Article

# Sheaf Mereology: Parts and Wholes in a Topos-Theoretic Setting

**Firstname Lastname [1], Firstname Lastname [2] and Firstname Lastname [2,*]**

[1] Affiliation 1; e-mail@e-mail.com
[2] Affiliation 2; e-mail@e-mail.com
* Correspondence: e-mail@e-mail.com; Tel.: (optional; include country code; if there are multiple corresponding authors, add author initials) +xx-xxxx-xxx-xxxx (F.L.)

**Abstract**

A single paragraph of about 200 words maximum. For research articles, abstracts should give a pertinent overview of the work. We strongly encourage authors to use the following style of structured abstracts, but without headings: (1) Background: place the question addressed in a broad context and highlight the purpose of the study; (2) Methods: describe briefly the main methods or treatments applied; (3) Results: summarize the article's main findings; (4) Conclusions: indicate the main conclusions or interpretations. The abstract should be an objective representation of the article, it must not contain results which are not presented and substantiated in the main text and should not exaggerate the main conclusions.

**Keywords:** mereology; fusions and integral wholes; sheaves; point-free topology; frames and locales; toposes; modality; merology logic; categorical logic

## 1. Introduction

Standard presentations of mereology tend to take what we might call a "parts-first" approach. You start by taking the parthood relation as primitive, and then you proceed by stipulating axioms that govern the relation. The goal is to choose your axioms well enough that the resulting models that satisfy your theory align nicely with the actual part-whole complexes that we encounter in the world around us.

By most standard accounts (e.g., [1], [2], or [3]), partisans of the parts-first approach have more or less agreed on a common (minimal) "core" known as "classical mereology." First, classicists adopt the following principles that govern the ordering of the parts:

- Parthood is reflexive, antisymmetric, and transitive (i.e., it is a partial order).

Second, classicists adopt the following decomposition principle that governs how wholes decompose:

- Wholes decompose into more than one proper part (i.e., parthood obeys some form of so-called "supplementation," "remainder," or "complementation" principle which says that no whole consists of only a single proper part — there must be some remainder or relative complement).

Third, classicists adopt the following principle that governs how parts fuse into wholes:

- Any collection of parts whatever forms a fusion (i.e. unrestricted fusion).

Classicists also require (either as an explicit axiom or as a consequence of the other axioms) some version of extensionality (TODO: consult Contoir's article on this and maybe cite it):

Received:
Revised:
Accepted:
Published:

**Citation:** Lastname, F.; Lastname, F.; Lastname, F. Title. *Philosophies* **2025**, *1*, 0. https://doi.org/

- If wholes have the same parts, then they're the same wholes.

With the above axioms fixed, the classicist can then define a number of other useful notions in the obvious ways, e.g.:

- Overlap and underlap
- Complement/difference
- Etc.

At first site, most of the classicist's principles can feel deeply intuitive. However, philosophers have objected to virtually all of them. Take for instance that parthood is transitive: if $x$ is a part of $y$ and $y$ is a part of $z$, then surely $x$ is a part of $z$. But there appear to be counter-examples. For example, my appendix is a part of me, and I am a member of the orchestra, but my appendix is not a member of the orchestra.

A standard response is to point out that this sort of objection exploits an ambiguity: we utilize different, more specialized notions of "parthood" when we talk about the integration of the parts of biological organisms vs. those of orchestras. My appendix is a part of me under one description (as a part of a biological organism), while I am a part of the orchestra under another (as a member of a musical ensemble).

Defenders of the classical axioms have said that the fact that we can partition the general notion of parthood into more specialized versions only shows that the above axioms do in fact characterize a general notion of parthood, characterized precisely by the above classical notions (TODO: cite Simons, Varzi, etc).

However, one can't help but feel that there is something circular about this response, since it turns on the assumption that the different notions of functional unity are species (or determinations, or partitions, or what have you) of a more generic relation. But the existence of that generic relation hasn't been established, and there is no reason to think that mereological pluralism isn't correct — namely, that there are many parthood relations, not one (TODO: cite Fine, etc.).

Another common objection to the classical approach revolves around composition principles. In particular, if we adopt unrestricted fusion, as the classical mereologist does, then we seem to get too many fusions. For instance, take the pencil on the table in front of me and your left knee. Are we really to believe that there is a fusion of that pencil and your left knee? Such a fusion would have two parts that live quite far apart (possibly even on different sides of the globe).

Defenders of the classical approach do have a response though: just because we may not have a word or concept that names the pencil+knee fusion, that doesn't mean it doesn't exist. (TODO: cite Varzi, etc.) Indeed, just as Moore attempt to show that extramental things exist by holding up his two hands and saying, "Here is one hand, here is the other," so too might one try to show that the pencil+knee fusion exists by saying, "there is the pencil, there is the knee."

Again though, one can't help but feel that there is something circular about this response. To appeal to the pencil+knee's fusion after its fused-ness has been questioned just brings the principle under scrutiny back into the mix. A better response would provide an independent reason to think that pencil+knee qualifies as more than a "mere Cambridge" fusion.

Whether these objections/responses constitute any real conceptual clarification or are inexorably stuck in semantic circularities is not something we want to decide here. We mention these points only because they illustrate something else: they illustrate that we seem to have intuitions not just about parts, but also about fusions. In the two objections just mentioned, we seem to have certain conceptions of integrated fusions somewhere in the back of our minds, and those seem to be driving the objections.

For example, the reason it seems wrong to say my appendix is not part of the orchestra is because we seem to think that biological organisms are integrated in a different way than orchestras. Similarly, the reason we can so easily think that a pencil and a hand don't fuse is because they don't integrate in one of the ways that we ambiently accept as legitimate.

That leads to the following question: if we have ambient intuitions about which collections count as fusions and which ones don't, then why not take a "fusions-first" approach to mereology? Instead of taking parthood as the primitive relation, and then try to work up to a notion of fusions, why not take fusions as the primitive relation, and then work backwards to parts?

Such an approach is far less common in the technical mereological literature. That raises yet another question: why has the "fusions-first" approach been so neglected? One proposal is that it might seem to be too unwieldy. One might say that there are just too many different ways that things can integrate into wholes/fusions, and so it is a hopeless task to try and enumerate them and offer any sort of a unifying taxonomy. (TODO: Cite Simons.)

Another reason might be that such a view would be inelegant and perhaps would even fail to qualify as an "explanation" of the part-whole phenomenon altogether. If you ask me to explain why various $X$s seem to exhibit the same (or sufficiently similar) properties, it would be quite dissatisfying if I said, "that's easy, there is no unifying explanation."

TODO: remove/rewrite these last two paragraphs. The "fusions-first" approach is not so uncommon in the literature as I just made it out to be. Mereotopology exists as a branch of hereology, for all intensive purposes, precisely because it is the fusions-first response to the classical parts-first approach. Cite Casati and Varzi, chapter 3 and others.

Fortunately, a "fusions-first" approach need not be as doomed as it may seem. In this paper, we claim that there is a satisfying "fusions-first" approach to mereology, and we present it in what follows. To accomplish this, we will build a bridge between category theory and philosophy. In particular, we will take well-known techniques used to manage the gluing-together of parts in algebraic geometry and topos theory, and we will apply those techniques to the realm of mereology.

### *1.1. From Parts to Sheaves*

To begin, we want to suggest that it is useful to draw a distinction between what one might call the algebra of parts on the one hand, and the integrity or gluing-together of the parts on the other. To get a sense of what this distinction means, and why it is useful, fix a part-whole complex to analyze (a statue of Dion, let's say), and let an enumeration of its parts be given. The classical principle of unrestricted fusion says that any combination of those parts glues into a fusion. In essence, this generates all possible combinations of parts. As such, it nearly yields a complete lattice, with overlap and underlap serving as the meet and join operators.

However, it only *nearly* yields a complete lattice because mereologists have been relunctant to allow a bottom element. Since mereologists are ontologists, they find a null element to be ontologically suspect. And indeed, what could an empty thing that is part of all other things possibly *be*? So, instead of admitting it into their mereological systems, classical mereologists have simply omitted it altogether. David Lewis ([4]) even went so far as to formulate a version of set theory that had no empty set.

Yet despite their suspicion of a null element, classical mereologists have not shied away from allowing all possible fusions to exist, as noted already. Since the bottom element of a lattice is the empty join, we can put the point like this: classical mereologists are ontologically conservative about empty joins, and ontologically permissive about non-empty joins. As Tarski pointed out long ago, the principles of classical mereology thus yield a boolean algebra, with the bottom element removed (TODO: cite).

But it is difficult to see the motivation here. On the one hand, if you want to be ontologically conservative, then why allow so many fusions? If we are going to be suspicious of an empty join, then wouldn't we also be suspicious of the fusion of (say) Dion's left hand and right knee? Conversely, if we are happy to admit the existence of entities like the fusion of Dion's left hand and right knee, then why not an empty join?

One way to diagnose the problem is to say that we, as classical mereologists, have confused the algebra of parts with the integrity of the wholes. We have defined the algebra of parts in a combinatorial way, but then at the same time, we tried to make that algebra do ontological work. But this inevitably pulls us in two directions. So, we end up letting the algebraic aspects of our parthood relation do ontological work (creating any fusion whatever), until it goes too far (e.g. the null element), at which point we try to pull back on the ontological reigns.

For another point of tension, consider extensionality. The classicist's axiom says if $x$ and $y$ have the same parts, then $x = y$. This is ontologically conservative: "no difference without a difference maker" (TODO cite Lewis). However, this flattens all structure, and so it judges that "tip" and "pit" cannot be different words, since they have the same parts, after flattening. But that of course feels wrong. These two words have a different ordering of letters, so why would we neglect that in determining their identity? Here too we don't want the combinatorics to do any ontological work, even though we're happy to let the join operation freely generate entities.

We can free ourselves from these sorts of tensions if we separate the algebra of parts from the integrity or gluing of the parts. Let us think of the lattice of parts merely as the abstract "parts space," i.e., as the set of all *possible* combinations of the given parts into larger pieces. Moreover, let us be clear that this does not do any ontological work. A "parts space" is just an abstract description of the various combinations of parts that could be. Think of it as a kind of mold that has slots that could be filled in with actual pieces. To specify an *actual* part-whole complex that occupies that parts space, we need to take a second step and fill in certain of those slots with actual stuff, and specify which of those pieces glue together into bigger pieces.

Once we have made this distinction, we can let the algebra of parts be an algebra, and we can even allow a bottom element without worry. As a component of the abstract parts space, the bottom element is no more a real thing than the join of any other arbitrary regions of the parts space. At the same time, when we specify which pieces really occupy the parts space, we can be as ontologically conservative or as permissive as we like. We have the freedom to provide gluing conditions that are as fine-grained as we need. For instance, we can say that certain pieces glue together, while others do not (e.g., Dion's right knee glues directly to his right femur, but not directly to his left hand). Moreover, we can let the identity conditions be determined by the gluing conditions, and so maintain structured extensionality (a difference in fusions comes from different parts, or different gluings).

Once we make the distinction between the background algebra of parts and the foreground integrity of the fusion, our task takes on a distinctive shape: now we find ourselves trying to coherently glue pieces together over an ambient space. And that is something known well to algebraic geometers and topos theorists: it is the task of constructing a sheaf over a space. For the algebraic geometer and topos theorist, sheaf theory provides a systematic framework for gluing together pieces over a space in such a way that the gluing is done coherently and consistently against the ambient structure of the underlying space. It stands to reason, then, that the mathematician's sheaf-theoretic techniques can be used profitably in mereology.

### 1.2. The Central Thesis

In this paper, we want to build a bridge between sheaf theory and mereology by importing some of those sheaf-theoretic techniques into the mereological setting. The central claim of this paper is thus: part-whole complexes can be usefully modeled as sheaves over locales. The key ideas are as follows:

- An algebra of parts tells us all the ways that parts can combine to form bigger wholes. In this sense, an algebra of parts generates the ambient "parts space" of an object, i.e. the abstract lattice-theoretic structure of *possible* combinations. But not all possible combinations actually glue together to form *actual* fusions. In many cases, we want to allow that only some of the parts glue together into an integrated whole/fusion.
- So, we then require a separate step where we, the mereologists, have to "fill in" the abstract parts space with actual parts: we have to specify which bits of stuff fill in or occupy which slots in that ambient parts space, and we have to specify how those various bits of stuff glue together to form integrated fusions.
- It is tempting to try to model the ambient parts space as a topology. However, topologies have points, and it is not clear that all of the part-whole complexes that we might wish to consider are usefully modeled with points. This limitation is easy to overcome if we generalize and move to the point-free setting: instead of a topology, we choose to model the ambient parts space as a locale (a point-free generalization of a topology).
- We use a sheaf to specify which bits of stuff inhabit an ambient locale and also to stipulate how those bits glue together. A sheaf is precisely an assignment of data to a topology or locale that coherently glues that data together. So, we choose to model the actual part-whole complex as a sheaf over the ambient locale.
- To specify a part-whole complex, then, we (the mereologists) simply need to define a sheaf over the given locale. The "data" that we assign to the ambient locale are the bits of actual stuff that inhabit that parts space, and the gluing condition specifies how those pieces glue together.
- This yields a straightforward procedure that can be used to model any part-whole complex: first, specify the ambient locale, i.e., the abstract space of parts that the part-whole complex in question inhabits; second, fill in that ambient space with actual pieces and say how they glue together; third, let the sheaf framework do the rest of the work. Then the glued sections of the sheaf turn out to be the fusions, whose parts are the smaller sections each fusion is glued from. This is an honest "fusions-first" approach.

There are two important benefits that come along for free when we take this approach.

- The sheaves over a locale form a topos. A topos is a special kind of category that you can do "parts"-like logic in. Indeed, every topos comes equipped with just such an internal logic. It turns out that this internal logic corresponds exactly to the correct mereological logic that governs the part-whole complexes that can be formed over that locale. So, there is no need to manually create a mereological logic to reason about the part-whole complexes that occupy the ambient locale. We get that for free.
- Modalities are natural operators that occur in sheaves, where they are easily defined and managed. These modalities interact correctly with the internal logic of the topos (and in fact are part of that internal logic). So we get mereological modalities for free too.

To our mind, the fact that these benefits come for free offers a compelling reason to adopt a sheaf-theoretic approach to part-whole complexes.

### 1.3. Literature

The literature on mereology is vast. What we might think of as formal mereology (axiomatized systems) goes back at least to Leśniewski's system called "Mereology" [cite] and Leonard and Goodman's "Calculus of Individuals" [cite], along with other contributions by Whitehead [cite], Tarski [cite], and others. Surveys of the resulting literature and ideas can be found in the now standard works by Simons and others (e.g. [1], [2], or [3]).

Topological concepts have been used in mereology for a long time (see the historical coverage and discussion in [1]). So-called "mereotopology" explicitly aims to characterize mereological questions in topological ways, especially using notions like boundaries, interiors, and connectedness. A standard introduction to modern mereotopology is [5].

However, despite its heavy reliance on topology, mereotopology has not (to our knowledge) utilized sheaf-theory in any significant way (nor has classical mereology). TODO: discuss Spivak's behavioral mereology/seven-sketches/temporal type theory ([**?** ], Moltmann's trope sheaves, Moltmann's mereology.

For sheaves, see [6], [7], [8], or [9].

For toposes, see [10], [11], [12], [7], [13]. TODO: discuss the idea of "deriving" the logic from the underlying structure, rather than "inventing" it axiomatically. Perhaps Moltmann's mereology is to be cited here.

TODO: discuss how mereological concepts are used in certain "fusion-first" approaches, e.g., Peter Simons and using "connectedness." Discuss Van Inwagen's special composition question ([14]).

TODO: discuss non-boolean approaches. Discuss boolean algebra stuff from Protow, the survey "Logic in Heyting Algebras," Moltmann's Heyting mereology.

### 1.4. Contributions

The contributions of this paper are as follows:

- We demonstrate a viable "fusions-first" approach to mereology.
- We separate the algebra of parts from the integrity of fusions.
- We build a bridge between mereological techniques of mathematics and philosophy. In particular:
  - We utilize sheaves to systematically manage coherence and gluing over parts spaces.
  - We acquire the correct mereological logics for free from the internal language of the underlying topos.

### 1.5. Plan of the Paper

The plan of this paper is as follows.

- Part 1: Sheaves
  - In Section 2, we introduce the relevant parts of sheaf theory that will be used in the rest of the paper.
  - In Section 3, we define part and whole in sheaf-theoretic terms, and we show how to model different kinds of part-whole complexes as sheaves.
  - In Section 4, we show how mereological modalities arise naturally in sheaves.
  - In Section 5, we discuss what classical mereological notions look like in the sheaf-theoretic setting.
- Part 2: Toposes
  - In Section 6, we introduce the relevant parts of category theory that are needed for topos theory.

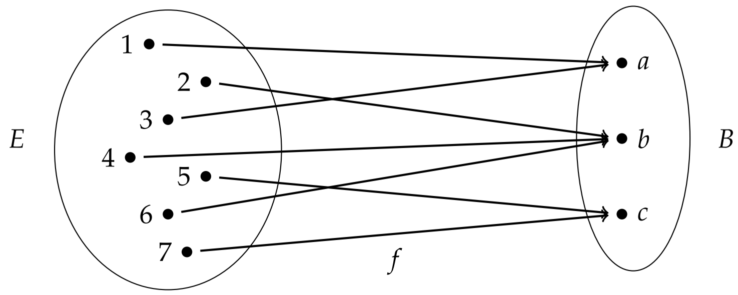
- In Section 7, we introduce topos theory and topos logic.
- in Section 8, we show how to do mereological logic with topos logic.
- In Section 9, we show how the mereological logic of any given setting arises (for free) from the internal logic of the underlying topos.

## 2. Sheaf-Theory

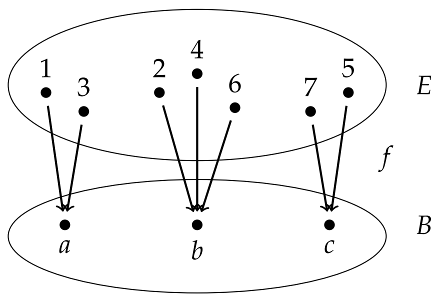
In this section, we cover the parts of sheaf theory that we will utilize in the rest of the paper. Readers familiar with sheaf theory can skip this section.

### 2.1. Fibers

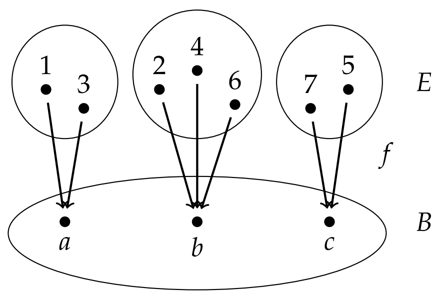
Suppose we have a map (function) $f : E \to B$ that looks something like this:



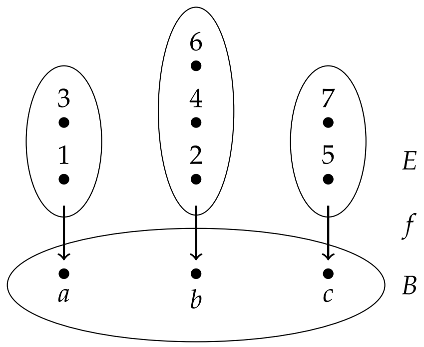

It is sometimes convenient to turn the diagram sideways and group together points in the domain that get sent to the same target, like so:




That makes the pre-images very easy to see. For any point in $B$, its pre-image is just the group of points sitting "over" it:



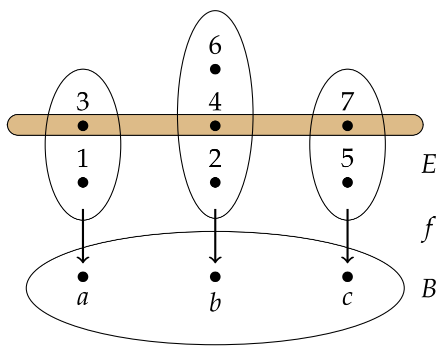

If we stack the points in each pre-image vertically, one on top of the other, we can then think of each pre-image as a kind of "stalk" growing over its base point:



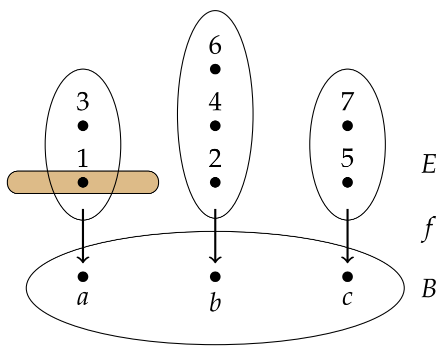

This gives rise to the idea of the "fibers" of a map. The fibers of a map are just its pre-images. For instance, the fiber over $b$ is $\{2, 4, 6\}$.

**Definition 1** (Fibers)**.** *Given a map $f : E \to B$ and a point $y \in B$, the fiber over $y$ is its pre-image $f^{-1}(y) = \{x \mid f(x) = y\}$. $B$ is called the "base space" (or "base" for short) of $f$, while the point $y$ is called the "base point" (or "base" for short) of the fiber.*

We can select a cross-section of one or more fibers by selecting a point from each of the fibers in question. For instance, we can take 3, 4, and 7 as a cross-section of the fibers $f^{-1}(a)$, $f^{-1}(b)$, and $f^{-1}(c)$:




We can also take cross-sections local to only some of the fibers. For instance, we can take 1 as a cross-section just of $f^{-1}(a)$:




**Definition 2** (Sections)**.** *Given a map $f : E \to B$ and a subset $C \subseteq B$ (i.e., a selection of base points in $B$), a section of $f$ (over $C$) is a choice of one element from each fiber over each base $x \in C$.*

**Remark 1.** *Since each point in a fiber amounts to a section over the fiber's base, the elements of a fiber are often just called the "sections" of the fiber.*

### *2.2. Spaces*

In the above examples, the base $B$ was a set. We often want to consider bases that have more structure, e.g., bases that have spatial structure.

In traditional topology, spaces are built out of the points of the space. Given a set of points, a topology on that set specifies which points belong in which regions of the space.

**Definition 3** (Topology)**.** *Let $X$ be a non-empty set, thought of as the "points" of the space. A topology on $X$ is a collection $T$ of subsets of $X$, thought of as the "regions" of the space (called the "open sets" or just the "opens" of $T$), that satisfy the following conditions:*

*(T1) The empty set and the whole set are open:*

$$\varnothing \in T, X \in T.$$

*(T2) Arbitrary unions of opens are open:*

$$\textit{if } \{U_i\}_{i \in I} \subseteq T, \textit{ then } \bigcup_{i \in I} U_i \in T.$$

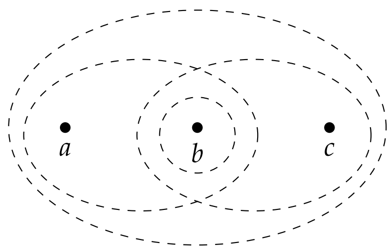
*(T3) Finite intersections of opens are open:*

$$\textit{if } U_1, \ldots, U_n \in T, \textit{ then } \bigcap_{i=1}^{n} U_i \in T.$$

These conditions encode the way that spatial regions are put together. For instance, it ensures that if two regions overlap, then their overlapping area is a region too (and indeed, that's what it means for regions to *overlap*: there's a region of space they have in common).

