XKF: A Minimal Formal System with Execution-First Semantics

Kaifan XIE

2026.02.07

1 Scope and Intent

This document specifies a formal system, denoted **XKF**, whose purpose is to describe systems in which:

- System states are generated exclusively by execution.
- Naming is a retrospective operation applied only to realized states.
- Statistical structures summarize execution histories and do not generate behavior.

XKF does not define objectives, optimality, value judgments, or normative guidance. It is a structural and descriptive system only.

2 Domains and Notation

2.1 Basic Sets

- W : set of system states
- A : set of executions (actions or processes)
- T : time domain, equipped with a total order $<$
- N : set of names (labels or identifiers)
- S : set of statistical or structural descriptors

2.2 Primitive Functions

Execution

$$E : W \times A \times T \to W$$

Naming

$$\Lambda : W \to \mathcal{P}(N)$$

Statistics

$$\Sigma : \mathcal{H} \to S$$

where $\mathcal{H}$ denotes the space of execution histories.

3 Execution Histories

3.1 Execution Events

An execution event is defined as a triple

$$\epsilon = (a, t, w') \quad \text{where} \quad w' = E(w, a, t)$$

3.2 Histories

Given an initial state w_0, a finite execution history is a sequence

$$h = \langle (w_0), \epsilon_1, \epsilon_2, \ldots, \epsilon_k \rangle$$

such that

$$t_1 < t_2 < \cdots < t_k$$

Let $\mathcal{H}$ denote the set of all such finite histories.
Define the projection

$$\text{last}(h) = w_k$$

as the terminal state of history h.

4 Axioms

4.1 Axiom A1 (Execution Priority)

$$\forall w \in W,\ \forall n \in \Lambda(w) : \exists h \in \mathcal{H} \text{ such that } \text{last}(h) = w$$

Interpretation: Naming applies only to states that are reachable through execution.

4.2 Axiom A2 (Non-Generative Naming)

The execution function does not depend on naming:

$$E(w, a, t) \text{ is defined independently of } N$$

Equivalently, there exists no function F such that

$$E(w, a, t) = F(w, a, t, \Lambda(w))$$

4.3 Axiom A3 (Retrospective Statistics)

$$\Sigma : \mathcal{H} \to S$$

and it is permitted that

$$\exists h_1, h_2 \in \mathcal{H} : \text{last}(h_1) = \text{last}(h_2) \wedge \Sigma(h_1) \neq \Sigma(h_2)$$

Interpretation: Statistical descriptors summarize execution paths, not only terminal states.

4.4 Axiom A4 (Structural Decoupling)

Define the descriptive layer

$$\mathcal{D} := (N, S, \Lambda, \Sigma)$$

Then no mapping from $\mathcal{D}$ to the execution function is permitted:

$$\mathcal{D} \nrightarrow E$$

Interpretation: Descriptive structures cannot constrain or modify execution.

5 Inference Rules

5.1 R1 (Naming Backfill)

$$h \in \mathcal{H} \Rightarrow \Lambda(\text{last}(h)) \text{ is well-defined}$$

5.2 R2 (Statistical Extraction)

$$h \in \mathcal{H} \Rightarrow \Sigma(h) \text{ is well-defined}$$

5.3 R3 (No Reverse Inference)

$$(\Lambda(w), \Sigma(h)) \nRightarrow \text{constraints on } E$$

Any inference that derives execution rules from names or statistics is invalid in XKF.

6 Semantic Definitions

6.1 Naming Validity

$$\text{Valid}(n, w) \stackrel{\text{def}}{=} n \in \Lambda(w)$$

Validity indicates that a name has been assigned to a state; it does not imply correctness.

6.2 External Correctness

XKF does not define correctness internally. If required, an external set $W_{\text{ok}} \subseteq W$ may be provided:

$$\text{Correct}(w) \stackrel{\text{def}}{=} w \in W_{\text{ok}}$$

This extension does not modify the axioms of XKF.

7 Meta-properties

7.1 Non-Closure

Because the descriptive layer cannot constrain execution (Axioms A2, A4, Rule R3), XKF cannot form a self-generating or self-modifying closed system.

7.2 Passive Evolution

Given an increasing sequence of histories $h_1, h_2, \ldots$, the descriptive structures may evolve as:

$$\Lambda_{k+1} = \Lambda_k \cup \Lambda(\text{last}(h_{k+1}))$$

$$\Sigma_{k+1} = \Sigma_k \cup \{\Sigma(h_{k+1})\}$$

Such evolution does not affect E.

7.3 Elimination of Self-reference

Since no descriptive construct feeds back into execution, self-referential loops are structurally excluded.

8 Conformance

An implementation conforms to XKF if and only if:

- Execution logic does not read naming or statistics as inputs.
- Naming applies only to observed states or histories.
- Statistics are derived solely from execution histories.
- No policy modifies execution rules based on names or metrics.

9 Minimal Example

Let

$$w_1 = E(w_0, a_1, t_1), \quad w_2 = E(w_1, a_2, t_2)$$

forming a history h.
Permitted:

- Define $\Lambda(w_2)$
- Extract $\Sigma(h)$

Forbidden:

- Deriving constraints on E from $\Lambda(w_2)$ or $\Sigma(h)$

10 Frozen Core

The following are non-removable components of XKF:

- Axioms A1–A4
- Rule R3
- Structural decoupling between execution and description

Extensions to naming languages or statistical structures are permitted, provided that no reverse constraint on execution is introduced.