FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

DOCTORAL THESIS

Barbora Hudcová

Complexity and Computational Capacity of Discrete Dynamical Systems

Department of Algebra

Supervisor of the doctoral thesis: Tomáš Mikolov, Ph.D.
Study programme: Mathematics
Study branch: Algebra, number theory, and mathematical logic

Prague 2024

I declare that I carried out this doctoral 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

My gratitude extends to many individuals I crossed paths with during my doctoral journey. I thank Tomáš Mikolov for suggesting this fascinating research topic to me and for mentoring me during the whole process. My special thanks belong to Jiří Tůma for his mathematical guidance during our numerous consultations. Thank you to Lenka Zdeborová and Stefano Nichele for welcoming me to their teams and supervising me during my internships abroad. Thank you to David Stanovský for his useful advice and encouragement to apply for the START grant.

I am very grateful to Freya Behrens, Jakub Krásenský, and Trym Lindell. Collaboration with them was an inspiring process that fuelled me with enthusiasm for research. Thank you to my great colleagues Hugo Cisneros, Jelle Piepenbrock, Teven LeScao, Kateryna Zorina, David Herel, Tom Glover, and Sidney Pontes-Filho. Last but certainly not least, my warmest thanks go to my friends and family who have been my unwavering pillars of support and comfort.

Title: Complexity and Computational Capacity of Discrete Dynamical Systems

Author: Barbora Hudcová

Department: Department of Algebra

Supervisor: Tomáš Mikolov, Ph.D., Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University

Advisor: doc. RNDr. Jiří Tůma, DrSc., Department of Algebra

Abstract: The central aim of this thesis is to study the concepts of “complexity” and “computational capacity” of discrete dynamical systems and to connect them to rigorously measurable properties. In the first part of the thesis, we propose a formal metric of a discrete system's complexity based on the numerical estimates of its asymptotic convergence time. We identify a critical region of systems corresponding to a phase transition from an ordered to a chaotic phase. Additionally, we complement this work by studying dynamical phase transitions of discrete systems analytically, using newly developed tools from statistical physics. Specifically, for a fixed discrete system, we demonstrate that varying its initial configurations can result in abrupt changes in the system's behaviour; and we describe exact positions of such transitions. The second part of this thesis is dedicated to analysing computational capacity of cellular automata via the notion of their relative simulation. Informally, we say that automaton $\mathcal{B}$ can simulate $\mathcal{A}$ if $\mathcal{B}$ can effectively reproduce any dynamics of $\mathcal{A}$. We introduce a specific notion of automata simulation and formalize it in algebraic language. This allowed us to answer open questions about the computational capacity of cellular automata using well-established algebraic results. Namely, we prove that certain classes of affine automata are very limited in terms of what they can simulate. Further, we characterize the simulation capacities of any canonical additive automaton with radius one.

Keywords: discrete dynamical systems cellular automata dynamical phase transitions computational capacity

Contents

1. Introduction

1.1 General Introduction

The field of complex systems is a fascinating area of research. Due to its broad scope and interdisciplinary nature, describing it concisely while doing it justice poses a challenge. In essence, a complex system consists of a large number of particles that interact locally and undergo spontaneous self-organisation into higher-level structures that evolve and behave qualitatively differently from their components. Remarkably, the self-organization occurs without a centralized controller governing the behaviour of the particles [153]. Examples of such a phenomenon are omnipresent and include the growth and organisation of cells during an organism's developmental phase, ant colonies, the organisation of birds who locally adapt to their neighbours and form flocks, or the interactions of individual people and markets that culminate in a global economy [104]. It is noteworthy that the field of complex systems is relatively nascent, with its inception often marked by the establishment of the Santa Fe Institute in 1984. This underscores the dynamic and evolving nature of this research area.

Abstract models exhibiting such complex behaviour include cellular automata, random Boolean networks, neural automata, and recurrent neural networks. Despite the simplicity of the underlying update rules of such systems, their iterative application often leads to intriguing behaviour where structures emerge and interact with each other in ways that are hard to predict. Consequently, there is a prevailing belief that such systems hold significant potential as models for artificial evolution – a process where structures emerge and grow in complexity in an open-ended manner. Moreover, there is a plethora of work exploring such systems as alternative models of efficient computation due to their massively parallel nature. These are just a few examples to motivate the theoretical studies of such models.

With the rising efficiency of computers, it has become popular to explore such models simply by running their simulations and observing the visualisations of their dynamics. Based on such observations, prominent figures in the field boldly asserted that models with "complex behaviour" exist at a critical phase transition between ordered and chaotic systems [84] and suggested that all sufficiently "complex systems" possess the capacity for universal computation [158]. There have been impressive results that support such claims; as an example, by giving elaborate proofs that certain cellular automata with intriguing visualisations can indeed simulate a computationally universal system [18, 27]. However, it yet remains a great challenge to rigorously prove such claims in their full generality. One of the reasons is that, due to the novelty and broad scope of the field, there is no consensus on what should be the formal definitions of notions such as "complexity", "emergence" or "computational capacity" [126]. This makes it especially challenging when proving negative results.

The central aim of this thesis is to delve into the concepts of "complexity" and "computational capacity" of discrete dynamical systems and to connect them to rigorously measurable properties. This overarching goal naturally divides the thesis into two distinct parts that we describe below in more detail.

