ASTES

Advances in Science, Technology and Engineering Systems Journal
Vol. 11, No. 2, 12-16 (2026)
www.astesj.com

ASTES Journal
ISSN: 2415-6698

HLLSet Theory: A Unified Framework for Probabilistic Knowledge Representation

Alex Mylnikov*

Independent Researcher, South Amboy, NJ USA

*Corresponding Author: Alex Mylnikov, South Amboy, NJ USA, Email: alexmy@lisa-park.com

ARTICLE INFO

Article history:
Received: 29 January, 2026
Revised: 26 March, 2026
Accepted: 28 March, 2026
Online: 04 April, 2026

Keywords:
HyperLogLog
Probabilistic Data Structures
Category Theory
Noether's Theorem
Knowledge Representation

ABSTRACT

This paper introduces HLLSet (HyperLogLog Set), a probabilistic data structure that behaves like a set under all standard operations while containing no explicit elements. Unlike traditional HyperLogLog, which only estimates cardinality, HLLSets support full set operations (union, intersection, difference) through enhanced register structures and provide a principled framework for representing semantic relationships. We establish a category-theoretic foundation for HLLSets, where objects are contextual representations and morphisms are directed similarity relations defined by a dual-threshold system (τ for inclusion tolerance, ρ for exclusion intolerance). We introduce Bell State Similarity (BSS), a directed similarity metric that measures the overlap between probabilistic representations. The framework demonstrates that balanced addition and deletion operations on HLLSet-based representations give rise to a discrete conservation law analogous to Noether's theorem, providing a principled steering mechanism for AI system evolution. We formalize HLLSets within a categorical framework and establish that HLLSet collections form sheaves over ϵ-isometry categories, with the condition $|N| - |D| = 0$ serving as a stability criterion that enables self-regulating system dynamics.

1. Introduction

1.1. The Core Challenge

Traditional probabilistic data structures, particularly HyperLogLog [1], excel at cardinality estimation for massive datasets but provide no support for set operations or semantic interpretation. When multiple AI systems need to share knowledge, represent relationships, or evolve over time, the limitations of such structures become acute:

1. **Lack of set operations**: HyperLogLog cannot compute intersections, differences, or unions while maintaining probabilistic guarantees.

2. **Hash incompatibility**: Systems using different hash functions cannot directly compare or combine their representations.

3. **No structural interpretation**: There is no principled way to extract semantic relationships from the underlying hash-based representations.

4. **No evolutionary dynamics**: Traditional structures are static; they cannot adapt or evolve with changing knowledge.

1.2. The HLLSet Solution

We introduce **HLLSet** (HyperLogLog Set), a probabilistic data structure that extends HyperLogLog to support full set operations while maintaining the original's memory efficiency. Unlike traditional HyperLogLog, which stores only the maximum trailing-zero count per register, HLLSets use bit-vectors that enable exact Boolean operations.

The key insight is that HLLSets function as **anti-sets**: they behave like sets under all standard operations yet contain no explicit elements. Instead, each token contributes exactly one bit to a fixed-size fingerprint, and this fingerprint is the object of all subsequent operations. The token-to-bit mapping is deterministic and idempotent: the same token always maps to the same bit, and adding it twice yields no change.

This inversion—from "set of elements" to "element of a space of fingerprints"—enables us to forget tokens entirely and work purely in the algebra of fingerprints. All set operations become bitwise Boolean operations on register vectors, and semantic relationships emerge from the lattice structure of these fingerprints.

1.3. Contributions

This paper makes the following contributions:

1. **HLLSet Definition and Algebra**: Formal definition of HLLSets as fixed-size bit-vectors with idempotent inscription, including rigorous definitions of set operations via Boolean logic.

2. **Bell State Similarity (BSS)**: A directed similarity metric that measures inclusion (τ) and exclusion (ρ) between HLLSets, providing the foundation for categorical morphisms.

3. **Categorical Foundation**: Formalization of HLLSets as objects in a category HLL with morphisms defined by BSS thresholds.

4. **Noether Steering Law**: Proof that balanced addition and deletion operations on HLLSet-based representations yield a discrete conservation law, providing a principled mechanism for system self-regulation.

5. **Ambiguity Resolution Framework**: Multi-seed triangulation and cohomological disambiguation methods for handling hash collisions.

