From Sets to Parts:
Operational Mereology via Event–Sourced Semantics

Flyxion

December 13, 2025

Abstract

Set theory has long served as the default foundational language of mathematics and computation, encoding structure through element–membership relations. This paper argues that the Spherepop calculus and kernel provide a viable alternative foundation grounded in mereology: a theory of part–whole relations realized through operational, replayable semantics. By replacing extensional membership with event–induced composition, Spherepop avoids several well-known pathologies of set-theoretic foundations while retaining sufficient expressive power for computation, logic, and structural reasoning.

1 The Foundational Role of Sets

Set theory has historically served as the default foundational language of mathematics and, by extension, of formal approaches to computation (Zermelo 1908; Fraenkel 1922; Jech 2003). Within this framework, collections are represented by sets, identity is determined extensionally by membership, structure is encoded through subsets and relations, and abstraction is achieved through comprehension principles.

These roles are not ontologically neutral. The membership relation is global and atemporal: whether an element belongs to a set is treated as a timeless fact, independent of any process by which the set may have been constructed. Extensionality collapses identity to total content, erasing distinctions grounded in construction history or operational provenance. From a logical perspective, this necessitates elaborate axiom systems to control paradoxes arising from unrestricted comprehension. From a computational perspective, it creates a mismatch between formal identity and operational behavior, since real systems evolve through time and record change explicitly.

In practice, modern computing systems rarely manipulate sets in this foundational sense. They operate on state transitions, event logs, transactional records, and replayable histories, all of which are inherently temporal (Gray and Reuter 1993; Fowler 2005). The continued use of set theory as a foundational language is therefore better understood as a historical inheritance than as a reflection of computational necessity.

Lemma 1 *Extensional membership is insufficient to characterize the identity of entities whose structure evolves through time.*

If identity is determined solely by membership, then two entities with identical instantaneous content must be identical. In operational systems, however, entities with identical state may nevertheless differ in provenance, authority, or future evolution. Membership therefore fails to capture distinctions that are preserved in replayable, history-sensitive systems.

2 Mereology as an Alternative Foundation

Mereology replaces the membership relation with a primitive notion of parthood. Rather than asking whether an object is an element of a set, mereology asks whether one entity is part of another. This shift replaces a static inclusion predicate with a relational notion better suited to composition, aggregation, and structural dependence (Leonard and Goodman 1940; Varzi 2023).

Historically, mereology has been proposed as a foundational alternative to set theory, particularly within nominalist programs seeking to avoid commitment to abstract entities such as sets or classes (Leśniewski 1967; Goodman 1951). However, classical mereological systems are typically presented axiomatically and timelessly. They specify general laws governing part–whole relations without explaining how such relations arise, change, or are enforced.

As a result, mereology has often remained a conceptual alternative rather than a practical replacement for set-theoretic foundations. What is missing is an operational account of how parts are introduced, how wholes are constructed, and how mereological facts persist or change over time. Spherepop supplies precisely this missing operational grounding.

Lemma 2 *An axiomatic mereology without a construction semantics cannot serve as a foundation for computational systems.*

Computation requires not only structural relations but also rules governing their introduction, modification, and persistence. A timeless axiomatization of parthood specifies constraints on completed structures but does not explain how those structures are produced or maintained by concrete operations.

3 Spherepop's Ontological Commitments

Spherepop adopts a deliberately minimal ontology grounded in explicit construction. Objects exist only insofar as they have been introduced by events. Relations between objects exist only insofar as they are induced by recorded operations. Events constitute the sole source of authoritative change, and replay provides the mechanism by which the state of the system at any moment is reconstructed.

There is no primitive notion of a set, no global membership relation, and no assumption of a completed universe of discourse. Structure is not postulated; it is accumulated. The meaning of any part–whole claim is determined by whether the replay of the event log witnesses a sequence of operations establishing that relation.

Replay is therefore not an auxiliary interpretive tool but the central semantic mechanism of the system. Ontological facts are inseparable from historical facts. To exist is to have been constructed; to be related is to have been related by an event.

Theorem 1 *Every ontological fact in Spherepop is reducible to a statement about replayed event history.*

By design, objects are introduced only by events, relations are induced only by events, and identity modifications are recorded only by events. Replay deterministically reconstructs all such facts from the event log. No ontological fact is introduced independently of this mechanism.

4 Part–Whole Relations as Event Semantics

In Spherepop, part–whole relations are not treated as primitive axioms but as the cumulative effect of explicit operational events. This places mereology within the semantics of execution rather than within a timeless logical theory. Objects are introduced, related, and sometimes identified through a sequence of recorded operations whose effects persist through replay.

A `POP` event brings a new object into existence. A `MERGE` event records that one object is subsumed by another, thereby establishing a part–whole relation that holds until modified by subsequent events. A `LINK` event introduces structured adjacency without subsumption, allowing for relational structure that does not collapse identity. Finally, a `COLLAPSE` event explicitly identifies objects that were previously distinct, recording an irreversible decision to treat them as one.

The significance of this design is that parthood is not a predicate evaluated against a completed universe, but a historical fact witnessed by replay. Whether an object is part of another at a given moment depends on whether the replay of the event log up to that moment establishes a containment path between them. Mereological facts are therefore time-indexed and contingent on construction history.

This operationalization aligns mereology with computational practice, where structure emerges from execution rather than from axiomatic closure (Fowler 2005; Shapiro et al. 2011).