In Part I, our focus is on investigating the complexity of discrete systems through the analysis of their dynamical properties. In Chapter 2, we propose a new measure of complexity for discrete systems that we call the *transient classification.* It is based on numerical estimates of the system's global dynamics. The method allows us to formally assess whether a given discrete dynamical system belongs to the "ordered", "chaotic", or "complex" regime. Compared to previous approaches, the method does not depend on any arbitrary choice of parameters, which makes the results robust and marks the method's significance. Expanding and complementing our initial work, we delve further into the exploration of global dynamics of discrete systems in Chapter 3 containing the paper [16]. This paper represents a collaborative effort with Freya

Behrens under the guidance of Lenka Zdeborová. Here, we analytically derive new results pertaining to the global dynamics of specific discrete systems. For that, we use sophisticated tools from statistical physics that were recently developed by Freya Behrens and Lenka Zdeborová exactly to enable answering such questions. Part I consists of two papers, each presented as an individual chapter:

[67] B. Hudcová, T. Mikolov, *Classification of Discrete Dynamical Systems Based on Transients*, Artificial Life, 27 (3-4), MIT Press, 220–245 (2022).

[16] F. Behrens, B. Hudcová, L. Zdeborová, *Dynamical Phase Transitions in Graph Cellular Automata*, submitted (2023). Available at arXiv:2310.15894.

Part II of this thesis is dedicated to analysing computational capacity of discrete systems, specifically focusing on cellular automata. We argue that a natural way to assess this is via the notion of cellular automata relative simulation. Informally, we say that automaton $\mathcal{B}$ can simulate $\mathcal{A}$ if $\mathcal{B}$ can effectively reproduce any dynamics of $\mathcal{A}$. Measuring the computational capacity of a given automaton then translates to assessing how many other automata can it simulate. To address this, we introduce a specific notion of automata simulation and formalize it in the algebraic language. This algebraic formalization is an important contribution of the thesis, as it allows us to answer open questions about the computational capacity of cellular automata using well-established algebraic results. To showcase this, in Chapter 4 containing our paper [66] we prove that certain classes of affine automata are very limited in terms of what they can simulate. This is a negative result that widely surpasses previous works in its generality. We further expand the results about affine automata in Chapter 5. Part II consists of two chapters, first of them contains the paper:

[66] B. Hudcová, J. Krásenský, *Simulation Limitations of Affine Cellular Automata*, submitted (2023). Available at arXiv:2311.14477.

The second chapter contains an original text that is yet to be extended into a paper.

An integral contribution of this thesis lies in exploring novel relationships between complex systems and other well-established fields. Concretely, we explore new links to particular methods from statistical physics in Part I and to universal algebra in Part II. The thesis showcases that both fields provide powerful tools for deriving new results about discrete dynamical systems. It remains an exciting question how far could such results be pushed in the future.

1.1.1 Contribution of the Authors

The texts of the papers included in this thesis were left mostly unchanged from the versions accepted (or submitted) to journals. Due to this, the notations can vary from chapter to chapter. Moreover, each paper contains its own introduction that does not reflect the developments in the field after the papers' publication. Below, we describe in detail for each of the papers the contribution of the authors.

The paper [67] was the work of Barbora Hudcová. Her supervisor Tomáš Mikolov is stated as a co-author, as this is the standard procedure in the field of computer science. The paper [16] is a result of a collaboration with Lenka Zdeborová's team during the present author's internship at EPFL, Lausanne. Whereas the paper [15] which develops new statistical physics tools to analyse dynamical systems was solely the work of Freya Behrens and Lenka Zdeborová with just a minor contribution of the present author; the subsequent paper [16] included in the thesis shares approximately equal contributions of both Freya Behrens and Barbora Hudcová. Freya Behrens provided fundamental insight from the field of statistical physics and derived the results of statistical physics methods. Barbora Hudcová provided the context and motivation from the field of complex systems and obtained the numerical results. Together, they chose the systems under study and identified their dynamical phases. Lastly, the paper [66] is a result of collaboration of the present author (around two thirds) with Jakub Krásenský (one third). Barbora Hudcová formalized the simulation notion into algebraic language, proved the elementary properties of the simulation, and wrote the content of the paper. Together with Jakub Krásenský they derived the results about limitations of affine automata, while Jakub Krásenský also provided the main author with insightful feedback on the first version of the manuscript. The last chapter with yet unpublished results is the work of Barbora Hudcová, with minor contributions from Jakub Krásenský.

In the rest of this introduction, we present some elementary definitions regarding complex systems and cellular automata in particular. This is followed by a brief history of the research on cellular automata. Subsequently, we concisely summarize the results from each chapter of the thesis while omitting most technical details. We complement each chapter's results by first providing a short overview of previous work to give some historical context and motivation, as well as to demonstrate the sheer breadth of work examining cellular automata and other dynamical systems.

1.2 Elementary Definitions

Though this thesis studies the dynamics of a variety of systems, the main focus remains on cellular automata due to both their popularity and simplicity of architecture. Informally, a Cellular Automaton (CA) can be perceived as a d-dimensional grid consisting of identical finite-state automata. They are all updated synchronously in discrete time steps based on an identical update function depending only on the states of automata in their fixed local neighbourhood. We give the formal definitions below; we note this section is an updated version of the original text in the present author's diploma thesis [65]. A great introduction to cellular automata can be found in [76].

Definition 1 (Cellular automaton)**.** Let $d \in \mathbb{N}$. We call $\mathbb{Z}^d$ the *d-dimensional cellular grid* and we say its elements are *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 call $S^{\mathbb{Z}^d}$ the *configuration space*.

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

Let $f : S^k \to S$ be a function. A d-dimensional *cellular automaton operating on an infinite grid* with neighbourhood N and *local rule* f is a dynamical system $\mathcal{A} = \left(S^{\mathbb{Z}^d}, F\right)$ where $F : S^{\mathbb{Z}^d} \to S^{\mathbb{Z}^d}$ is defined as:

$$F\left(c\right)\left(\mathbf{z}\right) = f\left(c(\mathbf{z} + \mathbf{n}_1), c(\mathbf{z} + \mathbf{n}_2), \dots, c(\mathbf{z} + \mathbf{n}_k)\right) \text{ for all } \mathbf{z} \in \mathbb{Z}^d,\, c \in S^{\mathbb{Z}^d}. \tag{1.1}$$

We call F the *global rule* of $\mathcal{A}$. We call the algebra $\mathbb{A} = (S, f)$ the *local algebra of $\mathcal{A}$* and we note that the local algebra, together with the dimension d and neighbourhood N, fully determines $\mathcal{A}$.