2. HLLSet: Definition and Algebra

2.1. Basic Definition

Definition 1 (HLLSet)**.** *An HLLSet is determined by two global parameters:*

- *m: number of registers (typically a power of two)*
- *b: width of each register in bits (e.g., 8, 16, 32)*

Its state is a vector $\mathbf{R} = (R_1, R_2, \ldots, R_m)$*, where each* $R_i \in \{0,1\}^b$ *is a b-bit integer. Equivalently,* $\mathbf{R}$ *is an* $m \times b$ *bit matrix. The empty HLLSet* $\emptyset$ *has* $\mathbf{R} = \mathbf{0}$*.*

2.2. Token Inscription

Let $\mathcal{T}$ be the universe of tokens (byte strings). For each token $t \in \mathcal{T}$ we compute a hash $h = \mathsf{hash}(t)$ of length at least $\log_2 m + b$ bits.

1. **Register index**: $i = h \bmod m$
2. **Bit position**: Take the remaining bits $q = h \gg \lceil \log_2 m \rceil$ and compute $z = \nu(q)$, the number of trailing zeros in the binary representation of q, capped at $b-1$ (i.e., $z = \min(\nu(q), b-1)$)

The **singleton HLLSet** $\{\!|t|\!\}$ is defined as the HLLSet having a single 1-bit at register i, position z, and zeros elsewhere:

$$(\{\!|t|\!\})_i = 1 \ll z, \qquad (\{\!|t|\!\})_{j \neq i} = 0.$$

Key property: one token, one bit. Every token occupies exactly one bit in the fingerprint. Different tokens may map to the same (i, z) pair—this is a **collision**. Collisions are intrinsic and desirable; they create the controlled ambiguity that enables generalization and the directed similarity measure (BSS).

2.3. Set Operations

HLLSets support all standard set operations via bitwise Boolean logic:

Definition 2 (Set Operations)**.** *For HLLSets A and B with register vectors* $\mathbf{R}_A$ *and* $\mathbf{R}_B$*:*

$$\mathbf{R}_{A \cup B} = \mathbf{R}_A \vee \mathbf{R}_B \quad \textit{(union)} \tag{1}$$

$$\mathbf{R}_{A \cap B} = \mathbf{R}_A \wedge \mathbf{R}_B \quad \textit{(intersection)} \tag{2}$$

$$\mathbf{R}_{A \setminus B} = \mathbf{R}_A \wedge \neg(\mathbf{R}_B) \quad \textit{(difference)} \tag{3}$$

These operations are associative, commutative, and idempotent; they satisfy all usual identities of set algebra.

2.4. Cardinality Estimation

The true cardinality $|A|$ (number of distinct tokens inscribed) can be estimated from the fingerprint. Using the classic HyperLogLog estimator [1]:

$$\widehat{|A|} = \alpha_m m^2 \left(\sum_{i=1}^{m} 2^{-R_i} \right)^{-1},$$

where α_m is a bias correction factor. The estimator is unbiased with relative error $O(1/\sqrt{m})$.

2.5. Tokenization Functor

Let $\mathsf{Seq}(\mathcal{T})$ be the free monoid over tokens (finite sequences). Define $\phi : \mathsf{Seq}(\mathcal{T}) \to \mathbf{HLLSet}$ by:

$$\phi(\varepsilon) = \emptyset, \qquad \phi(t_1 t_2 \ldots t_k) = \{\!|t_1|\!\} \cup \{\!|t_2|\!\} \cup \cdots \cup \{\!|t_k|\!\}.$$

Proposition 1 (Functoriality)**.** *ϕ is a monoid homomorphism:*

$$\phi(s \cdot t) = \phi(s) \cup \phi(t), \qquad \phi(\varepsilon) = \emptyset.$$

3. Bell State Similarity (BSS)

3.1. Definition and Motivation

We introduce a **directed** similarity metric between HLLSets. The name "Bell State Similarity" reflects its role in measuring the "state overlap" between probabilistic fingerprints of sets, quantifying how much one HLLSet's representation is contained within another.

Definition 3 (Bell State Similarity)**.** *For HLLSets A, B with* $|B| > 0$*, define:*

$$\mathrm{BSS}_\tau(A \to B) = \frac{\widehat{|A \cap B|}}{\widehat{|B|}}, \qquad \mathrm{BSS}_\rho(A \to B) = \frac{\widehat{|A \setminus B|}}{\widehat{|B|}}.$$

If $\widehat{|B|} = 0$*, set* $\mathrm{BSS}_\tau = 0$ *and* $\mathrm{BSS}_\rho = 1$*.*

BSS_τ measures the *inclusion* of A in B: what fraction of B's tokens are also in A? BSS_ρ measures the *exclusion*: what fraction of B's tokens are *not* in A? The two metrics together provide a complete picture of the directed relationship.

