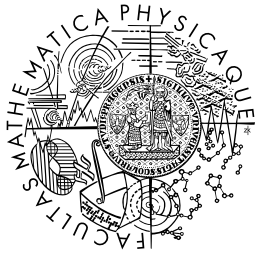


FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Barbora Hudcová

Complexity in Cellular Automata

Algebra Department

Supervisor of the master thesis: Ing. Tomáš Mikolov, Ph.D.

Study programme: Mathematics

Study branch: Mathematics for Information Technologies

Prague 2020

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ....

Author's signature

I would like to thank my supervisor, Ing. Tomáš Mikolov, Ph.D., for giving me a unique opportunity to develop a better idea of what the life of a researcher is like and for always making sure I don't lose sight of the big picture. I am very grateful to Doc. RNDr. Jiří Tůma, DrSc. for all his time and work he invested in me. Last but not least, I would like to thank prof. RNDr. Jaromír Antoch, CSc., Prof. RNDr. Petr Kůrka, CSc., and Mgr. Ondřej Týbl for their valuable advice on various topics. Thanks to my friends and family for being patient with me and supporting me.

Title: Complexity in Cellular Automata

Author: Barbora Hudcová

Department: Algebra Department

Supervisor: Ing. Tomáš Mikolov, Ph.D., department

Abstract: In order to identify complex systems capable of modeling artificial life, we study the notion of complexity within a class of dynamical systems called cellular automata. We present a novel classification of cellular automata dynamics, which helps us identify interesting behavior in large automaton spaces. We give a detailed comparison of our results to previous methods of dynamics classification. In the second part of the thesis, we study the backward dynamics of cellular automata. We present a novel representation of one-dimensional cellular automata, which can be used to charcterize all their garden of eden configurations. We demonstrate the usefulness of this method on examples.

Keywords: cellular automata, dynamical systems, classification of cellular automata, complex systems, garden of eden

Contents

Introduction

In the past decades, the interest in studying neural networks has been increasing rapidly. Along with this hype, they are often marketed as "the" artificial intelligence. However, they lack some fundamental properties to accomplish that. For example, it is challenging to implement a long-term memory in the neural network environment. A more fundamental problem is their inability to recognize similar tasks to those they have been trained for and adjust to perform well in them, [14]. This suggests that there is a need for a different paradigm that would bring us closer to the ambitious goal of designing real artificial general intelligence.

There is one field in particular that seems to promise an interesting and novel approach to searching for artificial general intelligence: the field of artificial life. It unifies researchers with different backgrounds such as mathematics, physics, computer science, or biology, most of whom wish to understand evolutionary mechanisms through the simulation of computer programs. The goal is to search for models that resemble some form of artificial evolution. Within their simulation, there would be complex structures spontaneously emerging and further growing in complexity, interacting with one another, and potentially exhibiting some form of intelligence. An iconic example is Tierra [20], a model where simple computer programs compete for memory and CPU time and exhibit interesting forms of parasitism and symbiosis. More recent attempts to model artificial life are for example [18] or [23]. As interesting as such approaches are, they are not rooted in formal theory. Possibly, such experiments might direct us towards the correct definitions of life, complexity, and emergence which are crucial and which we lack.

The field of complex systems offers more formal means of studying artificial life. Within it, scientists build upon the mathematical foundations of dynamical systems theory. The intriguing property of such systems is that even very simple rules can yield interesting behavior.

A reoccurring concept, as exhibited by the simple example of the logistic map, is that increasing a single parameter of a rule specifying the system's dynamics can change the dynamics from trivial and easily predictable to chaotic; more details can be found in [25]. Somewhere "between" these two modes of operation, we observe specific parameter values that cause the so-called complex behavior. Even though dynamical systems theory gives us a formal definition of chaos, as of yet, we have no conditions which would characterize complex behavior.

At least, to develop better intuition, we informally describe some properties shared by complex systems. Among them are the following:

- The system consists of many particles that are interacting locally but are eventually capable of some global synchronization.
- The system's underlying rule is nonlinear.
- When asked how the system's simulation would look like at time t, there are no significantly more efficient means to get an answer other than to simulate the system itself for t time steps.

Classic examples of complex systems found in nature include ant colonies, stock price fluctuations, or the neuron interactions in the human brain. In such

systems, we often see some "higher-order organisms forming" — organisms consisting of multiple basic building blocks of the system which nevertheless seem to act as individuals, an example being a flock of birds. The ultimate goal in the field of complex systems is to give a suitable formal definition of complexity or some sufficient conditions for it, to develop methods of automatically finding complex systems in large dynamical systems spaces, and to subsequently use those as models of artificial life.

In this thesis, we study an elementary class of dynamical systems, which are called cellular automata. They are probably the most classic example of systems capable of complex behavior. We examine their properties to develop a better understanding of the notion of complexity and propose a novel experimental method of classifying their dynamics.

In the first part of the thesis, we study forward dynamics of cellular automata. We present a novel method of classifying cellular automata dynamics; Section Transient Classification. We compare our results to previous classification schemes and demonstrate how our method can be used to automatically search for automata with interesting behavior. In the second part of the thesis, we present a novel representation of cellular automata related to their backward dynamics; Section Cayley Graphs of Cellular Automata. We use this representation to deduce certain phase-space properties of a chaotic automaton operating on a finite cyclic grid; Section Phase-Space Properties of Rule 45. Moreover, we present a characterization of all garden of eden configurations of a famous Turing complete automaton operating on an infinite grid; Theorem 38, Theorem 39. To the best of our knowledge, both results concerning the particular automata are new.

1. Introducing Cellular Automata

Before we define cellular automata, we present some basic notions and notation that we use throughout the thesis.

1.1 Preliminaries

We understand the set of natural numbers to be the set $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$. The set $\mathbb{N} \setminus 0$ is denoted by $\mathbb{N}^+$. $\mathbb{Z}$ denotes the set of integers.

Words
Let A be a finite non-empty set and $n \in \mathbb{N}$. A *word* w of length n over A is a sequence $w = w_0 w_1 \ldots w_{n-1}$ where each $w_i \in A$. We also define the *empty word* ϵ to be the empty sequence. A^* denotes the set of all words over A, including the empty word, and A^+ denotes the set $A^* \setminus \epsilon$. Any set $L \subseteq A^*$ is called a *language over* A. The length of w is denoted by $|w|$ and the set of all words of length n over A by A^n. The length of ϵ is 0.

An *infinite word* u over A is a mapping from $\mathbb{Z}$ to A, we write

$$u = \ldots u_{-2}u_{-1}u_0u_1u_2\ldots, \; u_i \in A \text{ for all } i \in \mathbb{Z}.$$

The set of all infinite words is denoted by $A^{\mathbb{Z}}$. A *left infinite word* l over A is a mapping from $\{\ldots, -3, -2, -1\}$ to A and we write $l = \ldots l_{-3}l_{-2}l_{-1}$. Analogously, a *right infinite word* r over A is a mapping from $\mathbb{N}$ to A and we write $r = r_0r_1r_2r_3\ldots$.

Let $u, v \in A^*$, $u = u_0u_1\ldots u_{n-1}$, $v = v_0v_1\ldots v_{m-1}$ for some $m, n \in \mathbb{N}^+$. We define the concatenation of u and v to be the word $u_0u_1\ldots u_{n-1}v_0v_1\ldots v_{m-1}$, which we denote by uv. The empty word is a neutral element with respect to concatenation. Let l be a left infinite word over A and r be a right infinite word over A. We can analogously define concatenations lr, lu, and ur.

With respect to the above notation, we say that $v \in A^+$ is a *subword* or a *block of* u if there exist $p, s \in A^*$ such that $u = pvs$. In such case, we write $v \sqsubseteq u$. If $|v| < |u|$, we say v is a *proper block* of u. If $p = \epsilon$, we say v is a *prefix* of u. And at last, if $s = \epsilon$, then we say that v is a suffix of u. We say that u is a block of an infinite word w over A if there exists a left and a right infinite word l, r over A such that $w = lur$.

Let $w \in A^+$ and $n \in \mathbb{N}$. For convenience, we write $(w)^n$ or w^n instead of $\underbrace{ww\ldots w}_{\text{n times}}$. By w^0 we understand the empty word. Let $p, s \in A^*$. The expression $p(w)^*s$ denotes the set $\{pvs \,|\, v \in \{w\}^*\}$. For $i, j \in \{0, 1, \ldots, |w| - 1\}$, $i < j$, we write $w_{[i,j]} = w_iw_{i+1}\ldots w_j$. An analogous notation is used for infinite words.

Functions
We denote $\mathbf{2} = \{0, 1\}$. A *Boolean function* is any function $f : \mathbf{2}^n \to \mathbf{2}$ for some $n \in \mathbb{N}^+$. We say a Boolean function is *balanced* if

$$|\{u \in \mathbf{2}^n \,|\, f(u) = 0\}| = |\{v \in \mathbf{2}^n \,|\, f(v) = 1\}| = 2^{n-1}.$$

Finite Automata
A *deterministic finite automaton* (DFA) is a 5-tuple $\mathcal{A} = (Q, A, \delta, q_0, F)$ where:

- Q is a finite set of *states*
- A is a finite set of *input symbols*
- $\delta : Q \times A \to Q$ is a *transition function*
- $q_0 \in Q$ is an *initial state*
- $F \subseteq Q$ is a set of *accept states*.

With respect to the above notation, we say that the automaton $\mathcal{A}$ *accepts a word* $w \in A^+$, $w = w_0 \dots w_{n-1}$, if there exists a sequence of states of length $n+1$, $(q_0, q_1, \dots, q_n) \subseteq Q^{n+1}$ such that $\delta(q_i, w_i) = q_{i+1}$ for all $i \in \{0, 1, \dots, n-1\}$ and $q_n \in F$. We say that ϵ is accepted by $\mathcal{A}$ if $q_0 \in F$.

Let A be a finite set. We call the language $L \subseteq A^*$ a *regular language* if there exists a deterministic finite automaton $\mathcal{A} = (Q, A, \delta, q_0, F)$ such that the set of words accepted by $\mathcal{A}$ is precisely L.

1.2 Introducing Cellular Automata

Informally, a cellular automaton (CA) can be perceived as a k-dimensional grid consisting of identical finite state automata. They are all updated synchronously in discrete time steps based on an identical transition function depending only on the states of automata in their local neighborhood. A more formal definition presented below is based on a great introduction to cellular automata by Jarkko Kari [9].

Let $d \in \mathbb{N}$. We define a *d-dimensional cellular grid* as $\mathbb{Z}^d$ and we call its elements *cells*. Let S be a finite set of *states*. A *configuration of the cellular grid* is a mapping

$$c : \mathbb{Z}^d \to S.$$

We denote the set of all d-dimensional configurations by $S^{\mathbb{Z}^d}$.

Let $k \in \mathbb{N}^+$. We define a *d-dimensional neighborhood of size k* to be a sequence $N = (\vec{n}_1, \vec{n}_2, \dots, \vec{n}_k)$ where each $\vec{n}_i \in \mathbb{Z}^d$ and $\vec{n}_i \neq \vec{n}_j$ for all $i \neq j$. Given such N, we can compute the *relative neighborhood of a cell* $\vec{z} \in \mathbb{Z}^d$ as $(\vec{z} + \vec{n}_1, \vec{z} + \vec{n}_2, \dots, \vec{z} + \vec{n}_k)$.

We define the *local update rule* to be a function that specifies the "next" state of a cell based on the "old" states of the cell's relative neighborhood. Formally, given a d-dimensional neighborhood of size k and a finite set of states S, it is a function

$$f : S^k \longrightarrow S.$$

A cellular automaton is given by a global update function that operates on the configurations. It updates the states of all cells of the configuration in parallel by "considering the states in every cell's relative neighborhood" and using the local update rule to obtain a "new" state of the cell.

Definition 1. *A cellular automaton operating on an infinite grid is a mapping given by:*

1. *a dimension* $d \in \mathbb{N}^+$ *of the cellular grid*
2. *a neighborhood* $N = (\vec{n}_1, \vec{n}_2, \ldots, \vec{n}_k) \subseteq (\mathbb{Z}^d)^k$ *of size* k *for some* $k \in \mathbb{N}^+$
3. *a finite set of states* S
4. *a local update rule* $f : S^k \to S$.

We denote such a cellular automaton by the tuple $(S^{\mathbb{Z}^d}, F)$ *where* F *is the global update function defined as:*

$$F : S^{\mathbb{Z}^d} \to S^{\mathbb{Z}^d}$$
$$c(\vec{z}) \mapsto f(c(\vec{z}+\vec{n}_1), c(\vec{z}+\vec{n}_2), \ldots, c(\vec{z}+\vec{n}_k)) \text{ for all } z \in \mathbb{Z}^d.$$

The most typical study cases are one, and two-dimensional CAs as their configurations can be easily visualized.

1.2.1 One-dimensional Cellular Automata

In the case of 1D CAs, we will simplify the notation. Let $N = (n_1, \ldots, n_k) \subseteq \mathbb{Z}^k$ be a neighborhood of some CA. If we put $r = \max\{|n_i| \,|\, i \in \{1, 2, \ldots, k\}\}$, we can notice that N can be embedded into a larger *symmetric neighborhood*, which is of the form $(-r, -r+1, \ldots, r-1, r)$. In such case, we call r the *radius* of the symmetric neighborhood.

For any 1D CA with neighborhood $N = (n_1, \ldots, n_k)$ and local rule $f : S^k \to S$, we can consider an analogous automaton with $N' = (-r, \ldots, r)$, $r = \max\{|n_i| \,|\, i \in \{1, 2, \ldots, k\}\}$, and a local function $f' : S^{2r+1} \to S$ defined as $f' = f \circ (\pi_{n_1}, \ldots, \pi_{n_k})$ where $\pi_i : S^{2r+1} \to S$ is the canonical projection to the i-th coordinate. Therefore, without loss of generality, we can assume that any 1D CA has such a symmetric neighborhood.

Suppose we have a local rule of a 1D CA $f : S^{2r+1} \to S$. We can extend f to a mapping $\widetilde{f} : S^* \dashrightarrow S^*$ defined as:

$$\widetilde{f}(u_0 u_1 \ldots u_n) = f(u_0, u_1, \ldots, u_{2r+1}) f(u_1, u_2, \ldots, u_{2r+2}) \ldots f(u_{n-2r}, \ldots, u_n)$$

for any $u_0 \ldots u_n \in \bigcup_{k=2r+1}^{\infty} S^k$.

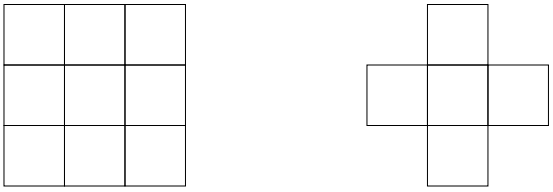
1.2.2 Two-dimensional Cellular Automata

When studying 2D CAs, there are two classic examples of neighborhoods that are considered. The Moore neighborhood is defined as

$$N_{\text{Moore}} = ((1,1), (1,0), (1,-1), (0,1), (0,0), (0,-1), (-1,1), (-1,0), (-1,-1)).$$

The simpler Von Neumann neighborhood is defined as

$$N_{\text{Von Neumann}} = ((1,0), (0,1), (0,0), (0,-1), (-1,0)).$$



Moore neighborhood Von Neumann neighborhood

Figure 1.1: Diagrams of two typical neighborhoods of two-dimensional cellular automata.

1.3 Cellular Automata on a Finite Grid

Given a cellular automaton $(S^{\mathbb{Z}^d}, F)$ and an initial configuration $u \in S^{\mathbb{Z}^d}$, we define the *trajectory of* u as:

$$(u, F(u), F^2(u), F^3(u), \ldots).$$

The process of obtaining a trajectory is often called a *simulation* or an *evolution* of the CA. Studying the *dynamics of a cellular automaton* simply means the studying of any property related to its global function, which is iterated on different initial conditions.

So far, we have considered CAs operating on infinite grids. Such automata are usually studied in fields such as symbolic dynamics for their interesting theoretical properties. However, when visualizing the cellular configurations, or when viewing CAs as a non-classical model of computation, we have to consider some form of a finite-sized cellular grid. Below, we present one possible method of reduction to a finite grid specifically for 1D and 2D CAs. The generalization of the method to higher dimensions is straightforward.

1.3.1 1D CAs Operating on a Finite Grid

In the case of 1D CAs, we consider the cellular grid to be $\mathbb{Z}_n$ and we simply compute the cells in the relative neighborhood modulo n. More formally, a *one-dimensional cellular automaton operating on a finite cyclic grid of size* $n \in \mathbb{N}^+$ is a tuple (S^n, F) where F is given by a local rule $f : S^{2r+1} \to S$ with a symmetric neighborhood with radius r and:

$$\begin{aligned} F : S^n &\to S^n \\ c_i &\mapsto f(c_{i-r \bmod n}, \ldots, c_{i+r \bmod n}) \end{aligned}$$

for all i $\in \mathbb{Z}_n$.

Let (S^n, F) be a one-dimensional CA operating on a cyclic grid and let $u \in S^n$ be an initial configuration. The trajectory $(u = F^0(u), F(u), F^2(u), \ldots)$ can be easily visualized in the following way. We define the *space-time diagram of the CA* (S^n, F) *simulated on* u *for* t *time steps* as a matrix $M \in S^{t \times n}$ where

$$(M)_{(i,j)} = F^i(u)_j, \qquad i \in \{0, 1, \ldots, t-1\},\ j \in \{0, 1, \ldots, n-1\}.$$

We write $F^0(u) \coloneqq u$.

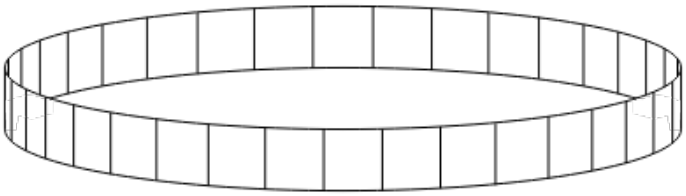


Figure 1.2: Schema of a one-dimensional cyclic cellular grid.

Figure 1.3: Space-time diagram of a one-dimensional CA operating on a cyclic grid of size 70, simulated for 50 time steps on a randomly chosen initial configuration.

Such space-time diagrams are typically visualized by identifying each state with a different color. In case when the set of states is the set $\mathbf{2} = \{0, 1\}$, the custom is to identify state 0 with white and state 1 with black color.

In Figure 1.3, we show the space-time diagram of an ECA with local rule $f : \mathbf{2}^3 \to \mathbf{2}$, $f(a, b, c) = (a + c) \bmod 2$.

