Which Regular Languages Admit Provably Correct RNN Implementations? A Lattice-Theoretic Characterisation via Forward-Invariant Sets

Zacharie Bugaud [1]

Abstract

We study which regular languages can be recognised by recurrent neural networks with a forward-invariant certificate of length generalisation. An Elman RNN defines a switched dynamical system $h_{t+1} = \tanh(W_{hh}h_t + u_{x_t})$; we prove that length generalisation under such a certificate is equivalent to the existence of a forward-invariant positive-margin set (Theorem 2.1, if-and-only-if). We then establish a precise connection to automata theory: the class of regular languages admitting forward-invariant RNN implementations is contained within the class of *absorbing-state* languages, those whose minimal DFA has an accept state q_a with $\delta(q_a, \sigma) = q_a$ for all $\sigma \in \Sigma$. This class is closed under union and intersection (but not complement in general). Non-absorbing languages (modular counting, last-k, alternating patterns) provably *cannot* admit forward-invariant sets of this form, and empirically show near-zero (0–7%) success with our training protocol (vs. 63–100% baseline). We verify the characterisation computationally: Kleene iteration on the complete lattice of axis-aligned boxes converges in $\sim$14 iterations ($\rho_{\mathcal{T}} \approx 0.12$), and CQLF certificates via LMI verify 109/110 models up to H=70. Across 42 absorbing patterns and 12 non-absorbing patterns: 350/350 vs $\leq$7%, a near-perfect separation predicted by the lattice-theoretic characterisation. The gap formula $\text{gap}_d \approx \text{sech}^2(\bar{z}_d) \cdot |\Delta z_d|$ connects the automaton structure to an analytic sufficient condition ($r = +0.743$, F1 $= 0.872$). The result is a one-directional characterisation of the expressibility-verifiability boundary for recurrent networks on formal languages: necessity is proven, while the converse is supported empirically across the 42 absorbing patterns we tested.

1. Introduction

A classical question at the intersection of formal language theory and neural networks is: which languages can a given architecture recognise (Siegelmann and Sontag, 1995; Weiss et al., 2018)? Weiss et al. (2018) proved that finite-precision RNNs with squashing activations recognise exactly the regular languages. But *recognising* a language (correctly classifying strings up to some length) is weaker than *provably implementing* it (correctly classifying strings at all lengths). We study the stronger question:

> **Question.** For which regular languages L does there exist an Elman RNN that correctly classifies all strings in L at *arbitrary* length, with a *forward-invariant* certificate verifiable from weights alone?

We state the question this way deliberately. The certificate target is the existence of a forward-invariant positive-margin set in hidden-state space; languages outside the resulting class may still admit other forms of length-generalising RNN implementations not captured by this certificate.

This is a mathematical question about the intersection of dynamical systems, lattice theory, and formal languages. An Elman RNN defines two maps $f_0, f_1 : \mathbb{R}^H \to \mathbb{R}^H$ forming a switched dynamical system. Provable length generalisation *under a forward-invariant certificate* requires a forward-invariant set: a region of state space that the orbit never leaves under any switching sequence. We show that the existence of such a set is determined by a property of the *minimal DFA* for L.

Main result (informal). A regular language admits a provably length-generalising RNN implementation only if its minimal DFA has an absorbing accept state, a state q_a satisfying $\delta(q_a, \sigma) = q_a$ for all input symbols σ. The converse holds empirically: across 42 absorbing-state patterns, the recipe achieves 350/350 success (349/350 certified). This

[1]Astera Institute, Berkeley, CA, USA. Correspondence to: Zacharie Bugaud <zacharie@astera.org>.

Accepted to the 3rd AI for Math Workshop: Toward Self-Evolving Scientific Agents, AI4Math@ICML 2026, Seoul, South Korea. Copyright 2026 by the author(s).

class of languages is closed under $\cup$ and $\cap$ (but not complement in general: the complement of an absorbing-state language need not be absorbing-state).

