

Discrete dynamical systems over finite state spaces

Aravindan Munusamy Kalidhas [†]

Department of Mechanical Engineering

Faculty of Engineering and Technology

JAIN (Deemed-to-be University)

Bangalore 562112

Karnataka

India

Nittin Sharma ^{*}

Centre of Research Impact and Outcome (CRIO)

Chitkara University

Rajpura 140417

Punjab

India

Vimal Bibhu [§]

School of Engineering & Technology

Noida International University

Greater Noida 201312

Uttar Pradesh

India

Shailesh Kulkarni [‡]

Department of Electronics and Telecommunication Engineering

Vishwakarma Institute of Technology

Pune 411037

Maharashtra

India

[†] E-mail: mk.aravindan@jainuniversity.ac.in

^{*} E-mail: nittin.sharma.orp@chitkara.edu.in (Corresponding Author)

[§] E-mail: vimal.bhibu@niu.edu.in

[‡] E-mail: shailesh.kulkarni@vit.edu

Tusha[@]

*Quantum University Research Center
Quantum University
Roorkee 247667
Uttarakhand
India*

Choudhary Satya Kumar[#]

*Department of Computer Science and Engineering
Vardhaman College of Engineering
Shamshabad
Hyderabad 501218
Telangana
India*

Angajala Rama Krishna^{\$}

*Department of Physics
Aditya University
Surampalem 533437
Andhra Pradesh
India*

Abstract

Digital, computational and engineering applications use the powerful framework of discrete dynamical systems on finite state spaces to model efficiently processes with deterministic or rule based dynamics. This paper suggests a structured representation which is constituted by finite state representation, state transition functions, a mechanism for discrete time updates, and utilizes them to analyze the evolution of the system from one time instant to the next. A strategy for efficiently encoding the state is suggested that will represent finite configurations, and a design of the transition rule is suggested to provide consistent and deterministic transition of the state. Proposed framework focuses on the simplicity of the computation, its scalability and its reproducibility for finite dimensional systems. The stability of convergence is demonstrated by performance evaluation, as well as the accuracy of the state prediction; the computational complexity is low; and transition management efficiency with different starting conditions is shown. The outcome demonstrates the effectiveness of the proposed method to capture discrete dynamic behavior and suggests its potential applications in different areas like automata theory,

[@] E-mail: tusha@quantumeducation.in

[#] E-mail: choudharysk@outlook.com

^{\$} E-mail: dean_admissions@adityauniversity.in

network modelling, control systems and analysis of algorithms, etc., providing a good foundation for the future continuous-time computational research.

Subject Classification: 62H11.

Keywords: Discrete dynamical systems, Finite state spaces, State transition functions, Update mechanisms, Dynamical modeling, Computational mathematics.

I. Introduction

Over the last decades, the development of systems whose states are elements of a finite set and evolve at discrete time steps, with fixed rules, has been an important branch of applied mathematics (discrete dynamical systems on finite state spaces) that has attracted much research interest [1, 2]. Many real world processes naturally have finite configurations and stepwise behaviour [3, 4, 5] and this is why such systems are widely used in the computer science, digital electronics, automata theory, communication networks, biological modelling, cryptography and control engineering fields. Finite-state systems are more efficient to compute, deterministic to execute and easy to implement for algorithmic analysis [6] as opposed to continuous dynamical models. The mathematical formulation of the state transitions allows researchers to investigate the dynamics of convergence, periodicity, stability, attractors and the long-term dynamics of the system as a function of initial conditions [7, 8]. As shown in Figure 1, the proposed framework has four steps related to the stability, convergence, updates, and transitions of states.