Theorem 2 *At any replay time t, the part–whole relation in Spherepop is uniquely determined by the prefix of the event log up to t.*

Events are the sole source of change, and replay deterministically applies them in order. Since part–whole relations arise only from the cumulative effect of these events, replay uniquely determines the induced mereology at time t.

5 Replacing Membership with Operational Containment

In set theory, the foundational structural assertion is the membership claim $x \in S$, which holds independently of any construction history. Spherepop replaces this assertion with an operational containment relation $x \leq y$, which holds if and only if replay demonstrates that x has been subsumed by, or structurally subordinated to, $y through a sequence of events$.

This replacement eliminates unrestricted comprehension. No object exists merely because a predicate can be formulated; objects exist only if they have been introduced by events. Likewise, no object possesses parts it did not explicitly acquire. The consequence is a strict alignment between ontology and construction: everything that exists has a witnessed origin.

Importantly, this also breaks the identification of identity with instantaneous structure. Two objects may induce identical part–whole graphs at a given replay point while remaining distinct by virtue of having different handles or different construction histories. Structural coincidence does not entail ontological identity.

This shift echoes nominalist critiques of set-theoretic foundations, which object to the reification of abstract collections without constructive grounding (Goodman 1951; Varzi 2023).

Lemma 3 *Operational containment does not entail extensional identity.*

If two objects have identical induced containment relations at a given replay time, then they are extensionally indistinguishable with respect to parthood. However, unless an explicit identification event has occurred, their distinct handles and event histories remain, preserving ontological distinction despite extensional coincidence.

6 Temporalized Identity

Set-theoretic identity is extensional and timeless: two sets are identical if and only if they have the same members. Spherepop replaces this criterion with a temporally indexed notion of identity grounded in construction and replay. An object is identified by its handle, by the events that introduced and modified it, and by the relations induced at a particular replay time.

Identity claims are therefore sensitive to time. At one replay point, two objects may be structurally indistinguishable; at another, they may diverge due to subsequent events. Explicit identity is asserted only through `COLLAPSE` events, which record a deliberate and authoritative decision to treat two previously distinct objects as one.

This temporalization resolves familiar puzzles associated with extensional collapse, such as the Ship of Theseus, by distinguishing persistence through history from instantaneous structural equivalence (Wiggins 2001).

Theorem 3 *Identity in Spherepop is a historically mediated relation rather than an extensional equivalence.*

Extensional equivalence depends solely on instantaneous structure, whereas Spherepop identity depends on handles and event history. Since event histories may differ even when structure coincides, identity cannot be reduced to extensional properties alone.

7 Authority and Mereological Disagreement

A further feature distinguishing Spherepop from classical foundations is the explicit role of authority. In distributed or collaborative settings, different agents may propose incompatible changes to part–whole structure. Spherepop treats such disagreement as an expected and manageable phenomenon rather than as a logical inconsistency.

Authorities submit proposed events asserting merges, links, or collapses. An arbiter orders these proposals, resolves conflicts, and commits a single authoritative sequence of events to the log. The resulting mereology is therefore not absolute in a metaphysical sense but procedurally determined by governance.

This mechanism replaces the implicit god's-eye perspective of classical set theory with an explicit account of how ontological commitments are made. Disagreement is resolved not by appeal to hidden axioms but by recorded decisions that are visible, inspectable, and replayable.

Theorem 4 *Mereological consistency in Spherepop is enforced procedurally rather than axiomatically.*

Conflicting proposals do not directly alter the event log. Only events committed by the arbiter affect replayed state. Consistency is therefore guaranteed by the ordering and acceptance rules governing event commitment, not by global logical constraints.

8 Interim Synthesis

Taken together, these features yield a mereology that is explicitly constructive, temporally grounded, and authority-aware. Parthood arises from execution, membership is replaced by witnessed containment, identity is historical rather than extensional, and disagreement is resolved procedurally. This reorientation prepares the ground for a reassessment of paradox, complexity, and abstraction, which are addressed in the following sections.

9 Why Russell–Style Paradoxes Cannot Be Formulated

A central motivation for axiomatic set theory has been the need to control paradoxes arising from unrestricted comprehension. RussellâĂŹs paradox is the paradigmatic case. The importance of Spherepop in this context is not that it resolves such paradoxes by modifying axioms, but that it prevents their formulation by eliminating the ontological and syntactic prerequisites on which they depend.

RussellâĂŹs paradox presupposes a universal domain of objects, a primitive membership relation applicable to all objects in that domain, and a comprehension principle allowing objects to be defined by predicates over that domain. On this basis, one defines the set

$$R = \{x \mid x \notin x\}$$

and derives a contradiction by asking whether $R \in R$.

Spherepop admits none of these prerequisites. There is no universal domain of objects, but only those objects that have been introduced by events up to a given replay time. There is no primitive membership relation; the closest analogue is the time-indexed part–whole relation $\leq_t$, which is meaningful only relative to replayed history. Finally, there is no mechanism for predicate-generated existence. Predicates may be evaluated against replayed state, but they do not introduce new objects.

As a result, even the syntactic form of RussellâĂŹs construction is unavailable. The expression $x \leq_t x$ is not paradoxical; it is simply a question about whether replay witnesses a cycle of containment. Such cycles are neither contradictory nor implicit. They can occur only if explicitly introduced by events and are therefore visible and auditable.

