphism if and only if Eq. (7.73) holds. For example, consider the case of database instances on a schema $\mathcal{C}$, say with 100 tables (one of which might be denoted $c \in \mathrm{Ob}(\mathcal{C})$) and 500 foreign key columns (one of which might be denoted $f\colon c \to c'$ in $\mathcal{C}$); see Eq. (3.2).

If S and T are two instances and f is a natural transformation between them, then we can ask the question of whether or not Eq. (7.73) holds. This simple formula is compiled by the Kripke-Joyal semantics into asking:

> Is it true that for every table $c \in \text{Ob}(\mathcal{C})$ and every row $s \in S(c)$ there exists a row $t \in T(c)$ such that $f(s) = t$?

This is exactly what it means for f to be surjective. Maybe this is not too impressive, but whether one is talking about databases or topological spaces, or complex ideas from algebraic geometry, Eq. (7.73) always compiles into the question of surjectivity. For topological spaces it would say something like:

> Is it true that for every open set $U \subseteq X$ and every section $s \in S(U)$ of the bundle S, there exists an open covering of $(U_i \subseteq U)_{i \in I}$ of U and a section $t_i \in T(U_i)$ of the bundle T for each $i \in I$, such that $f(t_i) = s\big|_{U_i}$ is the restriction of s to U_i?

7.5 A topos of behavior types

Now that we have discussed logic in a sheaf topos, we return to our motivating example, a topos of behavior types. We begin by discussing the topological space on which behavior types will be sheaves, a space called the *interval domain*.

Remark 7.75. Note that above, we were thinking very intuitively about time, e.g. when we discussed people being worried about the news. Now we will be thinking about time in a different way, but there is no need to change your answers or reconsider the intuitive thinking done above.

7.5.1 The interval domain

The interval domain $\mathbb{IR}$ is a specific topological space, which we will use to model intervals of time. In other words, we will be interested in the category $\mathbf{Shv}(\mathbb{IR})$ of sheaves on the interval domain.

To give a topological space, one must give a pair $(X, \mathbf{Op})$, where X is a set of 'points' and $\mathbf{Op}$ is a topology on X; see Definition 7.25. The set of points for $\mathbb{IR}$ is that of all finite closed intervals

$$\mathbb{IR} := \{[d, u] \subseteq \mathbb{R} \mid d \leq u\}.$$

For $a < b$ in $\mathbb{R}$, let $o_{[a,b]}$ denote the set $o_{[a,b]} := \{[d, u] \in \mathbb{IR} \mid a < d \leq u < b\}$; these are called *basic open sets*. The topology $\mathbf{Op}$ is determined by these basic open sets in that a subset U is open if it is the union of some collection of basic open sets.

Thus for example, $o_{[0,5]}$ is an open set: it contains every $[d, u]$ contained in the open interval $\{x \in \mathbb{R} \mid 0 < x < 5\}$. Similarly $o_{[4,8]}$ is an open set, but note that $o_{[0,5]} \cup o_{[4,8]} \neq o_{[0,8]}$. Indeed, the interval $[2, 6]$ is in the right-hand side but not the left.

Exercise 7.76.

1. Explain why $[2,6] \in o_{[0,8]}$.
2. Explain why $[2,6] \notin o_{[0,5]} \cup o_{[4,8]}$. ◊

Let **Op** denote the open sets of $\mathbb{IR}$, as described above, and let $\mathbf{BT} \coloneqq \mathbf{Shv}(\mathbb{IR}, \mathbf{Op})$ denote the topos of sheaves on this space. We call it the topos of *behavior types*.

There is an important subspace of $\mathbb{IR}$, namely the usual space of real numbers $\mathbb{R}$. We see $\mathbb{R}$ as a subspace of $\mathbb{IR}$ via the isomorphism

$$\mathbb{R} \cong \{[d,u] \in \mathbb{IR} \mid d = u\}.$$

We discussed the usual topology on $\mathbb{R}$ in Example 7.26, but we also get a topology on $\mathbb{R}$ because it is a subset of $\mathbb{IR}$; i.e. we have the subspace topology as described in Exercise 7.32. These agree, as the reader can check.

