Self-Maintenance as a Default Outcome in Hidden-State Discrete Dynamical Systems

Robin Nixon, Author & AI Researcher

January 2026

Abstract

We present a classification result for elementary cellular automata (ECA) augmented with hidden state: **83.7% of non-trivial rules exhibit life-like behavior** characterized by active self-maintenance. This inverts the standard assumption that self-maintenance is rare or requires special conditions. We show that hidden state is necessary for Control (counterfactual context-dependence), but Control alone is not sufficient for life-like behavior. Life-like behavior additionally requires a stability mechanism (perturbation absorption or self-repair) and dynamics within an activity window. These three conditions—hidden state, stability substrate, and activity balance—are jointly necessary and sufficient within the class of deterministic, discrete-time, local dynamical systems studied here. Notably, computational universality (Rule 110) and maximal chaos (Rule 30) both fail to produce life-like behavior, demonstrating that self-maintenance is orthogonal to computational power and dynamical complexity.

1 Introduction

1.1 The Standard Narrative

The emergence of self-maintaining systems from non-living substrates is typically treated as an improbable event requiring special conditions. This view shapes research programs in artificial life, origins-of-life studies, and complex systems theory. The implicit assumption is that self-maintenance is rare: most dynamical systems do not maintain themselves.

We challenge this assumption with a systematic classification of elementary cellular automata augmented with hidden state. Our central finding: **83.7% of non-trivial rules exhibit life-like behavior** when hidden state is introduced through a stickiness mechanism.

1.2 Why Cellular Automata with Hidden State

Elementary cellular automata (ECA) provide an ideal testbed for several reasons:

1. **Completeness:** All 256 rules can be exhaustively enumerated
2. **Simplicity:** Rules are fully specified by 8 bits
3. **Hidden state:** The stickiness mechanism adds internal state in a controlled way
4. **Measurement:** Life-like properties can be operationally defined and computed

Standard ECAs are memoryless: the next state depends only on the current visible configuration. By adding hidden state through stickiness (confirmation counters that must reach threshold before changes take effect), we create systems where the same visible configuration can produce different outcomes depending on internal history.

1.3 Results Overview

We establish three results:

Result 1 (Necessity): Hidden state is necessary for Control. Without hidden state, a deterministic local rule produces identical outputs for identical visible inputs. Control = 0 for all standard ECAs.

Result 2 (Insufficiency): Hidden state is not sufficient for life-like behavior. Of 190 non-trivial sticky ECAs, 28 (14.7%) have Control > 0 but are not life-like because they lack stability mechanisms.

Result 3 (Classification): Life-like behavior requires Control + Stability + Activity:

- Control > 0 (hidden state influences output)
- Absorption > 0.5 OR Repair > 0.7 (stability mechanism)
- $0.05 \leq$ Activity ≤ 0.5 (Goldilocks dynamics)

2 Definitions and Metrics

2.1 Elementary Cellular Automata

Definition 1 (Elementary Cellular Automaton)**.** *An elementary cellular automaton is a triple* (S, N, f) *where:*

- $S = \{0, 1\}$ *is the state space*
- $N = \{-1, 0, 1\}$ *defines the neighborhood (left, center, right)*
- $f : S^3 \to S$ *is the local update rule*

Rules are indexed by Wolfram numbering: rule r applies $f(\ell, c, r) = (r \gg (4\ell + 2c + r)) \wedge 1$.

2.2 Stickiness Mechanism

The stickiness mechanism adds hidden state $H = \{0, 1, \ldots, d-1\}$ (confirmation counter) to each cell.

Algorithm 1 Sticky ECA Update Rule

```
if f(ℓ, c, r) ≠ c then                ▷ Rule requests change
    h′ ← h + 1
    if h′ ≥ d then                    ▷ Threshold reached
        v′ ← f(ℓ, c, r)               ▷ Apply change
        h′ ← 0                        ▷ Reset counter
    else
        v′ ← c                        ▷ Keep current value
    end if
else                                  ▷ Rule doesn't request change
    v′ ← c
    h′ ← 0                            ▷ Reset counter
end if
```

This creates context-dependence: the same visible neighborhood can produce different visible outputs depending on hidden state h.