Below, we introduce some more terminology regarding the dynamics of cellular automata.

Definition 2 (Dynamics of Cellular Automata)**.** Let $d, k \in \mathbb{N}$, S be a finite set and $f : S^k \to S$ a function. Let $\mathcal{A} = \left(S^{\mathbb{Z}^d}, F\right)$ be a d-dimensional CA operating on an infinite grid with some neighbourhood $N \subseteq \left(\mathbb{Z}^d\right)^k$ and local rule f. For $c \in S^{\mathbb{Z}^d}$ and $t \in \mathbb{N}$, we define the *trajectory of the initial configuration c* as:

$$\left(c, F(c), F^2(c), F^3(c), \dots\right).$$

The process of obtaining a trajectory is often called a *simulation* or an *evolution* of the CA. Given the first t elements of a trajectory, $t \in \mathbb{N}$, we define its *space-time diagram* to be a matrix whose i-th (infinite) row is exactly $F^i(u)$, $0 \leq i \leq t$, where $F^0(c) = c$. Studying the *dynamics of a cellular automaton* simply means studying any property related to its global rule, which is iterated on different initial configurations. We define the *phase-space* of $\mathcal{A}$ to be a directed graph with nodes $S^{\mathbb{Z}^d}$ and edges $\{(c, F(c)) \mid c \in S^{\mathbb{Z}^d}\}$.

Typically, the most studied Cellular Automata (CAs) are in dimension 1 and 2. These are the cases we focus on in this thesis as well and thus, in what follows, we discuss more terminology regarding them.

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 neighbourhood of some CA. If we put $r = \max\{|n_i| \mid i \in \{1, 2, \ldots, k\}\}$, we can notice that N can be embedded into a larger *symmetric neighbourhood*, which is of the form $(-r, -r+1, \ldots, r-1, r)$. In such a case, we call r the *radius* of the symmetric neighbourhood.

For any 1D CA with neighbourhood $N = (n_1, \ldots, n_k)$ and local rule $f : S^k \to S$, we can consider an analogous automaton with $N' = (-r, \ldots, r)$, $r = \max\{|n_i| \mid i \in \{1, 2, \ldots, k\}\}$, 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 neighbourhood.

Let $S^{\mathbb{Z}}$ be the configuration space of some 1D CA and let $c \in S^{\mathbb{Z}}$. We typically write c_i instead of $c(i)$ for $i \in \mathbb{Z}$. In such a case, (1.1) simplifies into:

$$F(c)_i = f\left(c_{i-r}, c_{i-r+1}, \ldots, c_{i+r-1}, c_{i+r}\right) \text{ for all } i \in \mathbb{Z}, c \in S^{\mathbb{Z}}. \tag{1.2}$$

For practical purposes, when simulating a CA and depicting the space-time diagrams, typically only “finite configurations” of the CA are considered. There are multiple ways to reduce a CA to a finite grid, and we define one of the most classical ones below.

Definition 3 (1D CA operating on a cyclic grid)**.** Let $r, n \in \mathbb{N}$, S be a finite set and $f : S^{2r+1} \to S$. A *1D cellular automaton operating on a finite cyclic grid of size n* with neighbourhood r and *local rule f* is a dynamical system (S^n, F) where $F : S^n \to S^n$ is defined as in (1.2) but all indices are computed modulo n.

Cellular automata naturally split into families determined by their dimensionality, neighbourhood and state set. One of the smallest of such families are the *Elementary Cellular Automata (ECAs)* which are 1D CAs with states $\mathbf{2} \coloneqq \{0, 1\}$ and radius $r = 1$. They have received a lot of attention in the literature because despite the simple description of their local rules, ECAs contain automata with intriguing dynamics.

Each ECA is given by a Boolean function $f : \mathbf{2}^3 \to \mathbf{2}$. Hence, there are only 256 of them. 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 [159], which is now widely used. Respecting his terminology, we will identify each ECA with the Wolfram number of its local rule.

Example 4 (ECA 90)**.** Let $f : \mathbf{2}^3 \mapsto \mathbf{2}$ be defined as $f(x, y, z) = x + z$. This Boolean function has Wolfram number 90. In Figure 1.1, we show the space-time diagram of ECA 90 operating on a finite cyclic grid of size 100.

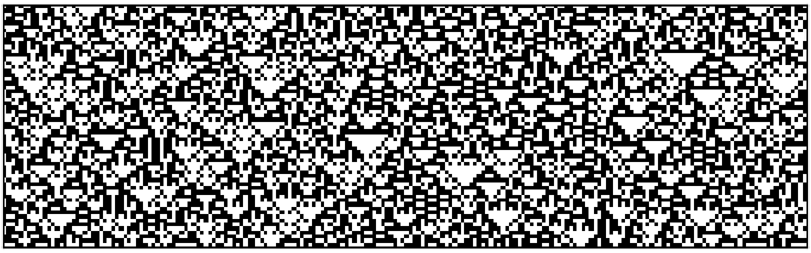


Figure 1.1: Space-time diagram of ECA 90 operating on a finite cyclic grid of 100. We typically identify the state 0 with white and state 1 with black colour. Each row of the depicted diagram corresponds with a configuration in the ECA's trajectory; thus, time is "progressing downwards".

