2 Prelude to Sheaves: Presheaves

In which we climb up the ladder of abstraction—scaling up the "morphisms are what's important" paradigm—by introducing morphisms between categories (functors) and morphisms between those (natural transformations), and then develop a store of examples of, and perspectives on, certain sorts of functors called presheaves, *a construction on which the central object of this book depends.*

It is often said that category theory privileges relations over objects, and in the last chapter you were exposed to what we might call the "morphisms are what's really important" perspective. As you meet more and more examples of categories, and especially as you begin to appreciate how a category is itself a mathematical construction with characteristic structure of its own intrinsic interest, a natural set of questions arise: if we treat categories themselves as objects, do we have a notion of morphisms between categories? And if so, what do the morphisms between categories look like? Is the model of preservation of structure still a good one? And if so, what is the relevant structure of a category that will have to be preserved by morphisms between categories?

Morphisms between categories are supplied by *functors*. Here is where the remarkable power of these categorical notions really starts to kick in, and where we lean even further into the "morphisms are what's really important" philosophy. To begin to appreciate where this might lead us, recall the arrow category, defined last chapter in definition 20, where one just takes the morphisms of a given category as the objects of a new category. What if, extending this idea, we now regarded functors as objects? Is there a notion of morphisms between functors? Yes, these are *natural transformations*. These constructions interact in an astoundingly rich manner, and we can continue further with this same controlled scaling procedure.

This chapter introduces functors and natural transformations, considers a wide variety of examples of functors and exhibits the many things we can do with them, introduces the concept of a *presheaf* (another name for a certain type of functor) and a category built out of presheaves, and uses a variety of examples of presheaves to further reflect on the different classes of things the presheaf does. The central object of this book, the sheaf, depends on this more general concept of the presheaf. In the introduction, we mentioned that sheaves effectively attach information locally to regions of some "space," doing so in a way that permits passage from local to global. Slightly more formally, the space can be made into a category, and so can the data. Then a presheaf will be a certain principled

association of the first category to the second one, where a sheaf will amount to a presheaf that satisfies some further conditions. This will all be substantially improved upon. For the present chapter, we focus on developing a good understanding of functors and presheaves.

2.1 Functors

If a category is a context for studying a specific type of mathematical object and the network of relations entertained between those objects, a *functor* is a principled way of comparing categories, translating the objects and actions of one category into objects and actions in another category in such a way that certain structural relations are preserved through this translation. As a way of moving in a controlled way *between* categories, one can initially think of a functor as doing any of the following things: specifying data locally; producing a "picture" of the source category inside the target category, modeling the one category or some aspect of that category within another; taking advantage of the methods available in the target category to analyze the source category; converting a problem in one category into another where the solution might be more readily apparent; realizing an abstract theory of some structured notion (such as a group) in a certain background or on a specific "stage"; forgetting or deliberately losing some information, perhaps in order to examine or identify those features more robust to variations or to ease computation. But underneath these different interpretations or uses is a very simple requirement: a functor just transforms objects and maps in the source category into objects and maps in the target category, in such a way that two equations (amounting to the preservation of the structure supplied by identities and composites) are satisfied. Functors also come in two flavors, depending on their direction or variance. Formally,

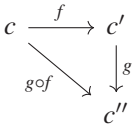
Definition 26 A *(covariant) functor* $F : \mathbf{C} \to \mathbf{D}$ between categories $\mathbf{C}$ and $\mathbf{D}$ is an assignment of

1. an object $F(c) \in \mathbf{D}$ for every object $c \in \mathrm{Ob}(\mathbf{C})$; and
2. a morphism $F(f) : F(c) \to F(c')$ in $\mathbf{D}$ for every morphism $c \to c'$ in $\mathbf{C}$,

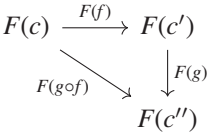
which assignments satisfy the following two axioms:

1. For any object c in $\mathbf{C}$, $F(\mathrm{id}_c) = \mathrm{id}_{F(c)}$ ("F of the identity on c is the identity on $F(c)$");
2. For any composable pair f, g in $\mathbf{C}$, $F(g) \circ F(f) = F(g \circ f)$.

Observe that this last condition just states that F of a composite of two morphisms in $\mathbf{C}$ is the composite (in $\mathbf{D}$) of their images under F, that is, whenever we have



commutative in $\mathbf{C}$, then the induced diagram

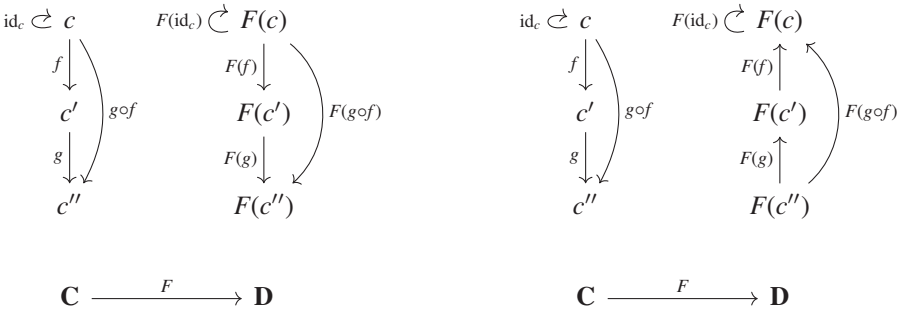


commutes in $\mathbf{D}$.

While such functors preserve the "direction" of morphisms—since the source of a morphism in **C** is assigned to the source of the image morphism in **D**, and the same for targets—there are plenty of constructions throughout mathematics that would supply us with examples of functors, in that they seem to do everything a functor does, except that they reverse direction by taking sources to targets and targets to sources. The notion of a *contravariant functor* lets us accommodate such things.

A *(contravariant) functor* F from category **C** to category **D** is defined in the same way on objects, but differently on morphisms (where the source and target are swapped). Explicitly, to each morphism $f : c \to c' \in \mathbf{C}$ a contravariant functor F assigns a morphism $F(f) : F(c') \to F(c) \in \mathbf{D}$. This assignment must satisfy the same identity axiom $F(\mathrm{id}_c) = \mathrm{id}_{F(c)}$ as above, but for any composable pair f, g in **C**, we must now have $F(f) \circ F(g) = F(g \circ f)$ (note the change in order of composition).

All the information of this definition is displayed below (the covariant case on the left and contravariant case on the right, and with identity maps omitted except for on one of the objects):



We will write $F : \mathbf{C} \to \mathbf{D}$, or $\mathbf{C} \xrightarrow{F} \mathbf{D}$, to indicate that F is a functor from **C** to **D**. Functors will usually be denoted with upper-case letters (F, G, etc.), though we may occasionally use a more evocative name to indicate what the functor does.

Remark 27 Observe that by simply reversing the direction of all the morphisms in the category **C**—that is, using the opposite category $\mathbf{C}^{op}$, as defined in definition 19—and then just using a *covariant* functor, we recover the notion of a contravariant functor on **C**. On account of this, in principle a contravariant functor $F : \mathbf{C} \to \mathbf{D}$ can always be replaced by a *covariant* one $F : \mathbf{C}^{op} \to \mathbf{D}$, using the opposite category for the source category. Accordingly, whenever we speak of "functor," we will by default mean *covariant functor*, and to handle contravariant functors we will just speak of (implicitly covariant) functors and use the opposite category for our source category.

The truth of the frequently cited claim of Eilenberg and Mac Lane that "the whole concept of a category is essentially an auxiliary one; our basic concepts are essentially those of a functor and of a natural transformation"[22] proves itself in time to anyone who works with

22. Eilenberg and Mac Lane (1945, 247). Natural transformations, the morphisms between functors, are introduced in the next section of this chapter.

categories. Before helping the reader to better appreciate this for themselves by exploring a variety of examples of functors, let us use the notion of a functor to introduce a few other items of interest.

In addition to their intrinsic interest, functors are of special interest to us because of their essential role in the definition of *presheaves*—a concept that, as the name suggests, will be rather important to the development of sheaves.

Definition 28 A (set-valued) *presheaf* on $\mathbf{C}$—where $\mathbf{C}$ is assumed to be a small category—is a functor $\mathbf{C}^{op} \to \mathbf{Set}$.[23]

As we will see, a presheaf can often be thought of as consisting of some specification or assignment of local data, according to the "shape" of the domain category; a sheaf will emerge as a special sort of presheaf in that its local data can be glued or patched together locally. Before addressing in more detail the nature of presheaves, we give another important definition and then provide some examples of functors in general.

By now the reader should be comfortable with the core idea of taking mathematical objects of a certain type together with their (possibly structure-preserving) morphisms and assembling this into a category. In this same spirit, functors define morphisms between categories, there is a natural notion of composition of functors as well as a functor that acts as the identity, and the composition of functors can be shown to respect the axioms on associativity and identities. So categories and functors can be assembled into a category of their own!

Definition 29 The *category of (small) categories*, denoted **Cat**, is the category that has

- objects: small categories;
- morphisms: functors between them.

To verify that this is indeed a category, we need to establish identity morphisms (functors), that the composition of two functors (when defined) leaves us with a functor, and that all this data satisfies the axioms on associativity and identity. Explicitly, we will need the following very boring, but ultimately quite useful, functor.

Definition 30 Given a category $\mathbf{C}$, the *identity functor* is the functor $\mathrm{id}_{\mathbf{C}} : \mathbf{C} \to \mathbf{C}$ that does what you would expect it to do. Explicitly, it takes an object to itself and a morphism to itself, that is,

- $\mathrm{id}_{\mathbf{C}}(c) = c$,
- $\mathrm{id}_{\mathbf{C}}(f) = f$.

As for composites: letting $\mathbf{C}$, $\mathbf{D}$, and $\mathbf{E}$ be (small) categories, and $F : \mathbf{C} \to \mathbf{D}$ and $G : \mathbf{D} \to \mathbf{E}$ be (covariant) functors, we can then define the composition of G with F, or composite functor $G \circ F : \mathbf{C} \to \mathbf{E}$, on objects c of $\mathbf{C}$ by $(G \circ F)(c) := G(F(c))$, and on morphisms $f : c \to c'$ of $\mathbf{C}$ by $(G \circ F)(f) := G(F(f))$. We can then show that the composition of two functors

23. Incidentally, as a presheaf is just a contravariant *functor* from a category $\mathbf{C}$ to **Set**, the reader may wonder why we give it two names. Such a reader might find useful the fun notion, used by the nLab authors, of a *concept with an attitude*—meant to capture those situations in math when one and the same concept is given two different names, one of the names indicating a specific perspective or attitude suggesting what to do with the objects, or the sorts of things one might expect to be able to do with them. In renaming a (set-valued) contravariant functor as a presheaf, then, we have a concept with an attitude, specifically looking forward to *sheaves*.

is a functor. Letting f, g be morphisms of **C** such that their composite $g \circ f$ is defined, we have

$$\begin{aligned}(G \circ F)(g \circ f) &= G(F(g \circ f)) \\ &= G(F(g) \circ F(f)) \\ &= G(F(g)) \circ G(F(f)) \\ &= (G \circ F)(g) \circ (G \circ F)(f),\end{aligned}$$

where the first and last lines are by definition of the composition of functors and the middle two use the fact that F and G are assumed to be functors themselves. Moreover, observe that for any object c of **C**, we have

$$\begin{aligned}(G \circ F)(\mathrm{id}_c) &= G(F(\mathrm{id}_c)) \\ &= G(\mathrm{id}_{F(c)}) \\ &= \mathrm{id}_{G(F(c))} \\ &= \mathrm{id}_{(G \circ F)(c)}.\end{aligned}$$

This shows that the composite $G \circ F$ of two functors is itself a (covariant) functor.[24] It is straightforward to show that, as morphisms, functors obey the associativity and identity axioms, thus ensuring that **Cat** indeed forms a category.

Remark 31 In section 1.3, we raised some size issues and important distinctions related to size. In considering possible definitions of the category of categories, the same sorts of size issues take on an even greater significance. In definition 29, we implicitly tried to get ahead of some of these lurking size issues by only counting small categories among our objects. Observe that the resulting category **Cat** itself will be locally small, yet not small. This is good, since it means that, being large itself, **Cat** won't be an object of itself, and so we skirt any issues analogous to Russell's paradox. However, in taking only small categories for objects, categories like **Set**, **Pos**, **Mon** will not be found among the *objects* of **Cat**. (Though they are subcategories.)

Naturally, this raises a question: if such nonsmall (large) categories cannot be found among the objects of **Cat**, is there a category that *does* include them among its objects? We will denote such a category, which admits large categories as objects and the functors between them as morphisms, by **CAT**. Again hoping to avoid problems analogous to the paradox discussed in section 1.3, we are inspired to stipulate that the objects in **CAT** still be locally small; defined thus, **CAT** for its part would not be locally small, and so it is not in danger of being an object of itself.

In this book, we won't worry too much further about differences between these two categories, and will mostly just find ourselves dealing with **Cat**.

2.1.1 Examples of Functors

Functors appear all over mathematics. But perhaps the lowest-hanging examples of functors can be found by looking at those established mathematical structures of a certain type

24. Relying on this fact, and in order to avoid certain notational infelicities like $(G \circ F)(g \circ f)$, we will occasionally use the juxtaposition notation GF for composite functors, where this is understood to be the same as $G \circ F$.

that individually assemble into a category and thereby supply us with a particularly simple way in to categories as objects of study in their own right. In these cases, we would expect that a functor between such objects, now each regarded as an individual category in its own right, would recover the usual important structure-preserving relations that are expected to obtain among such objects as they appear in their native setting. This is what is illustrated by the following two examples.

Example 32 Recall from example 3 (chapter 1) that a preorder $\mathcal{X} := (X, \leq)$ is traditionally defined as a set X together with a reflexive and transitive binary relation $\leq$. Recall also that we can transform a given preorder into a category $\mathcal{X}$ by defining, for every pair of objects $x, x' \in X$, the hom-set $\mathrm{Hom}_{\mathcal{X}}(x, x')$ as either empty (in case the pair (x, x') is not related by $\leq$) or as consisting of the unique morphism $x \to x'$ (just in case $x \leq x'$), making the composition formula completely determined. In other words, it is a category that has at most one morphism between any two objects.

If $\mathcal{X} := (X, \leq_X)$ and $\mathcal{Y} := (Y, \leq_Y)$ are two preorders, regarded as categories, then what is a functor $F : \mathcal{X} \to \mathcal{Y}$? First of all, it assigns an object x in the set $\mathrm{Ob}(\mathcal{X}) = X$ to the object $F(x) \in Ob(\mathcal{Y}) = Y$. In other words, it acts as a function on objects. Given a morphism $f : x \to x'$ in $\mathcal{X}$ (from which, from the traditional perspective of the preorder, we are just saying that $x \leq x'$), by the definition of a functor this will get sent to $F(x) \to F(x')$ in $\mathcal{Y}$, which will moreover be unique (and corresponds, at the level of the preorder, to saying that $F(x) \leq F(x')$). But this just says, at the preorder level, that $x \leq x'$ implies $F(x) \leq F(x')$. In other words, with the notion of a functor between preorders treated as categories we have recovered the usual notion of maps between preorders, namely a monotone map, as defined in definition 4. Moreover, it is easy to see that a contravariant functor between preorders regarded as categories will just recover the notion of an *antitone (order-reversing) map*, that is, whenever $x \leq x'$, then $f(x') \leq f(x)$.