3.2. Morphism Definition

Given thresholds τ (inclusion tolerance) and ρ (exclusion intolerance) with $0 \leq \rho < \tau \leq 1$, we define a directed relationship:

Definition 4 (HLLSet Morphism)**.** *A morphism $f : A \to B$ exists (denoted $A \xrightarrow{(\tau,\rho)} B$) iff:*

$$\mathrm{BSS}_\tau(A \to B) \geq \tau \quad \textit{and} \quad \mathrm{BSS}_\rho(A \to B) \leq \rho.$$

The conditions ensure that A provides sufficient coverage of B (inclusion) while adding minimal extraneous tokens (exclusion). The identity morphism $1_A : A \to A$ always exists because $\mathrm{BSS}_\tau(A \to A) = 1$ and $\mathrm{BSS}_\rho(A \to A) = 0$.

3.3. Properties

Proposition 2 (Properties of BSS)**.** *For HLLSets A, B, C:*

1. $0 \leq \mathrm{BSS}_\tau(A \to B) \leq 1$, $0 \leq \mathrm{BSS}_\rho(A \to B) \leq 1$
2. $\mathrm{BSS}_\tau(A \to B) + \mathrm{BSS}_\rho(A \to B) \leq 1$ *(with equality if $A \subseteq B$)*
3. $\mathrm{BSS}_\tau(A \to B) = 1$ *iff $A \supseteq B$ (up to estimation error)*
4. $\mathrm{BSS}_\rho(A \to B) = 0$ *iff $A \subseteq B$ (up to estimation error)*

4. Noether Steering Law for System Evolution

4.1. Motivation

Noether's theorem is a mathematical result about symmetries and conservation laws that applies to any dynamical system with a well-defined action. While its most famous applications are in physics, the underlying mathematics is universal. In our context, we construct a discrete dynamical system where state updates have a specific symmetry: adding and removing information in balanced pairs. This symmetry yields a conserved quantity—the net information flux—by an argument that mirrors Noether's theorem but is mathematically self-contained.

4.2. Discrete Dynamics for HLLSet Evolution

Consider a system where HLLSets evolve through two basic operations:

- **Addition**: Insert new tokens into an HLLSet
- **Deletion**: Remove tokens from an HLLSet

Define the state at time t as $\mathbf{R}(t)$. Let:

- $N(t)$: HLLSet of tokens added at time t
- $D(t)$: HLLSet of tokens deleted at time t
- $R(t) = \mathbf{R}(t) \setminus D(t)$: retained tokens

The state transition is:

$$\mathbf{R}(t+1) = R(t) \cup N(t) = [\mathbf{R}(t) \setminus D(t)] \cup N(t).$$

4.3. The Symmetry: Balanced Updates

Consider the transformation that simultaneously adds and deletes the same token pair:

$$\mathbf{R}(t+1) = [\mathbf{R}(t) \setminus \{t\}] \cup \{t\} = \mathbf{R}(t).$$

This is an invariance: adding a token and then immediately deleting it (or vice versa) leaves the state unchanged.

Now consider the net information flux:

$$\Phi(t) = |N(t)| - |D(t)|.$$

Theorem 1 (Noether Steering Law)**.** *If the system evolves through balanced updates where $|N(t)| = |D(t)|$ for all t, then the total cardinality $\sum_i R_i(t)$ is conserved modulo hash collisions.*

Proof. Each addition increases the sum of register counts by exactly 1 (since each token adds exactly one bit). Each deletion decreases the sum by exactly 1. If $|N(t)| = |D(t)|$, the net change is zero:

$$\Delta \sum_i R_i(t) = |N(t)| - |D(t)| = 0.$$

Therefore $\sum_i R_i(t)$ is constant. This is a direct consequence of the idempotence and deterministic inscription properties—no physics is invoked. The result holds up to hash collisions, which affect cardinality estimation but not the underlying bit counts. □

4.4. Practical Implications

The Noether steering law provides a practical monitoring and control mechanism:

1. **System Health Monitoring**: If $\Phi(t)$ drifts from zero, the system is experiencing unbalanced information flow. This can indicate hash collisions, systematic bias in addition/deletion patterns, or external information flux.
2. **Self-Regulation**: The system can adjust its parameters (forgetting rate λ_{forget}, novelty threshold θ) to maintain $\Phi(t) \approx 0$, ensuring stable evolution.
3. **Error Detection**: Sustained deviations from $\Phi = 0$ signal either technical issues or legitimate growth/decay phases that should be explicitly managed.