Theorem 5 *There exists no Spherepop expression or event sequence corresponding to the Russell set*

$$\{x \mid x \not\leq_t x\}.$$

Any such construction would require quantification over all objects, a primitive containment predicate independent of replay, and a comprehension principle turning predicates into objects. None of these mechanisms exist in Spherepop. Therefore the construction cannot be expressed.

The same analysis applies to other impredicative paradoxes, including the Burali–Forti paradox and ontological readings of CantorâĂŹs diagonal argument. In each case, paradox arises from reifying totality into an object. Spherepop enforces a strict separation between totality as a view and existence as a historical fact.

10 Computational Complexity: Power Sets and Event Logs

The foundational commitments of set theory carry significant computational consequences. The power set axiom asserts that for any set X there exists a set $\mathcal{P}(X)$ containing all subsets of X. Interpreted ontologically, this commits the system to exponential structure regardless of whether such structure is ever observed or used.

From a computational standpoint, this is untenable. Modern systems cannot afford to materialize, or even implicitly commit to, all possible substructures of a given entity. Spherepop replaces this commitment with append-only event logs. Let E_t denote the prefix of the event log up to replay time t. The size of E_t grows linearly with the number of events that have actually occurred. State is reconstructed by replaying E_t, and no object exists beyond those introduced by its events.

Structures that would correspond to subsets or collections in set theory are computed on demand as queries over replayed state. For example, the collection of all parts of an object is not itself an object, but the result of traversing the part–whole graph induced by replay. Computational cost is therefore incurred only when information is requested, not merely because it could be requested.

This approach aligns with established practice in databases, version control systems, and distributed data types, all of which treat logs rather than static collections as the source of truth (Gray and Reuter 1993; Fowler 2005; Shapiro et al. 2011).

Theorem 6 *Let E_t be a Spherepop event log of length n. Then the ontological size of the system is $O(n)$, and all derived structures are computable in time polynomial in n.*

Only events contribute to existence, and there are n such events. All further structure is derived by replay, traversal, or projection over the induced relations. No axiom or operation introduces super-polynomially many entities.

This stands in sharp contrast to power-set semantics, where exponential structure is implicit even when unused. Spherepop relocates complexity from ontology to computation, where it can be controlled, optimized, and reasoned about.

11 Category Theory as a View Layer over Event–Sourced Mereology

Category theory is often proposed as a foundational alternative to set theory because it abstracts away from element-level detail and emphasizes compositional structure (Mac Lane 1998; Lawvere and Schanuel 1997). These motivations resonate with SpherepopâĂŹs rejection of membership and its emphasis on relations. However, category theory typically presupposes a fixed collection of objects, timeless morphisms, and globally valid composition laws.

Spherepop admits categorical structure only derivatively. Given a replay time t, one may define a graph whose nodes are object handles existing at t and whose edges are relations induced by replay. From this graph, a category $\mathcal{C}_t$ may be constructed, with objects corresponding to handles and morphisms corresponding to paths generated by links and merges. Identity morphisms arise from trivial paths, and composition is given by path concatenation.

Crucially, this category exists only relative to replay. Morphisms are historical and contingent, and may be invalidated or altered by later events. Associativity and identity laws hold only insofar as replay produces a stable graph. Categorical axioms are therefore not foundational truths but emergent invariants of well-behaved histories.

Colimits illustrate this distinction particularly clearly. In category theory, colimits are guaranteed by axiom. In Spherepop, a merge corresponds to a colimit only when a `MERGE` event has been explicitly committed. Limits and pullbacks similarly arise as queries over replayed state rather than as new objects, unless deliberately constructed. This mirrors database joins rather than ontological closure.

Functors correspond naturally to views: mappings from the replay-induced category to categories of representations such as tables, graphs, or metrics. These views are time-indexed, non-authoritative, and discardable without ontological loss. Metadata evolution induces coherent transformations between such views, analogous to natural transformations, but these transformations remain subordinate to event authority.

Theorem 7 *Category-theoretic structure in Spherepop is observational rather than ontological.*

Objects and relations exist independently of any categorical interpretation. Categories, functors, and transformations are constructed from replayed state as descriptive artifacts. Removing these interpretations does not alter the underlying ontology.

12 Interim Synthesis

Across paradox, complexity, and abstraction, a consistent pattern emerges. Set-theoretic foundations encode possibility as existence, while Spherepop encodes existence as construction. Classical foundations rely on axiomatic restriction to control inconsistency; Spherepop relies on ontological discipline enforced by event semantics. Where set theory commits to exponential structure by fiat, Spherepop defers complexity to computation. And where category theory postulates universal constructions, Spherepop requires those constructions to be explicitly realized.

These differences are not merely philosophical. They reflect the structural realities of computation itself.

13 Why Russell–Style Paradoxes Cannot Be Formulated

A central motivation for axiomatic set theory has been the need to control paradoxes arising from unrestricted comprehension. RussellâĂŹs paradox is the paradigmatic case. The importance of Spherepop in this context is not that it resolves such paradoxes by modifying axioms, but that it prevents their formulation by eliminating the ontological and syntactic prerequisites on which they depend.

RussellâĂŹs paradox presupposes a universal domain of objects, a primitive membership relation applicable to all objects in that domain, and a comprehension principle allowing objects to be defined by predicates over that domain. On this basis, one defines the set

$$R = \{x \mid x \notin x\}$$

and derives a contradiction by asking whether $R \in R$.