Exercise 7.77. Show that a subset $U \subseteq \mathbb{R}$ is open in the subspace topology of $\mathbb{R} \subseteq \mathbb{IR}$ iff $U \cap \mathbb{R}$ is open in the usual topology on $\mathbb{R}$ defined in Example 7.26. ◊

7.5.2 Sheaves on $\mathbb{IR}$

We cannot go into much depth about the sheaf topos $\mathbf{BT} = \mathbf{Shv}(\mathbb{IR}, \mathbf{Op})$, for reasons of space; we refer the interested reader to Section 7.6. In this section we will briefly discuss what it means to be a sheaf on $\mathbb{IR}$, giving a few examples including that of the subobject classifier.

What is a sheaf on $\mathbb{IR}$? A sheaf S on the interval domain $(\mathbb{IR}, \mathbf{Op})$ is a functor $S\colon \mathbf{Op}^{\mathrm{op}} \to \mathbf{Set}$: it assigns to each open set U a set $S(U)$; how should we interpret this? An element $s \in S(U)$ is something that says is an "event that takes place throughout the interval U." Given this U-event s together with an open subset of $V \subseteq U$, there is a V-event $s\big|_V$ that tells us what s is if we regard it as an event taking place throughout V. If $U = \bigcup_{i \in I} U_i$ and we can find matching U_i-events (s_i) for each $i \in I$, then the sheaf condition (Definition 7.35) says that they have a unique gluing, i.e. a U-event $s \in S(U)$ that encompasses all of them: $s\big|_{U_i} = s_i$ for each $i \in I$.

We said in Section 7.5.1 that every open set $U \subseteq \mathbb{IR}$ can be written as the union of basic open sets $o_{[a,b]}$. This implies that any sheaf S is determined by its values $S(o_{[a,b]})$ on these basic open sets. The sheaf condition furthermore implies that these vary continuously in a certain sense, which we can express formally as

$$S(o_{[a,b]}) \cong \lim_{\epsilon > 0} S(o_{[a-\epsilon, b+\epsilon]}).$$

However, rather than get into the details, we describe a few sorts of sheaves that may be of interest.

Example 7.78. For any set A there is a sheaf $\mathtt{A} \in \mathbf{Shv}(\mathbb{IR})$ that assigns to each open set U the set $\mathtt{A}(U) \coloneqq A$. This allows us to refer to integers, or real numbers, or letters of an alphabet, as though they were behaviors. What sort of behavior is $7 \in \mathbb{N}$? It is the sort of behavior that never changes: it's always seven. Thus $\mathtt{A}$ is called the *constant sheaf on* A.

Example 7.79. Fix any topological space $(X, \mathbf{Op}_X)$. Then there is a sheaf F_X of *local functions from* $\mathbb{IR}$ *to* X. That is, for any open set $U \in \mathbf{Op}_{\mathbb{IR}}$, we assign the set $F_X(U) \coloneqq \{f \colon U \to X \mid f \text{ is continuous}\}$. There is also the sheaf G_X of local functions on the subspace $\mathbb{R} \subseteq \mathbb{IR}$. That is, for any open set $U \in \mathbf{Op}_{\mathbb{IR}}$, we assign the set $G_X(U) \coloneqq \{f \colon U \cap \mathbb{R} \to X \mid f \text{ is continuous}\}$.

Exercise 7.80. Let's check that Example 7.78 makes sense. Fix any topological space $(X, \mathbf{Op}_X)$ and any subset $R \subseteq \mathbb{IR}$ of the interval domain. Define $H_X(U) \coloneqq \{f \colon U \cap R \to X \mid f \text{ is continuous}\}$.

1. Is H_X a presheaf? If not, why not; if so, what are the restriction maps?
2. Is H_X a sheaf? Why or why not? ◊

Example 7.81. Another source of examples comes from the world of open hybrid dynamical systems. These are machines whose behavior is a mixture of continuous movements—generally imagined as trajectories through a vector field—and discrete jumps. These jumps are imagined as being caused by signals that spontaneously arrive. Over any interval of time, a hybrid system has certain things that it can do and certain things that it cannot. Although we will not make this precise here, there is a construction for converting any hybrid system into a sheaf on $\mathbb{IR}$; we will give references in Section 7.6.