1.3.2 2D CAs Operating on a Finite Grid

In the case of 2D CAs, we consider the cellular grid to be of the form $\mathbb{Z}_m \times \mathbb{Z}_n$ for some $m, n \in \mathbb{N}^+$ and analogously to the 1D case, we simply compute first coordinates of a cell's relative neighborhood modulo m and second coordinates modulo n. More formally, a *two-dimensional cellular automaton operating on a finite cyclic grid of size* $m \times n$ is a tuple $(S^{m \cdot n}, F)$ where F is given by a neighborhood $G = ((g_1^{(1)}, g_1^{(2)}), (g_2^{(1)}, g_2^{(2)}), \ldots, (g_k^{(1)}, g_k^{(2)}))$ and a local rule $f : S^k \to S$

and:

$$\begin{aligned} F:\ S^{m\cdot n} &\to S^{m\cdot n} \\ c(i,j) &\mapsto f(c(i+g_1^{(1)} \bmod m, j+g_1^{(2)} \bmod n), \\ &\qquad c(i+g_2^{(1)} \bmod m, j+g_2^{(2)} \bmod n), \\ &\qquad\qquad \vdots \\ &\qquad c(i+g_k^{(1)} \bmod m, j+g_k^{(2)} \bmod n)) \end{aligned}$$

for all $(i,j) \in \mathbb{Z}_m \times \mathbb{Z}_n$.

Loosely speaking, the topology of the two-dimensional finite grid is that of a torus.

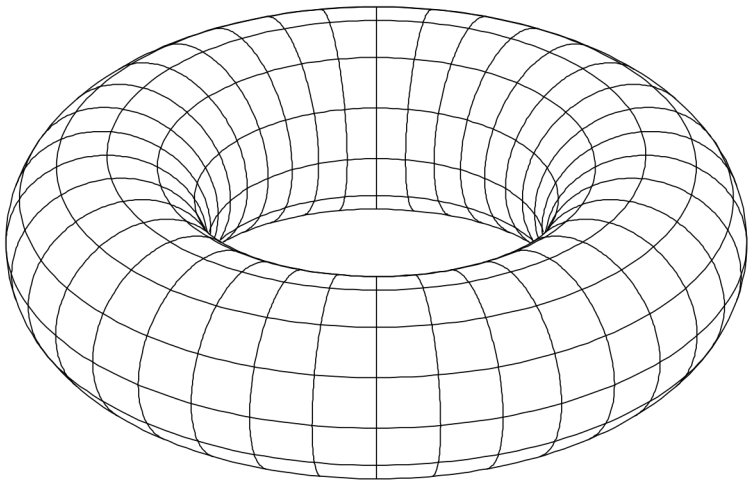


Figure 1.4: Schema of a two-dimensional cyclic cellular grid.

In the case of 2D CAs, their simulations are usually presented as animations where the frames represent consecutive configurations.

There are other "boundary conditions" that are being considered when studying the dynamics of cellular automata. Another frequent choice is to fix the "boundary cells" in a certain state. However, this choice seems more arbitrary. The advantage of computing the coordinates of cells in relative neighborhood modulo a set of fixed numbers is that all the cells have the same neighborhood.

1.4 Brief History of Cellular Automata

Cellular automata were first studied as models of self-replication. In the 1950s the great mathematician John Von Neumann was interested in the connections between biology and the new rising field of computational devices. His goal was to come up with a computational model, within which self-replication would be possible. After Stanislaw Ulam suggested the model of cellular automata for this purpose, Von Neumann was able to define a "universal constructor" within a 2D CA. That is a finite arrangement of cells, together with a possibly infinite tape, which is able to build any automaton whose description is encoded on the tape.

Therefore, the universal constructor can also build "itself". The crucial mechanism of self-replication is to use the information encoded on the tape twice. Firstly, to interpret it and to build the structure and secondly, to copy the information on the tape of the new structure. This was an intriguing idea as Von Neumann came up with this concept before the structure of DNA was discovered, and therefore, before the mechanisms of DNA's self-replication were known to humankind. This exciting part of history is described by the Nobel laureate Sydney Brenner [1].

In the cellular environment, it is easy to come up with trivial structures (such as a structure consisting of one cell only), which are able to self replicate. This is the reason why Von Neumann aimed to design a "universal constructor" — a structure being undoubtedly non-trivial as it has the same computational capabilities as a universal Turing machine. After Von Neuman's work was completed and published by Arthur Burks [17], it spurred many other variations on the design of self-replicating cellular structures. Many scientists argued that there is a large gap between "trivial" and "computationally universal" structures and that organisms in nature might not necessarily be "Turing complete". Therefore, many other much simpler structures were designed, a great overview was written by Reggia et al. [21], possibly the most famous of them being the loops designed by Christopher Langton [11].

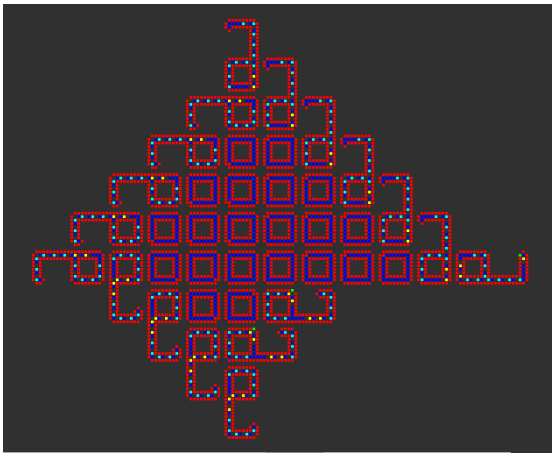


Figure 1.5: Illustration of a "colony" of Langton loops - simple structures in a 2D CA system, which are capable of self-replication.

Since the 1950s cellular automata have been studied for their various properties — as computational systems [15], as possible models for artificial life [12], as topological mappings [10], but mainly as dynamical systems capable of complex behavior [26], [28]. In the next chapter, we give a more detailed overview of the work related to classifying dynamics of CAs.

2. Classifying Cellular Automata

A crucial result helping us understand the notion of complexity would be a suitable classification of different CA dynamics. It could also navigate us toward a region of CAs with complex behavior. An ideal classification would be based on a rigorously defined and easily measurable property.

In this section, we describe three qualitatively different methods of CA classification to develop a better intuition for distinguishing different CA dynamics. Subsequently, we present a novel method, which we call the Transient classification, and compare our results to previous methods on a small class of CAs. We then show how the Transient classification can be used to search for complex automata in larger CA spaces.

2.1 CA Dynamics via Space-Time Diagrams

First, we introduce a small class of one-dimensional CAs often used to demonstrate the results of different classification methods.

Definition 2. *We define an elementary cellular automaton (ECA) to be a one-dimensional cellular automaton with radius* $r = 1$ *and the set of states* $\mathbf{2} = \{0, 1\}$.

Each ECA is given by a Boolean function $f : \mathbf{2}^3 \to \mathbf{2}$. Hence, there are only 256 of them. The size of the ECA class is the main reason it is frequently used when examining different properties of CAs. Given a property we want to measure, it is feasible to obtain results for each ECA and present an exhaustive comparison. As there are $k^{k^{2r+1}}$ one-dimensional CAs with k states and radius r, the size of such class grows steeply when either k, or r is increased. For instance, as soon as we increase the number of states to 3 and keep $r = 1$, we obtain $3^{3^3} \approx 10^{13}$ different cellular automata. For such vast classes, an exhaustive comparison is not feasible.

We have a natural bijection between the set of ternary Boolean functions $\{f : \mathbf{2}^3 \to \mathbf{2}\}$ and integers in the set $\{0, 1, \ldots, 255\}$ given simply by:

$$f \mapsto 2^0 f(0,0,0) + 2^1 f(0,0,1) + 2^2 f(0,1,0) + \ldots + 2^6 f(1,1,0) + 2^7 f(1,1,1).$$

We call such number the *Wolfram number of* f, after the notation introduced by Stephen Wolfram in [28], which is now widely used. Further, respecting his terminology, we will refer to each ECA as *rule* k where k is the corresponding Wolfram number of the ECA's local rule.

2.1.1 Wolfram's Classification

When observing the space-time diagrams of CAs, we notice that even simple automata can produce intriguing patterns. A very extensive study of cellular automata based on their visualizations was done by Stephen Wolfram in [28].

Therein, he established an intuitive, yet informal classification of CA dynamics based on the space-time diagrams. He distinguishes the following classes of behavior.

Class 1 ... space-time diagrams quickly resolve to a simple configuration repeated at every time step

Class 2 ... after a short time, there are only a few configurations repeating periodically in the diagram

Class 3 ... the diagrams look chaotic and no obvious pattern is formed

Class 4 ... the diagrams produce some localized structures, which interact with each other in complicated ways

The best way to understand the informal description is to observe examples given in Figure 2.1.

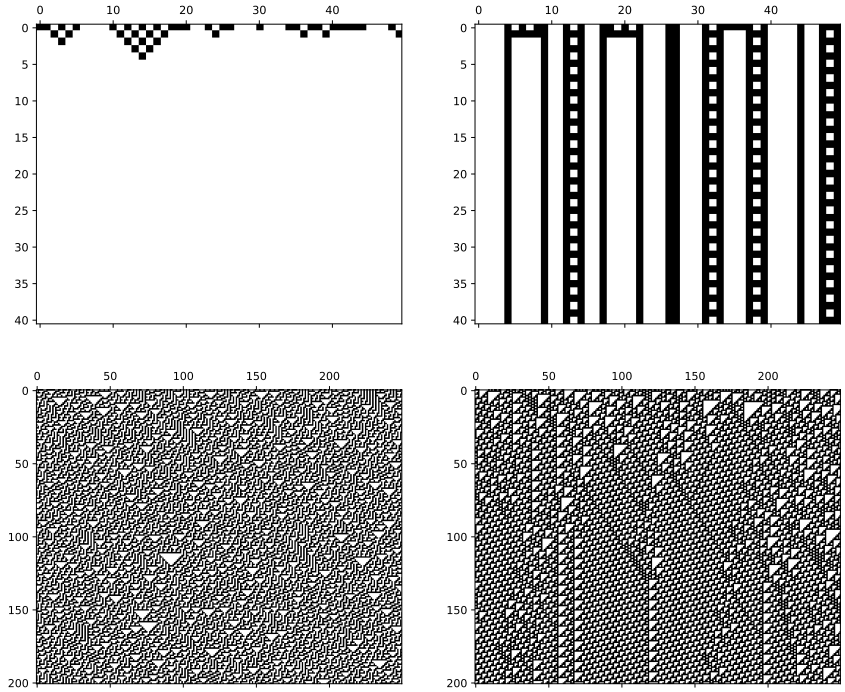


Figure 2.1: Space-time diagrams of rules from each Wolfram's class. Class 1 rule 32 is on the top left, Class 2 rule 108 on the top right. Both are simulated for 40 time steps on a grid of size 50. At the bottom row, we have Class 3 rule 30 on the left and Class 4 rule 110 on the right. The two are simulated for 200 steps on a grid of size 250.

Classes 1 and 2 are considered to have simple dynamics, whereas Classes 3 and 4 seem to have more intriguing properties. Class 3 is usually referred to as chaotic and Class 4 as complex. There are, however, only four ECA rules that are classified as complex. According to Wolfram, they are rules 110, 124, 137, and 193. They all have isomorphic phase-spaces. We will talk about the isomorphism of phase-spaces in more detail in Section 2.2.1. The argument, which supports Wolfram's claim that such rules are complex, is that they have been proven to be Turing complete [3]. The iconic examples of chaotic rules are rules 30 and 45.

Though informal, Wolfram's classification is widely referred to in the literature. The main issue is that we have no formal method to classify CAs in this way. Moreover, the behavior of some CAs can vary with different initial configurations. We have examples of this scenario even in the simple ECA class where e.g. rule 126 exhibits class 2 behavior when simulated from a particular set of initial conditions, and otherwise shows Class 3 behavior. This is illustrated in Figure 2.2. In fact, there is no exhaustive list of ECAs and their corresponding classes to be found. Wolfram rather presents only examples of rules from each class. The Transient classification we present in this chapter deals with such issues.

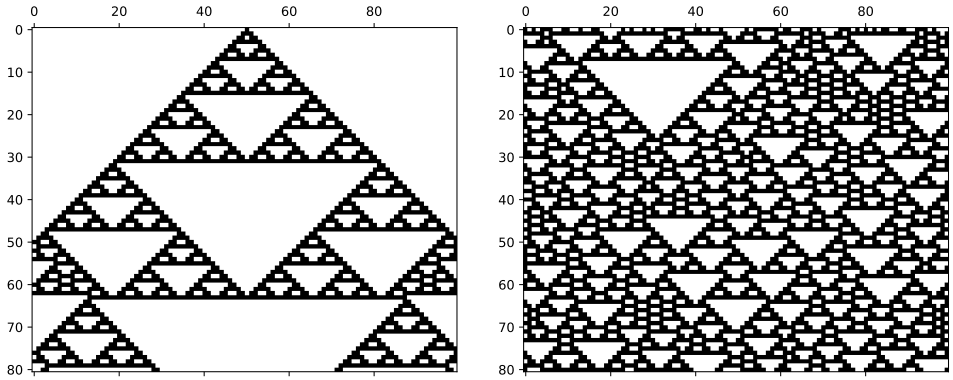


Figure 2.2: On the left, rule 126 is simulated on an initial condition consisting of a single 1 bit padded with 0's. On the right, the same rule is simulated with a random initial configuration.

2.1.2 Zenil's Classification

In [31], Hector Zenil presents an intuitive approach aiming to formalize Wolfram's observations of space-time diagrams. Let $(\mathbf{2}^n, F)$ be an ECA, $u \in \mathbf{2}^n$ an initial configuration, and $T \in \mathbb{N}^+$ a time threshold. Zenil generates the space-time diagram of u simulated for T time-steps and uses a compression algorithm in order to measure the size of the compressed diagram. Intuitively, Class 1 and 2 automata produce repetitive patterns in their space-time diagrams, and therefore, the compressed size of such diagrams is relatively small. For chaotic automata, the compressed size of their space-time diagrams should be large.

Concretely, Zenil puts $T = 200$, $n = 401$, and chooses the initial configuration u to have $u_{201} = 1$ and $u_i = 0$ for all the other positions. He chooses the compression algorithm to be Mathematica's function *Compress* from the zlib package. After obtaining the data for every ECA, Zenil uses a clustering technique to obtain two classes. In Figure 2.3, we show our reproduction of Zenil's results using Python.

When comparing this classification to Wolfram's, we get that the dominating "yellow" cluster corresponds to classes 1 and 2. The "purple" cluster with larger compression size roughly corresponds to Wolfram's classes 3 and 4.

We conducted multiple experiments presented in Figure 2.4, which show that Zenil's results are very sensitive to the choice of many parameters. We dare

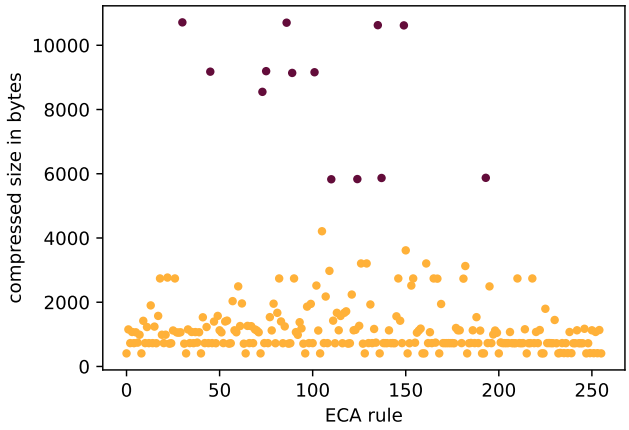


Figure 2.3: Reproduction of Zenil's results described in [31]. The two colors distinguish two CA classes obtained by a clustering technique KMeans in Python. We note that the sizes of the compressed diagrams do not match Zenil's results exactly. This is because we have used a slightly different compression algorithm - Python's gzip. However, the results are qualitatively the same, and the clusters we have obtained are identical to Zenil's.

say that Zenil chose the parameter values precisely to obtain a good match to Wolfram's classification. In vast CA spaces, where it is not feasible to examine every CA and mark it into one of Wolfram's classes by hand, it would not be clear how the parameter values should be chosen.

The method depends on the time threshold T, the choice of the initial condition, the compression algorithm, as well as the data representation. To illustrate the sensitivity of the results, we show two graphs in Figure 2.4 with different choices of the parameters.

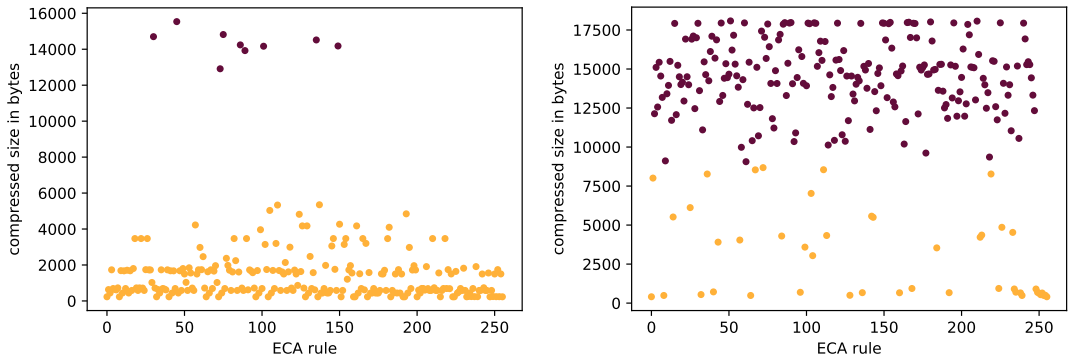


Figure 2.4: Graphs representing the results of Zenil's method when different parameter values were used. They demonstrate how sensitive the results are. On the left, the ECAs were simulated for 400 rather than 200 time steps. This causes rules 110, 124, 137, and 193 to no longer belong to the "purple" cluster. Therefore, such rules would not be marked as "interesting" by this method. On the right, the ECAs were simulated from a fixed, randomly chosen initial condition. In such case, we obtain entirely different clusters.

2.2 Global Dynamics of Cellular Automata

One of the issues of Wolfram's and Zenil's classifications is that they examine *local dynamics* of CAs - the classifications are dependent on the choice of an initial configuration. As there are different means of choosing the initial configuration, this adds arbitrariness to the method. In this section, we introduce the properties of cellular automata related to their *global* dynamics.

Let us consider a d-dimensional finite cyclic cellular grid $G = \mathbb{Z}_{m_1} \times \ldots \times \mathbb{Z}_{m_d}$, where each $m_i \in \mathbb{N}^+$. Let (S^G, F) be a CA operating on such a grid, let $u \in S^G$ and let $(F^0(u) = u, F(u), F^2(u), F^3(u), \ldots)$ be the trajectory of u.