Example 33 Recall from example 9 (chapter 1) that each monoid (and each group) can be regarded as its own category. Explicitly, we saw that a monoid $(M, e, \cdot)$ can be considered as a category $\mathcal{M}$ with one object and with hom-set equal to M, where the identity morphism comes from the monoid identity e and the composition formula from the monoid multiplication $\cdot : M \times M \to M$. Given two monoids $(M, e, \cdot)$ and $(N, e', \star)$, where these are regarded as one-object categories $\mathcal{M}$ and $\mathcal{N}$, we might hope that a (covariant) functor from one to the other would just recover the usual notion of a morphism between monoids, a *monoid homomorphism*, where this is defined as a map $\phi : M \to N$ that respects the structure in the sense that

$$\phi(m \cdot m') = \phi(m) \star \phi(m') \text{ and } \phi(e) = e'.$$

One can immediately see that the above equations are the same as defining a covariant functor between $\mathcal{M}$ and $\mathcal{N}$, when these are each regarded as a category. A contravariant functor from $\mathcal{M}$ to $\mathcal{N}$, for its part, is exactly a monoid morphism that flips the elements, that is, $\phi(m \cdot m') = \phi(m') \star \phi(m)$.

Since a group is just a monoid in which every element is invertible, a similar story can of course be told using groups. A group can be regarded as a category with one object such that every morphism is an isomorphism, and then a functor between such categories recovers the notion of a group homomorphism.

Before leaving this example, we might also mention a few other prominent functors relating the mathematical structures under consideration. There exists a functor $Core$: **Mon** $\rightarrow$ **Group** that ingests a monoid $(M, e, \cdot)$ and spits out the subset of invertible elements of that monoid—which of course leaves us with a group, typically called the *core* of the monoid M. There is a related functor **Cat** $\rightarrow$ **Grpd** sending a category **C** to the largest groupoid inside **C**, also called its core.[25] It's also worth mentioning that when we said that each monoid could be regarded as its own category we were really just appealing to the fact that there is a functor **Mon** $\rightarrow$ **Cat** that takes a monoid to its corresponding category!

Moving beyond examples where the functors pass between categories that are fundamentally the same type of structure, we need to begin to appreciate some of the other important things functors do.

Example 34 In many settings, one might want to transfer one system of objects that present themselves in a certain way in one context to another context where irrelevant or undesirable (e.g., noisy) features are suppressed, while simultaneously preserving certain basic qualitative features. In the definition of a category, and indeed in the definition of many mathematical objects, typically one specifies (1) underlying data, together with (2) some extra structure, which in turn may satisfy (3) some properties. One obvious thing to do when considering some category **C** is to deliberately lose or ignore some or all of the structure or the properties carried by the source category. This process informally describes *forgetful functors*, which provide us with a large source of examples.

There are many examples where **Set** is the target category, since many important categories are sets with some structure; however, forgetful functors need not have **Set** for the target category. For instance, since a group is just a monoid $(M, e, \cdot)$ with the extra property that every element $m \in M$ has an inverse, this means that to every group we can assign its underlying monoid and every group homomorphism will get assigned to a monoid homomorphism between its underlying monoids—this is carried out simply by forgetting the extra conditions on a group. Thus, there is a forgetful functor U: **Group** $\rightarrow$ **Mon**.

While the "forgetting" terminology might suggest some sort of (possibly pejorative) loss of information, another way of looking at the same process is that it extracts and emphasizes only the important features of the objects under study. An illustration of this comes from *detectors*, which in practice often act to forget or lose information carried by a signal, while preserving fundamental features of the underlying signal. This is exactly what is useful about such tools, since what is removed is clutter, leaving us with a compressed representation of the original information (with the effect that the result of applying the functor might be more robust to variations, more relevant to a particular application, simpler for computation, etc.).

A *signal* is essentially a collection of (local) measurements related to one another, and the topology associated with these measurements tells us how a measurement is affected by noise; for instance, a signal over a discrete set is typically either not changed by noise at all or it changes drastically, while a signal over a smoother space may depend less drastically

25. A groupoid is just like a group except that it can have more than one object, as the discussion of oidification in section 1.5 would suggest. More formally, a *groupoid* is a category such that every morphism is an isomorphism; a morphism between groupoids is also just a functor. As such, we could then define a group as a groupoid with only one object.

on perturbations.[26] As already anticipated, as a forgetful functor, a detector acts to *remove* something—specifically, it acts to remove topological structure from the signal (which may have the effect of quantizing signals)—but, as a functor, it should also preserve certain features of the signal.

As a specific and simple instance of this, consider a *threshold detector*.[27] A detector can just be regarded as a functor from some category of signal data into some subcategory of **Set**. We can describe a threshold detector as a detector that ingests a continuous (real-valued) function $f \in \mathbf{Cont}(\mathbb{R})$ and spits out the open set on which $f(x) > T$ for a given threshold $T \in \mathbb{R}$. The domain of this functor will be the category $\mathbf{Cont}(\mathbb{R})$ which has for objects the continuous real-valued functions and for morphisms $f \to g$ whenever $f(x) > g(x)$ for all $x \in \mathbb{R}$. The threshold detector is thus a functor D that assigns to each $f \in \mathbf{Cont}(\mathbb{R})$ the open set $D(f) = \{x \in \mathbb{R} \mid f(x) > T\}$, that is, it lands in the category $\mathbf{Open}(\mathbb{R})$ of open sets of $\mathbb{R}$, whose morphisms are given by subset inclusion. Moreover, one can see that if $f \to g$, then we will have $D(g) \subseteq D(f)$, making D a *contravariant* functor from $\mathbf{Cont}(\mathbb{R})$ to $\mathbf{Open}(\mathbb{R})$, that is, our threshold detector D is a functor $D : \mathbf{Cont}(\mathbb{R})^{op} \to \mathbf{Open}(\mathbb{R})$.

In general, forgetful functors frequently can tell us interesting things about the source category. For instance, we have a functor $U : \mathbf{Cat} \to \mathbf{Grph}$, which informs us that categories have underlying graphs. Recall from the definition of a directed graph (see example 5, chapter 1) that a graph $G = (V, A, s, t)$ just consists of a set V of vertices, a set A of edges (which will be directed, so we also call them arrows), and a pair of functions $s, t : A \to V$ codifying the direction of the edges (arrows) by assigning to each $a \in A$ its source vertex $s(a)$ and target vertex $t(a)$. In general, we have been displaying the objects and morphisms of a category as the vertices and edges of a directed graph, and we draw directed graphs the way we draw categories—which may superficially suggest graphs and categories are fundamentally "the same" sort of thing. But observe that, as defined, directed graphs just consist of vertices and edges—where the directedness of the edges is codified by information regarding designated source and target vertices—and a priori there is no notion of composition of edges and no identities to speak of.

To appreciate how the functor in question works, consider that, when defining a category, we could have equally defined a (small) category by saying that the data of the category involves a set of objects (let's denote this by C_0), a set of morphisms (denoted C_1), and a diagram $C_1 \underset{t}{\overset{s}{\rightrightarrows}} C_0$, together with some structure (composition and identities) and properties (identity and associativity axioms). Really, the additional structure and properties concerning composition and identities can be codified by supplementing the previous diagram with another map $i : C_0 \to C_1$ assigning identity arrows to each object, and another set $C_2 := \{(f, g) \in C_1 \times C_1 \mid t(f) = s(g)\}$ together with a partial operation capturing composition $C_2 \overset{\circ}{\to} C_1$. In this way, we are effectively describing a small category $\mathbf{C}$ with the diagram

$$C_2 \xrightarrow{\circ} C_1 \underset{t}{\overset{s}{\underset{\leftarrow i \rightarrow}{\rightrightarrows}}} C_0$$

26. All matters of topology—including the open sets and related topological matters invoked in the next paragraphs—are discussed extensively in chapter 4.

27. This idea for this threshold example comes from Robinson (2014).

where the expected behaviors of compositions and identities are codified by further equations. Comparing this formulation to the definition of a directed graph, it should come as no surprise that there is a functor sending categories to their underlying directed graph. What the functor U does is take the objects of a category to vertices of its underlying graph and the morphisms to the graph's directed edges (arrows), where the associated source and target functions of the graph agree with the source (domain) and target (codomain) assignments of the category. As there is no further structure or conditions having to do with composition of arrows or with identities in a graph, U basically "forgets" anything having to do with specifications of identities and composition. More explicitly, U can be seen as taking the category **C**—described by the diagram above—to the underlying graph

$$C_1 \underset{t}{\overset{s}{\rightrightarrows}} C_0$$

and then acting on morphisms (functors) by ignoring parts of the category diagram that contain information about i and $\circ$ (since graph homomorphisms, as simply a mapping of edges to edges and vertices to vertices that preserves sources and targets, effectively have nothing to say about matters of identities and compositions).

Forgetful functors often come paired with corresponding *free functors*. For instance, corresponding to the "underlying graph" functor U, there exists the *free category functor* $F : \mathbf{Grph} \to \mathbf{Cat}$, which we have in fact already met in example 15 (chapter 1). Recall that, given a directed graph G, we can create a category $\mathbf{Pth}(G)$, the category of paths through G, with objects the vertices of G and morphisms the paths through G. The resulting category of paths of a graph G, $\mathbf{Pth}(G)$, gives us the *free category generated by G*, which can be thought of as the result of freely adding to a given directed graph all paths (all possible composite arrows) as well as all the identity arrows. The resulting category has the same set of objects (i.e., vertices) as the original graph, but it will in general have a larger set of morphisms, for the hom-set $\mathrm{Hom}(v, v')$ in the resulting category will consist of all the paths in the graph G from v to v', which may include arrows that were not in the original graph. Graph homomorphisms then extend into unique (covariant) functors.

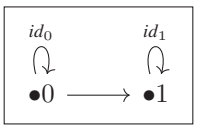
In short, this path construction gives rise to a functor $F : \mathbf{Grph} \to \mathbf{Cat}$, called the *free category functor*. $\mathbf{Pth}(G)$ is always the *largest* category generated by G. On the other hand, G also generates a *smallest* category by taking the quotient of $\mathbf{Pth}(G)$ by the relation that identifies two paths that share the same source and the same target. In this connection, any category **C** can be obtained as a quotient of the corresponding category of paths of its underlying graph, under the equivalence relation identifying two paths if and only if they have the same composite in **C**.[28]

Altogether, constructions and results established in the context of graphs can be applied to categories, once we forget about composition; and conversely, results concerning categories can be applied to graphs by simply replacing a graph by its category of paths. And these things are codified by the existence of the functors just described.

28. Getting ahead of ourselves somewhat, it is worth noting that the pair of functors $\mathbf{Cat} \underset{F}{\overset{U}{\leftrightarrows}} \mathbf{Grph}$ are related in a very special way, one that will be explained in the treatment of *adjunctions* in chapter 7. This relation is especially significant, for it forms an important adjunction that gives rise to a particular construction called a *monad* that is a starting point for the generalization to n-categories.

Example 35 We just saw that categories have underlying graphs. There is the important related notion of a *diagram* in a category **C**, a notion that in some sense captures a generalized idea of a subgraph of a given category's underlying graph.[29] A diagram is defined as a functor $F : \mathbf{J} \to \mathbf{C}$ where the domain, called the *indexing category* or *template*, is a small category. Typically, one thinks of the indexing category as a directed graph, that is, some collection of nodes and edges that serves as a template defining the shape of any realization of that template in **C** and that may also specify some commutativity conditions on the edges which are to be respected by **C**. Then a diagram can be regarded as something like an instantiation or realization in **C** of a particular template **J**. Each node in the underlying graph of the indexing category is instantiated with objects of **C**, while each edge is instantiated with a morphism of **C**. If we write the objects in the index category **J** as $i, j, \ldots$, and the values of the functor $F : \mathbf{J} \to \mathbf{C}$ in the form $F(i), F(j), \ldots$, then a diagram amounts to a family of objects $F(i)$ of **C** indexed by the nodes of **J** and a family of arrows $F(e)$ of **C** indexed by the edges of **J**. Accordingly, one sometimes speaks of a diagram F as a **J**-indexed set, or $\mathbf{J}^{op}$-parametrized set (depending on the variance of the functor). Functoriality demands that any of the composition relations (in particular, commutative diagrams) that obtain in **J** carry over (under the action of F) to the image in **C**.

We will have a lot more to say about this perspective later in this chapter. For now, let us look at a few concrete illustrations of this. First consider the category



of two objects and a single non-trivial morphism, often called **2**.[30] With such a category for our indexing category, a (set-valued) diagram yields a category that has as objects all the functions from one set to another set, and as morphisms the commutative squares between those arrow-objects. In more detail: a morphism from object $f : A \to B$ to object $g : C \to D$ will be a pair of functions $\langle h, k \rangle$ such that

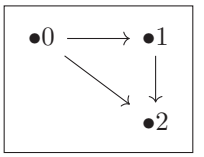
$$\begin{array}{ccc} A & \xrightarrow{h} & C \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle g} \\ B & \xrightarrow[k]{} & D \end{array}$$

commutes. Composition is component-wise, that is, $\langle j, l \rangle \circ \langle h, k \rangle = \langle j \circ h, l \circ k \rangle$, and the identity arrow for $f : A \to B$ will be the function pair $\langle \mathrm{id}_A, \mathrm{id}_B \rangle$. Does it look familiar? It should. This is just the *arrow category* introduced earlier, in definition 20!

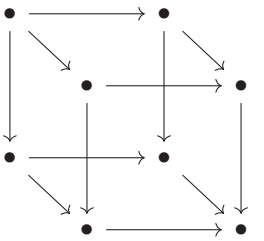
Suppose instead we take for our indexing category **3**, or **[2]**, the linear order category with length 2

29. A *subgraph* is just what it sounds like: a subset of a (directed) graph's edges (and associated vertices) that constitutes a (directed) graph. For undirected graphs, the notion is even more straightforward: graph G' is a subgraph of graph G when the vertex set V' of G' is a subset of the vertex set V of G and the edge set E' of G' is a subset of the edge set E of G. In other words, a subgraph is essentially a graph within a larger graph.

30. **2** is isomorphic to the linear order **[1]**, so one will occasionally see it go by that name.



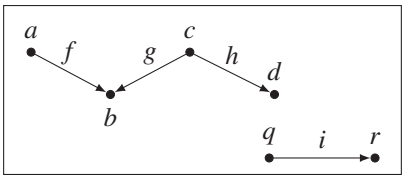
Then a diagram on this category just acts to pick out as objects commutative triangles. As a final example, taking the category $\mathbf{2} \times \mathbf{2} \times \mathbf{2}$ as our indexing category just serves to pick out as objects commutative cubes



in the target category.