We refer to sheaves on $\mathbb{IR}$ as behavior types because almost any sort of behavior one can imagine is a behavior type. Of course, a complex behavior type—such as the way someone acts when they are in love—would be extremely hard to write down. But the idea is straightforward: for any interval of time, say a three-day interval $(d, d+3)$, let $L(d, d+3)$ denote the set of all possible behaviors a person who is in love could possibly do. Obviously it's a big, unwieldy set, and no one would want to make precise. But to the extent that one can imagine that sort of behavior as occurring through time, they could imagine the corresponding sheaf.

The subobject classifier as a sheaf on $\mathbb{IR}$. In any sheaf topos, the subobject classifier Ω is itself a sheaf. It is responsible for the truth values in the topos. As we said in Section 7.4.1, when it comes to sheaves on a topological space $(X, \mathbf{Op})$, truth values are open subsets $U \in \mathbf{Op}$.

BT is the topos of sheaves on the space $(\mathbb{IR}, \mathbf{Op})$, as defined in Section 7.5.1. As always, the subobject classifier Ω assigns to any $U \in \mathbf{Op}$ the set of open subsets of U, so these are the truth values. But what do they mean? The idea is that every proposition, such as “Bob likes the weather” returns an open set U, as if to respond that Bob likes the weather “...throughout time period U.” Let’s explore this just a bit more.

Suppose Bob likes the weather throughout the interval $(0, 5)$ and throughout the interval $(4, 8)$. We would probably conclude that Bob likes the weather throughout the interval $(0, 8)$. But what about the more ominous statement “a single pair of eyes has remained watching position p.” Then just because it’s true on $(0, 5)$ and on $(4, 8)$, does not imply that it’s been true on $(0, 8)$: there may have been a change of shift, where one watcher was relieved from their post by another watcher. As another example, consider the statement “the stock market did not go down by more than 10 points.” This might be true on $(0, 5)$ and true on $(4, 8)$ but not on $(0, 8)$. In order to capture the semantics of statements like these—statements that take time to evaluate—we must use the space $\mathbb{IR}$ rather than the space $\mathbb{R}$.

7.5.3 Safety proofs in temporal logic

We now have at least a basic idea of what goes into a proof of safety, say for autonomous vehicles, or airplanes in the national airspace system. In fact, the underlying ideas of this chapter came out of a project between MIT, Honeywell Inc., and NASA [SSV18]. The background for the project was that the National Airspace System consists of many different systems interacting: interactions between airplanes, each of which is an interaction between physics, humans, sensors, and actuators, each of which is an interaction between still more basic parts. The same sort of story would hold for a fleet of autonomous vehicles, as in the introduction to this chapter.

Suppose that each of the systems—at any level—is guaranteed to satisfy some property. For example, perhaps we can assume that an engine is either out of gas, has a broken fuel line, or is following the orders of a human driver or pilot. If there is a rupture in the fuel line, the sensors will alert the human within three seconds, etc. Each of the components interact with a number of different variables. In the case of airplanes, a pilot interacts with the radio, the positions of the dials, the position of the thruster, and the visual data in front of her. The component—here the pilot—is guaranteed to keep these variables in some relation: “if I see something, I will say something” or “if the dials are in position `bad_pos`, I will engage the thruster within 1 second.” We call these guarantees *behavior contracts*.

All of the above can be captured in the topos **BT** of behavior types. The variables are behavior types: the altimeter is a variable whose value $\theta \in \mathbb{R}_{\geq 0}$ is changing continuously with respect to time. The thruster is also a continuously-changing variable whose value is in the range $[0, 1]$, etc.

