Article

Sheaf Mereology

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.)

Received:
Revised:
Accepted:
Published:

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

1. 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 1 (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 2 (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 1. *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 1. *From **??**, 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.

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

Definition 3 (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, $j_H(U_w \wedge U_v) = j_H(U_w) \wedge j_H(U_v)$. $\square$

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_ϵ: $j_H(U_\epsilon) = U_\epsilon$.*

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

Definition 4 (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?

2. 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*.

  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.

  check that we can model continuity in the locale in this way.

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

  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

  cite Normore, Arlig

  .

Todo list