As we will see, this diagram approach can be significantly generalized and can even be used to provide definitions of n-categories, specifying the data for an n-category as a diagram (presheaf) $A : \Sigma^{op} \to \mathbf{Set}$, where Σ is some category of shapes and the functor yields, for each shape, a set of "cells" of that shape.[31]

Example 36 Expanding on the previous perspective, assume we are given as indexing category $\mathbf{J} :=$



Now let the diagram $F : \mathbf{J} \to \mathbf{Set}$ be given on objects by

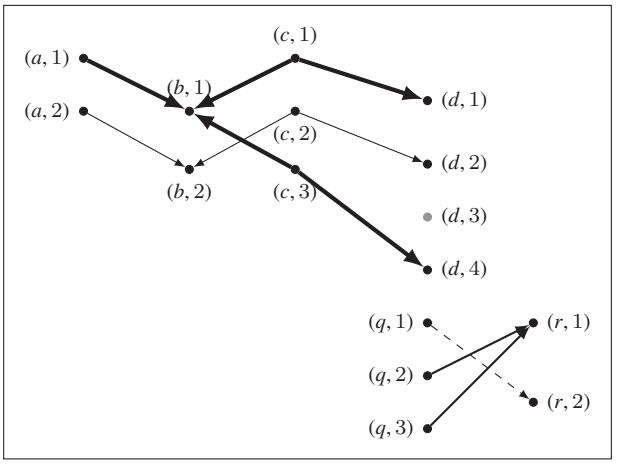
$$F(a) = \{1, 2\}, \quad F(b) = \{1, 2\}, \quad F(c) = \{1, 2, 3\},$$
$$F(d) = \{1, 2, 3, 4\}, \quad F(q) = \{1, 2, 3\}, \quad F(r) = \{1, 2\}$$

31. Treatment of n-categories is beyond the scope of this book. Instead, let us just observe how this diagram approach already suggests a more general definition of presheaves: for categories $\mathbf{C}$ and $\mathbf{J}$, a $\mathbf{C}$-presheaf on $\mathbf{J}$ can be defined as a contravariant functor from $\mathbf{J}$ to $\mathbf{C}$. Instead of taking presheaves to be functors taking values in **Set**, we can thus use other target categories, like the category of groups, rings, vector spaces, modules, and so on. While this more general definition is perfectly coherent (and is useful for achieving greater generality), presheaves are classically regarded as valued in **Set**. While this is not entirely necessary, as we just saw, there is also good reason for it. In brief, it has to do with the fact that the category of sets occupies a somewhat special place: as we will explore in chapter 6, the usual categories are *enriched* over sets, by which we effectively mean that given a pair of objects $X, Y \in \mathbf{C}$, we can form $\mathrm{Hom}_{\mathbf{C}}(X, Y)$, an object of **Set**. Moreover, another factor here has to do with the fact that only set-valued functors are *representable*. Both of these matters—that of enrichment, and representable functors—are covered in chapter 6. For now, however, it is worth noting that in most categories $\mathbf{C}$, the hom-sets $\mathrm{Hom}_{\mathbf{C}}(X, Y)$ are richer than just sets.

and on morphisms by

$$F(f) = 1 \mapsto 1, 2 \mapsto 2;$$
$$F(g) = 1 \mapsto 1, 2 \mapsto 2, 3 \mapsto 1;$$
$$F(h) = 1 \mapsto 1, 2 \mapsto 2, 3 \mapsto 4;$$
$$F(i) = 1 \mapsto 2, 2 \mapsto 1, 3 \mapsto 1.$$

This can be pictured as follows:



This realization affords us a concrete illustration of another important construction, the *category of elements*, which will be defined in chapter 3 and used to explain why there are different thicknesses of arrows in this picture.

There are also many functors that recover important established constructions that appear within the context of more specialized study of certain mathematical structures. The following is an example of that.

Example 37 *Graph coloring problems* are a commonly discussed class of problems in graph theory having to do with assigning colors to certain components of a graph subject to certain constraints. Such problems can ultimately be formulated as a problem of *vertex coloring*, where this is an assignment of "colors" (or any label) to a graph's vertices such that no two adjacent vertices share the same color, that is, so that whenever an edge connects two vertices, those vertices are assigned distinct colors. Moreover, one speaks of a coloring of a graph G by n colors as an *n-coloring* of the graph G. For an application of such a problem, the vertices of a graph might represent radio stations, where two vertices are adjacent (i.e., connected by an edge) whenever the stations are near enough to cause interference, so that a coloring would then amount to an assignment of noninterfering frequencies to the stations.

Recall the category of undirected (simple) graphs, **SmpGrph**, introduced in example 5 (chapter 1). The objects of such a category are the graphs that a graph theorist usually means by the word, and the morphisms are the usual (undirected) graph homomorphisms, that is, a graph homomorphism from a graph $G = (V, E)$ to a graph $G' = (V', E')$ is a function $f : V \to V'$ on the vertices such that if $\{x, y\}$ is an edge of G, then $\{f(x), f(y)\}$ is an

edge of G'. There is a contravariant functor from this category of undirected graphs to the category of sets—that is, a functor $nColor : \mathbf{SmpGrph}^{op} \to \mathbf{Set}$—that takes a graph to the set of n-colorings of its vertices, so that for any graph G, $nColor(G)$ will consist of the set of all n-colorings of G. Observe how an n-coloring of a graph G' together with a graph homomorphism $G \to G'$ will give rise to an n-coloring of G, illustrating the contravariance of this functor. We will return to this functor in later chapters.[32]

Here is an example of a different flavor, one that also ties together a number of constructions introduced thus far.

Example 38 There are natural language expressions that we use all the time to express that someone or something has a certain property *qua* (or *as*) one thing but not *qua* some other thing. For instance, one might say

John is fair *qua* businessman, but not *qua* politician,

or

Maria is inspirational *qua* teacher, but not *qua* basketball player.

We often make use of judgments involving the logic of *qua*. Suppose someone asks you whether your friend Abe is honest. You might answer, "Well, it depends: in some respects/aspects, Abe is honest; in other respects/aspects, not so much." Perhaps you know him to be an honest friend, and have no reason to suspect his dishonesty as a businessman, but you have serious doubts about his honesty as a card player. In attempting to form a judgment about your friend's honesty, you can argue about which of the aspects are relevant, or most relevant, and also about how Abe's behavior, under a particular aspect, should be interpreted (as honest or dishonest). However, in general, this sort of answer—"In some aspects, yes; in others, not so much"—and the ensuing discussion or debates make sense. Once agreement about these matters (which aspects are relevant, etc.) has been achieved, we may use these assessments to arrive at a global judgment about Abe's honesty.

We might conceptualize this situation category theoretically, using a particular *category of aspects* or *qua* category.[33] In this setting, we will be able to model things like the honesty of Abe under aspect A, and moreover model the assembly of global judgments of the type "Abe is honest," "Abe is not honest," "Abe is dishonest," and so on, from this data of judgments about Abe's honesty *qua* the various relevant aspects.

Let us first define something we could call the *nominal category* **CN**.

Definition 39 The *nominal category*, **CN**, has for

32. Looking ahead, a sheaf will be defined as a particular presheaf satisfying certain properties with respect to "covers" of the objects of the domain category (which we will meet more formally in chapter 4, dedicated to topology). Anticipating this, the *nColor* presheaf functor defined on a related subgraph category will turn out to give us a sheaf, since if some subgraphs G_i cover G, and if $\{c_i \in nColor(G_i) \mid i \in I\}$ is a family of colorings such that the colorings agree on intersections among the G_i, then the c_i's induce a unique coloring of the entire graph G, which essentially is what it means to have a sheaf in this instance.

33. The idea for this, and the key definitions provided below (as well as the example, with mostly trivial modifications), comes from Reyes et al. (1999).

- objects: CNs (count nouns) relevant to discussion, for example, "a student," "a coworker," "a husband," "a parent," "a family man," "a student, a coworker, and a family man," written as

$$\boxed{\text{a s}}, \boxed{\text{a c}}, \boxed{\text{a h}}, \boxed{\text{a p}}, \boxed{\text{a f}}, \boxed{\text{a scf}}.$$

- morphisms: "identification" postulates of the form (copula connecting nouns)

$$\boxed{\text{a f}} \xrightarrow{is} \boxed{\text{a p}}, \quad \boxed{\text{a scf}} \xrightarrow{is} \boxed{\text{a f}},$$

  where these are meant to capture the identifications frequently used in natural languages, such as "a family man is a parent," "a student, a coworker, and a family man is a family man," "a dog is an animal," and so on.

  Identity morphisms are those particular axiomatic identifications of the form

$$\boxed{\text{a f}} \xrightarrow{is} \boxed{\text{a f}}.$$

Composition is given by stringing together identifications in the obvious way, that is, whenever we have two arrows, the codomain of one as the domain of the other, we complete the graph by adding an arrow that is the composition of the two arrows, and where this corresponds to the common rule of inference in natural languages from things like "a human is a primate" and "a primate is a mammal" to "a human is a mammal."

Arrows of this category can be thought of as supplying a *system of identifications*, where this replaces a notion of *equality* between kinds (since equality is a relation that might be argued to obtain only between the members of a given kind). This category is assumed to be posetal, where this means there is at most one arrow between two objects.

We know from definition 20 (chapter 1) that for a category $\mathbf{C}$, we can form the *arrow category* of $\mathbf{C}$, denoted $\mathbf{C}^{\rightarrow}$. Moreover, it turns out that we can *identify* any object A of $\mathbf{C}$ with the object $A \xrightarrow{\text{id}_A} A$ in $\mathbf{C}^{\rightarrow}$. Using this arrow category construction, we can define our main category of interest.

Definition 40 The *qua category*[34] of **CN**, **Qua(CN)** (or just **Qua**), is defined as $(\mathbf{CN}^{\rightarrow})^{op}$.

Because of how **CN** was defined, we will have at most one morphism from an object A to an object B. When such a morphism exists, we can see $A \rightarrow B$ as A *qua* B, for example,

$$\boxed{\text{a f}} \longrightarrow \boxed{\text{a p}}$$

will be to look at "a family man *qua* a parent." Identifying $A \xrightarrow{\text{id}_A} A$ with A, we are thus identifying the count noun A with its global aspect A *qua* A, that is, *as itself*.

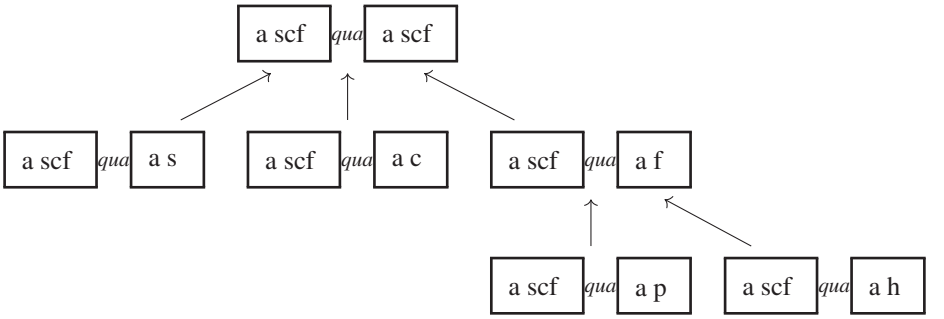
In modeling consideration of the various aspects of people, we will be interested in a particular subcategory of $(\mathbf{CN}^{\rightarrow})^{op}$, namely the co-slice category of objects under the global object. For instance, the category

34. This is called the *aspectual category* in Reyes et al. (1999).

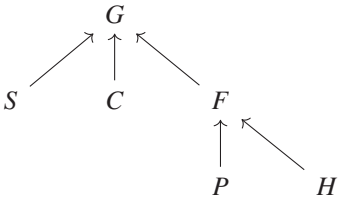
$$\boxed{\text{a scf}} \downarrow \mathbf{CN}$$

of objects under "a student, a coworker, a family man"—where this is identified with the global aspect $\boxed{\text{a scf}} \xrightarrow{qua} \boxed{\text{a scf}}$—forms a subcategory $\mathbf{A}$ of the *qua* category $(\mathbf{CN}^{\rightarrow})^{op}$.

For concreteness, suppose we have



where the identity maps and those induced by composition are left implicit. In this way, such an $\mathbf{A}$ will thus serve as a way of representing those aspects of a person, including for instance Abe, relevant to whether or not a certain predicable holds of them, like "honesty." This is something that will be evaluated "*qua* scf" (in terms of all their "hats," via the global aspect), "*qua* student," "*qua* family man," and so on, where this latter has two subaspects: "*qua* parent" and "*qua* husband." Abbreviating these aspects, then, we could have displayed $\mathbf{A}$ as



Before defining the relevant functor towards which we have been building, let us also record the following definition of a concept we will use here and throughout this book.

Definition 41 A functor $F : \mathbf{C}^{op} \to \mathbf{Set}$ is a *subfunctor* (*subpresheaf*) of a functor $G : \mathbf{C}^{op} \to \mathbf{Set}$ if

- for all $c \in \mathbf{C}$, $F(c) \subseteq G(c)$; and
- for all morphisms $f : c \to c'$ of $\mathbf{C}$, $F(f) : F(c') \to F(c)$ is the restriction of $G(f)$ to $F(c)$.

In other words, put more abstractly, for any $f : c \to c'$ in $\mathbf{C}$ there exists a commutative diagram

$$\begin{array}{ccc} F(c') & \xrightarrow{F(f)} & F(c) \\ \downarrow & & \downarrow \\ G(c') & \xrightarrow[G(f)]{} & G(c). \end{array}$$

For reasons that will be better appreciated after we have introduced a few other notions, we sometimes write $F \hookrightarrow G$ to indicate that F is a subfunctor of G.

Now, given a *qua* category and $\mathscr{P}$ a set of predicables that are applicable to the count nouns of **CN**—where predicables may be thought of for now as just involving grammatical expressions consisting of adjectives, verbs, or adjectival and verb phrases, including expressions such as "mortal" or "honest," and "to be a person," where this is derived from or sorted by a count noun—we define an *interpretation* of (**Qua**, $\mathscr{P}$) as a functor

$$X : \mathbf{Qua}^{op} \to \mathbf{Set}$$

together with a set

$$\{X_\phi \hookrightarrow X \mid \phi \in \mathscr{P}\}$$

of subfunctors of X that satisfy the following conditions:

1. $X\left(\boxed{\text{A}}\ qua\ \boxed{\text{B}}\right) = X\left(\boxed{\text{A}}\ qua\ \boxed{\text{A}}\right)$; and
2. $X_\phi\left(\boxed{\text{A}}\ qua\ \boxed{\text{B}}\right) = X_\phi\left(\boxed{\text{B}}\ qua\ \boxed{\text{B}}\right) \circ X\left(\boxed{\text{B}}\ qua\ \boxed{\text{B}} \to \boxed{\text{A}}\ qua\ \boxed{\text{A}}\right)$

Notice that since $\mathbf{Qua} = (\mathbf{CN}^{\to})^{op}$, then its dual, $\mathbf{Qua}^{op}$, is just $\mathbf{CN}^{\to}$, making X equivalently expressible as a functor

$$X : \mathbf{CN}^{\to} \to \mathbf{Set}.$$

That we can compare the interpretations of count nouns in fact forms the basis of the possibility of comparing the corresponding interpretations of predicables that are functorial. Using this notion of interpretation, and given a subcategory of **Qua** such as **A** above, we can restrict the interpretation to the subcategory. Fundamentally, the functoriality here can be understood as saying, for instance, if Abe is honest *qua* family man, then he is honest *qua* parent. Moreover, $\boxed{\text{a p}}$ will be interpreted as a set of parents, $\boxed{\text{a scf}}$ as a set of students who are also coworkers and family men. We will return to this example later in the chapter, and again in later chapters.

The next example is very important for the general theory that will be developed in later chapters, especially chapter 6.

Example 42 Let **C** be a category, and fix an object a of **C**. Then we can form the (covariant) *hom-functor* $\mathrm{Hom}_{\mathbf{C}}(a, -) : \mathbf{C} \to \mathbf{Set}$, which takes each object b of **C** to the set $\mathrm{Hom}_{\mathbf{C}}(a, b)$ of **C**-morphisms from a to b, and takes each **C**-morphism $f : b \to c$ to the following map between hom-sets:

$$\mathrm{Hom}_{\mathbf{C}}(a, f) : \mathrm{Hom}_{\mathbf{C}}(a, b) \to \mathrm{Hom}_{\mathbf{C}}(a, c), \tag{2.1}$$

which outputs $f \circ g : a \to c$ for input $g : a \to b$. In other words, the action on morphisms is given by *postcomposition*. This hom-functor will be defined for any object whenever the hom-sets of **C** are *small*, that is, whenever **C** is *locally small*. Intuitively, the set $\mathrm{Hom}_{\mathbf{C}}(a, b)$ can be thought of as the set of ways to pass from a to b within **C**, or the set of ways a "sees" b within the context or framework of **C**. Then, refraining from filling in the object b, it should be obvious how $\mathrm{Hom}(a, -)$ can be thought of as representing in a rather general fashion "where and how a goes elsewhere" or "how a sees its world." Given an object $a \in \mathbf{C}$, we say that the covariant functor $\mathrm{Hom}(a, -)$ is *represented by* a; for reasons we will

explore in chapter 6, this functor is also denoted Y^a (or sometimes h^a). It will turn out to be an important observation that instead of restricting ourselves to the hom-functor on a given a, we can assign to *each* object $c \in \mathbf{C}$ its hom-functor $\mathrm{Hom}(c, –)$, and then collect all these together.