As the set of all grid configurations S^G is finite, the trajectory eventually becomes periodic. More precisely, there exist $i, j \in \mathbb{N}$, $i < j$, such that $F^i(u) = F^j(u)$. Let $i \in \mathbb{N}$ be the smallest positive integer, for which there exist $j \in \mathbb{N}$, $j > i$, such that $F^i(u) = F^j(u)$. We call the sequence

$$(F^0(u) = u, F(u), F^2(u), \ldots, F^i(u))$$

the *transient of* u. We denote its length by t_u, i.e. $t_u = i$. Let $j \in \mathbb{N}$ be smallest such that $F^i(u) = F^j(u)$. We call the sequence

$$(F^i(u), F^{i+1}(u), F^{i+2}(u), \ldots, F^{j-1}(u))$$

the *attractor of* u and denote its length as a_u, $a_u = j - i$. We define the *rho value of* u as $\rho_u = t_u + a_u$.

The *phase-space* of the CA (S^G, F) is an oriented graph with vertices $V = S^G$ and edges $E = \{(u, F(u)), u \in S^G\}$. Such a graph is composed of components, each containing one attractor and multiple transients "leading" to the attractor. The phase-space completely characterizes the dynamics of the system. It is, however, infeasible to describe it for large n.

2.2.1 Equivalent Cellular Automata

Clearly, any two CAs with isomorphic phase-spaces have identical dynamics. We can use this fact to reduce the size of ECAs we examine. This will save us space and time when giving an exhaustive comparison of the results of the Transient classification for every ECA.

Let $\mathcal{A}$ denote the set of all local rules of elementary cellular automata. We have $|\mathcal{A}| = 256$. We define mappings $\pi : \mathcal{A} \to \mathcal{A}$ and $\sigma : \mathcal{A} \to \mathcal{A}$ as follows:

$$\pi(f)(a, b, c) = f(c, a, b) \quad \forall a, b, c \in \{0, 1\}$$
$$\sigma(f)(a, b, c) = 1 - f(1 - a, 1 - b, 1 - c) \quad \forall a, b, c \in \{0, 1\}.$$

Therefore, rule $\pi(f)$ is obtained from f by switching the role of the "left and right neighbor". Similarly, rule $\sigma(f)$ is obtained from f by switching the role of states 0 and 1.

We define a relation $\sim \subseteq \mathcal{A} \times \mathcal{A}$ so that $f \sim g$ if at least one of the following conditions holds:

$$f = g, \qquad f = \pi(g), \qquad f = \sigma(g), \qquad f = \sigma(\pi(g)).$$

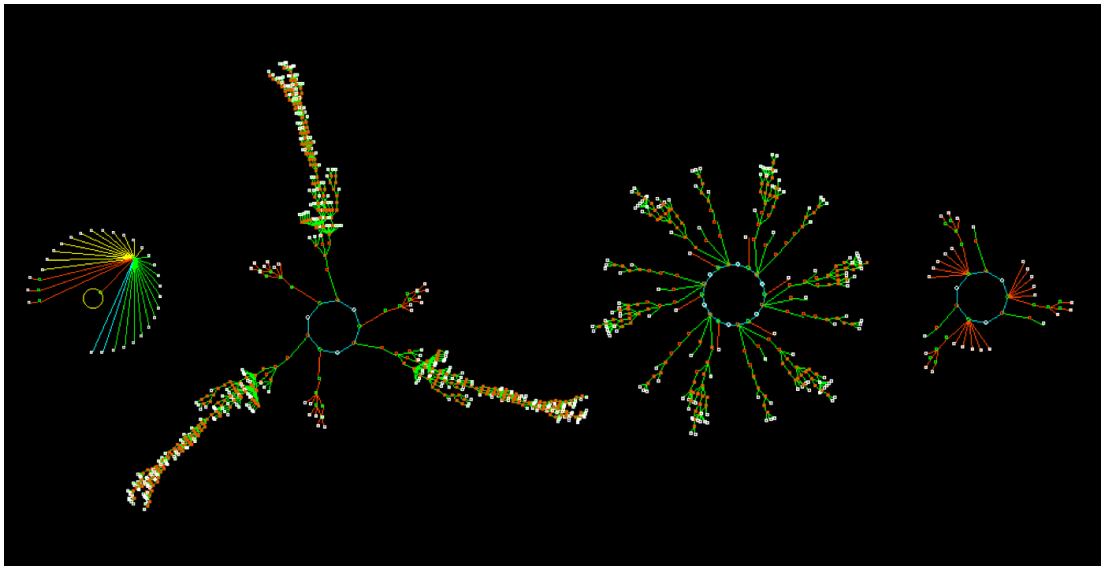


Figure 2.5: An intriguing phase-space visualization of ECA rule 110 operating on a cyclic grid of size 12. Only a part of the phase-space is shown. The direction of the edges is not depicted. Naturally, all edges point towards the "center cycle" in each component. The edges within each cycle are directed "clockwise". The software used to produce such illustrations is described in [29].

It can be easily verified that $\sim$ is an equivalence relation and that there are 88 equivalence classes of ECAs. Let $n \in \mathbb{N}^+$. We further define two mappings:

$$\Pi_n : \mathbf{2}^n \to \mathbf{2}^n, \quad \Pi(u)_i = u_{n-i}$$
$$\Sigma_n : \mathbf{2}^n \to \mathbf{2}^n, \quad \Sigma(u)_i = 1 - u_i.$$

Clearly, both mappings Π_n, Σ_n are bijective for every $n \in \mathbb{N}^+$. Moreover, let $(\mathbf{2}^n, F)$ and $(\mathbf{2}^n, G)$ be two ECA with local functions f and g, for which it holds that $f = \pi(g)$. Then,

$$G(u) = \Pi_n^{-1} F(\Pi_n(u)),$$

which gives us that Π_n is an isomorphism of the phase-spaces of the ECAs. By analogous arguments for the case when $f = \sigma(g)$ and $f = \sigma(\pi(g))$, we obtain the following observation.

Observation 3. *Let $(\mathbf{2}^n, F)$ be an ECA with local rule f and $(\mathbf{2}^n, G)$ an ECA with local rule g. If $f \sim g$ then the two ECAs have isomorphic phase-spaces.*

From now on, we will only consider the 88 ECA classes with unique dynamics. We choose the rule with the smallest Wolfram number from each equivalence class to represent it.

2.2.2 Z-parameter

In this subsection, we introduce a different method of classifying CA dynamics. We describe the Z-parameter — a simple parameter related to CAs' backward dynamics introduced by Andrew Wuensche [30]. One of the phase-space properties Wuensche has examined is the *maximum preimaging* of CAs defined below.

Definition 4. *Let A be a finite set and $n \in \mathbb{N}$. Let (A^n, F) be a cellular automaton operating on a finite cyclic grid of size n. We define the maximum preimaging of such automaton to be the number*

$$mp_n = \max_{u \in A^n} \{|F^{-1}(u)|\}.$$

We say that (A^n, F) is reversible, if $mp_n = 1$. That is, if F is injective.

As the maximum preimaging is usually infeasible to compute for automata operating on large grids, Wuensche introduces the Z-parameter, which loosely relates to the maximum preimaging parameter, and is very simple to compute. The main purpose of this subsection is to describe the Z-parameter as a different approach to classifying cellular automata.

Definition 5. *Let $f : \mathbf{2}^3 \to \mathbf{2}$ be a local rule of some ECA.*

We say that $ab \in \mathbf{2}^2$ is left deterministic for f, if for every output $u \in \mathbf{2}$ there exists a unique $c \in \mathbf{2}$ such that $f(a, b, c) = u$.

We say that $a \in \mathbf{2}$ is left deterministic for f if it holds that for every $u \in \mathbf{2}$ there exists a unique $b \in \mathbf{2}$ such that $f(a, b, c) = u$ for every $c \in \mathbf{2}$.

At last, we say that ϵ is left deterministic for f if it holds that for every u there exists a unique $a \in \mathbf{2}$ such that $f(a, b, c) = u$ for every $b, c \in \mathbf{2}$.

Analogically, we define right deterministic words.

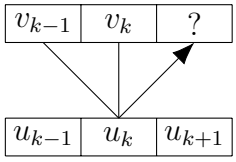
If f is a local rule of some ECA, for which $ab \in \mathbf{2}^2$ is left deterministic, then it must hold that $f(a, b, 0) = u$ and $f(a, b, 1) = 1 - u$. Therefore, in case every word of length two is left deterministic for f, we have that f is a balanced Boolean function.

Similarly, if $a \in \mathbf{2}$ is left deterministic for f, then $f(a, 0, c) = u$ and $f(a, 1, c) = 1 - u$ for every $c \in \mathbf{2}$. Hence, if both 0 and 1 are left deterministic for f, it also implies that f is balanced.

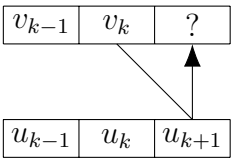
Definition 6. *Let (A^n, F), resp $(A^{\mathbb{Z}}, F)$, be a one-dimensional cellular automaton with local rule f. Let $u \in A^n$, resp $u \in A^{\mathbb{Z}}$. We say that $v \in A^*$ is a partial preimage of u if $\widetilde{f}(v) \sqsubseteq u$.*

Below, we illustrate the meaning of left deterministic words. Let f be a local rule of an ECA operating on an infinite grid. Suppose we have a configuration $u \in A^{\mathbb{Z}}$, for which we want to construct a preimage. Further, suppose we have already constructed a partial preimage $v = v_0 v_1 \dots v_k$. Suppose we would like to search for all partial images of u created by appending one letter to v, that is, obtain $v' = v_0 v_1 \dots v_k v_{k+1}$, so that v' is still a partial preimage of u. Then:

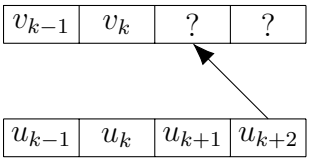
- In case $v_{k-1} v_k$ is left deterministic for f, v_{k+1} can be uniquely determined by v_{k-1}, v_k, and u_k.



- In case v_k is left deterministic for f, v_{k+1} can be uniquely determined by v_k and u_{k+1}.



- In case ϵ is left deterministic for f, v_{k+1} can be uniquely determined by u_{k+2}.



We can notice that if ϵ is left deterministic for $f : \mathbf{2^3} \to \mathbf{2}$, then there exists $u \in \mathbf{2}$ such that $f(0, b, c) = u$ and $f(1, b, c) = 1 - u$ for all $b, c \in \mathbf{2}$. Therefore, the values of f do not depend on the second and third input coordinates. This implies that there are no left deterministic words for f other than ϵ.

Moreover, in such a case, the corresponding ECA operating on a cyclic tape is reversible, as every configuration has exactly one preimage.

Definition 7. *Let $f : \mathbf{2}^3 \to \mathbf{2}$ be a local rule of some ECA. We define the probability "that each partial preimage can be prolonged uniquely to the right given two symbols" as the number*

$$p_2^l = \frac{1}{4} |\{ab \in \mathbf{2}^2, \ ab \text{ is left deterministic for } f\}|.$$

We define the probability "that each partial preimage can be prolonged uniquely to the right given one symbol" as the number

$$p_1^l = \frac{1}{2} |\{a \in \mathbf{2}, \ a \text{ is left deterministic for } f\}|.$$

Finally, we define

$$p_0^l = \begin{cases} 1 & \text{if } \epsilon \text{ is left deterministic for } f \\ 0 & \text{otherwise.} \end{cases}$$

We define $Z_l = p_2^l + (1 - p_2^l)p_1^l + p_0^l$. Analogously, we define Z_r. The Z-parameter for local rule f is defined as $Z = \max\{Z_l, Z_r\}$.

Clearly, $0 \leq p_i^l \leq 1$ for every $i \in \{0, 1, 2\}$. Moreover, if $p_0^l = 0$ then $0 \leq \mathrm{Z}_l = p_2^l + (1 - p_2^l)p_1^l \leq p_2^l + 1 - p_2^l = 1$. The other case when $p_0^l = 1$ implies that there are no non-empty left deterministic words for f and $p_1^l = 0$, $p_2^l = 0$. Therefore, $\mathrm{Z}_l = 1$. We get that $0 \leq \mathrm{Z}_l \leq 1$ and analogically, it holds that $0 \leq \mathrm{Z}_r \leq 1$ and therefore, we also get that $0 \leq Z \leq 1$.

Example. We will compute the Z-parameter of rule 110. The left and right Z-parameters can be easily computed from the table of the local rule.

u	111	110	101	100	011	010	001	000
$f(u)$	0	1	1	0	1	1	1	0

Computing Z_l:

$$p_2^l = \frac{1}{4}\,|\{11, 10, 00\}| = \frac{3}{4}$$
$$p_1^l = \frac{1}{2}\,|\emptyset| = 0$$
$$p_0^l = 0$$
$$\mathrm{Z}_l = \frac{3}{4} + 0 + 0 = 0.75$$

Computing Z_r:

$$p_2^r = \frac{1}{4}\,|\{11\}| = \frac{1}{4}$$
$$p_1^r = \frac{1}{2}\,|\{0\}| = \frac{1}{2}$$
$$p_1^r = 0$$
$$\mathrm{Z}_r = \frac{1}{4} + \frac{3}{4} \cdot \frac{1}{2} + 0 = 0.625$$

Hence, $Z = \max\{0.75, 0.625\} = 0.75$.

In fact, the Z-parameter is closely related to the algorithm Wuensche uses to compute CAs' phase-spaces. The algorithm first generates an attractor cycle by observing the trajectory of a chosen configuration. Then, it computes all preimages of all attractor configurations and iterates this process of preimage generation until the whole component is obtained. We describe a simplified version of the preimage generation process below.

The computation of a preimage of a configuration $u \in \mathbf{2}^n$, $u = u_0u_1 \dots u_{n-1}$, under a CA $(\mathbf{2}^n, F)$ with local rule f can be described by the following pseudocode.

1. initialize the queue of "possible preimages" by $\widetilde{f}^{-1}(u_0)$
2. repeat until the queue of possible preimages is empty:
   - take a partial preimage p from the queue of possible preimages
   - prolong p by all possible symbols $x \in \mathbf{2}$ so that px is still a partial preimage of u
   - if $|px| = n$, check whether $F(px) = u$, either discard px or save it as a valid preimage of u
   - if $|px| < n$, return px to the queue of partial preimages
3. return the set of valid preimages

Below, we describe the relationship between the maximum preimaging and Z-parameter. Lemma 8 describes the case when $Z = 1$.

Lemma 8. *Assume we have an ECA with local rule f such that $Z = 1$. Then, $mp_n \leq 4$ for any $n \in \mathbb{N}$.*

Proof. Without loss of generality we will suppose that $\mathrm{Z}_l = 1$.

In case when $p_0^l = 1$, we have that ϵ is left deterministic for f. In such case it is clear that every configuration has exactly one preimage and $mp_n = 1$ for every $n \in \mathbb{N}$.

In case when $p_0^l = 0$ we have that $1 = \mathrm{Z_L} = p_2^l + (1 - p_2^l)p_1^l$. This implies that either $p_1^l = 1$, or $p_2^l = 1$. That is, either every word of length 1 is left deterministic, or every word of length 2 is left deterministic. Both such scenarios imply that f is

balanced. Hence, $|\widetilde{f}^{-1}(a)| = 4$ for any $a \in \mathbf{2}$. Moreover, either case implies that every partial preimage of every configuration can be uniquely prolonged by one symbol on the right. Following the above algorithm for computing all possible preimages of a given configuration, we see that we obtain only 4 "candidate" preimages of a suitable length. After checking whether such candidates are indeed preimages of the given configuration, we get that any configuration has at most 4 preimages and $mp_n \leq 4$ for every $n \in \mathbb{N}$. □

Lemma 9 describes the case when $Z = 0$.

Lemma 9. *An ECA with local rule f has $Z = 0$ if and only of $mp_n = 2^n$ for every $n \in \mathbb{N}^+$.*

Proof. Let F_n denote the global function of the ECA operating on a cyclic grid of size n. Then $mp_n = 2^n$ for some $n \in \mathbb{N}^+$ if and only if F_n is constant. And further, F_n is constant for all $n \in \mathbb{N}^+$ if and only if f is constant.

It clearly holds that if f is constant there are no left or right deterministic words for f and $Z = 0$. It remains to prove the converse.

If $Z = 0$, there are no left or right deterministic words of either length 0, 1, or 2. The non-existence of left deterministic words of length two implies that $f(a, b, 0) = f(a, b, 1)$ for all $a, b \in \mathbf{2}$. The non-existence of the right ones implies that $f(0, b, c) = f(1, b, c)$ for all $b, c \in \mathbf{2}$. Therefore, $f(a, b, c)$ is only dependent on its second coordinate b, i. e. there exists a function $g : \mathbf{2} \to \mathbf{2}$ such that $f(a, b, c) = g(b)$ for all $a, b, c \in \mathbf{2}$. As there is no left deterministic word of length one for f, it holds that for all $a \in \mathbf{2}$ there exists $c \in \mathbf{2}$ such that $f(a, 0, c) = f(a, 1, c)$. Therefore it holds that g is constant, and consequently, that f is constant as well. □

In this sense, we have some correspondence between the Z-parameter and maximum preimaging. Intuitively, having $mp_n = 2^n$ for all n implies that all transients of all configurations have length at most 1, such ECAs belong to Wolfram's Class 1 and are rather uninteresting. On the other hand, ECAs with $Z = 1$ can possibly have long attractors or transients. In fact, some typical Wolfram Class 3 automata such as rule 30 or rule 45 have $Z = 1$. However, most ECAs have $0 < Z < 1$. From broader experiments, including the examination of 1D CAs with a neighborhood of size 5, Wuensche suggests that complex behavior typically occurs at $Z \approx 0.75$.

So far, we have given a non-exhaustive overview of different approaches of classifying CA dynamics. In the subsequent section, we present a novel classification method.

2.3 Transient Classification

2.3.1 Basic Concepts and Motivation

The key concept of our classification is that we consider a CA operating on a cyclic grid of increasing size and measure the asymptotic growth of its average transient length. In this subsection, we motivate this experiment.

For a given CA and a grid of size n, we randomly sample initial configurations $u \in \{0, 1\}^n$ and estimate the average transient length $\mu_n = \frac{1}{2^n} \sum_{u \in \{0,1\}^n} t_u$ by

a value $\widehat{\mu}_n$. With the help of computer simulations, we obtain a sequence of estimations $\widehat{\mu}_1, \widehat{\mu}_2, \ldots, \widehat{\mu}_k$ where k is as large as possible given computational resources that we have. Using linear regression, we find a function that best fits our data - we examine the fit to a constant, logarithmic, linear, polynomial, and exponential functions. The result could be interpreted as finding the fittest "asymptotic growth of the average transient length".