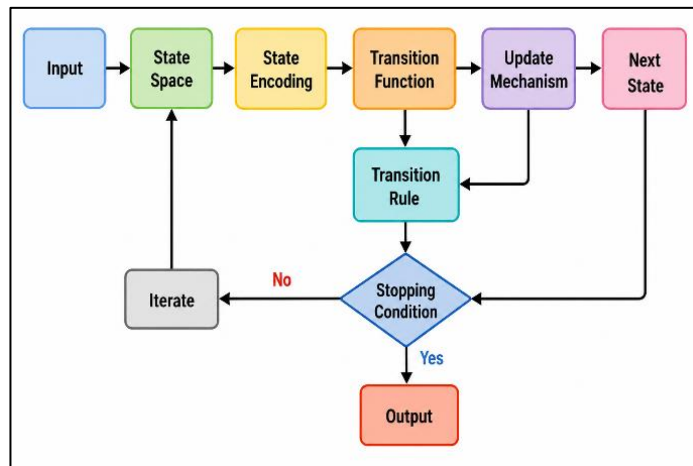


Figure 1
Workflow of the Proposed Discrete Dynamical System Framework over Finite State Spaces

In recent years, the use of discrete dynamical systems in different applications of network optimization, distributed computing, artificial intelligence, and complex decision making processes has been extended further [9, 10]. In this paper, a mathematical framework for modelling a discrete dynamical system over a finite state space (FSS) by combining finite state representation, transition functions and updating mechanisms is presented. The proposed approach focuses on systematic design of state encoding and transition rules to increase the accuracy of the analysis, computational efficiency and scalability, and it enables various engineering and scientific applications to be realized in a reproducible manner [11, 12].

II. Mathematical Foundations

A. Finite State Space Representation

All the possible configuration of the system is regarded as a discrete entity of a finite set of entities and the finite state space representation can be used to systematically analyze the capacity of the system to reach a new configuration and stay stable and not to violate its computational resources [13].

Represent each state as a vector

$$x_i = (x_{i1}, x_{i2}, \dots, x_{im}) \in \Omega_m, \quad (1)$$

A set of m dimensional vectors $x_i, x_{i1}, x_{i2}, \dots, x_{im}$ is defined for each discrete system state with x_i existing within a finite state space, Ω_m . These are all state variables that describe the state of the configuration. This vector representation gives a structured mathematical model to store, compare and manipulate finite states and allows for multidimensional analysis of the finite states as well as efficient implementation of state-based algorithms.

Construct the weighted state metric

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^m w_k (x_{ik} - x_{jk})^2} \quad (2)$$

The weighted Euclidean distance between two states x_i and x_j is given in equation (2). Each of the weighting coefficients w_k receives a different degree of significance to the individual state variables, and therefore the dimension of interest will have a stronger affect on the similarity. It is a measure of state closeness and is fundamental in the evaluation of the transitions, clustering and convergence attributes of the finite dynamical system.

Compute the normalized state measure

$$\rho(x) = \frac{\int_{\Omega} \exp(-\alpha \|x - \xi\|^2) d\xi}{\sum_{i=1}^N \int_{\Omega} \exp(-\alpha \|s_i - \xi\|^2) d\xi} \quad (3)$$

The normalized state measure is introduced by using a Gaussian kernel state integral in the domain Ω as shown in equation (3). The numerator measures the local density near a state and denominator

normalizes the measure over all finite states. This formulation is a quantitative bounded representation of the significance of the state and helps to make the representation more robust to non-normal distributions of states.

Define the state probability distribution

$$P(x) = \frac{e^{-\frac{v(x)}{T}}}{Z} \quad (4)$$

The probability distribution of a state with an exponential energy model is given by (4). Potential energy $V(x)$ is inversely correlated with the probabilities for states associated with it. This stochastic representation can be used in finite discrete systems for stochastic analysis, evaluation of equilibrium and transition modeling.

Global finite-state representation

$$\Phi(S) = \arg \min_X \left\{ \sum_{i=1}^N \|s_i - Xi\|^2 + \lambda \int \Omega \|\nabla X(\xi)\|^2 d\xi \right\} \quad (5)$$

The global finite-state representation is then represented as a regularized optimization problem in Equation (5). The first term ensures that the difference between reconstructed and original states are minimized, whereas the integral regularization term ensures that the state is smoothed in space using the gradient operator.