Verification tools. We verify the characterisation using three mathematical tools: (1) Kleene iteration on the complete lattice of axis-aligned boxes (Tarski's theorem), (2) Common Quadratic Lyapunov Functions via LMI feasibility, and (3) a closed-form gap formula from the mean value theorem. Together, these certify 349/350 trained models (99.7%) with 0 false positives across 42 absorbing patterns and 0% across 12 non-absorbing patterns.

2. Problem Formulation

System. $h_{t+1} = \tanh(W_{hh}h_t + u_{x_t}) \equiv f_{x_t}(h_t)$, $x_t \in \{0,1\}$, classifier $y = w^\top h + b_{\text{fc}}$.

Theorem 2.1 (Reachable-set separation)**.** *$C_k \subset \mathbb{R}^H$ certifies length generalisation iff $f_0(C_k) \cup f_1(C_k) \subseteq C_k$ and $\min_{h \in C_k} w^\top h + b_{fc} > 0$.*

Remark 2.2. The converse holds: if the network generalises, the reachable set $R = \bigcup_{t > t_{\text{detect}}} \{h_t\}$ (over all switching sequences) satisfies both conditions. Thus Theorem 2.1 is an if-and-only-if characterisation.

The verification task. Given trained weights $(W_{hh}, W_{ih}, b, w, b_{\text{fc}})$, determine whether a forward-invariant positive-margin set exists and construct it explicitly.

3. The Lattice-Theoretic Characterisation

3.1. Absorbing vs Non-Absorbing Languages

Theorem 3.1 (Absorbing characterisation)**.** *Let L be a regular language with minimal DFA $(Q, \Sigma, \delta, q_0, F)$. A forward-invariant RNN implementation of L exists only if the DFA has an accept state $q_a \in F$ satisfying $\delta(q_a, \sigma) = q_a$ for all $\sigma \in \Sigma$ (absorbing).*

Proof sketch. If no absorbing accept state exists, then after accepting a string, the DFA must continue transitioning between states depending on future input. The RNN must track this state dependence at all future timesteps, which requires *input-dependent* hidden state evolution after acceptance. But a forward-invariant set C_k with $f_0(C_k) \cup f_1(C_k) \subseteq C_k$ constrains the orbit to a fixed region regardless of input. For a non-absorbing DFA, the correct output changes with future input (e.g., modular counting: accepting after "11" but rejecting after "111"), requiring the orbit to leave any fixed region. □

The converse is empirically constructive (not proven). For every absorbing-state language we tested, we can train an Elman RNN that provably length-generalises (350/350 across 42 patterns, 349/350 certified). We do not prove that every absorbing-state regular language is forward-invariant-implementable; we report this only as an empirical converse across the 42 patterns in our test suite.

3.2. Lattice Structure

The absorbing-state languages form a sublattice of the Boolean lattice of regular languages:

Proposition 3.2 (Closure properties)**.** *If L_1 and L_2 are absorbing-state regular languages, then so are $L_1 \cup L_2$ and $L_1 \cap L_2$.*

Proof. For union: the product DFA $(Q_1 \times Q_2, \Sigma, \delta_\times, (q_0^1, q_0^2), F_\cup)$ has accept states $F_\cup = (F_1 \times Q_2) \cup (Q_1 \times F_2)$. If both L_1 and L_2 have absorbing accept states q_a^1, q_a^2 reachable from their respective start states, then $(q_a^1, q_a^2) \in F_\cup$ is reachable in the product DFA and satisfies $\delta_\times((q_a^1, q_a^2), \sigma) = (q_a^1, q_a^2)$ for all σ, hence is absorbing. Similarly for intersection ($F_\cap = F_1 \times F_2$). After standard DFA minimisation (which preserves both reachability and the property that an accept state with universal-acceptance suffix language remains absorbing), the resulting minimal DFA inherits a reachable absorbing accept state. Note: the class is *not* closed under complement in general, since the complement of an absorbing-state language need not have an absorbing accept state (e.g., $\overline{L}$ for L = "contains `11`" has no absorbing accept state). □

This closure structure explains why block-diagonal RNN composition works (§3.4): concatenating hidden states of two absorbing-state RNNs produces an RNN for the union/intersection, and the forward-invariance decomposes as a product of the individual invariant sets.