We can also form the *contravariant hom-functor* $\mathrm{Hom}_{\mathbf{C}}(–, a) : \mathbf{C}^{op} \to \mathbf{Set}$, for a fixed object a of $\mathbf{C}$, which takes each object b of $\mathbf{C}$ to the set $\mathrm{Hom}_{\mathbf{C}}(b, a)$ of $\mathbf{C}$-arrows from b to a, and takes each $\mathbf{C}$-arrow $f : b \to c$ to $\mathrm{Hom}_{\mathbf{C}}(f, a) : \mathrm{Hom}_{\mathbf{C}}(c, a) \to \mathrm{Hom}_{\mathbf{C}}(b, a)$, that is, outputting $g \circ f : b \to a$ for input $g : c \to a$, acting by *precomposition*. This functor can be thought of as representing "how a is seen by its world." Given an object $a \in \mathbf{C}$, we say that the contravariant functor $Y_a := \mathrm{Hom}(–, a)$ is *represented by* a. As above, instead of restricting ourselves to the hom-functor on a, we can ultimately let this functor vary over all the objects of $\mathbf{C}$.

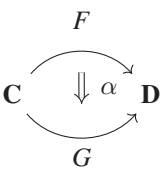
Throughout this book, we will see many more examples of functors. For now, though, we can continue to develop the main concepts, ascending once more in generality, now regarding functors as objects in a category, with morphisms given by certain *transformations* between the functors.

2.2 Natural Transformations

Functors are important for many reasons. In particular, as we will explore in chapter 3 and beyond, universal properties are given in terms of functors. Moreover, it is possible to use two functors to do a variety of important things, such as produce a new category from old categories. However, perhaps most important for our present purposes is the fact that functors can be composed, and there is a nice notion of comparing functors.

There may exist a variety of ways of embedding or modeling or instantiating one category within another, that is, there may exist many functors from one category to another. Sometimes these will be equivalent, but sometimes not. Moreover, the same blueprint may be realized in different ways, that is, there can be different functors that act the same way on objects. *Natural transformations* enable us to compare these realizations. If functors allow us to systematically import or transform objects from one category into another and thus translate between different categories, natural transformations allow us to compare the different translations in a controlled manner.

Definition 43 Given categories $\mathbf{C}$ and $\mathbf{D}$ and functors $F, G : \mathbf{C} \to \mathbf{D}$, a *natural transformation* $\alpha : F \Rightarrow G$, depicted in terms of its boundary data by the diagram



consists of the following:

- for each object $c \in \mathbf{C}$, an arrow $\alpha_c : F(c) \to G(c)$ in $\mathbf{D}$, called the c-component of α, the collection of which (for all objects in $\mathbf{C}$) defines the *components* of the natural transformation; and

- for each morphism $f : c \to c'$ in $\mathbf{C}$, the following square of morphisms (depicted on the right), called the *naturality square* for f, must commute in $\mathbf{D}$:

$$\begin{array}{ccc} c & & F(c) \xrightarrow{\alpha_c} G(c) \\ {\scriptstyle f}\downarrow & & {\scriptstyle F(f)}\downarrow \quad\quad \downarrow{\scriptstyle G(f)} \\ c' & & F(c') \xrightarrow[\alpha_{c'}]{} G(c') \end{array}$$

The collection of natural transformations from F to G is sometimes denoted by $Nat(F, G)$.

Let's step back and unpack this a little. In the general context of a category, we think of each arrow of a category as a way of comparing two objects. As we climb the ladder of abstraction and introduce functors, we think of functors as distinct ways of comparing two categories. As we suggested earlier—though we will refine this in the coming sections—one way of viewing a functor from $\mathbf{C}$ to $\mathbf{D}$ is as supplying some sort of "picture" of $\mathbf{C}$ within the world of $\mathbf{D}$. Taking the next step on the ladder of abstraction, a natural transformation effectively compares two functors. Following Goldblatt (2006), one way of building intuition of the idea of a transformation from F to G is to imagine that, within $\mathbf{D}$, you have to superimpose or slide the picture of $\mathbf{C}$ given by F onto the picture of $\mathbf{C}$ given by G, where you make use of the structure of $\mathbf{D}$ in carrying out this translation. Minimally, to compare a picture given by F to one given by G, we ought to assign to each object c of $\mathbf{C}$ an arrow in $\mathbf{D}$ going from the image $F(c)$ of c under F to the image $G(c)$ of c under G. For a given c, we could name this "component" of the transformation by $\alpha_c : F(c) \to G(c)$, to track which object we are attending to. Collecting these together, for all the objects of $\mathbf{C}$, gives us the *components* of our transformation α. But there surely needs to be something else to this process, beyond just an indexed family of arrows. As functors, we also have information about how F and G act on the morphisms of $\mathbf{C}$, and we will need to use this to complete the pictures supplied by F and G. Incorporating this information, for each morphism $f : c \to c'$ of $\mathbf{C}$, we automatically end up with diagrams like the naturality square depicted above. As a final step, asking that such diagrams *commute* (in $\mathbf{D}$) is exactly the "natural" thing to do, if we want to continue with the same sort of notion of structure-preservation that we have been developing on lower rungs of the ladder of abstraction.

We will see a variety of examples of natural transformations in the coming sections. For now, let us make a few other useful observations and definitions. First of all, observe that the same notion works for functors of the other common variance as well. If F and G are both contravariant functors, then the same definition applies, except the vertical arrows are reversed in each of the naturality squares.

Now, if natural transformations are ways of comparing functors, at the extreme we ought to be able to use this notion to develop a refined idea of when two functors are fundamentally the *same* functor. The following notion helps us do just that—and, as we will see, a lot more.

Definition 44 A *natural isomorphism* is a natural transformation $\alpha : F \Rightarrow G$ for which every one of the components $\alpha_c : F(c) \to G(c)$ is an isomorphism (in the target category). In other words, each α_c has an inverse $\alpha_c^{-1} : G(c) \to F(c)$, where these inverses form the components of a natural transformation α^{-1} from G to F.

When α is such a natural isomorphism, we denote this by writing $\alpha : F \cong G$.

Let's push this further. We already know that we can compose functors, and we have also met the identity functor on a category. Combining these ingredients together with the notion of a natural isomorphism provides us with a more refined notion of when two categories are *equivalent*.

Definition 45 An *equivalence of categories* consists of a pair of functors $F : \mathbf{C} \to \mathbf{D}$, $G : \mathbf{D} \to \mathbf{C}$ together with the natural isomorphisms $\eta : \mathrm{id}_{\mathbf{C}} \cong G \circ F$ and $\epsilon : F \circ G \cong \mathrm{id}_{\mathbf{D}}$. Another way of saying this is that the functors are inverse to each other "up to natural isomorphism of functors." The categories $\mathbf{C}$ and $\mathbf{D}$ are then said to be *equivalent* if there exists an equivalence of categories between them, and in such cases we write $\mathbf{C} \simeq \mathbf{D}$.

Remark 46 Recall the category-theoretic notion of an isomorphism from definition 10 (chapter 1). By running that definition on the category of categories (say, **Cat**), we get the concept of an *isomorphism of categories*, where this is a pair of functors $F : \mathbf{C} \to \mathbf{D}$, $G : \mathbf{D} \to \mathbf{C}$ such that their composites *equal* the respective identity functors, that is, $G \circ F = \mathrm{id}_{\mathbf{C}}$ and $F \circ G = \mathrm{id}_{\mathbf{D}}$. This will moreover induce a bijection between the objects of each category as well as between their morphisms.

The overly restrictive notion of an isomorphism of categories should be compared to the more relaxed (and ultimately superior) criterion of identity or "sameness" of two categories we find in the notion of an equivalence of categories. Even rather simple categories can be found to illustrate the point that, as far as categories are concerned, the notion of isomorphism of categories is not the right thing to consider—many categories may indeed appear "the same" in all important respects, yet they won't be isomorphic (often for reasons that appear to be trivial or irrelevant). Grothendieck, who first introduced the notion of equivalence of categories, accordingly stressed the differences between the two notions, and how the notion of an isomorphism of categories is, in many cases of practical interest, not at all a useful notion. He was in part motivated to this because he was dealing mainly with *functor categories* (introduced below), and among such categories it is common to find categories that appear to be the same in every important categorical respect, yet are not isomorphic.

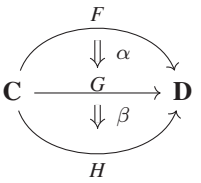
If we know from constructions like the arrow category that it is sensible to take things like functors as objects, and if natural transformations can be regarded as morphisms between functors, this data should let us form a new category! Indeed, this gives us the following important category.

Definition 47 For any fixed pair of categories $\mathbf{C}$ and $\mathbf{D}$, we can form the *functor category*, denoted by $\mathbf{D}^{\mathbf{C}}$ (or, less commonly, by $Fun(\mathbf{C}, \mathbf{D})$), where this has for

- objects: all the functors from $\mathbf{C}$ to $\mathbf{D}$;
- morphisms: all the natural transformations between such functors.

Of course, to exhibit this as a category, we need identities and composites, and to show that the relevant axioms are satisfied. For a functor $F : \mathbf{C} \to \mathbf{D}$, we can assign its identity natural transformation $\mathrm{id}_F : F \Rightarrow F$ to be the natural transformation whose components are each identities. For composition of morphisms in $\mathbf{D}^{\mathbf{C}}$, we use the following notion:

Definition 48 Let $\alpha : F \Rightarrow G$ and $\beta : G \Rightarrow H$ be natural transformations between the parallel functors F, G, and H, from $\mathbf{C}$ to $\mathbf{D}$, as depicted by the diagram:



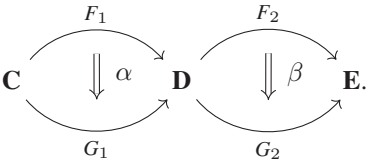
There is a natural transformation $\beta \circ \alpha : F \Rightarrow H$, where this is defined on components: components $(\beta \circ \alpha)_c := \beta_c \circ \alpha_c$ are taken to be the composites of the components of α and β.

You might think of this, at the level of the components and their naturality squares, as a matter of pasting the naturality squares together into rectangles.

For reasons discussed briefly in the remark to follow, this sort of composition is usually referred to as *vertical composition*.

One can verify that this composition operation satisfies the two axioms of a category, by checking it on components and using the fact that composition automatically satisfies these properties in **D** (since it is a category). We thus have a category, and a rather important one at that, as we will see.

Remark 49 After you have sat with categories and natural transformations, it may occur to you that the notion of natural transformations seems to support another kind of composition, distinct from that given in definition 48. Indeed, imagine we move "horizontally" via functors from one category **C** to **D** and then again via functors from **D** to **E**, as depicted in the following diagram:



Horizontal composition uses the symbol $\diamond$ and gives $\beta \diamond \alpha : F_2 \circ F_1 \Rightarrow G_2 \circ G_1$, whose component at c of **C** is defined as the composite of the following commutative square:

$$\begin{array}{ccc} F_2F_1(c) & \xrightarrow{\beta_{F_1c}} & G_2F_1(c) \\ {\scriptstyle F_2(\alpha_c)}\downarrow & \overset{(\beta\diamond\alpha)_c}{\searrow} & \downarrow{\scriptstyle G_2(\alpha_c)} \\ F_2G_1(c) & \xrightarrow[\beta_{G_1c}]{} & G_2G_1(c). \end{array}$$

While we will not be needing this notion of horizontal composition in this text, it is used to further generalize the definition of a category, defining a 2-*category*, a starting point for higher and higher climbs up the ladder of abstraction.

For our purposes, perhaps the most important thing to note here is that since presheaves are just (contravariant) functors, so that we are given a notion of a morphism of presheaves from F and G as just a natural transformation $\alpha : F \Rightarrow G$, we can form the presheaf functor category.

Definition 50 The *presheaf category*, denoted $\mathbf{Set}^{\mathbf{C}^{op}}$ or $\mathbf{PreSh}(\mathbf{C})$, is the (contravariant) functor category having for objects all functors $F : \mathbf{C}^{op} \to \mathbf{Set}$, and for morphisms $F \to G$

all natural transformations $\theta : F \Rightarrow G$ between such functors. As a natural transformation, such a θ will assign to each object c of $\mathbf{C}$ a function $\theta_c : F(c) \to G(c)$, and do so in such as way as to make all diagrams

$$\begin{array}{ccc} d & & F(d) \xrightarrow{\theta_d} G(d) \\ {\scriptstyle f}\uparrow & \qquad & {\scriptstyle F(f)}\downarrow \qquad\quad \downarrow{\scriptstyle G(f)} \\ c & & F(c) \xrightarrow[\theta_c]{} G(c) \end{array}$$

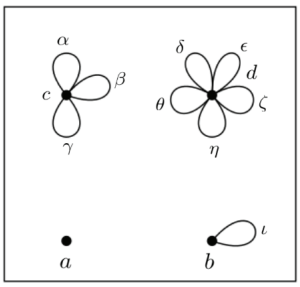
commute for each $f : c \to d$ in $\mathbf{C}$.

Example 51 For $\mathbf{J}$, an arbitrary category viewed as a template or indexing category for $\mathbf{C}$, we can define another sort of functor category by looking at the category $\mathbf{C}^{\mathbf{J}}$ of $\mathbf{J}$-diagrams in $\mathbf{C}$, where each object is a functor $F : \mathbf{J} \to \mathbf{C}$, and for two such objects F, G, a morphism of $\mathbf{C}^{\mathbf{J}}$ from F to G is a natural transformation between the functors.

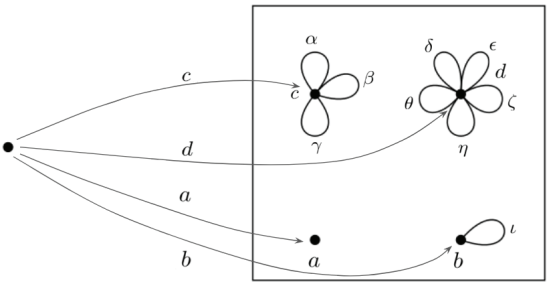
Especially on account of the important place that presheaf categories will occupy in our story, we will see many more examples of natural transformations in action, in a variety of contexts. For now, let us delve a little deeper into what it is to be a presheaf.

2.3 Seeing Structures as Presheaves

When we work with a mathematical structure, it is common to try to approach it in terms of its elements. In general, it is very natural to want to break things down by decomposing more complicated structures into their components—and elements, like points, are one sort of component we seem especially ready to recognize as such. But in certain settings, one needs to consider *figures of a more general shape* than points. Points, after all, might be regarded as just a particularly simple kind of "shape." For instance, suppose you are presented with the structure X:



This X depicts what is called a *bouquet*. Figures in the bouquet X with the shape "point" can be regarded as maps $\bullet \to X$, each of which map names a point in X:



We use the generic shape • to locate and name all the distinct point-like figures of X, that is, via a particular map $\bullet \xrightarrow{a} X$ we name with a one of the various •-like components of X. Altogether, the data of our point-like figures in X really just amounts to a set

$$X(\bullet) = \{a, b, c, d\},$$

which you might read as saying "X realizes a, b, c, and d as its figures of shape •."

But you could not hope to understand all that this structure X is just by considering the point-like figures! After all, points are not the only sort of figural component in X. Thus, there are in fact many distinct bouquets that may even have the same set of points (yet will look rather different!). So we also need a way of picking out those figures in X whose shape is that of a "loop." Similar to our point-like figures in X, we can pick out and name loops via maps from the generic shape into X:

In other words, the data here is captured by the set

$$X(\text{  }) = \{\alpha, \beta, \gamma, \delta, \epsilon, \zeta, \eta, \theta, \iota\},$$

which you might read as saying "X realizes $\alpha, \beta, \gamma, \ldots$ as its figures of shape  ."