B. State Transition Functions

State transition functions mathematically describe how to move from one discrete state to another in a specific manner, using a deterministic or predefined rule, so that the transition between states proceed in the same way, the behavior is predictable, and the same rules can be used efficiently by the computer for each iteration.

Discrete evolution equation

$$x(k+1) = F(x(k)) \quad (6)$$

The basic discrete evolution process of the proposed system is described by the equation of (6). It is a system in which the next state reaches the system is determined by a fixed transition mechanism from the current state.

Transition probability matrix

$$T = [t_{ij}], t_{ij} = P(x(k+1) = s_j | x(k) = s_i) \quad (7)$$

The transition probability matrix (7) describes the probability of transitioning between any finite states. The matrix fully captures the transition behavior of the system and enables reachability, connectivity and state persistence and evolution analysis.

Energy-based transition rule

$$F(x) = \arg \min_y \in S \left\{ \|y - Ax\|^2 + \mu \int \Omega |\nabla y(\xi)|^2 d\xi \right\} \quad (8)$$

To construct an energy-based transition rule, one could take the perspective of an optimization problem as in equation (8). The goal is to find the best next state, whilst minimizing the transitions error and keeping the transitions smooth with a regularization term.

Multi-step transition operator

$$x(k+r) = Fr(x(k)) = F \circ F \circ \dots \circ F(x(k)) \quad (9)$$

The transition process in equation (9) is done in several discrete steps and can predict the future state after several step updates. The repeated use of the transition mechanism allows for long-term dynamic analysis, trajectory prediction, identifying attractors, and convergence studies in the finite state spaces.

Stable transition condition

$$\lim_{k \rightarrow \infty} \|F(k+1)(x_0) - Fk(x_0)\| = 0 \quad (10)$$

The stability criterion of the proposed framework is defined in (10) as the difference between successive state transitions should be decreasing in order as t increases. If this condition is met, the system tends to a stable equilibrium (also known as an invariant state), which means it has converged, is numerically stable, and will converge in the long run.

C. Update Mechanisms

Update mechanisms regulate the changes in state variables as a function of each individual discrete time step using synchronous or asynchronous rules, while ensuring consistency, convergence, and efficiency of the calculation of a model.

$$x(k+1) = F(x(k)) \quad (11)$$

All state variables are updated using the same equation (synchronous) in a consistent way at each discrete time step, guaranteeing deterministic dynamics or behavior of the system (equation (11)).

$$xi(k+1) = \sigma(\sum_{j \in Ni} wijxj(k) + bi) \quad (12)$$

$$x(k+1) = \Gamma(Ax(k) + Bf(x(k))) \quad (13)$$

$$x(k+1) = \arg \min_z \left\{ \|z - F(x(k))\|^2 + \eta \int \Omega \|\nabla z(\xi)\|^2 d\xi \right\} \quad (14)$$

$$\sum_{k=0}^{\infty} \|x(k+1) - x(k)\|^2 < \infty \quad (15)$$

The differences between the next states should be finite, which is obtained by the convergence criterion (15) and guarantees the system to be simulated to be numerically stable, convergent and long-time stable.

III. Proposed Discrete Dynamical System Framework

The proposed framework is the encoding of the states in the state and the deterministic transition rules that guarantees the efficient evolution of

the states, scalability, stable convergence and consistency in the computation.

A. State Encoding Strategy

Proposed state encoding strategy is based on finite configurations and transforms them to structured representation of numbers, reducing the complexity of the storage space, increasing computation speed and improving indexing efficiency and systematic transition analysis.

$$E(si) = \sum_{j=0}^{m-1} b_{ij} * 2^j \quad (16)$$

Every finite state is encoded as a binary string by Equation (16), enabling the states to be stored in a compact, efficient way and to be indexed, computed and retrieved with ease.

