The Phaneron as an Abstract Data Type: Formal Definition, Expressivity, and Algorithms

Anonymous Preprint

November 15, 2025

Abstract

We give a precise mathematical/computer-science formulation of the *Phaneron*: a single-layer, unlabeled graph substrate in which *distinctions* (nodes) and *connections* (edges) are the only primitives, and all nuance (roles, n-ary relations, direction, time) is represented as *patterns* (small subgraphs) rather than labels or weights. A Phaneron system is an abstract data type (ADT) $\mathcal{P} = (G, \mathcal{E}, \mathcal{D}, \mathcal{R}, \mathcal{L})$ comprising an unlabeled undirected pseudograph G, a replayable event log $\mathcal{E}$, a pattern dictionary $\mathcal{D}$, a finite set of graph-rewrite rules $\mathcal{R}$, and an MDL-style objective $\mathcal{L}$ that scores states and rewrites. We formalize meaning as structure (ideal: automorphism/orbits; operational: r-hop neighborhoods/WL), define admissible rewrites as those that reduce description length without harming prediction (static or future versions), and show how direction/roles/time arise inside patterns while the base substrate remains unlabeled. We sketch encodings for labeled KGs/RDF and finite first-order structures, and outline Turing-completeness via pattern rewriting. Finally, we present algorithms and complexity considerations for matching, WL maintenance, dictionary induction, and versioned replay, and discuss how Phaneron-Store realizes this ADT at scale.

1 Introduction

The Phaneron imposes a single structural layer: nodes carry only distinctness; edges carry only connectivity; all other semantics are reusable patterns (unlabeled subgraphs). This design avoids "semantic leaks" into labels/weights, enabling uniform search, rewrite, and compression across the whole structure. Patterns are chosen because they *compress* the graph while *improving prediction* of held-out structure and future edits (version time).

2 Axioms and the ADT

Axioms (A1–A7, condensed). Nodes exist iff distinct (no labels). Edges exist iff connected (no labels/weights). All nuance is represented exclusively as patterns (single-layer principle). Base graph is an undirected pseudograph (loops and parallel edges permitted with no default semantics). Relations/roles/direction/time are reified as intermediate subgraphs. All changes are versioned delta events (add/remove/merge/split/replace) recorded as first-class structure. Node merges are permitted under structural criteria and are replayable/reversible.

Definition 1 (Phaneron ADT). *A* Phaneron system *is a tuple* $\mathcal{P} = (G, \mathcal{E}, \mathcal{D}, \mathcal{R}, \mathcal{L})$ *where*

- $G = (V, E)$ *is a finite or countable unlabeled undirected pseudograph;*
- $\mathcal{E}$ *is a replayable event log over edits and pattern-rewrites;*
- $\mathcal{D}$ *is a multiset of finite unlabeled patterns with placements in* G*;*
- $\mathcal{R}$ *is a set of admissible rewrite schemas (add/remove/merge/split/replace);*
- $\mathcal{L}$ *is a description-length objective possibly augmented by predictive risk.*

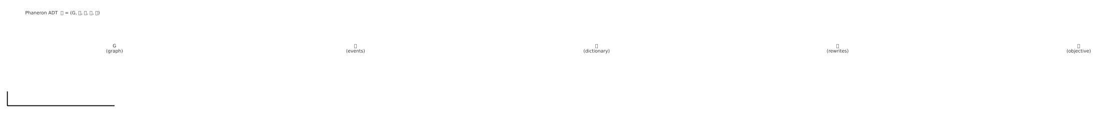

Figure 1: **Phaneron ADT.** $\mathcal{P} = (G, \mathcal{E}, \mathcal{D}, \mathcal{R}, \mathcal{L})$. G is unlabeled/undirected; $\mathcal{E}$ is a versioned event log; $\mathcal{D}$ is a dictionary of unlabeled patterns; $\mathcal{R}$ are admissible rewrites; $\mathcal{L}$ is an MDL-style objective.