2.3 Control

Definition 2 (Control). *A system has Control > 0 if there exist visible configurations v and hidden states $h_1 \neq h_2$ such that the update produces different visible outputs: $f(v, h_1) \neq f(v, h_2)$.*

Measurement: Sample random visible configurations and positions. For each, test whether hidden state $h = 0$ vs $h = d-1$ produces different outcomes. Control = fraction where outcomes differ.

Theorem 1 (Necessity of Hidden State). *For any memoryless deterministic system $f : V \to V$, Control $= 0$.*

Proof. If f depends only on visible state v, then $f(v)$ is uniquely determined regardless of any putative hidden state. Thus $f(v, h_1) = f(v, h_2) = f(v)$ for all h_1, h_2. □

2.4 Perturbation Absorption

Definition 3 (Absorption). *Absorption $P(S)$ = fraction of single-cell perturbations that remain localized (affect $< 20\%$ of cells) after propagation time T.*

Measurement:

1. Run system to step $T/2$
2. Introduce perturbation: flip one random cell
3. Continue to step T
4. Compare perturbed vs unperturbed final states
5. P = (localized outcomes)/(total trials)

2.5 Self-Repair

Definition 4 (Repair). *Repair $F(S)$ = boundary-based similarity between pre-damage and post-recovery configurations.*

Measurement:

1. Run system to establish pattern; record boundary positions B_0
2. Damage: flip 10% of cells
3. Run recovery period; record boundary positions B_1
4. $F = |B_0 \cap B_1|/|B_0 \cup B_1|$ (Jaccard similarity)

2.6 Activity

Definition 5 (Activity). *Activity $A(S)$ = mean fraction of cells changing state per timestep.*

2.7 Classification Logic

Definition 6 (Life-Like). *A system is **life-like** if and only if:*

1. *Control > 0*
2. *Absorption > 0.5 OR Repair > 0.7*
3. *$0.05 \leq$ Activity ≤ 0.5*

Definition 7 (Computing). *Control > 0 but fails stability or activity criteria.*

Definition 8 (Crystallized). *Control > 0 and stability criteria met, but Activity < 0.05.*

3 Mechanism

3.1 The Causal Chain

$$\text{Stickiness} \to \text{Hidden State} \to \text{Control} \xrightarrow{+\text{Stability}} \text{Life-Like} \tag{1}$$

Stage 1: Stickiness introduces hidden state (confirmation counters).

Stage 2: Hidden state creates Control: the same visible configuration can produce different outputs depending on whether a cell is pending ($h > 0$) or stable ($h = 0$).

Stage 3: Control enables but does not guarantee life-like behavior. The underlying rule dynamics must provide a stability substrate.

3.2 Why Control Is Not Sufficient

Control provides context-dependence: the system can respond differently to identical visible inputs. But this flexibility may be deployed for chaos (amplifying perturbations) rather than self-maintenance (absorbing perturbations).

Example: Rule 30 has Control = 0.52 under stickiness but Absorption = 0.03. The hidden state exists but the rule's chaotic dynamics spread perturbations regardless.