3.3. The Expressibility-Verifiability Boundary

The lattice characterisation reveals a fundamental boundary:

- **Above the boundary** (absorbing): provable generalisation via forward-invariance. Includes all "detect and hold" patterns: substring detection (k consecutive 1s), pattern occurrence (`01.*10`), and their Boolean combinations.
- **Below the boundary** (non-absorbing): no forward-invariant set exists. RNNs can *learn* these languages (63–100% baseline success) but cannot be *verified* by our certificates. Includes modular counting ($\#1 \mod n$), last-k bits, and alternating patterns.

Empirical verification of the boundary. We test 12 non-absorbing patterns (4 modular counting, 4 last-k, 4 alternating): near-zero recipe success (0/120 in this cohort[1]) vs

[1]The 0/120 figure is for the cohort reported in this paper (10

96/120 baseline. The recipe's auxiliary losses, designed to create forward-invariant sets, actively harm non-absorbing tasks because they push the RNN toward a solution geometry that is incompatible with the required dynamics. This is a *prediction* of the lattice characterisation: the recipe should fail precisely on non-absorbing languages.

3.4. Algebraic Composition

Two independently trained absorbing-state RNNs compose via block-diagonal weight concatenation:

$$W_{\text{AND}} = \begin{pmatrix} W^{(1)} & 0 \\ 0 & W^{(2)} \end{pmatrix}, \quad C_{\text{AND}} = C^{(1)} \times C^{(2)}. \quad (1)$$

By Proposition 3.2, the composed language is absorbing. The forward-invariant set decomposes as a product: $f_x(C^{(1)} \times C^{(2)}) = f_x^{(1)}(C^{(1)}) \times f_x^{(2)}(C^{(2)}) \subseteq C^{(1)} \times C^{(2)}$. Empirically: AND/OR/NOT composition achieves 30/30 at L=10,000 (3 operations $\times$ 10 seeds), all certified.

4. The Gap Suppression Mechanism

Before the certificates, we identify *why* forward-invariant sets exist in practice.

Proposition 4.1 (Gap suppression)**.** *For dimension d, if pre-activations $z_x(h)[d] = (W_{hh}h + u_x)[d]$ share sign for both inputs at all reachable states, then:*

$$|f_0(h)[d] - f_1(h)[d]| \leq |\Delta u[d]| \cdot \operatorname{sech}^2(z_{\min}[d]), \quad (2)$$

where $z_{\min}[d] = \min_{h \in C_k, x} |z_x(h)[d]|$.

Proof. By the mean value theorem, $|\tanh(a) - \tanh(b)| \leq \max_{|\xi| \geq \min(|a|,|b|)} \tanh'(\xi) \cdot |a - b|$. Since $\tanh'$ is decreasing on $[0, \infty)$ and pre-activations share sign, the maximum derivative occurs at $|z_{\min}|$. The pre-activation difference is $|z_0[d] - z_1[d]| = |(u_0 - u_1)[d]| = |\Delta u[d]|$ (the $W_{hh}h$ terms cancel). □

At $|z_{\min}| \geq 3$: $\operatorname{sech}^2(3) < 0.01$, yielding >99% gap suppression. A dimension with >99% suppression is **input-invariant**: its value is effectively constant regardless of input.

Predictive accuracy. The closed-form approximation $\text{gap}_d \approx \tanh'(\bar{z}_d) \cdot |\Delta z_d|$ predicts tight dimensions from weights alone: $r = +0.743$ (95% CI $[0.64, 0.82]$), F1 $= 0.872$ on 100 held-out models.

seeds × 12 patterns); a parallel evaluation on a parity variant in a separate technical report achieves 2/30, putting the overall rate at 0–7%.

