Appendix A — Minimal Mathematical Formalism

SYF v2.0

A.1 Notation and Domains

Let:

- $t \in T$ denote discrete time steps (or observation indices)
- N be the space of admissible observations (noise-derived signals)
- $N_t \in N$ be the observation available at step t

Define a canonicalization operator:

$$C : N \to X$$

where X is a fixed canonical input space (e.g., fixed-length representations).

Define a deterministic invariant extractor:

$$T : X \to R$$

where $R \subset$ ■ (or ■, or fixed-point scalars) is the invariant codomain.

The Systemic Fire invariant is:

$$R_t = T(C(N_t))$$

A.2 Core Axioms (Formal)

Axiom 1 — Determinism

For any $N \in N$, the value $R = T(C(N))$ is unique.

$$\forall N \in N, \exists! R \in R \text{ such that } R = T(C(N))$$

No randomness, sampling, or time-dependent configuration is permitted in C or T.

Axiom 2 — Read-Only Observation (No Actuation)

SYF produces an output signal only. It does not produce actions. Formally, SYF defines no actuation function A and no policy π. The system is strictly observational:

$$SYF : N \to R$$

Axiom 3 — No Feedback Loop

The output R_t is not an input to subsequent canonicalization or extraction.

$$\forall t, (C, T) \text{ are independent of } \{R_0, \ldots, R_{t-1}\}$$

Equivalently, C and T are time-invariant mappings with no internal state.

Axiom 4 — Boundedness (Invariant Range)

There exist fixed bounds $R_{min} < R_{max}$ such that:

$$\forall t, R_t \in [R_{min}, R_{max}]$$

These bounds are construction constraints, not learned parameters.

Axiom 5 — Public Verifiability

Given N_t and the published definitions of C and T, any observer can recompute R_t.

$$Verify(N_t, R_t) \blacksquare (R_t = T(C(N_t)))$$

A.3 Boundary Conditions and Fail-Closed Rule

Define an admissibility predicate $Valid : N \to \{0, 1\}$ such that if $Valid(N_t) = 0$, the computation must not yield an extrapolated value.

Fail-closed rule:

$$Valid(N_t) = 0 \Rightarrow output = \perp$$

where $\perp$ denotes a defined "reject/undefined" output state.

$\perp$ is a non-numeric terminal symbol and MUST NOT be mapped to any numeric value.

No heuristics, interpolation, or fallback estimates are allowed.

A.4 FirePlank as a Safety Floor (Constraint Form)

Let S be the space of permissible system states for an external system that uses R_t. SYF itself does not control S; it only provides a measurable invariant.

Define a floor threshold $\theta \in (R_{min}, R_{max})$.

FirePlank constraint (informal): below a coherence threshold, downstream systems must not transition into states that allow unbounded behavior.

Minimal formal expression (as a guard condition):

$$R_t \leq \theta \Rightarrow Guard(S) \text{ enforces restricted transitions}$$

The FirePlank is therefore a non-inflationary floor constraint: it does not add capability, reward, or value; it only restricts reachable transitions under low-coherence conditions.

A.5 Phoenix as a Stabilization Condition (Non-Narrative)

Phoenix is a defined stabilization mode triggered only by boundary proximity.

Let $\varepsilon > 0$ be a small margin.

Trigger condition:

$$R_t \in [R_{min}, R_{min} + \varepsilon] \cup [R_{max} - \varepsilon, R_{max}] \Rightarrow Stabilize()$$

Constraints on Stabilize:

- It must not restore historical state
- It must not introduce reward
- It must not increase long-term capability
- It must not create a feedback dependency for C or T

Phoenix is therefore a thermodynamic correction, not recovery.

Phoenix provides no liveness or progress guarantee. It only prevents invariant discontinuity.

A.6 Minimal Failure Modes (Formal)

Invalid input:

$$Valid(N_t) = 0 \Rightarrow \perp$$

Boundary saturation:

$$R_t \to R_{min} \text{ or } R_{max} \Rightarrow Stabilize()$$

Non-computability: If C or T cannot be computed for any reason, output must be $\perp$, not an estimate.

A.7 Non-Goals (Formal Exclusions)

SYF does not define:

- any objective function J
- any policy π
- any reward r
- any learning rule Δ
- any governance function g
- any stateful memory $M_{t+1} = f(M_t, \cdot)$

Formally, the law is complete as:

$$SYF = (C, T, [R_{min}, R_{max}], Valid, \perp)$$

A.8 Interpretation

SYF treats safety as a property of reachable state space under invariant bounds. It does not attempt to make systems benevolent; it makes unsafe escalation structurally unreachable for any system built to respect the bounds.