

Pattern formation of generalized fuzzy elementary cellular automaton

Seiryu Shimizu¹ and Tetsuji Tokihiro¹

¹Department of Mathematical Engineering, Musashino University, 3-3-3 Ariake, Koto-ku, 135-8181, Tokyo, Japan.

Abstract

We propose a general method for constructing a fuzzy cellular automaton from a given cellular automaton. Unlike previous approaches that use fuzzy distinctive normal form, whose update function is restricted to third-order polynomials, our method accommodates a wide range of fuzzification functions, enabling the generation of diverse and complex time-evolution patterns that are unattainable with simpler heuristic models.

We demonstrate that phase transitions in pattern formation can be observed by changing the parameters of the fuzzification function or the mixing ratio between two distinct evolution rules of elementary cellular automata. Remarkably, the resulting generalized fuzzy elementary cellular automata exhibit rich dynamical properties, including stable manifolds and chaos, even in minimal systems composed of just three cells.

Keywords: fuzzy cellular automaton, elementary cellular automaton, fuzzification

1 Introduction

A cellular automaton (CA) is a model of computation and a dynamical system that consists of a grid of cells, where each cell can take one of a finite number of states [1]. The system evolves in discrete time steps, with the state of each cell in the next step being determined by a fixed set of rules based on its current state and those of its neighboring cells. Even if its time evolution rule is simple, a CA can exhibit very complex behavior, generate a variety of patterns, and effectively capture how complex global patterns can arise from simple, local interactions. Hence, it is often used as a mathematical model for a natural or a social phenomenon [2].

An elementary cellular automaton (ECA) is the simplest possible type of CA [3]. It is a one-dimensional two-state three-neighboring CA, that is, the cells are arranged in a single line, each cell can only be in one of two states, typically represented as 0 (off/white) and 1 (on/black), and the state of a cell in the next generation is determined by its own state and the states of its immediate left and right neighbors from the current generation.

Despite its simplicity, it can generate very complex behavior, including fractal and chaotic patterns. It can be used as a mathematical model for pattern formation in chemical or biological systems. It also gives a fundamental mathematical model, a special case of the totally asymmetric simple exclusion process (TASEP), for traffic flow [4].

Although CAs have been used as models for natural and/or social phenomena, the state of a cell in a CA is limited to a discrete set, such as $\{0, 1\}$, which has the advantage of being easy to implement on a computer. However, many real-world phenomena inherently involve uncertainty, ambiguity, or continuous quantities, which cannot be adequately represented by only two states. A fuzzy cellular automaton (FCA) is a system that attempts to overcome this limitation of the discrete model by extending the CA's state space from a discrete set to a continuous interval through a process called *fuzzification*[5]. For a two-state CA (Boolean CA), a systematic method of fuzzification is to express the update rule in disjunctive normal form and replace it as [6]

$$\neg a \rightarrow 1 - a, \quad a \wedge b \rightarrow a \cdot b, \quad a \vee b \rightarrow a + b - a \cdot b \text{ or } \min[a + b, 1].$$

While it is almost impossible to predict the long-term behavior of a Boolean CA like an ECA, there have been reports of cases where mathematical analysis of global behavior and fixed points is possible for fuzzification of some ECAs [7–10]. Furthermore, various applications of FCA to real-world systems have been actively discussed. Key areas include pattern recognition and information processing (e.g., noise removal, edge detection, and compression in image analysis; dataset generation via feature extraction in machine learning [11–13], and the modeling of complex systems (such as urban growth, traffic flow, and natural disasters like forest fires [14–16].

However, the aforementioned FCA based on fuzzy disjunctive normal form (FDNF) generally exhibits only monotonic behavior, even when the original CA displays complex dynamics. Consequently, when a CA that models natural or social phenomena is fuzzified, the characteristic features of those phenomena are lost. Meanwhile, in the current application of FCA to real-world systems, state transition rules (fuzzy transition rules) are typically determined empirically or through methods like genetic algorithms. This makes it extremely difficult to leverage the advantages of FDNF, such as its straightforward numerical computation programs and analytical techniques.

For a given Boolean CA, its fuzzification into a FCA may be defined by the following conditions:

1. The state of an FCA cell takes values in the continuous interval $[0, 1]$.
2. The FCA's fuzzy transition rule must coincide with the original CA rule when all cell states are restricted to $\{0, 1\}$.

While FCAs constructed using FDNF satisfy these conditions, they suffer from the disadvantage mentioned previously. In this paper, we propose a method to construct an FCA by generalizing FDNF through the use of *fuzzification functions*. The resulting FCA construction avoids this drawback. We analyze the FCAs obtained using this method from several perspectives and explore their potential applications. The remainder of this paper is organized as follows. Section 2 reviews the construction of FCAs via FDNF. Section 3 details the method for generating generalized fuzzy elementary automata (GFECAs) by applying fuzzification functions to ECAs. In Section 4, we present several examples of pattern formation from GFECAs. Section 5 analyzes a phenomenon resembling a phase transition—observed by varying parameters in a fuzzification function—using effective dimension related to the ℓ^p -norm. Section 6 investigates the dynamics of minimal three-cell GFECAs, which display non-trivial temporal evolution, based on their orbits and attractors. Finally, Section 7 is devoted to concluding remarks.

2 Fuzzification of ECAs via FDNF

Let us consider ECAs, which are one-dimensional Boolean CAs where the neighborhood of a cell consists of itself and its two immediate neighbors (a neighborhood size of three). There are $2^8 = 256$ possible time evolution rules for ECAs, conventionally numbered from 0 to 255. An ECA governed by rule k is referred to as the rule k ECA or ECA k for short. When we denote by $u_j^t \in \{0, 1\}$ ($j \in \mathbb{Z}$, $t \in \mathbb{Z}_{\geq 0}$) the value of the j cite at time step t , the update rule of the rule k ECA is expressed by a function $F_k : \{0, 1\}^{\times 3} \rightarrow \{0, 1\}$ as

$$u_j^{t+1} = F_k(u_{j-1}^t, u_j^t, u_{j+1}^t) \quad (1)$$

Since the function F_k is determined by the $2^3 = 8$ values of 0 or 1, a_0, a_1, \dots, a_7 , as shown in the Table 1, the rule number k is defined by

$$k = a_0 \cdot 2^0 + a_1 \cdot 2^1 + a_2 \cdot 2^2 + a_3 \cdot 2^3 + a_4 \cdot 2^4 + a_5 \cdot 2^5 + a_6 \cdot 2^6 + a_7 \cdot 2^7$$

| (x, y, z) | $(0,0,0)$ | $(0,0,1)$ | $(0,1,0)$ | $(0,1,1)$ | $(1,0,0)$ | $(1,0,1)$ | $(1,1,0)$ | $(1,1,1)$ |
|----------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| $F_k(x, y, z)$ | a_0 | a_1 | a_2 | a_3 | a_4 | a_5 | a_6 | a_7 |