**Remark 2.** *The regions of a topology form a complete lattice. T1 says that the lattice has a top ($\top$) and bottom ($\bot$), T2 says that for any selection of opens there is a join, and T3 says that for any finite selection of opens there is a meet.*

**Example 1.** *Let $X = \{a, b, c\}$. One possible topology is: $T = \{\varnothing, \{b\}, \{a, b\}, \{b, c\}, \{a, b, c\}\}$. If we draw dashed circles around each of the opens (regions), ignoring the empty set, we get:*




*There are two regions $\{a, b\}$ and $\{b, c\}$ that overlap at $b$ (so $\{b\}$ is a region in $T$ too). There is also the full region $\{a, b, c\}$, which is the union of the smaller regions.*

*We can draw $T$ as a Hasse diagram, which shows that the regions form a lattice:*

$$\begin{tikzcd}[row sep=small, column sep=small]
 & \{a,b,c\} \arrow[dl, dash] \arrow[dr, dash] & \\
\{a,b\} \arrow[dr, dash] & & \{b,c\} \arrow[dl, dash] \\
 & \{b\} \arrow[d, dash] & \\
 & \varnothing &
\end{tikzcd}$$

The lattice structure suggests that much of what is important about a space is not so much its points, but rather its opens/regions. This leads to the idea that topology-like reasoning can be done without the points. So, we can generalize: take a topology, and drop the points. That leaves just the opens/regions, which we call a frame (or locale).

**Definition 4** (Frames/locales). *A frame (synonymously, a locale) $\mathbb{L}$ is a partially ordered set $L$ (we call its elements "opens" or "regions") that satisfies the following conditions:*

*(L1) $L$ is a complete lattice:*

- *Every subset $S \subseteq L$ has a join, denoted $\bigvee S$.*
- *Every finite subset $F \subseteq L$ has a meet, denoted $\bigwedge F$.*

*(L2) Finite meets distribute over arbitrary joins:*

$$a \wedge \bigwedge_{i \in I} b_i = \bigwedge_{i \in I} (a \wedge b_i), \text{ for all } a \in L \text{ and all families } \{b_i\}_{i \in I} \subseteq L.$$

*Define $V \preccurlyeq U$ (read "$V$ is included in $U$") by $a = a \wedge b$.*

**Remark 3.** *The fact that $V \preccurlyeq U$ is equivalent to $a = a \wedge b$ means we can deal with the opens of a frame algebraically (via $\wedge$ and $\vee$ operations), or order-theoretically (via the $\preccurlyeq$ relation), whichever is more convenient.*

**Remark 4.** *The category of locales is defined as the dual/opposite of the category of frames (see* **??** *below), and so frames and locales are quite literally the very same objects. In practice, frames are often used for algebraic purposes, and locales are used for (generalized) spatial purposes. Here, we will have no reason to distinguish these two roles, and so we will use the names "frame" and "locale" interchangeably.*

### *2.3. Presentations of locales*

Locales have presentations much like groups and other algebraic structures. To give the presentation of a locale, specify a set of generators and relations.

**Definition 5** (Presentations). *A presentation $\langle G, R \rangle$ of a locale $\mathbb{L}$ is comprised of:*

*(P1) A set of generators $G = \{U_k, U_m, \ldots\}$.*
*(P2) A set of relations $R \subseteq G \times G$ on those generators.*

*The locale $\mathbb{L}$ presented by $\langle G, R \rangle$ is the smallest one freely generated from $G$ which satisfies $R$.*

To calculate the locale that corresponds to a presentation, start with the generators, then take all finite meets and all arbitrary joins that satisfy $R$ (and of course L1 and L2).

**Example 2.** *Let a locale $\mathbb{L}$ be given by the presentation $\langle G, R \rangle$ where:*

- $G = \{\perp, U_1, U_2, U_3\}$.
- $R = \{\perp \preccurlyeq U_1, U_1 \preccurlyeq U_2, U_1 \preccurlyeq U_3\}$.

*There are four generators ($\perp$, $U_1$, $U_2$, and $U_3$), and $\perp$ is below $U_1$ while $U_1$ is a sub-region of $U_2$ and $U_3$. Since $U_1$ is a sub-region of both $U_2$ and $U_3$, $U_1$ is their meet:*

- $U_1 = U_2 \wedge U_3$.

*At this point, we have generated this much of the locale:*

$$
\begin{tikzcd}
U_2 & & U_3 \\
& U_1 \arrow[ul, dash] \arrow[ur, dash] & \\
& \perp \arrow[u, dash] &
\end{tikzcd}
$$

*$R$ says nothing to constrain joins, so we need to join everything we can. In this case, we need to join $U_2$ and $U_3$:*

$$
\begin{tikzcd}
& U_2 \vee U_3 & \\
U_2 \arrow[ur, dash] & & U_3 \arrow[ul, dash] \\
& U_1 \arrow[ul, dash] \arrow[ur, dash] & \\
& \perp \arrow[u, dash] &
\end{tikzcd}
$$

*There are no further joins or meets that aren't already represented in the picture. For instance, all further non-trivial meets are already accounted for:*

- $U_1 \wedge \perp = \perp$.
- $U_2 \wedge U_1 = U_1$ *and* $U_3 \wedge U_1 = U_1$.
- $U_2 \wedge \perp = \perp$ *and* $U_3 \wedge \perp = \perp$.
- $(U_2 \vee U_3) \wedge U_2 = U_2$ *and* $(U_2 \vee U_3) \wedge U_3 = U_3$.
- $(U_2 \vee U_3) \wedge U_1 = U_1$.
- $(U_2 \vee U_3) \wedge \perp = \perp$.

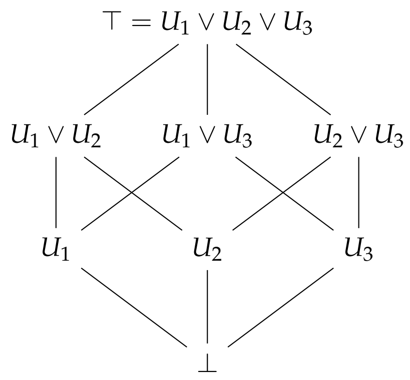
*Similarly, all other non-trivial joins are also already accounted for:*

- $\perp \vee U_1 = U_1$.
- $\perp \vee U_2 = U_2$ *and* $\perp \vee U_3 = U_3$.
- $\perp \vee (U_2 \vee U_3) = U_2 \vee U_3$.
- $U_1 \vee U_2 = U_2$ *and* $U_1 \vee U_3 = U_3$.
- $U_2 \vee (U_2 \vee U_3) = U_2 \vee U_3$ *and* $U_2 \vee (U_3 \vee U_3) = U_2 \vee U_3$.

**Example 3.** *Let $\mathbb{L} = \langle G, R \rangle$ be given by:*

- $G = \{U_1, U_2, U_3\}$.
- $R = \varnothing$.

*We have three generators ($U_1$, $U_2$, and $U_3$), and there are no relations restricting how those generators are related. Thus, the locale that is freely generated from this presentation is isomorphic to the power set of three elements:*




A presentation provides the most "minimal" information from which the rest of the locale is generated. It might be tempting to think that each generator is an atomic region, but that is not quite right. Some generators are reducible to others.

**Definition 6** (Meet-irreducibility)**.** *Given a presentation $\mathbb{L} = \langle G, R \rangle$, a region $U \in G$ is meet-reducible ("reducible" for short) if it is the non-trivial meet of other regions:*

$$U = W \wedge V \implies (V = U \text{ or } W = U).$$

*$U$ is meet-irreducible ("irreducible" for short) if it is not meet-reducible.*

Intuitively, a generator is reducible if it can be expressed as the meet of strictly larger regions, which occurs exactly when it is their overlap.

**Example 4.** *Take the locale from Example 2. $U_1$ is the overlap of $U_2$ and $U_3$, and $U_2$ and $U_3$ are strictly larger regions than $U_1$, so $U_1$ is reducible.*

*By contrast, $U_2$ and $U_3$ are irreducible, because they cannot be expressed as the meet of two strictly larger regions. Similarly, $\bot$ is irreducible, because it is not the meet of two strictly larger regions either (it is the meet of only one strictly larger region, namely $U_1$).*

We can see the minimal irreducible generators of a locale as its atomic regions.

**Definition 7** (Atomic regions)**.** *Given a presentation $\mathbb{L} = \langle G, R \rangle$, define the atomic regions of $\mathbb{L}$, denoted $Atoms(\mathbb{L})$, as the minimal irreducible generators, i.e. those generators $g \in G$ that satisfy the following two conditions:*

*(A1)* Meet-irreducibility. *$g$ is meet-irreducible.*
*(A2)* Minimality. *There is no strictly smaller meet-irreducible $h$ with $h \preccurlyeq g$.*

### *2.4. Presheaves*

Above we considered the fibers of a map $f : E \to B$, where $E$ and $B$ were sets. We can also consider fibers over locales, where the fibers respect the locale's structure. This is called a presheaf. A presheaf is an assignment of data to each of a locale's regions that is "stable under restriction," i.e., that respects zooming in and out.

**Definition 8** (Presheaf)**.** *Let $\mathbb{L}$ be a locale, and let $Morphs(\mathbb{L})$ be $\{\langle A, B \rangle \mid A \preccurlyeq B \in \mathbb{L}\}$. A presheaf on $\mathbb{L}$ is a pair $\langle F, \{\rho^B_A\}_{\langle A,B \rangle \in Morphs(\mathbb{L})} \rangle$, where:*

- *$F$ assigns to each region $U \in L$ some data $F(U)$.*
- *$\{\rho^B_A\}_{\langle A,B \rangle \in Morphs(\mathbb{L})}$ is a family of maps $\rho^B_A : F(B) \to F(A)$ (called "restriction maps"), each of which specifies how to restrict the data over $F(B)$ down to the data over $F(A)$.*

*All together,* $\langle F, \{\rho_A^B\}_{\langle A,B\rangle \in Morphs(\mathbb{L})}\rangle$ *must satisfy the following conditions:*

*(R1) Restrictions preserve identity:*

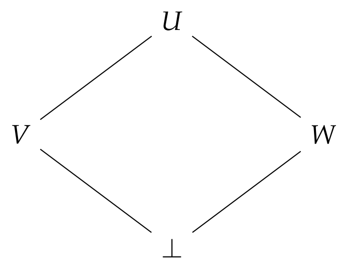
$$\rho_U^U = id_U \text{ (the identity on } U\text{), for every } U \in \mathbb{L}.$$

*(R2) Restrictions compose:*

$$\text{If } A \preccurlyeq B \text{ and } B \preccurlyeq C, \text{ then } \rho_A^C = \rho_A^B \circ \rho_B^C.$$

Since $F$ assigns data $F(U)$ to each region $U \in \mathbb{L}$, we can think of the $F(U)$s as the "fibers" over $\mathbb{L}$, and the restriction maps as "zoom in" maps that go from bigger fibers down to smaller fibers.

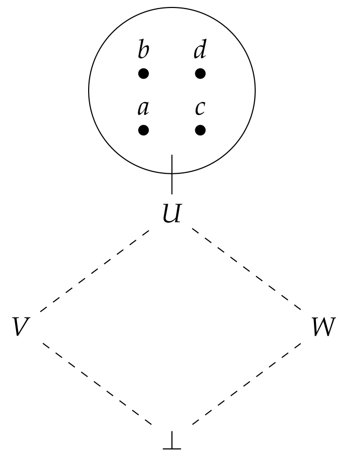
**Example 5.** *Let* $\mathbb{L}$ *be a locale* $\{\perp, W, V, U\}$ *with the following structure:*



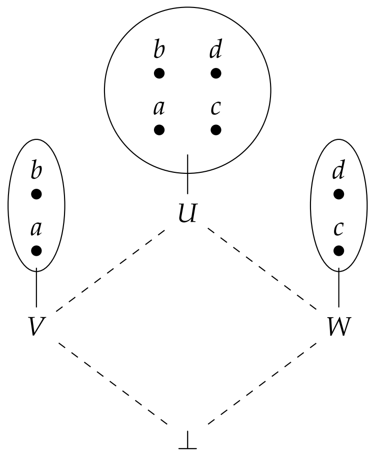

*Next, let's define a presheaf F as follows:*

- $F(U) = \{a, b, c, d\}$, $F(V) = \{a, b\}$, $F(W) = \{c, d\}$, $F(\perp) = \{*\}$.
- *Define* $\rho_V^U$ *as the projection (send a to a, b to b, and the rest can go anywhere), and similarly for* $\rho_W^U$*. Let* $\rho_\perp^U$*,* $\rho_\perp^V$*, and* $\rho_\perp^W$ *send their data to* $\{*\}$*, and let the rest be identities.*

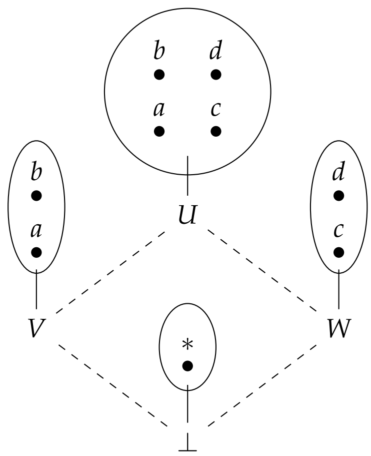
*We can see F's assignments as fibers over* $\mathbb{L}$ *by drawing them over the regions they are assigned to. For instance, over U we have* $F(U)$*, i.e.,* $\{a, b, c, d\}$*:*



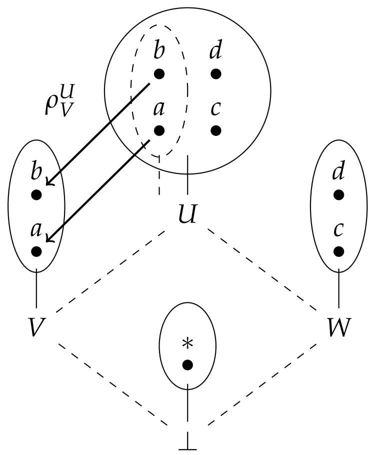

*Similarly, over V and W, we have* $F(V) = \{a, b\}$ *and* $F(W) = \{c, d\}$*:*



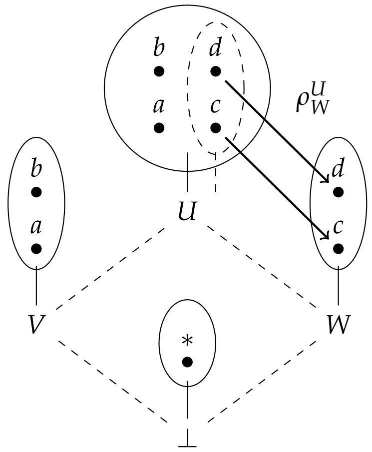

*Finally, over $\perp$, we have a singleton set:*



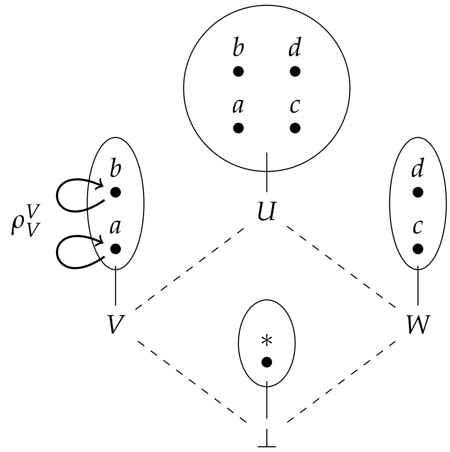

*The restriction maps show how to "zoom in" on the parts (sub-fibers) of any given fiber. For instance, we can see that the fiber over $V$ is contained in the fiber over $U$. The restriction map just projects that sub-fiber out, thereby showing us how to "zoom in" on that sub-fiber:*



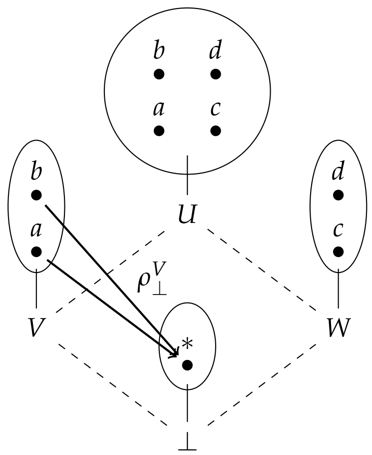

*It's similar for the fiber over $W$:*



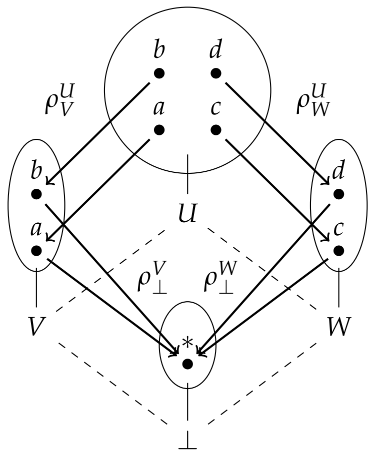

*Restricting a fiber to itself is just the identity on the fiber:*




*The other restriction maps restrict down to the singleton set. For instance:*




*All of this makes it clear that the structure of the presheaf data that sits in the fibers over $\mathbb{L}$ mimics (respects) the structure of the base locale:*




**Remark 5.** *In line with Remark 1, the elements of each fiber $F(U)$ are usually just called the "sections" of $F(U)$. For instance, $c$ is a section of $F(W)$, as is $d$.*

### *2.5. Sheaves*

The definition of a presheaf requires only that the data be stable under restriction (zooming in on a region). It does not require that the data fit together across different regions (fibers).

A sheaf is a presheaf with an added gluing condition: whenever you have compatible data on overlapping fibers, there must be a unique way to glue it together into data over the union. In other words, the data in the fibers must agree on overlap and combine coherently.

To get at this idea, let's first define a cover. A cover of a region $U$ is a selection of sub-regions that covers $U$ in its entirety. The chosen sub-regions don't leave any part of $U$ exposed.

**Definition 9** (Cover)**.** *Let $\mathbb{L}$ be a topology or a locale, and let $U$ be a region of $\mathbb{L}$. A cover of $U$ is a family $\{U_i\}_{i\in I} \subseteq \mathbb{L}$ such that:*

$$U = \bigvee_{i\in I}\{U_i\}.$$

*In other words, a cover of $U$ is a family of regions that join together to form $U$.*

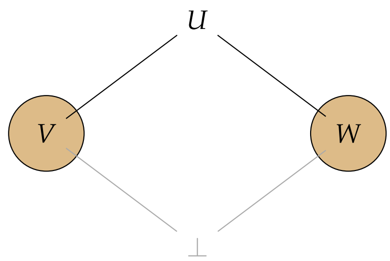
**Example 6.** *Take the topology from Example 1: $T = \{\varnothing, \{b\}, \{a,b\}, \{b,c\}, \{a,b,c\}\}$. A cover of $\{a,b,c\}$ is $\{a,b\}$ and $\{b,c\}$, because altogether, $\{a,b\}$ and $\{b,c\}$ cover all of the points in $\{a,b,c\}$.*

*Another cover of $\{a,b,c\}$ is $\{\{a,b\},\{b,c\},\{b\}\}$. Although $\{b\}$ is redundant here, this choice of sub-regions still entirely covers $\{a,b,c\}$ as required.*

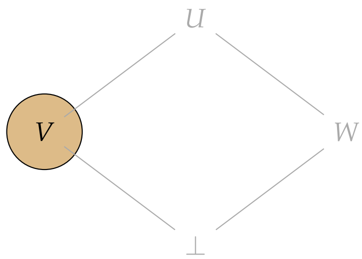
**Example 7.** *In the context of frames, where there are no points, a cover of $U$ is just a selection of sub-regions of $U$ that together join together to form $U$. Take the locale from Example 5:*

$$\begin{tikzcd}
 & U \arrow[dl, dash] \arrow[dr, dash] & \\
V \arrow[dr, dash] & & W \arrow[dl, dash] \\
 & \perp &
\end{tikzcd}$$

*A cover of $U$ is $\{V, W\}$, since $U = \bigvee\{V, W\}$:*




*A cover of V is just $\{V\}$:*



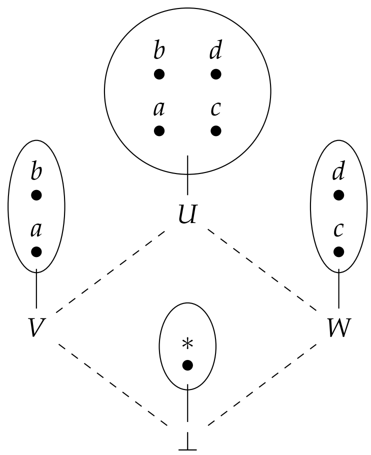

**Remark 6.** *A cover over the least element of a locale (or a topology) is empty (the empty set), because there are no regions (or points) to cover.*

Given a presheaf $F$ over a locale $\mathbb{L}$, if we have a cover $\{U_i\}_{i \in I}$ of some portion of $\mathbb{L}$, there is a corresponding family of fibers $\{F(U_i)\}_{i \in I}$ over that cover. We can pick one section (i.e., one element) from each such fiber to get a slice of elements that spans all of the fibers over that cover. Let us call such a choice a selection of patch candidates.

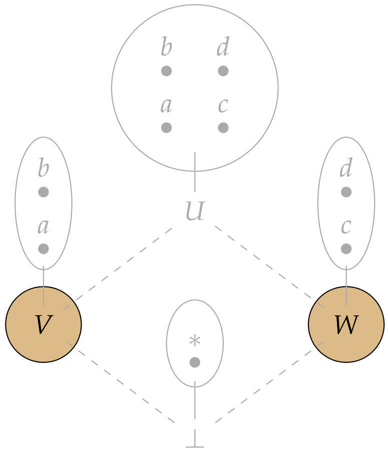
**Definition 10** (Patch candidates)**.** *Given a presheaf F and a cover $\{U_i\}_{i \in I}$ with a corresponding family of fibers $\{F(U_i)\}_{i \in I}$, a selection of patch candidates $\{s_i\}_{i \in I}$ is a choice of one section $s_i$ from each $F(U_i)$:*

$$\{s_i\}_{i \in I} = \{s_i \mid s_i \in F(U_i) \textit{ for each } F(U_i) \in \{F(U_i)\}_{i \in I}\}.$$

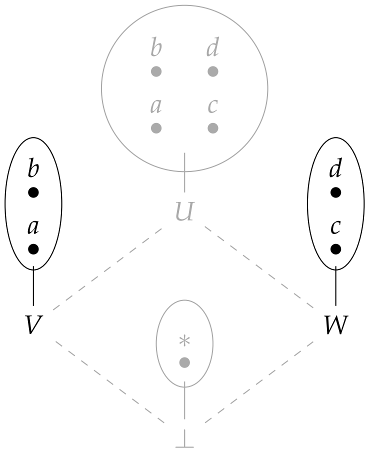
**Example 8.** *Take the presheaf from Example 5:*



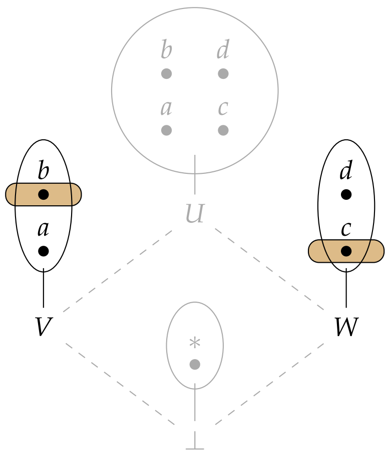

*Let $\{V, W\}$ be the cover of interest:*




*Over this cover, we have a corresponding family of fibers:*




*A selection of patch candidates is a choice of one section (element) from each fiber. For instance, we might pick b from $F(V)$ and c from $F(W)$:*




*Similarly, we might pick $\{a, d\}$, $\{b, d\}$, or $\{a, c\}$, each of which is a valid selection of patching candidates.*

**Remark 7.** *Consider the empty cover. Since there are no sub-regions below the least element of a locale, there are no patch candidates we could choose for the empty cover either. Hence, any selection of patch candidates for the empty cover is $\varnothing$.*

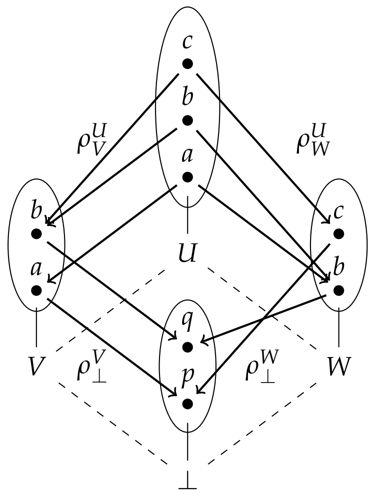
A selection of patch candidates might fit together, or they might not. We say they are compatible if they fit together, i.e., if they agree on overlaps. To check this, take any pair of patch candidates, and check if they restrict to the same data on their overlap.

