Concept Flyer — Shunyaya True Logic (STL)

When Truth Is Structurally Disciplined

Status: Public Open Standard (v2.0)
Release Date: February 16, 2026
Nature: Deterministic Structural Governance Layer
License: Open Standard — specification may be implemented freely; provided “as is” without warranty or liability.
Caution: Research, observability, and structural experimentation only. Not a predictive or safety-critical engine.

The Problem — Boolean Logic Assumes Instant Truth

Classical logic states:

```
Truth(P) ∈ {TRUE, FALSE}
```

This assumes truth is an instantaneous terminal state.

But in real systems — physical, digital, financial —
states evolve structurally before they stabilize.

Thresholds collapse instantly.
Structure does not.

Boolean logic is exact.
But it has no discipline before collapse.

The Shift — From Instant Truth to Structurally Valid Collapse

Shunyaya True Logic (STL) does not replace Boolean logic.

It introduces a structural topology prior to collapse:

```
T5 = { Z0, Eplus, S, Eminus, Zstar }
```

Where:

- `Z0` — declared structural zero
- `Eplus` — emerging toward TRUE
- `S` — stable TRUE regime
- `Eminus` — destabilizing transition
- `Zstar` — stable FALSE regime

Truth remains binary.
Collapse is governed.

The Core Invariant — Structure Never Alters Truth

Collapse mapping:

```
phi_T(S) = TRUE
phi_T(Zstar) = FALSE
phi_T(Z0) = UNDEFINED
phi_T(Eplus) = UNDEFINED
phi_T(Eminus) = UNDEFINED
```

If `state ∈ {S, Zstar}`:

`phi_T(state)` equals the original Boolean evaluation.

Boolean compatibility is preserved exactly.

When structurally stable, STL reduces to classical logic.

What STL Enforces — Deterministic Collapse Discipline

STL introduces:

- Stability window `W`
- Upper threshold `tau_s`
- Lower threshold `tau_l`
- Sensitivity bound `eps`
- Deterministic structural evolution

Transitional states cannot collapse.

No probability.
No fuzzy truth.
No learning.

No hidden state.
No tolerance thresholds.

Fully replay-verifiable.

What STL Prevents — Premature Truth Assignment

Naive Boolean thresholding collapses instantly at boundary crossing.

STL requires structural stability over window `W`.

Let:

```
E_total = total Boolean crossing events
E_premature = collapse events occurring before structural stabilization is
satisfied
```

Structural Admissibility Discipline:

```
SAD(P) = 1 - (E_premature / E_total)
```

SAD(P):

- Does not redefine truth
- Does not alter classification
- Does not introduce prediction

It measures timing alignment only.

STL does not guess truth.
It enforces structural validity before truth is declared.

Verified Deterministic Evidence

STL has been replay-verified across:

- Controlled phase-transition traces (ice-like)
- Threshold-edge stress tests
- Operator preservation validation
- De Morgan and involution stability
- Public financial dataset (SPX drawdown)
- Deterministic negative control traces

All runs satisfy:

```
B_A = B_B
```

Replay equivalence requires:

- Byte-identical CSV/TXT artifacts
- Identical T5 classifications
- Identical operator outputs
- Identical collapse mapping
- Identical MANIFEST.sha256

Replay equivalence is structural proof.

Practical Benefits — Why STL Matters

1. Deterministic Auditability
Every collapse decision is replay-verifiable.
No interpretive ambiguity.

2. Premature Collapse Prevention
Truth cannot be declared before structural stability.

3. Zero Modification to Classical Logic
Boolean algebra remains intact and untouched.

4. Finite Structural Vocabulary
`|T5| = 5`
State space is bounded and closed.

5. No Probabilistic Risk Layer
No randomness.
No confidence scores.
No heuristic arbitration.

6. Deployment Simplicity
Minimal parameters: `W`, `tau_s`, `tau_l`, `eps`.
No model training required.

7. Layered Governance Architecture
Admissibility can be separated cleanly from evaluation.

STL introduces discipline without introducing complexity.

What STL Is Not

STL does not:

- Replace Boolean logic
- Introduce probabilistic truth
- Redefine computation
- Act as a predictive inference engine
- Simulate physics
- Guarantee safety outcomes

Boolean logic remains terminal truth logic.
STL governs when terminal truth is structurally admissible.

Architectural Position

Layered stack:

```
SSRL → Participation (Admissibility)
STL → Collapse Discipline
Boolean → Terminal Truth
```

This separation ensures:

- No premature evaluation
- No collapse without admissibility
- No alteration of classical truth

Truth remains binary.
Structure becomes disciplined.

The Closing Principle

Boolean logic answers:
“Is it TRUE or FALSE?”

STL answers first:
“Is it structurally admissible to decide?”

Truth is not replaced. Truth is governed.

Shunyaya True Logic (STL) is a conservative, deterministic structural extension of Boolean algebra — introducing topology before collapse, while preserving truth exactly at stability.

OMP