Table 1 ECA transition table

For example,

$$184 = 0 \cdot 2^0 + 0 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + 1 \cdot 2^4 + 1 \cdot 2^5 + 0 \cdot 2^6 + 1 \cdot 2^7,$$

and the Rule 184 ECA, which is a well-known traffic flow model, is defined from the Table 2.

| (x, y, z) | (0,0,0) | (0,0,1) | (0,1,0) | (0,1,1) | (1,0,0) | (1,0,1) | (1,1,0) | (1,1,1) |
|--------------------|---------|---------|---------|---------|---------|---------|---------|---------|
| $F_{184}(x, y, z)$ | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |

Table 2 Time evolution rule of Rule 184 ECA.

Figure 1 illustrates the time evolution of rule 184 ECA for $0 \leq t \leq 50$ on a lattice of width 50 under a periodic boundary condition. The evolution proceeds downwards from the initial configuration (top row), where white and black cells denote states 0 and 1, respectively.

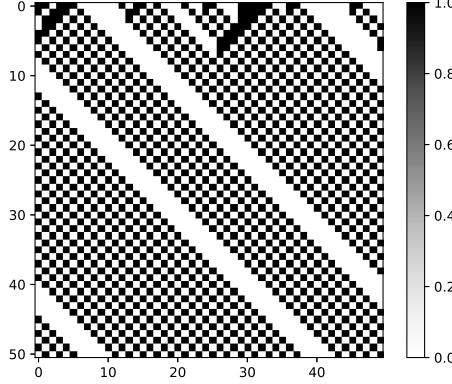


Fig. 1 A time evolution pattern of rule 184 ECA.

Now we consider the fuzzification of ECAs using FDNF [6]. From Eq. (1) and Table 1, the update function of an ECA is written in disjunctive normal form as

$$\begin{aligned}
F_k(x, y, z) = & a_0 ((\neg x) \wedge (\neg y) \wedge (\neg z)) \vee a_1 ((\neg x) \wedge (\neg y) \wedge z) \\
& \vee a_2 ((\neg x) \wedge y \wedge (\neg z)) \vee a_3 ((\neg x) \wedge y \wedge z) \\
& \vee a_4 (x \wedge (\neg y) \wedge (\neg z)) \vee a_5 (x \wedge (\neg y) \wedge z) \\
& \vee a_6 (x \wedge y \wedge (\neg z)) \vee a_7 (x \wedge y \wedge z)
\end{aligned} \tag{2}$$

Here, we use the following notation:

$$\begin{aligned}
A \vee a_j (\dots) &= \begin{cases} A \vee \dots & (a_j = 1) \\ A & (a_j = 0) \end{cases} \\
a_j (\dots) \vee B &= \begin{cases} \dots \vee B & (a_j = 1) \\ B & (a_j = 0) \end{cases}
\end{aligned}$$

The FDNF employs the following fuzzy mappings for Boolean operations: $\neg x \mapsto 1 - x$, $x \wedge y \mapsto x \cdot y$, and $x \vee y \mapsto \min[x + y, 1]$, and the update function $F_k(x, y, z)$ turns

into a polynomial of order 3 as

$$f_k(x, y, z) = a_0(1-x)(1-y)(1-z) + a_1(1-x)(1-y)z + a_2(1-x)y(1-z) + a_3(1-x)yz + a_4x(1-y)(1-z) + a_5x(1-y)z + a_6xy(1-z) + a_7xyz \quad (3)$$

Since $0 \leq f_k(x, y, z) \leq 1$ for $(x, y, z) \in [0, 1]^3$, and $f_k(x, y, z) = F_k(x, y, z)$ for $(x, y, z) \in \{0, 1\}^3$, $f_k(x, y, z)$ serves as an update function of the fuzzy elementary cellular automaton (FECA). For instance, application of Eq. (3) to the rule 184 ECA yields

$$\begin{aligned} f_{184}(x, y, z) &= (1-x)yz + x(1-y)(1-z) + x(1-y)z + xyz \\ &= x - xy + yz \end{aligned} \quad (4)$$

Let us examine time evolution patterns of the FECAs. In contrast to the clear structure seen in Figure 1, the time evolution of rule 184 FECA (Figure 2) appears largely uniform. This is because the system is provably destined to become a monotone pattern when the number of cells is odd, and a checkerboard pattern when it is even[15].

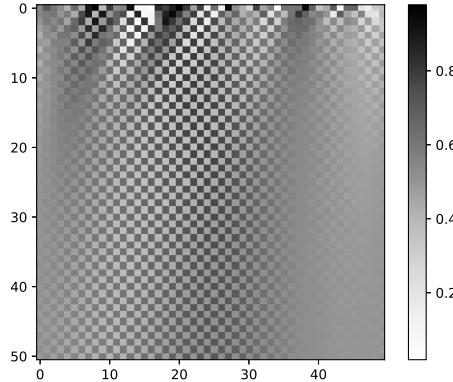


Fig. 2 A time evolution pattern of rule 184 FECA obtained by FDNF. The number of cells is 50, the initial values are chosen randomly from $[0, 1]$, and a periodic boundary condition is imposed.

Figures 3 and 4 show the results of applying FDNF to rules 30 and 90, respectively. The upper diagrams depict the original ECA with complex patterns, while the lower ones show their fuzzified counterparts. The preceding results show that fuzzifying an ECA via FDNF yields only trivial or monotonic dynamical systems. Consequently, a different approach is required to construct a meaningful FCA model that exhibits complex behavior.