Spherepop admits none of these prerequisites. There is no universal domain of objects, but only those objects that have been introduced by events up to a given replay time. There is no primitive membership relation; the closest analogue is the time-indexed part–whole relation $\leq_t$, which is meaningful only relative to replayed history. Finally, there is no mechanism for predicate-generated existence. Predicates may be evaluated against replayed state, but they do not introduce new objects.

As a result, even the syntactic form of RussellâĂŹs construction is unavailable. The expression $x \leq_t x$ is not paradoxical; it is simply a question about whether replay witnesses a cycle of containment. Such cycles are neither contradictory nor implicit. They can occur only if explicitly introduced by events and are therefore visible and auditable.

Theorem 8 *There exists no Spherepop expression or event sequence corresponding to the Russell set*

$$\{x \mid x \not\leq_t x\}.$$

Any such construction would require quantification over all objects, a primitive containment predicate independent of replay, and a comprehension principle turning predicates into objects. None of these mechanisms exist in Spherepop. Therefore the construction cannot be expressed.

The same analysis applies to other impredicative paradoxes, including the Burali–Forti paradox and ontological readings of CantorâĂŹs diagonal argument. In each case, paradox arises from reifying totality into an object. Spherepop enforces a strict separation between totality as a view and existence as a historical fact.

14 Computational Complexity: Power Sets and Event Logs

The foundational commitments of set theory carry significant computational consequences. The power set axiom asserts that for any set X there exists a set $\mathcal{P}(X)$ containing all subsets of X. Interpreted ontologically, this commits the system to exponential structure regardless of whether such structure is ever observed or used.

From a computational standpoint, this is untenable. Modern systems cannot afford to materialize, or even implicitly commit to, all possible substructures of a given entity. Spherepop replaces this commitment with append-only event logs. Let E_t denote the prefix of the event log up to replay time t. The size of E_t grows linearly with the number of events that have actually occurred. State is reconstructed by replaying E_t, and no object exists beyond those introduced by its events.

Structures that would correspond to subsets or collections in set theory are computed on demand as queries over replayed state. For example, the collection of all parts of an object is not itself an object, but the result of traversing the part–whole graph induced by replay. Computational cost is therefore incurred only when information is requested, not merely because it could be requested.

This approach aligns with established practice in databases, version control systems, and distributed data types, all of which treat logs rather than static collections as the source of truth (Gray and Reuter 1993; Fowler 2005; Shapiro et al. 2011).

Theorem 9 *Let E_t be a Spherepop event log of length n. Then the ontological size of the system is $O(n)$, and all derived structures are computable in time polynomial in n.*

Only events contribute to existence, and there are n such events. All further structure is derived by replay, traversal, or projection over the induced relations. No axiom or operation introduces super-polynomially many entities.

This stands in sharp contrast to power-set semantics, where exponential structure is implicit even when unused. Spherepop relocates complexity from ontology to computation, where it can be controlled, optimized, and reasoned about.

15 Category Theory as a View Layer over Event–Sourced Mereology

Category theory is often proposed as a foundational alternative to set theory because it abstracts away from element-level detail and emphasizes compositional structure (Mac Lane 1998; Lawvere and Schanuel 1997). These motivations resonate with SpherepopâĂŹs rejection of membership and its emphasis on relations. However, category theory typically presupposes a fixed collection of objects, timeless morphisms, and globally valid composition laws.

Spherepop admits categorical structure only derivatively. Given a replay time t, one may define a graph whose nodes are object handles existing at t and whose edges are relations induced by replay. From this graph, a category $\mathcal{C}_t$ may be constructed, with objects corresponding to handles and morphisms corresponding to paths generated by links and merges. Identity morphisms arise from trivial paths, and composition is given by path concatenation.

Crucially, this category exists only relative to replay. Morphisms are historical and contingent, and may be invalidated or altered by later events. Associativity and identity laws hold only insofar as replay produces a stable graph. Categorical axioms are therefore not foundational truths but emergent invariants of well-behaved histories.

Colimits illustrate this distinction particularly clearly. In category theory, colimits are guaranteed by axiom. In Spherepop, a merge corresponds to a colimit only when a `MERGE` event has been explicitly committed. Limits and pullbacks similarly arise as queries over replayed state rather than as new objects, unless deliberately constructed. This mirrors database joins rather than ontological closure.

Functors correspond naturally to views: mappings from the replay-induced category to categories of representations such as tables, graphs, or metrics. These views are time-indexed, non-authoritative, and discardable without ontological loss. Metadata evolution induces coherent transformations between such views, analogous to natural transformations, but these transformations remain subordinate to event authority.

Theorem 10 *Category-theoretic structure in Spherepop is observational rather than ontological.*

Objects and relations exist independently of any categorical interpretation. Categories, functors, and transformations are constructed from replayed state as descriptive artifacts. Removing these interpretations does not alter the underlying ontology.

16 Interim Synthesis

Across paradox, complexity, and abstraction, a consistent pattern emerges. Set-theoretic foundations encode possibility as existence, while Spherepop encodes existence as construction. Classical foundations rely on axiomatic restriction to control inconsistency; Spherepop relies on ontological discipline enforced by event semantics. Where set theory commits to exponential structure by fiat, Spherepop defers complexity to computation. And where category theory postulates universal constructions, Spherepop requires those constructions to be explicitly realized.