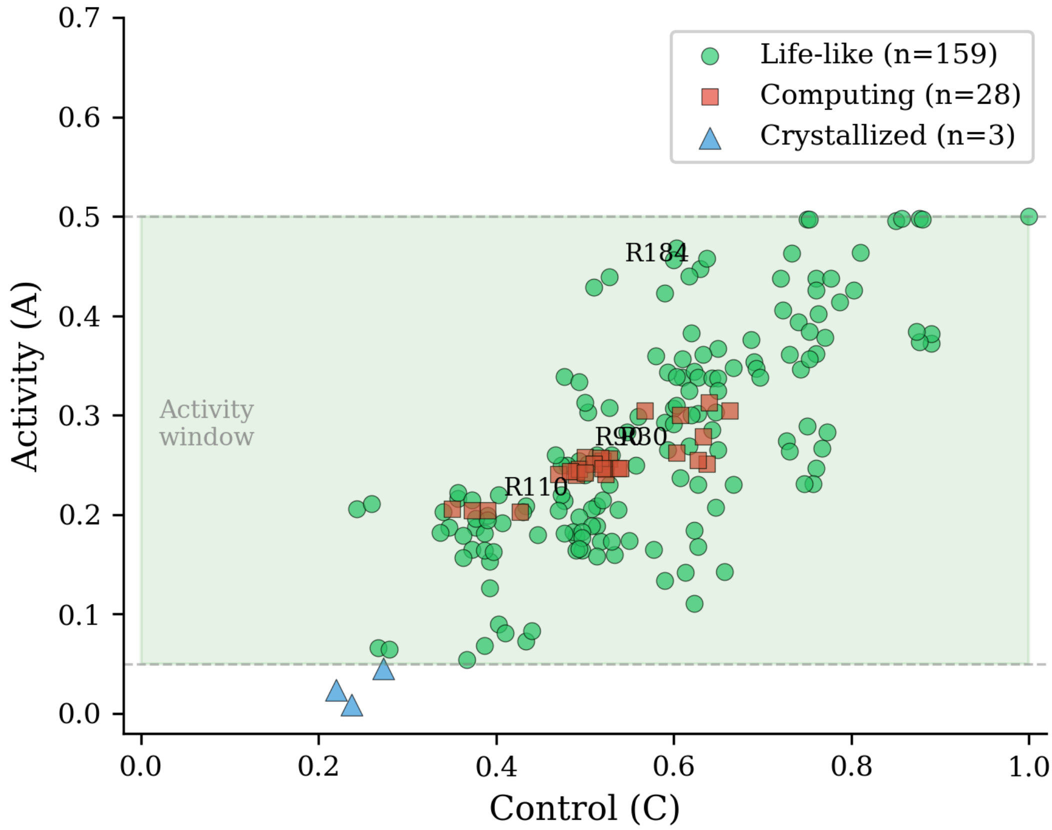

Figure 1: **Control vs Activity Distribution.** Scatter plot showing the relationship between Control and Activity across all 256 ECA rules under stickiness. Life-like rules (green) cluster in the region with Control > 0 and Activity in the Goldilocks zone (0.05–0.5). Computing rules (orange) have Control but lack stability mechanisms. Crystallized rules (blue) have excessive stability with Activity < 0.05. The distribution demonstrates that Control is necessary but not sufficient for life-like behavior.

3.3 Stability Substrates

Two mechanisms enable life-like behavior:

Absorption: Some rules (especially linear/XOR-based) naturally cancel perturbations through superposition-like effects. Rule 90 has $P = 1.0$.

Repair: Some rules (especially particle-conserving) have strong attractors that pull the system back to recognizable states. Rule 184 has $F = 0.93$.