**Definition 11** (Compatible patch candidates)**.** *Given two fibers $F(U_i)$ and $F(U_j)$ and a patch candidate from each, $s_i \in F(U_i)$ and $s_j \in F(U_j)$, $s_i$ and $s_j$ are compatible if they restrict to the same data on their overlap $U_i \wedge U_j$:*

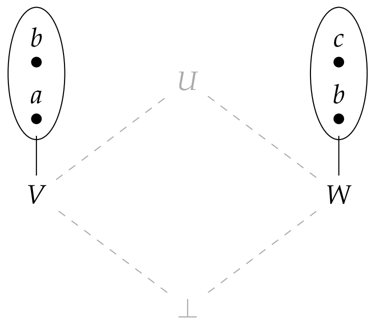
$$\rho^{U_i}_{U_i \wedge U_j}(s_i) = \rho^{U_j}_{U_i \wedge U_j}(s_j).$$

*A selection of patch candidates $\{s_i\}_{i \in I}$ is compatible if all of its members are pair-wise compatible.*

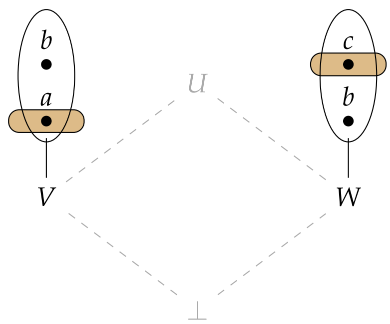
**Example 9.** *Consider the following presheaf $F$:*



*Take the cover $\{V, W\}$ and its corresponding fibers:*

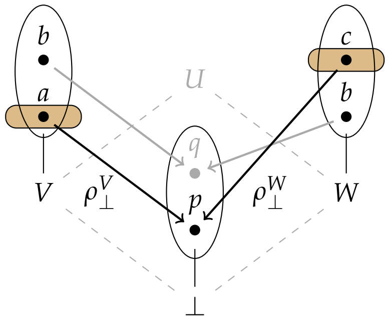


*Suppose we pick $\{a, c\}$ for patch candidates:*

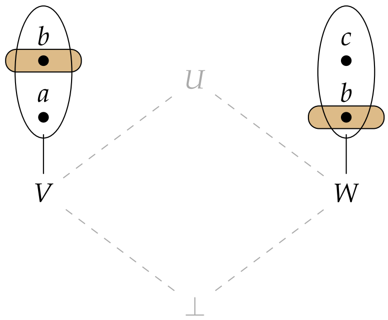


*Is this selection compatible? We have to check if they agree on their overlap. The overlap $V \wedge W$ is $\perp$. Where does $\rho^V_\perp$ send our chosen patch candidate $a$? It sends it to $p$, since $\rho^V_\perp(a) = p$.*

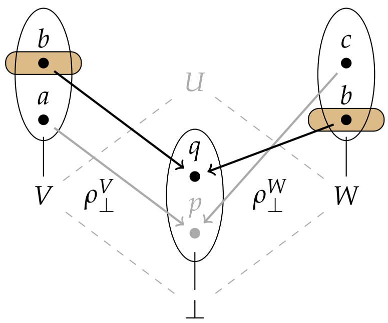
*Where does $\rho^W_\perp$ send our other chosen patch candidate $b$? It also sends it to $p$, since $\rho^W_\perp(c) = p$. On the overlap $\perp$ then, $\rho^V_\perp(a) = \rho^W_\perp(c)$, so $a$ and $c$ are compatible. This is easy to see in the diagram, since $a$ and $b$ both get sent to the same place:*



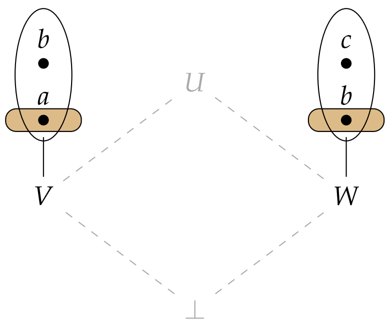

*Now suppose we pick $\{b, b\}$ for patch candidates:*



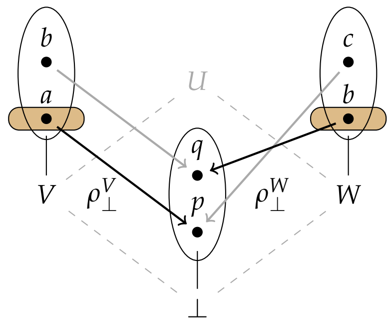

*These are also compatible. They agree on their overlap (both restrict to $q$):*




*Finally, suppose we pick $\{a, b\}$ for patch candidates:*




*These are not compatible. They do not agree on their overlap:*




**Remark 8.** *Consider the empty cover. Since any selection of patch candidates for the empty cover is empty, compatibility is satisfied vacuously.*

*As an analogy, if you ask your class to turn off all cell phones but nobody brought a cell phone to glass, then your request is satisfied vacuously: there is simply nothing that needs to be done to make it happen. It's similar with the empty cover: since there are patch candidates to check, compatibility is achieved vacuously.*

If a selection of patch candidates $s_i, \ldots, s_k$ across a cover of $U$ is compatible, we say those patches glue together if if there's a section $s$ in $F(U)$ that restricts down to exactly those patches.

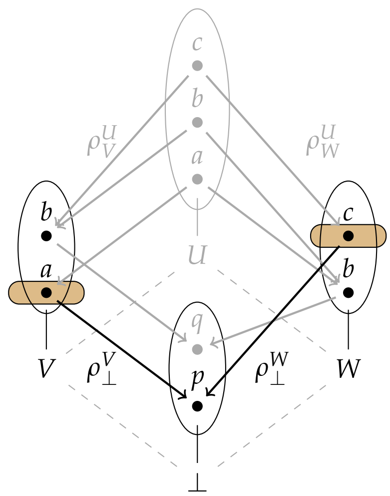
**Definition 12** (Gluing). *Given a presheaf F and a selection of compatible patch candidates $\{s_i\}_{i \in I}$ for a cover $\{U_i\}_{i \in I}$, $\{s_i\}_{i \in I}$ glue together only if there is a section $s \in F(U)$ that restricts down to $s_i$ on each fiber $F(U_i)$ of the cover, i.e., only if s is such that:*

$$\rho^U_{U_i}(s) = s_i, \text{ for each } i \in I.$$

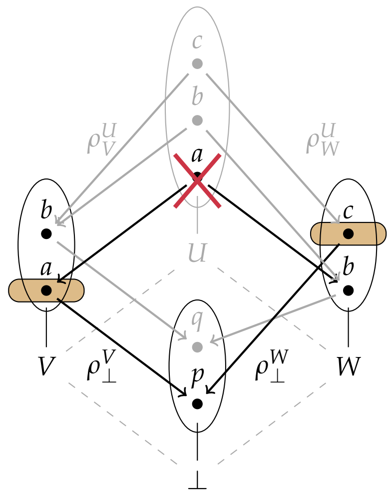
*As a matter of terminology, if a section $s \in F(U)$ is glued from patches $\{s_i\}_{i \in I}$, we say that s is a global section of the cover, and each $s_i$ is a local section of the cover. We may also say variously that s is* glued from *those patches, that s is* composed *of those pathes, that those patches* compose *s, or that gluing those patches* yields *s.*

*A selection of patches $\{s_i\}_{i \in I}$ glues uniquely if there is one and only one such section $s \in F(U)$ that is glued from them.*

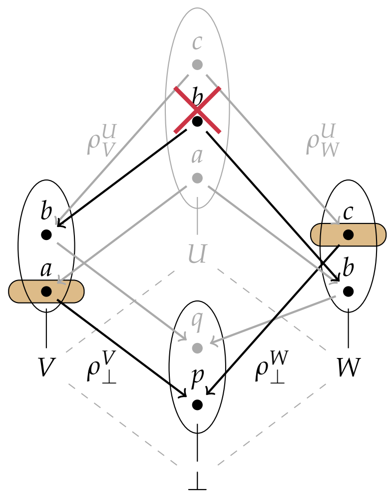
**Example 10.** *Take the presheaf from Example 9, and consider the cover $\{V, W\}$ again. Take the the selection of patches $\{a, c\}$, which are compatible because they agree on overlap:*



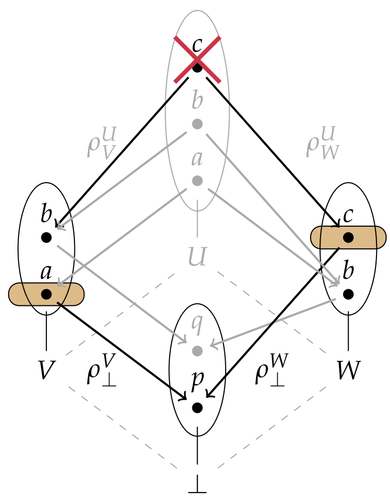
*Even though $a$ and $c$ are compatible, they do not glue together, because there is no section in $F(U)$ that restricts down to them. Consider $a \in F(U)$ first. It restricts to $a \in F(V)$ on the left, but it does not restrict to $c \in F(W)$ on the right:*




*As for $b \in F(U)$, it restricts to neither $a \in F(V)$ on the left nor $c \in F(W)$ on the right:*

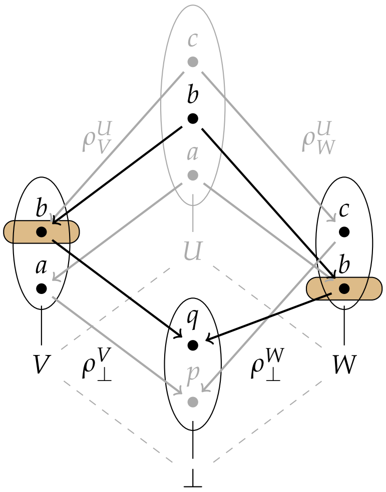



*Finally, $c \in F(U)$ restricts to $c \in F(W)$ on the right, but not to $a \in F(V)$ on the left:*



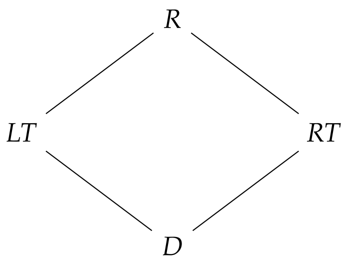

*Thus, none of $a$, $b$, or $c$ in $F(U)$ are glued from $\{a, c\}$, because none of them decompose into $a$ on the left and $c$ on the right.*

*Now suppose we pick $\{b, b\}$ for patch candidates. These do glue together (trivially), because there is a section in $F(U)$ (namely $b \in F(U)$) that restricts down to $b \in F(V)$ on the left and $b \in F(W)$ on the right:*




**Example 11.** *Consider an example that glues together behaviors. Imagine a toy robot that looks something like a small tank: it has tracks on the left and right sides, and the two tracks are connected by a single drive controller. The controller either drives at a constant speed, or it sits idle. When it drives, it turns both tracks at the same speed.*

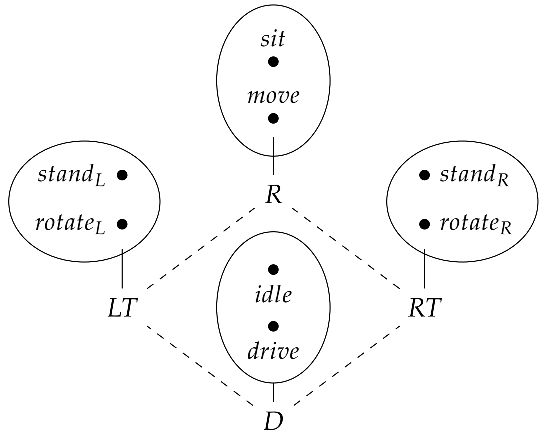
*Let's represent the robot as a locale. Let $LT$ and $RT$ be the left and right track assemblies respectively, let $D$ be the drive controller that is shared by $LT$ and $RT$, and let $R$ be the whole robot (the join of $LT$ and $RT$). As a picture:*




*For a presheaf, let's assign to each region the behaviors that are locally observable at that region:*

- *The drive controller $D$ can either drive (turn) or sit idle.*
- *The left track assembly can each either rotate$_L$ or stand$_L$ still.*
- *The right track assembly can also either rotate$_R$ or stand$_R$ still.*
- *The entire robot can either move (forward) or sit stationary.*

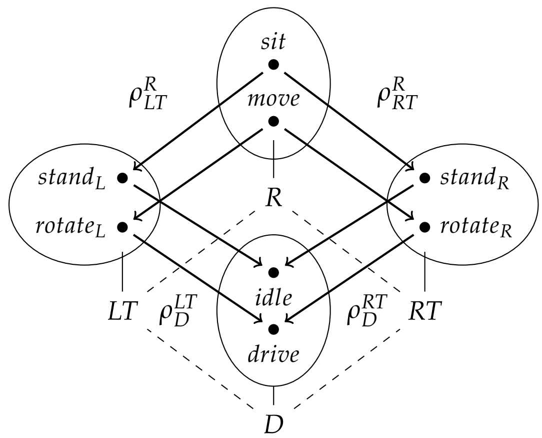
*In a picture:*




*For the restriction maps, let's say that they restrict the observable behavior of a larger region to the observable behavior of the smaller region. For instance, if you are observing the whole robot moving forward (move), and you then "zoom in" on the left track assembly, you'll see those tracks rotating (rotate$_L$).*

- $\rho^{R}_{LT}(sit) = stand_L$, $\rho^{R}_{LT}(move) = rotate_L$.
- $\rho^{R}_{RT}(sit) = stand_R$, $\rho^{R}_{RT}(move) = rotate_R$.
- $\rho^{LT}_{D}(stand_L) = idle$, $\rho^{LT}_{D}(rotate_L) = drive$.
- $\rho^{RT}_{D}(stand_R) = idle$, $\rho^{RT}_{D}(rotate_R) = drive$.

*In a picture:*




*Now take the cover $\{LT, RT\}$ of R. The patch candidates $\{rotate_L, rotate_R\}$ are compatible, because they agree on overlap (they both restrict down to drive). But they also glue uniquely, yielding move. In other words, the robot's forward motion is patched together precisely from the two pieces of its cover, namely the left tracks rotating (rotate$_L$) and the right tracks rotating (rotate$_R$).*

*Similarly, the Robot's sit behavior is also glued from the two pieces of its cover, namely the left track assembly standing still (stand$_L$) and the right track assembly standing still (stand$_R$).*

*Thus, there are two global sections of R's behavior: moving forwards (patched together from its left and right motions), or standing still (patched together from its left and right lack of motion).*

We can now state what it is to be a sheaf. A sheaf is a presheaf that satisfies a special gluing condition: namely, that for every cover, every compatible selection of patch candidates glues together uniquely.

**Definition 13** (Sheaf)**.** *A presheaf F is a sheaf iff it satisfies the following condition (called "the gluing condition"):*

*(G0) For every cover $\{U_i\}_{i \in I}$ of a region $U$ and every selection of patch candidates $\{s_i\}_{i \in I}$ for that cover, if $\{s_i\}_{i \in I}$ can glue, then $\{s_i\}_{i \in I}$ glue to yield a unique $s \in F(U)$.*

**Remark 9.** *There is a subtlety regarding what sheaves look like over the least element of a locale. Note that the gluing condition is formulated as an implication. That is to say, it says that, for every cross-section of patch candidates,* if *that cross-section can glue,* then *it glues in exactly one way.*

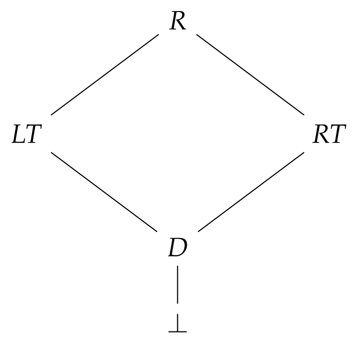
*Next, consider the fact that the cover over the least region of a locale is an empty cover. Since there are no patch candidates that need to be checked for compatibility, there is nothing that needs to be done to get a "selection of gluable patch candidates." Hence, the antecedent of the gluing condition is satisfied vacuously over the least element of the locale.*

*But since the empty cover satisfies the antecedent of the gluing condition vacuously, it follows that if a presheaf is to qualify as a sheaf, it must ensure that the consequent is satisfied over the empty cover as well. In other words, it must assign a unique glued section (a singleton set) to the least region of the locale. So, even though a* presheaf *may assign a larger set of data to the least element of a locale, a* sheaf *always assigns a singleton to that region.*

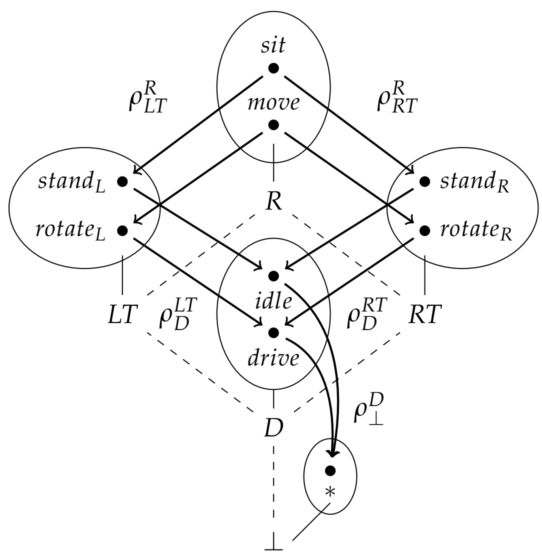
**Example 12.** *The presheaf from Example 9 fails to be sheaf, because as we saw in Example 10, there is a compatible selection of patch candidates (namely, $\{a, c\}$) which fails to glue. To be a sheaf, every compatible selection of patch candidates must glue.*

**Example 13.** *The presheaf from Example 11 fails to be a sheaf, because it does not assign a singleton to the lowest region of the underlying locale. In that example $D$ is the lowest region of the locale, and $F(D) = \{p, q\}$, a set containing two elements. Hence, $F$ fails to be a sheaf.*

*However, suppose we add a distinct bottom element to the locale:*




*If we assign a singleton set to $\perp$ (so that $F(\perp) = \{*\}$, say), then $F$ looks like this:*



*This modification ensures that $F$ qualifies as a sheaf, since it ensures that* all *gluable selections of patch candidates (including the empty one) glue uniquely.*

### 2.6. A Canonical Sheaf Construction

Not every presheaf is a sheaf, since some presheaves fail the gluing condition. However, there is a canonical procedure called "sheafification" that turns any presheaf into a sheaf. To sheafify a presheaf, add any missing sections that glue, then quotient sections that are locally indistinguishable. The result is guaranteed to be a sheaf, by construction.

For our purposes, there is a simplified version of sheafification that we can use to construct sheaves that model part-whole complexes in a natural way. Given a presentation of a locale, the recipe to build such a sheaf over it goes like this:

1. Assign local data to atomic regions.
2. Specify a gluing condition.
3. Recursively glue more and more pieces together until you can't glue any more.

Let's make this more precise. Given a presented locale, we can uniquely write each region as the join of its atomic regions.

**Definition 14** (Atomic indices)**.** *Let $\mathbb{L} = \langle G, R\rangle$ be a presented locale with $G = \{U_1, \ldots, U_n\}$. Let $\mathbb{A} \subseteq \{0, \ldots, n\}$ be the indices of the atomic regions of $G$.*

*For any $U \in \mathbb{L}$, define its atomic support (denoted $I(U)$, or just $I$ for short) as:*

$$I(U) = \{i \in \mathbb{A} \mid U_i \preccurlyeq U\}$$

*Then $U$ can be written canonically as $U_{I(U)}$, the join of its atomic supports:*

$$U_{I(U)} = \bigvee_{i \in I(U)} U_i.$$

A gluing condition is a family of predicates that say when a selection of patch candidates glue.

**Definition 15** (Gluing condition)**.** *A gluing condition $\mathcal{G}$ is a family of predicates*

$$\mathcal{G}_U : \prod_{i \in I(U)} F(U_i) \to \{\text{true}, \text{false}\},$$

*one for each region $U \in \mathbb{L}$, that collectively satisfy the following coherence conditions:*

*(G1)* Local data is glued. *If $U_k \in Atoms(\mathbb{L})$ and $F(U_k) = \{\langle b_k\rangle\}$, then*

$$\mathcal{G}_{U_k}(\langle b_k\rangle) = \text{true}.$$

*(G2)* Downward stability. *If $\mathcal{G}_U(\langle b_i\rangle_{i \in I(U)}) =$ true, then for each $V \preccurlyeq U$, we must have:*

$$\mathcal{G}_V(\rho^U_V(\langle b_i\rangle_{i \in I(U)})) = \text{true}.$$

*(G3)* Upward stability. *Given a selection of patch candidates $\langle b_i\rangle_{i \in I(U)}$, if $\mathcal{G}_{U_i \vee U_j}(\langle b_i, b_j\rangle) =$ true for each $i, j \in I(U)$, then we must have:*

$$\mathcal{G}_U(\langle b_i\rangle_{i \in I(U)}) = \text{true}.$$

A sheaf can be generated from a gluing condition by starting with some local data on the atomic regions and then gluing all pieces together that satisfy the gluing condition.

**Definition 16** ($\mathcal{G}$-sheaves)**.** *Given a gluing condition $\mathcal{G}$ and local data $F(U_k) = \{\langle b_k \rangle\}$ for each atomic region $U_k$, define for each region $U$:*

$$F(U) = \{\langle b_i \rangle_{i \in I(U)} \in \prod_{i \in I(U)} F(U_i) \mid \mathcal{G}_U(\langle b_i \rangle_{i \in I(U)}) = \text{true}\}.$$

*For $V \preccurlyeq U$, define the restriction map*

$$\rho^U_V : F(U) \to F(V) \quad \textit{as} \quad \rho^U_V(\langle b_i \rangle_{i \in I(U)}) = \langle b_i \rangle_{i \in I(V)}.$$

*Set $F(\bot) = \{\langle\rangle\}$, the empty tuple.*

**Remark 10.** *Alternatively, given some local data and a gluing condition, define a presheaf over the given locale, call it $F_\wp$, that assigns all combinations of local data to each region:*

$$F_\wp(U) = \prod_{i \in I(U)} F(U_i).$$

*Then filter by the gluing condition. That produces the same sheaf.*

We must check that Definition 16 really defines a sheaf.

**Theorem 1** ($\mathcal{G}$-sheaves are presheaves)**.** *Given a gluing condition $\mathcal{G}$ and an assignment of local data to the atomic regions of the underlying locale, the corresponding $\mathcal{G}$-sheaf is a presheaf.*

**Proof.** We must show that restrictions preserve identities and composition.

- *Identities.* $\rho^U_U$ projects to the same index set, so $\rho^U_U = id_{F(U)}$.
- *Composition.* $\rho^U_V$ restricts to the fiber over $V$, and $\rho^V_W$ restricts to the fiber over $W$, so $\rho^V_W \circ \rho^U_V = \rho^U_W$.

We must also show that the restrictions are well defined.

- For $V \preccurlyeq U$, if $\langle b_i \rangle_{i \in I(U)} \in F(U)$, then by (G2) $\langle b_i \rangle_{i \in I(V)}$ satisfies $\mathcal{G}_V$, so $\rho^U_V$ is well-defined. □

**Theorem 2** ($\mathcal{G}$-sheaves are sheaves)**.** *Given a gluing condition $\mathcal{G}$ and an assignment of local data to the atomic regions of the underlying locale, the corresponding $\mathcal{G}$-sheaf satisfies the gluing condition G0.*

**Proof.** We must show that every gluable selection of patch candidates $\langle b_i \rangle_{i \in I(U)}$ glues to yield a unique section in $F(U)$. Assume that we have a compatible selection of patch candidates $\langle b_i \rangle_{i \in I(U)}$. Then:

- Existence: we assumed the patch candidates are compatible. By (G3) then, $\langle b_i \rangle_{i \in I(U)} \in F(U)$.
- Uniqueness: let $s = \langle b_i \rangle_{i \in I(U)} \in F(U)$. If another section $t = \langle b_i \rangle_{i \in I(U)} \in F(U)$ were glued from the same components, then $s = t$, since both restrict to the same supports. □

Throughout the rest of this paper, we will use $\mathcal{G}$-sheaves to model part-whole complexes, but that is only for simplicity of exposition. Any sheaf over a locale would do just as well.