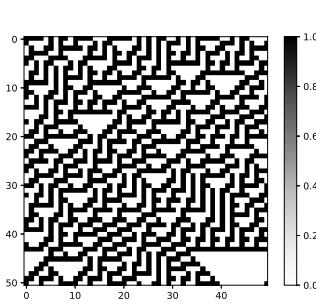


Fig. 3 Rule 30 ECA and corresponding FECA.

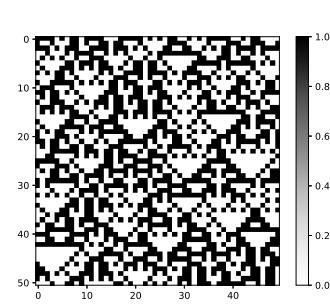


Fig. 4 Rule 90 ECA and corresponding FECA

3 Fuzzification function and generalized fuzzy elementary cellular automata

Let us remind that a Boolean FCA constructed from a CA should satisfy the two conditions: (i) the value of its cell is in $[0, 1]$, and (ii) time evolution rule coincides with that of the CA when its initial values are in $\{0, 1\}$. Hence we define

Definition 1 (GFECA)

A generalized fuzzy elementary cellular automaton (GFECA) is defined as a system satisfying the following two conditions:

- (1) It is a one-dimensional discrete dynamical system with three neighbors, where the state values are real numbers in $[0, 1]$.
- (2) The update function $f_k : [0, 1]^3 \rightarrow [0, 1]$ matches the ECA rule F_k at the endpoints.

Condition (2) of Definition 1 means that

$$(x, y, z) \in \{0, 1\}^3 \Rightarrow f_k(x, y, z) = F_k(x, y, z).$$

To generalize the update functions f_k from those constructed via FDNF, we define a set of *fuzzification functions* as follows.

Definition 2 (Set Ω)

$$\Omega := \{u \mid u : [0, 1] \rightarrow [0, 1], u(0) = 0, u(1) = 1\}$$

An important property of the fuzzification functions is found from the following proposition, which follows directly from the definitions of f_k and Ω .

Proposition 1 Let $g, u, v, w \in \Omega$. Then,

$$\tilde{f}_k(x, y, z) := g(f_k(u(x), v(y), w(z))) \quad (5)$$

defines the update rule

$$u_n^{t+1} = \tilde{f}_k(u_{n-1}^t, u_n^t, u_{n+1}^t) \quad (6)$$

which gives a GFECA corresponding to rule k .

Several examples of fuzzification functions are shown in Figs. 5 (a)-(f). If we use

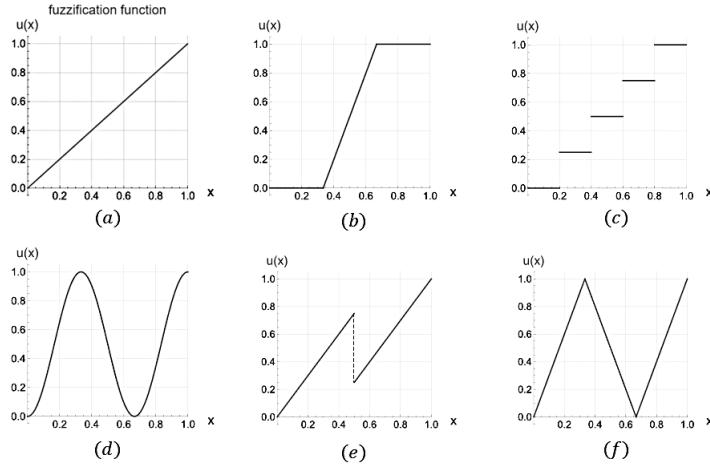


Fig. 5 Examples of fuzzification functions.

the function shown in Fig. 5 (a) for the fuzzification functions u, v, w, g in Eq. (6), then $\tilde{f}_k = f_k$ and the original FECA via FDNF is recovered. On the other hand, if we use the function shown in Fig. 5 (b), the update function \tilde{f}_k tends to take values 0 or 1 and its time evolution pattern is similar to that of the CA. If we use the other fuzzification functions shown in Fig. 5 (c)-(f), it is difficult to predict what kinds of patterns will emerge. In the next section, we will examine pattern formation using several types of fuzzification functions.

4 Pattern formation by GFECAs

As Proposition 1 indicates, the choice of fuzzification functions yields infinitely many types of GFeca. This section presents some of the representative evolution patterns that emerge.

1. FECA converging to ECA behavior

Let g be the identity function, and define $u(x) = v(x) = w(x)$ as:

$$u(x) = \begin{cases} 0 & (0 \leq x \leq \frac{1}{3}) \\ 3x - 1 & (\frac{1}{3} < x \leq \frac{2}{3}) \\ 1 & (\frac{2}{3} < x \leq 1) \end{cases} \quad (7)$$

This fuzzification function is illustrated in Fig. 5 (b).

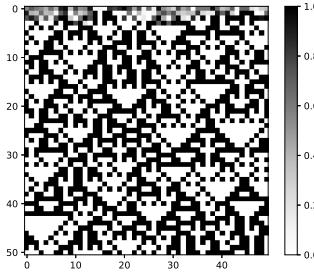


Fig. 6 90FECA using Eq. (7)

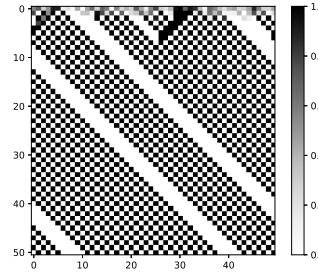


Fig. 7 184FECA using Eq. (7)

Using Eq. (7), we obtain rule 90 and 184 GFECAs. Figures 6 and 7 show time evolution patterns of the rule 90 and 184 GFECAs respectively thus obtained. We used random initial values in the interval $[0, 1]$ with periodic boundary conditions. Although initial values vary from 0 and 1, they eventually converge to binary values over time, reproducing the original ECA behavior. In the extreme case where $u(x)$ is defined as a step function:

$$u(x) = \begin{cases} 0 & (0 \leq x \leq \frac{1}{2}) \\ 1 & (\frac{1}{2} < x \leq 1) \end{cases}$$

the update function \tilde{f}_k takes only 0 or 1, causing the time evolution pattern for $t \geq 1$ to coincide with that of the corresponding ECA for any initial condition.