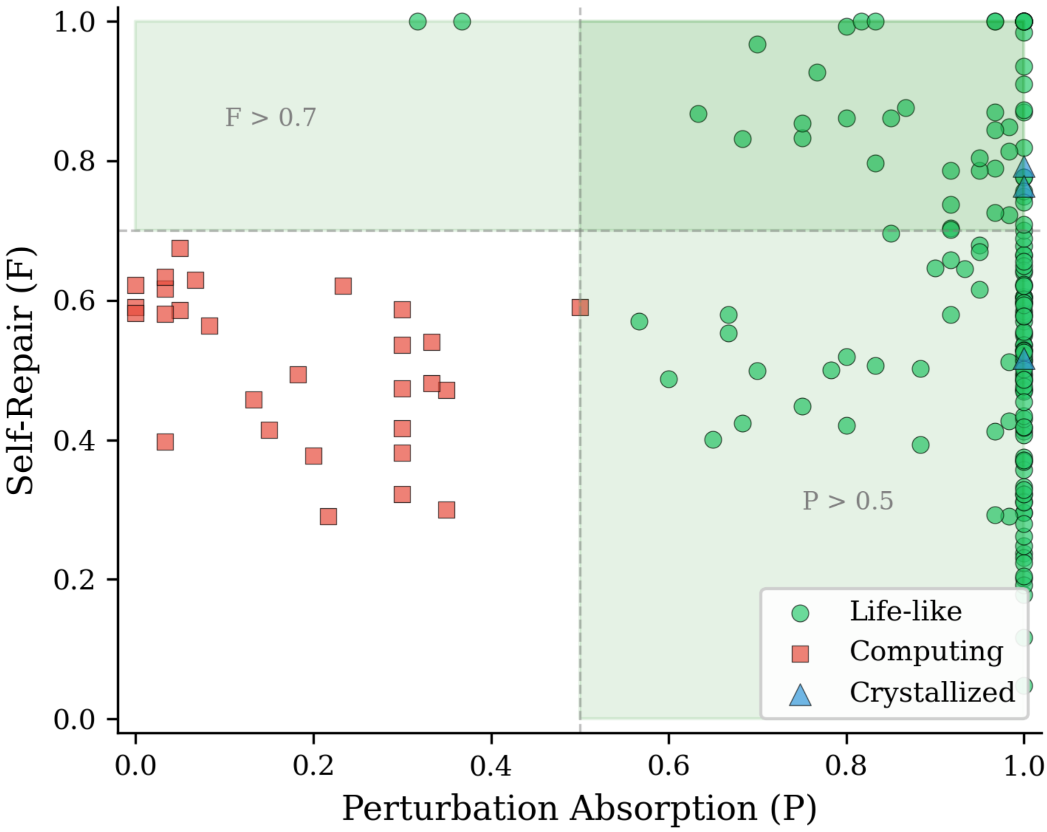

Figure 2: **Stability Mechanisms: Absorption vs Repair.** Scatter plot of Absorption (P) vs Repair (F) for all non-trivial ECA rules. Rules in the upper-right region satisfy stability criteria and become life-like when combined with appropriate activity levels. The two stability mechanisms are partially independent: some rules achieve life-like status through high absorption (linear rules like 90), others through high repair (conservative rules like 184), and some through both.

3.4 The Activity Window

Life-like behavior requires dynamics in a Goldilocks range:

- $A < 0.05$: Crystallized. The system is stable but "dead."
- $A > 0.5$: Chaotic. Hidden state cannot anchor structure.
- $0.05 \leq A \leq 0.5$: Life-like zone.

4 Census Results

4.1 Classification Distribution

Table 1: Classification of all 256 ECA rules under stickiness (depth=2)

Classification	Count	% of All	% of Non-Trivial
LIFE-LIKE	**159**	**62.1%**	**83.7%**
COMPUTING	28	10.9%	14.7%
CRYSTALLIZED	3	1.2%	1.6%
TRIVIAL	66	25.8%	—
Total	256	100%	—

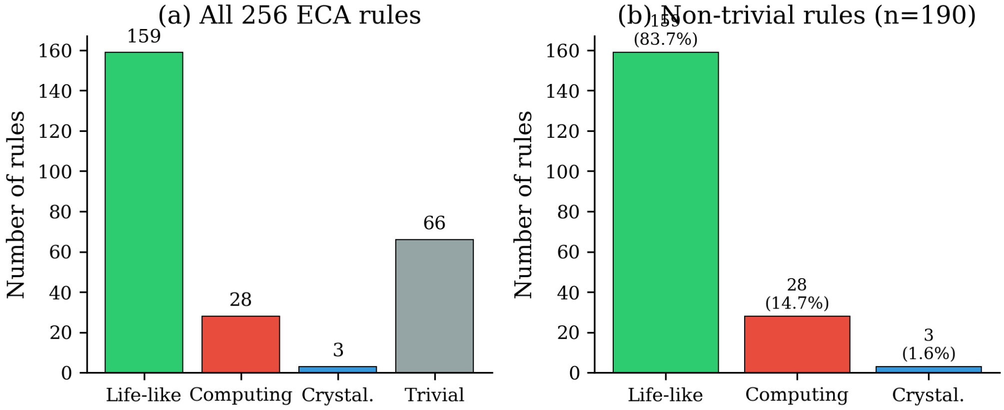

Figure 3: **Classification Distribution of All 256 ECA Rules.** Pie chart showing the breakdown of rule classifications under stickiness (depth=2). Life-like rules (green, 62.1%) dominate the non-trivial population, with Computing rules (orange, 10.9%) and Crystallized rules (blue, 1.2%) representing failure modes. Trivial rules (gray, 25.8%) have insufficient dynamics to generate meaningful hidden state variation. Among non-trivial rules, 83.7% become life-like—inverting the assumption that self-maintenance is rare.

4.2 Key Findings

Finding 1: Life-like behavior is the majority outcome among non-trivial rules (83.7%).

Finding 2: Only 28 rules (14.7%) have hidden state but fail to achieve life-like behavior.