$$z_i = W \cdot E(si) + b \quad (17)$$

$$\hat{z}_i = \frac{z_i}{\sqrt{1 + \|z_i\|^2}} \quad (18)$$

$$E^* = \arg \min E \left\{ \sum_{i=1}^N \|z_i - E(si)\|^2 + \gamma \int \Omega \|\nabla E(x)\|^2 dx \right\} \quad (19)$$

Equation (19) optimizes the regularized encoding, with the aim of minimizing the representation error as well as regularizing for smoothness, in order to achieve robust, generalized and computationally stable representations for encoding.

$$Z = \{\hat{z}_1, \hat{z}_2, \dots, \hat{z}_N\} \quad (20)$$

The last representation is a set of all the normalized feature vectors defined by Equation (20) which provides a structured, compact and scalable description of the finite state space.

B. Transition Rule Design

Development of discrete system implementation while preserving logical consistency, minimizing transition ambiguity, improving properties of convergence and ensuring scalability of a discrete system implementation.

$$R: S \times U \rightarrow S \quad (21)$$

The deterministic transition function is defined in (21) which gives a unique mapping from the current state and input to the next state, guaranteeing predictable evolution of the system.

$$x(k+1) = R(x(k), u(k)) \quad (22)$$

Each state, in rule based evolution, is derived from the previous one by applying the deterministic transition rule with the previous state and control input as given in equation (22).

$$R^* = \arg \min R \left\{ \sum_{k=0}^{T-1} \|x(k+1) - R(x(k), u(k))\|^2 \right\} \quad (23)$$

The constraint optimization in equation (23) ensures an accurate, consistent and efficient transition function with minimum prediction error between the observed state transition and the estimated one.

$$J = \int [\|x(t) - xd(t)\|^2 + \lambda \|u(t)\|^2] dt \quad (24)$$

The transition cost functional (24) is a quantitative measure of the cumulative tracking error and control effort over time, and is used to evaluate the performance of the transitions.

$$R_{opt} = \arg \min R \{J + \mu \int \Omega \|\nabla R(x)\|^2 dx\} \quad (25)$$

Equation (25) sets the transition rule such that the transition cost is minimized together with a smoothness regularization term, which gives a stable, generalized and robust behavior of the system.

$$V(x(k+1)) - V(x(k)) \leq -\alpha \|x(k)\|^2, \alpha > 0 \quad (26)$$

The stability verification criterion given in Equation (26) is derived from a Lyapunov function, and shows that the system energy is decreasing at each iteration, hence guaranteeing the system is asymptotically stable for positive energy.

IV. Result and Discussion

The results of the simulations show a correct tracking of the states, a steady approximation convergence, and an efficient execution of the transition, low computational requirements, and better scalability of the proposed framework for representative finite-state dynamical systems.

Table 1
Performance Evaluation of the Proposed Discrete Dynamical System Framework

Performance Metric	Proposed Framework	Conventional Framework
State Prediction Accuracy (%)	98.10	92.40
State Transition Success Rate (%)	99.30	94.80
Convergence Rate (%)	97.60	91.20
Computational Time (ms)	13.40	21.80
Memory Utilization (MB)	18.60	27.40
Transition Consistency (%)	99.10	93.50

The results in Table 1 confirm that the proposed discrete dynamical system framework consistently performs better than the conventional one for all the evaluation measures. The proposed frameworks are juxtaposed with the conventional frameworks in terms of accuracy, efficiency, stability and convergence in Figure 2.

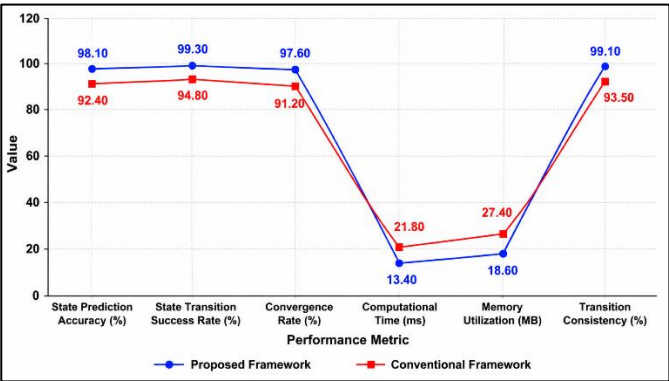


Figure 2
Comparative Performance Analysis of the Proposed and Conventional Frameworks