These differences are not merely philosophical. They reflect the structural realities of computation itself.

17 Conclusion

This paper has argued that the foundational role traditionally played by set theory can be assumed instead by an operational mereology grounded in event-sourced semantics. Rather than treating collections, membership, and abstraction as primitive, the Spherepop framework replaces them with construction, replay, and part–whole relations that are explicit, temporal, and inspectable.

The central shift is ontological as well as formal. In Spherepop, existence is historical rather than axiomatic; identity is induced by events rather than extensional content; and structure is something that happens rather than something that is postulated. Set-theoretic notions reappear only as derived views over replayed state, useful for analysis but no longer foundational.

By examining the relationship between Spherepop and classical mereology, type theory, set theory, and category theory, this paper has shown that many of the expressive strengths traditionally attributed to set-theoretic foundations are preserved while several of their most problematic features are eliminated. Unrestricted comprehension, power-set ontology, and impredicative self-reference are not repaired by additional axioms; they are rendered inexpressible by design. Consistency arises from constructional discipline rather than logical restriction.

The computational argument reinforces this conclusion. Set-theoretic foundations encode worst-case complexity directly into ontology, committing systems to structures whose size and cost bear no relation to actual use. Event-sourced foundations grow in proportion to what has been constructed rather than what is merely possible. Spherepop therefore aligns foundational commitments with the realities of computation, where history, replay, and authority are unavoidable.

Spherepop does not reject existing mathematical frameworks. Type theory, category theory, and set theory remain indispensable descriptive tools. What changes is their status: they become languages for reasoning about constructed structure rather than sources of existence. Foundations are relocated from abstract universes to executable processes.

In replacing sets with parts, Spherepop does not propose a new metaphysical picture so much as a new discipline of construction. What exists is what has been built. What is known is what can be replayed. And what is abstract is what is derived, not assumed.

A From Zermelo–Fraenkel Axioms to Operational Mereology

This appendix makes explicit the correspondence between the axioms of Zermelo–Fraenkel set theory and the operational mereology implemented by the Spherepop kernel. The purpose is not to refute ZF, but to demonstrate that its axioms either become unnecessary, are realized operationally, or are relegated to derived views.

In ZF, the primitive ontology consists of sets together with the membership relation. In Spherepop, the primitive ontology consists of object handles, event records, a replay operation mapping event prefixes to states, and relations induced by replay. There is no primitive membership relation; all structure is event-induced.

The axiom of extensionality identifies sets by total membership. Spherepop admits no such global extensional equality. Two objects are identical only if they share a handle or if an explicit `COLLAPSE` event equates them. Objects with identical induced parts may remain distinct if their histories differ, making identity historical rather than extensional.

The axiom of the empty set has no analogue. Spherepop has no distinguished empty object; non-existence is represented simply by non-construction. Pairing and union correspond to explicit construction via `POP` and `MERGE` events, with no implicit aggregation. The power set axiom has no counterpart at all. Spherepop deliberately forbids the automatic reification of all possible substructures, preventing both combinatorial explosion and non-constructive existence.

Separation and replacement survive only at the level of views. Predicates and transformations generate queries over replayed state rather than new objects, unless explicitly materialized by events. Foundation is enforced by temporal order: no object may depend on events that occur after its creation, though explicit cycles may be introduced deliberately. Choice is realized operationally through authority-mediated arbitration, where selections among competing proposals are explicit, contextual, and auditable.

Theorem 11 *Every ZF axiom corresponds either to an explicit construction rule, a replay invariant, or a derived view in Spherepop.*

Each axiom is either realized by an event type, enforced by replay order, or replaced by a query over constructed structure. No axiom requires independent ontological postulation.

B A Sheaf–Theoretic Interpretation of Event–Sourced Mereology

Spherepop admits a natural interpretation in sheaf-theoretic terms, which clarifies the relationship between local construction, global consistency, and replay. This interpretation is not foundational but explanatory: it provides a geometric language for reasoning about event-sourced structure.

Let the event log be totally ordered by time, and let prefixes of the log determine states via replay. Consider the poset category whose objects are event prefixes ordered by inclusion. To each prefix, associate the induced mereological structure at that replay point. This assignment defines a presheaf of structures over the event poset.

Restriction maps correspond to replay truncation: restricting a state to an earlier prefix simply forgets later events. Gluing corresponds to the consistency of replay: if two prefixes agree on their overlap, then their induced structures agree on that overlap as well.

The sheaf condition expresses determinism. Given a compatible family of local states induced by overlapping prefixes, there exists a unique global state obtained by replaying their union. This captures, in geometric language, the fact that replay deterministically reconstructs structure from history.

From this perspective, views correspond to sections of derived sheaves, obtained by projecting the base mereological sheaf into representational domains such as graphs, tables, or metrics. Authority corresponds to control over which local sections are permitted to glue, enforcing global consistency through governance rather than through axiomatic constraint.

Theorem 12 *The replay semantics of Spherepop defines a sheaf over the poset of event prefixes.*

Replay restriction defines presheaf structure. Determinism and prefix compatibility ensure unique gluing of compatible local states, satisfying the sheaf condition.

This sheaf-theoretic perspective reinforces the central thesis of the paper. Structure is local, historical, and compositional. Global coherence is achieved not by postulating completed universes, but by enforcing compatibility across constructed parts.

