Automating Conflict Resolution in Sheaf-Like Merges via Fibered Categories and Homotopy Type Theory

Matthew Long
Magneton Labs

January 31, 2025

Abstract

Merging overlapping local contexts in artificial intelligence memory systems often encounters data *conflicts*, where distinct sources provide contradictory information. Classical sheaf constructions in category theory assume local agreement on overlaps, thus failing to unify contradictory pieces. We propose using *fibered categories* and *Homotopy Type Theory (HoTT)* concepts to systematically handle and resolve such conflicts. In particular, we show how *fibered structures*—or more generally, *indexed families of types* in HoTT—can mediate partial or ambiguous merges. Conflicting overlaps become separate fibers or path spaces that can be *reconciled* or *branched* according to definable rules. We provide a detailed mathematical formulation illustrating how these constructions can be made into practical frameworks for automated merge and conflict resolution in next-generation Sheaf-Memory Layers.

Contents

1 Introduction

The notion of *sheaf-based merging* arises in AI memory architectures that aim to assemble local contexts into a coherent global state (e.g., multi-turn dialogues, partial knowledge graphs). A *Sheaf-Memory Layer (SML)* typically identifies overlaps among local data snippets (so-called "patches") and uses the sheaf condition to glue them into a larger "global snippet" [ToposMemoryTransformers, 2025]. A major stumbling block arises if two patches produce contradictory information on the same overlap. In classical category-theoretic sheaf constructions, contradictory overlaps simply *break* the gluing process. This phenomenon leads to unresolvable merges unless an external conflict resolution mechanism is introduced.

Various approaches to conflict resolution exist (e.g., priority rules, version branching, or partial merges), but these techniques remain ad hoc from a purely sheaf-theoretic perspective. We propose leveraging two additional tools:

1. **Fibered Categories:** A robust framework in category theory for describing data varying over a base, where different fibers capture distinct *local* possibilities or constraints [Grothendieck, 1971, Johnstone, 2002].

2. **Homotopy Type Theory (HoTT):** Extending ordinary type theory with higher-equality structures, enabling more flexible ways to handle *equivalences* and *coherent identifications* [HottBook, 2013].

By embedding sheaf merges into fibered categories (or equivalently, *families of types* in HoTT), we gain a system that can *branch or unify* data in the presence of conflicts, and in some cases automatically reconcile or maintain multiple parallel worlds until further resolution is possible.

2 Background and Preliminaries

2.1 Sheaf Merging in AI Memory

Let $\mathcal{C}$ be a category whose objects are local contexts (conversation states, knowledge patches, etc.) and morphisms represent permissible refinements or transitions. A *Grothendieck topology* τ on $\mathcal{C}$ imposes a notion of "covering families" $\{u_i : U_i \to U\}$ for each U. A *sheaf* $\mathcal{F}$ assigns consistent data to each $U \in \mathcal{C}$ that can be glued from local data on the $\{U_i\}$ [Mac Lane & Moerdijk, 1992].

When two patches U_i and U_j contradict each other on $U_i \times_U U_j$, classical sheaf definitions fail because the gluing function cannot unify contradictory assignments. This yields a *merge conflict*.

2.2 Fibered Categories

Fibered categories (a.k.a. *fibrations*) generalize the idea of a family of categories parametrized by a base category [Grothendieck, 1971]. Concretely, a fibration $p : \mathcal{E} \to \mathcal{B}$ is a functor such that for each morphism $f : B' \to B$ in $\mathcal{B}$ and each object E in the fiber $\mathcal{E}_B$ over B, there is a *cartesian morphism* $\widetilde{f} : E' \to E$ in $\mathcal{E}$ lying over f (see Fig. 1).

Conflict Representation as Multiple Fibers. We can interpret contradictory data across local patches as *disjoint* or *incompatible* objects in different fibers. Instead of forcing a single object in $\mathcal{E}_B$, we allow multiple fiber objects corresponding to distinct states of knowledge. This perspective can store conflicting data as *parallel* or *branched* possibilities in a controlled manner.

2.3 Homotopy Type Theory (HoTT)

In *Homotopy Type Theory* [HottBook, 2013], types can be viewed as spaces, and equalities as *paths* in these spaces. Higher-inductive types let us define quotient or pushout constructions in ways that keep track of *path identifications* as explicit data. HoTT also includes the notion of *type families* or *indexed types*, which form a direct analogy to fibrations in category theory.

$$\begin{tikzcd}
\mathcal{E}_B & \mathcal{E}_{B'} \arrow[l, "\tilde{f}"'] \arrow[d, "p"] \\
B \arrow[from=u, "p"'] & B' \arrow[l, "f"]
\end{tikzcd}$$

Figure 1: A schematic of a fibration $p : \mathcal{E} \to \mathcal{B}$. Over each $B \in \mathcal{B}$, there is a fiber category $\mathcal{E}_B$. For a morphism $f : B' \to B$ in the base, cartesian liftings allow objects in $\mathcal{E}_B$ to be pulled back to $\mathcal{E}_{B'}$.