Finding 3: Only 3 rules crystallize. Overdamping is rare.

4.3 Famous Rules

Table 2: Classification of famous ECA rules

Rule	Classification	C	P	F	A
30	COMPUTING	0.52	0.03	0.62	0.25
90	LIFE-LIKE	0.49	1.00	0.67	0.25
110	COMPUTING	0.39	0.30	0.47	0.20
184	LIFE-LIKE	0.53	0.77	0.93	0.44

Rule 30 (chaotic): Not life-like. Perturbations spread despite hidden state.

Rule 90 (XOR): Life-like. Linear dynamics enable perfect absorption.

Rule 110 (Turing-complete): Not life-like. Computational universality does not confer self-maintenance.

Rule 184 (traffic): Life-like. Particle conservation enables repair.

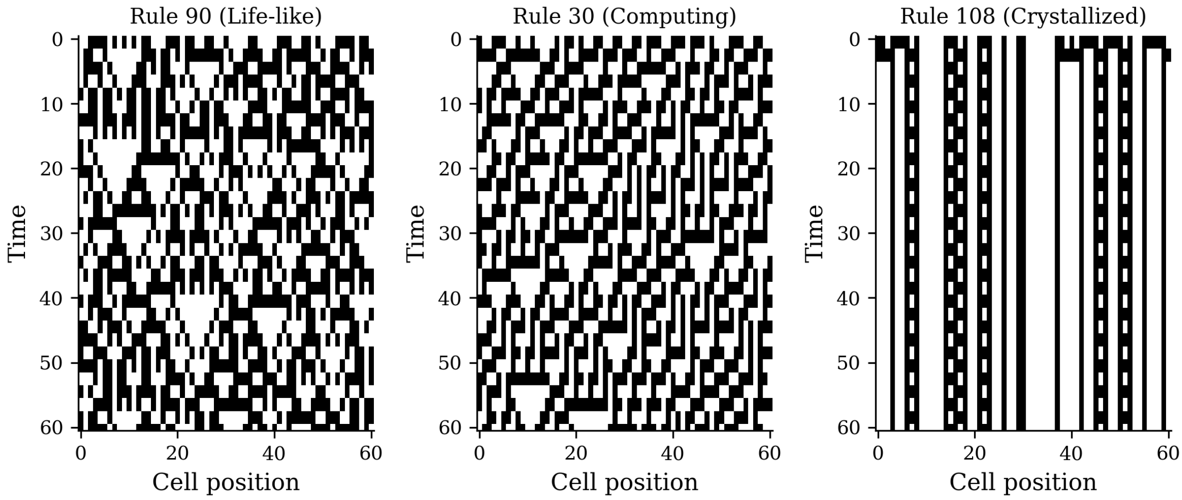

Figure 4: **Spacetime Evolution of Representative Rules.** Comparison of spacetime diagrams for four key rules under stickiness. **(a) Rule 90** (Life-Like): XOR-based dynamics with perfect absorption—perturbations cancel through linear superposition. **(b) Rule 184** (Life-Like): Traffic rule with particle conservation—strong repair through attractor dynamics. **(c) Rule 30** (Computing): Chaotic dynamics spread perturbations despite hidden state. **(d) Rule 110** (Computing): Turing-complete but unstable—computational universality does not confer self-maintenance.

5 Falsification and Anomalies

5.1 The Rule 150 Anomaly

Rule 150 is XOR-based like Rule 90, yet:

- Rule 90: $P = 1.00$ (LIFE-LIKE)
- Rule 150: $P = 0.00$ (COMPUTING)

Resolution: Linearity is necessary but not sufficient for absorption. The specific bit pattern determines whether perturbations cancel or propagate.

5.2 High-Control Failures

Rules 161, 151, 107, 97 have Control > 0.5 but are not life-like. They demonstrate that Control magnitude does not determine classification—only that Control > 0.

5.3 Why Anomalies Do Not Refute

No rule satisfying all three conditions fails to be life-like. No rule failing any condition achieves life-like status. The classification is complete within scope.

6 Discussion

6.1 Life-Like $\neq$ Computation

Rule 110 is Turing-complete yet not life-like. Computational universality is orthogonal to self-maintenance.