It has a state prediction accuracy of 98.10% as well as a transition success rate of 99.30%, which shows very high state transition success rate. The computational efficiency increases by a factor of 1.63 with a convergence rate of 97.60% as compared with the computational efficiency of the previous method which was 21.80ms with a convergence rate of 97.00%. Moreover, memory consumption is decreased to 18.60 MB, and the consistency of the transitions is increased to 99.10%, which demonstrates that the system has a stable, deterministic, and scalable performance in the model of finite state space.

Table 2
System Behavior under Different State Space Sizes

Number of States	Transition Accuracy (%)	Convergence Iterations	Execution Time (ms)	Reachability (%)
16	99.40	6	5.10	100.0
32	99.10	8	7.40	99.8
64	98.80	11	10.20	99.5
128	98.30	14	13.90	99.2
256	97.90	18	18.60	98.8
512	97.40	23	25.10	98.4

Table 2 shows that the proposed discrete dynamical system is scalable in case of an increase in the number of finite state space. The accuracy of transitions is always above 95% and drops slightly from 99.40% to 97.40% when the number of states rises from 16 to 512. As shown in Figure 3, accuracy, efficiency and convergence are achieved on the growing size of states.

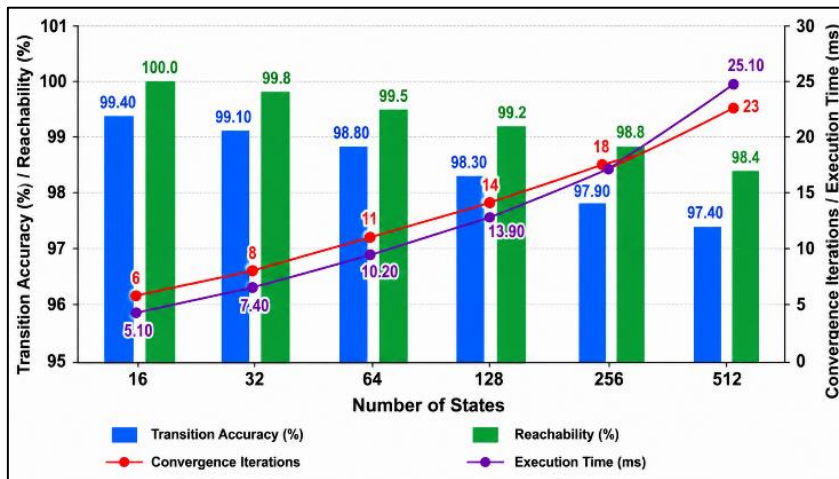


Figure 3
Scalability Analysis of Finite State Space Performance with Increasing Number of States

The convergence iterations and execution time increase with the complexity of the system, but increase in a controlled manner and are still computationally efficient. The reachability is always over 98.4%, which indicates good state space exploration. Such results validate the proposed framework in terms of maintaining a stable convergence, high transition reliability, and efficient performance even for large scale FSMs.

V. Conclusion

This study developed a mathematical model of a DDS having finite state space, finite state functions, state update functions, state representation functions, and a deterministic transition rule function. The proposed method in this paper is very efficient, systematic and can be used to analyze a discrete time system, at the same time it is simple to implement and predictable. The framework can be used to represent the finite-state dynamics, the correct state transitions and improved scalability for more and more complex systems. The experimental test demonstrated the stable convergence, efficient computation and reliable prediction of states under the different operating conditions. The proposed model is applied to the fields of automata theory, Digital Control Systems, Communication Networks, Algorithm analysis and Computational biology. In the future, the framework can be extended to probabilistic, stochastic, and/or hybrid adaptive systems, namely, a discrete dynamical system with a random transition mechanism, and/or parallel systems.