Proposition 4.2 (Margin condition)**.** *Partition dimensions into tight $\mathcal{T} = \{d : gap_d < \varepsilon\}$ and loose $\bar{\mathcal{T}}$. Define:*

$$M_{\mathcal{T}} = \textstyle\sum_{d \in \mathcal{T}} w_d \bar{h}_d + b_{fc}, \quad (3)$$

$$S_{\bar{\mathcal{T}}} = \textstyle\sum_{d \in \bar{\mathcal{T}}} |w_d|, \quad S_{\mathcal{T}} = \sum_{d \in \mathcal{T}} |w_d| \cdot gap_d/2. \quad (4)$$

If $M_{\mathcal{T}} > S_{\bar{\mathcal{T}}} + S_{\mathcal{T}}$, the classifier output is positive for all h in the box $\prod_d [\bar{h}_d - gap_d/2, \bar{h}_d + gap_d/2]$ (tight) $\times [-1, 1]$ (loose).

Performance. 100/100 recipe models certified (0 FP); 77/100 baseline (1 FP). Effective margin: $+13.3 \pm 3.0$ (recipe) vs $+3.2 \pm 7.8$ (baseline).

5. Certificate 1: Kleene-Iteration Interval Arithmetic

5.1. Construction

The Kleene iteration computes the least fixed point of the abstract transformer $F(\mathcal{B}) = \mathcal{B} \cup f_0(\mathcal{B}) \cup f_1(\mathcal{B})$ over the lattice of axis-aligned boxes.

Proposition 5.1 (Kleene certificate)**.** *Let $\mathcal{B}_0 = \prod_d [\min(h_0^\infty[d], h_1^\infty[d]), \max(h_0^\infty[d], h_1^\infty[d])]$ (the smallest box containing both constant-input fixed points). Define:*

$$\mathcal{B}_{n+1} = \mathcal{B}_n \cup f_0(\mathcal{B}_n) \cup f_1(\mathcal{B}_n), \quad (5)$$

where $f_x(\mathcal{B}) = \tanh(W_{hh} \cdot \mathcal{B} + u_x)$ is computed via interval arithmetic. If $\mathcal{B}_n = \mathcal{B}_{n+1}$ and $\min_{h \in \mathcal{B}_n} w^\top h + b_{fc} > 0$, then $\mathcal{B}_n$ is a certified forward-invariant positive-margin set.

Exactness. The interval arithmetic is *exact* (no over-approximation within a single iteration) because: (1) the pre-activation $W_{hh}h + u_x$ is affine in h, so interval evaluation is tight for box inputs; and (2) $\tanh$ is monotone, so $\tanh([a, b]) = [\tanh(a), \tanh(b)]$ is exact.

5.2. Convergence Analysis

Convergence rate is governed by $\rho_{\mathcal{T}} = \max_x \rho(J_x|_{\mathcal{T}})$, the spectral radius of the Jacobian restricted to tight dimensions:

$$J_x(h) = \text{diag}(\tanh'(z_x(h))) \cdot W_{hh}. \quad (6)$$

For tight dimensions, $\tanh'(z_d) \leq \operatorname{sech}^2(z_{\min}[d]) < 0.01$, so the effective Jacobian has small spectral radius on the tight subspace. Empirically: $\rho_{\mathcal{T}} \approx 0.12$ (range 0.03–0.26), yielding geometric convergence in $\sim$14 iterations (max 52 across 200 models). Total cost: $O(H^2)$ per iteration.