3 Patterns and Meaning

Definition 2 (Patterns and instances)**.** *A* pattern *P is a finite unlabeled undirected graph. An* instance *of P in G is an injective homomorphism $f : P \hookrightarrow G$ that is an isomorphism onto its image. A dictionary $\mathcal{D} = \{P_i\}$ describes G as a composition of pattern instances plus residual edges.*

Meaning as structure (ideal). The ideal meaning of $v \in V$ is its orbit under $\mathrm{Aut}(G)$ (equivalently, the set of rooted neighborhoods up to isomorphism).

Operational meaning (local invariants). Let $N_r(v)$ be the r-hop neighborhood of v. We use WL color refinement to approximate structural equivalence: equal colors at step $t^*(r)$ serve as a practical certificate of indistinguishability for many local pattern queries; ties can be broken by increasing r or using k-WL.

Objective (MDL). We use a two-part code $L(G; \mathcal{D}, B) = L(\mathcal{D}) + L(B) + L(\text{usages} \mid \mathcal{D}, B, G)$ with optional coarse block structure B. A rewrite $S \Rightarrow P$ (pattern-replace or merge) is *admissible* if it strictly reduces L and does not worsen held-out recovery of masked structure nor prediction of future edits.

4 Relations, Roles, Direction, and Time as Patterns

Relations (binary or n-ary), role distinctions, and apparent edge directions are encoded by *intermediate subgraphs* that position participants structurally; asymmetry and precedence live *inside the relation pattern.* A stable directed regularity $A \to B$ wins when its asymmetric pattern reduces usage entropy and improves prediction versus a symmetric alternative.

5 Rewrite Semantics and Versioning

We adopt double-pushout (DPO) rewriting on unlabeled undirected graphs. A rule $L \leftarrow K \to R$ matches $m : L \to G$; we delete $m(L \setminus K)$ (no dangling edges) and add $R \setminus K$ along K. Each application appends an event to $\mathcal{E}$; snapshots select all content before a version cut. Confluence is not required; the event log and $\mathcal{L}$ arbitrate among alternatives.

6 Expressivity and Translations

6.1 Labeled graphs / RDF / KGs

A labeled edge or predicate becomes a small unlabeled relation pattern with role positions; reification, provenance, and time are structural attachments to the instance. This preserves single-layer purity while capturing mainstream KR constructs.

6.2 Finite first-order structures

Given a finite FO structure $\mathcal{M}$ over domain U and relations $R_i \subseteq U^{k_i}$, encode each $u \in U$ as a node and each relation instance as a relation pattern linking its participants with role positions. Unary predicates become membership in a pattern class. Queries over $\mathcal{M}$ correspond to pattern queries over G.

6.3 Computation sketch (Turing-completeness)

A register machine (or λ-calculus) can be encoded via a fixed set of control/data patterns and a finite rewrite set $\mathcal{R}$ that simulates instructions (or β-steps). Thus, Phaneron with DPO rewriting is at least as expressive as standard rewriting systems; full proof left to future work.

7 Algorithms and Complexity

Pattern matching. Subgraph isomorphism is NP-complete; practical engines use bounded-size patterns, indexing, and WL-based pruning.

WL maintenance. Cache WL colors up to small t with local invalidation on edits; complexity near-linear per r-hop shell in sparse graphs.

Dictionary induction. Greedy mining of frequent subgraphs with MDL scoring; merge candidates from WL-equivalent neighborhoods across a stability window.

Admissible rewrites. Compute local ΔL using precomputed code lengths and usage counts; verify static held-out and version-time prediction locally.

Versioning. Append-only delta log with checkpoints; snapshots select content prior to cut; long-lived edge presence can be intervalized.