The guaranteed relationships—behavior contracts—are given by predicates on variables. For example, if the pilot will always engage the thruster within one second of the display dials being in position `bad_pos`, this can be captured by a predicate $p\colon \texttt{dials} \times \texttt{thrusters} \to \Omega$. While we have not written out a formal language for p, one could imagine the predicate $p(D, T)$ for D : $\texttt{dials}$ and T : $\texttt{thrusters}$ as

$$\forall(t : \mathbb{R}).\, @_t\big(\texttt{bad\_pos}(D)\big) \Rightarrow \\ \exists(r : \mathbb{R}).\, (0 < r < 1) \wedge \forall(r' : \mathbb{R}).\, 0 \leq r' \leq 5 \Rightarrow @_{t+r+r'}\big(\texttt{engaged}(T)\big). \quad (7.82)$$

Here $@_t$ is a modality, as we discussed in Definition 7.69; in fact it turns out to be one of type 3. from Proposition 7.71, but we cannot go into that. For a proposition q, the statement $@_t(q)$ says that q is true in some small enough neighborhood around t. So (7.82) says "starting within one second of whenever the dials say that we are in a bad position, I'll engage the thrusters for five seconds."

Given an actual playing-out-of-events over a time period U, i.e. actual section $D \in \texttt{dials}(U)$ and section $T \in \texttt{thrusters}(U)$, the predicate Eq. (7.82) will hold on certain parts of U and not others, and this is the truth value of p. Hopefully the pilot upholds her behavior contract at all times she is flying, in which case the truth value will be $\texttt{true}$ throughout that interval U. But if the pilot breaks her contract over certain intervals, then this fact is recorded in Ω.

The logic allows us to record axioms like that shown in Eq. (7.82) and then reason from them: e.g. if the pilot and the airplane, and at least one of the three radars upholds its contract then safe separation will be maintained. We cannot give further details here, but these matters have been worked out in detail in [SS18]; see Section 7.6.

7.6 Summary and further reading

This chapter was about modeling various sorts of behavior using sheaves on a space of time-intervals. Behavior may seem like it's something that occurs now in the present, but in fact our memory of past behavior informs what the current behavior means. In order to commit to anything, to plan or complete any sort of process, one needs to be able to reason over time-intervals. The nice thing about temporal sheaves—indeed sheaves on any site—is that they fit into a categorical structure called a topos, which has many useful formal properties. In particular, it comes equipped with a higher-order logic with which we can formally reason about how temporal sheaves work together when combined in larger systems. A much more detailed version of this story was presented in [SS18]. But it would have been impossible without the extensive edifice of topos theory and domain theory that has been developed over the past six decades.

Sheaf toposes were invented by Grothendieck and his school in the 1960s [AGV71] as an approach to proving conjectures at the intersection of algebraic geometry and number theory, called the Weil conjectures. Soon after, Lawvere and Tierney recognized that toposes had all the structure necessary to do logic, and with a whole host of other category theorists, the subject was developed to an impressive extent in many directions. For a much more complete history, see [McL90].

There are many sorts of references on topos theory. One that starts by introducing categories and then moves to toposes, focusing on logic, is [McL92]. Our favorite treatment is perhaps [MM92], where the geometric aspects play a central role. Finally, Johnstone has done the field a huge favor by collecting large amounts of the theory into a single two-volume set [Joh02]; it is very dense, but an essential reference for the serious student or researcher. For just categorical (Kripke-Joyal) semantics of logic in a topos, one should see either [MM92], [Jac99], or [LS88].

We did not mention domain theory much in this chapter, aside from referring to the interval domain. But domains, in the sense of Dana Scott, play an important role in the deeper aspects of temporal type theory. A good reference is [Gie+03], but for an introduction we suggest [AJ94].

In some sense our application area has been a very general sort of dynamical system. Other categorical approaches to this subject include [JNW96], [HTP03], [AS05], and [Law86], though there are many others.

We hope you have enjoyed the seven sketches in this book. As a next step, consider running a reading course on applied category theory with some friends or colleagues. Simultaneously, we hope you begin to search out categorical ways of thinking about familiar subjects. Perhaps you'll find something you want to contribute to this growing field of applied category theory, or as we sometimes call it, the field of compositionality.