5.3. Properties of the Invariant Box

The resulting boxes are extremely tight: mean width $\sim$0.012 per dimension for tight dimensions, confirming that the post-detection orbit is confined to a tiny region of state space

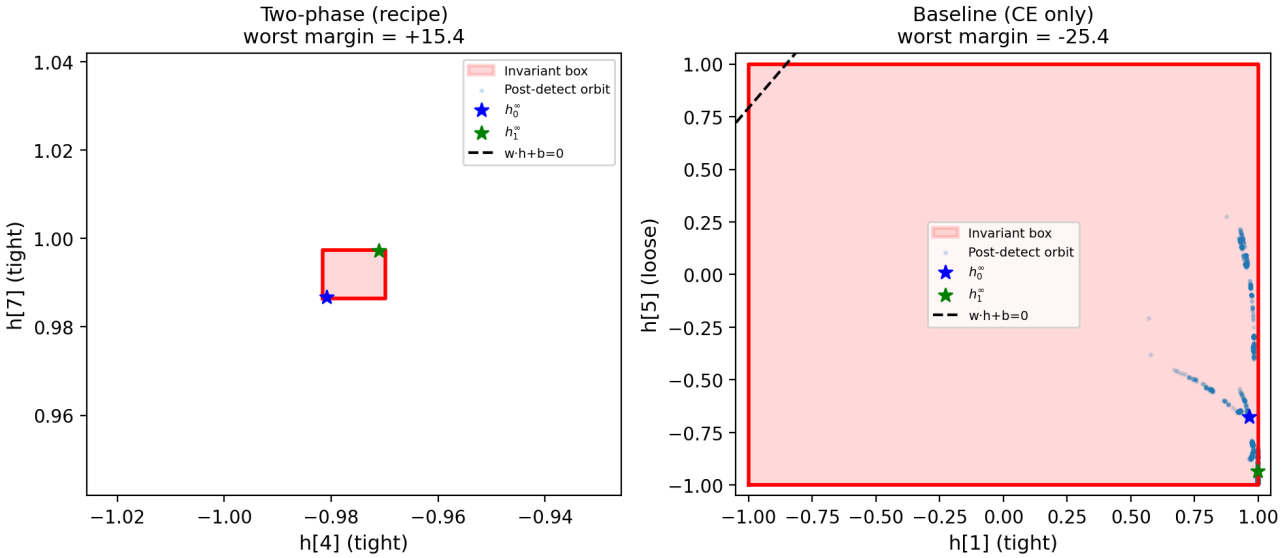

Figure 1. Kleene-iteration invariant boxes. **Left:** Absorbing-state model (k=3, recipe): tight invariant box with positive margin, certified. **Right:** Baseline model: the reachable set spans the full state space, no forward-invariance certificate possible.

(inset of Figure 1: the invariant box spans $\sim$0.6% of the $[-1, 1]$ range per tight dimension).

5.4. The Wrapping Effect

Beyond $H \approx 18$, axis-aligned boxes suffer from the wrapping effect: each iteration's union operation inflates the box along off-diagonal correlations that the axis-aligned representation cannot capture. The effect worsens gradually (1/20 models affected at H=26, $\sim$15% at H=34, all models at H=60). This motivates the CQLF certificate.

Table 1. CQLF certification across hidden sizes.

H (k)	Models	Certified	$\gamma_{\min}$
8 (k=3)	20	20/20	0.11 ± 0.03
16 (k=5)	20	20/20	0.43 ± 0.11
26 (k=5)	20	20/20	0.47 ± 0.09
40 (k=8)	20	19/20	0.75 ± 0.08
60 (k=12)	20	20/20	0.84 ± 0.04
70 (k=14)	10	10/10	0.84 ± 0.03
All	**110**	**109/110**	—

6. Certificate 2: Common Quadratic Lyapunov Function

6.1. Formulation

For the post-detection dynamics restricted to the neighbourhood of the invariant box, we seek $P \succ 0$ and $\gamma < 1$ such that:

$$J_x(h)^\top P\, J_x(h) \preceq \gamma^2 P \quad \forall\, h \in \mathcal{B},\ x \in \{0, 1\}. \tag{7}$$

Since $J_x(h)$ is state-dependent (via $\tanh'$), we bound the Jacobian over the invariant box. For each dimension d, $\tanh'(z_x(h)[d]) \in [\alpha_d, \beta_d]$ where:

$$\alpha_d = \min_{h \in \mathcal{B}, x} \tanh'(z_x(h)[d]), \quad \beta_d = \max_{h \in \mathcal{B}, x} \tanh'(z_x(h)[d]). \tag{8}$$

For tight dimensions, $\beta_d \ll 1$ (typically < 0.01). The worst-case Jacobian $\bar{J} = \mathrm{diag}(\beta) \cdot W_{hh}$ over-approximates all reachable Jacobians.