But do we then have enough information to reconstruct X? Well, many bouquets may have the same set of loops, but the home point at which they live may be different. You would not regard these as the same thing. To fully capture X, then, we also need a way of extracting the data of where each loop-shaped figure lives, that is, *how the loop-shaped figures relate to the point-shaped figures*. Corresponding to the inclusion of the generic shape "point" in the generic shape "loop"

$$\bullet \xrightarrow{i} \text{  },$$

we should then have a map

$$X(\text{loop}) \xrightarrow{X(i)} X(\bullet)$$
$$l \mapsto p,$$

taking a loop l to the point p at which it is stationed, and so informing us about which loops get stationed at which points. For instance, this will tell us that

$$X(i)(\alpha) = c,$$

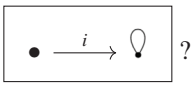
or "the loop-shaped component named α lives at the point-shaped component named c." The equations telling us which point each of the loops are assigned supply us with what are called the *incidence relations*.

With all this information—the set $X(\bullet)$ and $X(\text{loop})$, together with a map describing how the loop elements in the latter set are sent to the points in the former set—it would seem that we will be able to recover the whole of the information of the bouquet X itself.

But described in this way—and this is the point!—what else have we been saying but that X itself can just be regarded as a presheaf

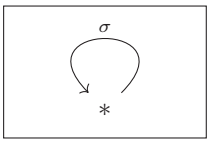
$$X : \mathbb{B}^{op} \to \mathbf{Set},$$

where the indexing category $\mathbb{B}$ is

$\mathbb{B} :=$  ?

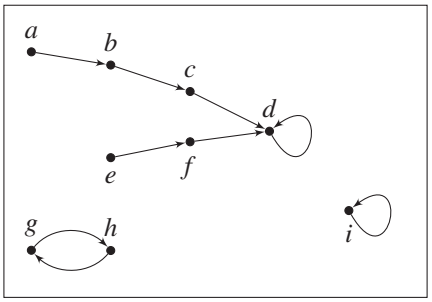
In general, what we are starting to develop is the extension of an idea whereby we consider maps with codomain X as *figures* in X, where the domain of a figure is regarded as its *shape* or *type*, and where the incidence relations giving such an X its structure, and describing how the figures of distinct shapes relate, are then specified in terms of a map between the figures. In this same manner, if our domain or source category is regarded as consisting of some generic "shapes," related in some particular way (such as points included in pointed loops), then the result of realizing or figuring those shapes, via the image of a specified presheaf X, can be imagined as a "container" holding onto the various concrete realizations or instantiations of the different generic shapes supplied by the source category, where the natural relation that obtains between the underlying shapes is respected by the associated figures realized in the target category.[35]

In a similar way, suppose we instead take for our indexing category of shapes the single object $*$ and all the morphisms generated by iterations of σ, that is, the free monoid on one generator (σ),

$\mathbb{E} :=$ 

35. This *figure* perspective, applied to presheaves, is advocated by Lawvere, especially in Lawvere and Schanuel (2009). This approach is taken even further in the delightful Reyes, Reyes, and Zolfaghari (2008), which much of the remainder of this chapter is inspired by and which is highly recommended to the reader who finds the topics and discussions of this chapter of particular interest.

Then presheaves X on this arise as dynamical systems (evolutive sets) or automata, where X supplies the set of possible states, and the given endomap σ gives rise to the evolution of states (think the change in internal state that results after the passage of one unit of time, or as a result of pressing the "button" σ on the outside of a machine). In other words, if X is a presheaf on $\mathbb{E}$, we think of its image as a container containing a set of figures—shaped in the form of dots, corresponding to various instantiations of the object $*$ of $\mathbb{C}$, and in the form of arrows between certain of those dots, corresponding to the endomap σ of $\mathbb{E}$—with a process taking each element to a next stage or next element. In this way, we might end up with an X such as

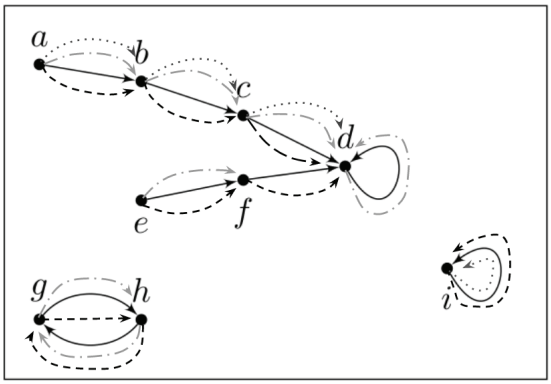


Altogether, as we will see, $\mathbf{Set}^{\mathbb{E}^{op}}$, the category of presheaves on $\mathbb{E}$, with objects like X, will itself be none other than the category of evolutive sets or dynamical systems.

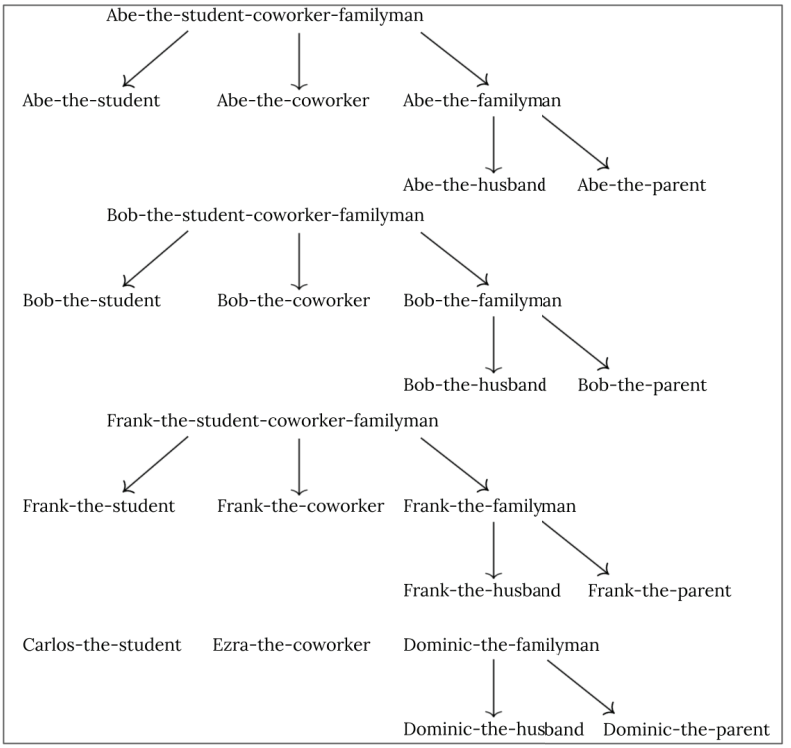
Similarly, if we instead took as our indexing shape category the category of n-evolving sets, that is, $\mathbb{E}_n$, freely generated by n nonidentity morphisms:

$$\mathbb{E}_n := \quad \sigma_1 \circlearrowright \; * \; \circlearrowright \cdots \quad \sigma_2 \circlearrowright \; \circlearrowright \sigma_n$$

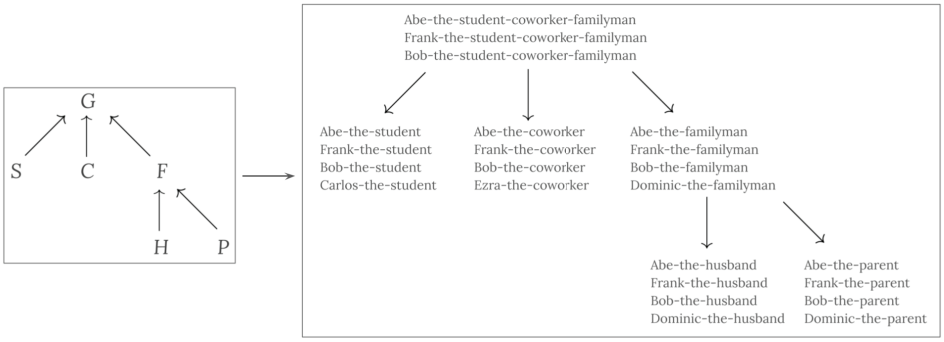
then the container of $\mathbb{E}_n$-shaped figures would have figures similar to the picture of a presheaf on $\mathbb{E}$, except with (up to) n different processes carrying one $*$-figure to the next, for example,



A similar approach can also be taken when using more distinct indexing categories. For instance, recall the particular subcategory **A** of **Qua** from example 38. Applying an interpretation functor $X : \mathbf{A}^{op} \to \mathbf{Set}$ then results in a container of **A**-shaped figures, for example,



As before, we could regard $X(S)$, for instance, as picking out or naming those who are "shaped like," or of type, *student*. If we organize things a bit, grouping together those people that are picked out as conforming to the same shape (in this case, their role), then what is fundamentally going on here can be displayed more sensibly as



An interpretation X of such an $\mathbf{A}$ is really just an object of the presheaf category $\mathbf{Set}^{\mathbf{A}^{op}}$ together with a set of subfunctors corresponding to the predicables of $\mathscr{P}$. Morphisms of this category are just natural transformations from an interpretation X to an interpretation Y such that the restrictions to $X_\phi \hookrightarrow X$ can be factored through $Y_\phi \hookrightarrow Y$.

As in the previous examples, the idea here is that the domain category supplies the generic shape according to which figures or instantiations of such shape are organized, and where the overall realization of such figures as figures of such a shape, constrained to relate in a way that respects the underlying relations of the shapes of the domain, is accomplished via the functor.

Generalizing from such examples, the story we are starting to tell is that for a presheaf $P : \mathbf{C}^{op} \to \mathbf{Set}$, for certain $\mathbf{C}$ it is often natural to think of the result of applying P as leaving us with some sort of container of $\mathbf{C}$-shaped figures, where the various objects c of $\mathbf{C}$ are thought of as supplying the *generic figures*, or templates of a figure, that are then *instantiated* or *figured* or *realized* in **Set**; for example, $P(c)$ is some particular *set* of instantiations or figures of c-shape. A functor is fundamentally a way of turning objects and structured connections in one world $\mathbf{C}$ into objects and maps in another world $\mathbf{D}$, and doing it in such a way that certain equations are satisfied (where these equations code for the preservation of structure, or compatibility of the transformation with the composition of maps in $\mathbf{D}$). Another way of thinking of a presheaf is thus as a *realization* of $\mathbf{C}$ in **Set**.

So far, we have mostly dwelt on how the presheaf operates *on objects*. Obviously, as a functor, we must also consider the action specified by the (contravariant) functor, that is, how it acts *on morphisms*. The basic idea will be that, following the figures of generic shape c interpretation, a morphism in $\mathbf{C}$ from one object to another will give rise to a *change of figures*, where this means, more precisely, that if we have a figure x of shape c (i.e., $x \in P(c)$) and a figure y of shape c' (i.e., $y \in P(c')$), then asking about the effect of changes of figures amounts, at the level of the presheaf, to asking to what extent the figures are incident or overlap (and what this overlap structure looks like) or otherwise relate. But the same idea would apply to less geometrical settings, for instance using an indexing category that is more time-like, and adopting the interpretation of $P(c)$ as a set existing *at*

stage c. Then the morphisms of **C**, when acted on by the presheaf, would model *varying the stage*, or transitions between stages—so that, overall, the functor can be interpreted as supplying a picture of a set *varying through time*. In the next section, we start to look more closely at interpretations such as the incidence relation, as well as some others such as the variable-set interpretation—as always, via examples. Such examples will enable us to start to think more systematically about presheaves and their action. Moreover, in building on some of the examples thus far, looking closer at the resulting presheaf category, we will see some further examples of natural transformations.

2.4 The Presheaf Action

It is not uncommon to see in the literature on presheaves references to *right* **C**-sets (which are the same as *left* $\mathbf{C}^{op}$-sets). Similarly, one will sometimes hear talk of a presheaf's *right action*. We will think of there being *four characteristic kinds of cohesivity or variability* presented by presheaf categories in accordance with four main ways the right action of the presheaves in question can be found to operate. But before discussing these interpretations of a presheaf (illustrating them each through select examples), it may be useful to further explain the reference to the presheaf action as a *right action*, in case it is not already clear why one can see this being referred to as an *action* (and, moreover, why the action is *right*).

2.4.1 Right Action Terminology

A presheaf is ultimately just a *functor* (one with a particular variance). At least with the usual set-valued presheaves, in applying a given functor P to each of the objects of the domain category **C**, we just get a bunch of *sets*, $P(c), P(c')$, and so on, indexed by the objects of **C**. The functoriality of the given presheaf P, then, just means that for every map $f : c' \to c$ in **C**, we will have a *function*—since we are landing in **Set**, after all!—$P(f) : P(c) \to P(c')$ going the other way. So we just have a function that takes the elements x of the *set* $P(c)$, that is, the set P *seen at stage* c (or "seen in the shape of" c), to elements of the *set* $P(c')$, that is, P *seen at stage* c' (or "seen in the shape of" c'). In other words, for each element x of $P(c)$ and each map $c' \xrightarrow{f} c$ of **C**, there is an associated element xf of $P(c')$.[36]

Now, the *contravariance* of the functor of course means that the functor applied to a composite $f \circ g$, where $c'' \xrightarrow{g} c' \xrightarrow{f} c$, should be the same as the functor first acting on f then acting on g. In other words, in terms of the elements $x \in P(c)$, f, g, and xf as above, whenever $c'' \xrightarrow{g} c' \xrightarrow{f} c$, we must have $x(f \circ g) = (xf)g$ in $P(c'')$. Moreover, functors must respect identities. But all this data essentially means that we are dealing with what, in other settings, one would call a *right action* of **C** on the underlying set (formed by the presheaf P), and where this right action expresses the incidence relations or transitions among the various figures x, x', and so on.

In those other contexts, if we have some mapping $X \times A \to X$, it is common to refer to such a map as a *right action* of A on X. Usually A is some monoid or group (and X is a set). The basic idea is that A is thought of as furnishing a set of "buttons" that control the states

36. The reason for writing the element xf this way is explained below.

of X, while the given action $X \times A \xrightarrow{\alpha} X$ is regarded as supplying us with the data of a state-machine or automaton. Considering a particular "button" a then gives rise to an endomap of X, specifically $\alpha(-, a)$, where this means that for each element x of X, its image $\alpha(x, a)$ under the action map α is just a new element of X. "Pressing" a once takes a particular state x into the state $\alpha(x, a)$; pressing it twice takes x to $\alpha(\alpha(x, a), a)$; and so on. Of course, we can also press a different button (i.e., take a different object a' of A). Combining things, we can press one button and then another. This will mean: suppose we are in state x and button a is pressed and then button a', the resulting state will be $\alpha(\alpha(x, a), a')$. As is common, we can choose, notationally, to represent the result of the action $\alpha(x, a) = x \cdot a$, which you might read as "having pressed a on state x."

In a similar fashion, with presheaves we speak of a *right action* of $\mathbf{C}$ on a set P that is partitioned into sorts coming from the objects of $\mathbf{C}$ (i.e., parameterized by the objects of $\mathbf{C}$). Being a "right action" here means that whenever we have an arrow $f : c' \to c$ in $\mathbf{C}$ and an element $x \in P(c)$—that is, an element of the set P of sort c—then xf yields an element of P of sort c' subject to the conditions

$$x \, \mathrm{id}_c = x;$$

$$x(f \circ g) = (xf)g \text{ whenever } c'' \xrightarrow{g} c' \xrightarrow{f} c \in \mathbf{C}.$$

We may write the action in the form of concatenation; that is, xf is short for $x \cdot f$ where the action $\alpha : \mathbf{Set} \times \mathbf{C} \to \mathbf{Set}$ is defined as $\alpha(x, f) = x \cdot f$ and the set in question is actually just the disjoint union $\coprod_{c \in Ob(\mathbf{C})} P(c)$.