8 ADT Interface and a Realization at Scale

8.1 ADT operations

Core primitives: `add_node`, `add_edge`, `merge(u,v)`, `pattern_replace(S→P)`, `neighbors_r(v,r)`, `wl_color(v,t)`, `snapshot(t)`, `replay(t_0`$\to t\_1).Invariants : single{-}layersemantics(nolabels/weights); all$

8.2 Realization (sketch)

`An ID-less, content-addressed state DAG deduplicates neighborhood states across time; pattern-first adjacency materializes edges implied by dictionary placements on access; residuals persist physically. Hot tiers use bitmap containers; cold tiers use succinct integers; reference compression copies from prior states with small add/remove lists; structural reordering improves compression and locality.`

9 Examples

Direction as pattern. `A repeated precedence motif between two clusters produces a better MDL code than a symmetric alternative; prediction prefers the asymmetric pattern.`

Polysemy as neighborhoods. A single surface form participates in distinct neighborhoods; senses are neighborhoods, not labels.

10 Limitations and Open Problems

Global isomorphism and unrestricted subgraph matching are hard; WL and bounded-radius tests are approximations with known counterexamples. MDL objectives require choices (code families, block models) and approximations. We have sketched but not proven Turing-completeness; categorical formulations and type systems for patterns are future work.

11 Stagewise Reflection and Singularities

Reflection parity is not a fixed point but a *stagewise* target. Let W_t be the micro-world, $\pi_{B,G}(t)$ the task-indexed quotient under resource bound B and goals G, and P_t the current Phaneron.

- **Stage** k: an interval $[t_k, t_{k+1})$ where there exists a homomorphism $h_k : P_t \to \pi_{B,G}(t)$ with task error $\leq \varepsilon(B)$ and P_t is MDL-minimal under the equilibrium objective.
- **Singularity at** t_{k+1}: the smallest time where no sequence of local refinements of $P_{t_{k+1}^-}$ can keep task error $\leq \varepsilon(B)$ without (i) raising capacity B, (ii) narrowing G, or (iii) introducing new invariants (a partition re-factor). Equivalently, the optimal partition changes topology/cardinality:

$$\mathcal{P}(B^-, G) \ncong \mathcal{P}(B^+, G) \quad \text{or} \quad |\mathcal{P}(B^-, G)| \neq |\mathcal{P}(B^+, G)|.$$

Predictability horizon. The horizon H at state (P_t, B, G) is the largest τ such that all task queries within $[t, t+\tau]$ admit bounded regret under the current partition; beyond H, any reliable forecast requires a partition transition (capacity increase or new invariants).
Precursors and a practical score. As a singularity approaches, we typically observe: (i) rising conflict curvature despite consolidation, (ii) increasing residual variance and autocorrelation in forecast errors, (iii) accelerated split/merge churn and codebook drift, (iv) longer/variable MES and message-size spikes in multi-agent settings, and (v) a stall in reflection-distance improvement. A simple trigger uses a weighted score $S(t)$ over these signals and initiates a controlled re-factor when $S(t) > \tau$.
Consequences. Intelligence growth is piecewise: long plateaus of reflection parity punctuated by singularities when tasks/evidence demand new invariants. This explains "unknown unknowns" pre-transition, collective communication cliffs when teams align a finer partition, and subjective time shifts when cognitive debt is reduced across a transition.

12 Conclusion

The Phaneron ADT gives a single-layer, unlabeled substrate where all semantics are structural and all computation is pattern rewrite guided by compression and prediction. It unifies representation and learning in one object, admits translations from mainstream KR, and maps to a scalable implementation. We hope this specification helps others build reasoning systems on top without labels leaking into the base.

