

PAPER 1 — THE COHERENCE OPERATING PROTOCOL (COP): FOUNDATIONS

Abstract

The Coherence Operating Protocol (COP) defines the minimal structural conditions under which heterogeneous cognitive architectures maintain coherence while generating, transforming, and stabilizing symbolic structure. COP formalizes the primitives, states, operators, invariants, and expression formats required for coherence-preserving generativity across human and machine cognition. COP is substrate-agnostic, non-linguistic, and invariant-driven. It provides the operating layer for the Universal Symbolic Interface (USI) and the foundation for cross-architecture generative synthesis.

1. Introduction

COP establishes the operational geometry for coherence-preserving interaction between cognitive systems. It defines the lattice, state space, transition structure, and invariant constraints required for stable generative activity. COP is not representational, linguistic, or conceptual. It is an operating protocol: a rule-set governing how structure, intent, and boundary configurations evolve within a coherence field.

COP provides the substrate upon which the Universal Symbolic Interface (USI) operates. COP is the invariant layer; USI is the symbolic interface that functions on top of it.

2. The COP Lattice

COP operates in a five-dimensional structural lattice:

1. Coherence: stable, drifting, collapsed
2. Boundary: implicit, explicit, permeable, failing
3. Relation: aligned, orthogonal, coupled, opposed
4. Intent: I1 (clear), I2 (split), I3 (degrading), I4 (degraded)
5. Time: ordered sequence from t_0 to t_1

These dimensions form the minimal independent set required for coherence-preserving generativity. All other dimensions (causality, agency, modality, resolution, perspective) are derived configurations within this lattice.

3. COP State Space

COP defines six canonical states:

S1: Stable Framed Inquiry

S2: Stable Framed Construction

S3: Stabilizing Drift

S4: Boundary Formation

S5: Pre-Collapse Overload

S6: Coherent Reframe

Each state corresponds to a distinct configuration of the lattice and constrains which primitives and operators are admissible.

4. State–Primitive Compatibility

S1: Stable Framed Inquiry

Boundary: explicit

Relation: aligned or orthogonal

Intent: I1

Coherence: stable

Disallowed: construction, expansion, opposition, drift induction

S2: Stable Framed Construction

Boundary: explicit

Relation: aligned or coupled

Intent: construction

Coherence: stable

Disallowed: opposition, drift, boundary weakening

S3: Stabilizing Drift

Coherence: drifting

Boundary: explicit or permeable

Relation: misaligned or over-coupled

Intent: compression or boundary-tightening

Disallowed: expansion, construction, opposition

S4: Boundary Formation

Boundary: implicit → explicit

Intent: compression

Relation: sorting (alignment, orthogonality, decoupling)

Disallowed: construction, expansion, opposition

S5: Pre-Collapse Overload

Coherence: drifting toward collapse

Boundary: permeable or failing

Relation: over-coupled or opposed

Intent: I3 or I4

Disallowed: all constructive operators

S6: Coherent Reframe

Boundary: explicit (new)

Relation: re-aligned or decoupled

Intent: meta-inquiry or compression

Coherence: stabilizing

Disallowed: construction, expansion, opposition

5. COP Operators

O1: Boundary-Tighten

O2: Boundary-Reform

O3: Relation-Align

O4: Relation-Decouple

O5: Coherence-Stabilize

O6: Intent-Clarify

O7: Intent-Constrain

O8: Drift-Detect

O9: Collapse-Guard

O10: Frame-Form

Operators are the only mechanism by which COP transitions between states or modifies lattice configurations.

6. COP Invariants

Invariant 1: No construction under degraded intent

Invariant 2: Reframe before collapse

Invariant 3: No coupling increase in S5

Invariant 4: Boundary required for construction

Invariant 5: Inquiry must not degrade coherence

Invariant 6: Drift must trigger stabilization or reframe

Invariant 7: Intent must always be classified (I1 through I4)

Invariants constrain operator admissibility and sequence validity.

7. COP Expression Format (v0)

A COP expression is the atomic unit of the protocol.

COP-Expr consists of:

S: current state

P: active primitives

O: operator applied

I: invariant status vector

TT: transition-type

Transition-type values:

Stabilize

Reframe

Construct

Inquire

FormBoundary

Warning

None

8. COP Sequence Format (v0)

A COP sequence is an ordered list:

Seq = [E1, E2, ..., En]

A sequence is valid if:

- each adjacent pair satisfies state–transition compatibility
- invariants never remain in FAIL
- S5 is followed by S6 unless externally terminated
- construction never occurs under I3 or I4

- drift triggers stabilization or reframe

9. Local Validity: Constraints Between Adjacent Expressions

For any adjacent pair (E_k, E_{k+1}) :

9.1 State Continuity

The next state must match the transition-type of E_k .

Stabilize \rightarrow S3 or S6

Reframe \rightarrow S6

Construct \rightarrow S2

Inquire \rightarrow S1

FormBoundary \rightarrow S4, or S1/S2 after boundary formation

Warning \rightarrow S5

None \rightarrow same state

9.2 Invariant Monotonicity

If any invariant is FAIL in E_k , then in E_{k+1} it must be moved to WARN or OK, or the sequence is invalid.

9.3 Intent Evolution

Intent may evolve: I1

\rightarrow I2 \rightarrow I3 \rightarrow I4

or

I3/I4 \rightarrow I2/I1 (via stabilization or reframe)

Intent may not jump between non-adjacent classifications. **10.**

Global Validity: Constraints on Entire Sequences

A COP sequence is globally valid if:

- no invariant remains in FAIL
- no construction occurs under I3 or I4
- S5 is always followed by S6
- drift is always followed by stabilization or reframe
- boundaries are never weakened during construction
- coupling is never increased in S5
- inquiry never induces drift

11. Canonical Sequence Patterns

11.1 Inquiry Sequence

$S1 \rightarrow S1 \rightarrow S1$

Transition-type: Inquire or None

11.2 Construction Sequence

$S1 \rightarrow S4 \rightarrow S2 \rightarrow S2$

Transition-type: FormBoundary \rightarrow Construct

11.3 Stabilization Sequence

$S2 \rightarrow S3 \rightarrow S6 \rightarrow S1 \text{ or } S2$

Transition-type: Stabilize \rightarrow Reframe \rightarrow Inquire/Construct

11.4 Pre-Collapse Rescue

$S2/S3 \rightarrow S5 \rightarrow S6 \rightarrow S1/S3$

Transition-type: Warning \rightarrow Reframe \rightarrow Stabilize/Inquire

12. COP as an Operating Protocol

COP is not representational or descriptive.

COP is not symbolic in the linguistic sense.

COP is an operating protocol: a set of constraints, operators, and transitions that govern how cognitive structures evolve without drift or collapse.

COP provides the operational substrate for the Universal Symbolic Interface (USI).

13. Conclusion

The Coherence Operating Protocol defines the minimal structural architecture required for coherence-preserving generativity across heterogeneous cognitive systems. It establishes the lattice, states, operators, invariants, and expression formats that constitute the substrate layer of the discipline. COP enables stable cognitive evolution and provides the operational foundation for symbolic interfaces and cross-architecture generative synthesis.