The idea, then, is that given an element $x \in P(c)$ for some $c \in \mathbf{C}$, such an x will be acted on by all the morphisms $c' \xrightarrow{f} c$ in $\mathbf{C}$, and in such a way that composite morphisms act as above. In asking what the value of a function $f : c' \to c$ in $\mathbf{C}$ at such an element x will look like, we are asking about $P(f)(x)$. Regarding this in terms of a right action $\alpha(x, f) = x \cdot f$, we have for composite maps (which we write here with juxtaposition notation), $\alpha(x, fg) := x \cdot (fg) = \alpha(\alpha(x, f), g) = (x \cdot f) \cdot g$. If we agree, notationally, then, to let $x \cdot f = P(f)(x)$,[37] it is evident that the contravariance of the functor P is equivalent to specifying that $\mathbf{C}$ acts (and does so *on the right*) on P (regarded as a set). This is evident since

$$\alpha(x, fg) = \alpha(\alpha(x, f), g)$$

$$x \cdot (fg) = \alpha(x \cdot f, g)$$

$$x \cdot (fg) = (x \cdot f) \cdot g$$

$$P(fg)(x) = P(g)(P(f)(x))$$

$$P(f \circ g)(x) = P(g) \circ P(f)(x).$$

Having established the reasoning behind that terminological and notational choice, let us now consider more closely the various interpretations this presheaf action takes on in practice and explore what the natural transformations will look like for various presheaf categories we have already introduced.

37. That f gets written on the right of x here not only is meant to reveal the underlying right action, but it is a good notational choice since it accords with the induced notation for a composite arrow $f \circ g$ as $x \cdot (f \circ g) = (x \cdot f) \cdot g$.

2.4.2 Four Ways of Acting as a Presheaf

We will think of there being *four characteristic kinds of cohesivity or variability* presented by presheaf categories in accordance with four main ways the right action can be found to operate:

1. As *processual*, for example, as passing from sets indexed by one stage to sets indexed by another. Here, objects of **C** play the role of stages; for every c in **C**, the set $P(c)$ is the set of elements of P *at stage* c, while the morphisms model transitions between stages.
2. As *extracting boundaries* (or picking out components), for example, using the source and target map to pick out the vertices of a graph's edge, picking out lower-dimensional boundaries of simplices (generalizations of triangles or tetrahedrons to arbitrary dimensions). For something like a space that consists of points, edges, triangles, and so on, in *changing figures* we pass from higher-dimensional figures to lower-, so that, for instance, the action works by extracting the endpoints of an edge or extracting the edges of a triangle.
3. As *conditions on how different "probes" of a space relate to each other*, for example given a category **C** of geometrical figures or spaces of some sort, a presheaf X is regarded as a rule assigning to each object U (each "test space") of **C** the set $X(U)$ of admissible maps from U into a generalized space or geometrical object X—giving "probes of X by U"—and the presheaf action then concerns how maps from one test space U to another test space V induce maps of sets $X(V) \to X(U)$, ultimately codifying how probes of X by V transform into probes of X by U.
4. As *restriction*, for example, whenever some sort of topology is involved, where the data specified over or about a larger region can be restricted to the data specified over a region included in the former region.[38]

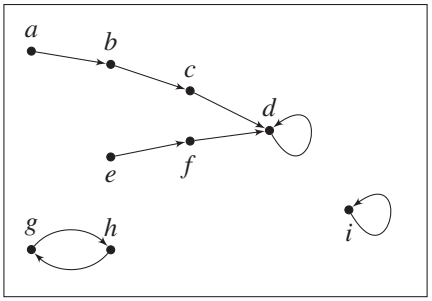
We illustrate these four action perspectives, in order, via specific examples.

Example 52 We discussed earlier how presheaves on **C**, as functors, can be thought of as providing a set of figures with the shape of the indexing category for each object in **C** and a *process* operator for each morphism in **C**. For each a in $\mathbf{C}^{op}$, the resulting set $F(a)$ is a set of elements of F at stage a, while each arrow between objects in $\mathbf{C}^{op}$ induces a transition map between the varying set F at stage a and the varying set F at stage b (for an arrow from b to a), so that, altogether, we are regarding the objects of **C** as playing the role of stages of $F : \mathbf{C}^{op} \to \mathbf{Set}$ and F itself as a *set that varies through the stages*.

This perspective of the action as exemplifying a kind of *process* is nicely illustrated by considering presheaves on a variety of finite indexing categories. For instance, consider the case of finitely free monoids. We said in section 2.3 that if $\mathbb{E}$ is free monoid on one generator σ, or the additive monoid of natural numbers, then $\mathbf{Set}^{\mathbb{E}^{op}}$, the category of presheaves on $\mathbb{E}$, is none other than the category of evolutive sets or dynamical systems. Objects of

38. It is not uncommon to see presheaves and sheaves introduced exclusively via this fourth approach—and, indeed, our first look at sheaves in chapter 5 falls under this umbrella—but the first three perspectives are also important to consider, especially since the first two often involve examples with finitely generated categories (and, as such, provide a good stock of simple and computationally tractable examples) and the third achieves a level of generality that, were it pursued to its end, would ultimately let us speak of sheaves in "higher dimensions."

$\mathbf{Set}^{\mathbb{E}^{op}}$ consist of a set X equipped with a "process" endomap. For objects X of $\mathbf{Set}^{\mathbb{E}^{op}}$, in other words, X supplies the set of possible states, and the given endomap σ gives rise to the evolution of states. Referring back to our earlier such X,



the idea is that, with this picture, we are displaying the presheaf consisting of $X(*) = \{a, b, c, d, e, f, g, h\}$ and where σ (i.e., $* \to *$) acts for instance on the figure a (i.e., on $* \xrightarrow{a} X$) to produce the figure $b : * \to X$; that is, $X(\sigma) : X(*) \to X(*)$ takes the particular $*$-figure given the name "a" to the particular $*$-figure given the name "b."[39]

Then a morphism in this entire presheaf category from a presheaf X (with endomap named α) to another object (presheaf) Y (with endomap β) will be the expected "equivariant map" (the usual map of relevance for such mathematical structures) in $\mathbf{Set}^{\mathbb{E}^{op}}$, that is, just a natural transformation $(X, \alpha) \xrightarrow{f} (Y, \beta)$, where this preserves the structure in the sense that $f \circ \alpha = \beta \circ f$.

We also saw how the same story is easily generalized to the category of n-evolving sets, that is, $\mathbb{E}_n$, freely generated by n nonidentity morphisms, so that the container of $\mathbb{E}_n$-sets would have figures similar to the above picture, except with (up to) n different processes carrying one $*$-figure to the next. We could also further consider finitely generated monoids, such as $\mathbb{E}_{1,R}$, where certain relations are imposed on the indexing category. For instance, taking $\mathbb{E}_{1,R}$ with one object and nonidentity morphism σ obeying some relation R—say the relation $\sigma^2 = \mathrm{id}_*$—the resulting category of presheaves on $\mathbb{E}_{1,R}$ gives rise to what is usually called, in other contexts, the category of *involution sets*. We could generalize this to any presentation of a monoid $\mathcal{M}$, $\mathbb{E}_{n,R}$, for n generators (i.e., sigmas), and R relations, ultimately leading us to show that the usual *Cayley graph* for a group is nothing other than a presheaf on the category $\mathbb{E}_{n,R}$.

In this context, we can take the opportunity to highlight that presheaves on a monoid are just equivalent to the usual *right actions* on a set by a monoid (which is in part responsible for the "right action" terminology). Recall that a monoid, viewed as a category $\mathcal{M}$ with just one object $*$, will imply that a set-valued functor on $\mathcal{M}$ yields just *one set*, $F(*) \in \mathrm{Ob}(\mathbf{Set})$. We must also supply, though, a function from $\mathrm{Hom}_{\mathcal{M}}(*, *)$ to $\mathrm{Hom}_{\mathbf{Set}}(F(*), F(*))$, that is, from M to $\mathrm{Hom}_{\mathbf{Set}}(F(*), F(*))$. In general, given a set A, for any sets X, Y, there is a

39. Incidentally, the notation $* \xrightarrow{a} X$ will be fully justified by the Yoneda results, covered in chapter 6. Looking ahead to that, for any object of $\mathbf{Set}^{\mathbf{C}^{op}}$, that is, some presheaf F, and any object c of $\mathbf{C}$, the set of elements of F of sort or type c can be naturally identified with the set of $\mathbf{Set}^{\mathbf{C}^{op}}$-morphisms from $\mathrm{Hom}_{\mathbf{C}}(-, c)$ to F, which is precisely what justifies the abuse of notation that alternately treats the elements of F of sort c as a morphism $c \to F$ in $\mathbf{Set}^{\mathbf{C}^{op}}$, letting any $c \to F$ be interpreted as a particular figure in F of sort c.

bijection

$$\mathrm{Hom}_{\mathbf{Set}}(X \times A, Y) \xrightarrow{\cong} \mathrm{Hom}_{\mathbf{Set}}(X, Y^A)$$

where $Y^A := \mathrm{Hom}_{\mathbf{Set}}(A, Y)$ the set of functions from A to Y. Moving between these two equivalent formulations via the bijection is sometimes called "currying." Our function from M to $\mathrm{Hom}_{\mathbf{Set}}(F(*), F(*))$ just belongs to $\mathrm{Hom}_{\mathbf{Set}}(M, F(*)^{F(*)})$. Currying, this is the same as a function $M \times F(*) \to F(*)$. Functors preserve identities by definition, so the monoid action law concerning the unit element e is satisfied, while the composition law for functions provides the other monoid action law. This shows that each monoid action is nothing other than a set-valued functor. Depending on the variance of the functor from a monoid M to **Set**, we get the left (covariant) or right (contravariant) M-sets.[40]

The variability provided by the right action in each of the above examples is fundamentally *processual*. This perspective is even clearer in an important related example, where we consider sets varying over some time-like linearly ordered category, such as over $\mathbf{N}$ (the linearly ordered set of natural numbers $\mathbb{N}$, regarded as a category), in terms of a functor. With such a category as our indexing category, objects in $\mathbf{Set}^{\mathbf{N}}$ are just sets varying through n successive stages, that is, $(n-1)$-tuples of maps:

$$X : X_0 \xrightarrow{f_0} X_1 \xrightarrow{f_1} X_2 \xrightarrow{f_2} \cdots X_{n-2} \xrightarrow{f_{n-2}} X_{n-1}.$$

The functor $\mathbf{N} \to \mathbf{Set}$ picks a sequence $X_0 \to X_1 \to \cdots$ of sets X_n and functions $X_n \to X_{n+1}$. A morphism between two such objects (sequences) is a sequence of functions, for example,

$$\begin{array}{ccccc} X_0 & \longrightarrow & X_1 & \longrightarrow & X_2 \cdots \\ \downarrow & & \downarrow & & \downarrow \\ Y_0 & \longrightarrow & Y_1 & \longrightarrow & Y_2 \cdots \end{array}$$

such that each individual square commutes. More generally, we have an $\mathbf{N}$-indexed family of functions $(f_i : X_i \to Y_i)_{i \in N}$ compatible with the maps, that is, whenever $i \leq j$, this square commutes:

$$\begin{array}{ccc} X_i & \xrightarrow{\alpha_{ij}} & X_j \\ {\scriptstyle f_i}\downarrow & & \downarrow{\scriptstyle f_j} \\ Y_i & \xrightarrow[\beta_{ij}]{} & Y_j \end{array}$$

This is equivalently just to describe a natural transformation between X and Y viewed as functors. As such, these transition functions $\alpha_{ij} : X_i \to X_j$ for each $i \leq j$ should satisfy

- $\alpha_{ik} = \alpha_{jk} \circ \alpha_{ij}$ whenever $i \leq j \leq k$
- $\alpha_{ii} = \mathrm{id}_{X_i}$ for all i.

Composing would look just as you would expect:

40. Just as for monoids, a *group action* on a set $S \in \mathrm{Ob}(\mathbf{Set})$ is just a functor $G \to \mathbf{Set}$ that sends the single object of G to the set S. Right G-sets are the same as the presheaf category $\mathbf{Set}^{G^{op}}$.

$$\begin{array}{ccccccccccc}
X & & X_0 & \xrightarrow{\alpha_{01}} & X_1 & \xrightarrow{\alpha_{12}} & X_2 & \xrightarrow{\alpha_{23}} & X_3 & \cdots \\
{\scriptstyle f}\downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & \\
Y & & Y_0 & \xrightarrow{\beta_{01}} & Y_1 & \xrightarrow{\beta_{12}} & Y_2 & \xrightarrow{\beta_{23}} & Y_3 & \cdots \\
{\scriptstyle g}\downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & \\
Z & & Z_0 & \xrightarrow{\gamma_{01}} & Z_1 & \xrightarrow{\gamma_{12}} & Z_2 & \xrightarrow{\gamma_{23}} & Z_3 & \cdots
\end{array}$$

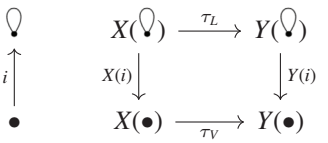
where we require that each individual square commutes. The basic idea here is that once an element is in a set, for example, $x \in X_t$, it *remains there*, that is, $\alpha_{tt'}(x) \in X_{t'}$. However, certain elements $a, b \in X_t$ could become identified in the long run (so the αs do not have to be injective); additionally, new elements can appear over time (something that is expressed by the fact that the maps do not have to be surjective).

The resulting category $\mathbf{Set}^{\mathbf{N}}$ that we have been describing is just a presheaf category $\mathbf{Set}^{\mathbf{C}^{op}}$, taking $\mathbf{N}^{op}$ as our $\mathbf{C}$, where $\mathbf{N}^{op}$ of course has natural numbers for objects and for morphisms $n \to m$ the pairs $\langle n, m \rangle$ such that $n \geq m$. Part of the power of this way of seeing this construction is that there is not really any need to restrict attention to linear orders. Thus, we could similarly consider the functor category $\mathbf{Set}^{\mathcal{P}}$ of sets varying over a preorder or poset $\mathcal{P}$, something we take up in later chapters. Here, too, it is entirely sensible to regard the resulting functor objects as $\mathcal{P}$-variable sets, since we have sets varying according to the shape of the order supplied by $\mathcal{P}$. For instance, the category $\mathbf{Set}^{\mathbf{R}}$, with $\mathbf{R}$ the ordered set of real numbers regarded as a category, has for objects sets varying through real time. More generally, the idea of a set varying over an ordered (poset or preordered) set is really all just a specialization of the general idea of a set "varying over" some arbitrary small category.

Example 53 Moving beyond examples using single-object categories such as the monoids introduced in the previous example, we might also consider presheaves on categories with more than one object. For instance, consider the presheaf from the very beginning of section 2.3, where the indexing category was

$$\mathbb{B} := \boxed{V \xrightarrow{\ i\ } L},$$

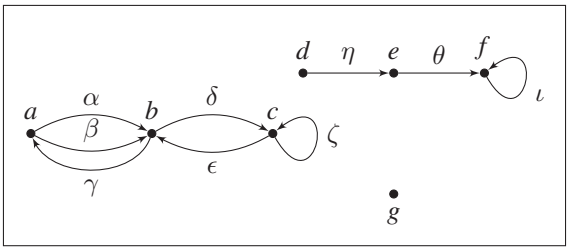
the object V standing for vertex and L for loop, and the single (nonidentity) morphism i including the vertex in the vertex of the loop. The presheaves on this $\mathbb{B}$ yielded bouquets, or those structures with any number of loops stationed at vertices. A particular presheaf X on this indexing category, then, gives all the data of a particular bouquet X, entirely described by a set $X(V)$ of vertices and a set $X(L)$ of loops, together with a function $X(L) \xrightarrow{X(i)} X(V)$ that acts to pick out the vertex of each of the loops. The action $\gamma \cdot i = c$ or $X(i)(\gamma) = c$, where $\gamma \in X(L)$, just extracts the appropriate vertex (boundary) of the loop in question. Finally, a natural transformation from one bouquet (presheaf on $\mathbb{B}$) X to another bouquet (presheaf on $\mathbb{B}$) Y will just amount to a rule that sends loop-figures in X to loop-figures in Y, point-figures of X to point-figures of Y, and does so in such a way that it preserves the incidence relations. In other words, $\tau : X \Rightarrow Y$ is a natural transformation making the diagram



commute, recovering the appropriate notion of a mapping between bouquets.