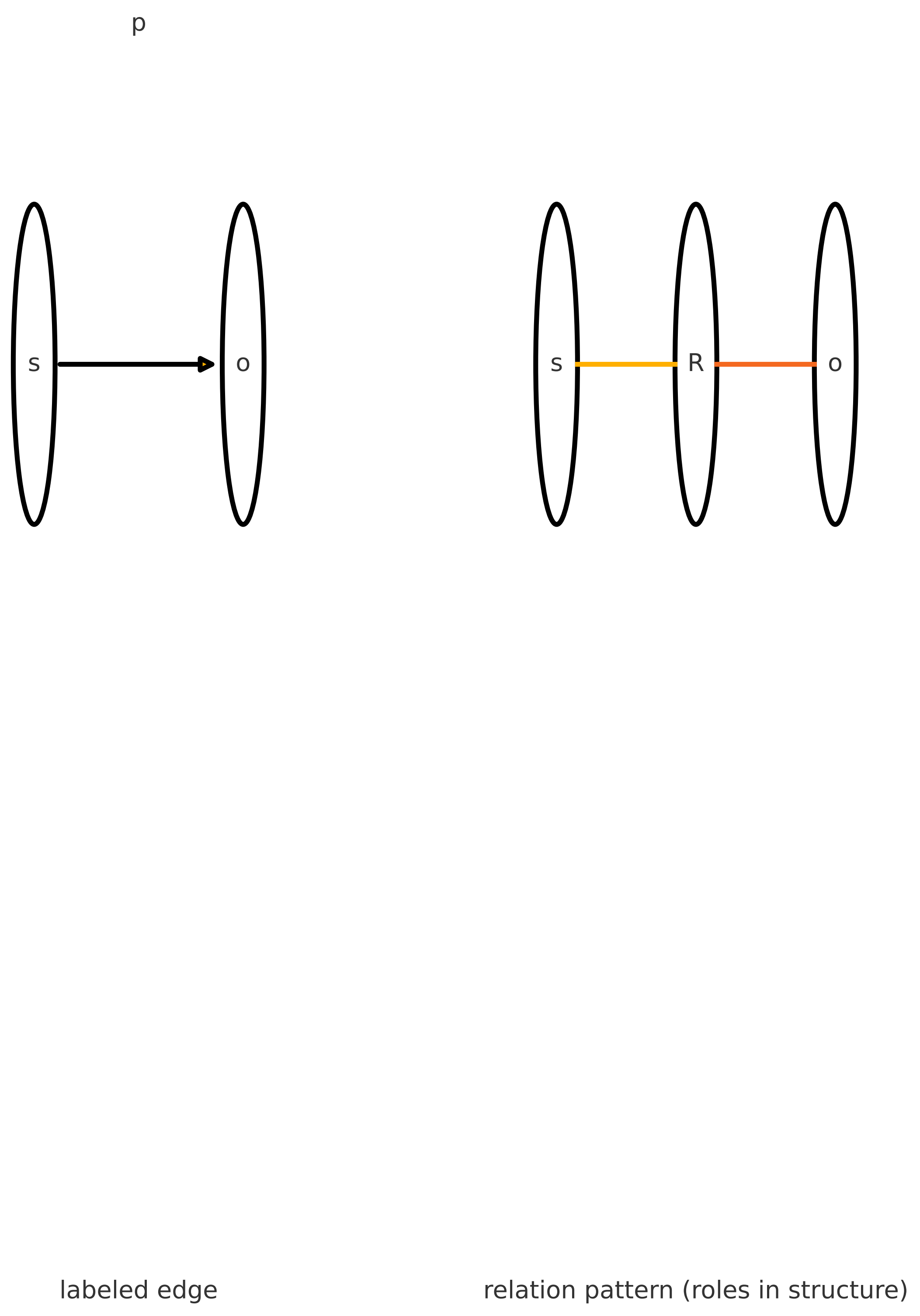

Figure 2: **From labeled triples to patterns.** A labeled edge $s \xrightarrow{p} o$ is represented by an unlabeled intermediate node R with role positions; direction and roles are structural, not labels.

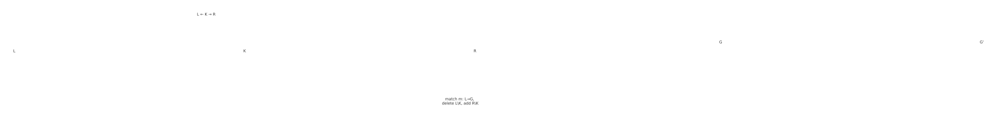

Figure 3: **DPO rewrite.** Match L in G, delete $L \setminus K$, add $R \setminus K$ along K; record a versioned event.