In non-classical models of computation [24] the process of traversing CA's transients can be perceived as the process of self-organization, in which information can be aggregated in an irreversible manner. The attractors are then viewed as memory storage units, from which the information about the output can be extracted. This is explored in [7]. The average transient growth then corresponds to the average computation time of the CA. CAs with bounded transient lengths can only perform trivial computation. On the other hand, CAs with exponential transient growth can be interpreted as inefficient computation models.

In the context of artificial evolution, we can view the local rule of a CA as the physical rule of the system, whereas the initial configuration as the particular "setting of the universe", which is then subject to evolution. If we are interested in finding CAs capable of complex behavior automatically, it would be beneficial for us if such behavior occurred on average, rather than having to select the initial configurations carefully from some narrow region. The probability of finding such special initial configurations would be extremely low as the overall number of configurations grows exponentially with increasing grid size. This motivates our study of the growth of average transient lengths rather than the maximum transient lengths.

We note that transients of CAs have been examined before as in [30], [22]. However, we are not aware of an attempt to compare the "asymptotic growth" of transients for different CAs. We present the classification method in more detail using the simple class of ECAs. Subsequently, we show how the method can be used in more general CA spaces.

2.3.2 Data Sampling and Regression Fits

Suppose we have an ECA operating on a large grid of size n. In such case, computing the average transient length μ_n is infeasible. Therefore, we randomly sample initial configurations $u_1, u_2, \ldots, u_m$ and estimate μ_n by $\frac{1}{m}\sum_{i=1}^{m} t_{u_i}$. It remains to estimate the number of samples m so that the error $|\frac{1}{m}\sum_{i=1}^{m} t_{u_i} - \mu_n|$ is reasonably small.

More formally, we fix $n \in \mathbb{N}$ and let (C_n, U_n) be a discrete probability space where $C_n = \{0,1\}^n$ is the set of all n-bit configurations and U_n is a uniform distribution. Let $X : C_n \to \mathbb{N}$ be a random variable, which sends each u to its transient length t_u. This gives rise to a probability distribution of transient lengths on $\mathbb{N}$ with mean $E(X)$ and variance $var(X)$. We can notice that

$$\begin{aligned} E(X) &= \sum_{i=0}^{\infty} U_n\{u \in C_u | X(u) = i\} \cdot i \\ &= \sum_{i=0}^{\infty} \sum_{u \in C_n,\, X(u)=i} \frac{1}{2^n} \cdot i \\ &= \sum_{u \in C_n} \frac{1}{2^n} X(u) \\ &= \mu_n. \end{aligned}$$

Our goal is to obtain a good estimate of $E(X)$ by the Monte Carlo method ([19]).

Let $(X_1, X_2, \ldots, X_m)$ be a random sample of independent, identically distributed random variables, $X_i \overset{d}{=} X$ for all i. Let $\mu_n^{(m)} = \frac{1}{m}\sum_{i=1}^{m} X_i$ be the sample mean and $\sigma_n^{(m)} = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}(X_i - \mu_n^{(m)})^2}$ the sample standard deviation. As $var(X) < \infty$, we have by the Central limit theorem that the distribution of $\sqrt{m}\mu_n^{(m)}$ converges to a normal distribution when $m \to \infty$. Moreover, the interval

$$\left(\mu_n^{(m)} - u_{1-\frac{\alpha}{2}}\frac{\sigma_n^{(m)}}{\sqrt{m}}, \mu_n^{(m)} + u_{1-\frac{\alpha}{2}}\frac{\sigma_n^{(m)}}{\sqrt{m}}\right)$$

where u_β is the β quantile of the normalized normal distribution, covers μ_n for m large with probability approximately $1-\alpha$. We will take $\alpha = 0.05$. Hence, with probability approximately 95%

$$|\mu_n - \mu_n^{(m)}| < u_{0.975}\frac{\sigma_n^{(m)}}{\sqrt{m}}.$$

From the nature of our data, both the values $E(x) = \mu_n$ and $var(X)$ tend to grow with increasing grid size. Therefore, to employ a general method of estimating the number of samples, we consider the relative error $\frac{|\mu_n - \mu_n^{(m)}|}{\mu_n}$. We obtain that with probability around 95%

$$\frac{|\mu_n - \mu_n^{(m)}|}{\mu_n} < u_{0.975}\frac{\sigma_n^{(m)}}{\sqrt{m}\mu_n}.$$

We approximate the value of μ_n by $\mu_n^{(m)}$ and finally obtain the following criterion. For m sufficiently large such that

$$u_{0.975}\frac{\sigma_n^{(m)}}{\sqrt{m}\mu_n^{(m)}} < \epsilon \tag{2.1}$$

we have that $\mu_n^{(m)}$ differs from μ_n by at most $\epsilon \cdot 100\%$ with probability approximately 95%.

In practice, we put $\epsilon = 0.1$ and produce the observations in batches of size 20 until condition (2.1) is met. For each ECA we obtain a dataset of the form $(\widehat{\mu}_n)_{n=n_{min}}^{n_{max}}$ where $\widehat{\mu}_n$ is the estimate of the average transient length on the grid of size n. In case of ECAs, we put $n_{min} = 4$ and keep generating data until a time bound is reached.

Subsequently, we perform regression fits to constant, logarithmic, linear, polynomial, and exponential functions. We measure the fitness by the R^2 score and pick the fit with the highest R^2 value. Below we describe the pseudocode of the algorithm, which provides a clear method for the ECA classification.

1. $time\,bound := T_{max}$
2. for $n = 4, 5, 6, 7, \ldots$ do:
   (a) $error = 1$, $samples = empty\,array$
   (b) while $error > 0.1$:
   - generate transient lengths of 20 randomly chosen initial configurations on a grid of size n and append them to the array of *samples*
   - if the process of generating the transient length of a single configuration takes more than T_{max}, exit and go to 3
   - update the error of the samples

   (c) compute $\widehat{\mu}_n$
3. we have obtained average transient estimates $(\widehat{\mu}_4, \widehat{\mu}_5, \ldots, \widehat{\mu}_N)$ for increasing grid size; we perform linear regression to const, log, lin, poly and exp functions and mark the R^2 score of each fit
4. if the maximum of the marked R^2 scores is greater than 0.9, classify the CA to the corresponding class; else, mark the CA as unclassified

2.3.3 Transient Classifiation of ECAs

For most ECAs we were able to generate data for grids as large as 400. We found a good fit with $R^2 > 90\%$ for most ECAs and obtained four major classes named Bounded, Log, Lin, and Exp Class. The exception are rules in classes Affine and Fractal, which were not fitted to any functions due to their specific form. Below, we give a more detailed description of each class.

Bounded Class: 27/88 rules (30.68%). The average transient lengths were bounded by a constant independent of the grid size. This suggests that the long term dynamics of such automata can be predicted efficiently.

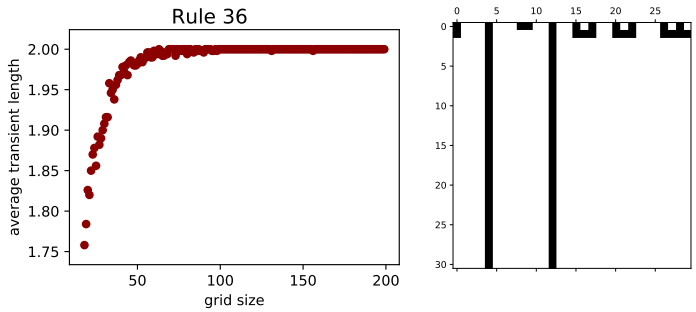


Figure 2.6: Bounded Class rule 36. The average transient plot is on the left, the space-time diagram on the right.

Log Class: 39/88 rules (44.32%). The largest ECA class exhibits possible logarithmic average transient growth. The event of two cells at a large distance "communicating" is improbable for this class.

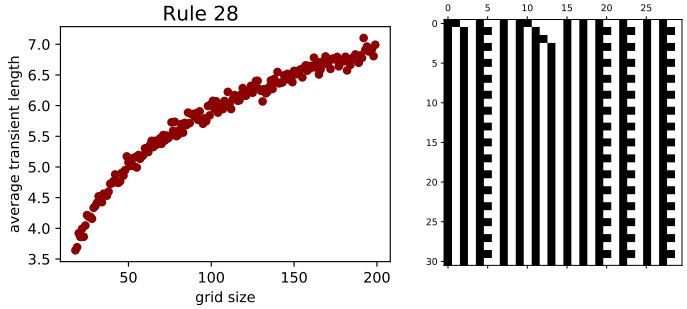


Figure 2.7: Log Class rule 28. The average transient plot is on the left, the space-time diagram on the right.

Lin Class: 8/88 rules (9.09%). On average, information can be aggregated from cells at an arbitrary distance. This class contains automata whose space-time diagrams resemble some sort of computation. This is supported by the fact that it contains two rules known to have a non-trivial computational capacity: rule 184, which computes the majority of black and white cells, and rule 110, which is the only ECA proven to be Turing complete.

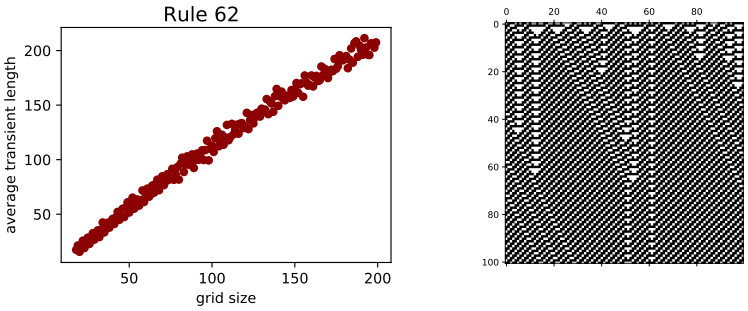


Figure 2.8: Lin Class rule 62. The average transient plot is on the left, the space-time diagram on the right.

We note that rules in this class are not necessarily complex as the interesting behavior seems to correlate with the slope of the linear growth. Most of the Class Lin rules have only a very gradual incline. In fact, the only two rules with such a slope greater than 1, rules 110 and 62, seem to be the ones with the most interesting space-time diagrams.

We are aware of the issue that average transients of rules in Lin Class might turn out to grow logarithmically or exponentially given enough data samples. This could explain that the behavior of ECAs in Lin Class depends on the slope of the transient growth. However, obtaining enough data points to distinguish the asymptotic growth might not be computationally feasible. Therefore, the class could be interpreted as consisting of rules, which might have a logarithmic or exponential growth, but this could not be decided given only a limited amount

of data points. However, given such limited data, the best fit for such rules seems to be to a linear function.

Exp Class: 6/88 rules (6.82%). This class has a striking correspondence to automata with chaotic behavior. Visually, there seem to be no persistent patterns in the configurations. Not only the transients, but also the attractor lengths are significantly larger than for other rules. This class contains rules 45, 30, and 106 whose transients grow the fastest, as well as rules 54, 73, and 22. We note that the average transient length of rule 45 seems to grow exponentially only for grids of even size. For grids of an odd size, rule 45 seems to be reversible. We will prove the reversibility of rule 45 operating on grids of odd size in the subsequent chapter.

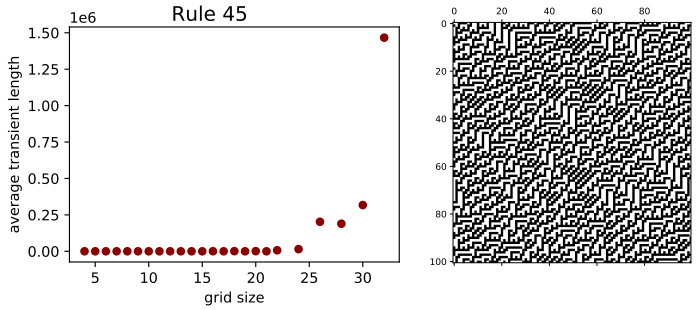


Figure 2.9: Exp Class rule 45. The average transient plot is on the left, the space-time diagram on the right.

Affine Class: 4/88 rules (4.55%). This class contains rules 60, 90, 105, and 150 whose local rules are affine Boolean functions. Such automata can be studied algebraically and predicted efficiently. It was shown in [13] that the transient lengths of rule 90 depend on the largest power of 2 that divides the grid size. Therefore, the measured data did not fit any of the functions above but formed a rather specific pattern.

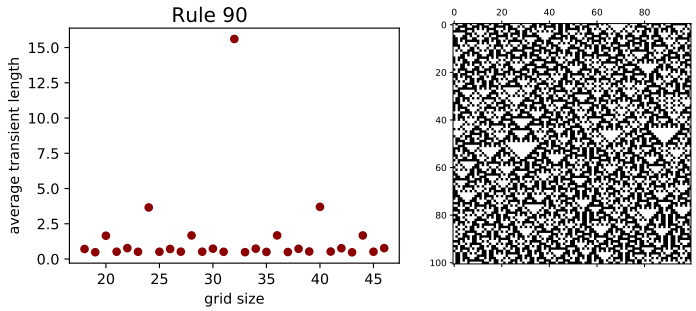


Figure 2.10: Affine Class rule 90. The average transient plot is on the left, the space-time diagram on the right.

Fractal Class: 4/88 rules (4.55%). This class contains rules 18, 122, 126, and 146, which are sensitive to initial conditions. Their evolution either produces a

fractal structure resembling a Sierpinski triangle or a space-time diagram with no apparent structure. We could say such rules oscillate between easily predictable behavior and chaotic behavior. Their average transients and periods grow quite fast, which makes it difficult to gather data for larger grid sizes.

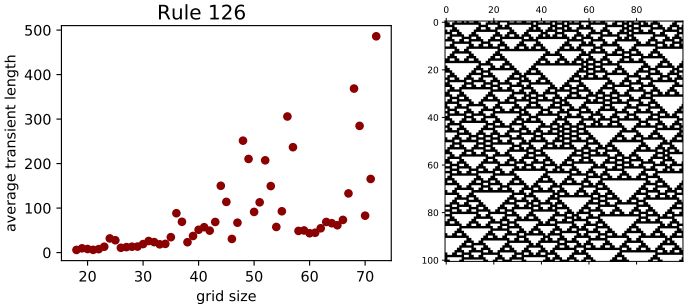


Figure 2.11: Fractal Class rule 126. The average transient plot is on the left, the space-time diagram on the right.

2.3.4 Discussion

We have performed a simple linear regression minimizing the mean square error. One of the assumptions, which is used when proving that such method guarantees to find the best fit, is homoscedasticity of the data. In our case, when observing the plots of the average transient estimations, we could notice that for some ECAs the data seems to have an increasing variance with increasing grid size (e.g. rules 62 or 126). In such a case, we are not guaranteed to have found functions with the best fit. However, as we only use a simple linear regression method, this is in practice not considered to be a significant problem. Observing the data manually, it would be difficult to imagine better fits for the data than the ones found by the regression method.

We note that we have also tried to measure the asymptotic growth of the average attractor size $\frac{1}{2^n}\sum_{u\in\{0,1\}^n} a_u$, as well as the average of the rho value $\frac{1}{2^n}\sum_{u\in\{0,1\}^n} \rho_u$. However, we obtained data points that could not be fitted to simple functions well. This is due to the fact that many automata have attractors consisting of a configuration, which is shifted by one bit to the left, resp. right, at every time step. The size of such an attractor then depends on the greatest common divisor of the size of the period of the attractor and the grid size. This causes oscillations of the measured data. We conclude that such phase-space properties are not suitable for this classification method.

Below, we compare our results to other classifications described earlier. Exhaustive comparison for each ECA is presented in Figure 2.12.

Wolfram's Classification - Discussion The significance of our results stems precisely from the fact that the Transient Classification corresponds to Wolfram's so well. As it is not clear for many rules, which Wolfram class they belong to, the main advantage is that we provide a formal criterion, upon which this could be decided.

In particular, rules in Classes Bounded and Log correspond to rules in either Class 1 or 2. Class Exp corresponds to the chaotic Class 3 and Class Lin contains

Classification Comparison				
ECA	**Transient**	**Wolf.**	**Zenil**	**Wuen.**
0	bounded	1	1 or 2	0
1	bounded	2	1 or 2	0.25
2	bounded	2	1 or 2	0.25
3	bounded	2	1 or 2	0.25
4	bounded	2	1 or 2	0.25
5	bounded	2	1 or 2	0.5
6	log	2	1 or 2	0.5
7	log	2	1 or 2	0.75
8	bounded	1	1 or 2	0.25
9	lin	2	1 or 2	0.5
10	bounded	2	1 or 2	0.5
11	log	2	1 or 2	0.75
12	bounded	2	1 or 2	0.5
13	log	2	1 or 2	0.75
14	lin	2	1 or 2	0.75
15	bounded	2	1 or 2	1
18	fractal	2/3	1 or 2	0.5
19	bounded	2	1 or 2	0.625
22	exp	2/3	1 or 2	0.75
23	log	2	1 or 2	0.5
24	bounded	2	1 or 2	0.5
25	lin	2	1 or 2	0.75
26	log	2	1 or 2	0.75
27	log	2	1 or 2	0.75
28	log	2	1 or 2	0.75
29	bounded	2	1 or 2	0.5
30	exp	3	3	1
32	log	1	1 or 2	0.25
33	log	2	1 or 2	0.5
34	bounded	2	1 or 2	0.5
35	log	2	1 or 2	0.625
36	bounded	2	1 or 2	0.5
37	log	2	1 or 2	0.75
38	bounded	2	1 or 2	0.75
40	log	1	1 or 2	0.5
41	log	2	1 or 2	0.75
42	bounded	2	1 or 2	0.75
43	lin	2	1 or 2	0.5
44	log	2	1 or 2	0.75
45	exp	3	3	1
46	bounded	2	1 or 2	0.5
50	log	2	1 or 2	0.625
51	bounded	2	1 or 2	1
54	exp	2/4	1 or 2	0.75

Classification Comparison				
ECA	**Transient**	**Wolf.**	**Zenil**	**Wuen.**
56	log	2	1 or 2	0.75
57	lin	2	1 or 2	0.75
58	log	2	1 or 2	0.75
60	affine	2	1 or 2	1
62	lin	2	1 or 2	0.75
72	bounded	1	1 or 2	0.5
73	exp	3/4	3	0.75
74	log	2	1 or 2	0.75
76	bounded	2	1 or 2	0.625
77	log	2	1 or 2	0.5
78	log	2	1 or 2	0.75
90	affine	2	1 or 2	1
94	log	2	1 or 2	0.75
104	log	1	1 or 2	0.75
105	affine	2	1 or 2	1
106	exp	3	1 or 2	1
108	bounded	1	1 or 2	0.75
110	lin	4	4	0.75
122	fractal	2/3	1 or 2	0.75
126	fractal	2/3	1 or 2	0.5
128	log	1	1 or 2	0.25
130	log	2	1 or 2	0.5
132	log	2	1 or 2	0.5
134	log	2	1 or 2	0.75
136	log	1	1 or 2	0.5
138	bounded	2	1 or 2	0.75
140	log	2	1 or 2	0.625
142	lin	2	1 or 2	0.5
146	fractal	2/3	1 or 2	0.75
150	affine	2	1 or 2	1
152	log	2	1 or 2	0.75
154	bounded	2/3	1 or 2	1
156	log	2	1 or 2	0.75
160	log	1	1 or 2	0.5
162	log	2	1 or 2	0.75
164	log	2	1 or 2	0.75
168	log	1	1 or 2	0.75
170	bounded	2	1 or 2	1
172	log	2	1 or 2	0.75
178	log	2	1 or 2	0.5
184	lin	2	1 or 2	0.5
200	bounded	1	1 or 2	0.625
204	bounded	2	1 or 2	1
232	log	1	1 or 2	0.5

Figure 2.12: Comparing classifications of the 88 unique ECAs.