For our purposes, the thing to note in the above example is how the presheaf action is one that amounts to an operation of *boundary extraction*. The presheaf action operates by extracting from a loop-figure the vertex-figure to which it is attached, an operation it is very natural to think of as *taking the boundary*, or extracting the simpler elements that form the components of given (higher-dimensional) figures.

For another example of this type, consider a (directed, multi)graph X



As we have been doing with the other examples, we can regard this as a functor, that is, as being generated by a presheaf on a particular indexing category. Moreover, there is then the obvious action representing the "boundary extraction" of the source and target vertices (boundaries) from a given arrow. More explicitly, take for indexing category the category consisting of two nonidentity arrows (the identities again left implicit),

$$\mathcal{G} := \boxed{V \underset{t}{\overset{s}{\rightrightarrows}} A}$$

where the arrows s, t go from an object V (think vertex) to another object called A (think arrow). Regarding X as a presheaf on $\mathcal{G}$, then, is straightforward: the presheaf $X : \mathcal{G}^{op} \to \mathbf{Set}$ just assigns a set of vertex-shaped objects, a set of arrow-shaped objects, and functions $X(A) \xrightarrow{X(s)} X(V)$ and $X(A) \xrightarrow{X(t)} X(V)$, where the function $X(s)$ just assigns to each arc its source vertex and the function $X(t)$ picks out each arc's target vertex, thus giving us a presheaf action that can naturally be thought of as performing a sort of boundary extraction.

More explicitly, for our given graph X displayed above, the data (including some of the action data) is

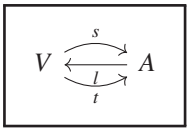
$$X(V) = \{a, b, c, d, e, f, g\}$$

$$X(A) = \{\alpha, \beta, \gamma, \delta, \epsilon, \zeta, \eta, \theta, \iota\}$$

$$X(s)(\alpha) = X(s)(\beta) = a, X(s)(\gamma) = X(s)(\delta) = b, X(s)(\epsilon) = X(s)(\zeta) = c, \ldots.$$

By taking presheaves like X for our objects, and natural transformations between such functors for our morphisms (which preserve the incidence relations), we recover the usual graph morphisms, that is, graph homomorphisms—showing that the presheaf category $\mathbf{Set}^{\mathcal{G}^{op}}$ is none other than **Grph** (actually, provided we have multigraphs, it is **Quiv**), consisting of irreflexive directed graphs. We can perform a similar analysis for other sorts of graphs, for

instance reflexive graphs. A graph is reflexive provided each vertex v is assigned a designated edge $v \to v$. Equivalently, in terms of quivers, a reflexive quiver has a designated identity edge $\mathrm{id}_X : X \to X$ on each object X. For reflexive graphs, we would take for our indexing category

$\mathcal{G}' :=$ 

which consists of two nonidentity arrows, just as in $\mathcal{G}$, but now with an extra, third arrow (l for "looped edges") going in the other direction from the two already given. This indexing category is subject to the following equations:

$$l \circ t = \mathrm{id}_V = l \circ s.$$

Using this as our indexing category, $\mathbf{Set}^{\mathcal{G}'^{op}}$ recovers the category of reflexive (directed, multi)graphs, **rGrph** (or **rQuiv**).[41] Maps of reflexive graphs, that is, natural transformations between the presheaf objects of $\mathbf{Set}^{\mathcal{G}'^{op}}$, must not only respect the source and target maps, but also the extra piece of structure given by l.

As a final observation on this sort of example, let us briefly note that we could generalize all this to (n-uniform) "hypergraphs" taking values in "multisets"—where a *hypergraph* is a generalization of a graph, where edges can join any number of vertices, and where a *multiset* is a generalization of the standard "set," where there can be multiple occurrences of a given element—or still other graph structures. Moreover, as we discussed in example 34, each category can be regarded as a directed graph with some structure. In order to begin to appreciate the third perspective, we could generalize this and consider the n-dimensional analogue of a directed graph, that is, via so-called globular shapes.[42]

Definition 54 For $n \in \mathbb{N}$, an *n-globular set* X is a diagram

$$X(n) \underset{t}{\overset{s}{\rightrightarrows}} X(n-1) \underset{t}{\overset{s}{\rightrightarrows}} \cdots \underset{t}{\overset{s}{\rightrightarrows}} X(1) \underset{t}{\overset{s}{\rightrightarrows}} X(0)$$

of sets and functions such that $s(s(x)) = s(t(x))$ and $t(s(x)) = t(t(x))$ for all $m \in \{2, \ldots, n\}$ and $x \in X(m)$.

But an n-globular set can also be defined as a presheaf on the category $\mathbb{G}_n$ generated by the objects and arrows

$$n \underset{\tau_n}{\overset{\sigma_n}{\leftleftarrows}} n-1 \underset{\tau_{n-1}}{\overset{\sigma_{n-1}}{\leftleftarrows}} \cdots \underset{\tau_2}{\overset{\sigma_2}{\leftleftarrows}} 1 \underset{\tau_1}{\overset{\sigma_1}{\leftleftarrows}} 0$$

which moreover satisfy the equations $\sigma_m \circ \sigma_{m-1} = \tau_m \circ \sigma_{m-1}$ and $\sigma_m \circ \tau_{m-1} = \tau_m \circ \tau_{m-1}$ for all $m \in \{2, \ldots, n\}$.

In short, the category of n-globular sets can also be defined as the presheaf category $\mathbf{Set}^{\mathbb{G}_n^{op}}$. For X an n-globular set, we call elements of $X(m)$ the m-cells of X: for instance, $a \in X(0)$ is a dot or vertex labeled by a; $f \in X(1)$ is an arrow with a source and target boundary;

41. Observe that there is an obvious forgetful functor $U : \mathbf{rQuiv} \to \mathbf{Quiv}$ from reflexive graphs to irreflexive graphs, where this acts by neglecting the structural component l.

42. The following definition and page or so of discussion can be skimmed or skipped on a first reading and resumed with the next example.

$\alpha \in X(2)$ looks just like a natural transformation arrow satisfying certain relations; a three-cell $x \in X(3)$ an arrow between arrows of the natural transformation type, and so on. In this way, various prominent mathematical constructions including the likes of simplicial sets, cubical sets, and globular sets can be construed as examples of presheaf categories. The basic idea in all this is that one selects a category **C** of cell shapes with morphisms "face inclusions" and "degeneracies"; then, as above, one produces a presheaf category $\mathbf{Set}^{\mathbf{C}^{op}}$, and the boundary extraction action perspective will generally fit such situations. But such constructions also encourage the (third) view that for a category **C**, whose objects can be regarded as certain geometrical figures or spaces of a certain sort and its morphisms as structure-preserving morphisms between those spaces, presheaves on such a category give rise to spaces modeled on **C** in the sense that they are probed by the objects of **C**.

The idea with this third perspective can be roughly sketched as follows.[43] If **D** is taken to be, for example, the category of sets or certain topological spaces,[44] and if we regard the indexing category **C** as some category supplying the shapes or generic (geometrical) figures, then the presheaf category $\mathbf{D}^{\mathbf{C}^{op}}$ will be a (generally large) category that will include more general geometrical or spatial objects that are probable or testable with the help of **C**. Altogether, this provides a perspective according to which a presheaf on **C** can be seen as a very general space *modeled on* **C**, where this otherwise unknown space emerges from "probing" it with the known objects of **C**.

In referring to "probes" of a hypothetical space X (for now, just think of some generic space, not necessarily a topological space in the strict sense), we are really thinking of all the ways of mapping into X using the objects of **C**. In other words, if you start with a test space U (an object in **C**) and are returned a set $X(U)$, we are thinking of this set $X(U)$ as designating the set of ways U can be mapped into X, supplying the probes or ways of testing X with U. However, these probes alone will not usually suffice to give you a very discriminating or complete understanding of the space X. To attain a more complete picture, you also need to know about how the different tests or probes of the space relate to one another. This is what the presheaf action takes care of. If you have a map $f : U \to V$ in **C**, then given some (probe) element $V \xrightarrow{p} X$ of $X(V)$, precomposing with f (acting on the right) will induce a map going in the other direction $X(V) \xrightarrow{X(f)} X(U)$ that will tell you how probes of the space X by V change into probes by U; and with *that* information, you can get an accurate picture of what X itself is. In this way, in describing a generalized space modeled on the objects of **C**, we are in fact describing nothing but a presheaf X on **C**, where each presheaf is a rule assigning to each $U \in \mathbf{C}$ the set $X(U)$ of admissible maps from U into the space X, and where this comes with the information of how certain probes of X change into other probes of X. One of the purposes of doing this is that by probing a big space with a number of smaller or simpler test spaces, not only can we model parts of the space into which we are mapping but we can ultimately look to piece together the small or partial tests into information about tests with bigger test spaces, arriving at a picture of the overall space of interest. This is of particular importance since the information such probes

43. This general perspective largely follows Lawvere; see, for instance, Lawvere (2005).

44. One usually starts with thinking about topological spaces, that is, the category **Top**, but really we just need it to be a category and for this category to support some notion of how certain objects can be *covered* by other objects. There is much more on this in chapter 4 and in the final chapters.

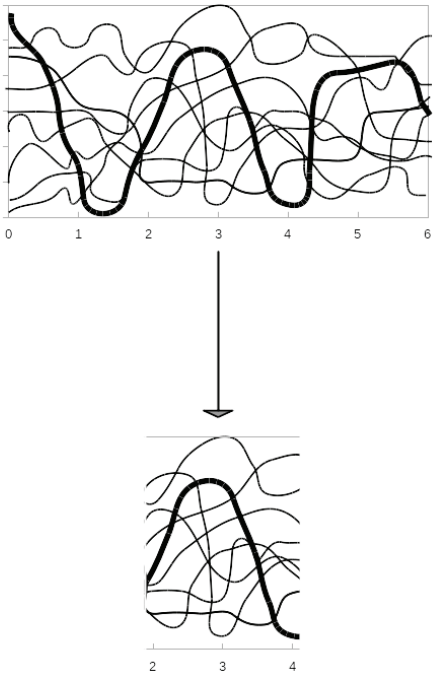
gather turns out to be most useful precisely when the presheaves are in fact sheaves, that is, satisfy some further consistency conditions.[45]

Example 55 To illustrate the last (but arguably most significant) perspective on the presheaf action—namely, action by *restriction*—we can begin by considering the construction of a presheaf on the partial order of open sets $\mathcal{O}(X)$ (or **Open**), for X a topological space.[46] A presheaf on X is just a functor $F : \mathcal{O}(X)^{op} \to \mathbf{Set}$. For each open $U \subseteq X$, we then think of the set $F(U)$ as the set that results from assigning set-values or data throughout or over all of U. An open subset $V \subseteq U$ can be seen in terms of an inclusion arrow $V \hookrightarrow U$ when regarded in the poset $\mathcal{O}(X)$ seen as a category, so applying the (contravariant) functor F will give us a function that passes from the data assigned throughout or specified over U (the generally "larger" region) to the data assigned throughout the subregion V, in a process aptly called the *restriction*, and typically denoted by $\rho_V^U : F(U) \to F(V)$ (or just $F(V \hookrightarrow U)$). Especially when the particular application involves looking at all the *functions* of a certain type (e.g., continuous functions) defined throughout that region U, given an element $f \in F(U)$, one sometimes denotes $\rho_V^U(f)$ by $f|_V$ and speaks of the *restriction* of f from U to V, treated like the usual restriction of a function along a part of its domain.

As a first illustration of such a functor, we can consider the set of all continuous real-valued functions, that is, functions from $U \subseteq X$ to $\mathbb{R}$. Importantly, when there is an inclusion of opens $V \subseteq U$, we will have a restriction $\rho_V^U : \mathbf{Top}(U, \mathbb{R}) \to \mathbf{Top}(V, \mathbb{R})$, which just sends $f : U \to \mathbb{R}$ to $f|_V : V \to \mathbb{R}$. The presheaf here thus acts to *restrict* the collection of functions given over some region (say, $(0, 6)$) down to the open subsets of that region (say, $(2, 4)$ in particular), as suggested by the following picture:

45. The reader intrigued by this admittedly more subtle perspective may find the extended discussion in nLab Authors (2019) particularly illuminating; the paragraph above leans on this discussion.

46. Topological spaces and the poset of open sets of a space are discussed in ample detail in chapter 4.



The action of this presheaf is thus given by *restriction*, an action that is clearly functorial. Observe, also, how each of the collections $\mathbf{Top}(U, \mathbb{R})$ of continuous functions in fact assembles into a ring structure, meaning that the restriction functions will in fact be ring homomorphisms—as such, this presheaf $\mathbf{Top}(-, \mathbb{R})$ is actually a presheaf of rings (as opposed to being valued in **Set**).[47]

For another restriction-type example, but of a rather different flavor, start by considering for regions the set J of jurisdictions with their subjurisdictions, that is, $(J, \subseteq)$ a preorder.[48] We can consider that within the set of *possible laws*—where laws are just treated as propositions, that is, objects of the preorder **Prop** regarded as a category whose objects are logical formulas or propositions and whose morphisms are (equivalence classes of) *proofs* or derivations that one proposition implies another—some of these laws are being followed by all people in the region. To each jurisdiction V, then, we can assign a set $R(V)$ consisting of whatever laws are being respected by all the people throughout V. In other words,

47. Another standard way of producing presheaves on a space arises by taking the *local sections* of a continuous function $p : E \to X$, via the "local section functor." A local section of p is a continuous function $s : U \to E$ from an open subset U of X to E, such that $p \circ s(x) = x$ for all $x \in U$. If we let $\Gamma(p)(U) = \{s : U \to E \mid s \text{ is a local section of } p\}$, then by considering that whenever $V \subseteq U$ we can restrict local sections over U to local sections over V, we see that this defines a presheaf on X. Don't worry if this doesn't make sense yet. We have a lot more to say about this local section approach in chapters 5 and 8.

48. This example comes from Spivak (2014, 263-264). See also Awodey (2010) for more on the category of propositions that is used in the example.

laws are being assigned *locally* to each jurisdiction; after all, a law is dictated to be valid only within a specific region. If V is a subjurisdiction of U, that is, $V \subseteq U$, then any law respected throughout U is obviously respected throughout V, so we can *restrict* from the laws respected throughout U to those respected throughout V. Clearly any law respected throughout the state of Illinois will be respected throughout any county in Illinois, so we can regard such a law given over Illinois from the restricted perspective of a county in that state. But observe that the converse is not true! A law respected throughout a part of Illinois need not be respected throughout all of Illinois.

Here we have a local assignment of data to the space of jurisdictions that moreover obeys the property that whenever we have a region V included in U, then the action of the presheaf works in the opposite direction: it takes data assignments given throughout U and *restricts* them to (the same) data assignments now given throughout V, the smaller region. The idea to keep in mind here is this: if you have some data (like a list of those laws being respected by everyone) assigned to some region (like Illinois), and you have another list of those laws being respected by everyone in some subregion included in Illinois (like Cook County), then you will expect that the list of laws respected by everyone throughout Cook County will be (equal if not) larger than the list of laws respected by everyone throughout Illinois. In a larger region, there are more chances for the data not to fit—for example, for someone to fail to respect that law—than there is in a smaller region.

The main takeaway is that in the previous construction, we have made use of two key ingredients: (1) a local assignment of data to a space (each of the laws in the "respected laws data" is expected to hold throughout all of the jurisdiction region to which it is assigned); and (2) a natural operation of restriction (induced by the natural inclusion relation governing the overall space of jurisdictions) allowing us to move from the data assigned throughout a region to data assigned throughout subregions. Formally, these two ingredients just specify what we need to have a functor R that is *contravariant*, that is, we have been describing a presheaf $R : J^{op} \to \mathbf{Prop}$.