2. Functions with a gap at $\frac{1}{2}$

We define a class of GFECA by setting the functions g, u, v , and w identically to $g(x)$:

$$g(x) = \begin{cases} ax & (0 \leq x \leq \frac{1}{2}) \\ ax - a + 1 & (\frac{1}{2} < x \leq 1) \end{cases}, \quad (8)$$

where a is a parameter in the range $1 \leq a \leq 2$. As illustrated in Fig. 5(e), this function introduces a discontinuity gap of size $a-1$ at $x = \frac{1}{2}$. For the specific case where $a = 1$, the gap vanishes, and the GFECA becomes equivalent to the FECA via FDNF.

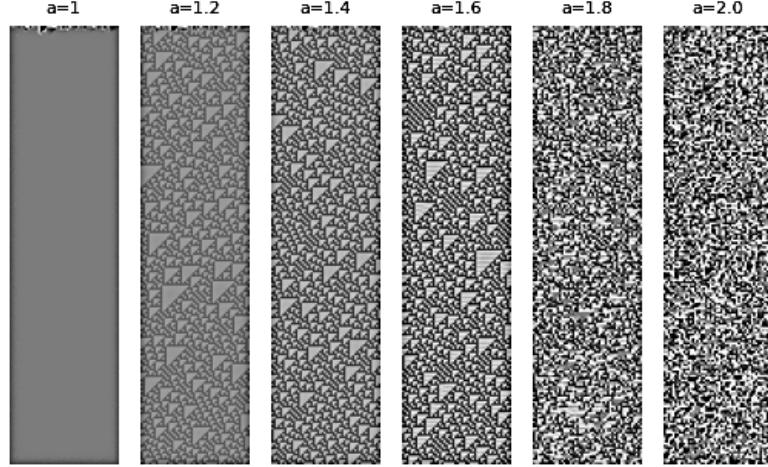


Fig. 8 Time evolution patterns of the rule 102 GFECA with fuzzification function Eq. (8). The parameter $a = 1.0, 1.2, 1.4, 1.6, 1.8, 2.0$.

Figure 8 shows how the time-evolution patterns of the rule 102 GFECA change as a function of the parameter a . The simulations use a system of 50 cells over a time interval of $0 \leq t \leq 200$, with the parameter a set to 1.0, 1.2, 1.4, 1.6, 1.8, and 2.0. For $a = 1.0$, the system evolves into a uniform state as expected. As a increases, however, characteristic patterns emerge. In the range $1.1 \leq a \leq 1.6$, the patterns are notably similar to those of the standard rule 102 ECA (Fig. 9). For $a \geq 1.6$, this distinct structure disappears, giving way to a random-like pattern. We observe similar behavior in case of the rule 195 GFECA.

Figure 10 shows the rule 184 GFECA obtained by the same fuzzification functions and parameters. For $a = 1.0$, the system evolves into a uniform state as expected too. As a increases, a labyrinthine pattern emerges around $1.1 \leq a \leq 1.3$, and changes to a random-like pattern for $1.4 \leq a$. The similar behavior is observed in the rule 226 GFECA.

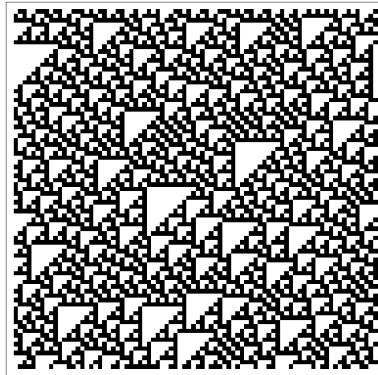


Fig. 9 A time evolution pattern of the rule 102 ECA

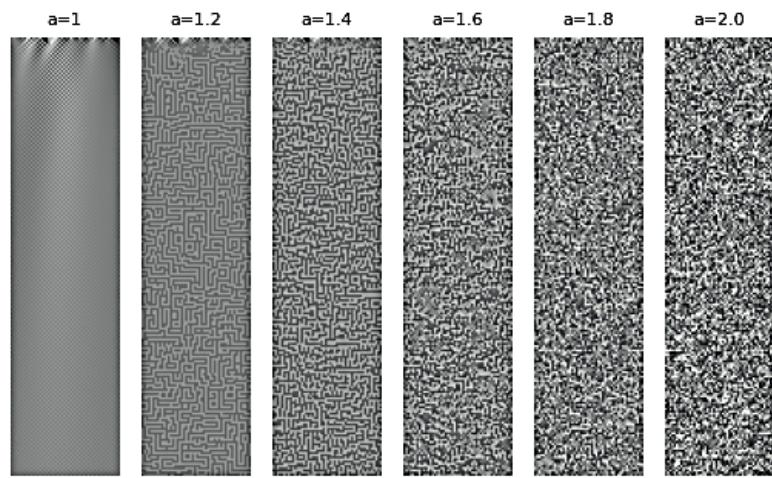


Fig. 10 Time evolution patterns of the rule 184 GFECA. Fuzzification functions and parameters are the same as those in Fig. 8

5 FCAs by mixing of ECA rules

By omitting condition (2) from Definition 1, we can define a more general class of one-dimensional, three-neighbor FCAs. A key method for constructing such automata involves creating a convex combination of the evolution rules from different GFECAAs. Specifically, given a set of update functions \tilde{f}_{k_j} ($j = 1, 2, \dots, l$) of GFECAAs, we can define a new, composite function \tilde{f} as:

$$\tilde{f}(x, y, z) := \sum_{j=1}^l \alpha_j \tilde{f}_{k_j}(x, y, z)$$

where $\forall j$, $\alpha_j \geq 0$, and $\sum_{j=1}^l \alpha_j = 1$. This construction guarantees that the resulting function \tilde{f} satisfies the condition (1) of the definition 1. It is important to note that this method constitutes a global mixing of rules. This is distinct from local mixing schemes, where different update rules might be applied at various points in space and time.

Let us consider a simple case where the fuzzy update function is constructed from a convex combination of two standard ECA update functions, f_{k_1} and f_{k_2} . For a mixing parameter $\alpha \in [0, 1]$, this new function is defined as:

$$\tilde{f}_{k_1 k_2}^{(\alpha)}(x, y, z) := \alpha f_{k_1}(u(x), u(y), u(z)) + (1 - \alpha) f_{k_2}(u(x), u(y), u(z)). \quad (9)$$

Here, the function $u(x)$, as defined in Eq. (7), maps the fuzzy cell states to crisp values suitable for the ECA rules.