Class 4 together with some Class 2 rules. We mention an interesting discrepancy: rule 54, which is possibly considered by Wolfram to be Turing complete, belongs to the Class Exp. This might suggest that computations performed by this rule can be, on average, quite inefficient.

Zenil's Classification - Discussion Zenil's Classification of ECAs offers a great formalization of Wolfram's and seems to correspond to it roughly. The main issue of this method is that the outcome of the classification is very sensitive to the choice of many parameters. Compared to the Transient Classification, it is less fine grained. In addition, it uses a clustering technique that requires data of multiple automata to be mutually compared in order to give rise to different classes. In contrast, the Transient Class can be determined for a single automaton without any context. Lastly, most compression algorithms are meant to compress one-dimensional or two-dimensional objects. Therefore, it is not straightforward how to extend Zenil's method to classify CAs from higher dimensional CA spaces.

The Transient Classification has no such issue and, in principle, can be extended to an arbitrary space of CAs.

Wuensche's Z-parameter - Discussion Wuensche suggests that complex behavior occurs around $Z = 0.75$, which agrees with the fact that Lin Class rules with a steep slope (rules 110, 62, and 25) have precisely this Z value. However, the $Z = 0.75$ is in fact quite frequent. This suggests that thanks to its simplicity, the Z parameter can be used to narrow down a vast space of CA rules when searching for complexity. However, more refined methods have to be subsequently applied to find concrete CAs with interesting behavior.

2.3.5 Transient Classification of 2D CAs

So far, we have examined the toy model of ECAs. The actual usefulness of the classification would stem from its application to more general CA spaces where it could be used to discover automata with interesting behavior. We, therefore, applied it on a subset of two-dimensional CAs with Moore neighborhood and 3 states to see whether 2D automata would exhibit such evident transient growths.

When measuring the average transient length of 2D CAs, we only consider the cyclic grids to be of the form $\mathbb{Z}_n \times \mathbb{Z}_n$, $n \in \mathbb{N}^+$. In contrast to the 1D case, we measure the average transient growth with respect to n rather than the size of the grid, which is n^2. This is motivated by the fact that in a $n \times n$ grid, the greatest distance between two cells depends linearly on n rather than quadratically.

To reduce the vast automaton space, we only considered such automata whose local rules are invariant to all the symmetries of the neighborhood (rotations and reflections of a 3×3 square). As there are still 3^{2861} such symmetrical 2D CAs, we randomly sampled 10 000 of them.

Classification of 2D 3-state CAs (10 000 samples)	
Transient Class	Percentage of CAs
Bounded Class	0%
Log Class	18.21%
Lin Class	1.17%
Poly Class	1.03%
Exp Class	72.62%
Unclassified	6.97%

Table 2.1: Classification of 10 000 randomly sampled symmetric 2D 3-state CAs.

We were able to classify 93.03% of 10 000 sampled automata with a time bound of 40 seconds for the computation of one transient length value on a single CPU. We estimate that most CAs are unclassified due to such computation time restriction or due to rather strict conditions we imposed on a good regression fit. We obtained the same significant classes, the Log, Lin, and Exp Class. However, in this case, the Exp Class seems to dominate the rule space. Another interesting difference is that a new class was observed — the polynomial class that contains rules whose transients grow approximately quadratically. Moreover, our results suggest that the occurrence of bounded class CAs in 2D is much scarcer as we found no such CAs in our sample.

We observed the space-time diagrams of randomly sampled automata from each class to infer its typical behavior. On average, the Log Class automata quickly enter attractors of small size. Lin Class exhibit emergence of various local structures. For automata with more gradual incline, such structures seem to die out quite fast. However, automata with steeper slopes exhibit complex interactions of such structures. The Poly class automata with a steep slope seem to produce spatially separated regions of chaotic behavior against a static background. In the case of more gradual slopes, some local structures emerge. Finally, the Exp Class seems to be evolving chaotically with no apparent local structures. We present various examples of CA evolution dynamics in the form of GIF animations here[1].

These observations suggest that the region of Lin Class with a steep slope and Poly class with a more gradual incline seems to contain a non-trivial ratio of automata with complex behavior. In this sense, the Transient Classification can assist us to automatically search for complex automata similarly to [2] where interesting automata were discovered by measuring the growth of structured complexity using a data compression approach.

2.3.6 Transient Classification of Well Known CAs

We were interested whether some well-known complex automata from larger CA spaces would conform to the transient classification as well. Surprisingly, the result is positive.

Game of Life As the left plot in Figure 2.13 suggests, the famous cellular automaton called Game of Life [5], which is proved to be Turing complete, seems to fit the Lin Class. This is confirmed by the linear regression fit with $R^2 \approx 98.4\%$.

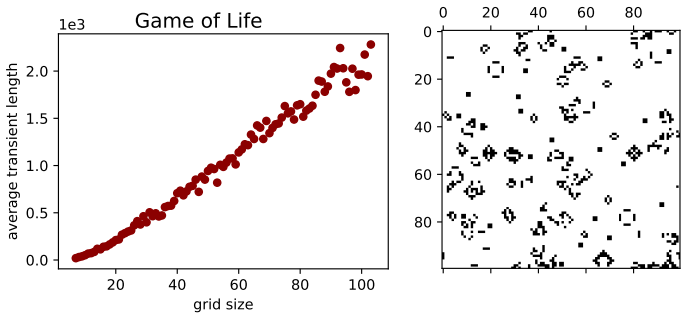


Figure 2.13: Game of Life. Average transient growth plot is on the left. On the right, we show a diagram at time 200 started from a random initial configuration.

Genetically Evolved Majority CA In [16], it is studied how genetic algorithms can evolve CAs capable of global coordination. The authors were able to find a 1D CA denoted as ϕ_{par} with two states and radius $r = 3$, which is quite successful at computing the majority task with the output required to be of the form of a homogenous state of either all 0's or all 1's.

[1] `https://bit.ly/ca_animations`

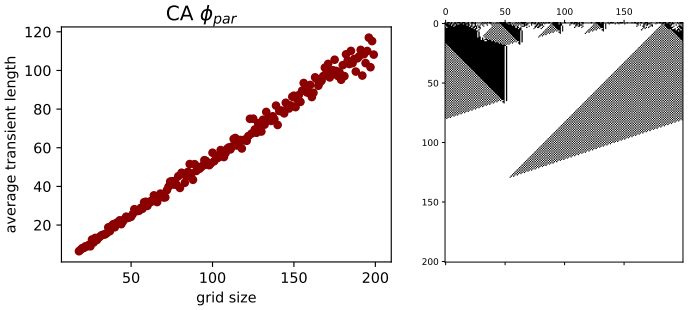


Figure 2.14: Cellular automaton ϕ_{par}. The average transient growth plot is on the left. On the right, we show a space-time diagram simulated from a random initial configuration.

This CA seems to belong to the Lin Class, which is confirmed by the linear regression fit with $R^2 \approx 99.2\%$.

Totalistic 1D 3-state CA A totalistic CA is any CA whose local rule depends only on the number of cells in each state and not on their particular position. Wolfram studied various CA classes, one of them being the totalistic 1D CAs with radius $r = 1$ and 3 states $S = \{0, 1, 2\}$. Considering only totalistic CAs significantly reduces the number of 1D CAs with radius 1 and 3 states and also makes it possible to represent the local rules by a short description.

In [28], Wolfram presents a list of possibly complex CAs from this class. We applied the Transient Classification to such CAs and learned that most of them were classified as logarithmic. This agrees with our space-time diagram observations that the local structures in such CAs "die out" quite quickly. Nonetheless, some of the CAs were classified as linear. An example of such a CA is in Figure 2.15 where the linear regression fit has $R^2 \approx 97.63\%$.

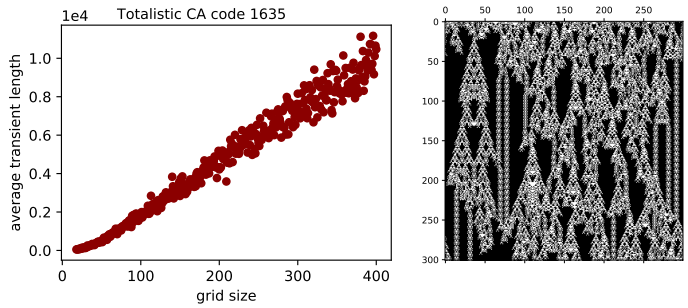


Figure 2.15: Totalistic cellular automaton with code 1635. The average transient growth plot is on the left. On the right, we show a space-time diagram of the evolution from a random initial configuration.

3. Preimages of Cellular Automata

So far, we have studied the "forward dynamics" of cellular automata. That is, we have observed the trajectories of configurations obtained when time is progressing forward. In this chapter, we explore another approach - we study the properties of CAs' phase-spaces related to the "backward dynamics" of the system. More concretely, we study CAs' properties related to the number of preimages of different configurations. An ultimate goal would be to relate the forward and backward dynamics of CAs in order to develop a better understanding of the notion of complexity within the model of cellular automata.

We have already described some work of Andrew Wuensche, who has examined CAs' backward dynamics. In this chapter, we describe a novel representation of 1D CAs, which we call the *Cayley graph of a CA*, and we demonstrate its usefulness by using it to obtain some new results about the phase-space properties of a particular ECA. We note that all the methods described in this chapter are applicable to any one-dimensional cellular automaton; however, for simpler notation, we present them in the context of elementary cellular automata.

3.1 Preimage Relations of Cellular Automata

In this section, we define the preimage relations of CAs and describe how do they relate to the number of preimages of CA configurations. This notion is subsequently used to define the Cayley graphs of CAs.

Definition 10. *For an ECA with local rule f we define two binary relations $R_0^f, R_1^f \subseteq \mathbf{2}^2 \times \mathbf{2}^2$ called the preimage relations as follows:*

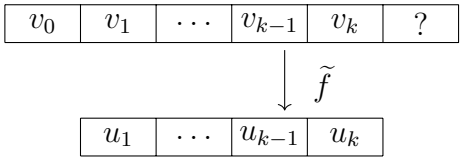
$$R_0^f = \{(ab, bc) \mid f(a, b, c) = 0\}$$
$$R_1^f = \{(ab, bc) \mid f(a, b, c) = 1\}.$$

If the local rule f is clear from the context, we omit the superscript in the notation of the relations.

We illustrate the meaning of these relations in the case where the ECA operates on an infinite grid. Let $u \in \mathbf{2}^{\mathbb{Z}}$ be a configuration, for which we want to construct a preimage. Suppose we have already constructed a partial preimage $v \in \mathbf{2}^+$, $v = v_0 v_1 v_2 \dots v_k$, and that we would like to append one letter to v, that is, obtain $v' = v_0 v_1 \dots v_k v_{k+1}$, so that v' would still be a partial preimage of u. That is, we search for $v_{k+1} \in \mathbf{2}$ such that $f(v_{k-1} v_k v_{k+1}) = u_k$. Then, all the possible symbols we could append to v are exactly symbols $x \in \mathbf{2}$ satisfying $(v_{k-1} v_k, v_k x) \in R_{u_k}$.

Below, we relate the properties of the preimage relations to the existence of preimages of cellular automata configurations. For that, we introduce the following notation.

Let B be an arbitrary set. We define the *diagonal relation on B* as $diag(B) = \{(b, b) \mid b \in B\}$. For a binary relation $R \subseteq B \times B$ we define its *fixed points* as the elements of the set $R \cap diag(B)$.



Let $R, S \subseteq B \times B$ be binary relations on B. We define the composition of R and S as a binary relation

$$R \circ S = \{(r, s) \mid \text{there exists } x \in B \text{ such that } (r, x) \in R \text{ and } (x, s) \in S\}.$$

This way, the definition, unfortunately, does not correspond to the function composition but for our purposes, it is much easier to work with. Clearly, the composition operation is associative, and therefore, we can write the composition of $n \in \mathbb{N}^+$ binary relations $R_0, R_1, \ldots, R_{n-1} \subseteq B \times B$ as $R_0 \circ R_1 \circ \ldots \circ R_{n-1}$. We define the inverse relation of R as

$$R^{-1} = \{(s, r) \mid (r, s) \in R\}$$

and we define the *domain of* R as:

$$dom(R) = \{r \mid \text{there exists an } s \in B \text{ such that } (r, s) \in R\} \subseteq B.$$

Suppose we have an ECA and its two preimage relations $R_0, R_1 \subseteq \mathbf{2}^2 \times \mathbf{2}^2$. Let $u \in \mathbf{2}^+$, $u = u_0 u_1 \ldots u_{n-1}$ for some $n \in \mathbb{N}^+$. We define the relation

$$R_u \coloneqq R_{u_0} \circ R_{u_1} \circ \ldots \circ R_{u_{n-2}} \circ R_{u_{n-1}}.$$

Let $(a, b) \in R_u$. We call a sequence $(w_0 \coloneqq a, w_1, \ldots, w_{n-1}, w_n \coloneqq b)$ where each $w_i \in \mathbf{2}^2$ a *witness* for u and (a, b) if it satisfies that:

$$(w_i, w_{i+1}) \in R_{u_i} \text{ for all } i \in \{0, 1, \ldots, n-1\}.$$

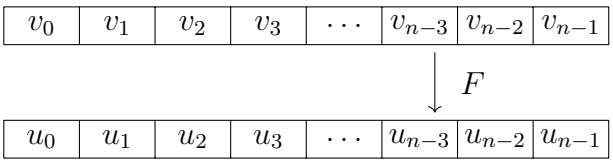
Lemma 11. *Let $(\mathbf{2}^n, F)$ be an elementary cellular automaton operating on a finite grid of size $n \in \mathbb{N}$, and let R_0, R_1 be its preimage relations. Then, a configuration $u \in \mathbf{2}^n$, $u = u_0 u_1 \ldots u_{n-1}$, has a preimage under F if and only if the relation $R_u = R_{u_0} \circ R_{u_1} \circ \ldots \circ R_{u_{n-1}}$ has a fixed point. Moreover, we have a natural bijection between the set of preimages of u and the set of witnesses for u and all fixed points of R_u. The bijection is given by:*

$$\beta : F^{-1}(u) \longrightarrow \bigcup_{(a,a) \text{ fixed point of } R_u} \{w \in (\mathbf{2}^2)^{n+1} \mid w \text{ witness for } u \text{ and } (a, a)\}$$

$$\beta(v_0 v_1 \ldots v_{n-2} v_{n-1}) = (v_{n-1} v_0, v_0 v_1, \ldots, v_{n-2} v_{n-1}, v_{n-1} v_0).$$

Proof. Let us fix a configuration $u \in \mathbf{2}^n$ for some $n \in \mathbb{N}^+$. Clearly, the mapping β is injective. We show that it indeed maps each preimage of u to a witness for u and some fixed point of R_u. Subsequently, we show that β is surjective.

Suppose $v \in \mathbf{2}^n$ is a preimage of u, $v = v_0 v_1 \ldots v_{n-1}$.



Then clearly:

$$(v_{n-1}v_0, v_0v_1) \in R_{u_0} \quad \Leftrightarrow f(v_{n-1}, v_0, v_1) = u_0. \tag{3.1}$$
$$(v_0v_1, v_1v_2) \in R_{u_1} \quad \Leftrightarrow f(v_0, v_1, v_2) = u_1 \tag{3.2}$$
$$\vdots \tag{3.3}$$
$$(v_{n-3}v_{n-2}, v_{n-2}v_{n-1}) \in R_{u_{n-2}} \Leftrightarrow f(v_{n-3}, v_{n-2}, v_{n-1}) = u_{n-2} \tag{3.4}$$
$$(v_{n-2}v_{n-1}, v_{n-1}v_0) \in R_{u_{n-1}} \Leftrightarrow f(v_{n-2}, v_{n-1}, v_0) = u_{n-1}. \tag{3.5}$$

Therefore, we get that $(v_{n-1}v_0, v_{n-1}v_0) \in R_u$ and indeed, the sequence $(v_{n-1}v_0, v_0v_1, \ldots, v_{n-2}v_{n-1}, v_{n-1}v_0)$ is a witness for u and $(v_{n-1}v_0, v_{n-1}v_0)$. Conversely, suppose we have a witness for u and a fixed point $(a, a) \in R_u$ given by $(w_0 = a, w_1, \ldots, w_{n-1}, w_n = a)$. Then, there exist elements $v_i \in \mathbf{2}$, $i \in \{0, 1, \ldots, n-1\}$ such that

$$\begin{aligned} w_1 &= v_0v_1 \\ w_2 &= v_1v_2 \\ &\vdots \\ w_{n-1} &= v_{n-2}v_{n-1} \\ w_n &= v_{n-1}v_0 = w_0. \end{aligned}$$

From the equivalences (3.1) - (3.5), it is clear that $v_0v_1 \ldots v_{n-1}$ is a preimage of u. Hence, β is a bijection. □

Definition 12. *Let $(\mathbf{2}^n, F)$, resp. $(\mathbf{2}^{\mathbb{Z}}, F)$ be an elementary cellular automaton operating on a cyclic grid of size n, resp. on an infinite grid. We say that a configuration $u \in \mathbf{2}^n$, resp. $u \in \mathbf{2}^{\mathbb{Z}}$ is a garden of eden (GOE) configuration if u has no preimage under F.*

The primary goal of this chapter is to study how to characterize all the garden of eden configurations of a given ECA as this question is by no means straightforward to answer. In the case of an ECA operating on a cyclic grid, the ECA's phase-space properties, including the number of GOE configurations, can vary dramatically depending on the grid size.