## 3. Modeling Part-Whole Complexes as Sheaves

As noted in Section 1, the central claim of this paper is that we can model part-whole complexes as sheaves over locales. In particular, the locale provides the abstract parts space of "regions" the pieces can occupy, the sheaf assigns actual pieces to those regions, and the gluing condition determines when pieces fuse.

We can thus define the core mereological concepts of part and whole in sheaf-theoretic terms. Regarding wholes, we can identify fusion with gluing: to say that some pieces fuse or form a "fusion" is just to say that they are glued together. Regarding parts, to say that a piece is a "part" is just to say that it is a part of a fusion. In other words, the parts of a fusion are just the pieces from which it is glued together.

**Definition 17** (Fusions and parts)**.** *We say that a section $s \in F(U)$ is a fusion iff*

$$\mathcal{G}_U(s) = \text{true}.$$

*Given $t \in F(V)$ and $s \in F(U)$ with $V \preccurlyeq U$ and $V \neq \bot$, we say $t$ is a part of $s$, denoted $t \sqsubseteq s$, iff*

$$\mathcal{G}_U(s) \quad \textit{and} \quad \rho^U_V(s) = t.$$

**Remark 11.** *$V \neq \bot$ because no parts can occupy $\bot$. The least region of the locale represents the combinatorial idea of no regions at all, and so it cannot be populated by any parts (hence in a $\mathcal{G}$-sheaf the sole section over $\bot$ is the singleton $\langle\rangle$).*

Sheaf theory thus provides a systematic framework with which to model a large variety of part-whole complexes in a "fusions-first" manner. In the rest of this section, we illustrate with examples.

**Example 14.** *Consider a building with a west room, an east room, and a hallway between them. For simplicity, let us consider only the floors of the building (ignore walls, ceilings, and so on). The ambient locale is given by the presentation*

- $\mathbb{L} = \langle G, R \rangle = \langle \{WR, H, ER\}, \varnothing \rangle$

*where*

- $WR$ = *west room*
- $H$ = *hallway*
- $ER$ = *east room*

*As a Hasse diagram:*

$$\begin{tikzcd}
& WR \vee H \vee ER \arrow[dl, dash] \arrow[d, dash] \arrow[dr, dash] & \\
WR \vee H \arrow[d, dash] \arrow[dr, dash] & WR \vee ER \arrow[dl, dash] \arrow[dr, dash] & H \vee ER \arrow[dl, dash] \arrow[d, dash] \\
WR \arrow[dr, dash] & H \arrow[d, dash] & ER \arrow[dl, dash] \\
& \bot &
\end{tikzcd}$$

*All of the generators are atomic, since none overlap (there are no meets among the generators):*

- $Atoms(\mathbb{L}) = \{WR, H, ER\}$

*Let us define a $\mathcal{G}$-sheaf $F$ that models the flooring of this building. For data, let there be the following available flooring materials:*

- $M = \{wood, tile, \ldots\}$

*For a gluing condition, let us say that sections glue if they contain the same materials:*

- $\mathcal{G}_U(\langle b_i \rangle_{i \in I(U)}) = \text{true}$ *iff* $b_i = b_j$ *for every* $i, j \in I(U)$.

- false *otherwise*

*We must check that this is a legitimate gluing condition.*

**Proof.** We must show that $\mathcal{G}$ satisfies the coherence conditions (G1)–(G3).

G1 *Local atomic data*. Trivial.

G2 *Downward stability*. We must show that if $\mathcal{G}_U(\langle b_i \rangle_{i \in I(U)})$ = true, then $\mathcal{G}_V(\rho^U_V(\langle b_i \rangle_{i \in I(U)}))$ = true for every $V \preccurlyeq U$. Assume the antecedent. Then $\rho^U_V(\langle b_i \rangle_{i \in I(U)}) = \langle b_i \rangle_{i \in I(V)}$.

- *Case 1*: if the length of $\langle b_i \rangle_{i \in I(V)} = 1$, it glues by (G1).
- *Case 2*: if the length of $\langle b_i \rangle_{i \in I(V)} \geqslant 2$, then by the assumption, $b_i = b_j$ for every $i, j \in I(V)$, so they glue.

G3 *Upward stability*. Given a selection of compatible patch candidates $\langle b_i \rangle_{i \in I(U)}$, we must show that if $\mathcal{G}_{U_i}(\langle b_i, b_j \rangle)$ = true for each $i, j \in I(U)$, then $\mathcal{G}_U(\langle b_i \rangle_{i \in I(U)})$ = true. Assume the antecedent. Since for every $i, j \in I(U)$, $b_i = b_j$ by the assumption, $\langle b_i \rangle_{i \in I(U)}$ glues. □

*For the atomic regions, fix a choice of local data:*

- $F(WR) = \{\langle wood \rangle\}$
- $F(H) = \{\langle wood \rangle\}$
- $F(ER) = \{\langle tile \rangle\}$

*Extend compatible data to meets, of which there is only* $\perp$, *so:*

- $F(\perp) = \{\langle\rangle\}$

*Recursively extend data to joins via gluing:*

- $F(WR \vee H) = \{\langle wood, wood \rangle\}$, *since* $F(WR) = F(H) = \{\langle wood \rangle\}$, *and* $wood = wood$.
- $F(WR \vee ER) = \varnothing$, *since* $F(WR) = \{\langle wood \rangle\}$, $F(ER) = \{\langle tile \rangle\}$, *and* $wood \neq tile$.
- $F(H \vee ER) = \varnothing$, *since* $F(H) = \{\langle wood \rangle\}$, $F(ER) = \{\langle tile \rangle\}$, $wood \neq tile$.
- $F(WR \vee H \vee ER) = \varnothing$, *since* $F(WR \vee H) = \{\langle wood, wood \rangle\}$, $F(H \vee ER) = F(WR \vee ER) = \varnothing$, *and* $wood \neq \varnothing$.

*In this building, there are two maximal fusions:*

- *The flooring of the west room and the hallway glue into one piece that covers both.*
- *The flooring that covers the east room is (trivially) glued into a single piece, namely itself.*

*Thus, the flooring of this building is really a collection of two independent fusions: the wooden floor that covers the east room and hallway, and the tiled floor that covers the east room. That implies:*

- *To separate the floors of the east room and hallway, you would have to use a saw to cut them, since they are fused. They are not merely sitting next to each other. Rather, they make up a single (fused) piece.*
- *By contrast, to separate the hallway and the east room, you would not need to cut them, since they are not fused. They simply happen to be sitting next to each other.*

*The parts of the fusions are clear:*

- *The wooden floor that covers the west room and the hallway has two parts: the wooden floor that covers the west room, and the wooden floor that covers the hallway.*
- *The tiled floor of the east room has no parts (in this locale), since it is not the fusion of other fusions.*

In the previous example, none of the atomic regions overlapped. The locale was discrete, and thus the sheaf was free to glue or not glue pieces as it saw fit. The story is different if there are overlaps in the locale itself. Overlaps in the locale require overlaps in the sheaf.

**Example 15.** *Consider the floor of a single room. Let us say that the regions of interest are its west half, its east half, and a six inch span where they overlap.*

*The ambient locale of this kind of space can be given by the presentation*

- $L = \langle G, R \rangle = \langle \{\perp, WH, O, EH\}, \{\perp \preccurlyeq O, O \preccurlyeq WH, O \preccurlyeq EH\} \rangle$

*where*

- $WH$ *= west half*
- $O$ *= overlap*
- $EH$ *= east half*

*The atomic sections of this locale are:*

- $WH$
- $EH$
- $\perp$

*In particular,* $O$ *is not an atomic region, because it is the non-trivial overlap of* $WH$ *and* $EH$*.*

*Here is the Hasse diagram:*

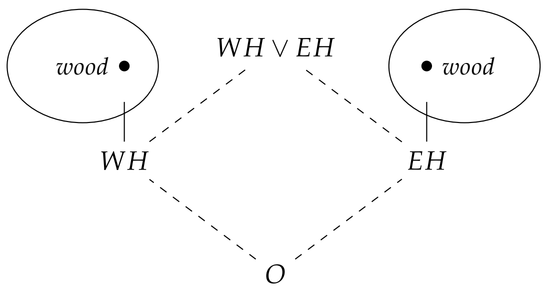
$$
\begin{tikzcd}
& WH \vee EH \arrow[dl, dash] \arrow[dr, dash] & \\
WH \arrow[dr, dash] & & EH \arrow[dl, dash] \\
& O \arrow[d, dash] & \\
& \perp &
\end{tikzcd}
$$

*Let us define a* $\mathcal{G}$*-sheaf* $F$ *that models the flooring of this room, using the same gluing condition from Example 14.*

*For the atomic regions, let us assign wood to both halves:*

- $F(WH) = \{\langle wood \rangle\}$
- $F(EH) = \{\langle wood \rangle\}$

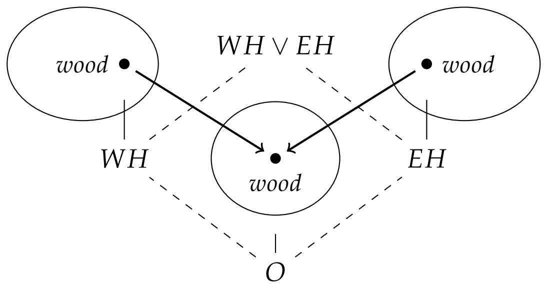
*In a picture (omitting* $\perp$ *for simplicity):*




*For the meet (the overlap* $O$*), the two halves restrict to the same thing:*

- $F(O) = \{\langle wood \rangle\}$

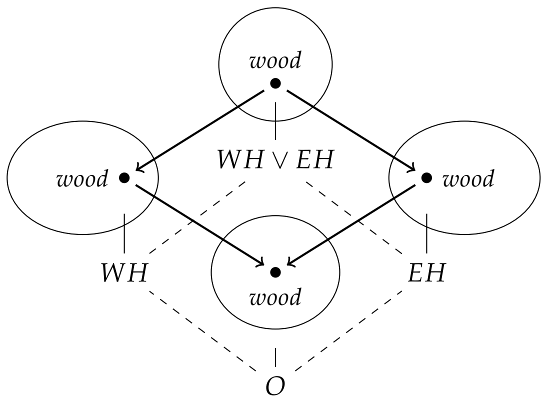
*Thus:*




*For the join, the west and east halves glue, since they're made from the same flooring materials and agree on their overlap:*

- $F(WH \vee EH) = \{\langle wood \rangle\}$

*Thus:*

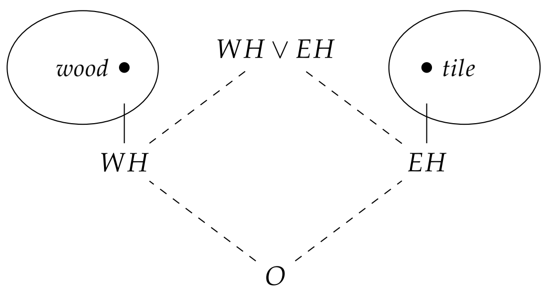



*The maximal fusion is a single piece of wooden flooring that covers the whole room. Its parts are the west and east halves, and (transitively) their overlap. The west and east halves themselves have a shared part, the strip of overlap.*

**Example 16.** *To illustrate a failed attempt to build a sheaf, let us take the locale and gluing condition from Example 15, but let's assign different flooring materials to the atomic regions:*

- $F(WH) = \{\langle wood \rangle\}$
- $F(EH) = \{\langle tile \rangle\}$

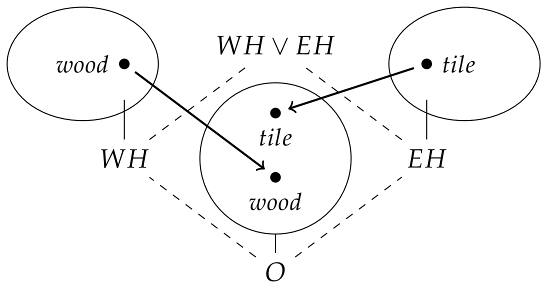
*As a picture:*




*Next, we attempt to extend this data to meets, which requires that we restrict to the overlap, and then filter out anything that can't glue. Here,* $F(WH)$ *restricts to* $\{\langle wood \rangle\}$, *and* $F(EH)$ *restricts to* $\{\langle tile \rangle\}$:

- $\rho_O^{WH}(\langle wood \rangle) = \langle wood \rangle$
- $\rho_O^{EH}(\langle tile \rangle) = \langle tile \rangle$

*Thus:*




*However, these cannot glue, because they are not the same. We see here that the data of WH and EH disagree on the overlap. Hence, we are unable to construct a coherent sheaf. This illustrates how sheaf theory requires and manages coherent gluing at all levels. Because it requires that pieces glue together coherently at every level of "zoom," it prevents us from ever putting together an incoherent part-whole complex in the first place.*

*It is worth spelling the failure out explicitly. Since $WH$ and $EH$ disagree on their overlap, $F$ cannot assign anything to $O$, so:*

- $F(O) = \varnothing$

*But that renders the restrictions $\rho_O^{WH}$ and $\rho_O^{EH}$ undefined, thereby severing our ability to zoom in and out. Thus, the system as a whole becomes incoherent.*

*Intuitively, this makes sense. If the western and eastern halves of a room were truly floored with different materials, then they would not overlap. There would be some sort of boundary between them where the one's materials end and the other's materials begin. But here, the ambient locale doesn't allow that possibility. In* this *particular locale, the western and eastern halves* do *overlap, so the sheaf must assign pieces to the different regions coherently, i.e., it must assign pieces that agree on their overlap.*

The previous two examples were spatial. But parts come in non-spatial guises too, and sheaves can model them just as well.

**Example 17.** *Suppose we say that human society (under some description) consists of the mesh of a specified set of relationships between the people that participate in that society.*

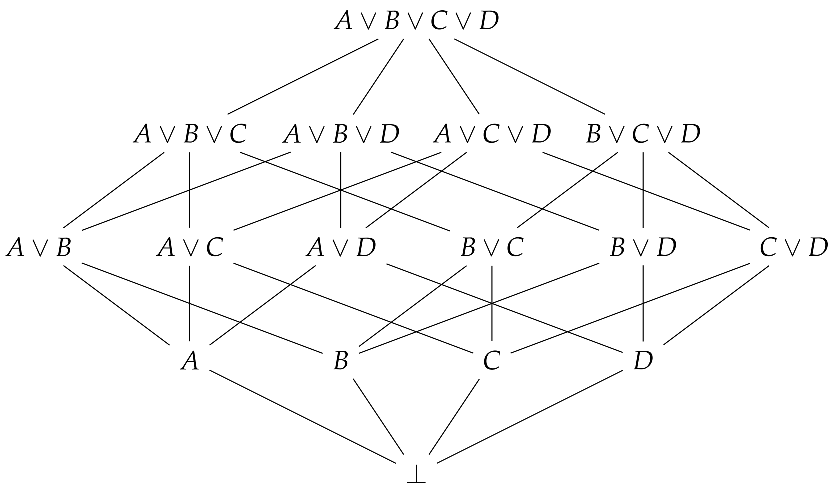
*Let $P$ be the population in question (a finite set of individual people), and let the regions of our locale be subsets of such individuals. Then the ambient locale is given by the presentation:*

- $\mathbb{L} = \langle G, R \rangle = \langle P, \varnothing \rangle$

*For concreteness, suppose:*

- $P = \{$ *A (Alice), B (Bob), C (Carol), D (Denny)* $\}$

*Then the Hasse diagram is isomorphic to the powerset of P:*




*All of the generators are atomic, since there are no meets among the generators:*

- $Atoms(\mathbb{L}) = \{A, B, C, D\}$

*Let us define a $\mathcal{G}$-sheaf F that models the mesh of a selected set of relationships over P. To do that, let us first specify a set R that picks out the (binary, symmetric) relationships of interest:*

- $R = \{f$ *(friends), m (married),* $\ldots\}$

*For convenience, if $U, V \in P$, $r \in R$, and U and V stand in relationship r, we will write $r(U, V)$.*

*For a gluing condition, let us say that sections glue if they are connected by the same relations:*

- $\mathcal{G}_U(\langle b_i \rangle_{i \in I(U)}) =$ true *iff for every $r \in R$, $r(U_i, U_j) \in b_i$ iff $r(U_j, U_i) \in b_j$, for every $i, j \in I(U)$*
- false *otherwise*

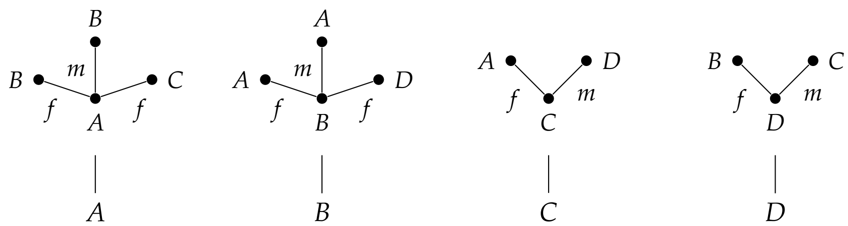
*We must check that this is a legitimate gluing condition.*

**Proof.** The proof is the same as before. □

*For the atomic regions, let us fix a choice of local data by assigning to each person the relations they stand in, e.g.:*

- $F(A) = \{\langle \{f(A, B), f(A, C), m(A, B)\} \rangle\}$
- $F(B) = \{\langle \{f(B, A), m(B, A), f(B, D)\} \rangle\}$
- $F(C) = \{\langle \{f(C, A), m(C, D)\} \rangle\}$
- $F(D) = \{\langle \{f(D, B), m(D, C)\} \rangle\}$

*To visualize this data, we can picture each fiber as a mini-graph:*




*For example, in the fiber over A:*

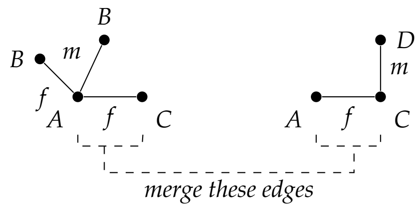
- *The f-labeled edge from A to B represents $f(A, B)$: A and B are friends.*
- *The m-labeled edge from A to B represents $m(A, B)$: A and B are married.*
- *The f-labeled edge from A to C represents $f(A, C)$: A and C are friends.*

*Next, we must extend compatible data to meets, of which there is only ⊥, so:*

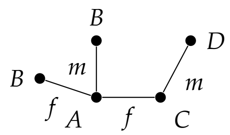
- $F(\bot) = \{\langle\rangle\}$

*Finally, we must extend atomic data to binary joins via gluing. The gluing condition essentially says that mini-graphs can be glued along shared edges, provided that they share exactly the same edges. To see how this works, consider (for example) the mini-graphs over $A$ and $C$:*


*Can these be glued? The answer is yes, because they share exactly one edge, namely the one labeled $f$. If you rotate the graphs sideways a bit, you can see how they can be merged along $f(A, C)$:*



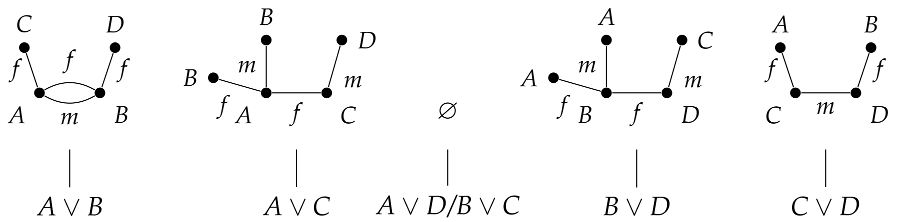

*Merging along $f(A, C)$ yields the following glued graph:*



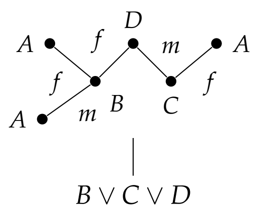

*By gluing binary joins in this fashion, we get:*

- $F(A \vee B) = \{\langle\{f(A, B), m(A, B), f(A, C)\}, \{f(B, A), m(B, A), f(B, D)\}\rangle\}$
- $F(A \vee C) = \{\langle\{f(B, A), m(B, A), f(B, D)\}\rangle, \{f(C, A), m(C, D)\}\}$
- $F(A \vee D) = \varnothing$
- $F(B \vee C) = \varnothing$
- $F(B \vee D) = \{\langle\{f(B, A), m(B, A), f(B, D)\}, \{f(D, B), m(D, C)\}\rangle\}$
- $F(C \vee D) = \{\langle\{f(C, A), m(C, D)\}, \{f(D, B), m(D, C)\}\rangle\}$

*As pictures:*




*Having glued joins of two regions, we must next glue joins of three atomic regions. For instance, take $B \vee C \vee D$. We can glue $B \vee C$ trivially (because they share no edges), we can glue $C \vee D$ along their shared $f$-edge, and we can glue $B \vee D$ along their shared $f$-edge. That yields:*



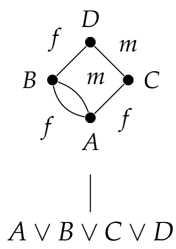
*By gluing all joins of three atomic regions in this fashion, we get:*

- $$F(A \vee B \vee C) = \left\{ \begin{array}{l} \langle \quad \{f(A,B), f(A,C), m(A,B)\}, \\ \quad\;\; \{f(B,A), m(B,A), f(B,D)\}, \\ \quad\;\; \{f(C,A), m(C,D)\} \quad \rangle \end{array} \right\}$$
- $$F(A \vee B \vee D) = \left\{ \begin{array}{l} \langle \quad \{f(A,B), f(A,C), m(A,B)\}, \\ \quad\;\; \{f(B,A), m(B,A), f(B,D)\}, \\ \quad\;\; \{f(D,B), m(D,C)\} \quad \rangle \end{array} \right\}$$
- $$F(A \vee C \vee D) = \left\{ \begin{array}{l} \langle \quad \{f(A,B), f(A,C), m(A,B)\}, \\ \quad\;\; \{f(C,A), m(C,D)\}, \\ \quad\;\; \{f(D,B), m(D,C)\} \quad \rangle \end{array} \right\}$$
- $$F(B \vee C \vee D) = \left\{ \begin{array}{l} \langle \quad \{f(B,A), m(B,A), f(B,D)\}, \\ \quad\;\; \{f(C,A), m(C,D)\}, \\ \quad\;\; \{f(D,B), m(D,C)\} \quad \rangle \end{array} \right\}$$

*At the top-most join, gluing four regions, we get:*

- $$F(A \vee B \vee C \vee D) = \left\{ \begin{array}{l} \langle \quad \{f(A,B), f(A,C), m(A,B)\}, \\ \quad\;\; \{f(B,A), m(B,A), f(B,D)\}, \\ \quad\;\; \{f(C,A), m(C,D)\}, \\ \quad\;\; \{f(D,B), m(D,C)\} \quad \rangle \end{array} \right\}$$

*As a picture:*




*The resulting sheaf yields a fused mesh of relationships over the population, which is glued together from smaller meshes over smaller subsets of the population.*

- *Each atomic fiber is a part of the whole (human society), and its data encodes the internal (relational) structure of that part.*
- *Mereological overlap is then modeled by shared relationships: two parts overlap if their relational graphs intersect coherently.*
- *Failure to glue (as in $F(A \vee D) = \varnothing$ and $F(B \vee C) = \varnothing$) reflects mereological separation: the atomic regions in question cannot be fused because they are not related in this mesh.*

For another example, consider processes. A process (or more generally any sequence of events, states, etc.) can be seen as a part-whole complex too.