Since the ECA functions f_{k_1} and f_{k_2} have codomains of $\{0, 1\}$, it follows that the new function is correctly bounded, i.e., $0 \leq \tilde{f}_{k_1 k_2}^{(\alpha)} \leq 1$. The parameter α smoothly interpolates between the two base rules. At the extreme values of α , the mixed rule collapses to one of the original ECA rules:

$$\tilde{f}_{k_1 k_2}^{(0)}(x, y, z) = f_{k_2}(u(x), u(y), u(z)) \quad \text{and} \quad \tilde{f}_{k_1 k_2}^{(1)}(x, y, z) = f_{k_1}(u(x), u(y), u(z)).$$

This allows us to observe how the dynamics of the system change as the rules are mixed.

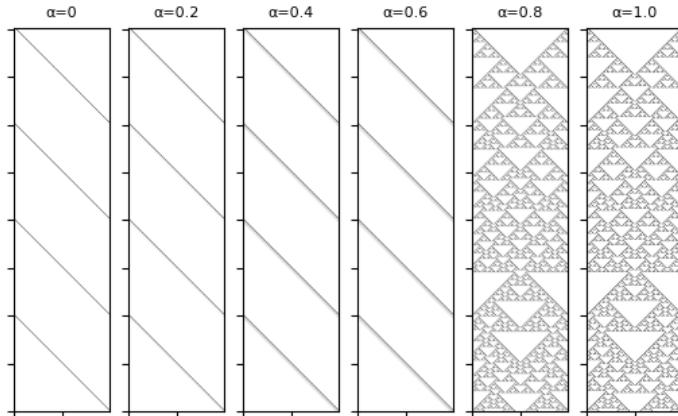


Fig. 11 Time evolution patterns by mixing the rule 184 FECA and 90 FECA using Eq. (9)

Figure 11 illustrates the dynamics that emerge from mixing the ECA rules $k_1 = 184$ and $k_2 = 90$. The simulation was performed on a lattice of 50 cells for 200 time steps (i.e., $0 \leq t \leq 200$). The initial condition was set by assigning the leftmost cell a value of 0.6 while all other cells were set to 0. The panels in the figure, from left to right, show the time evolution patterns corresponding to values of the mixing parameter $\alpha \in \{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$.

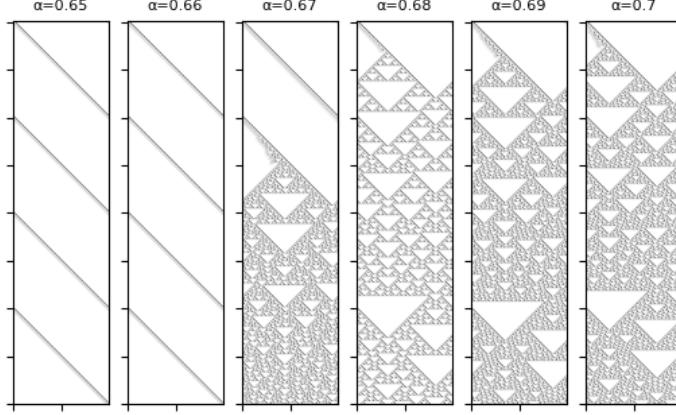


Fig. 12 Pattern transitions from $\alpha = 0.65$ to 0.70 in Fig. 11.

A magnified view, shown in Figure 12, focuses on the behavior around $\alpha \approx 0.66$, revealing a sharp, phase-transition-like shift in the dynamics. Similar critical transitions were also observed in other rule combinations, typically near $\alpha \approx 0.33$ and $\alpha \approx 0.66$.

To quantify the complexity of the evolving patterns, we introduce an order parameter d_α . This parameter is calculated for an analysis domain D containing M cells, where the state of a cell at position (i, j) is given by ϕ_{ij} .

First, we compute the mean square of the cell states, C_α , to serve as a normalization constant:

$$C_\alpha := \frac{1}{M} \sum_{(i,j) \in D} \phi_{ij}^2 \quad (10)$$

Next, using C_α , we define a normalized moment of order $2p$, a quantity related to ℓ^p -norm, for any real number $p \geq 1$:

$$N_\alpha(p) := \frac{1}{M} \sum_{(i,j) \in D} \left(\frac{\phi_{ij}}{\sqrt{C_\alpha}} \right)^{2p} \quad (11)$$

Finally, the *effective dimension* d_α is defined in the limit as $p \rightarrow \infty$:

$$d_\alpha := \lim_{p \rightarrow \infty} \left(2 - \frac{2}{p-1} \log_M N_\alpha(p) \right) \quad (12)$$

The resulting value, d_α , is a single scalar that characterizes the spatial complexity of the observed patterns, as illustrated in Fig. 13. To build intuition for this measure, we will now examine its behavior in three idealized cases.

1. Point-like (0-d) pattern

Consider a case where a single cell has a value $a > 0$ and all other $M - 1$ cells are

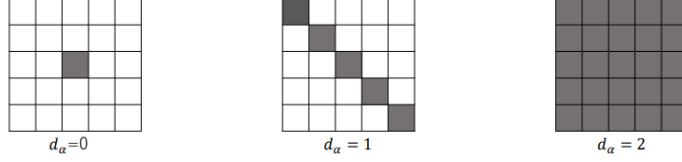


Fig. 13 When a pattern takes values mainly in a d dimensional region, effective dimension $d_\alpha \approx d$.

zero. The normalization constant and the moment are calculated as:

$$C_\alpha = \frac{1}{M}(a^2) = \frac{a^2}{M}$$

$$N_\alpha(p) = \frac{1}{M} \left(\frac{a}{\sqrt{C_\alpha}} \right)^{2p} = \frac{1}{M} \left(\frac{a}{a/\sqrt{M}} \right)^{2p} = \frac{1}{M} (\sqrt{M})^{2p} = M^{p-1}$$