So far, from Lemma 11, we have obtained that $u \in \mathbf{2}^+$ is a GOE configuration of a given ECA operating on a cyclic grid with local rule f if and only if R_u^f contains no fixed points. Below, we present a preliminary for the characterization of GOE configurations of ECAs operating on infinite grids.

Lemma 13. *Let $(\mathbf{2}^{\mathbb{Z}}, F)$ be an elementary cellular automaton given by a local rule f with preimage relations R_0 and R_1. Then, a configuration $u \in \mathbf{2}^+$ has a preimage under $\widetilde{f}$ if and only if the relation R_u is nonempty. Moreover, we have a bijection between preimages of u under $\widetilde{f}$ and the set of witnesses for u and all elements in the relation R_u.*

Proof. Analogous to the proof of Lemma 11. □

Definition 14. *Let A be a finite alphabet, $r \in \mathbb{N}$ and $f : A^{2r+1} \to A$ a local rule of a 1D cellular automaton. We call such words, which have no preimage under $\widetilde{f}$, orphans of $\widetilde{f}$.*

3.2 Cayley Graphs of Cellular Automata

We define a *relational monoid on* $\mathbf{2}$ as a monoid $(\mathcal{P}(\mathbf{2} \times \mathbf{2}), \circ, \mathbf{1})$ of all binary relations on $\mathbf{2}$ with $\circ$ being the composition operation of relations and $\mathbf{1} = \{(00, 00), (01, 01), (10, 10), (11, 11)\}$ being the neutral element with respect to $\circ$.

Definition 15. *Let f be a local rule of some ECA. We define the preimage monoid of f*

$$\mathcal{M}^f = (M^f, \circ, \mathbf{1})$$

as a submonoid of the relational monoid on $\mathbf{2}$ generated by the preimage relations R_0^f, R_1^f.

Although we define the preimage monoid only for ECAs, it can be defined for any one-dimensional cellular automaton. Given a finite set A and a local rule $f : A^{2r+1} \to A$ of some 1D CA, the size of the preimage monoid is bounded by the number of binary relations on the set $A^{2r} \times A^{2r}$, which is equal to $2^{|A|^{4r}}$. In the case of ECAs, the upper bound on the size of the monoid is 2^{16}.

Before we define the Cayley graph of an ECA, we define the notion of a labeled directed graph to be a triplet $\mathcal{G} = (V, E, l)$, where V is the set of vertices, $E \subseteq V \times V$ is the set of edges, and $l : E \to L$ is a mapping from the set of edges to some finite set L of edge labels.

Let $\mathcal{G} = (V, E, l)$ be a labeled directed graph. We define a *walk of length* $n \in \mathbb{N}^+$ in $\mathcal{G}$ to be a sequence $w = (v_0, e_1, v_1, e_1, \ldots, e_n, v_n)$ where

- $v_i \in V$ for each $i \in \{0, 1, \ldots, n\}$,
- $e_i = (v_{i-1}, v_i) \in E$ for each $i \in \{1, 2, \ldots, n\}$.

In such a case, we say that the walk w *starts* at the vertex v_0 and *terminates* in the vertex v_n. Note that in a walk, both vertices and edges can be repeated. Let v, $w \in V$. We say that w is a *successor of* v if there exists a walk in $\mathcal{G}$ starting at v and terminating in w.

Definition 16. *Let f be the local rule of some ECA. We define the Cayley graph of the ECA as a labeled directed graph $\mathcal{C}^f = (V^f, E^f, l^f)$ where:*

- *$V^f = M^f$ is the set of elements of the preimage monoid of f,*
- *there is an edge (R, S) labeled by $x \in \{0, 1\}$ if and only if $R \circ R_x = S$.*

Clearly, the Cayley graph of any ECA has at most 2^{16} vertices. From each vertex $R \in V^f$ there are exactly two edges pointing towards vertices corresponding to $R \circ R_0$ and $R \circ R_1$, which are labeled by 0 and 1, respectively.

Let f be the local rule of some ECA. The key property of the Cayley graph (V^f, E^f, l) is that we have a correspondence between ECA configurations $u \in \mathbf{2}^+$ and walks in the Cayley graph starting at $\mathbf{1}$ and terminating at the vertex R_u. More concretely, each configuration $u \in \mathbf{2}^n$ uniquely determines a walk $(v_0 = \mathbf{1}, e_1, v_1, \ldots, e_n, v_n)$ such that $l(v_i, v_{i+1}) = u_i$ for all $i \in \{0, 1, \ldots, n-1\}$. Moreover, the walk satisfies that $v_n = R_u$. Therefore, given the Cayley graph of some ECA together with a configuration $u \in \mathbf{2}^+$, we can decide whether u has

a preimage on the cyclic grid by "following the walk given by u in the Cayley graph" and examining whether the terminal vertex of such a walk, R_u, has a fixed point.

Just from the definition of a Cayley graph, we obtain a simple result about the garden of eden configurations of ECAs, and by a straightforward generalization, of any 1D CA.

Observation 17. *Let F_n denote the global rule of an ECA operating on a cyclic grid of size n. Then, the set of all its garden of eden configurations*

$$L = \{u \in \mathbf{2}^+ \mid u \text{ is a garden of eden configuration for } (\mathbf{2}^{|u|}, F_{|u|})\}$$

forms a regular language.

Proof. We fix an ECA and denote its Cayley graph as $\mathcal{C} = (V, E, l)$ and the set of all its GOE configurations as L.

The language L corresponds exactly to the walks starting at $\mathbf{1}$ and terminating in any node corresponding to a relation without any fixed points. We define the DFA accepting L in a straightforward way - we set its set of states to be V, the set of input symbols to be $\mathbf{2}$, and the initial state to be $\mathbf{1}$. We define the transition function $\delta : V \times \mathbf{2} \to V$ as $\delta(v, a) = v \circ R_a$ and the set of accepting states to be $F = \{R \in V \mid R \cap diag(\mathbf{2}) = \emptyset\}$. The accepting language of such a DFA is precisely L. □

3.2.1 Matrix Representation of Cayley Graphs

In this section, we describe how the preimage relations can be represented as matrices.

General Matrix Monoid

We present a new structure, the *general matrix monoid*, which gives us information not only about the existence of a preimage of a configuration but also about the number of such preimages. However, there are ECAs, for which the general matrix monoid is infinite, which we demonstrate on an example.

First, we formally define a mapping T that transforms any number in its binary representation to its decimal one.

$$T : \mathbf{2}^+ \longrightarrow \mathbb{N}$$
$$T(u_0 u_1 \dots u_{n-1} u_n) = \sum_{i=0}^{n} 2^{n-i} u_i.$$

In particular, we have that $T(00) = 0$, $T(01) = 1$, $T(10) = 2$, and $T(11) = 3$.

Definition 18. *Given a binary relation $R \subseteq \mathbf{2} \times \mathbf{2}$ we define its matrix representation to be a four by four binary matrix $M_R \in \mathbf{2}^{4\times 4}$ where*

$$(M_R)_{(T(a),T(b))} = \begin{cases} 1, & \text{if } (a, b) \in R \\ 0, & \text{otherwise.} \end{cases}$$

Definition 19. *Let R_0, $R_1 \subseteq \mathbf{2} \times \mathbf{2}$ be the two preimage relations of some ECA. We define the general matrix monoid, $\mathcal{G} = (G, \cdot, I)$, of the ECA as a monoid generated by matrices $M_0 \coloneqq M_{R_0}, M_1 \coloneqq M_{R_1} \in \mathbb{N}^{4\times4}$ where the matrix multiplication $\cdot$ is considered over the semiring $\mathbb{N}$ and $I \in \mathbb{N}^{4\times4}$ is the identity matrix.*

Given the general matrix monoid $\mathcal{G} = (G, \cdot, I)$ of some ECA, generated by $M_0, M_1 \in \mathbb{N}^{4\times4}$, we define $M_u \coloneqq M_{u_0} \cdot M_{u_1} \cdot \ldots \cdot M_{u_{n-1}}$. In Lemma 20, we describe what information is encompassed in the matrix M_u.

Lemma 20. *Suppose we have an ECA operating on a cyclic grid of size $n \in \mathbb{N}^+$. Let $\mathcal{G} = (G, \cdot, I)$ be its general matrix monoid and let $u \in \mathbf{2}^n$. Then, $(M_u)_{(T(a),T(b))}$ denotes the number of witnesses for u and (a, b).*

Proof. We prove this by induction on n. The claim is straightforward for $n = 1$. Suppose that we have $n > 1$ and that the claim holds for all $w \in \mathbf{2}^+$, $|w| \leq n$. Let $u \in \mathbf{2}^{n+1}$, $u = u_0 \ldots u_n$. We put $u' = u_1 \ldots u_n$. Then, $(M_u)_{(T(a),T(b))} = (M_{u_0} \cdot M_{u'})_{(T(a),T(b))} = \sum_{x\in\mathbf{2}^2} (M_{u_0})_{(T(a),T(x))} \cdot (M_{u'})_{(T(x),T(b))}$. Each summand gives us the number of witnesses for u_0 and (a, x) times the number of witnesses for u' and (x, b) for a particular $x \in \mathbf{2}^2$. Those are exactly the witnesses for u, which "pass through x". Therefore, the sum gives us exactly the number of witnesses for u and (a, b). □

Corollary 21. *Suppose we have an ECA $(\mathbf{2}^n, F)$ operating on a cyclic grid of size $n \in \mathbb{N}^+$ with local rule f. Let $\mathcal{G} = (G, \cdot, I)$ be its general matrix monoid and let $u \in \mathbf{2}^n$. Then, $Tr(M_u)$ denotes the number of preimages of u under F.*

From Lemma 20 we obtain a straightforward algorithm, which computes the number of preimages of a configuration $u \in \mathbf{2}^n$ in linear time with respect to n, given a local rule of an ECA operating on a cyclic grid.

To see that the general matrix monoid can be infinite for some CAs, let us consider ECA rule 0. Clearly, for each $n \in \mathbb{N}^+$, the configuration 0^n has 2^n preimages on the cyclic grid and therefore, $lim_{n\to\infty} Tr(M_{0^n}) = \infty$. This implies that the general matrix monoid of rule 0 is infinite.

Finite Matrix Monoid

The Cayley graph of any ECA can be efficiently generated by a computer program. In order to do that, it is useful to represent the Cayley graph by a *finite matrix monoid*. We define the *matrix of ones* as $J = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$ and the *minimum of two matrices* $A, B \in \mathbb{N}^{4\times4}$ to be the matrix $min(A, B)$ where $min(A, B)_{(i,j)} = min\{A_{(i,j)}, B_{(i,j)}\}$ for $i, j \in \{0, 1, 2, 3\}$.

We define a binary operation $\bullet : \mathbb{N}^{4\times4} \times \mathbb{N}^{4\times4} \to \mathbb{N}^{4\times4}$ as follows:

$$A \bullet B = min(A \cdot B, J).$$

As we work with matrices with non-negative entries, it can easily be shown that $\bullet$ is an associative operation. Moreover, for any four by four binary matrix $A \in \mathbf{2}^{4\times4}$ it holds that $A \bullet I = min(A \cdot I, J) = min(A, J) = A = I \bullet A$. Therefore, the triplet $(\mathbf{2}^{4\times4}, \bullet, I)$ forms a monoid.

Definition 22. *Let us consider an ECA with preimage relations R_0 and R_1. We define the finite matrix monoid $\mathcal{F} = (F, \bullet, I)$ to be a submonoid of the monoid $(\mathbf{2}^{4\times 4}, \bullet, I)$ generated by M_{R_0} and M_{R_1}.*

Let us consider an ECA with preimage relations R_0 and R_1 whose preimage monoid is denoted as $\mathcal{M} = (M, \circ, \mathbf{1})$ and whose finite matrix monoid is denoted as $\mathcal{F} = (F, \bullet, I)$. Then, we have a natural bijection between the two monoids defined as:

$$\begin{aligned} \phi : \mathcal{M} &\longrightarrow \mathcal{F} \\ R &\longmapsto M_R. \end{aligned}$$

We show that ϕ is a homomorphism. Let $R, S \in M$. Then, $(M_{R\circ S})_{(T(a),T(b))} = 1$ if and only if there exists some $x \in \mathbf{2}$ such that $(a, x) \in R$ and $(x, b) \in S$. On the other hand, $(M_R \bullet M_S)_{(T(a),T(b))} = 1$ if and only if there exists some $x \in \mathbf{2}$ such that $(M_R)_{(T(a),T(x))} = 1$ and $(M_S)_{(T(x),T(b))} = 1$, which holds if and only if $(a, x) \in R$ and $(x, b) \in S$. Therefore, $(M_{R\circ S})_{(T(a),T(b))} = (M_R \bullet M_S)_{(T(a),T(b))}$ for all $a, b \in \mathbf{2}$.

The relations with no fixed points correspond to matrices with only 0's on their main diagonal. Therefore, given an ECA operating on a cyclic grid of size n and a configuration $u = u_0 \dots u_{n-1}$, we can efficiently decide whether u has a preimage by computing the matrix $M_{R_u} = M_{R_{u_0}} \bullet M_{R_{u_1}} \bullet \dots \bullet M_{R_{u_{n-1}}}$ and examining its diagonal.

3.3 Phase-Space Properties of Rule 45

We demonstrate the usefulness of the Cayley graph on the example of ECA rule 45. When examining the phase-space of rule 45 in Chapter 2, we have noticed that for small n, the average transient length seems to grow exponentially for n even. However, the rule seems reversible when operating on a cyclic grid of odd size. In this chapter, we will prove the reversibility of rule 45 for a cyclic grid of odd size using the Cayley graph representation. In fact, we will be able to characterize all configurations with a given number of preimages and also give a formula for computing the number of garden of eden configurations as a parameter of the grid size.

For the rest of this section, we fix f to be the local rule of ECA 45.

u	111	110	101	100	011	010	001	000
$f(u)$	0	0	1	0	1	1	0	1

We can notice that for each b, c and $u \in \mathbf{2}$ there is a unique $a \in \mathbf{2}$ such that $f(a, b, c) = u$. Using the terminology of Definition 5, every word $bc \in \mathbf{2}^2$ is right deterministic for f. This implies that every partial preimage can be uniquely prolonged by one symbol on the left. This property can be observed from the preimage relations R_0, R_1 for f, which are depicted in Figure 3.1 and, for which it holds that both R_0^{-1} and R_1^{-1} are functions. Therefore, it also holds that R_u^{-1} is a function for any $u \in \mathbf{2}^+$.

Lemma 23. *Let $(\mathbf{2}^n, F)$ denote the ECA rule 45 operating on a cyclic grid of size n. It holds that every $u \in \mathbf{2}^n$ has exactly one witness for every $(a, b) \in R_u$.*

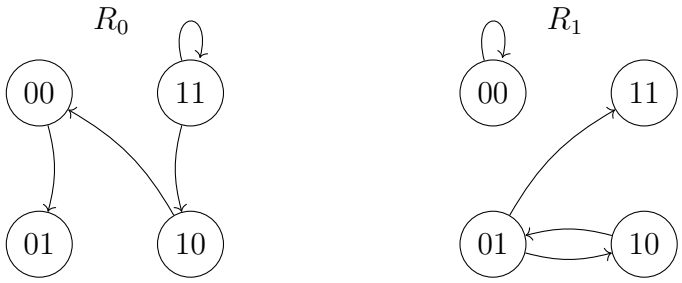


Figure 3.1: Preimage relations of rule 45.

Proof. Let $u = u_0u_1 \dots u_{n-1}$, $n \in \mathbb{N}^+$ and let $(a, b) \in R_u$. Since $R_{u_i}^{-1}$ is a function for all $i \in \{0, \dots, n-1\}$ it holds that there is exactly one sequence $(w_0, w_1, \dots, w_n = b)$ such that $w_i \in \mathbf{2}$ and $(w_i, w_{i+1}) \in R_{u_i}$ for all i. Since there must be some witness for u and (a, b), we have that $w_0 = a$, which completes the proof. □

Corollary 24. *For rule 45 operating on a cyclic grid, it holds that every word $u \in \mathbf{2}^+$ has exactly as many preimages as there are fixed points of the relation R_u.*

Before we present the Cayley graph of rule 45, we make one more observation in order to simplify the graph.

Lemma 25. *Let $(\mathbf{2}^n, F)$ denote the ECA rule 45 operating on a cyclic grid of size n. Let $u \in \mathbf{2}^n$ be such that there exists $v \sqsubseteq u$, for which R_v^{-1} is a constant function. Then, u has exactly one preimage under F.*

Proof. We will write $u = pvs$ where $p, s \in \mathbf{2}^*$. We need to show that R_u has exactly one fixed point. As the fixed points of R_u and R_u^{-1} are identical, it suffices to show that R_u^{-1} has exactly one fixed point.

We have that $R_u^{-1} = R_s^{-1} \circ R_v^{-1} \circ R_p^{-1}$ where both R_s^{-1} and R_p^{-1} are functions and R_v^{-1} is a constant function. This implies that their composition R_u is also a constant function, and therefore, has exactly one fixed point. □

Our goal is to use the Cayley graph of rule 45 to characterize all configurations with a given number of preimages. Using Lemma 25, we know that whenever there is an edge in the Cayley graph pointing towards a vertex R corresponding to the inverse of a constant function, we can disregard the vertex as well as all of its successors because all the walks starting at $\mathbf{1}$ and passing through R correspond to configurations with exactly one fixed point.