**Example 18.** *Imagine a scenario where something can do one of two things repeatedly: at each step, it can do one thing ("option a") or another thing ("option b"), and then repeat the choice again.*

*To model this, fix a finite alphabet $\Sigma = \{a, b\}$, with "a" for "option a" and "b" for "option b." Then let $\Sigma^*$ be the set of all finite sequences (words) over $\Sigma$, with $\epsilon$ denoting the empty sequence. For instance, the sequence aab represents the sequence of length 3 that picks "option a" first, then "option a" again, and then finally "option b."*

*Let us say that $\Sigma^{\leqslant n}$ is the set of all finite sequences less than length $n$, and let us say that $\Sigma^{=n}$ is the set of finite sequences of exactly length $n$. Hence:*

- $\Sigma^{=0} = \{\epsilon\}$.
- $\Sigma^{=1} = \Sigma^{\leqslant 1} = \{\epsilon, a, b\}$.

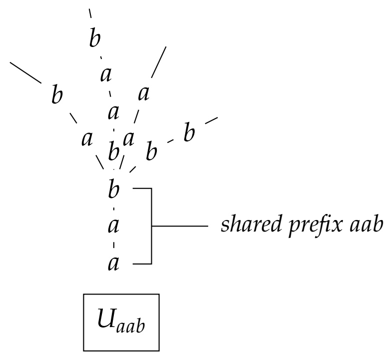
- $\Sigma^{\leqslant 2} = \{\epsilon, a, b, aa, bb, ab, ba\}$.
- $\Sigma^{=2} = \{aa, bb, ab, ba\}$.
- *Etc.*

*Given sequences* $w, v \in \Sigma^{\leqslant n}$ *with* $length(w) \leqslant length(v)$, *let us write* $w \subseteq v$ *to denote that* $w$ *is a prefix of* $v$, *as in* $aab \subseteq aabc$.

*Next, define a topology over* $\Sigma^{\leqslant}$ *by setting the open sets to be sequences that share a prefix:*

- $U_w = \{v \in \Sigma^{\leqslant} \mid w \subseteq v\}$.

*So* $U_w$ *consists of all sequences that continue* $w$. *For instance, if* $w = aab$, *then we might picture* $U_w$ *as a kind of bouquet or bundle of sequences that are all bound at their shared stem (aab) but then branch out in different directions:*




*We can form a locale from this topology. Let* $\mathbb{L}$ *be the locale given by the presentation* $\langle G, R\rangle$, *where:*

- $G = \{U_w \mid w \in \Sigma^n\}$, *i.e., each open is a generator.*
- $R = \{U_w \preccurlyeq U_v \mid v \subseteq w\}$, *i.e., bouquets with longer prefixes are lower.*

*For example, given* $\Sigma^{\leqslant 2}$, *we have the following generators:*

- $G = \{U_\epsilon, U_a, U_b, U_{aa}, U_{bb}, U_{ab}, U_{ba}\}$.

*Here are some of the relations:*

- $U_{aa} \preccurlyeq U_a$ *and* $U_{ab} \preccurlyeq U_a$, *since "a" is a prefix of aa and ab.*
- $U_{bb} \preccurlyeq U_b$ *and* $U_{ba} \preccurlyeq U_b$, *since "b" is a prefix of bb and ba.*
- *Every generator is lower than* $U_\epsilon$, *since* $\epsilon$ *(the empty sequence) is a prefix of every sequence.*

*The Hasse diagram looks like this:*

$$\begin{tikzcd}
 & & & U_\epsilon \arrow[dll, no head] \arrow[drr, no head] & & & \\
 & U_a \arrow[dl, no head] \arrow[dr, no head] & & & & U_b \arrow[dl, no head] \arrow[dr, no head] & \\
U_{aa} \arrow[drrr, no head] & & U_{ab} \arrow[dr, no head] & & U_{ba} \arrow[dl, no head] & & U_{bb} \arrow[dlll, no head] \\
 & & & \bot & & &
\end{tikzcd}$$

*Think of moving upwards in this locale as forgetting information about (or alternatively, as committing less to) the history of the sequence. For example, think of* $U_{ab}$ *as a region where we know that "a" happened first and then "b" happened, but think of* $U_a$ *as a region where we know only that "a" happened first and we don't know what happened after that. The top element is* $U_\epsilon$, *which means we don't know anything about the sequence of actions. The* $\bot$ *element indicates not that we know nothing, but that there is no sequence at all.*

*Notice that implication moves upwards: $U_{ab}$ implies $U_a$ because if I know (at $U_{ab}$) that "a" happened first and then "b" happened, then I certainly know that "a" happened first.*

*Further, no generator is the non-trivial overlap of other generators, so every generator is an atomic region:*

- $Atoms(\mathbb{L}) = G$

*As with any locale, we can write each region canonically as the join of its atomic regions:*

- $U_{I(U)} = \bigvee_{i \in I(U)} U_i$

*But here, what this means is that we can canonically write each region as the join of its "most specified" prefixes. For instance, $I(U_a) = \{aa, ab\}$, so:*

- $U_a = U_{I(U_a)} = \bigvee\{U_{aa}, U_{ab}\}$.

*This makes sense. Since $U_a$ is a region where we know only that "a" happened first, it is the join of all maximal continuations that begin with "a."*

*This particular locale is interesting because it models the "process space" of any 2-stage sequence that can make one of two choices at each stage. Let us now assign some actual processes to this ambient space, using a $\mathcal{G}$-sheaf.*

*Imagine a machine $m$ that can run multiple concurrent processes, all of whom share the same memory. For simplicity, let us suppose that the machine has two registers ($R = \{r_1, r_2\}$), each of which can hold one bit (1 or 0). So, at any point in time the machine's memory state $S : \{0,1\} \times \{0,1\}$ can be one of the following:*

- $S = \{\langle 0,0\rangle, \langle 1,0\rangle, \langle 0,1\rangle, \langle 1,1\rangle\}$, *with initial state* $s_0 = \langle 0,0\rangle$.

*We can think of the concurrent processes of interest as a selection of programs that we want to run on the machine all at the same time. In terms of behavior, let us say that each program-run reads a word from its input stream, one character at a time, and in response to each character, it takes one of the following actions $A$: it writes a value (1 or 0) to one of the registers, it writes (possibly distinct) values to both registers, or it does nothing and leaves the registers as they are:*

- $A = \{\{r1 \mapsto v\}, \{r2 \mapsto v\}, \{r1 \mapsto v, r2 \mapsto w\}, \varnothing\}$, *where* $v, w \in \{0,1\}$.

*We can define a process (program trace) as a map from $n$-length words to $n$-length sequences of write actions, where we require that such maps agree on prefixes (since a process responding to ab and aa would do the same thing on the first a). This way, a program trace records for each input stream the sequence of write actions that result. For concreteness, here are two such traces:*

- $f : \Sigma^{=2} \to A \times A$
  - $f(aa) = \langle\{r1 \mapsto 1\}, \{r1 \mapsto 0\}\}\rangle$
  - $f(ab) = \langle\{r1 \mapsto 1\}, \{r2 \mapsto 0\}\}\rangle$
  - $f(bb) = \langle\{\{r2 \mapsto 0\}, \{r2 \mapsto 1\}\}\rangle$
  - $f(ba) = \langle\{\{r2 \mapsto 0\}, \{r1 \mapsto 1\}\}\rangle$
- $g : \Sigma^{=2} \to A \times A$
  - $g(aa) = \langle\{r2 \mapsto 0\}, \{r2 \mapsto 0\}\}\rangle$
  - $g(ab) = \langle\{r2 \mapsto 0\}, \varnothing\}\rangle$
  - $g(bb) = \langle\{\{r1 \mapsto 1\}, \{r1 \mapsto 0\}\}\rangle$
  - $g(ba) = \langle\{\{r1 \mapsto 1\}, \{r1 \mapsto 1\}\}\rangle$

*Let us now say that concurrent processes are compatible if they "play well" together, i.e., they share resources (memory) consistently. In particular, given two processes $f$ and $g$, let us say:*

- *$f$ and $g$ are compatible at stage $n$ if they write to different registers.*
- *$f$ and $g$ are compatible at stage $n$ if they write the same value to the same register.*

- *$f$ and $g$ conflict at stage $n$ if they write different values to the same register.*

*We can formalize this notion as a gluing condition that says a selection of patch candidates glue at $U_w$ if they play well up to $w$. Fix a selection of programs $P = \{f, g, \ldots\}$ to run on the machine, then:*

- *Given a selection of patch candidates $\langle (b_{i,p})_{p \in P} \rangle_{i \in I(U_w)}$ over a region $U_w$ with trace length $|w|$, $\mathcal{G}_{U_w}(\langle b_i \rangle_{i \in I(U_w)}) = \text{true}$ iff the following condition holds. Write $b_{i,p}[m]$ to denote the write actions of process $p$ in region $i$ at stage $m$. Then, require that at each stage $m \leqslant |w|$ and every register $r \in R$, the set*

$$\{v \in \{0,1\} \mid \exists i \in I(U_w), p \in P \textit{ where } r \mapsto v \in b_{i,p}[m]\}$$

  *has cardinality at most 1. In other words, two values are not written to the same register.*
- $\mathcal{G}_{U_w}(\langle b_i \rangle_{i \in I(U_w)}) = \text{false}$ *otherwise.*

*We must check that this is a legitimate gluing condition.*

**Proof.** We must check that $\mathcal{G}_{U_w}$ is downwards and upwards stable.

- *Downwards stability*. Assume that $\mathcal{G}_{U_w}(\langle b_i \rangle_{i \in I(U_w)}) = \text{true}$. Then $\mathcal{G}_{U_i}(\langle b_i \rangle) = \text{true}$ for each $i \in I(U_w)$ since by $\mathcal{G}_{U_w}$, every $b_i, b_j$ play well on their prefixes.
- *Upwards stability*. Assume $\mathcal{G}_{U_{\{i,j\}}}(\langle b_i, b_j \rangle) = \text{true}$ for all $i, j \in I(U_w)$. Then $\mathcal{G}_{U_w}(\langle b_i \rangle_{i \in I(U_w)}) = \text{true}$ since no $i, j$ conflict on writes. □

*With a gluing condition at hand, we can now define a $\mathcal{G}$-sheaf $F$. Let our selection of processes be $P = \{f, g, \ldots\}$. Then, we can fix the atomic data (omitting outer brackets to avoid clutter):*

- $F(U_{aa}) = (f(aa), g(aa)) = (\langle \{r1 \mapsto 1\}, \{r1 \mapsto 0\} \rangle, \langle \{r2 \mapsto 0\}, \{r2 \mapsto 0\} \rangle)$.
- $F(U_{ab}) = (f(ab), g(ab)) = (\langle \{r1 \mapsto 1\}, \{r2 \mapsto 0\} \rangle, \langle \{r2 \mapsto 0\}, \varnothing \rangle)$.
- $F(U_{bb}) = (f(bb), g(bb)) = (\langle \{r2 \mapsto 0\}, \{r2 \mapsto 1\} \rangle, \langle \{r1 \mapsto 1\}, \{r1 \mapsto 0\} \rangle)$.
- $F(U_{ba}) = (f(ba), g(ba)) = (\langle \{r2 \mapsto 0\}, \{r1 \mapsto 1\} \rangle, \langle \{r1 \mapsto 1\}, \{r1 \mapsto 1\} \rangle)$.

*There are no meets among the generators beyond $\perp$, so:*

- $F(\perp) = \{\langle\rangle\}$.

*Next, we must extend $F$ to joins via gluing. So, for each $U_w$, we must assign:*

- $F(U_w) = \{\langle (b_{i,p})_{p \in P} \rangle_{i \in I(U_w)} \mid \mathcal{G}_{U_w}(\langle (b_{i,p})_{p \in P} \rangle_{i \in I(U_w)}) = \text{true}\}$.

*Let's compute $F(U_a) = F(U_{aa} \vee U_{ab})$. To determine if $(f(aa), g(aa))$ and $(f(ab), g(ab))$ glue, we need to check that they do not write conflicting values.*

- *Stage 1 (at the shared prefix "a"): $f(aa)$ and $f(ab)$ write $1$ to $r1$, while $g(aa)$ and $g(ab)$ write $0$ to $r2$. Since $f$ and $g$ write to different registers, there is no conflict.*
- *Stage 2: $f(ab)$ and $g(aa)$ write the same value (namely, 0) to $r2$, $f(aa)$ writes $0$ to $r1$, and $g(ab)$ does nothing, so there are no conflicts.*

*Hence, $(f(aa), g(aa))$ and $(f(ab), g(ab))$ glue to form a unique section:*

- $F(U_a) = (f(aa), g(aa)), (f(ab), g(ab))$

*Now let's compute $F(U_b) = F(U_{bb} \vee U_{ba})$. To determine if $(f(bb), g(bb))$ and $(f(ba), g(ba))$ glue, we need to again check for conflicting writes:*

- *Stage 1 (at the shared prefix "b"): $f(bb)$ and $f(ba)$ write $0$ to $r2$, while $g(bb)$ and $g(ba)$ write $1$ to $r1$, so there is no conflict.*
- *Stage 2: $f(bb)$ and $f(ba)$ write $1$ to different registers, so they do not conflict with each other, while $f(ba)$ and $g(ba)$ write $1$ to $r1$, so they do not conflict either. However, $g(bb)$ writes $0$ to $r1$, which conflicts with $g(ba)$'s and $f(ba)$'s attempt to write $1$ to the same register.*

*Since we have a conflict,* $(f(bb), g(bb))$ *and* $(f(ba), g(ba))$ *fail to glue over* $U_b$*. Notice:*

- *The processes* $f$ *and* $g$ *agree locally at* $U_a$*.*
- *By contrast, they disagree locally at* $U_b$*.*
- *There is no global section that glues together all of* $f$ *and* $g$*'s behavior at the top* $U_\epsilon$*, thus* $f$ *and* $g$ *are not globally compatible processes.*

*This sort of example illustrates how sheaves can model processes, concurrency, and resource conflicts. Here the processes were programs running on a simple machine, but they could just as easily be biological processes competing for resources, etc.*

*Whatever the concrete details may be, this example captures how local behavior can be integrated and extended over larger regions of the process space. One might naively think that the "parts" of such systems are the processes. But there is a different way to slice it: if you want to talk about the integrity of the "whole" of a concurrent system, you need to talk about how that involves coherent, integrated behavior that is functionally united locally across the various "regions" and "stages" of the system's evolution.*

TODO:

- Add example: something over a continuous interval/timeline? E.g., maybe something over a timeline (the frame of opens taken from the usual topology of R)? Maybe we can define a gluing condition for a mass of clay over time that says pieces glue if they agree on overlaps, so that the whole lump of clay can have parts replaced over time but as a whole it never breaks into fragments? Maybe the "closure" is even a modality.
- Add example: Socrates and seated Socrates?
- Note Spivak et al's behavioral mereology is an example of a $\mathcal{G}$-sheaf (and check the details to make sure that's really true).

## 4. Modalities in the Sheaf-theoretic Setting

In the context of sheaves, modalities manifest as $j$-operators (also called local operators). A $j$-operator is a closure operator on the underlying locale.

**Definition 18** ($j$-operators)**.** *Given a locale* $\mathbb{L}$*, a* $j$*-operator on* $\mathbb{L}$ *is a closure operator* $j : \mathbb{L} \to \mathbb{L}$ *satisfying the following conditions:*

*(J1)* Inflation. $U \preccurlyeq j(U)$.
*(J2)* Idempotence. $j(j(U)) = j(U)$.
*(J3)* Meet-preservation. $j(U \wedge V) = j(U) \wedge j(V)$.

A $j$-operator induces a $j$-sheaf.

**Definition 19** ($j$-sheaves)**.** *Given a sheaf* $F$ *over a locale* $\mathbb{L}$ *and a* $j$*-operator* $j : \mathbb{L} \to \mathbb{L}$*, the corresponding* $j$*-sheaf, denoted* $F_j$*, is given by:*

$$F_j = F(j(U)).$$

**Remark 12.** *In a sheaf, there are a variety of other modalities beyond the traditional alethic ones (necessity and possibility). Any closure operator qualifies as a modality of some description.*

**Example 19.** *From Example 17, recall the mesh of human relationships modeled by a* $\mathcal{G}$*-sheaf* $F$ *defined over the presented locale* $\mathbb{L} = \langle P := \{A, B, C, D\}, \varnothing \rangle$*. Let us define a family of "reachability" modalities over this mesh.*

*For each $r \in R$, write $\rightsquigarrow_r$ for the reflexive and transitive closure of $r$ on the generators. Hence, $\rightsquigarrow_f$ is the transitive closure of friendship on the generators, and $\rightsquigarrow_m$ is the transitive closure on marriage.*

*Then for each $r \in R$, define $j_r$ inductively:*

- Base case. *For each generator $U \in G$, set $j_r$ to the join of all other generators reachable via $r$:*

$$j_r(U) := \bigvee \{V \mid U \rightsquigarrow_r V\}$$

- Inductive step. *Extend to arbitrary joins $U_{I(U)}$:*

$$j_r(U_{I(U)}) := \bigvee_{i \in I(U)} j_r(U_i)$$

*We need to check that this is a j-operator.*

**Proof.** We check (J1)–(J3) from the definition.

TODO. Do the base case, then the inductive step. □

*Intuitively, this operator expands every region $U$ to the largest region that is reachable from $U$ by $r$. In other words, it expands each subset of society to the largest subset of society that is connected by $r$. Hence, $j_f(U)$ yields all those who are connected to $U$ through a chain of friends, while $j_m(U)$ yields all those who are connected to $U$ through a chain of marriage (which in a monogamous society will yield only married couples but in a polygamous society may be more revealing).*

*Applying $j_f$ (for instance) to $\mathbb{L}$ yields the following:*

- *$j_f(A) = A \vee B \vee C \vee D$, because $A \rightsquigarrow_f A$, $A \rightsquigarrow_f B$, $A \rightsquigarrow_f D$, and $A \rightsquigarrow_f C$.*
- *$j_f(B) = A \vee B \vee C \vee D$, because $B \rightsquigarrow_f B$, $B \rightsquigarrow_f D$, $B \rightsquigarrow_f A$, and $B \rightsquigarrow_f C$.*
- *Similar for $j_f(C)$ ad $j_f(D)$.*
- *$j_f(\bot) = \bot$.*

*Hence, everyone in this mini-society is connected through friends (or friends-of-friends, etc.). Notice also that everyone is connected* immediately*, i.e., at the first application of $j_f$.*

*When it comes to marriage, the situation is different. Applying $j_m$ yields:*

- *$j_m(A) = A \vee B$, because $A \rightsquigarrow_m A$ and $A \rightsquigarrow_m B$.*
- *$j_m(B) = A \vee B$, because $B \rightsquigarrow_m B$ and $B \rightsquigarrow_m A$.*
- *$j_m(C) = C \vee D$, because $C \rightsquigarrow_m C$ and $C \rightsquigarrow_m D$.*
- *Similar for $j_m(D)$.*
- *$j_m(A \vee B) = A \vee B$, since $A$ and $B$ are already connected.*
- *$j_m(C \vee D) = C \vee D$, since $C$ and $D$ are already connected.*
- *$j_m(A \vee C) = A \vee B \vee C \vee D$, since from $A$, $A$ can reach $B$ (i.e., $A \rightsquigarrow_m B$) and from $C$, $C$ can reach $D$ (i.e., $C \rightsquigarrow_m D$).*
- *Similar for the rest.*

*In contrast to the $j_f$ modality, the $j_m$ modality keeps the $A, B$ component separate from the $C, D$ component at all regions (sub-populations) that don't include a member of both couples, just as we would expect.*

*Now that we have defined $j_f$ and $j_m$, we can construct a modal overlay for each that we can use to filter the original mesh:*

- *Define the friendship mesh as $F_f$, filtered by $j_f$, i.e., set $F_f(U) := F(j_f(U))$.*
- *Define the marriage mesh as $F_m$, filtered by $j_m$, i.e., $F_m(U) := F(j_m(U))$.*

**Example 20.** *Recall the example of concurrent processes $f$ and $g$ from Example 18. We can define an* "already happened" *modality that captures what has definitely occurred so far.*

**Definition 20** (Already-happened operator). *Let $j_H$ be given by:*

$$j_H(U_w) := \bigvee\{U_v \mid v \subseteq w\},$$

*i.e., the join of all opens corresponding to prefixes of $w$ (including $w$ itself).*

*Intuitively, $j_H(U_w)$ is the region that remembers everything that has already happened along $w$. It is a closure operator that closes upwards by collecting all shorter prefixes.*

*We must check that this is a legitimate $j$-operator.*

**Proof.** We check (J1)–(J3).

J1 *Inflation*. $U_w \preccurlyeq j_H(U_w)$ holds because $U_w$ is included among the prefixes being joined.
J2 *Idempotence*. Applying $j_H$ again adds no new prefixes, so $j_H(j_H(U_w)) = j_H(U_w)$.
J3 *Meet-preservation*. The meet of two regions corresponds to their longest shared prefix, whose prefixes are all of the prefixes collected by $j_H$. Hence, $j_H(U_w \wedge U_v) = j_H(U_w) \wedge j_H(U_v)$. □

*Applying $j_H$ to the generators of $\mathbb{L}$:*

- *For $U_{aa}$: $j_H(U_{aa}) = U_\epsilon \vee U_a \vee U_{aa}$.*
- *For $U_{ab}$: $j_H(U_{ab}) = U_\epsilon \vee U_a \vee U_{ab}$.*
- *For $U_{ba}$: $j_H(U_{ba}) = U_\epsilon \vee U_b \vee U_{ba}$.*
- *For $U_{bb}$: $j_H(U_{bb}) = U_\epsilon \vee U_b \vee U_{bb}$.*
- *For $U_a$: $j_H(U_a) = U_\epsilon \vee U_a$.*
- *For $U_b$: $j_H(U_b) = U_\epsilon \vee U_b$.*
- *For $U_\epsilon$: $j_H(U_\epsilon) = U_\epsilon$.*

*Since $j_H$ filters each region to everything that is already determined in that region, we can use it to define an overlay of $F$*

$$F_H(U_w) := F(j_H(U_w)),$$

*so that sections at $U_w$ remember only what has happened along all prefixes of $w$.*

**Example 21.** *Recall the example of concurrent processes $f$ and $g$ from Example 18. We can define a safety ("nothing bad happens") modality as a $j$-operator that identifies the largest safe extensions of a given region.*

**Definition 21** (Safety operator). *Let us say that a region $U$ is safe if all processes in $F(U)$ play well together, i.e., if there are no write conflicts. Then let $j_S : \mathbb{L} \to \mathbb{L}$ be given by:*

$$j_S(U) := \begin{cases} \bigvee\{V \mid U \preccurlyeq V \text{ and } V \text{ is safe}\} & \text{if this join is non-empty} \\ U & \text{otherwise.} \end{cases}$$

*Intuitively, $j_S(U)$ inflates $U$ to the largest region that is guaranteed safe starting from $U$.*

*We must check that $j_S(U)$ is a legitimate $j$-operator.*

**Proof.** We check (J1)–(J3).

J1 *Inflation*. By construction, $U \preccurlyeq j_S(U)$ whenever $U$ has any safe parent regions, otherwise $j_S(U) = U$.
J2 *Idempotence*. Applying $j_S$ more than once does not change the result, since appyling it once takes the join of all safe parents. Hence, $j_S(j_S(U)) = j_S(U)$.