1.2.2 Two-dimensional Cellular Automata

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

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

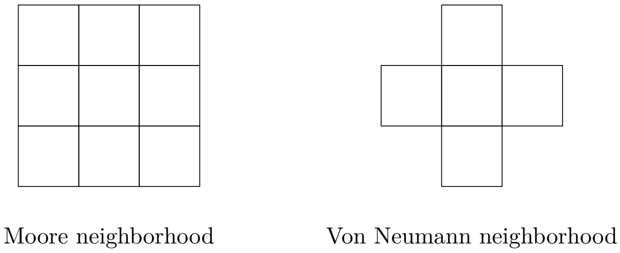


Figure 1.2: Diagrams of two typical neighbourhoods of two-dimensional cellular automata.

Analogously to the one-dimensional case, we can define 2D CAs operating on finite grids. In this case, the finite grid is parametrized by a tuple $(n, m) \subseteq \mathbb{N}^2$. In such a case, in (1.1) the first coordinates of the indices are computed modulo n and the second coordinates modulo m. We say that *grid is of size nm.* Loosely speaking, the topology of the two-dimensional finite grid is that of a torus. In the case of 2D CAs, their space-time diagrams are usually presented as animations, where the frames represent consecutive configurations.

1.2.3 Attractors and Transients

Let $\mathcal{A}$ be a CA operating on a finite grid of size $n \in \mathbb{N}$ with a global rule F and state set S. Since S is finite, the configuration space S^n is as well and therefore, every trajectory $(c, F(c), F^2(c), F^3(c), \ldots)$, $c \in S^n$ becomes eventually periodic. I.e., there exist i, $j \in \mathbb{N}$, $i < j$, such that $F^i(c) = F^j(c)$. Let $i \in \mathbb{N}$ be the smallest such that there exist $j \in \mathbb{N}$, $i < j$, such that $F^i(c) = F^j(c)$. We call the sequence $(F^0(c) = c, F(c), F^2(c), \ldots, F^i(c))$ the *transient of* c. Let $j \in \mathbb{N}$ be smallest such that $F^i(c) = F^j(c)$. We call the sequence $(F^i(c), F^{i+1}(c), F^{i+2}(c), \ldots, F^{j-1}(c))$ the *attractor of* c.

1.3 Brief History of Cellular Automata

While the exploration of cellular automata and related discrete dynamical systems within the context of complexity sciences is relatively recent, the body of work is extensive and characterized by its interdisciplinary nature. In this section, we provide a succinct overview of key results pertinent to this thesis.

The popularity of cellular automata owes much to the groundbreaking work of John von Neumann who was fascinated with designing a non-trivial, self-replicating machine. Inspired by Stanislaw Ulam's suggestion to explore cellular automaton environments, von Neumann successfully realized this vision, introducing what he termed the *universal constructor*. The universal constructor comprised a potentially infinite tape and an ensemble of cells representing the "processor". Its functionality involved interpreting the information on the tape as a blueprint for a new machine, constructing it, and finally, duplicating the information onto the tape of the newly created machine. Its non-triviality was undoubtable, as the processor implemented a universal Turing machine. Once such a constructor received its own description on the tape, it would reproduce itself. Von Neumann's visionary insight to employ the tape information twice — first for interpretation and then for duplication — preceded the discovery of DNA's structure [20].

Arthur Burks later published von Neumann's completed work [112], sparking numerous variations in the design of self-replicating structures. This influential work played a pivotal role in the exploration of artificial life and self-replication. Despite these advancements, scientists debated the existence of a significant gap between "trivial" and "computationally universal" structures. Some argued that natural organisms might not necessarily be Turing complete. Therefore, many other much simpler structures were designed, a great overview was written by Reggia et al. [129], a famous example being the loops designed by Christopher Langton [82].

The popularity of cellular automata was solidified with John Conway's invention of Game of Life [46] – a 2D CA with a very simple update rule that showcases fascinating behaviour. When simulated from various initial configurations, one can observe the emergence of various intricate patterns. The dynamics of Game of Life sustains an enduring allure for researchers and enthusiasts alike.

Since the seminal works of von Neumann and Conway, cellular automata have been extensively explored across diverse research domains such as evolutionary algorithms [105, 120], parallel computation [10], cryptography [53, 75], topological and symbolic dynamics [81, 132], ergodic theory [58, 156], and computational mechanics [60, 136]. In the next section, we discuss in more detail the work on classifying CA dynamics and measuring their complexity, as that is particularly relevant for Part I of this thesis.

1.4 Complexity Measures of Cellular Automata

One of the most widely referred measures of complexity is due to Wolfram who in [158] proposed four "universality classes" of CA dynamical behaviour and claimed that each CA belongs to one such class; we cite them from [158] below:

1) Evolution leads to a homogeneous state.

2) Evolution leads to a set of separated simple stable or periodic structures.

3) Evolution leads to a chaotic pattern.

4) Evolution leads to complex localized structures, sometimes long-lived.

Figure 1.3 illustrates the four classes of CA dynamics.

Wolfram claims that Class 4 is of particular interest as it encompasses all automata with complex behaviour, and further speculates that all such systems are "capable of universal computation". Clearly, the classification is heuristic which makes it challenging to convincingly show that a particular CA belongs to a certain class, especially when it comes to Class 4. This has been addressed in [29] where Culik and Yu suggest formal definitions roughly corresponding