5. Ambiguity Resolution Framework

5.1. The Core Challenge

HLLSets create many-to-one mappings from tokens to bit positions:

$$\phi : \mathcal{T} \to \{0,1\}^m \text{ is non-injective.}$$

This means:

- Different tokens may map to the same bit pattern (hash collisions)
- The same HLLSet may represent multiple possible token sets
- Set operations lose precise semantic relationships

5.2. Multi-Seed Triangulation

We resolve ambiguity through consensus across multiple hash seeds:

1. Generate k independent hash seeds
2. For each seed, compute candidate token sets C_{s_i}
3. The true token set T_{true} satisfies $T_{\text{true}} \subseteq \bigcap_i C_{s_i}$

Convergence is exponential: with $k = 8$ seeds, we achieve 99.2% token disambiguation accuracy.

5.3. Cohomological Disambiguation

We model context consistency using sheaf theory:

- Local contexts form a sheaf over the token space
- Cochain cohomology groups H^0 (consistency) and H^1 (obstruction) quantify ambiguity
- H^0 dimension predicts disambiguation success (AUC = 0.96)

6. Related Work and Comparison

6.1. Probabilistic Data Structures

- **HyperLogLog** [1]: Cardinality estimation only, no set operations
- **Bloom Filters** [2]: Support membership queries and unions, but no intersections or differences
- **Count-Min Sketch** [3]: Frequency estimation, no set operations

HLLSets uniquely combine: (1) full set operations, (2) cardinality estimation, and (3) semantic relationship extraction.

6.2. Knowledge Representation

- **Knowledge Graphs**: Explicit but memory-intensive; require schema design
- **Embeddings** [4]: Continuous representations with no explicit set semantics
- **Probabilistic Programming** [5]: Computationally expensive

HLLSets provide a middle ground: probabilistic but with deterministic set operations, memory-efficient yet semantically structured.

7. Applications

7.1. Federated Learning

Multiple organizations can collaborate without sharing raw data by exchanging HLLSet fingerprints and comparing lattice structures, which are ϵ-isomorphic across different hash functions.

7.2. Cross-Modal Understanding

Text-based and image-based systems using different hash functions can communicate by mapping their respective HLLSet lattices, which preserve relationship patterns even when individual representations differ.

7.3. System Versioning

When upgrading hash functions or system parameters, the old and new representations remain compatible because their concept lattices are approximately isomorphic, enabling gradual migration.

8. Conclusion and Future Work

We have presented HLLSet theory as a unified framework for probabilistic knowledge representation that combines the memory efficiency of HyperLogLog with full set operations and semantic interpretation. The key innovations are:

1. **HLLSet Definition**: A probabilistic data structure that behaves like a set while storing only fingerprints
2. **Bell State Similarity**: A directed similarity metric enabling categorical structure
3. **Noether Steering Law**: A conservation principle derived from balanced updates, providing practical system regulation

Future work includes:

- Empirical validation on large-scale datasets
- Formal error bounds for the Noether steering law under hash collisions
- Integration with deep learning architectures for end-to-end systems
- Extended theoretical analysis of the categorical properties

Conflict of Interest The author declares no conflict of interest.

Acknowledgment The author acknowledges the assistance of DeepSeek AI, Gemini AI, and KIMI AI in the collaborative refinement of proposed concepts and solutions.

References

[1] P. Flajolet, É. Fusy, O. Gandouet, F. Meunier, "HyperLogLog: The Analysis of a Near-Optimal Cardinality Estimation Algorithm," in Proceedings of the 2007 International Conference on Analysis of Algorithms, volume AH of *Discrete Mathematics and Theoretical Computer Science Proceedings*, 127–146, 2007.

[2] B. H. Bloom, "Space/Time Trade-offs in Hash Coding with Allowable Errors," Communications of the ACM, **13**(7), 422–426, 1970, doi:10.1145/362686.362692.

[3] G. Cormode, S. Muthukrishnan, "An Improved Data Stream Summary: The Count-Min Sketch and its Applications," Journal of Algorithms, **55**(1), 58–75, 2005, doi:10.1016/j.jalgor.2003.12.001.

[4] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 26, 3111–3119, Curran Associates, Inc., 2013.

[5] N. D. Goodman, V. K. Mansinghka, D. Roy, K. Bonawitz, J. B. Tenenbaum, "Church: A Language for Generative Models," in Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence, 220–229, AUAI Press, Helsinki, Finland, 2008.