J3 *Meet-preservation*. For any $U$ and $V$, since $U \wedge V$ is $U$ or $V$,

$$j_S(U \wedge V) = \bigvee\{W \mid U \wedge V \preccurlyeq W \text{ and } W \text{ safe}\} = j_S(U) \wedge j_S(V). \quad \square$$

*Let's apply $j_S$ to the generators of $\mathbb{L}$:*

- *$j_S(U_{aa}) = U_a$ since its safe parent regions are $U_{aa}$ and $U_a$.*
- *Similarly, $j_S(U_{ab}) = U_a$.*
- *$j_S(U_{ba}) = U_{ba}$ because the only safe parent of $U_{ba}$ is $U_{ba}$ itself.*
- *Similarly, $j_S(U_{bb}) = U_{bb}$.*

*Now extend it to joins:*

- *$j_S(U_a) = j_S(U_{aa}) \vee j_S(U_{ab}) = U_a \vee U_a = U_a$.*
- *$j_S(U_b) = U_b$ since $U_b$ is unsafe (there are conflicts among its generators) and thus no further extension can be safe.*
- *$\perp$ is trivially fixed: $j_S(\perp) = \perp$.*

*Notice:*

- *Each generator $U_w$ represents a part of a process's history.*
- *The operator $j_S$ identifies the largest safe fusion containing $U_w$, i.e., the maximal extension of the part where processes play well together.*
- *If no safe extensions exist (as in $U_b$), then $j_S(U_b)$ doesn't get bigger, indicating that safety cannot be guaranteed any further beyond this part.*
- *Hence, $j_S$ captures a mereological notion of integrity, showing which combinations of parts form consistent wholes and which do not.*

TODOs:

- Add example: A statue and the lump of clay?

## 5. Classical Mereological Notions in the Sheaf-theoretic Setting

In this section, we provide a discussion of what classical notions of mereology look like in the sheaf-theoretic setting.

- *Cambridge fusions*. Sheaves handle Cambridge fusions correctly.
- *Mere collections*. The collection of all dogs. Is that a "whole"? Well, we could build a sheaf whose atomic regions are filled with dogs, none of which glue. Then we have a collection of dogs, but no glued object. That matches exactly the intuition: yes, we have a "collection" (we built a sheaf for it, after all), but the internals of that sheaf reveal that it's *merely* a collection, i.e., that its parts are not glued.
- *Co-habitating fusions*. Sheaves allow multiple fusions to occupy the same locale, without being glued. For instance, in the sheaf of real-valued functions over real number line, there are many functions that glue together, and occupy the same locale.
- *Non-boolean algebra*. The parts space is Heyting, not Boolean. We're not saddled with such a strong complement operation. You can pick a locale that is Boolean if you need it, but this framework doesn't require it. In fact, the positive logic of a locale is "geometric logic."
- *Reflexivity, antisymmetry, and transitivity*. These are guaranteed. Locally, of course, you may not have transitivity. But globally, it's a theorem. [Check that.]
- *Distributivity*. TODO: do the glued sections of a sheaf have to be distrubitive? Only inside what glues (since we glue pairwise, so every $i \vee j$ of the cover.
- *An empty element*. There is a need for a bottom element in the *algebra* of parts, but a sheaf need not contain any such thing. There is no need here to try and construct

awkward mathematical structures that do algebra on parts but yet don't have a bottom element because our ontological intuitions tell us there can be no such thing. That confuses two issues: algebra and integrity. So here we separate those cleanly, and the algebra can do algebra while the sheaf can do integrity. [In a sheaf you CAN'T assign an empty element to bottom, for coherence, so the bottom element is special...need to say more about that and figure it out.]

- *Supplementation principles*. Sheaves don't constrain one way or another. [Is that really true? Maybe it's better to say that it doesn't force any supplementation principles, which might provide a reason to call into question whether supplementation is another one of those ideas that is about integrity of parts but has been confused with the algebra of parts.]
- *Ordering of parts*. Consider that "pit" and "tip" have the same parts but are different words. These differences can be handled by different sheaves over a 3-stage prefix-ordered locale as in the example of concurrent processes. Note that we retain extensionality.
- *Extensionality*. Classical mereology's notion of extensionality essentially flattens any structure and is thus overly aggressive. This is why extensionality is so controversial. The sheaf-theoretic perspective retains extensionality, but is much more nuanced. [Here too, I suspect that mereological discussions of extensionality have confused the algebra of parts and the integrity of wholes.]
- *Gunk and atoms*. You can model continuity and gunky parts if you so desire. You just need a sober space to do it. TODO: check that we can model continuity in the locale in this way. TODO: can you do continuity only in the sheaf data, without an underlying continuous decompositon in the locale? I would think that if you can't infinitely decompose into smaller opens around a point in the locale, you couldn't do such a thing in the sheaf data?
- *Priority of wholes*. The framework is agnostic as to whether you take an Aristotelian-Thomistic approach (TODO: cite Aquinas, Arlig, and that guy who wrote that recent book defending the Aristotelian view).
- *The whole is greater than its parts*. The framework is agnostic as to whether you want to be a Scotist and say that the whole is something over and above its parts (cite Cross) or an Ockhamist who says it is not (TODO: cite Normore, Arlig).

## 6. Category Theory

In this section, we cover the parts of category theory that we will use in the remainder of the paper. Readers familiar with category theory can skip this section.

Since at least Aristotle's time, philosophers have been interested in placing objects of the same kind into categories. Modern category theorists do this too: a category is a collection of objects that are all the same in kind. But category theory has a further requirement: not only must you tell us what the objects are in your category, you must also tell us about the maps between them, so that we know how to relate/compare those objects. Further, those maps must compose in well-behaved ways.

**Definition 22** (Categories)**.** *A category $\mathbb{C}$ consists of the following data:*

1. *A collection of objects $X$, $Y$, $Z$, and so on, denoted $Objs(\mathbb{C})$*
2. *A collection of morphisms $f : X \to Y$, $g : Y \to Z$, and so on, denoted $Morphs(\mathbb{C})$, each with a designated domain (an object) and codomain (an object), such that:*
   (a) *There is an identity morphism $id_X : X \to X$ for each object $X$.*

(b) *Morphisms compose, i.e., if $f : X \to Y$ and $g : Y \to Z$ are morphisms in $\mathbb{C}$ such that $f$'s codomain matches $g$'s domain, then their composite $g \circ f : X \to Z$ (pronounced "$g$ after $f$") is a morphism in $\mathbb{C}$ too.*

*The objects and morphisms of $\mathbb{C}$ must satisfy the following conditions:*

*(K1) Composing with an identity has no effect, i.e. for any morphism $f : X \to Y$,*

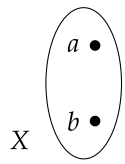
$$id_Y \circ f = f \text{ and } f = f \circ id_X.$$

*(K2) Composition is associative, i.e., for any morphisms $f : X \to Y$, $g : Y \to Z$, and $h : Z \to W$,*

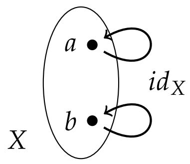
$$(h \circ g) \circ f = h \circ (g \circ f).$$

**Example 22.** *The category $\mathbb{S}$et of sets and total maps has all sets for its objects and all total function between them for its morphisms. To check that it is a category, we need to confirm that it satisfies all the requirements.*

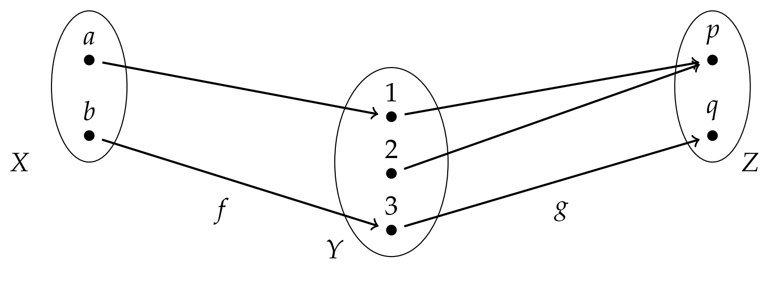
*First, does every object have an identity? Yes, because for every set $X$, there is an identity function $id_X : X \to X$ given by $id_X(x) = x$. For instance, take the following set $X$:*



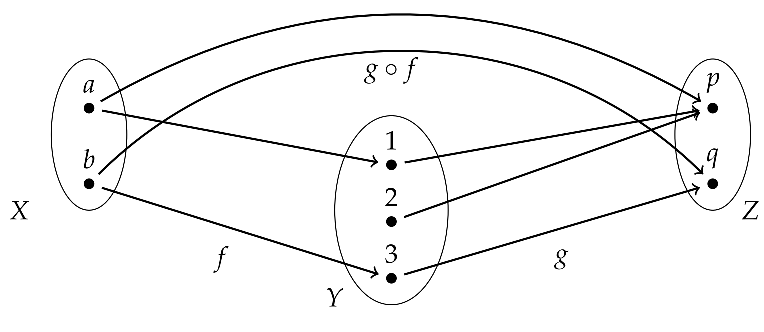

*There is an identity function that sends each point of $X$ to itself:*




*Do morphisms compose? Yes: the morphisms of this category are functions, and functions compose. For instance, take the following $f : X \to Y$ and $g : Y \to Z$:*




*Their composite $g \circ f : X \to Z$ is a function too:*

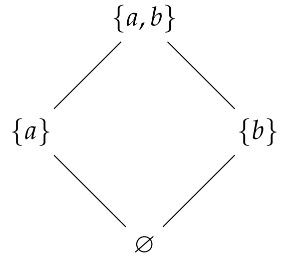


*Further, it is clear that composing with an identity has no effect, since identity functions do not shuffle around any points. It is also clear that function composition is associative: it does not matter if I take $f$ first and then $h \circ g$, or $g \circ f$ first and then $h$. I get to the same results either way. So $\mathbb{S}et$ is a genuine category.*

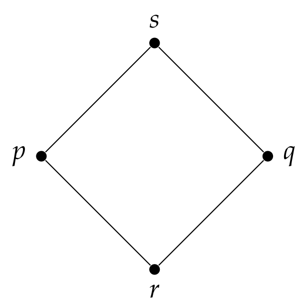
**Example 23.** *The category $\mathbb{T}op$ of topological spaces and continuous maps is the category whose objects are all topological spaces and whose morphisms are the continuous maps between them.*

**Example 24.** *The category $\mathbb{F}rm$ of frames and frame homomorphisms is the category whose objects are frames and whose morphisms are frame homomorphisms, i.e., maps that preserve arbitrary joins and finite meets.*

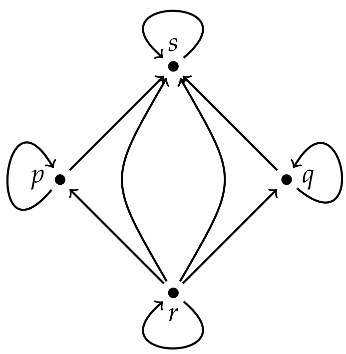
**Example 25.** *A poset can be seen as a category. For example, consider the powerset of $\{a, b\}$, ordered by inclusion. The Hasse Diagram looks like this:*




*The fact that the elements of this diagram are sets is inessential. We could relabel them with arbitrary names to make the structure of the underlying poset obvious:*




*Hasse diagrams only show the minimal information needed to understand the poset. A poset is reflexive and transitive, but we don't draw lines for the reflexive and transitive steps, to avoid clutter. Thus, for instance, in this case, although $p \leqslant p$, we don't draw a line to represent it in the Hasse diagram. However, if we were to draw the full poset, we would get something that looks like this:*



*This is category. It's objects are the points ($r$, $p$, $q$, and $s$), and its morphisms are the arrows in the picture. It has identities (each reflexive loop), morphisms compose (because a poset is transitive), and so on.*

*This illustrates that the morphisms in a category need not be functions, nor do they need to be function-like. Here, we put a morphism between two objects just to state a fact about the poset: a morphism from $p$ to $s$ simply means $p \leqslant s$, i.e., that "$p$ is lower than $s$ in the ordering."*

**Definition 23** (Opposite categories). *For any category $\mathbb{C}$, define the opposite category $\mathbb{C}^{op}$ as the category with the same objects as $\mathbb{C}$, but whose morphisms and composition are turned around. That is:*

1. *For any morphism $f : X \to Y$ in $\mathbb{C}$, the corresponding morphism in $\mathbb{C}^{op}$ is $f^{op} : Y \to X$.*
2. *For any morphisms $f : X \to Y$, $g : Y \to Z$, and their composite $g \circ f : X \to Z$ in $\mathbb{C}$, the correspond morphisms and composite in $\mathbb{C}^{op}$ are $f^{op} : Y \to X$, $g^{op} : Z \to Y$, and $(g \circ f)^{op}$ $= f^{op} \circ g^{op} : Z \to X$.*

**Example 26.** *The category $\mathbb{L}oc$ of locales is defined as the opposite category of $\mathbb{F}rm$. Thus, the objects of $\mathbb{L}oc$ are the same objects as $\mathbb{F}rm$, which is why we can call a frame a locale or vice versa.*

### 6.1. Functors

Functors are maps between categories that preserve categorical structure. That is, a functor maps one category to another in such a way that it preserves the identities and composition of the original category, so that you end up picking out a kind of "image" of the first category in the second category.

**Definition 24** (Functors). *A functor $F : \mathbb{J} \to \mathbb{C}$ is comprised of the following data:*

1. *A mapping of objects to objects, i.e., for each object $X \in \mathbb{J}$, $F(X)$ is an object in $\mathbb{C}$.*
2. *A mapping of morphisms to morphisms, i.e., for each morphism $f : X \to Y$ in $\mathbb{J}$, $F(f) : F(X) \to F(Y)$ is a morphism in $\mathbb{C}$.*

*Further, these mappings must satisfy the following conditions:*

*(F1) $F$ preserves identities, i.e.,*
$$F(id_X) = id_{F(X)}.$$

*(F2) $F$ preserves composition, i.e. for any $f : X \to Y$ and $g : Y \to Z$ in $\mathbb{C}$,*

$$F(g \circ f) = F(f) \circ F(g).$$

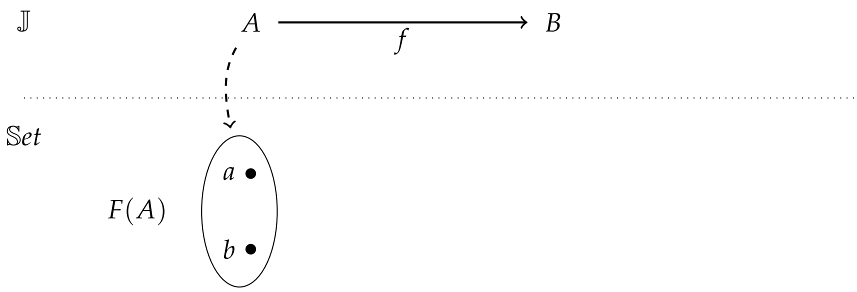
*A functor $F : \mathbb{J} \to \mathbb{C}$ is also called a diagram, because it picks out $\mathbb{J}$-shaped figures of $\mathbb{C}$. $\mathbb{J}$ is then called the indexing category, because the objects and morphisms of $F$'s image are indexed by the objects and morphisms of $J$.*

*A functor to or from an opposite category (e.g., $F : \mathbb{J}^{op} \to \mathbb{C}$ or $F : \mathbb{J} \to \mathbb{C}^{op}$) is called a contravariant functor because the morphisms go in opposite directions on either side of the functor.*

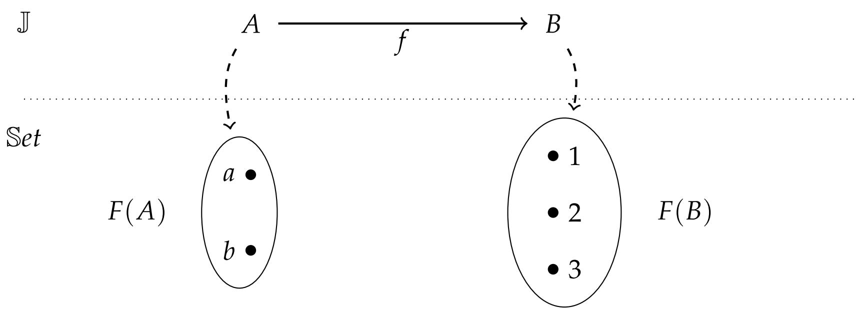
**Example 27.** *Suppose $\mathbb{J}$ consists of two objects and one non-trival morphism, something like this (we do not draw identity morphisms):*

$$\mathbb{J} \qquad\qquad A \xrightarrow[f]{} B$$

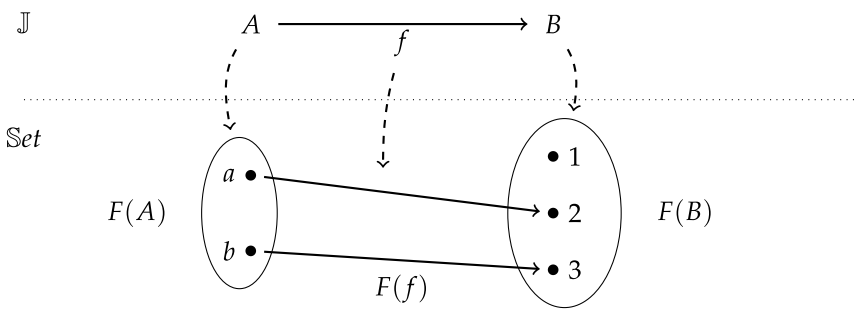
*Then a functor $F$ from $\mathbb{J}$ to $\mathbb{S}et$ uses $\mathbb{J}$ as a kind of template: it picks out a set $F(A)$ for $A$, a set $F(B)$ for $B$, and a function $F(f) : F(A) \to F(B)$ for $f$. For instance, to build such an $F$, for $A$ we might pick the set $\{a, b\}$:*



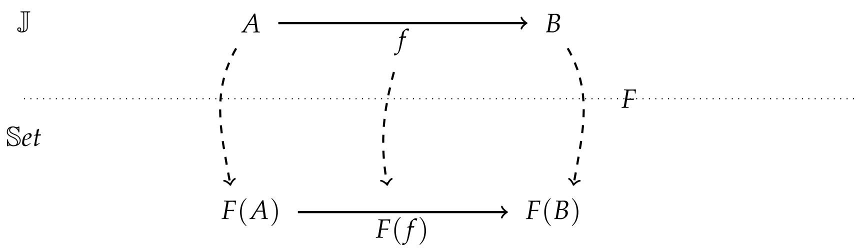

*Then for B we might pick the set* $\{1, 2, 3\}$*:*




*Finally, for $f$ we might pick the function that sends a to 2 and b to* 3*:*




*Or, to draw the same picture without the internal details:*




*This makes it clear that F picks out a $\mathbb{J}$-shaped piece of $\mathbb{S}et$: namely, a piece of $\mathbb{S}et$ that consists of two objects with one non-trivial morphism between them.*

*We could construct a different functor $G : \mathbb{J} \to \mathbb{S}et$ by picking different sets with a different morphism between them, in which case we would thereby pick out a different $\mathbb{J}$-shaped piece of $\mathbb{S}et$.*

**Remark 13.** *Functors do not always pick out* isomorphic *$\mathbb{J}$-figures. A functor can collapse parts of $\mathbb{J}$ and still preserve its identities and composition. For instance, a functor can send all objects of $\mathbb{J}$ to the one-element set $\{*\}$ in $\mathbb{S}et$ and all morphisms of $\mathbb{J}$ to its identity $* \mapsto *$.*

**Remark 14.** *A contravariant functor from an indexing category $\mathbb{J}$ to $\mathbb{S}et$ is a presheaf. The signature of such a functor is $F : \mathbb{J}^{op} \to \mathbb{S}et$. To see why this is a presheaf, consider the following. $\mathbb{J}^{op}$ is* *the base category, and $F$ assigns to each object $U \in \mathbb{J}^{op}$ a set. This is the data over the fiber of $U$. For each morphism $f : V \to U$ in $\mathbb{J}^{op}$, $F$ sends $f$ to a morphism going the other direction: $F(f) : F(U) \to F(V)$. These are the restriction maps. The functor laws F1 and F2 ensure that the restriction maps are unital and transitive as required.*

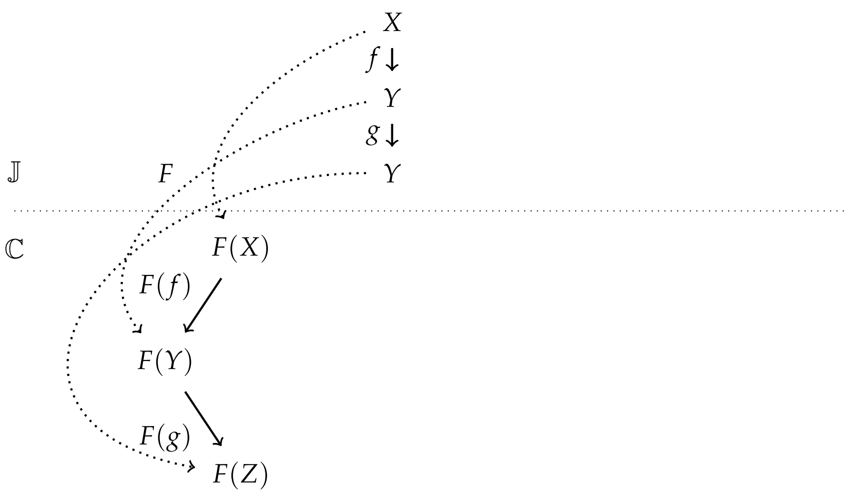
### *6.2. Natural transformations*

A natural transformation maps one $\mathbb{J}$-figure to another. It does this by connecting up all of the objects of the first figure with those of the second in lock-step.

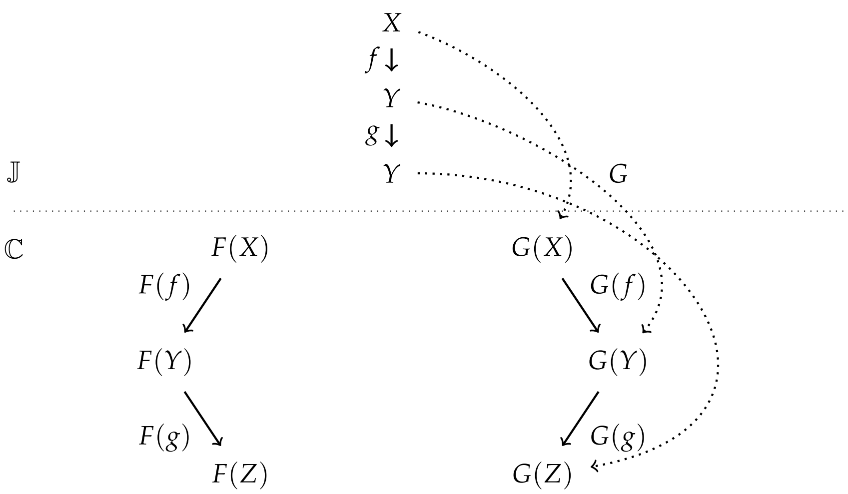
Visually, we can think of constructing a natural transformation as follows. Suppose we have an indexing category $\mathbb{J}$ that looks something like this:

$$\mathbb{J} \qquad \begin{array}{r} X \\ f\downarrow \\ Y \\ g\downarrow \\ Y \end{array}$$

Next, suppose we have a diagram $F$ of $\mathbb{J}$ over on the left:



Then suppose we have another diagram $G$ of $\mathbb{J}$ over on the right:



in other words, we have two diagrams of $\mathbb{J}$ side by side in $\mathbb{C}$:

$$\begin{tikzcd}
& F(X) \arrow[dl, "F(f)"'] & & G(X) \arrow[dr, "G(f)"] & \\
F(Y) \arrow[dr, "F(g)"'] & & & & G(Y) \arrow[dl, "G(g)"] \\
& F(Z) & & G(Z) &
\end{tikzcd}$$

We can map the one diagram ($F$) to the other ($G$) as follows. First, for each object in the left-hand $\mathbb{J}$-figure, pick a morphism that goes over to the corresponding object in the right-hand $\mathbb{J}$-figure. For instance, for the $X$-component, pick a morphism that goes from $F(X)$ to $G(X)$ (call it $\alpha_X$):

$$\begin{tikzcd}
& F(X) \arrow[dl, "F(f)"'] \arrow[rr, dashed, "\alpha_X"'] & & G(X) \arrow[dr, "G(f)"] & \\
F(Y) \arrow[dr, "F(g)"'] & & & & G(Y) \arrow[dl, "G(g)"] \\
& F(Z) & & G(Z) &
\end{tikzcd}$$

Then, for the $Y$-component, pick a morphism that connects $F(Y)$ to $G(Y)$:

$$\begin{tikzcd}
& F(X) \arrow[dl, "F(f)"'] \arrow[rr, dashed, "\alpha_X"'] & & G(X) \arrow[dr, "G(f)"] & \\
F(Y) \arrow[dr, "F(g)"'] \arrow[rrrr, dashed, "\alpha_Y"'] & & & & G(Y) \arrow[dl, "G(g)"] \\
& F(Z) & & G(Z) &
\end{tikzcd}$$

Finally, for the $Z$-component, pick a morphism that connects $F(Z)$ to $G(Z)$:

$$\begin{tikzcd}
& F(X) \arrow[dl, "F(f)"'] \arrow[rr, dashed, "\alpha_X"'] & & G(X) \arrow[dr, "G(f)"] & \\
F(Y) \arrow[dr, "F(g)"'] \arrow[rrrr, dashed, "\alpha_Y"'] & & & & G(Y) \arrow[dl, "G(g)"] \\
& F(Z) \arrow[rr, dashed, "\alpha_Z"'] & & G(Z) &
\end{tikzcd}$$

By doing this, we connect each object from the left-hand figure with an object on the right-hand figure.