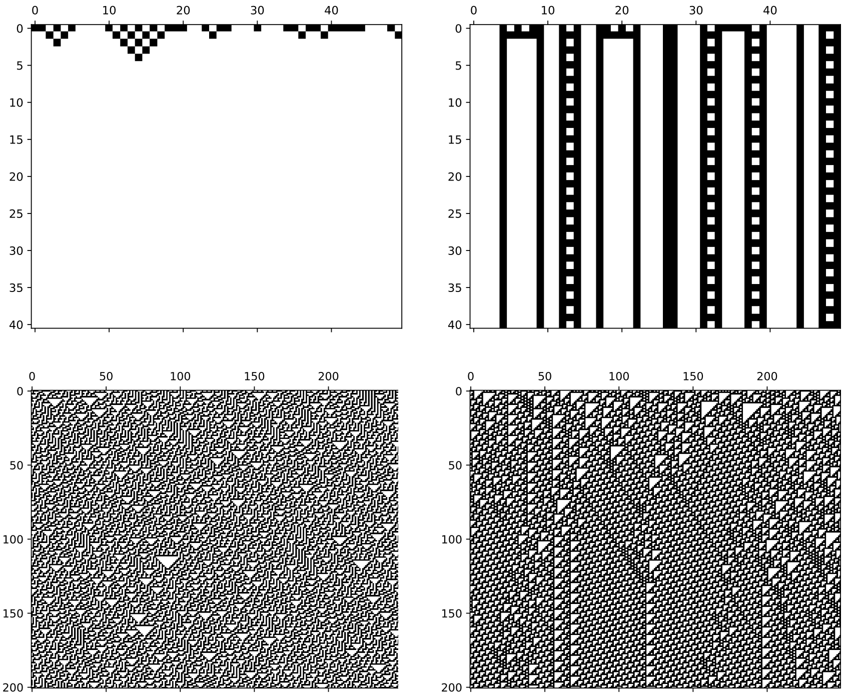


Figure 1.3: Space-time diagrams of CAs from each Wolfram's class. *(Top left)* Class 1 ECA 32. *(Top right)* Class 2 ECA 108. *(Bottom left)* Class 3 ECA 30. *(Bottom right)* Class 4 ECA 110.

to Wolfram's. Before we describe them, we introduce some terminology. In what follows, we will focus solely on one-dimensional automata.

Let $\mathcal{A} = \left(S^{\mathbb{Z}}, F\right)$ be a 1D CA with local rule $f : S^k \to S$ for some $k \in \mathbb{N}$. We say that a state $q \in S$ is *stable* if $f(q, q, \ldots, q) = q$. Let $s \in S$. A configuration with all but finitely many cells in the state s is called an *s-finite configuration.* A configuration with all its cells in the state s is called a *homogeneous configuration of s.* Culik and Yu propose the following hierarchy: $\mathcal{A}$ belongs to Class One if there exists a stable state $s \in S$ such that all the CA's s-finite configurations evolve into the homogeneous configuration of s. It belongs to Class Two if there exists a stable state $s \in S$ such that each trajectory starting from an s-finite configurations becomes eventually periodic. It belongs to Class Three if there exists a stable state $s \in S$ such that for any pair of s-finite configurations c_1, c_2 it is decidable whether c_2 belongs to the trajectory of c_1. Lastly, Class Four consists of all one-dimensional automata. The authors claim that their Class k corresponds to the union of Wolfram's classes up to class k for each $k \in \{1, 2, 3, 4\}$. Among other fundamental results, the authors prove that it is undecidable whether a given CA belongs to Class One, Two or Three.

Despite the fundamental undecidability result, it is a meaningful pursuit to design formal metrics of CA complexity that would be applicable in practice and that would approximate the four classes well enough. Such metrics would allow us to automatically search for automata with complex behaviour in vast CA spaces as well as to elucidate the characteristic properties of Wolfram's "mysterious class 4". Below, we briefly summarize some of the most classical approaches to designing such a metric.

1.4.1 Description Length of Space-Time Diagrams

Intuitively, complex cellular automata are the ones that produce intriguing patterns in their space-time diagrams. Thus, a natural approach to measuring their complexity is to analyse the "description length" of their space-time diagrams. Once suitable means of "describing" the diagrams are fixed, it is to be expected that the description length should be the largest for chaotic systems that produce space-time diagrams with no apparent patterns, the smallest for ordered systems that quickly converge to short attractors, whereas complex automata should have intermediate values. Such a method using a compression algorithm from Mathematica has been explored in [165] by Zenil. He assesses the compression size of each ECA's space-time diagrams (considering a finite cyclic grid) and further applies a clustering algorithm on the results, obtaining CA classes roughly corresponding to Wolfram's classification.

The description length of the space-time diagrams has also been measured by information-theoretic means. In the simplest case, given a trajectory $(c, F(c), \ldots, F^t(c))$ for some configuration c and 1D CA global rule F operating on a finite cyclic grid of a fixed size n, each of the grid cells $i \in \mathbb{Z}_n$ can be identified with a random variable X_i whose outcomes comprise the sequence $c_i, F(c)_i, \ldots, F^t(c)_i$. Then, one can measure the average Shannon entropy H [137] over all the random variables [126]. Again, intuitively, this value should be maximal for chaotic space-time diagrams with no obvious patterns and it should be minimal for ordered systems, while complex system should have intermediate values. Some works use the transformation $H(1-H)$ arguing that in this fashion, the obtained value is maximal for complex systems [89, 133].

1.4.2 Typical Convergence Time

In his seminal work [153] Langton explores the complexity of CAs through the *convergence time to their typical behaviour.* He considers CAs with states $\{0, 1\}$ operating on a finite cyclic grid of a fixed size. His definition of "typical behaviour" encompasses two cases:

1. The system's actual attractor is reached.
2. The system follows a trajectory for a prolonged time (e.g., 100 time-steps) where each configuration has "similar properties" to the rest. Namely, Langton computes the *density* of each configuration in the trajectory (i.e. its average number of 1s). If it holds that each configuration's density does not differ from the trajectory's average by more than 1 %, he assesses the system has reached its typical behaviour.

Langton initializes the configurations uniformly randomly and computes the average time until the system's "typical behaviour" is reached, whichever of the two cases occurs first. He argues that ordered systems converge quickly to their attractors and chaotic systems have short transients to their typical behaviour as well. The complex phase can be identified with an explosion of the transient length.