C A BNF Grammar for the Spherepop Calculus

This appendix presents a Backus–Naur Form (BNF) grammar for the Spherepop calculus. The purpose of the grammar is not to define the ontology of Spherepop, which is determined by event semantics and replay, but to specify a concrete surface language for expressing Spherepop programs, logs, and utilities. The grammar therefore describes admissible syntax, while semantic meaning is supplied exclusively by replay.

The grammar is intentionally minimal. It encodes only those constructs that correspond directly to kernel-recognized events and annotations. Higher-level abstractions, views, and derived analyses are treated as external layers and are not part of the core calculus.

C.1 Lexical Structure

We assume an infinite set of identifiers, written as alphanumeric strings beginning with a letter. Literals are restricted to integers, strings, and opaque payloads whose internal structure is not interpreted by the kernel.

Whitespace and comments are ignored except as separators.

C.2 Top-Level Structure

A Spherepop program consists of a sequence of event declarations. Evaluation of a program corresponds to replaying these events in order.

$$\langle \mathit{program} \rangle ::= \langle \mathit{event} \rangle^*$$

The meaning of a program is the state induced by replaying all events in sequence.

C.3 Events

Events are the sole authoritative source of state change. Each event has a type and a fixed arity.

$$\langle \mathit{event} \rangle ::= \langle \mathit{pop} \rangle \mid \langle \mathit{merge} \rangle \mid \langle \mathit{link} \rangle \mid \langle \mathit{collapse} \rangle \mid \langle \mathit{meta} \rangle$$

C.4 Object Introduction

A `POP` event introduces a new object handle into existence.

$$\langle \mathit{pop} \rangle ::= \texttt{POP}\ \langle \mathit{handle} \rangle$$

Semantically, this event asserts the existence of a fresh object identified by the given handle. No object exists prior to such an event.

C.5 Containment and Aggregation

A `MERGE` event asserts that one object becomes part of another.

$$\langle \mathit{merge} \rangle ::= \texttt{MERGE}\ \langle \mathit{handle} \rangle\ \langle \mathit{handle} \rangle$$

The first handle denotes the part, the second denotes the whole. The induced part–whole relation holds at all replay times following the event, unless altered by subsequent events.

C.6 Relational Adjacency

A `LINK` event introduces a relation between two objects without subsumption.

$$\langle \mathit{link} \rangle ::= \texttt{LINK}\ \langle \mathit{handle} \rangle\ \langle \mathit{handle} \rangle\ [\langle \mathit{label} \rangle]$$

The optional label provides a typed or named relation. Links do not imply parthood and do not affect containment unless interpreted by higher-level views.

C.7 Explicit Identification

A `COLLAPSE` event explicitly identifies two objects.

$$\langle \mathit{collapse} \rangle ::= \texttt{COLLAPSE}\ \langle \mathit{handle} \rangle\ \langle \mathit{handle} \rangle$$

Semantically, this event asserts that the two handles are to be treated as identical from this replay point forward. Identity is therefore historical and explicit.

C.8 Metadata Annotation

Metadata events attach non-authoritative annotations to objects or relations.

$$\langle \mathit{meta} \rangle ::= \texttt{META}\ \langle \mathit{target} \rangle\ \langle \mathit{key} \rangle\ \langle \mathit{value} \rangle$$

$$\langle \mathit{target} \rangle ::= \langle \mathit{handle} \rangle \mid \langle \mathit{relation} \rangle$$

Metadata does not introduce new objects or relations. It may influence views, queries, or authority decisions, but has no ontological force unless acted upon by subsequent events.

C.9 Identifiers and Literals

$$\langle \mathit{handle} \rangle ::= \langle identifier \rangle$$

$$\langle \mathit{label} \rangle ::= \langle identifier \rangle$$

$$\langle \mathit{key} \rangle ::= \langle identifier \rangle$$

$$\langle \mathit{value} \rangle ::= \langle \mathit{integer} \rangle \mid \langle \mathit{string} \rangle \mid \langle \mathit{opaque} \rangle$$

C.10 Semantic Discipline

Several constraints are enforced semantically rather than syntactically:

An event may reference only handles introduced by earlier POP events. Replay order induces temporal well-foundedness. No grammatical form permits predicate-generated existence. No construct introduces implicit collections or totalities.

These constraints ensure that the grammar is compatible with the constructive, replay-based mereology developed in the body of the paper.

Theorem 13 *Every well-formed Spherepop program induces a unique replay state and a unique temporal mereological structure.*

The grammar permits only event sequences. Replay is deterministic, and events are the sole source of structure. Therefore each program corresponds to a unique induced state and part–whole relation at every replay time.

C.11 Remarks on Extensibility

The grammar presented here is intentionally minimal. Additional surface constructs, such as macros, control flow, or query syntax, may be layered atop this core without altering its semantics. Such extensions remain non-authoritative unless they ultimately reduce to sequences of kernel-recognized events.

This separation preserves the foundational principle of Spherepop: syntax may be extended freely, but ontology is fixed by replay.

C.12 Summary

The BNF grammar of Spherepop defines a small, explicit, and executable calculus whose primitives correspond directly to kernel operations. By restricting syntax to constructions that have clear operational meaning, the grammar reinforces the central thesis of the paper: existence, structure, and identity arise from events, not from abstraction.