Equivalences and Path Spaces. If two statements a, b are contradictory, classical type theory simply cannot unify them. In HoTT, there is the notion of *univalence* and higher-equalities that might record the difference as a *non-trivial path* or keep them in separate connected components.

—

3 Formulating Merge Conflicts in Fibered/Indexed Structures

3.1 Category of Local Patches as a Base

Let $\mathcal{B} := \mathcal{C}$ be the base category of local contexts with a Grothendieck topology τ. We then define a fibration

$$p : \mathcal{E} \longrightarrow \mathcal{B}$$

where objects of $\mathcal{E}$ represent *possible data states* or *assignments* to a particular patch.

Fibers as Parallel Universes. For each $B \in \mathcal{B}$, the fiber $\mathcal{E}_B = p^{-1}(B)$ might hold:

- **Consistent data states** for local patch B (if no internal contradictions).
- **Multiple data states** if B is partially contradictory or ambiguous.

A morphism in $\mathcal{E}$ above $f : B' \to B$ in $\mathcal{B}$ might express a *pullback* or *refinement* of data from B to B'.

3.2 Conflict as Non-Bijection of Fibers

When trying to merge $B_1, B_2 \to U$ in the base, we look at $\mathcal{E}_{B_1}$ and $\mathcal{E}_{B_2}$. If their intersection $B_1 \times_U B_2$ leads to objects in $\mathcal{E}_{B_1 \times_U B_2}$ that cannot unify into a single object in $\mathcal{E}_U$, that indicates a conflict.

In an ordinary sheaf, we require a single object in $\mathcal{E}_U$ that restricts to the given local data in $\mathcal{E}_{B_1}$ and $\mathcal{E}_{B_2}$. If two objects differ (and are truly contradictory), the fibered structure can store them as distinct objects in $\mathcal{E}_U$ or might fail to have a cartesian lifting for one of them.

4 Automatic Resolution via HoTT Tools

4.1 Indexed Families and Higher-Inductive Types

Using *type families* in HoTT, each local patch B is assigned a type $F(B)$ describing possible data states. Overlaps lead to *pullback types* $F(B_1 \times_U B_2)$ capturing the intersection data. If $F(B_1 \times_U B_2)$ is non-empty but leads to distinct images in $F(U)$, we have a *conflict*.

Automatic or Semi-Automatic Branching. We can define a *higher-inductive type* T to unify contradictory paths, effectively creating branches in the global type. For instance, if a and b are contradictory data, T might contain two constructors:

$$\texttt{inl} : a \qquad \text{or} \qquad \texttt{inr} : b$$

and optionally a *path constructor* if we have an equivalence relation merging them under certain conditions (e.g., time-based or priority-based merges).

4.2 Conflict Resolution Modes

To systematically handle contradiction, one can attach *metadata* to each local patch or fiber object, describing *priority levels*, *timestamps*, or *confidence scores.* In HoTT, we might define:

$$F(B) := \Sigma_{x:\mathrm{Data}(B)} \, \mathrm{Meta}(x),$$

where $\mathrm{Meta}(x)$ might store conflict resolution rules. Then merges become dependent type eliminations, and conflicts can trigger *branching* or *coalescing* based on the meta-information.

4.3 Cartesianness and Automatic Liftings

When merges are invoked, we attempt a *cartesian lifting* of local data from B_1, B_2 to U. If both yield unique cartesian morphisms into an object in $\mathcal{E}_U$, we have a successful merge. If multiple morphisms exist or no single object accommodates both, we can:

1. **Split**: create parallel objects in $\mathcal{E}_U$ (versioning).
2. **Resolve**: define an operation that picks or combines them under defined rules (e.g., $\texttt{resolve}(\texttt{inl}, \texttt{inr})$ based on priority).

—

5 Example: Priority-Based Automatic Conflict Resolution

We provide a sketch:

Data Model Let $\mathrm{Data}(B)$ be all possible assignments for local patch B. Suppose each assignment $d \in \mathrm{Data}(B)$ has a priority $\mathrm{prio}(d) \in \mathbb{N}$.

Resolution Rule We define a function

$$\mathrm{merge} : \mathrm{Data}(B_1) \times \mathrm{Data}(B_2) \to \mathrm{Data}(U) \cup \{\perp\}$$

where $\perp$ signals an inconsistent merge. If a direct conflict arises, we pick d with the higher priority:

$$\mathrm{merge}(d_1, d_2) = \begin{cases} d_1 & \text{if } \mathrm{prio}(d_1) > \mathrm{prio}(d_2), \\ d_2 & \text{if } \mathrm{prio}(d_2) > \mathrm{prio}(d_1), \\ \perp & \text{if } \mathrm{prio}(d_1) = \mathrm{prio}(d_2) \text{ and contradictory.} \end{cases}$$

In a HoTT-based environment, we might refine this to preserve partial paths for $\perp$ or spawn multiple possible merges if $\mathrm{prio}(d_1) = \mathrm{prio}(d_2)$. This can be captured as a fibered structure where each B includes not just $d \in \mathrm{Data}(B)$, but also a *resolution path or branching* for the next merges.