1.4.3 Self-Organized Criticality and Avalanche Distributions

Another popular measure of CA complexity is based on the famous yet controversial concept of *self-organized criticality* introduced by Bak et al. in 1980s [9, 155]. The core of this concept is the claim that many systems in nature "self-organize towards a critical state" which has spurred active debate in various fields ranging from statistical mechanics to neuroscience [13, 94]. We illustrate this on a simple example of a sandpile model used in the seminal paper [9]. Consider a 2D CA $\mathcal{A}$ defined on a finite cyclic grid with the von Neumann neighbourhood. A cell's state can be any natural number representing the number of sand grains on that specific location. The update function is parametrized by a threshold $K \in \mathbb{N}$: if a cell's number of grains exceeds K, the cell state is decreased by four, and all its neighbouring cell states are increased by 1; otherwise the cell's state is preserved. The automaton is initialized randomly and simulated until an attractor is reached. The authors then perturb the attractor by randomly choosing a cell and changing its state. This then causes an *avalanche*; i.e., a process until a new attractor is reached. The authors measure the "avalanche size", i.e., the number of cells that change their state during the avalanche. By repeating this process, they obtain a probability distribution

$D(s)$ that an avalanche of size s occurs. They argue that $D(s) \sim s^{-\tau}$ for some $\tau \in \mathbb{R}^+$; i.e., the distribution obeys a power law where an arbitrarily large avalanche is essentially possible – such a property is considered typical for a system at a critical point. Similar phenomena have been observed in well-known complex CAs such as Game of Life [8]. The method has since been used to automatically search for CAs with power-law distributions of their avalanches as a means to automatically generate systems with complex behaviour [125].

1.4.4 Discussing the Gaps

All such metrics approximate a property that is in its full generality undecidable, therefore, each of them naturally comes with its own set of shortcomings. One of the most prominent ones is that all methods described above depend on the size of cyclic grid chosen for the experiment. Some of the methods further rely on other parameters such as the length of the trajectories generated for analysing the description length of CA space-time diagrams or a time frame to asses that a system has reached its typical behaviour in Langton's method. In general, dependence of the results on the specific parameter values is rather underexplored in the literature. As an example, in [67] included in Chapter 2 we study in detail Zenil's classification and show that the different parameter values do change the resulting CA clusters qualitatively.

Furthermore, most of the methods assign a specific real-numbered value to each automaton, representing aspects such as its space-time description length, average entropy of cell values or typical convergence time. However, interpreting this value in isolation for a single automaton can be challenging. The value only becomes meaningful when compared to the values of other automata within a given family. By comparing results and forming clusters, these values can be interpreted as indicative of certain CA classes. Consequently, analysing a single automaton from an extensive CA family not yet explored in the literature presents a challenge.

In the following sections, we outline the transient classification we propose, aiming to address both of the aforementioned issues.

1.5 Transient Classification

In this section, we briefly summarize our work from Chapter 2.

The transient classification is a method which aims to formally assess whether a given discrete dynamical system belongs to the "ordered" (Wolfram's classes 1 and 2), "chaotic" (class 3) or "complex" (class 4) regime. The main significance of the method is that it does not depend upon an arbitrary choice of any parameters which makes the results robust. We briefly explain the process below; more details can be found in Chapter 2.

Let us consider a sequence of automata all in a fixed dimension $d \in \mathbb{N}$ given by a fixed neighbourhood and local rule, operating on finite cyclic grids of growing sizes $n_1 < n_2 < \ldots$:

$$\mathcal{A}_1 = (S^{n_1}, F_1), \mathcal{A}_2 = (S^{n_2}, F_2), \mathcal{A}_3 = (S^{n_3}, F_3), \mathcal{A}_4 = (S^{n_4}, F_4), \ldots.$$

Let us fix $i \in \mathbb{N}$. Clearly, $F_i : S^{n_i} \to S^{n_i}$ is a deterministic function operating on a finite set and thus, each of the trajectories $c, F_i(c), F_i^2(c), \ldots,$ $c \in S^{n_i}$ becomes eventually periodic. Hence, its transient length $t(c)$ can be simply measured by running a program that computes the trajectory. Thus, we can compute the *average transient length* of $\mathcal{A}_i$ as: $T(\mathcal{A}_i) := \frac{1}{|S|^{n_i}} \sum_{c \in S^{n_i}} t(c)$. For n_i large, going through all the initial configurations is not feasible, so we simply estimate the value $T(\mathcal{A}_i)$ using the Monte Carlo sampling method. The aim of the transient classification is to compute the average transient lengths $T(\mathcal{A}_1), T(\mathcal{A}_2), \ldots$ for a CA operating on finite cyclic grids of growing size in order to estimate its asymptotic growth; this is illustrated in 1.4.

In practice, we generate a finite part of the sequence $T(\mathcal{A}_1), T(\mathcal{A}_2), \ldots, T(\mathcal{A}_B)$ where B is an upper bound imposed by our computational limitations, and examine different regression

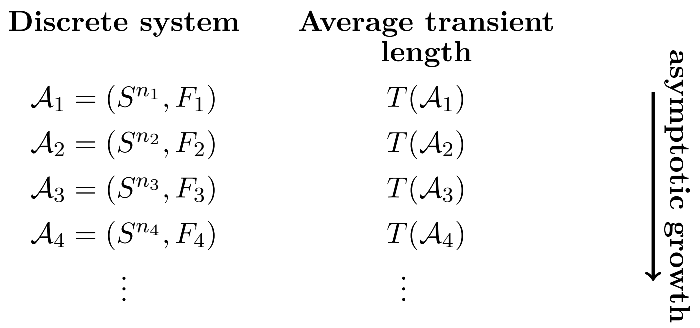


Figure 1.4: Diagram depicting the asymptotic growth of average transient lengths of a sequence of cellular automata operating on finite grids of growing size.

fits of the data. Specifically, we evaluate the fit to constant, logarithmic, linear, polynomial, and exponential functions. We pick the best fit with respect to the R^2 score and obtain the classes: Bounded, Log, Lin, Poly, and Exp. If the score of the fit to all such functions is low (i.e., $R^2 < 85\%$), we say the system is Unclassified. Surprisingly, we found a very good fit to one of the classes with $R^2 > 90\%$ for most systems we examined.