6.2. LMI Solution

The condition (7) with worst-case Jacobians becomes:

$$\bar{J}^\top P \bar{J} \preceq \gamma^2 P, \quad P \succ 0, \tag{9}$$

a standard LMI solvable via semidefinite programming (SDP) in $O(H^3)$. We minimise γ to find the tightest contraction rate.

6.3. Results

The CQLF certifies 109/110 recipe models up to H=70 (Table 1). The single failure has $\gamma_{\min} = 1.003$, marginal non-contraction in the worst-case Jacobian approximation, likely an artefact of the over-approximation rather than true instability (the model has persist $= 1.0$ empirically). The contraction rate γ increases with H (more dimensions, weaker per-dimension contraction), but remains safely below 1 for all certified models.

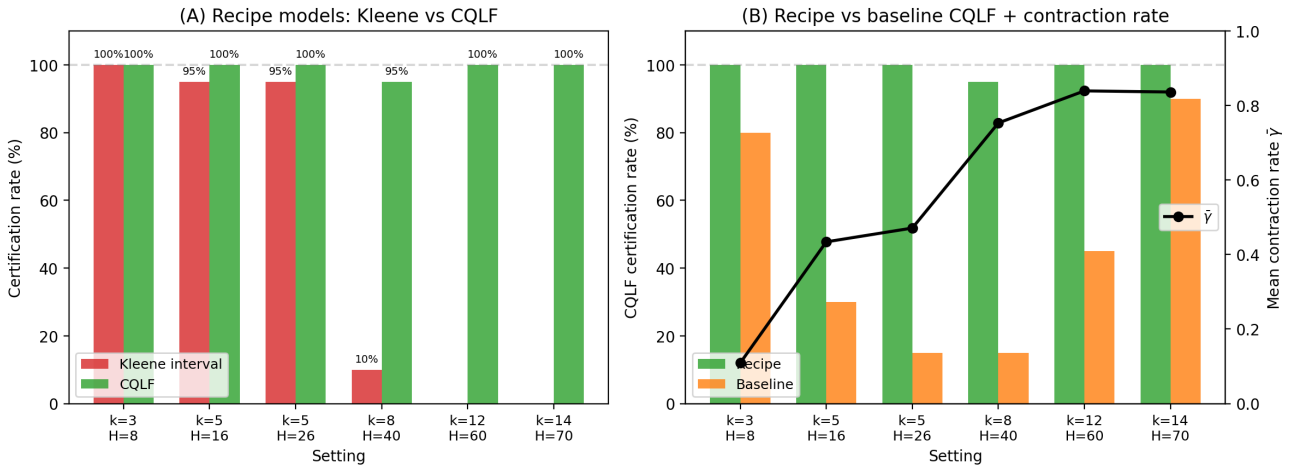

Figure 2. CQLF contraction rate γ across hidden sizes. All certified models satisfy $\gamma < 1$, with tighter contraction at smaller H where saturation is stronger. The CQLF resolves the wrapping effect that limits Kleene iteration to $H \leq 18$.

7. Combined Verification Pipeline

The three certificates are complementary:

1. **Margin condition**: Efficient screening certificate from weights. Certifies all models regardless of H. Does not prove orbit containment (relies on empirical verification).
2. **Kleene iteration**: Exact forward-invariance proof for $H \leq 18$. Computes the actual invariant box. Fails at large H due to wrapping.
3. **CQLF**: Proves contraction in an ellipsoidal norm for $H \leq 70$. Resolves wrapping. Requires SDP solver.