Substituting this into the definition for the effective dimension yields:

$$d_\alpha = \lim_{p \rightarrow \infty} \left(2 - \frac{2}{p-1} \log_M(M^{p-1}) \right) = 2 - \frac{2(p-1)}{p-1} = 0$$

2. Line-like (1-d) pattern

For a one-dimensional structure (e.g., a line) in the 2D domain, the number of active cells is approximately \sqrt{M} . Assuming \sqrt{M} cells have the value a and the rest are zero:

$$C_\alpha = \frac{1}{M}(\sqrt{M} \cdot a^2) = \frac{a^2}{\sqrt{M}}$$

$$N_\alpha(p) = \frac{1}{M} \sum_{i=1}^{\sqrt{M}} \left(\frac{a}{\sqrt{C_\alpha}} \right)^{2p} = \frac{\sqrt{M}}{M} \left(\frac{a}{a/\sqrt{M}} \right)^{2p} = M^{-1/2} (M^{1/4})^{2p} = M^{\frac{p-1}{2}}$$

This configuration correctly results in a dimension of 1:

$$d_\alpha = \lim_{p \rightarrow \infty} \left(2 - \frac{2}{p-1} \log_M(M^{\frac{p-1}{2}}) \right) = 2 - \frac{2}{p-1} \cdot \frac{p-1}{2} = 1$$

3. Space-filling (2-d) pattern

Finally, if all M cells have a uniform value a , then $C_\alpha = a^2$ and $N_\alpha(p) = 1$. This gives $\log_M N_\alpha(p) = 0$, which yields $d_\alpha = 2$.

These examples demonstrate that d_α provides a quantitative index that corresponds to the intuitive spatial dimension of a pattern.

Let us use the above effective dimension d_α to investigate the transition in the time-evolution patterns of the GFECA 90 and GFECA 184 mixing rules as a function of the parameter α . Figure 14 plots d_α calculated over the time interval $450 \leq t \leq 500$. For $0 \leq \alpha \leq 0.6$, $d_\alpha \approx 1.0$, while for $\alpha \geq 0.7$, $d_\alpha \approx 1.6$. Between these regions, d_α increases in a step-like manner, though with significant fluctuations. A smaller, secondary jump is also observed around $\alpha \approx 1/3$.

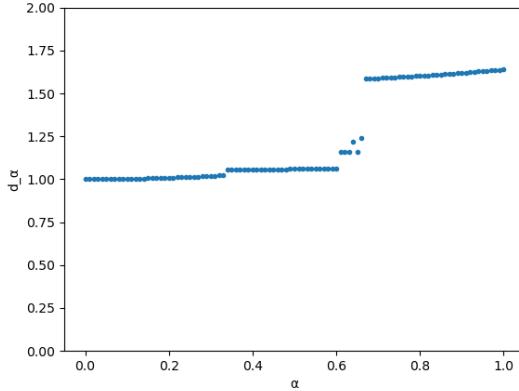


Fig. 14 Change of effective dimension d_α for the time evolution patterns of the mixing rule 184 GFECA and 90 GFECA.

This behavior is likely caused by the kinks in the fuzzification function $u(x)$ at $x = 1/3$ and $x = 2/3$. These features in $u(x)$ appear to induce the corresponding changes observed in d_α around $\alpha \approx 1/3$ and $\alpha \approx 2/3$.

Therefore, the effective dimension d_α serves as a useful order parameter for quantitatively describing these pattern changes. Furthermore, we note that the function $N_\alpha(p)$, or its logarithm $\log N_\alpha(p)$, contains more detailed information and could be used to investigate subtler pattern features.

6 GFECA dynamics with small degrees of freedom

Cellular automata with a small number of cells typically do not exhibit complex patterns. For example, an ECA with only three cells exhibits periodic time evolution over its 8 possible states, with the period's maximum length of 8 depending on the boundary conditions. However, a GFECA can exhibit complex time evolution if its fuzzification functions are chosen properly. As an example, consider a GFECA with three cells where the update function Eq. (6) uses the identity function for $g(x)$, and the functions $u(x)$, $v(x)$, $w(x)$ are identical and given by

$$u(x) = \begin{cases} 3x & (0 \leq x \leq \frac{1}{3}) \\ -3x + 2 & (\frac{1}{3} < x \leq \frac{2}{3}) \\ 3x - 2 & (\frac{2}{3} < x \leq 1) \end{cases} \quad (13)$$

The fuzzification function Eq. (13) is shown in Fig. 5 (f). Let x_t , y_t and z_t be the values of the three cells at time t . Then, the dynamics of rule k GFECA are governed

by the following system of equations:

$$\begin{cases} x_{t+1} = \tilde{f}_k(z_t, x_t, y_t) \\ y_{t+1} = \tilde{f}_k(x_t, y_t, z_t) \quad (t = 0, 1, 2, \dots) \\ z_{t+1} = \tilde{f}_k(y_t, z_t, x_t) \end{cases} \quad (14)$$

Starting from a generic initial state (x_0, y_0, z_0) , the orbit generated by this system evolves within the three-dimensional unit cube, $[0, 1]^3$. The characteristics of this orbit are determined by the specific rule k . In particular, its asymptotic points often form a stable manifold, providing a basis for classifying different GFECAs.

For any given update function $\tilde{f}_k(x, y, z)$, permuting its inputs x, y, z can generate up to six distinct variations. For instance, a new rule k' can be defined from rule k as follows:

$$f_{k'}(x, y, z) := f_k(y, z, x)$$

When all such variations are treated as symmetrically equivalent, the 256 ECA rules can be classified into 80 distinct groups. Examples of these groups include the single-member group $\{1\}$, the three-member group $\{2, 4, 16\}$, and the six-member group $\{138, 196, 176, 140, 162, 208\}$.

This equivalence is based on the functional form. The other rules in a group are generated by permuting the inputs of an update function. For example, consider the update function for rule 2:

$$f_2(x, y, z) = (1 - x)(1 - y)z$$

The other functions in its group, for rules 4 and 16, can be derived by swapping the inputs of f_2 :

$$\begin{aligned} f_4(x, y, z) &= x(1 - y)(1 - z) \quad (\text{by swapping } x \leftrightarrow z \text{ in } f_2) \\ f_{16}(x, y, z) &= (1 - x)y(1 - z) \quad (\text{by swapping } y \leftrightarrow z \text{ in } f_2) \end{aligned}$$

Similarly, permuting the inputs of $f_{138}(x, y, z) = xyz + (1 - x)z$ generates the other rules in its group, such as f_{196} and f_{176} .