There seems to be good correspondence between the union of the classes Bounded and Log and the union of Wolfram's Classes 1 and 2; we mark the set of such systems as the *ordered phase*. Furthermore, the Exp class seems to correspond well with Wolfram's Class 3; marking a *chaotic phase*. Between the two phases we observe systems whose transients seem to grow linearly or polynomially with the grid size. Such systems form a region that can be interpreted as a "phase transition region" between the ordered and chaotic phase. We measured multiple CAs generally considered as complex, such as Game of Life, and the results suggest that such systems belong to this region. This supports our hypothesis that the Lin and Poly Classes roughly correspond to Wolfram's Class 4. The trend we have observed, which seems to hold across various families of discrete dynamical systems, is shown in Figure 1.5.

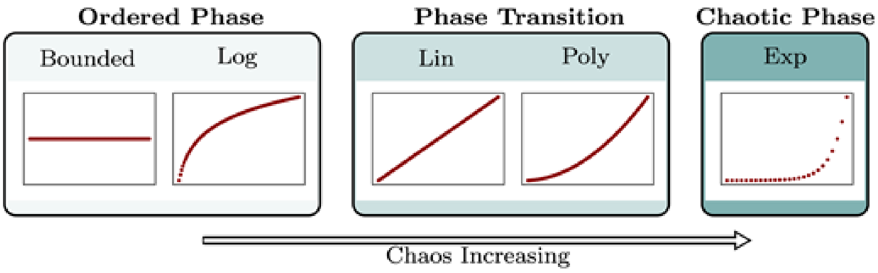


Figure 1.5: General trend of the transient classification results.

Thus, the method allows us to classify the dynamics of a single automaton without relying on a choice of any arbitrary parameters and yields a good approximation to Wolfram's classes. Whereas the transient classification seems to be a good tool for approximating the complexity in discrete systems, we also discuss some of its drawbacks below.

Computationally Demanding Nature of the Classification The most obvious one is that the method merely approximates the asymptotic behaviour of a system from finitely many data points and does not guarantee us the true asymptotic transient growth. For some systems, the transient growth might correspond to more complicated functions but we have deliberately chosen the classes Bounded, Log, Lin, Poly, and Exp to be quite robust and coarse to have clearer boundaries between them. The uncertainty of the true asymptotic growth is especially relevant for the Lin and Poly Classes which identify the critical phase transition region. Such systems might turn out to be logarithmic or exponential, and it might merely be the case that we have not detected this due to our limited data. However, in such a case, such systems would exhibit significantly slower convergence to their asymptotic behaviour than systems in other classes which is a typical property of a system at a phase transition. In addition, from our experiments, it appears that the distribution of transient lengths for such systems tends to have a long tail, thus it is hard to estimate the true average well which adds to the computational intensity of the classification method.

Coarse Sampling of Initial Configurations We now describe a more nuanced issue which is nevertheless very important to mention. Given a dynamical system and its size $n \in \mathbb{N}$, the Monte Carlo Method for estimating its average transient relies on sampling the initial configurations uniformly randomly. For a system with binary states where the initial configurations belong to $\{0,1\}^n$, this leads to a certain bias: most configurations sampled by such means are “balanced”, i.e., they contain the same number of 0s and 1s. Thus, the classification gives us information about the dynamics of the system which is always initiated from a special region of its initial configurations. Obviously, if the system is initiated differently, its dynamics can differ fundamentally and such variance is not grasped by the classification. However, in order to gain intuitive insight into a system’s dynamics, the most classical method used in literature is to randomly sample an initial configuration of a CA and observe the space-time diagram generated from such a configuration. Thus, the classification simply mimics this procedure. If it is required to study the dynamics of a system initialized from a different region, one can simply sample the initial configurations uniformly from that specific region to obtain the results about the system’s transients in the chosen regime.

1.5.1 Phase Transitions

Chapter 3 of the thesis can be seen as a natural continuation of the work on transient classification, which in particular addresses the shortcomings described above. Before we summarize the results of Chapter 3 in the next section, we highlight an important difference between the approaches in Chapters 2 and 3.

In the transient classification, we fix a specific region of initial configurations and proceed by studying the behaviour of various discrete dynamical systems. We group together systems with qualitatively similar behaviour and identify a phase transition region in the space of the discrete dynamical systems.

In contrast, in Chapter 3, we fix a specific discrete dynamical system: as an example, this could be an *anti-conformist GCA* that we introduce in Chapter 3. For such a system, we study its behaviour while varying the system’s initial configurations. A suitable parameter of the configuration space $\{0,1\}^n$ is the configuration density; i.e., its average number of 1s. Using methods from statistical physics, we show there can be abrupt changes in the system’s behaviour as we increase the configuration density. Thus, in Chapter 3 we identify phase transitions in the space of the initial configurations. The difference of the context in which we use the notion “phase transition” between Chapters 2 and 3 is illustrated in Figure 1.6.

1.6 Dynamical Phase Transitions in Graph Cellular Automata

The contribution of Chapter 3 is two-fold:

- We showcase sophisticated, newly developed tools from statistical physics and demonstrate they yield new analytical results about dynamics of discrete dynamical systems. Specifically, we use them to analyse systems very close in architecture to cellular automata that we call *Graph Cellular Automata* (GCAs).