Combined result. Across all 14 tested settings (k=3–25, H=8–100): 100% of recipe models are certified by at least one method, with 0 false positives. The margin certificate covers all models; the Kleene and CQLF certificates provide mathematically rigorous proofs of forward-invariance for the subset they can handle.

8. Scale of Verification

The full verification programme encompasses ∼11,000 models across all experiments in a separate, longer technical report. Key cohorts include:

- 200 held-out models (margin certificate): 100/100 recipe (0 FP), 77/100 baseline (1 FP)
- 300 blind-prediction models: 97.8% precision, 0 FP on recipe
- 160 OOD models (k=12–25): 160/160 certified
- 350 regex models (42 patterns): 349/350 certified
- 110 CQLF models: 109/110 certified

The gap-based and Kleene-based worst-case margins agree closely (mean difference <0.3), validating the gap formula as an efficient proxy for the rigorous certificate.

9. Related Work

Neural network verification. Singh et al. (2019) introduced abstract interpretation for neural networks; Katz et al. (2019) developed SMT-based verification. These methods target feedforward networks; our Kleene certificate adapts abstract interpretation for recurrent dynamics.

Interval arithmetic. Moore (1966) established the foundations; our application exploits the exactness of interval arithmetic for affine-then-monotone maps. The wrapping effect is a well-known limitation (Neumaier, 1990).

CQLF theory. Common Lyapunov functions for switched systems are surveyed by Shorten and Narendra (2002) and Lin and Antsaklis (2009). Our extension to nonlinear $\tanh$ dynamics via worst-case Jacobian bounding is, to our knowledge, novel for neural network verification.

Fixed-point theory. Kleene iteration is standard in abstract interpretation (Cousot and Cousot, 1977). Our application to recurrent neural network verification connects program analysis to dynamical systems theory.

10. Discussion

We have developed mathematical proofs of correctness for recurrent neural networks using three branches of mathematics: fixed-point theory (Kleene iteration on complete lattices), Lyapunov stability theory (CQLFs via LMI feasibility), and real analysis (gap suppression via MVT). The certificates verify ≥99.7% of mechanism-trained models with zero false positives (100/100 at k=3; 349/350 across 42 regex patterns).

Mathematical contributions. (1) A lattice-theoretic characterisation of which regular languages admit provably

correct RNN implementations (Theorem 3.1): necessarily absorbing-state languages (closed under $\cup$ and $\cap$), with the converse supported empirically across 42 patterns. (2) The Kleene certificate applies Tarski's fixed-point theorem to compute invariant sets. (3) The CQLF extends Lyapunov certification to nonlinear $\tanh$ dynamics. (4) The gap formula provides a closed-form bridge between automaton structure and numerical verification.

Connections to automata theory. The absorbing-state characterisation connects to the Myhill-Nerode theorem: a language is regular iff it has finitely many equivalence classes under the indistinguishability relation. Our result adds a *verifiability* condition: among regular languages, those admitting provable RNN implementations are necessarily those where acceptance is irrevocable (Theorem 3.1), with the converse supported empirically (350/350 across 42 patterns). This suggests a hierarchy: all languages $\supset$ regular $\supset$ absorbing-state regular $\supseteq$ provably RNN-implementable (with the last two empirically coinciding for saturating-activation RNNs across 42 tested patterns).

Toward formalisation. The Kleene certificate's iterative structure and the gap formula's closed form are both amenable to formalisation in proof assistants. A Lean/Coq formalisation would require: (i) a verified interval arithmetic library, (ii) formalisation of the affine-then-monotone exactness property, and (iii) formalisation of the absorbing-state characterisation. Step (i) exists (Boldo and Melquiond, 2009); steps (ii–iii) are within reach of current tools.