D Small-Step Operational Semantics for the Spherepop Calculus

This appendix derives a small-step operational semantics for the Spherepop calculus specified in Appendix C. The goal is to make precise, in an execution-oriented form, the claim that all ontology in Spherepop is induced by replay. In particular, the semantics is presented as a deterministic transition system on configurations consisting of an event stream and a mutable replay state.

The presentation is intentionally kernel-aligned. It treats events as the only authoritative updates, regards metadata as non-ontological annotation, and makes identity collapse explicit. The semantics is not intended to replace higher-level logics or type systems; rather, it provides the minimal execution substrate on which such systems may be interpreted.

D.1 Syntax (Recalled)

Let Evt denote the set of events generated by the grammar:

$$e ::= \texttt{POP}(x) \mid \texttt{MERGE}(x,y) \mid \texttt{LINK}(x,y,\ell) \mid \texttt{COLLAPSE}(x,y) \mid \texttt{META}(\tau,k,v),$$

where x, y range over handles, ℓ ranges over optional labels, τ ranges over targets (handles or relations), and k, v are metadata keys and values.

A program is a finite sequence $P = e_1 e_2 \cdots e_n$.

D.2 Replay State and Configuration

A replay state is a tuple

$$\Sigma \;=\; \langle H,\; \preceq,\; L,\; \approx,\; M \rangle,$$

whose components have the following intended meaning.

The set H is the set of existing handles. The relation $\preceq \subseteq H \times H$ is a directed relation representing induced containment (parthood or subsumption edges). The relation $L \subseteq H \times H \times \mathsf{Lbl}$ represents labeled links. The relation $\approx \subseteq H \times H$ is an explicit identification relation generated by collapses. The component M is a partial map assigning metadata to targets.

To model representative choice induced by collapse, we assume a total function

$$\mathsf{rep}_\Sigma : H \to H$$

that maps each handle to its chosen representative under $\approx$. Operationally, this can be realized by union-find with a deterministic tie-break rule (for example, smallest handle by a fixed ordering). The semantics below is parameterized by the choice of rep, but determinism is preserved once that choice is fixed.

A configuration is a pair

$$\langle P, \Sigma \rangle,$$

where P is the remaining event sequence to be replayed and Σ is the current state.

The small-step evaluation relation is written

$$\langle P, \Sigma \rangle \;\to\; \langle P', \Sigma' \rangle,$$

and its reflexive transitive closure is written $\to^*$.

D.3 Auxiliary Normalization

Because collapse induces identification, subsequent events should be interpreted modulo representatives. Define normalization of handles in an event with respect to state Σ by replacing each handle x with $\mathsf{rep}_\Sigma(x)$. For brevity, we write $\widehat{e}_\Sigma$ for the normalized form of e under Σ.

Similarly, for any binary relation $R \subseteq H \times H$, define normalization under Σ by mapping all endpoints through rep_Σ and removing duplicates.

D.4 Transition Rules

The rules below define $\to$ by consuming the head event and updating state.

POP.

$$\frac{x \notin H}{\langle \texttt{POP}(x)\ P,\ \langle H, \preceq, L, \approx, M\rangle\rangle \to \langle P,\ \langle H \cup \{x\}, \preceq, L, \approx, M\rangle\rangle}$$

If $x \in H$, the event is either treated as an error or as a no-op depending on kernel policy. For a foundational semantics, it is useful to model it as an error; for an engineering semantics, as an idempotent no-op. We record both options below.

POP (Idempotent Variant).

$$\frac{x \in H}{\langle \texttt{POP}(x)\ P,\ \Sigma\rangle \to \langle P,\ \Sigma\rangle}$$

MERGE.

$$\frac{x, y \in H}{\langle \texttt{MERGE}(x, y)\ P,\ \Sigma\rangle \to \langle P,\ \Sigma[\preceq := \preceq \cup\{(\mathsf{rep}_\Sigma(x), \mathsf{rep}_\Sigma(y))\}]\rangle}$$

This rule records a directed containment edge from the representative of x to the representative of y.

LINK.

$$\frac{x, y \in H}{\langle \texttt{LINK}(x, y, \ell)\ P,\ \Sigma\rangle \to \langle P,\ \Sigma[L\ :=\ L \cup \{(\mathsf{rep}_\Sigma(x), \mathsf{rep}_\Sigma(y), \ell)\}]\rangle}$$

COLLAPSE.

$$\frac{x, y \in H}{\langle \texttt{COLLAPSE}(x, y)\ P,\ \Sigma\rangle \to \langle P,\ \mathsf{collapse}(\Sigma, x, y)\rangle}$$

The function collapse updates $\approx$ by identifying x and y, chooses a representative, and normalizes the remaining relations accordingly. Formally, let $\Sigma = \langle H, \preceq, L, \approx, M\rangle$. Define

$$\approx' = \mathsf{cl}(\approx \cup\{(x, y)\}),$$

where cl is the smallest equivalence relation containing the added pair. Let $\mathsf{rep}_{\Sigma'}$ be the representative function induced by $\approx'$.

Then define

$$\preceq' = \mathsf{norm}_{\mathsf{rep}_{\Sigma'}}(\preceq), \qquad L'\ =\ \mathsf{norm}_{\mathsf{rep}_{\Sigma'}}(L),$$

and update metadata by redirecting targets through representatives:

$$M'\ =\ \mathsf{norm}_{\mathsf{rep}_{\Sigma'}}(M).$$

Finally,

$$\mathsf{collapse}(\Sigma, x, y)\ =\ \langle H, \preceq', L', \approx', M'\rangle.$$

This is the operational content of identity-as-event: collapse is not a derived equivalence, but an executed act that rewrites subsequent interpretation.