Here are some typical examples of asymptotic behavior and stable manifolds.

A class of GFECAs has orbits that converge to a single fixed point. Some rules converge to the same point regardless of the initial state. For example, orbits under rule 8 always converge to $(0, 0, 0)$, and those under rule 238 always converge to $(1, 1, 1)$. In other cases, the fixed-point attractor depends on the initial conditions. For instance, rule 150 converges to a fixed point, but which point it is depends on the starting values. This behavior is also found in rules 60, 90, and 233.

A common behavior is the convergence of an orbit to a periodic cycle. For example, consider the rule 3 GFECAs, which has the update function:

$$f_3(x, y, z) = (1 - x)(1 - y)$$

As shown in Fig. 15, the orbits of the GFECA in the cube $[0, 1]^3$ converges to a period-two orbit consisting of the points $(0, 0, 0)$ and $(1, 1, 1)$. The figure's left panel plots the points from $t = 0$ to 1000, and the right panel plots the final stable orbit from $t = 9000$ to 10000. Similar behavior is observed in GFECA with rule 1, 5, 7, and 17.

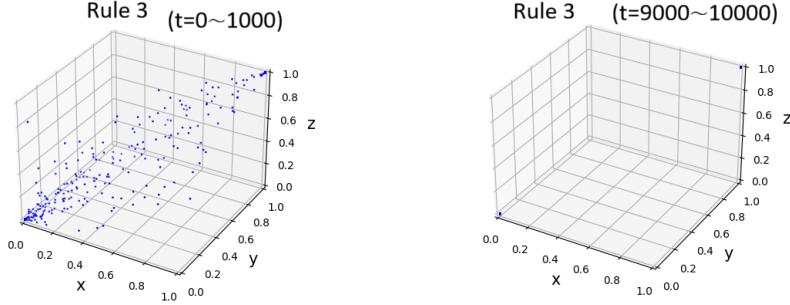


Fig. 15 Orbit of rule 3 GFECA with three cells. It converges to a period-two orbit.

Other rules converge to longer cycles. For instance, rule 210 converges to a period-six orbit (Fig. 16). Similar periodic convergence is observed in the GFECAs with rule 154, 156, and 198.

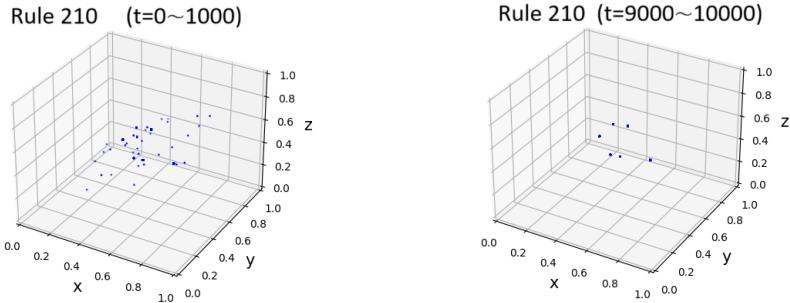


Fig. 16 Orbit of rule 210 GFECA with three cells. It converges to a period-six orbit.

Certain classes of GFECAs possess higher-dimensional stable manifolds. Figure 17 illustrates this by plotting orbits from $t = 9000$ to 10000, each starting from a generic initial state. In panels (a)–(c), the attractors asymptotically converge to one-dimensional line segments. In contrast, panels (d)–(f) display the characteristic, complex structures of higher-dimensional attractors. The system can also exhibit chaotic orbits, as demonstrated in Fig. 18. This behavior is readily explained by the update function for rule 85. Since $f_{85}(x, y, z) = 1 - y$, the fuzzified update function is:

$$\tilde{f}_{85}(x, y, z) = 1 - u(y),$$

where $u(y)$ is the inherently chaotic fuzzification function from Eq. (13). Consequently, the GFECA under rule 85 inherits this chaotic dynamics directly.

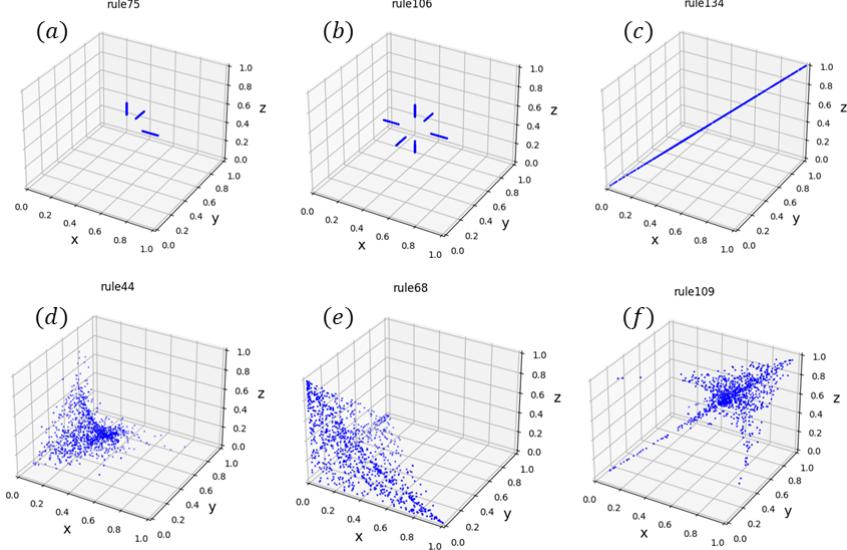


Fig. 17 Attractors of the GFECA with three cells. The orbits from $t = 9000$ to $t = 10000$ are plotted. The rules are (a) 75, (b) 106, (c) 134, (d) 44, (e) 68 and (f) 109.