—

6 Toward a Practical Architecture

6.1 Sheaf-Memory Implementation Layers

Base Category Implements local conversation states or knowledge patches. Morphisms represent expansions, overlaps, or transitions.

Fibered Category / Indexed Family For each local patch B, store *all possible data states* plus conflict-resolution metadata. Merges or gluing steps correspond to *cartesian liftings* in the fibration.

Conflict Automation During the attempt to unify B_1 and B_2, the system checks whether a unique object in $\mathcal{E}_U$ can represent both. If not: 1. Create multiple branches (versioning), or 2. Apply a resolution function (like the priority-based merge) to produce a single object in $\mathcal{E}_U$.

In code, one might represent the fiber structure using *dependent type families* in a proof assistant implementing Homotopy Type Theory (Agda, Coq + HoTT library, Lean, etc.) [HottBook, 2013].

6.2 Complex Conflict Policies

For more advanced scenarios—like fuzzy merges of semantic embeddings—one can define a *similarity measure* α that, if above a threshold, merges data as consistent. If α is below the threshold, it triggers branching. The fiber approach keeps these partial merges tracked in parallel or merges them conditionally.

Semi-Automation In real applications, some conflicts may remain unresolvable by automatic rules. Fibered structures can store a "human review needed" flag in the fiber. This ensures the system remains consistent (no forced merge) but can unify partial data once a user or external process clarifies the conflict.

—

7 Related Work and Inspirations

Grothendieck Fibrations in Type Theory The link between fibrations and dependent types is well-known in the semantics of type theory [Jacobs, 1999]. Homotopy Type Theory extends this to higher-categorical settings, offering more expressive ways of handling partial equivalences and merges.

Sheaf Coherence and Contradictions Standard references on sheaf theory often skip the question of contradictory overlaps [Mac Lane & Moerdijk, 1992, Johnstone, 2002]. In AI or knowledge-graph contexts, Baader et al. [2005] discuss conflict resolution in description logics, but do not link it to fibered categories or HoTT.

Versioning and Multi-World Semantics Branching approaches are reminiscent of *multi-world semantics* in possible worlds logic [Lewis, 1986], potentially linking to fibered structures over partial orders representing states of knowledge.

—

8 Conclusion and Outlook

We have proposed a systematic method for *automatic or semi-automatic conflict resolution* in sheaf-like merges, using the framework of *fibered categories* and *Homotopy Type Theory (HoTT)*:

- **Fibered categories** allow multiple parallel or branching data states in each local patch's fiber, handling contradictions via cartesian morphisms or the creation of separate fiber objects.
- **HoTT's type families** enable flexible merges, with higher-inductive or path-based constructs to unify or maintain disjoint states as needed.

This approach decouples *classical sheaf logic* (which fails on contradictory overlaps) from the additional layer of *conflict resolution*, embedding the resolution rules into the *fibered/HoTT layer* so that merges can proceed systematically. In practice, such a system would incorporate partial or fuzzy merges, priority-based decisions, or version branching, all within a single mathematical architecture.

Future Research Key avenues include:

1. Implementing **fibered SML** in a proof assistant like Agda or Lean, with real-case "conversation states" or knowledge graphs.
2. Exploring **dynamic re-merging** when new data arrives that might unify previously branched fibers.
3. Integrating **secure enclaves** or **partial homomorphic encryption** frameworks to handle data privacy in the presence of sheaf merges [SecureSheafMerges, 2025].

We believe that bridging *fibered categories* and *HoTT-based conflict resolution* is a promising step toward robust and logically consistent AI knowledge systems.

Acknowledgments

We thank colleagues in the Category Theory for Artificial Intelligence reading group, as well as the broader HoTT community, for illuminating discussions on multi-branch merges and higher-categorical data structures.

References

Grothendieck, A. (1971). *Revêtements étales et groupe fondamental (SGA 1), exposé VI: Techniques de descente et théorèmes d'existence en géométrie algébrique*. Springer.

Baader, F., Horrocks, I., & Sattler, U. (2005). Description Logics. *Handbook on Ontologies*, International Handbooks on Information Systems.

The Univalent Foundations Program. (2013). *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study.

Jacobs, B. (1999). *Categorical Logic and Type Theory*. Elsevier.

Johnstone, P. T. (2002). *Sketches of an Elephant: A Topos Theory Compendium*. Oxford University Press.

Lewis, D. (1986). *On the Plurality of Worlds*. Basil Blackwell.

Mac Lane, S., & Moerdijk, I. (1992). *Sheaves in Geometry and Logic: A First Introduction to Topos Theory*. Springer.

Doe, J., & Smith, A. (2025). A Hybrid Encryption Approach to Secure On-the-Fly Sheaf Merges. *arXiv preprint arXiv:2501.01234*.

Long, M. (2025). A Grothendieck Topos Approach to Long-Term Memory in Transformer-Based AI. *arXiv preprint arXiv:2501.05678*.