A natural transformation is just such a family of connecting wires. It is a family because there are many of them: there is one such connecting wire for each "component" (object) of the figure. Thus, it is a family $\{\alpha_i\}_{i \in Objs(\mathbb{J})}$. We can think of it as a stack of bridges: each "bridge" (morphism) lets us travel from a component in the left-hand figure to the corresponding component in the right-hand figure.

However, we cannot pick just any set of "bridges" and call it a natural transformation. The choice of bridges has to be "natural," which means that our choice of bridges has to keep paths through the two figures in lock-step. In other words, we must be able to travel along a morphism in either figure, and it won't matter if we go over the bridge before or after we go down. We'll get to the same place either way.

For instance, in our picture, if we selected our connecting bridges correctly, then it won't matter if we go down $F(f)$ in the left figure and then go over the bridge $\alpha_Y$, or whether we go over the bridge $\alpha_X$ first and then go down $G(f)$. Either way, we'll get the same results. In other words, this square of the picture must commute (i.e., both paths through the diagram must be equal):

$$
\begin{tikzcd}
& F(X) \arrow[rr, dashed, "\alpha_X"'] \arrow[dl, "F(f)"'] & & G(X) \arrow[dr, "G(f)"] & \\
F(Y) \arrow[rrrr, dashed, "\alpha_Y"'] & & & & G(Y)
\end{tikzcd}
$$

In order to qualify as a natural transformation, our choice of bridges has to be such that *all* such squares in the stack of bridges commute. This requirement is called the naturality condition. It is a fairly strict requirement. It is not always possible to find a natural transformation from one diagram to another.

**Definition 25** (Natural transformations)**.** *Given two diagrams $F, G : \mathbb{J} \to \mathbb{C}$, a natural transformation $\alpha$ is a family of morphisms $\{\alpha_i : F(i) \to G(i)\}_{i \in Objs(\mathbb{J})}$ in $\mathbb{C}$. Each such $\alpha_i$ is called a "component" of $\alpha$, or the "i-component" of $\alpha$.*

*Further, the components of $\alpha$ must be chosen in such a way that they satisfy the following naturality condition. For any morphism $f : X \to Y$ in $\mathbb{J}$, $\alpha_Y \circ F(f) = G(f) \circ \alpha_X$. In other words, for every morphism $f : X \to Y$ in $\mathbb{J}$, the following must commute:*

$$
\begin{tikzcd}
F(X) \arrow[r, dashed, "\alpha_X"'] \arrow[d, "F(f)"'] & G(X) \arrow[d, "G(f)"] \\
F(Y) \arrow[r, dashed, "\alpha_Y"'] & G(Y)
\end{tikzcd}
$$

**Remark 15.** *Given an indexing category $\mathbb{J}$ and another category $\mathbb{C}$, the diagrams (functors) from $\mathbb{J}$ to $\mathbb{C}$ form a category, denoted $\mathbb{C}^{\mathbb{J}}$ or $[\mathbb{J}, \mathbb{C}]$. The objects of the category are the diagrams, and the morphisms are natural transformations. Since functors of the shape $F : \mathbb{J}^{op} \to \mathbb{S}et$ are presheaves, a category of $\mathbb{S}et$-valued diagrams $\mathbb{S}et^{\mathbb{J}^{op}}$ (alternatively written $[\mathbb{J}^{op}, \mathbb{S}et]$) is called a presheaf category, or a category of presheaves.*

### *6.3. Limits*

TODO.

### *6.4. (Co)Terminals, (Co)Products, (Co)Pullbacks*

TODO.

### *6.5. Exponentials*

TODO? I think the fibrational setting is easier to describe. Maybe do that instead.

# 7. Topos Theory

In this section, we cover the parts of topos theory that we will utilize in the rest of the paper. Readers familiar with topos theory can skip this section.

A topos is often described as a category that has enough internal structure to it that you can do "sets"-like reasoning in it. In other words, it has an internal logic that very much resembles the kind of first-order logic that we often model in universes of sets.

However, it is important to recognize that the key feature of first-order logic is its ability to speak about *subsets* of sets. Indeed, in first-order logic, a predicate $P(x)$ is typically said to hold when $x$ belongs to the subset of things that satisfy $P$. This is why one can say that first-order logic really boils down to a system for reasoning about the "parts" of sets (their subsets).

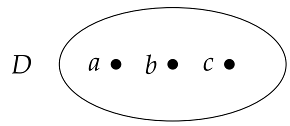
Topos theory generalizes this considerably. Topos theory characterizes categories in which we can reason not just about sets and their "parts" (subsets), but about a great variety of other kinds of objects and their "parts" (subobjects). In the category $\mathbb{S}et$, the objects are sets, and the "parts" or subobjects of a set are its subsets. But in other categories the appropriate "part" or subobject might be different. For instance, in the category of directed multi-graphs, the objects are directed multi-graphs, and the "parts" or subobjects of such a graph are its subgraphs.

Whatever sort of objects they may be, if they are the objects of a topos, then the topos has an internal logic that provides a consistent way to reason about those objects and their "parts" (subobjects). So, it is perhaps more appropriate to say that a topos is a category that has enough internal structure that we can do "parts"-like reasoning in it. A topos is a category that has an internal "parts"-like reasoning system built-in for free.

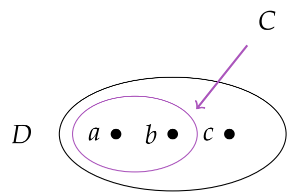
We will not give a definition of a topos, since the details are not important for our purposes. There are a number of different equivalent definitions of a topos, and all are easy to find in the literature (see, for instance, [10], [7], or [12]). Here, we will focus on explaining how notions like predicates and logical connectives are realized in the internal logic of a topos.

## *7.1. Subobjects*

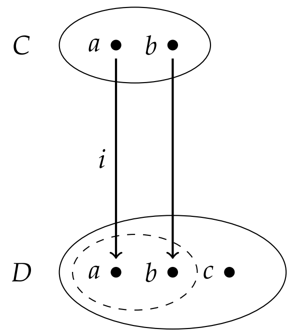
In the category $\mathbb{S}et$, it is natural to think of a "part" of a set as a subset. For example, suppose $D = \{a, b, c\}$:



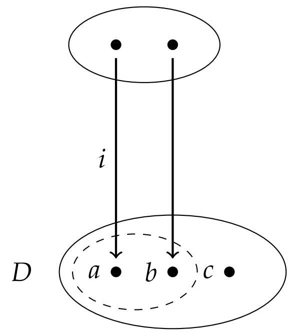

Now consider the "parts" of $D$. For instance, take $C = \{a, b\}$. We can picture the fact that $C$ is a "part" of $D$ by drawing a circle around $C$'s elements to make it clear that the elements of $C$ live "inside" $D$:



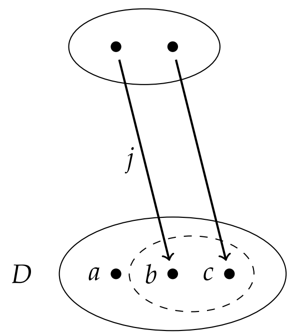

Thinking of $C$ as a "part" of $D$ in this way is fine, but it is very specific to the way that sets work. We can generalize by thinking about $C$ as an insertion map $i : C \to D$ that injects $C$ into $D$:




Once we think of a subset as an insertion map, it becomes clear that the names of the elements in $C$ are irrelevant. In order to pick out the subset $\{a, b\}$, a function from any two-element set that picks out $a$ and $b$ will do:




Further, we can pick another subset of $D$ with a different insertion map. For instance, we can pick out $\{b, c\}$ as follows:




The essential feature of an insertion map is that it does not collapse any information. In other words, it is a mere pass-through: i.e., it keeps things distinct and doesn't equate things that aren't already equated. Or, to put it the other way around, if two things aren't already equal, an insertion map won't make them equal. A morphism that has this property is called a monomorphism.

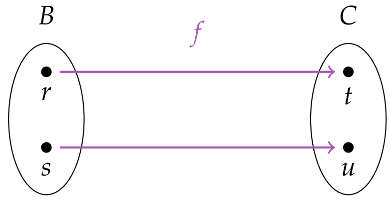
**Definition 26** (Monomorphism)**.** *A morphism $i : C \to D$ in $\mathbb{C}$ is a monomorphism iff:*

$$\forall B \in \mathbb{C} \text{ and } \forall f, g : B \to C \in \mathbb{C}, f \neq g \text{ implies } i \circ f \neq i \circ g.$$

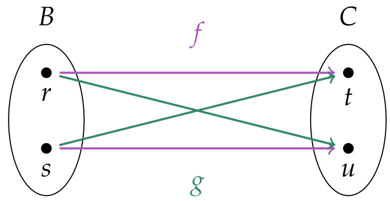
*Equivalently:*

$$\forall B \in \mathbb{C} \text{ and } \forall f, g : B \to C \in \mathbb{C}, i \circ f = i \circ g \text{ implies } f = g.$$

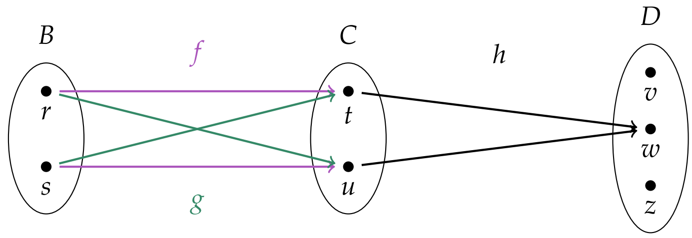
**Example 28.** *Consider a function $f : B \to C$ that looks like this:*




*Suppose we also have another, non-equal function $g : B \to C$:*

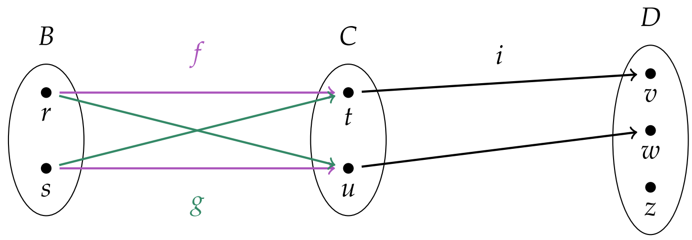



*Since $f$ and $g$ are not equal, they send their inputs to different places. However, if we compose them with a function that collapses points, we can make them send their inputs to the same place. For instance, suppose we follow $f$ and $g$ with the following function $h : C \to D$:*




*Since, $h$ collapses information — that is to say, since it sends both $t$ and $u$ to the same output $w$ — by composing $f$ and $g$ with $h$, we can make them send their inputs to the same place too (namely, $w$). Since we can make a non-equal $f$ and $g$ yield equal results by composing them with $h$, we know that $h$ therefore collapses information, and so $h$ is* not *a monomorphism.*

*Now consider what happens when we compose $f$ and $g$ with a monomorphism $i : C \to D$:*




*Here, we can see that $i$ is just a pass-through: it does not collapse any points but rather simply passes on whatever $f$ and $g$ give it, keeping the inputs it receives distinct. So unlike $h$, $i$ cannot ever help $f$ and $g$ send their inputs to the same place.*

*Further, we can see that $i$ will behave this way with* any *pair of non-equal morphisms $f$ and $g$: if they already result in distinct outputs, $i$ will keep them distinct. That is what it means for $i$ to be a monomorphism.*

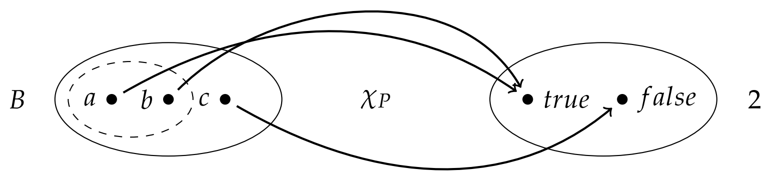
The previous examples were taken from the category $\mathbb{S}et$, but the definition of a monomorphism does not depend on the nature of the sets and functions involved. It works in any category.

Monomorphisms therefore give us a general way to talk about the subobjects of objects, in any category whatever. We can thus identify the subobjects of any object with the monomorphisms into it.

**Definition 27** (Subobject)**.** *Given an object $D$ in a category $\mathbb{C}$, a subobject $i$ of $D$ is a monomorphism $i : C \to D$.*

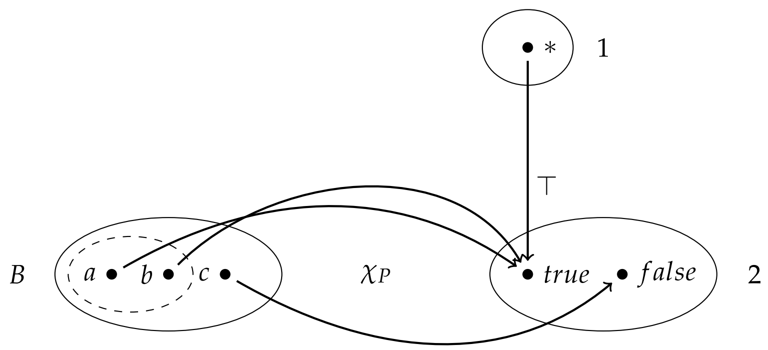
### *7.2. Subobject classifiers*

A predicate $P$ on o domain $B$ is often defined as a function, let's call it $\chi_P$, from $B$ to $2 = \{true, false\}$, where elements of $B$ that satisfy the predicate $P$ are sent to "$true$" and those that don't are sent to "$false$." For instance, suppose we have a set $B = \{a, b, c\}$, and a predicate $P$ with $P(a) = true$, $P(b) = true$, and $P(c) = false$. For concreteness, suppose $B$ is the set of students in a class, and $P(x)$ is the predicate expressing that a student is passing the class. As a picture:

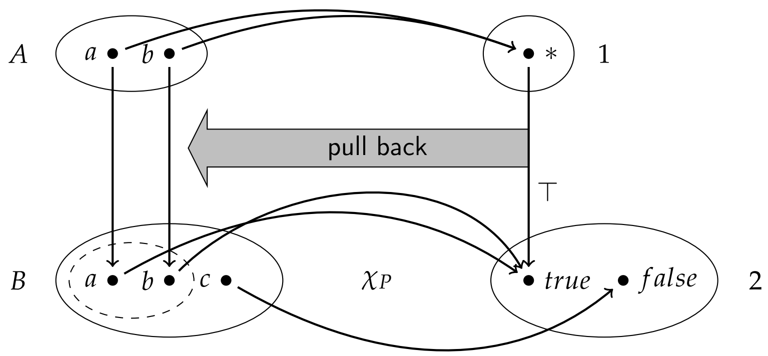



In essence, $\chi_P$ is a characteristic function: it "classifies" a subset $A$ of $B$ by telling us which elements of $B$ belong in that subset $A$ and which ones do not. In this case, the subset that gets classified by $\chi_P$ is $A = \{a, b\}$, i.e., the subset of students who are passing the class. This subset is precisely the extension of the predicate $P$ — it is the set of elements in $B$ that satisfy the predicate $P$, i.e., the set of students in the class who are passing.

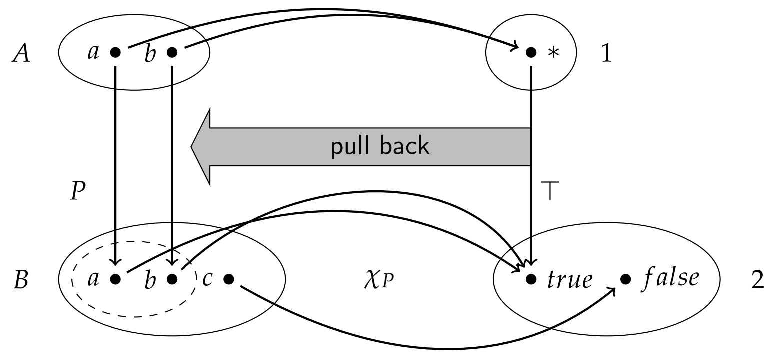
Categorically, we can find $A$ through a pullback. The key feature of a characteristic function like $\chi_P$ is that it sends the elements that it classifies to *true*. So, let's pick out *true* with a morphism (call it $\top$ for "true") from the terminal object 1 to 2:




Next, pull back $\top : 1 \to 2$ along $\chi_P$ to get the preimage of $\chi_P$ on *true*:



This gives us another way to think about the predicate $P$. It is the map on the left-hand side of the pullback square that inserts $A$ into $B$. In other words, it is the subobject (monomorphism) on the left-hand side of the pullback. Let's call it $P : A \to B$:




$P$ and $\chi_P$ give us two perspectives on the same thing. Taken in itself, $P$ identifies a subset of $B$ (e.g., a subset of students in the class). By contrast, $\chi_P$ has more of a logical interpretation: given a student $x$ in $B$, it tells you if they satisfy the predicate $P$ (i.e., it tells you if it is true that they are passing or not). The pullback makes it clear how these two maps — $P$ and $\chi_P$ — are essentially related to truth values.

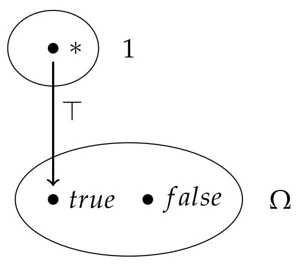
What is crucial about this pullback construction is the object 2 of truth-values on the right, along with the monomorphism $\top$ that picks out the "true" part of 2. In $\mathbb{S}et$, we can pull back $\top$ along any morphism into 2 to identify the corresponding predicate on 2.

This construction generalizes to other categories beyond $\mathbb{S}et$. In other categories too, it often happens that we have an object of truth values, call it $\Omega$, and a monomorphism $\top : 1 \to \Omega$ that picks out the "true" part of $\Omega$, which is such that we can pull back along any morphism into $\Omega$ to get a predicate on it.

**Definition 28** (Subobject Classifier)**.** *In a category $\mathbb{E}$ with a terminal object $1$, a subobject classifier is an object, denoted $\Omega$, along with a morphism $\top : 1 \to \Omega$ which is such that, for any monomorphism $P : A \to B$, there is a unique morphism $\chi_P : B \to \Omega$ that makes the following a pullback:*

$$
\begin{tikzcd}
A \arrow[r] \arrow[d, "P"'] & 1 \arrow[d, "\top"] \\
B \arrow[r, "\chi_P"'] & \Omega
\end{tikzcd}
$$

**Example 29.** *In $\mathbb{S}et$, as we saw, $\Omega$ is the set of boolean truth values $\{true, false\}$, with $\top : 1 \to \Omega$ picking out true:*



*Note that $\top$ is a monomorphism. It picks out a genuine subset of $\Omega$: namely $\{true\} \subseteq \Omega$.*

### *7.3. An Extended Example*

In more complicated categories, the subobject classifier works just as it does in $\mathbb{S}et$, but the set of truth values might be different, and the morphisms may be more complex. This is particularly useful when it is so binary a matter whether something satisfies a predicate.

In this section, we will consider how the subobject classifier works in a category of diagrams. We will omit proofs, and proceed in an expositional style. Readers familiar with subobject classifiers can skip this section.

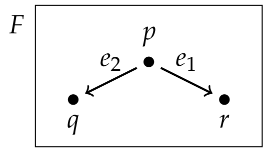
The diagrams we are interested in here are the ones whose indexing category $\mathbb{J}$ has the following shape:

$$E \underset{t}{\overset{s}{\rightrightarrows}} V$$

$\mathbb{S}et$-valued diagrams of this shape (i.e., functors from $\mathbb{J}$ to $\mathbb{S}et$) correspond to directed multi-graphs. Each such functor sends $E$ to a set of edges, it sends $V$ to a set of vertices, and what it picks for $s$ and $t$ send edges to their respective source and target vertices. For instance, let $F : \mathbb{J} \to \mathbb{S}et$ be given as follows:

- $F(E) = \{e_1, e_2\}$
- $F(V) = \{p, q, r\}$
- $F(s)(e_1) = p$, $F(s)(e_2) = p$
- $F(t)(e_1) = r$, $F(t)(e_2) = q$

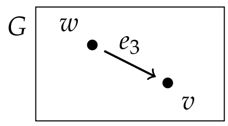
If we draw that graph, we get this:



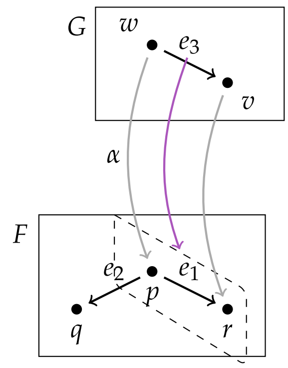

Now suppose we have another functor $G : \mathbb{J} \to \mathbb{S}et$, given as follows:

- $G(E) = \{e_3\}$
- $G(V) = \{w, v\}$
- $G(s)(e_3) = w$
- $G(t)(e_3) = v$

If we draw $G$ as a graph, we get this:




$G$ can be inserted into $F$ as a subgraph. Since $F$ and $G$ are diagrams in the category $\mathbb{S}et^{\mathbb{J}}$, morphisms between them are natural transformations. Hence, an insertion map $\alpha : G \to F$ will be a natural transformation that inserts $G$ directly into $F$ without collapsing any of $G$. There are two such insertions here, but let us pick the one that sends $w$ to $p$, $v$ to $r$, and $e_3$ to $e_1$:




Now that we have picked out a subobject $G$ of $F$, suppose next that we want to define a predicate $P$ on $F$ whose extension is precisely the subobject $G$. For instance, suppose $F$ describes a fork $p$ on a bike path, with $e_1$ going to the park $r$ and $e_2$ going to the museum $q$. Given this interpretation of $F$, let us introduce a predicate $P$ that means something like "is on the route to the park." The extension of this predicate is exactly the subgraph $G$.

But how exactly do we define such a predicate, formally? With sets, a predicate takes a point (an element of a set) as input and then it tells you in response whether that point satisfies the predicate or not. More exactly, it takes a point $x$ and returns "$true$" or "$false$," depending on whether $x$ is in the extension (subobject) of the predicate.

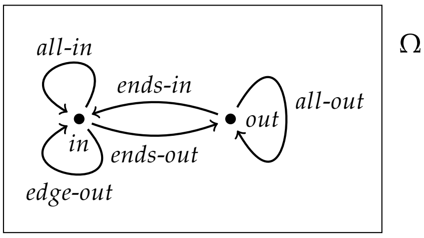
To do something similar for graphs, we have to generalize:

- A predicate on sets takes only *one* kind of input (a point in a set), but a predicate on graphs needs to take *two* kinds of inputs (vertices and edges).
- A predicate on sets tells you if a point is in the *subset* of the predicate's domain, but a predicate on graphs needs to tell you if a vertex or edge is in the *subgraph* of the predicate's domain.
- A predicate on sets gives you a binary "in" or "out" answer (either the point is *in* the predicate's extension or it is *out*), but a predicate on graphs needs to give a more variegated answer:
  - With respect to vertices, a given vertex is either inside or outside of a subgraph.
  - With respect to edges, we have five options:
    * The edge lives entirely outside of the subgraph (its source and target vertices are not part of the subgraph at all).
    * The edge starts outside the subgraph and ends up inside the subgraph (its source is outside the subgraph, and its target is inside the subgraph).
    * The edge starts inside the subgraph and ends up outside the subgraph (its source is inside the subgraph, but its target is outside the subgraph).
    * The edge travels outside the subgraph and then comes back in (its source and target are inside the subgraph, but the edge itself is outside the subgraph).
    * The edge lives entirely inside the subgraph (its source and target vertices are inside the subgraph, and the edge itself stays inside the subgraph and does not travel outside the subgraph).