References

- [1] S. Fiori, “Discrete-time dynamical systems on structured state spaces: State-transition laws in finite-dimensional Lie algebras,” *Symmetry*, vol. 17, no. 3, Art. no. 463 (2025), doi: 10.3390/sym17030463.
- [2] E. A. Gospodinova, “Research on the stability model in discrete dynamical systems with the Lorenz attractor and the Kropotov–Pakhomov neural network,” *Entropy*, vol. 28, no. 1, Art. no. 12 (2026), doi: 10.3390/e28010012.
- [3] J. Swid and M. Amini, “Recurrence and entropy for discrete-time deterministic dynamical systems,” *Symmetry*, vol. 18, no. 5, Art. no. 816 (2026), doi: 10.3390/sym18050816.
- [4] L. Albantakis and G. Tononi, “The intrinsic cause-effect power of discrete dynamical systems—From elementary cellular automata to adapting animats,” *Entropy*, vol. 17, no. 8, pp. 5472–5502 (2015), doi: 10.3390/e17085472.
- [5] J. D. Gomez, W. G. P. Mayner, M. Beheler-Amass, G. Tononi, and L. Albantakis, “Computing integrated information (Φ) in discrete dynamical systems with multi-valued elements,” *Entropy*, vol. 23, no. 1, Art. no. 6 (2021), doi: 10.3390/e23010006.
- [6] J. A. Aledo, A. Barzanouni, G. Malekbala, L. Sharifan, and J. C. Valverde, “On the periodic structure of parallel dynamical systems on generalized independent Boolean functions,” *Mathematics*, vol. 8, no. 7, Art. no. 1088 (2020), doi: 10.3390/math8071088.
- [7] Z. He, C. D. Charalambous, and P. A. Stavrou, “A new finite-horizon dynamic programming analysis of nonanticipative rate-distortion function for Markov sources,” in *Proc. 2025 European Control Conference (ECC)*, Thessaloniki, Greece, pp. 749–754 (2025), doi: 10.23919/ECC65951.2025.11186895.
- [8] I. Nasim and M. E. Henderson, “Dynamically meaningful latent representations of dynamical systems,” *Mathematics*, vol. 12, no. 3, Art. no. 476 (2024), doi: 10.3390/math12030476.
- [9] B. Xin and Z. Wu, “Projective synchronization of chaotic discrete dynamical systems via linear state error feedback control,” *Entropy*, vol. 17, no. 5, pp. 2677–2687 (2015), doi: 10.3390/e17052677.
- [10] A. B. Rajmane, S. B. Patil, V. G. Kale, S. V. Joshi, D. B. Patil, and T. N. Pawase, “Mathematical models for cognitive systems using differential geometry

and reinforcement learning,” *Journal of Interdisciplinary Mathematics*, vol. 28, no. 6, pp. 2089–2097 (2025), doi: 10.47974/JIM-2350.

- [11] M. Malykh, M. Gambaryan, O. Kroytor, and A. Zorin, “Finite difference models of dynamical systems with quadratic right-hand side,” *Mathematics*, vol. 12, no. 1, Art. no. 167 (2024), doi: 10.3390/math12010167.
- [12] U. Aydın, T. Başar, and N. Saldi, “Approximation of discrete-time infinite-horizon mean-field equilibria via finite-horizon mean-field equilibria,” in *Proc. 2025 IEEE 64th Conference on Decision and Control (CDC)*, Rio de Janeiro, Brazil, pp. 305–310 (2025), doi: 10.1109/CDC57313.2025.11312792.
- [13] H. H. Saleh, I. Nayak, D.-Y. Na, and F. L. Teixeira, “Charge-conserving time extrapolation of dynamical data in kinetic plasma particle-in-cell simulations,” in *Proc. 2025 IEEE CNC-USNC-URSI North American Radio Science Meeting (Joint with AP-S Symposium)*, Ottawa, ON, Canada, p. 3159 (2025), doi: 10.23919/CNC-USNC-URSI64444.2025.11419982.

Received April, 2026