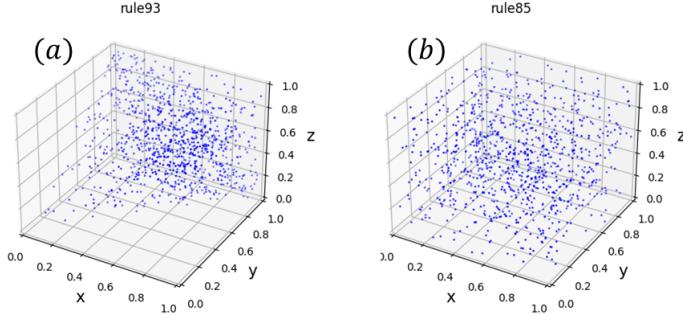


Fig. 18 Chaotic orbits of the GFECA with three cells. The orbits from $t = 9000$ to $t = 10000$ are plotted. The rules are (a) 93 and (b) 85.

This study focused on a three-cell GFECA—the minimal configuration for a three-neighbor FCA—using only the fuzzification function in Eq. (13). Increasing the number of cells would create dynamical systems with more degrees of freedom, likely leading to more complex evolutionary patterns and asymptotic behaviors. Furthermore, exploring a wider variety of fuzzification functions is a promising avenue for discovering richer and more diverse dynamical systems within the GFECA framework.

7 Concluding Remarks

This paper introduces a general method for constructing a FCA from any given Boolean CA. Since a multi-state CA can be replaced by equivalent Boolean CAs (for example, see [17]), the method is also applicable to multi-state CAs. Unlike the heuristic FDNF method, our approach not only retains the characteristic patterns of the original CA but also generates more complex patterns by using diverse fuzzification functions. We applied this method to ECAs, creating GFECAs. We demonstrated that by selecting the fuzzification function, these GFECAs exhibit controlled pattern transformations based on the function's parameters. Furthermore, our method facilitates the mixing of ECA rules, which leads to observable phase transitions as the mixing ratio is altered. These transitions were quantitatively confirmed using the effective dimension related to ℓ^p -norm. Notably, the resulting GFECAs display non-trivial temporal evolution even in minimal systems of just three cells. In essence, we have developed a general fuzzification method that preserves the complex behaviors of CAs, is mathematically tractable, and is expected to expand their utility as a modeling tool for natural and social phenomena.

Future research will focus on integrating our FCA framework with machine learning to explore new applications. This integration will enable the systematic and automatic discovery and calibration of fuzzy transition rules, for instance, by embedding FCAs within neural networks to learn transition functions or by informing deep learning architectures. By establishing an effective method for modeling localized interactions under uncertainty, this work could be applied to diverse fields such as materials science, biology, and sociology. Addressing these challenges and exploring these applications will be the primary focus of our future work.

Acknowledgements. This work was supported by Arithmer inc. and JSPS KAKENHI Grant Number JP23K22408.

References

- [1] Neumann, J.: Theory of Self-Reproducing Automata. University of Illinois Press, Urbana, IL, USA (1966)
- [2] Wolfram, S.: A New Kind of Science. Wolfram Media, Inc., Champaign, IL (2002)
- [3] Wolfram, S.: Theory and Applications of Cellular Automata. World Scientific Press, Singapore, Singapore (1986)
- [4] Nagel, K., Schreckenberg, M.: A cellular automaton model for freeway traffic. *Journal de Physique I* **2**(12), 2221–2229 (1992) <https://doi.org/10.1051/jp1:1992277>
- [5] Cattaneo, G., Finelli, M., Margara, L.: Cellular automata in fuzzy backgrounds. *Physica D: Nonlinear Phenomena* **105**(1-2), 105–120 (1997)

- [6] Betel, H., Flocchini, P.: On the relationship between boolean and fuzzy cellular automata. *Electronic Notes in Theoretical Computer Science* **252**, 5–21 (2009)
- [7] Flocchini, P., Geurts, F., Mingarelli, A., Santoro, N.: Convergence and aperiodicity in fuzzy cellular automata: revisiting rule 90. *Physica D: Nonlinear Phenomena* **142**(1-2), 20–28 (2000)
- [8] Mingarelli, A.B.: Fuzzy rule 110 dynamics and the golden number. *WSEAS Transactions on Computers* **2**(4), 1102–1107 (2003)
- [9] Mingarelli, A.B.: A classification scheme for fuzzy cellular automata with applications to ECA. *Journal of Cellular Automata* **5**(4-5), 445–467 (2010)
- [10] Yamamoto, K., Takahashi, D.: Asymptotic solutions to a fuzzy elementary cellular automaton of rule number 38. *JSIAM Letters* **15**, 93–96 (2023) <https://doi.org/10.14495/journal.15.93>
- [11] Nayak, D.R., Patra, P.K., Mahapatra, A.: A survey on two dimensional cellular automata and its application in image processing. *International Journal of Computer Applications* **98**(19), 32–36 (2014). Published in the proceedings of the International Conference on Emergent Trends in Computing and Communication (ETCC-2014)
- [12] Sadjadi, E.N., Zadeh, D.S., Moshiri, B., Herrero, J.G., López, J.M.M., Fernández, R.: Application of Smooth Fuzzy Model in Image Denoising and Edge Detection. *Mathematics* **10**(14), 2421 (2022) <https://doi.org/10.3390/math10142421>
- [13] Tangsakul, S., Wongthanavasu, S.: Deep Cellular Automata-Based Feature Extraction for Classification of the Breast Cancer Image. *Applied Sciences* **13**(10), 6081 (2023) <https://doi.org/10.3390/app13106081>
- [14] Foroutan, E., Moud, M.-A., Dastani, Y.: Urban growth modeling based on cellular automata with transition rules optimized using genetic fuzzy systems. *Transactions in GIS* **26**(7), 3253–3276 (2022) <https://doi.org/10.1111/tgis.12992>
- [15] Higashi, K., Satsuma, J., Tokihiro, T.: Rule 184 fuzzy cellular automaton as a mathematical model for traffic flow. *Japan Journal of Industrial and Applied Mathematics* **38**(2), 579–609 (2021)
- [16] Nishida, Y., Watanabe, S., Fukuda, A., Yanagisawa, D.: Fuzzy cellular automata with complete number-conserving rule as traffic-flow models with bottleneck. *JSIAM Letters* **14**, 143–146 (2022) <https://doi.org/10.14495/journal.14.143>
- [17] Nishida, Y., Watanabe, S., Fukuda, A., Watanabe, Y.: q-vfca: q-state vector-valued fuzzy cellular automata. *Journal of Cellular Automata* **15**(3), 207–222 (2020)