The graph can be examined in Figure 3.2. There are three vertices in the graph, which represent relations without fixed points. Let V_1 denote the terminal vertex of the walk starting at $\mathbf{1}$ given by 01. Similarly, let V_2 denote the terminal vertex of the walk given by 1101 and V_3 the terminal node of the walk given by 10.

$$V_1 = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \; V_2 = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \; V_3 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

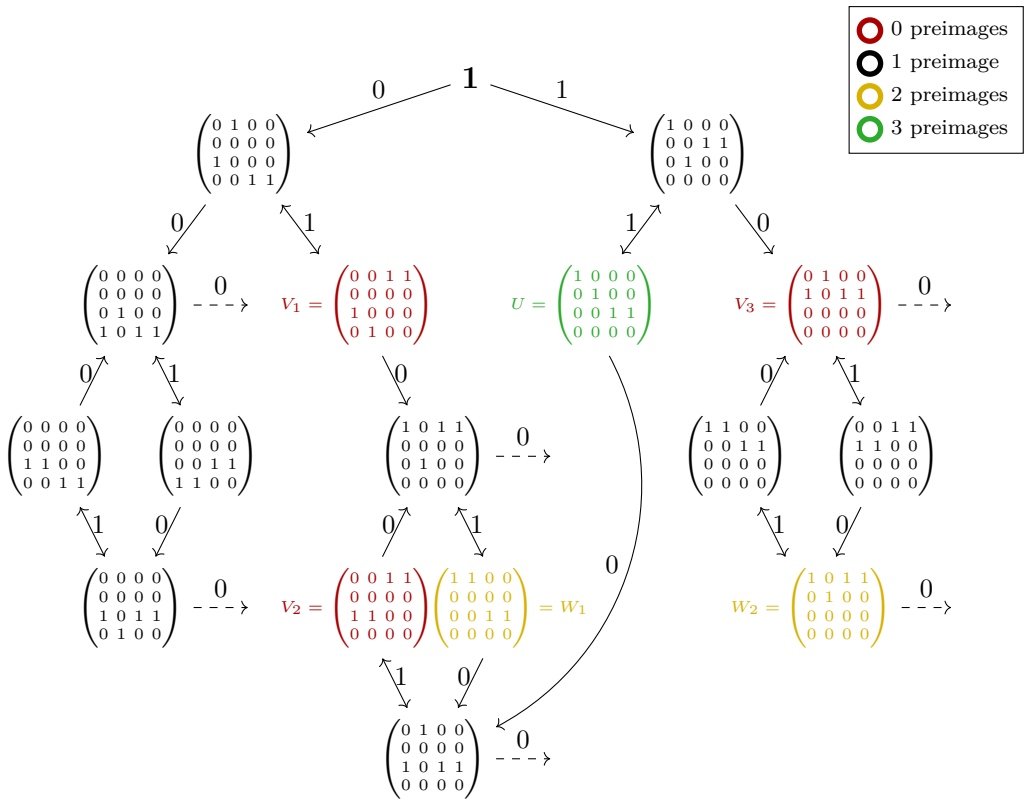


Figure 3.2: Cayley graph of rule 45. Vertices representing constant functions and their successors are not depicted. The edges pointing at such vertices are marked as dashed.

As the structure of the Cayley graph of rule 45 is very simple, the form of configurations corresponding to walks starting at **1** and terminating in either V_1, V_2, or V_3 can be observed directly from Figure 3.2. Below, we give a characterization of such walks.

1. It is straightforward that all the configurations that describe such walks terminating at V_1 form the set $01(11)^*$.

2. The walks terminating at V_2 either pass through V_1, or through the "green vertex" U. In the first case, the set of configurations corresponding to such paths is $01(11)^*01(11)^*01(11)^*\Big(01(11)^*01(11)^*\Big)^*$. In the second case, it is $11(11)^*01(11)^*\Big(01(11)^*01(11)^*\Big)^*$.

3. Lastly, there are walks terminating at V_3, which are the ones corresponding to configurations from the set $10\Big(1(11)^*01(11)^*0\Big)^*$.

We see that in all cases, the configurations consist of an odd number of 0's where each two 0's are separated by a block consisting of an odd number of 1's. In case 1, the configurations only consist of a single 0. All the configurations that consist of any of $2k+3$, $k \in \mathbb{N}$, 0's are described by the first scenario in case 2. All the configurations that consist of any of $2k+1$, $k \in \mathbb{N}$, 0's are described by the second scenario in case 2 together with case 3.

Corollary 26. *The configurations, which have no preimages under the ECA rule 45 operating on a finite cyclic grid, are exactly those consisting of an odd number of single 0's, each two 0's being separated by a block of an odd number of 1's.*

We briefly mention an outline of a formal proof of the above characterization of walks terminating at vertices without fixed points. Let $\mathcal{C} = (V, E, l)$ denote the Cayley graph of rule 45. It clearly holds that for all $R, S \in V$, $dom(R \circ S) \subseteq dom(R)$. Hence, if a vertex S is a successor of vertex R, it must hold that $dom(S) \subseteq dom(R)$.

We have that $dom(V_1) = \{00, 10, 11\}$. We can easily check that $dom(R_1) = \{00, 01, 10\}$ and that $dom(V_1) \nsubseteq dom(R_1)$. Therefore, V_1 cannot be a successor of R_1.

Further, we can notice that $dom(R_0 \circ R_0) \subsetneq dom(R_0) = dom(V_1)$ and therefore, V_1 cannot be a successor of $R_0 \circ R_0$.

We have that $V_1 = R_0 \circ R_1$ and $dom(V_1 \circ R_0) \subsetneq dom(V_1)$ and therefore, we can never reach V_1 by any walk passing through $V_1 \circ R_0$.

Therefore, the only way to reach V_1 from R_0 is to "continue from R_0 by an edge labeled as 1 to the vertex $R_0 \circ R_1 = V_1$". Further, the only way to reach V_1 from V_1 is to "continue from V_1 by an edge labeled as 1 to the vertex $V_1 \circ R_1 = R_0$". And lastly, V_1 can only be reach from the vertices $\mathbf{1}, R_0$, and V_1. From this, we can easily obtain that all the walks starting at $\mathbf{1}$, which terminate in V_1, correspond to configurations from the set $01(11)^*$. Similarly, we can approach the characterization of walks starting at $\mathbf{1}$, which terminate at V_2 or at V_3.

Analogously, we can describe all the configurations with exactly two preimages as the ones corresponding to walks terminating at either vertex W_1 - the terminal vertex of the walk given by 0101, which starts at $\mathbf{1}$, or W_2 - terminal vertex of the walk given by 1010.

$$W_1 = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \; W_2 = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

1. The walks terminating at W_1 either pass through V_1, or through the "green vertex" U. In the first case, the configurations corresponding to such walks are of the form $01(11)^*01(11)^*\Big(01(11)^*01(11)^*\Big)^*$. In the second case, we have configurations of the form $11(11)^*01(11)^*01(11)^*\Big(01(11)^*01(11)^*\Big)^*$.

2. The walks terminating at W_2 are the ones corresponding to configurations from the set $1(11)^*01(11)^*0\Big(1(11)^*01(11)^*0\Big)^*$.

Analogously, we obtain a second corollary.

Corollary 27. *The configurations, which have exactly two preimages under the ECA rule 45 operating on a finite cyclic grid, are exactly configurations, which consist of an even, nonzero number of single 0's, each two 0's separated by a block of an odd number of 1's.*

Furthermore, we can see that all configurations with exactly three preimages form the set $11(11)^*$. They are exactly those configurations consisting of a non-zero even number of 1's. Combining the results, we obtain Theorem 28.

Theorem 28. *ECA rule 45 operating on a cyclic grid of odd size is reversible.*

Proof. Examining the vertices in the Cayley graph of rule 45, we see that every configuration has either 0, 1, 2, or 3 preimages. We have given the characterization of all configurations having 0, 2, or 3 preimages, from which we can see that all such configurations are of even size. Therefore, if a configuration is of an odd size, it must have exactly one preimage. This implies that rule 45 operating on a cyclic grid of odd size is reversible. □

3.3.1 Rule 45 as a Random Mapping

In Chapter 2, we have seen that rule 45 possesses typical properties of a chaotic rule - its space time diagrams seem to form no obvious structures and its average transient length seems to grow exponentially for a grid of even size. In the literature, chaotic cellular automata are often connected to "random behavior". In fact, Stephen Wolfram has patented the idea of using the chaotic ECA rule 30 as a pseudorandom generator[1] [27]. In this subsection, we use the results about the phase-space properties of rule 45 that we obtained so far to compare the rule to a random mapping.

We define a random mapping space to be a triplet $(X, \mathcal{F}, P)$ where X is some finite set of cardinality $N \in \mathbb{N}^+$, $\mathcal{F}$ is the set of all functions $T : X \to X$ and P is the uniform probability distribution over $\mathcal{F}$. By $g : \mathcal{F} \to \mathbb{N}$ we denote the random variable determining the number of GOE configurations for a particular function. Formally, $g(F) = |\{u \in X \mid F^{-1}(u) = \emptyset\}|$ for $F \in \mathcal{F}$. The statistical properties of random mappings have been widely studied in the past. In [4], the following results have been obtained:

$$\mathbb{E}(g) \approx \frac{N}{e} \qquad \text{as } N \to \infty \tag{3.6}$$

We consider the random mapping space to be $(X, \mathcal{F}, P)$ where $X = \mathbf{2}^n$ for $n \in \mathbb{N}^+$. Then, for an ECA $(\mathbf{2}^n, F)$ we have that $F \in \mathcal{F}$ and $N = |X| = 2^n$. In such case, we have that $\mathbb{E}(g) \approx \frac{2^n}{e}$. The goal of this subsection is to compare the number of GOE configurations of a random mapping and of rule 45. Therefore, we need to establish the number of GOE configurations of length $n \in \mathbb{N}^+$ for this rule.

We have already established that all the garden of eden configurations of rule 45 have an even length and that they are exactly those configurations that contain an odd number of 0's, each two 0's separated by a region of an odd number of 1's.

Let us fix $n \in \mathbb{N}^+$ even, and let $u \in \mathbf{2}^n$ be a GOE configuration of rule 45. We fix $l \in \mathbb{N}^+$ to be the largest odd number such that $l \leq \frac{n}{2}$; l is, therefore, the largest number of 0's that u can contain. There must be some $i \in \mathbb{Z}_n$ such that $u_i = 0$. Then, for all j such that $u_j = 0$, it holds that j has the same parity as i. Therefore, each GOE configuration $v \in \mathbf{2}^n$ satisfies exactly one of the following two cases.

[1] hyperlink to the patent register: `https://patents.google.com/patent/US4691291A/en`

1. For every $i \in \mathbb{Z}_n$ $u_i = 0$ implies that i is even. In such a case, there are $\binom{\frac{n}{2}}{k}$ different GOE configurations containing k 0's on even positions. Therefore, the total number of such configurations is

$$\binom{\frac{n}{2}}{1} + \binom{\frac{n}{2}}{3} + \ldots + \binom{\frac{n}{2}}{l}.$$

2. For every $i \in \mathbb{Z}_n$ $u_i = 0$ implies that i is odd. In such a case, there are $\binom{\frac{n}{2}}{k}$ different GOE configurations containing k 0's on odd positions. Therefore, the total number of such configurations is again

$$\binom{\frac{n}{2}}{1} + \binom{\frac{n}{2}}{3} + \ldots + \binom{\frac{n}{2}}{l}.$$

To simplify the sum, we observe the following.

$$\binom{\frac{n}{2}}{0} + \binom{\frac{n}{2}}{1} + \binom{\frac{n}{2}}{2} + \ldots + \binom{\frac{n}{2}}{\frac{n}{2}} = 2^{\frac{n}{2}} \tag{3.7}$$

$$\binom{\frac{n}{2}}{0} - \binom{\frac{n}{2}}{1} + \ldots + (-1)^{\frac{n}{2}}\binom{\frac{n}{2}}{\frac{n}{2}} = (1 + (-1))^{\frac{n}{2}} = 0. \tag{3.8}$$

Substracting (3.8) from (3.7), we get that

$$\binom{\frac{n}{2}}{1} + \binom{\frac{n}{2}}{3} + \ldots + \binom{\frac{n}{2}}{l} = 2^{\frac{n}{2}-1}.$$

Therefore, for a grid of even size n, the total number of garden of eden configurations is $2 \cdot 2^{\frac{n}{2}-1} = 2^{\frac{n}{2}}$. As $2^{\frac{n}{2}} << \frac{2^n}{e}$, we can conclude that for a large grid of even size, the number of GOE configurations of rule 45 is much smaller than the expected value of the number of GOE configurations of a random mapping. Therefore, rule 45 does not have one of the fundamental statistical properties of a random mapping.

3.4 Cellular Automata as Topological Mappings

So far, we have studied the phase-space properties of 1D CAs operating on a cyclic grid of finite size. In the following, we will examine infinite garden of eden configurations. This section, however, shows that due to the simplicity of 1D CAs, the task in fact reduces to studying finite structures. To obtain this reduction, we introduce some basic facts from the field of symbolic dynamics. This section is based on the results presented in [10].

Let $A = \{a_1, \ldots, a_n\}$ be a finite set and consider the discrete topological space $(A, \mathcal{P}(A))$. We define $(A^{\mathbb{Z}}, \mathcal{T})$ to be the space equipped with product topology. A classical basis of $(A^{\mathbb{Z}}, \mathcal{T})$ is the system

$$\mathcal{B} = \{ \prod_{i=-\infty}^{\infty} A_i \,|\, A_i \subseteq A,\ A_i = A \text{ for all but finitely many } i\}.$$

As $(A, \mathcal{P}(A))$ is compact, we get from Tychonoff's theorem that $(A^{\mathbb{Z}}, \mathcal{T})$ is compact as well.

Definition 29. *Let A be a finite set, $u \in A^+$, and $z \in \mathbb{Z}$. A cylinder is a set of the form $[u]_z = \{x \in A^{\mathbb{Z}} \mid x_{z+i} = u_i \text{ for all } 0 \leq i < |u|\}$.*

Let $\pi_i : A^{\mathbb{Z}} \to A$ be the i-th projection, let $u \in A^+$, $u = u_0 u_1 \dots u_{n-1}$, and let $z \in \mathbb{Z}$. Clearly, $[u]_z = \bigcap_{i=0}^{n-1} \pi_{z+i}^{-1}(u_i)$ where each set $\pi_{z+i}^{-1}(u_i)$ is clopen. Hence, every cylinder is a finite intersection of clopen sets and is itself clopen. In the following Lemma 30, we show that the set of all cylinders forms a basis for $(A^{\mathbb{Z}}, \mathcal{T})$.

Lemma 30. *Let $\mathcal{C} = \{[u]_z \mid u \in A^+, z \in \mathbb{Z}\}$ be the set of all cylinders. Then, $\mathcal{C}$ forms a basis for $(A^{\mathbb{Z}}, \mathcal{T})$.*

Proof. Let $B \in \mathcal{B}$ and $x \in B$. We will show that there exists $C \in \mathcal{C}$ such that $C \subseteq B$ and $x \in C$. This will imply that $\mathcal{C}$ is a "finer system" than $\mathcal{B}$ and, therefore, a basis for $(A^{\mathbb{Z}}, \mathcal{T})$. Let $i_1, i_2, \dots, i_k \in \mathbb{Z}$, $i_1 < i_2 < \dots < i_k$ be such that $\pi_i(B) = A$ for all $i \in \mathbb{Z} \setminus \{i_1, i_2, \dots, i_k\}$. We set $u = x_{[i_1, i_k]}$. Then, the cylinder $[u]_{i_1}$ contains x and it clearly holds that $[u]_{i_1} \subseteq B$. □

Given a cellular automaton $(A^{\mathbb{Z}}, F)$, we will show that the mapping $F : A^{\mathbb{Z}} \to A^{\mathbb{Z}}$ is a continuous mapping of topological spaces.

Lemma 31. *Let $(A^{\mathbb{Z}}, F)$ be a cellular automaton with radius $r \in \mathbb{N}$ and a local rule $f : A^{2r+1} \to A$. Then, the mapping $F : A^{\mathbb{Z}} \to A^{\mathbb{Z}}$ is a continuous endomorphism of the topological space $(A^{\mathbb{Z}}, \mathcal{T})$.*

Proof. It suffices to show that $F^{-1}(C)$ is open for every cylinder $C \in \mathcal{C}$. Let $u \in A^+$ and $z \in \mathbb{Z}$. It is is straightforward that $F^{-1}([u]_z)$ is open because $F^{-1}([u]_z) = \bigcup_{v \in \widetilde{f}^{-1}(u)} [v]_{z-r}$ is a union of open sets. □

Let A be any finite set, $(A, \mathcal{P}(\mathcal{A}))$ the discrete topological space, and $(A^{\mathbb{Z}}, \mathcal{T})$ the space with product topology. We define the *shift operator* $\sigma : A^{\mathbb{Z}} \to A^{\mathbb{Z}}$ by $\sigma(u)_i = u_{i+1}$ for $u \in A^{\mathbb{Z}}$. In fact, there is a fundamental result first stated by Hedlund [6], which characterizes cellular automata to be exactly those continuous endomorphisms of $(A^{\mathbb{Z}}, \mathcal{T})$, which commute with the shift operator, i.e. $F(\sigma(u)) = \sigma(F(u))$.

We can finally prove Theorem 32, which explains why we only need to examine finite patterns when studying the garden of eden configurations of a CA operating on an infinite grid.

Theorem 32. *Let A be a finite set, $(A^{\mathbb{Z}}, F)$ a cellular automaton operating on an infinite grid, and let f be its local rule with radius r. A configuration $u \in A^{\mathbb{Z}}$ is a garden of eden configuration if and only if it contains an orphan.*

Proof. Suppose $a \in A^{\mathbb{Z}}$ is not gargen of eden. Then there exists $v \in A^{\mathbb{Z}}$ such that $F(v) = u$. Let w be any finite subword of u, $w = u_{[i,j]}$ for some $i, j \in \mathbb{Z}$. Then $v_{[i-r, j+r]}$ is a preimage of w under $\widetilde{f}$. Therefore, u contains no orphan.

Conversely, suppose u contains no orphan. Then, $u_{[-n,n]}$ has a preimage under $\widetilde{f}$ for all $n \in \mathbb{N}$. Therefore, each $F^{-1}(u_{[-n,n]}) = \bigcup\{[v]_{-n-r} \mid v \in \widetilde{f}^{-1}(u_{[-n,n]})\}$ is a nonempty closed set. From compactness of the space $(A^{\mathbb{Z}}, \mathcal{T})$, the intersection $\bigcap_{n \in N} F^{-1}(u_{[-n,n]})$ is nonempty and contains some $x \in A^{\mathbb{Z}}$, which is a preimage of u. □

Combining Theorem 32 and Lemma 13 we get that in order to describe all garden of eden configurations of an ECA operating on an infinite tape, it is sufficient to characterize all words $u \in \mathbf{2}^+$, $u = u_0u_1 \dots u_{n-1}$ for some $n \in \mathbb{N}^+$, such that the relation $R_{u_0} \circ R_{u_1} \circ R_{u_2} \circ \dots \circ R_{u_{n-1}}$ is empty.

We say that a CA $(A^{\mathbb{Z}}, F)$ is surjective, if F is a surjective mapping. As a corollary, using the Cayley diagrams, we get a very simple proof of the known fact about the decidability of surjectivity for 1D CAs.

Corollary 33. *It is decidable whether a one-dimensional cellular automaton is surjective.*

Proof. For a CA with local rule $f : A^{2r+1} \to A$, it suffices to construct its Cayley graph whose size is bounded by $2^{|A|^{4r}}$. Then, the CA is surjective if and only if the Cayley graph does not contain a vertex corresponding to the empty relation. This can be easily checked. □

It is interesting how various problems get significantly more difficult when considering CAs in higher dimensions. As an example, Kari [8] has shown that the problem of surjectivity is undecidable for 2D CAs.