This makes it clear that the logic of graphs is more discriminating than the logic of sets. With a predicate on sets, if you ask it whether a given input satisfies a predicate, it gives you a simple yes or no answer, because the given element either is or is not inside the given subset. This makes sense, of course. There is no more structure to sets than the points, so there is no more to say about the matter beyond stating whether the given point is or is not in the extension of the predicate.

A graph is different. There is much more structure to a directed graph, and so if you ask a predicate whether a given input (a vertex or an edge) satisfies that predicate, it cannot give you such a simple answer. It can tell you whether a given vertex falls inside the extension of the predicate with a yes/no answer, but if you ask it about an edge, it cannot give you a simple yes/no answer. It must instead tell you the degree to which the edge falls inside the extension of the predicate, since an edge can be partly in, partly out, etc.

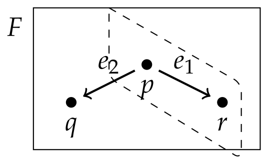
A simple binary set $\{true, false\}$ is thus not sufficient to serve as the object of truth values in this category of directed multi-graphs. Instead, $\Omega$ in this category has to look like the following:



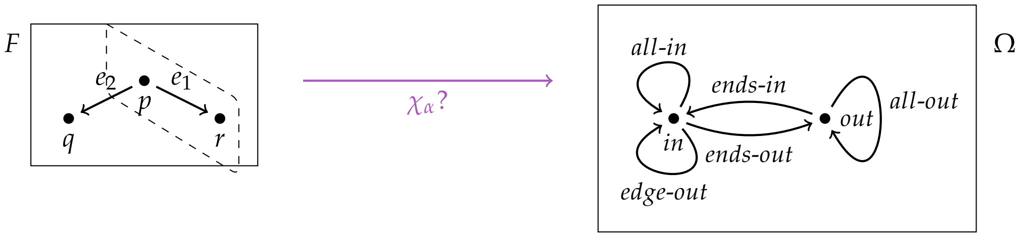

There are two vertices in this graph, labeled "*in*" and "*out*," and they represent whether a given vertex is inside or outside of the given subgraph. There are five edges, each of which represents one of the five options described above:

- The edge labeled "*all-out*" represents an edge that is outside the given subgraph and whose vertices are outside it too. Notice that both its source and target vertices are "*out*," i.e., they are vertices that live outside the given subgraph.
- The edge labeled "*ends-in*" represents an edge that starts outside the given subgraph and ends inside the subgraph. Notice that its source vertex is "*out*" and its target vertex is "*in*."
- The edge labeled "*ends-out*" represents an edge that starts inside the given subgraph and ends outside the subgraph. Notice that its source vertex is "*in*" and its target vertex is "*out*."
- The edge labeled "*edge-out*" represents an edge that starts inside the given subgraph, travels outside the subgraph, and then travels back in. Notice that its source and target vertices are both "*in*."
- The edge labeled "*all-in*" represents an edge that lives inside the given subgraph and whose vertices are also inside the subgraph.

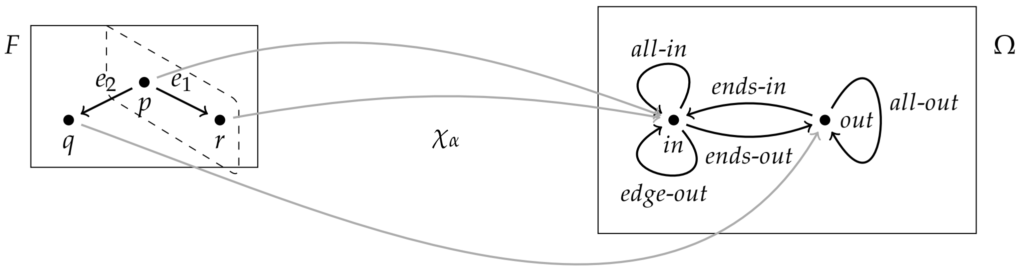
With this version of $\Omega$, we can classify all of the different ways a vertex or edge in a predicate's domain can be part of a a given subgraph. For instance, consider $F$ and the subgraph $G$:




With sets, we can define a characteristic morphism into $\Omega$ that "classifies" which points in the predicate's domain belong in the given subset. But in this setting, how do we define a characteristic morphism $\chi_\alpha$ into $\Omega$ that "classifies" which parts of $F$ belong in $G$?

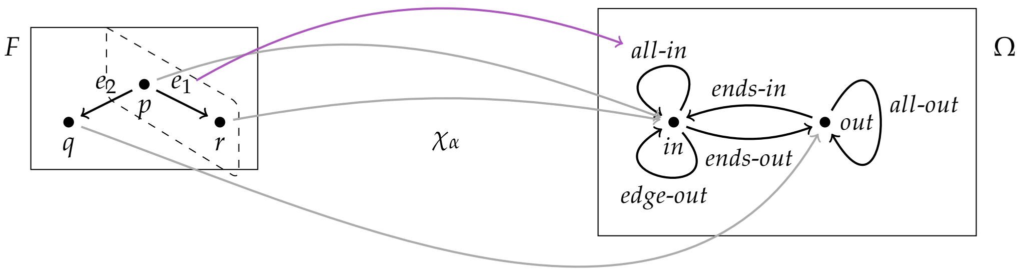



First, we send the vertices $p$ and $r$ to "*in*" (since they are inside $G$) and we send the vertex $q$ to "*out*," since it is outside $G$:



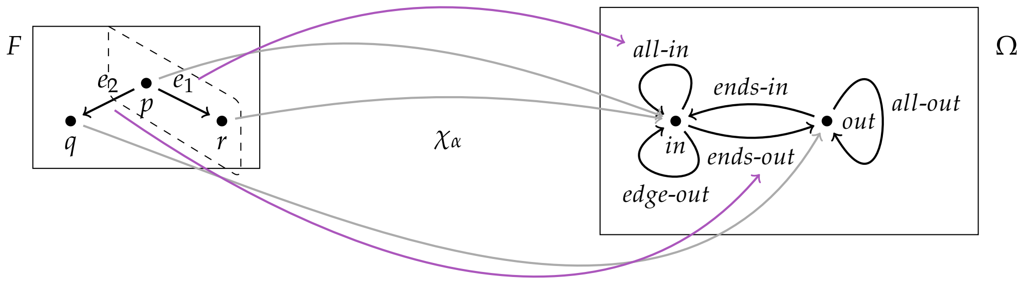

Hence, by this classification, $p$ and $r$ fall under the extension of the predicate, while $q$ does not. This matches what we expect. Is the fork $p$ on the route to the park (i.e., does $P(p)$ hold)? Yes, and the same goes for the park $q$ ($P(q)$ holds). But the museum $q$ is not on the route to the park, so $P(q)$ does not hold.

Next, we send $e_1$ to "*all-in*" because it lives inside $G$:



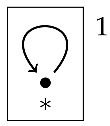

This matches our intuitions about the meaning of the predicate "is on the route to the park" too. Since $e_1$ is the path from the fork $p$ to the park $r$, it makes sense that $P(e_1)$ holds.

Finally, we send $e_2$ to "*ends-out*" because although it begins *inside* of $G$ (at $p$), it leaves $G$ and ends up outside of $G$ (at $q$):



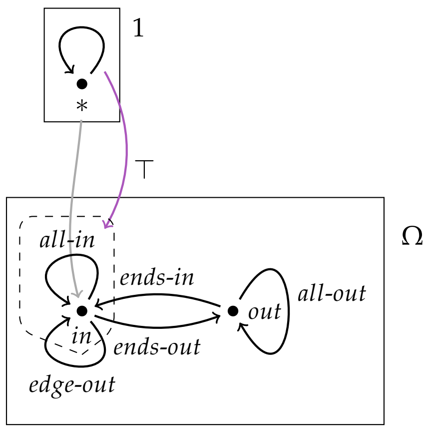

This also classifies $e_2$ appropriately. Does the path to the museum fall on the route to the park? Well, the beginning part of it does, at the fork $p$. But after that, it does not. So $P(e_2)$ does not hold unequivocally. But nor does it fail to hold unequivocally. The edge $e_2$ is classified as being partly in and partly out of the extension of the predicate.

What is the unequivocal "true" part of $\Omega$? That is to say, what is the subobject of $\Omega$ that represents "true" unequivocally? With sets, we pick out the "$\{true\}$" subobject of $\Omega$ with a monomorphism from the terminal object 1. In this category, the terminal object is the one-vertex graph with a single edge:

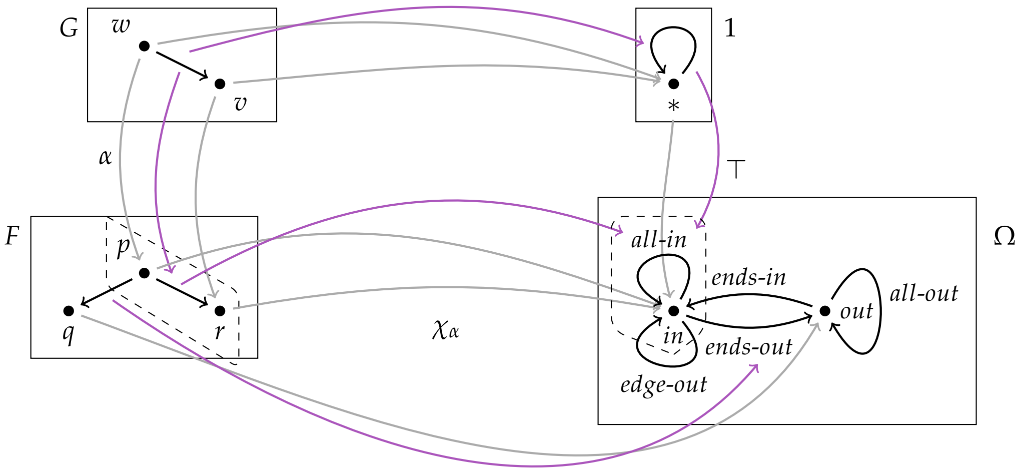



The $\top$ monomorphism from 1 into $\Omega$ has to pick out the "true" subobject of $\Omega$, but in this case that means it will have to pick out the subgraph of $\Omega$ that represents when a given input (vertex or edge) is unequivocally inside the extension of the predicate.

Some of the possible cases represented by $\Omega$ are not entirely "in" the extension of the predicate, e.g. when an edge is partly in, and partly out. The only *unequivocal* case is when an edge and both of its vertices are *entirely* inside the given subgraph. Thus, $\top$ must pick out the subgraph consisting of the vertex labeled "*in*" and the edge labeled "*all-in*":




This is the subobject classifier for the category at hand, and it yields the correct characteristic morphisms when we pull $\top$ back. For instance, in the case of $F$ and $G$:




### 7.4. Logical Connectives

In a topos, predicates are realized as characteristic morphisms into the subobject classifier $\Omega$. A characteristic arrow, say, $\chi_\alpha$, "classifies" or identifies a subobject $\alpha$ as a par tof the predicate's domain. Logical connectives generally take one or more such things and say whether they are true in combination.

For example, suppose we have $P(r)$ and $Q(s)$. The connective "*and*" takes $P$ and $Q$ and tells us whether they are both true together. In a topos, this is realized as a morphism $and : \Omega \times \Omega \to \Omega$.

Specifically, start with the product of $\Omega$, namely $\Omega \times \Omega$. Since "*and*" requires that both conjuncts are true, take the pairing $\langle \top, \top \rangle : 1 \to \Omega \times \Omega$, i.e., the morphism from the terminal object 1 that picks out the "true" part of both sides of $\Omega \times \Omega$:

$$\begin{tikzcd} & 1 \arrow[d, "{\langle \top, \top \rangle}"] \\ \Omega \times \Omega & \Omega \times \Omega \end{tikzcd}$$

This is just like $\top : 1 \to \Omega$, which picks out the "true" part of $\Omega$, except $\langle \top, \top \rangle$ goes to the *product* of $\Omega$, and so it picks out both "true" parts.

Next, suppose we have $\langle \pi_1, \pi_2 \rangle$, where $\pi_1$ is the first projection and $\pi_2$ is the second projection:

$$\begin{tikzcd} & 1 \arrow[d, "{\langle \top, \top \rangle}"] \\ \Omega \times \Omega \arrow[r, "{\langle \pi_1, \pi_2 \rangle}"'] & \Omega \times \Omega \end{tikzcd}$$

Now, pull pack $\langle \top, \top \rangle$ along $\langle \pi_1, \pi_2 \rangle$ to find the pre-image $T$:

$$\begin{tikzcd} T \arrow[r] \arrow[d, "i"'] \arrow[dr, phantom, "\text{pull back}"] & 1 \arrow[d, "\langle \top, \top \rangle"] \\ \Omega \times \Omega \arrow[r, "\langle \pi_1, \pi_2 \rangle"'] & \Omega \times \Omega \end{tikzcd}$$

Since we pulled back from $\langle \top, \top \rangle$, we end up with a monomorphism $i : T \to \Omega \times \Omega$ that inserts exactly the subobject of $\Omega \times \Omega$ that is "true" in both components.

But note that $i$ is itself a subobject, i.e., it inserts $T$ into $\Omega \times \Omega$. In a topos, there is always a unique morphism $\chi_i$ that characterizes this subobject, and we can obtain it by pulling back $\top$. Hence, if we start with the subobject classifier:

$$\begin{tikzcd} 1 \arrow[d, "\top"] \\ \Omega \end{tikzcd}$$

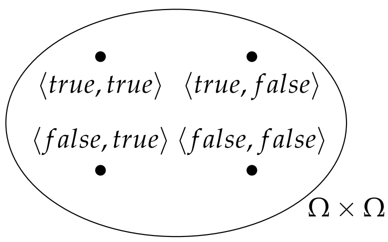
We can then get $\chi_i$ by pulling back to $i : T \to \Omega \times \Omega$:

$$\begin{tikzcd} T \arrow[r] \arrow[d, "i"'] \arrow[dr, phantom, "\text{pull back}"] & 1 \arrow[d, "\top"] \\ \Omega \times \Omega \arrow[r, "\chi_i"'] & \Omega \end{tikzcd}$$

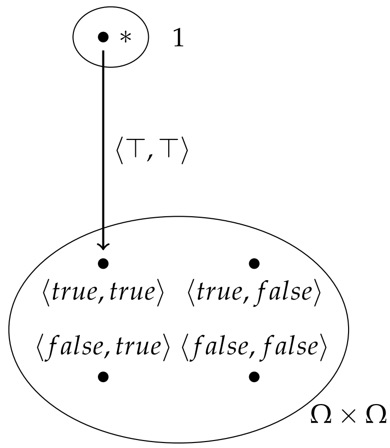
Since $\chi_i$ classifies $T$, and $T$ is the subject of $\Omega \times \Omega$ that has "true" in both components, it follows that $\chi_i$ just is the "*and*" morphism: it characterizes logical conjunction. In other words, we might as well rename $i$ to *and*, since it picks out "both true components," and then we can rename $\chi_i$ to $\chi_{and}$.

**Definition 29** (Conjunction)**.** *Conjunction is classified by $\chi_{and} : \Omega \times \Omega \to \Omega$, where $and : T \to \Omega \times \Omega$ is obtained by pulling back $\langle \top, \top \rangle : 1 \to \Omega \times \Omega$ along $\langle \pi_1, \pi_2 \rangle : \Omega \times \Omega \to \Omega \times \Omega$.*

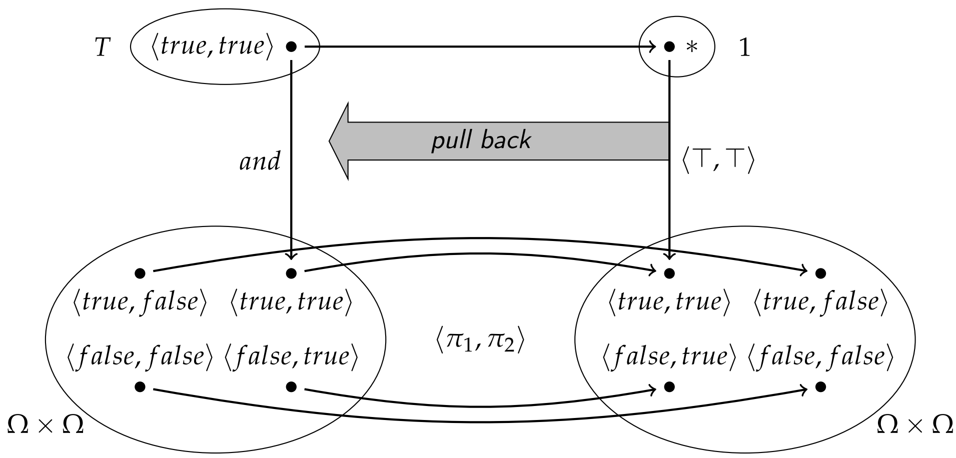
**Example 30.** *In $\mathbb{S}et$, conjunction has a well-known truth table: a conjunction is true when both conjuncts are true. Thus, we need to identify the subobject of $\Omega \times \Omega$ that has both components true. To construct this, start with the product of $\Omega$:*



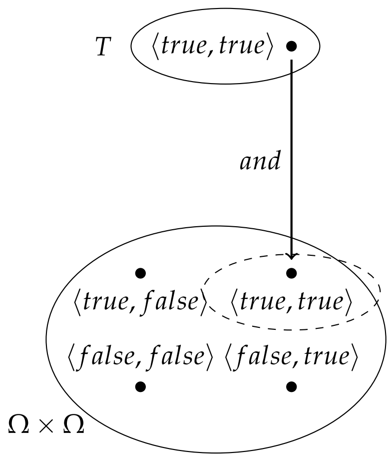
*Use* $\langle \top, \top \rangle$ *from the terminal object* $1$ *to select* $\{true, true\}$*:*



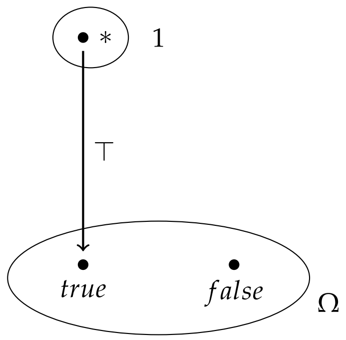

*To get the preimage/subobject of* $\langle \top, \top \rangle$*, pull back:*



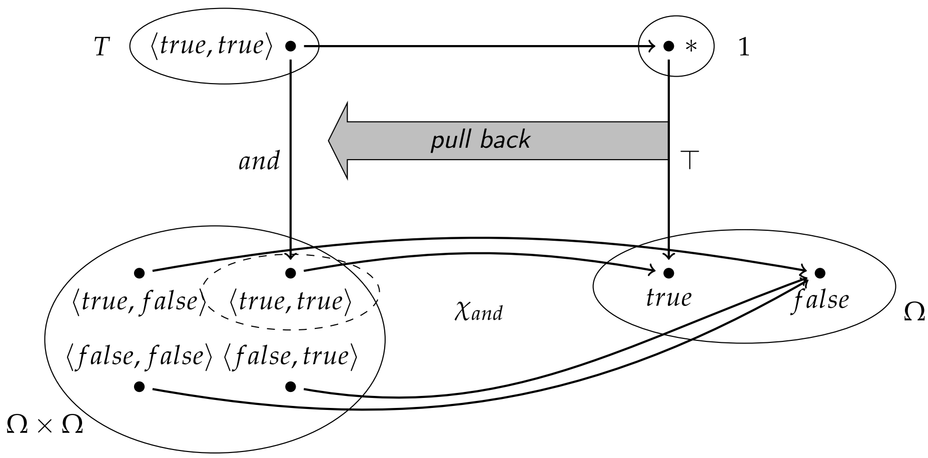

*That yields the subobject/monomorphism that picks out* $\langle true, true \rangle$*:*




*Then, to get the characteristic morphism for "and," take the subobject classifier:*




*Then pull back to get the desired characteristic morphism:*




*This construction can be repeated in any topos to yield the $\chi_{and}$ morphism.*

Other connectives can be constructed in similar ways. We will not go into that here, but for details, see any of the standard references already mentioned.

### *7.5. The Internal Language of the Topos*

TODO

## 8. Mereological Reasoning in Toposes

TODO

### *8.1. Parts and Wholes, Logically*

TODO: define wholes in the internal language, and define the parthood relation in the internal language. Formulate some definitions like overlap.

## 9. Mereological Logics For Free

TODO: prove some theorems like reflexivity, antisymmetry, and transitivity. discuss supplementation? extensionality? atoms?

## 10. Conclusions

TODO

**Author Contributions:** For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used "Conceptualization, X.X. and Y.Y.; methodology, X.X.; software, X.X.; validation, X.X., Y.Y. and Z.Z.; formal analysis, X.X.; investigation, X.X.; resources, X.X.; data curation, X.X.; writing—original draft preparation, X.X.; writing—review and editing, X.X.; visualization, X.X.; supervision, X.X.; project administration,

X.X.; funding acquisition, Y.Y. All authors have read and agreed to the published version of the manuscript.”, please turn to the CRediT taxonomy for the term explanation. Authorship must be limited to those who have contributed substantially to the work reported.

**Funding:** This research received no external funding.

**Acknowledgments:** In this section you can acknowledge any support given which is not covered by the author contribution or funding sections. This may include administrative and technical support, or donations in kind (e.g., materials used for experiments). Where GenAI has been used for purposes such as generating text, data, or graphics, or for study design, data collection, analysis, or interpretation of data, please add “During the preparation of this manuscript/study, the author(s) used [tool name, version information] for the purposes of [description of use]. The authors have reviewed and edited the output and take full responsibility for the content of this publication.”

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Simons, P. *Parts: A Study in Ontology*; Clarendon Press: Oxford, UK, 1987.
2. Varzi, A. Mereology. In *The Stanford Encyclopedia of Philosophy*; Zalta, E., Ed.; Stanford University: Stanford, CA.
3. Cotnoir, A.; Varzi, A. *Mereology*; Oxford University Press: Oxford, UK, 2021.
4. Lewis, D. *Parts of Classes*; Basil Blackwell: Oxford, UK, 1991.
5. Casati, R.; Varzi, A. *Parts and Places*; MIT Press: Cambridge, MA, 1999.
6. Tennison, B.R. *Sheaf Theory*; Cambridge University Press: Cambridge, UK, 1975.
7. Mac Lane, S.; Moerdijk, I. *Sheaves in Geometry and Logic: A First Introduction to Topos Theory*; Springer: New York, 1994.
8. Rosiak, D. *Sheaf Theory through Examples*; MIT Press: Cambridge, MA, 2022.
9. Wedhorn, T. *Manifolds, Sheaves, and Cohomology*; Springer, 2016.
10. Goldblatt, R. *Topoi: The Categorical Analysis of Logic*, 2nd ed.; Elsevier: Amsterdam, 1984.
11. McLarty, C. *Elementary Categories, Elementary Toposes*; Clarendon Press: Oxford, UK, 1992.
12. Borceux, F. *Handbook of Categorical Algebra 3: Categories of Sheaves*; Cambridge University Press: Cambridge, UK, 1994.
13. Lawvere, W.; Schanuel, S. *Conceptual Mathematics: A First Introduction to Categories*; Cambridge University Press: Cambridge, UK, 1997.
14. Van Inwagen, P. *Material Beings*; Cornell University Press: Ithaca, NY, 1990.