META.

$$\frac{\text{the target } \tau \text{ is well-formed in } \Sigma}{\langle \texttt{META}(\tau, k, v)\ P,\ \Sigma\rangle \to \langle P,\ \Sigma[M := M \cup \{(\mathsf{rep}_\Sigma(\tau), k) \mapsto v\}]\rangle}$$

Metadata does not affect $H, \preceq, L, \approx$ and is therefore non-ontological unless later used by authority or view logic.

D.5 Derived Notions: Temporal Parthood

The semantics above defines the induced containment relation after replaying a prefix. For a program prefix $P_{\leq t} = e_1 \cdots e_t$, define the replayed state by

$$\langle P_{\leq t}, \Sigma_0 \rangle \to^* \langle \varepsilon, \Sigma_t \rangle,$$

where Σ_0 is the initial empty state.

Define time-indexed parthood by reachability in the induced containment graph:

$$x \leq_t y \quad \text{iff} \quad \mathsf{rep}_{\Sigma_t}(x) \text{ reaches } \mathsf{rep}_{\Sigma_t}(y) \text{ via } \preceq_t^* .$$

This makes explicit the earlier claim that parthood is a historical fact witnessed by replay.

D.6 Determinism and Replay Equivalence

Theorem 14 (Determinism) *Fix a deterministic representative-selection policy for* rep*. For any configuration* $\langle P, \Sigma \rangle$*, there exists at most one* $\langle P', \Sigma' \rangle$ *such that* $\langle P, \Sigma \rangle \to \langle P', \Sigma' \rangle$*.*

Each rule matches only the head constructor of the event sequence and prescribes a unique state update. The only potential non-determinism arises in representative choice during collapse; once a deterministic policy is fixed, the update is unique.

Theorem 15 (Replay Equivalence) *If two programs share the same event prefix* $P_{\leq t}$*, then replaying that prefix from the same initial state yields the same state* Σ_t*.*

By determinism, evaluation of the shared prefix proceeds via the same unique transition sequence, producing the same state.

D.7 Safety Properties Induced by the Semantics

The small-step semantics enforces the foundational constraints emphasized in the main text.

Lemma 4 (Existence by Construction) *If* $x \in H_t$ *in the replayed state* Σ_t*, then there exists* $i \leq t$ *such that* $e_i = \mathtt{POP}(x)$*.*

The only rule that adds elements to H is `POP`. Therefore membership of x in H_t implies that some `POP`(x) occurred in the prefix.

Lemma 5 (No Predicate-Generated Objects) *There is no derivation step that introduces a new handle except via* `POP`*.*

Inspection of the transition rules shows that only the `POP` rule updates H. All other rules update only relations or metadata.

Lemma 6 (Explicit Identity) *If* $\mathsf{rep}_{\Sigma_t}(x) = \mathsf{rep}_{\Sigma_t}(y)$ *and* $x \neq y$*, then some* `COLLAPSE` *event relating the equivalence classes of* x *and* y *occurred at or before time* t*.*

Handles become identified only by updates to $\approx$, and only `COLLAPSE` modifies $\approx$.

D.8 Summary

The small-step semantics provides a concrete, execution-aligned foundation for the philosophical claims advanced in the paper. It formalizes the principle that ontology is induced by replay, that parthood is time-indexed and constructed, that identity is explicit and historical, and that metadata is non-ontological unless elevated by further events or governance. Higher-level logical, categorical, or sheaf-theoretic interpretations may be layered on top of this semantics, but none are required for the core meaning of the calculus.

References

[1] E. Zermelo. Untersuchungen über die Grundlagen der Mengenlehre I. *Mathematische Annalen*, 65:261–281, 1908.

[2] A. Fraenkel. Zu den Grundlagen der Cantor–Zermeloschen Mengenlehre. *Mathematische Annalen*, 86:230–237, 1922.

[3] T. Jech. *Set Theory.* Springer, 3rd edition, 2003.

[4] S. Leśniewski. On the Foundations of Mathematics. In S. McCall (ed.), *Polish Logic 1920–1939.* Oxford University Press, 1967.

[5] H. S. Leonard and N. Goodman. The Calculus of Individuals and Its Uses. *Journal of Symbolic Logic*, 5(2):45–55, 1940.

[6] N. Goodman. *The Structure of Appearance.* Harvard University Press, 1951.

[7] A. C. Varzi. Mereology. In *The Stanford Encyclopedia of Philosophy*, Winter 2023 Edition.

[8] E. Husserl. *Logical Investigations*, Vol. II. Routledge, 2001.

[9] D. Wiggins. *Sameness and Substance Renewed.* Cambridge University Press, 2001.

[10] J. Gray and A. Reuter. *Transaction Processing: Concepts and Techniques.* Morgan Kaufmann, 1993.

[11] M. Fowler. Event Sourcing. *martinfowler.com*, 2005.

[12] M. Shapiro, N. Preguiça, C. Baquero, and M. Zawirski. Conflict-Free Replicated Data Types. *SSS 2011*, LNCS 6976.

[13] S. Mac Lane. *Categories for the Working Mathematician.* Springer, 1998.

[14] F. W. Lawvere and S. Schanuel. *Conceptual Mathematics.* Cambridge University Press, 1997.