3.5 Rule 110

Rule 110 is of particular interest to researchers as it is so far the only ECA proven to be Turing complete and, therefore, undoubtedly regarded as complex. This makes this rule subject to various studies of its properties. Even though researchers often give examples of garden of eden configurations of rule 110 operating on either cyclic or infinite grid, so far, we have not seen a complete characterization of all garden of eden configurations of this rule. In this section, we present such characterization in case the rule operates on an infinite grid. The characterization of "cyclic GOE configurations" can be obtained analogously.

For the rest of this section, we fix f to be the local rule with Wolfram number 110.

Observation 34. *Let $u, v \in \mathbf{2}^*$ be such that $v \sqsubseteq u$. Then, if v is an orphan of $\widetilde{f}$, u is also its orphan.*

Proof. Let us write $u = rvs$ for some $r, s \in \mathbf{2}^*$. Then $R_u = R_r \circ R_v \circ R_s$. Since $R_v = \emptyset$ we obtain that $R_u = R_r \circ \emptyset \circ R_s = \emptyset$. Hence, u is an orphan. □

In the following text, we will construct a language $G \subseteq \mathbf{2}^*$, which consists of all "canonical" orphans of $\widetilde{f}$. By "canonical", we mean that:

- Any orphan of $\widetilde{f}$ contains some $u \in G$ as a subword.
- Each $u \in G$ contains no other $v \in G$ as a subword.

Using Observation 34, we get that orphans are exactly those words $u \in \mathbf{2}^*$, which contain some $v \in G$ as a subword. And further, using Theorem 32, we get that all garden of eden configurations of rule 110 operating on an infinite grid are exactly the words $u \in \mathbf{2}^{\mathbb{Z}}$, which contain some $v \in G$ as a subword. In the next subsection, we construct the language G.

3.5.1 Construction of Canonical Orphans

The local rule f with Wolfram number 110 is given by the following table.

u	111	110	101	100	011	010	001	000
$f(u)$	0	1	1	0	1	1	1	0

For the rest of this section, we fix R_0, R_1 to be the relations R_0^f, R_1^f. They have the following form:

$$R_0 = \{(00,00),(10,00),(11,11)\}$$
$$R_1 = \{(00,01),(01,10),(01,11),(10,01),(11,10)\}.$$

We will examine the powers of both relations. Clearly

$$R_0^2 = \{(00,00),(10,00),(11,11)\} = R_0. \tag{3.9}$$

For R_1 the situation is not as simple:

$$\begin{aligned}
R_1^2 &= \{(00,10),(00,11),(01,01),(01,10),(10,10),(10,11),(11,01)\}\\
R_1^3 &= \{(00,01),(00,10),(01,01),(01,10),(01,11),(10,01),(10,10),(11,10),\\
&\quad (11,11)\}\\
R_1^4 &= \{(00,01),(00,10),(00,11),(01,01),(01,10),(01,11),(10,01),(10,10),\\
&\quad (10,01),(11,01),(11,10)\}\\
R_1^5 &= \{(00,01),(00,10),(00,11),(01,01),(01,10),(01,11),(10,01),(10,10),\\
&\quad (10,01),(11,01),(11,10),(11,11)\}.
\end{aligned}$$

We see that $R_1^5 = C := \{00,01,10,11\} \times \{01,10,11\}$. C contains all pairs from $\mathbf{2}^2 \times \mathbf{2}^2$ except the elements of the form $(ab,00)$, $a,b \in \mathbf{2}$.

Lemma 35. *$R_1^n = C$ for all $n \in \{5,6,7,\ldots\}$.*

Proof. We will prove this by induction on n. For $n = 5$ the statement clearly holds. Suppose that $n > 5$ and that $R_1^{n-1} = C$. It suffices to show that $R_1 \circ C = C$. Let $a \in \{00,01,10,11\}$ and $c \in \{01,10,11\}$. Then, there exists $b \in \{00,01,10,11\}$ such that $(a,b) \in R_1$. As $c \neq 00$ it holds that $(b,c) \in C$. Therefore $(a,c) \in R_1 \circ C$ and $R_1 \circ C = C$. □

Definition 36. *We define a mapping $P : \mathbf{2}^* \to \mathbf{2}^*$, which maps each word u to the word constructed from u by substituting all subwords in u of the form 1^k, $k \leq 6$ by 1^5 and all subwords 0^l, $l \geq 2$ by 0. We say that $P(u)$ is a reduced word.*

A word $w \in \mathbf{2}^*$ is reduced if and only if it contains no blocks of the form 1^6 or 0^2. From the equality (3.9) and Lemma 35, we have that $R_u = R_{P(u)}$ for all $u \in \mathbf{2}^*$. Therefore, when searching for orphans, it suffices to examine reduced words.

Lemma 37. *$R_{0(1110)^k10} = R_{010}$ for any $k \in \mathbb{N}$.*

Proof. It suffices to prove that $R_{0111010} = R_{010}$ and use induction on k. This equality is easily verified by computing that both sides are equal to $\{(11,00)\}$. □

Using the previous Lemma 37, we can notice that $R_{10(1110)^k10} = \{(01, 00)\}$ and since $R_0 = \{(00, 00), (10, 00), (11, 11)\}$ we see that

$$R_{010(1110)^k10} = R_0 \circ R_{10(1110)^k10} = \emptyset \text{ for any } k \in \mathbb{N}.$$

Therefore, any reduced word $u \in \mathbf{2}^*$ containing a subword of the form $010(1110)^k10$ for some $k \in \mathbb{N}$ is an orphan. In the following Theorem 38, we prove the converse.

Theorem 38. *If a reduced word $u \in \mathbf{2}^*$ does not contain a subword of the form $01(0111)^k010$ for some $k \in \mathbb{N}$, then it is not an orphan of $\widetilde{f}$.*

Proof. Let $u \in \mathbf{2}^*$ be a reduced word satisfying the assumption. We want to show that $R_u \neq \emptyset$. Without loss of generality we can assume that u is of the form $1v$ for some $v \in \mathbf{2}^*$. In case u is not of this form, we consider the word $1u$ and notice that this word still satisfies the assumption. In such a case, we will prove that $R_{1u} \neq \emptyset$. From Lemma 34 it will follow that $R_u \neq \emptyset$.

It can easily be shown that if u is of the form 1^n for any $n \in \mathbb{N}^+$ then $R_u \neq \emptyset$. Therefore, we can assume the reduced word u contains at least one 0 and, therefore, is of the form

$$1^{k_1}01^{k_2}0 \dots 1^{k_n}01^e \text{ where n} \in \mathbb{N}^+, \ 1 \leq k_i \leq 5 \text{ for all } i, \text{ and } 0 \leq e \leq 5.$$

Further, we will write u in the form $u = b_1b_2 \dots b_m1^e$ where each block satisfies exactly one of the following cases. Informally, case Case 1 deals with all the occurrences of the block 010 in u and case Case 2 deals with the rest.

Case 1 $b = 1^l0(1110)^k10$ where $k \in \mathbb{N}$ is largest possible (therefore, it cannot happen that $l = 3$); and $l \in \{1, 2, 3, 4, 5\}$ denotes the maximum number of 1's, which form a prefix of the block $0(1110)^k10$ in u.

Case 2 $b = 1^k0$ and b is not a subword of any block from case Case 1; either b is a prefix of u and $b = 1^k0$, $k \geq 1$, or b is not a prefix of u and $k \geq 2$ is maximum possible (in such a case, u must contain the block $0b$ as a subword).

Below, we show that the form $u = b_1 \dots b_m1^e$ can always be constructed. We write u in the following form and describe the construction by induction on n.

$$u = \underbrace{1^{k_1}0}_{c_1}\underbrace{1^{k_2}0}_{c_2} \dots \underbrace{1^{k_n}0}_{c_n}1^e$$

If $n = 1$ then we put $b_1 = c_1$, which is a block from Case 2, and we are done.

Suppose that $n > 1$ and that any reduced word $1^{l_1}01^{l_2}0 \dots 1^{l_{n'}}01^{e'}$, $n' < n$, can be written in the desired form.

If $c_n = 10$ then there exists a k such that the block $c_kc_{k+1} \dots c_n$ is a block from Case 1. Hence, we put $b_m = c_kc_{k+1} \dots c_n$ and we get that $u = c_1c_2 \dots c_{n'}b_m1^e$ where $n' = k - 1$. If $c_n \neq 10$ then c_n is a block from Case 2 and we put $u = c_1 \dots c_{n'}b_m1^e$ where $n' = n-1$. Hence, we have constructed $u = c_1 \dots c_{n'}b_m1^e$ where $n' < n$ and we use the induction assumption on the word $c_1 \dots c_{n'}$.

As we have constructed the form $u = b_1 \dots b_m1^e$, we can examine the relations R_{b_i} for each block b_i in u. The main goal is to show that whenever the block b_i is not a prefix of u then $(00, 00) \in R_{b_i}$. We study all the possible cases below.

Case A Let $b = b_i$, $i \geq 2$, be a block, which is not a prefix of u. We show that $(00, 00) \in R_b$.

Case A1 We assume b is not a prefix and it is of the form $b = 1^l0(1110)^k10$ for some $k \in \mathbb{N}$, $l \in \{1, 2, 3, 4, 5\}$. In such a case, $l \neq 3$ because k is the maximum number of blocks 1110 contained in b. Moreover $l \neq 1$ otherwise b would be an orphan and u would violate the assumption. We already have that $R_{0(1110)^k10} = \{(11, 00)\}$. For $l \in \{2, 4, 5\}$ it holds that $(00, 11) \in R_1^l$. Therefore, $(00, 00) \in R_b$.

Case A2 We assume b is not a prefix and it is of the form $b = 1^k0$ where $k \geq 2$. For $k \geq 2$ it holds that $(00, 10) \in R_1^k$ and since $(10, 00) \in R_0$ we obtain that $(00, 00) \in R_b$.

Case B The remaining case we need to examine is when $b = b_1$ is the prefix of u. Our goal is to show that then there exist c, $d \in \mathbf{2}$ such that $(cd, 00) \in R_b$.

Case B1 Suppose b is a prefix of the form $b = 1^l0(1110)^k10$, $k \in \mathbb{N}$, $l \in \{1, 2, 3, 4, 5\}$.

- if $l \in \{2, 4, 5\}$ we already know that $(00, 00) \in R_b$ from Case A1
- if $l \in \{1, 3\}$ then it holds that $(01, 11) \in R_1^l$ and since $(11, 00) \in R_{(0111)^k010}$ we get that $(01, 00) \in R_b$.

Case B2 We are left with the case when b is a prefix of u, $b = 1^k0$, $k \geq 1$. Then for each k there exist $c, d \in \mathbf{2}$ such that $(cd, 10) \in R_1^k$. Since $(10, 00) \in R_0$, we have that $(cd, 00) \in R_b$.

To summarize, for $u = b_1b_2 \dots b_m1^e$ as defined above, we have that there exist c, $d \in \mathbf{2}$ such that $(cd, 00) \in R_{b_1b_2\dots b_m}$.

For $e \in \{1, 2, 3, 4, 5\}$, there exist g, $h \in \mathbf{2}$ such that $(00, gh) \in R_1^e$.

Altogether, if $e = 0$ then $R_u = R_{b_1} \circ R_{b_2} \circ \dots \circ R_{b_m} \ni (cd, 00)$ for some c, $d \in \mathbf{2}$. If $e \in \{1, 2, 3, 4, 5\}$ then $R_u = R_{b_1} \circ R_{b_2} \circ \dots \circ R_{b_m} \circ R_1^e \ni (cd, gh)$ for some c, d, g, $h \in \mathbf{2}$. Therefore, $R_u \neq \emptyset$. □

Finally, we can describe all canonical orphans in the following Theroem 39.

Theorem 39. *An infinite word $u \in \mathbf{2}^{\mathbb{Z}}$ is a garden of eden configuration of rule 110 if and only if it contains a subword from the set*

$$\begin{aligned} G &= \{010^{k_1}1110^{k_2} \dots 0^{k_{n-1}}1110^{k_n}10 \mid n \in \mathbb{N},\ k_i \in \mathbb{N}^+ \text{ for all } i \in \{1, 2, \dots, n\}\} \\ &= 01\left(0^+111\right)^*0^+10. \end{aligned}$$

Proof. We know that a reduced word is an orphan of $\widetilde{f}$ if and only if it contains a subword of the form $010(1110)^k10$ for some $k \in \mathbb{N}$. Now it suffices to compute $P^{-1}\{010(1110)^k10 \mid k \in \mathbb{N}\} = 0^+10^+\left(1110^+\right)^*10^+$. As the multiple 0's forming

the prefix and suffix of words in the set are unnecessary we finally define the canonical language

$$G = 01\Big(0^+111\Big)^*0^+10.$$

Clearly for any two words in G it holds that one is not a subword of the other.

Let $u \in \mathbf{2}^{\mathbb{Z}}$ be such that u contains some $v \in G$ as a subword. Then, $R_v = R_{P(v)} = \emptyset$. Therefore, v is an orphan and u is a garden of eden configuration of rule 110.

Conversely, let $u \in \mathbf{2}^{\mathbb{Z}}$ be a GOE configuration of rule 110. Let $v \in \mathbf{2}^*$ be an orphan of $\widetilde{f}$ contained in u. Then, the reduced word $P(v)$ is also an orphan and from Theorem 38 it must contain a subword of the form $01(0111)^k010$ for some $k \in \mathbb{N}$. Then, v must contain a subword from the set $P^{-1}\{01(0111)^k010 \,|\, k \in \mathbb{N}\} = G$. Therefore, u contains a subword from the set G. □

3.6 Discussion

We presented a novel representation of 1D CAs called the Cayley graphs of cellular automata, which are related to the backward dynamics of the system. We used the Cayley graphs to obtain the characterization of garden of eden configuration for the ECA rule 45 operating on a cyclic grid to demonstrate its usefulness. We used a slightly different approach to characterize all garden of eden configurations of the Turing complete rule 110 operating on an infinite grid. Due to the results of Kari [8] we know that such characterization of GOE configurations is impossible for CAs in higher dimensions as the problem of surjectivity is undecidable for them.

Conclusion

The model of cellular automata has been intensively studied by many researchers in the past due to their intriguing complex behavior. This makes it surprising that there is still so much to explore about it. We have demonstrated that by proposing a novel classification of CA dynamics, which corresponds to Wolfram's classification surprisingly well.

We believe the ultimate goal to understand complex behavior of various dynamical systems is to propose suitable formal definitions of fundamental notions in complex systems theory. We hope that the Transient classification, as well as our studies related to CAs backward dynamics, can help us understand such notions better.

Bibliography

[1] S. Brenner. *My Life in Science*. BioMed Central Ltd, 2001.

[2] H. Cisneros, J. Sivic, and T. Mikolov. Evolving structures in complex systems. 2019.

[3] M. Cook. Universality in elementary cellular automata. *Complex Systems*, 15, 01 2004.

[4] P. Flajolet and A. Odlyzko. Random Mapping Statistics. *Advances in Cryptology — EUROCRYPT '89*, 1990.

[5] M. Gardener. Mathematical games: the fantastic combinations of john conway's new solitaire game "life. 1970.

[6] G. A. Hedlund. Endomorphisms and automorphisms of the shift dynamical system. *Mathematical systems theory*, 3:320–375, 1969.

[7] K. Kaneko. Complexity in basin structures and information processing by the transition among attractors. 01 1985.

[8] J. J. Kari. Reversibility and surjectivity problems of cellular automata. *Journal of Computer and System Sciences*, 48:149 – 182, 1991.

[9] J. J. Kari. *Basic Concepts of Cellular Automata*, pages 3–24. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[10] P. Kůrka. *Topological and Symbolic Dynamics*. Société mathématique de France, 2003.

[11] C. G. Langton. Self-reproduction in cellular automata. *Physica D: Nonlinear Phenomena*, 10(1):135 – 144, 1984.

[12] C. G. Langton. Studying artificial life with cellular automata. *Physica D: Nonlinear Phenomena*, 1986.

[13] O. Martin, A. Odlyzko, and S. Wolfram. Algebraic properties of cellular automata. *Communications in Mathematical Physics*, 93, 06 1984. doi: 10.1007/BF01223745.

[14] T. Mikolov, A. Joulin, and M. Baroni. A roadmap towards machine intelligence. *ArXiv*, abs/1511.08130, 2015.

[15] M. Mitchell. Computation in cellular automata: A selected review. 1996.

[16] M. Mitchell, J. Crutchfield, and R. Das. Evolving cellular automata with genetic algorithms: A review of recent work. *First Int. Conf. on Evolutionary Computation and Its Applications*, 1, 05 2000.

[17] J. V. Neumann and A. W. Burks. *Theory of Self-Reproducing Automata*. University of Illinois Press, USA, 1966.

[18] C. Ofria and C. Wilke. Avida: A software platform for research in computational evolutionary biology. *Artificial Life*, 10(2):191–229, 2004.

[19] A. B. Owen. *Monte Carlo theory, methods and examples.* 2013.

[20] T. S. Ray. An approach to the synthesis of life. 1991.

[21] J. Reggia, S. Armentrout, H. Chou, and Y. Peng. Simple systems that exhibit self-directed replication. *Science (New York, N.Y.)*, 259:1282–7, 03 1993.

[22] C. Saclay and H. Gutowitz. Transients, cycles, and complexity in cellular automata. *Physical Review A*, 44, 12 1994.

[23] L. Soros and K. Stanley. Identifying necessary conditions for open-ended evolution through the artificial life world of chromaria. *Artificial Life Conference Proceedings*, (26):793–800, 2014.

[24] S. Stepney. *Nonclassical Computation — A Dynamical Systems Perspective*, pages 1979–2025. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[25] S. H. Strogatz. *Nonlinear Dynamics and Chaos.* CRC Press, 2000.

[26] S. Wolfram. Statistical mechanics of cellular automata. *Rev. Mod. Phys.*, 55:601–644, 1983.

[27] S. Wolfram. Random sequence generation by cellular automata. *Advances in Applied Mathematics*, 7(2):123 – 169, 1986.

[28] S. Wolfram. *A New Kind of Science.* Wolfram Media, 2002.

[29] A. Wuensche. Exploring discrete dynamics. The DDLab manual. 2019.

[30] A. Wuensche and M. Lesser. The global dynamics of celullar automata: An atlas of basin of attraction fields of one-dimensional cellular automata. *J. Artificial Societies and Social Simulation*, 4, 01 2001.

[31] H. Zenil. Compression-based investigation of the dynamical properties of cellular automata and other systems. *Computing Research Repository - CORR*, 19, 10 2009.