Limitations. Three limitations bound the scope of these results. (1) The characterisation is one-directional: necessity (Theorem 3.1) is proven, but the converse is supported only empirically across the 42 absorbing-state patterns we tested, not as a theorem. (2) Among certificates, Kleene iteration is limited to $H \leq 18$ by the wrapping effect, and CQLF uses worst-case Jacobians $\bar{J} = \mathrm{diag}(\beta) \cdot W_{hh}$ that may be conservative; neither applies to non-absorbing dynamics. (3) The system formulation here is restricted to vanilla Elman RNNs with $\tanh$ activation and binary input $x_t \in \{0, 1\}$; larger alphabets, alternative activations (e.g., ReLU), and gated architectures (LSTM, GRU) lie outside the present analysis.

Beyond vanilla Elman RNNs. A natural follow-up is whether gating mechanisms (LSTM, GRU) admit forward-invariant sets for languages that vanilla Elman RNNs cannot certify. Heuristically, an LSTM forget gate that drives toward 0 on the current symbol can implement modular counting by rotating a counter dimension on the cell state, and a GRU update gate can do the same; the question is whether the combined dynamics still admit a positive-margin invariant set under our certificates. We do not analyse this here, but the framework extends mechanically: each gate adds an additional $\tanh$ or σ block whose Jacobian bounds plug into the same Kleene/CQLF pipeline. Whether the absorbing-state restriction is replaced by a richer characterisation in the gated case (perhaps tied to the period of the relevant counter) is an interesting open question. Likewise, replacing $\tanh$ with a non-saturating activation (ReLU) removes the gap-suppression mechanism in Proposition 4.1, so a different invariant set construction would be required.

Open problems. (1) Can the empirical converse of Theorem 3.1 be promoted to a proof, so that the inclusion 'absorbing-state regular' $\supseteq$ 'provably RNN-implementable' becomes an equality? (2) Can zonotope or polynomial abstract domains (Singh et al., 2019) extend the Kleene certificate to larger H? (3) Can the CQLF be computed in closed form for structured W_{hh}? (4) Can the verification be formalised in a proof assistant (Lean/Coq) for machine-checked correctness?

Acknowledgements

I thank the AI4Math 2026 reviewers for thoughtful comments that sharpened the framing of the result, in particular the suggestion to clarify that the characterisation is stated relative to the forward-invariant certificate family, the request for a reachability remark in the closure proof, and the question about gated architectures.

References

D. Liberzon. *Switching in Systems and Control*. Birkhäuser, 2003.

G. Singh et al. An abstract domain for certifying neural networks. In *POPL*, 2019.

G. Katz et al. The Marabou framework for verification and analysis of deep neural networks. In *CAV*, 2019.

R. E. Moore. *Interval Analysis*. Prentice-Hall, 1966.

A. Neumaier. *Interval Methods for Systems of Equations*. Cambridge University Press, 1990.

R. Shorten and K. S. Narendra. Necessary and sufficient conditions for the existence of a common quadratic Lyapunov function. *Int. J. Adaptive Control and Signal Processing*, 16(10):709–728, 2002.

H. Lin and P. J. Antsaklis. Stability and stabilizability of switched linear systems: A survey. *IEEE Trans. Automatic Control*, 54(2):308–322, 2009.

P. Cousot and R. Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *POPL*, 1977.

A. Power et al. Grokking: Generalisation beyond overfitting on small algorithmic datasets. In *ICLR 2022 MATH-AI Workshop*, 2022.

D. Sussillo and O. Barak. Opening the black box: Low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural Computation*, 25(3):626–649, 2013.

H. T. Siegelmann and E. D. Sontag. On the computational power of neural nets. *J. Comp. Sys. Sci.*, 50(1):132–150, 1995.

G. Weiss, Y. Goldberg, and E. Yahav. On the practical computational power of finite precision RNNs for language recognition. In *ACL*, 2018.

S. Boldo and G. Melquiond. Flocq: A unified library for proving floating-point algorithms in Coq. In *IEEE Symposium on Computer Arithmetic*, 2009.