With an eye toward sheaves, the restriction-style action is in some sense the most decisive of the four perspectives considered, or at least the most immediately relevant to the subsequent initial presentation of sheaves in terms of sheaves on topological spaces. Thus, it pays to understand it well. We could dwell at length on the notion of restriction and its relation to some of the other key basic categorical concepts that we will meet—for instance, that restrictions are not "right cancellable" in general, and that it is easy to construct many examples of *distinct mappings* that have equal restrictions to the same part, that is, mappings f, g such that $f|_i = g|_i$ but $f \neq g$. For now, a few general observations concerning restriction may be worth stressing.

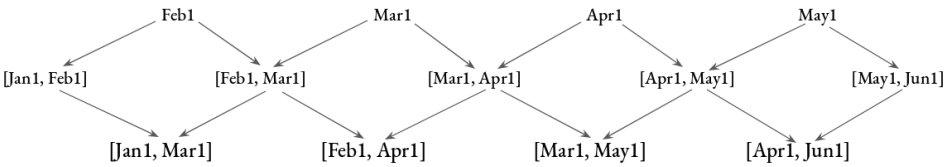
Given an inclusion $V \hookrightarrow U$, restriction tells us that we can take some $x \in F(U)$ and restrict that data assignment to the part of U that makes up V, and this will leave us with a viable data assignment on V. It should be easy to see, both intuitively and precisely (on the model of function restriction), how this amounts to a restriction. However, it is important to recognize what is *not* going on: it would be a mistake to take the language of restriction as implying that, at the level of the presheaf itself, we are passing from a (generally) bigger set of data to a smaller one—"restricting our attention" as it were. Strictly speaking, at the level of the maps between the presheaf data $F(U)$ and $F(V)$, this is *not* what is going on.

This should already be evident from close consideration of the laws example, where the set of respected laws $R(U)$ specified over a larger region U will actually typically be *smaller* than the set of respected laws $R(V)$ specified over the subregion $V \subseteq U$. The same is true of the continuous functions: it is easier to be continuous on a smaller region; that is, over a bigger region there will be more opportunities to fail to be continuous. It is perhaps an unnecessary warning, but the point is that, at the level of the presheaf maps themselves, for example, moving from $R(U)$ to $R(V)$, we are not generally dealing with a restriction in the sense of moving from a bigger *dataset* (set of value assignments) to a smaller dataset. Confining our attention to U and V as regions or components of a space, it makes sense to think of these objects (regions) as *constraints* of sorts, according to which V, a subregion, amounts to a weaker constraint on any data specified locally over the regions. It should be evident that given any inclusion of a smaller region into a larger region, more data will generally be able to satisfy the weaker constraint (corresponding to the smaller region) than will satisfy the stronger one (corresponding to the larger region). Yet, at the level of a particular data assignment, we can regard the presheaf maps as amounting to a restriction of that data along inclusions of subregions. The next and final example of this chapter should help to further clarify this.

Example 56 Consider time intervals as objects and morphisms given by inclusions, yielding a category we will denote $\mathcal{T}$. For concreteness, suppose we consider the period spanning from January 1 (at midnight) of 2018 until June 1 (at midnight) of 2018. Then, suppose this period is decomposed into various two-month intervals

$$[Jan1, March1], [Feb1, April1], [March1, May1], [April1, June1],$$

that together "cover" the entire half-year period from $Jan1$ through to the end of May.[49] The natural overlapping subintervals produce the following overall structure on the system of intervals ordered by inclusion (as indicated by the inclusion arrows):



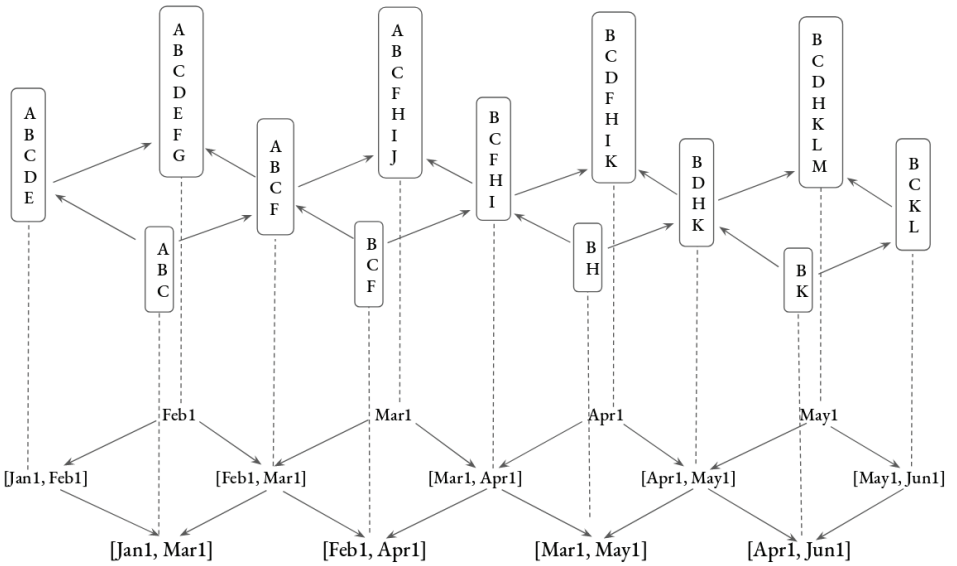
where the single dates like Feb1 are short for the degenerate interval $[Feb1, Feb1]$ representing the instant of midnight on February 1.

Now, for a particular company with some (generally fluctuating) stockpile of products, we can define a contravariant functor $S : \mathcal{T}^{op} \to \mathbf{Set}$ that assigns to each time period $[t, t']$ the products that are in stock throughout the *entire interval* $[t, t']$ (where, for the moment and for simplicity, we just imagine that a product is simply present or absent, say, as if the only important question was whether the company had at least one item of the product or had none, ignoring the question of quantity). $S([t, t'])$ is thus just the set of products the company has in stock throughout the entirety of the time period $[t, t']$.

49. For now, you can think of this notion of "covering" in a naive way. We will be precise about this sort of thing below, starting in chapter 4.

Then, for any inclusion of time periods $i : [t, t'] \hookrightarrow [u, u']$, the functor S acts (contravariantly) *by restriction*, mapping each stocked item onto itself. Clearly, any product present in the company's store throughout the bigger time interval $[u, u']$ must be present as well throughout any subinterval $[t, t']$. But this tells us that the "list" of products recorded as present throughout the bigger time interval $[u, u']$ is in general likely to be shorter or smaller than the list of products assigned to the smaller time interval $[t, t']$.

A particular presheaf on such a $\mathcal{T}$ might then be given by something like the following (where each of the A, B, C, and so on, sitting over each interval-object, represents one of the products held by the company throughout the entire interval):



Inspecting the diagram, one can see that the sets of value assignments (of products present) throughout each interval are generally smaller over larger regions (time intervals); thus, strictly attending just to the part of the diagram sitting on top of the network of time intervals, the presheaf arrows in fact generally go from smaller sets to larger ones.[50]

A final thing to realize is that, in general, restriction along an inclusion is *not necessarily either surjective or injective* (despite what a naive understanding of the language of restriction might seem to imply, for instance, suggesting at least surjectivity). An easy counterexample is provided by the following.[51] As was seen in an earlier example, restriction of continuous functions is continuous. However, take the (poset) category $\mathbf{A}$ that consists of just two objects, U and C with the single nonidentity (inclusion) map $U \to C$, where U is the open interval $(0, 1)$ and C is the closed interval $[-1, 1]$. Now let the presheaf $F : \mathbf{A}^{op} \to \mathbf{Set}$ act on objects as follows: $F(U)$ is the set of all continuous real-valued functions given over the open interval $(0, 1)$ and $F(C)$ is the set of all continuous real-valued functions over the closed interval $[-1, 1]$. Then the induced presheaf action $F(C) \to F(U)$ is clearly by restriction; yet, it should be evident that this particular restriction process can

50. This relates to the warning discussed just before this example.

51. This counterexample is derived from Lawvere and Rosebrugh (2003).

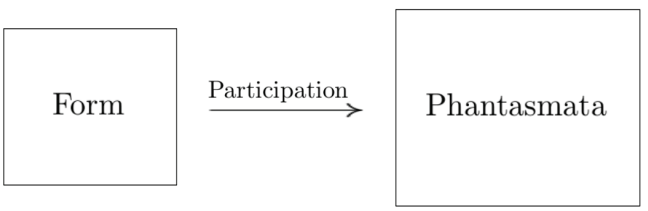
be neither surjective nor injective. It is not surjective since there are functions that remain continuous over $(0, 1)$ while having discontinuities at either or both "end points"—in particular, at 0—so that such functions cannot come from any continuous functions specified over all of $[-1, 1]$. It is not injective since there exist distinct continuous functions given over all of $[-1, 1]$, each of whose restrictions to $(0, 1)$ are identical.

2.5 Philosophical Pass: The Four Action Perspectives

Box 2.1
On the Presheaf

The general understanding of presheaves developed in section 2.3 might be thought of in terms of Plato's notion of the *form* or *shape* (*eidos*) of something, that structural schema according to which the concrete realizations thereof are organized. This form also supports a great variety of realizations or "manifestations" (*phantasmata*), and the plurality of particular manifestations of it populates a world that acts as some sort of "receptacle" for such instantiations of the forms. The process by which the manifestations are unfolded according to the structural schema of the form is what Plato would call the *participation* (*methexis*) of the form. The form is held to be invariant, its components sufficiently generic, and altogether it is fundamentally simpler (and so, in the end, more *intelligible*) than its many realizations or manifestations.

Applied to presheaves, the gist here is that the "generic figures" supplied by the domain category **C** act as something like the form, while the value assignments $P(c)$ for each object of **C** supply something like the concrete appearances or manifestations of the static components of the form, and the presheaf action enforces dynamic relationships between the various manifestations modeled on the invariant generic relationships between the components of the form. The presheaf P itself, on this way of seeing things, would then be nothing other than the process of manifestation or participation of the form in concrete particulars, and to understand how the concrete manifestations and their components respect among themselves the relations that obtain between the components of the form itself is just to understand the general functoriality of the functor P.



Presheaves accordingly supply a uniform framework for capturing, in an at once compressed and illuminating way, many structures that appear throughout math and that can otherwise appear, in their traditional presentation, rather complicated or haphazard (frequently leading to a complicated or haphazard description).

Many important mathematical structures and categories—including some of those already discussed, such as dynamical systems, bouquets, graphs, hypergraphs, and more—arise as a presheaf category consisting of contravariant functors on some simple indexing category, where the result of applying the functor to the objects of the indexing category yields what we naturally think of as containers (in **Set**) holding on to various manifestations or figures

each of which conforms to the shape or form determined by the generic figures populating the indexing category (one for each of its objects), and where the changes of figure indicated by the indexing category (given by its morphisms) are respected by the figures of the container. While this perspective is perhaps most appropriate, or easy to countenance, when the objects and morphisms of the indexing category **C** have some sort of geometrical interpretation, it is a surprisingly useful perspective even in more general cases.

As for the four perspectives on the presheaf action, the fundamental idea that they share is that the domain category **C** plays the role of specifying the general internal structure or schema—in the form of the figure-types or shapes, the glue, the nature of the internal dynamic, or the locality of data assignments—in which all the sets in $\mathbf{Set}^{\mathbf{C}^{op}}$ must participate. The resulting presheaf category in each case has for its objects different instantiations or realizations of the general form supplied by **C**. The other way of looking at this is that the domain category plays the role of a parameter specifying in an invariant form how (temporal, dynamic, geometric) variation or cohesion is to take place, while the target category (**Set**) serves as the container or arena holding on to all the particular values or results of trying to "participate in" or "realize" this form of variation. One might accordingly think of a presheaf itself as mediating between the invariance or fixedness of a structure "outside of time" belonging to the domain category, on the one hand, and its multifarious concrete presentations or manifestations "in time," on the other.

Most of the presheaf functors above are valued in **Set**, which can be useful in taming many problems or otherwise complicated structures, and there are good reasons for the central role **Set** plays in classical category theory, largely accounting for why presheaves classically take values in **Set**. (Again, these reasons have to do mainly with the Yoneda results, covered in chapter 6.) However, it is worth emphasizing that, philosophically speaking, presheaves are anything but the static and qualitatively barren objects the usual set-theoretical perspective on sets as "bags of fixed objects" might encourage us to believe.

Considering presheaves as coming equipped with an action that is *processual* recaptures a dynamic perspective in which objects are not regarded as static collections; instead, the sets are seen as evolving through stages, either merely temporally or in accordance with an internal dynamic (as in cases of evolutions subject to certain equations). Against the generally discrete and static context of sets, this restores a more continuous and dynamic perspective. The *boundary-extraction* perspective, for its part, reveals the incidence relations, relations that together describe something like how the overall structure "holds together" or coheres. Against the usual set-theoretical perspective of sets of objects grouped together more or less arbitrarily into a set that cannot internally differentiate objects or discern important qualitative (or dimensional) features of those objects or their modes of relation, this perspective restores a kind of continuity in telling us how the various components of a structure can be regarded as glued or stitched together from other (lower-dimensional) components. The third perspective lets us regard a space in terms of all the ways of probing it from the outside and thinking about the entire space in terms of how these various probes behave with respect to one another. This perspective, similar to that of the *relationism* of Yoneda (discussed in chapter 6), is more "continuous" in the sense that it insists that we understand something in terms of all the relations or perspectives on it, instead of as something set off on its own or intelligible by itself. Finally, acting via *restriction*, presheaves open onto a range of relationships between the parts of a whole. In general, such a relationship emerges as "regular" in the sense that in passing from data specified over some containing region to data over a subregion, there remains a kind of conformity or identity of the data given over the parts in relation to the same rule or function describing the containing region. This perspective thus opens onto the notion of a conformity

of parts of a whole to a single rule or idea (as opposed to the usual set-theoretical consideration of a whole independently of the specific way, beyond whether or not a part *belongs*, the whole enforces relationships among the parts).

In short, while we are able to benefit from certain tame properties of **Set**, the presheaf perspective lets us recapture a generally more dynamic, nuanced, and continuous (in a general sense) perspective on many structures of interest.

This is a section of doi:10.7551/mitpress/12581.001.0001

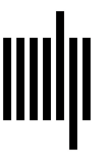
Sheaf Theory through Examples

By: Daniel Rosiak

Citation:

Sheaf Theory through Examples
By: Daniel Rosiak
DOI: 10.7551/mitpress/12581.001.0001
ISBN (electronic): 9780262370424
Publisher: The MIT Press
Published: 2022

The open access edition of this book was made possible by generous funding and support from Arcadia – a charitable fund of Lisbet Rausing and Peter Baldwin, and MIT Press Direct to Open



The MIT Press

The open access edition of this book was made possible by generous funding from Arcadia—a charitable fund of Lisbet Rausing and Peter Baldwin.

The MIT Press would like to thank the anonymous peer reviewers who provided comments on drafts of this book. The generous work of academic experts is essential for establishing the authority and quality of our publications. We acknowledge with gratitude the contributions of these otherwise uncredited readers.

This book was set in LaTeX by the author.

Library of Congress Cataloging-in-Publication Data

Names: Rosiak, Daniel, author.
Title: Sheaf theory through examples / Daniel Rosiak.
Description: Cambridge, Massachusetts : The MIT Press, [2022] | Includes bibliographical references and index.
Identifiers: LCCN 2021058949 | ISBN 9780262542159 (paperback)
Subjects: LCSH: Sheaf theory.
Classification: LCC QA612.36 .R67 2022 | DDC 514/.224–dc23/eng20220521
LC record available at https://lccn.loc.gov/2021058949