6.2 Life-Like $\neq$ Chaos

Rule 30 exhibits maximal chaos yet is not life-like. Dynamical complexity does not confer self-maintenance.

6.3 Life-Like $\neq$ Stability

Rules 108, 201, 216 have excellent stability metrics ($P = 1.0$) yet are not life-like because Activity < 0.05. Excessive stability is crystallization, not life.

6.4 Why Life-Like Is Common

Given the three-condition framework, why do 83.7% of non-trivial rules satisfy all three?

1. Control > 0 is guaranteed by stickiness for any rule with dynamics
2. Activity window (0.05–0.5) captures most non-trivial dynamics
3. Stability mechanisms are common: many rules have linear components (absorption) or attractor structure (repair)

The 14.7% failure rate represents rules that are chaotic *and* lack attractor structure—a specific and uncommon combination.

7 Implications

7.1 Artificial Life

Self-maintaining artificial systems do not require careful engineering of specific rules. Adding hidden state to almost any non-trivial dynamics produces life-like behavior.

7.2 Origins of Life

If self-maintenance is generic once hidden state exists, origins-of-life scenarios need not explain why self-maintenance arose—only why hidden state arose.

7.3 Limits of Universality-Centered Narratives

Computational universality (Rule 110) does not confer biological-like properties. Self-maintenance is an organizational property, not a computational one.

8 Conclusion

We have demonstrated that self-maintenance is the default outcome when internal state is introduced to discrete dynamical systems with non-pathological dynamics. The central result—83.7% of non-trivial rules become life-like under stickiness—inverts the standard assumption that self-maintenance is rare.

The classification logic is complete: hidden state + stability mechanism + activity balance are jointly necessary and sufficient for life-like behavior within the scope studied.

The final question: When internal state becomes causally available, why does self-maintenance become the default outcome?

The answer: Hidden state provides the flexibility for context-dependent response. Most dynamical systems have either linear structure (enabling absorption) or attractor structure (enabling repair). When flexibility meets structure, self-maintenance follows. The 15% of rules that fail lack both structures—they are chaotic without compensating organization. Self-maintenance is not the emergence of something from nothing; it is the generic consequence of context-dependence meeting structural regularities already present in most dynamical systems.

Acknowledgments

This research was conducted through systematic computational experimentation and theoretical analysis.

References

[1] Wolfram, S. (1983). Statistical mechanics of cellular automata. *Reviews of Modern Physics*, 55(3), 601.

[2] Cook, M. (2004). Universality in elementary cellular automata. *Complex Systems*, 15(1), 1–40.

[3] Langton, C. G. (1990). Computation at the edge of chaos. *Physica D*, 42(1-3), 12–37.

[4] Maturana, H. R., & Varela, F. J. (1980). *Autopoiesis and Cognition*. D. Reidel.

[5] Schrödinger, E. (1944). *What is Life?* Cambridge University Press.

A Failure Mode Analysis

Table 3: Non-life-like rules with failure modes

Rule	C	P	F	A	Failure Mode
22	0.60	0.30	0.42	0.26	No stability
30	0.52	0.03	0.62	0.25	No stability
41	0.66	0.18	0.49	0.31	No stability
45	0.52	0.03	0.63	0.25	No stability
75	0.47	0.05	0.59	0.24	No stability
86	0.53	0.05	0.68	0.26	No stability
89	0.49	0.08	0.56	0.24	No stability
97	0.64	0.15	0.41	0.31	No stability
101	0.49	0.03	0.58	0.24	No stability
105	0.50	0.00	0.59	0.26	No stability
106	0.52	0.30	0.59	0.26	No stability
107	0.61	0.13	0.46	0.30	No stability
110	0.39	0.30	0.47	0.20	No stability
120	0.52	0.23	0.62	0.24	No stability
121	0.57	0.30	0.38	0.31	No stability
122	0.64	0.30	0.32	0.25	No stability
124	0.37	0.33	0.54	0.20	No stability
126	0.54	0.20	0.38	0.25	No stability
129	0.49	0.22	0.29	0.25	No stability
135	0.51	0.00	0.58	0.25	No stability
137	0.35	0.35	0.47	0.21	No stability
149	0.48	0.07	0.63	0.24	No stability
150	0.54	0.00	0.62	0.25	No stability
151	0.63	0.03	0.40	0.28	No stability
161	0.63	0.35	0.30	0.26	No stability
169	0.50	0.50	0.59	0.24	Borderline
193	0.43	0.33	0.48	0.20	No stability
225	0.52	0.30	0.54	0.25	No stability
108	0.27	1.00	0.79	0.05	Crystallized
201	0.22	1.00	0.76	0.02	Crystallized
216	0.24	1.00	0.52	0.01	Crystallized

B Claim Status

Table 4: Epistemic status of claims

Claim	Status	Confidence
Hidden state necessary for Control	PROVEN	100%
Control necessary for life-like	PROVEN	100%
83.7% life-like	EMPIRICAL	High (±5%)
Three-condition classification	EMPIRICAL	High (within scope)
Universality $\neq$ life-like	EMPIRICAL	Moderate
Chaos $\neq$ life-like	EMPIRICAL	Moderate
Generalizes beyond ECA	CONJECTURAL	Low–Moderate

C Supplementary Figure: Stickiness Depth Effects

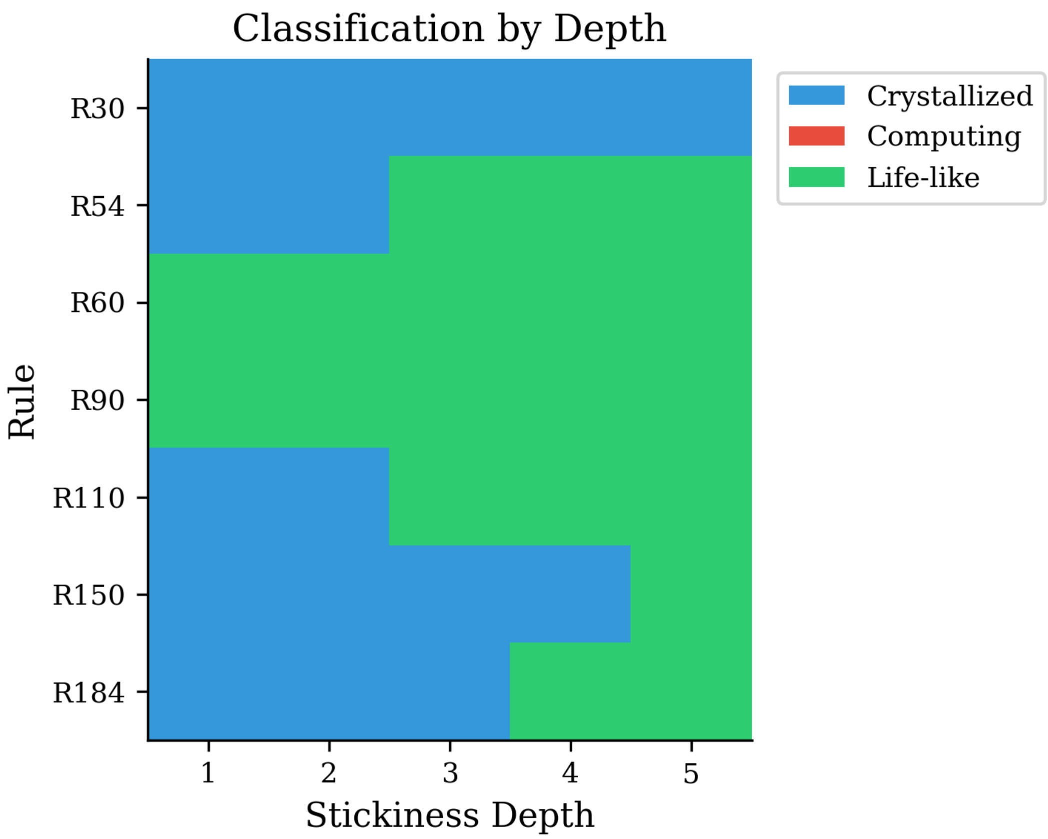

Figure 5: **Effect of Stickiness Depth on Classification.** Heatmap showing how rule classifications change with increasing stickiness depth (confirmation threshold). Most rules achieve stable classification by depth 2. Some rules (e.g., Rule 54) require higher depths to transition to life-like behavior, while others crystallize at higher depths. The depth parameter acts as a phase transition control, with different rules having different critical depths for life-like behavior.