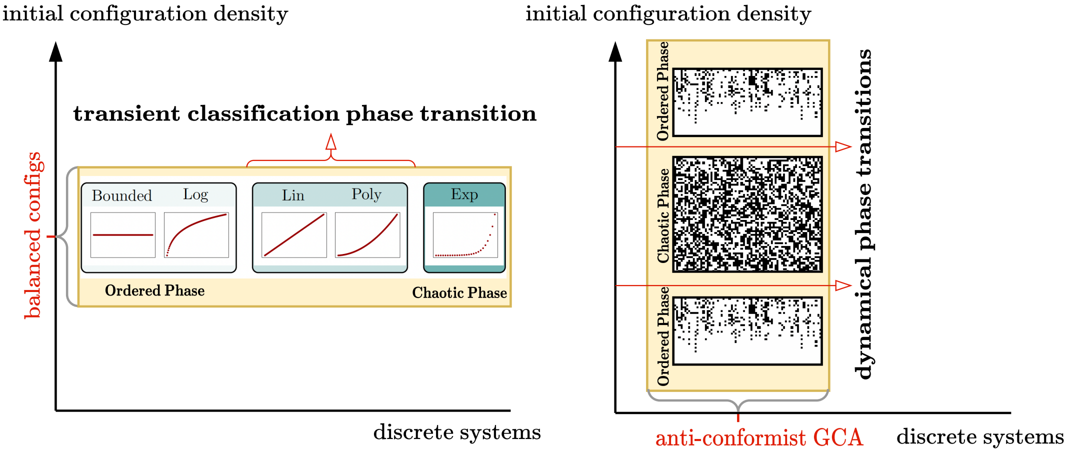


Figure 1.6: We can view each system to be given by its update rule (x-axis) together with a specific region of initial configurations (y-axis); yielding essentially a 2-dimensional space. *(Left)* In the transient classification, for a fixed y value, we traverse the x-axis. *(Right)* In contrast, in Chapter 3, for a fixed x value, we traverse the y axis.

- We study a specific family of GCAs for which we reinforce the well-known fact: the initialisation of a system matters. Concretely, we identify qualitatively different dynamical behaviours of a given GCA that we call *dynamical phases*. Further, we demonstrate the GCA's phase can abruptly change as we increase the initial configuration density. The tools we use allow us to precisely describe the value at which such a dynamical phase transition occurs in the limit when the GCA's size grows to infinity.

We describe the two points in more detail below.

1.6.1 Randomized Discrete Dynamical Systems

As previously discussed, obtaining analytically exact general results regarding the dynamics of cellular automata poses a great challenge, given that various properties of CA dynamics have been proven to be undecidable [29, 74, 77]. However, one can study discrete dynamical systems related to cellular automata that have a more "randomized architecture". Then, it becomes possible again to derive analytically correct results about the typical dynamics of the system in the limit when its size goes to infinity, using tools from statistical physics. Below, we describe a simple example of such a result, preceded by an introduction to relevant terminology.

Graph Terminology By a *directed graph* of size $n \in \mathbb{N}$ we understand the tuple $G = (V, E)$ where $V = \{1, \ldots, n\}$ is the set of nodes and $E = \{(i, j) \mid i, j \in V\}$ is the set of edges. For each node i, we define its *neighbourhood* to be the set $\partial_i = \{j \mid (j, i) \in E\} \subseteq V$; and we define the indegree of i as $d(i) = |\partial_i|$.

By an *undirected graph* of size n we understand the tuple $G = (V, E)$ where $V = \{1, \ldots, n\}$ is the set of nodes and $E = \{\{i, j\} \mid i, j \in V\}$ is the set of edges. For each node $i \in V$ we define the *neighbourhood of* i to be the set $\partial_i = \{j \mid \{i, j\} \in E\} \subseteq V$; and we define the degree of i as $d(i) = |\partial_i|$. We say an undirected graph is *d-regular* if each node has degree d.

Let G be a (either directed or undirected) graph with n nodes and let S be a finite set of *states*. Each node i can be assigned a state $x_i \in S$; we represent such an assignment by the sequence $\mathbf{x} = x_1 \ldots x_n \in S^n$ and call it an *S-configuration* or just a *configuration*.

Random Boolean Networks Let $G = (V, E)$ be a directed graph with $n \in \mathbb{N}$ nodes and let $f_1, \dots, f_n$ be a sequence of Boolean functions such that $f_i : \{0, 1\}^{d(i)} \to \{0, 1\}$ for each $i \in V$. For each node $i \in V$ we fix a particular ordering of the nodes in its neighbourhood: $(i_1, \dots, i_{d(i)})$. A *Boolean network* $\mathcal{B}$ *of size* n given by $G, f_1, \dots, f_n$, and the orderings of the neighbourhoods is a discrete dynamical system $(\{0, 1\}^n, F)$ where $F : \{0, 1\}^n \to \{0, 1\}^n$ is defined as follows for each configuration $\mathbf{x} \in \{0, 1\}^n$ and each $i \in V$:

$$F(\mathbf{x})_i = f_i(\mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_{d(i)}}).$$

The *average connectivity of* $\mathcal{B}$ is defined as $\frac{\sum_{i=1}^{n} d(i)}{n}$. Given $N \in \mathbb{N}$ and $K \in \mathbb{R}^+$ we say that $\mathcal{B}$ is an *N-K Random Boolean Network (RBN)* if it is a Boolean network that has been uniformly randomly sampled from all the Boolean networks of size N with average connectivity K.

It has been shown that given a fixed K one can analyse the dynamics of a typical N-K random Boolean network with $N \mapsto \infty$. Specifically, using simple approaches from statistical physics (in this case, the mean field calculations and annealed approximations) one can study how does the Hamming distance between two randomly chosen initial configurations evolve over time. It has been shown in [38, 91] that for $N \mapsto \infty$ if $K < 2$ the distance of the configurations converges to 0 marking an "ordered phase", whereas for $K > 2$ the configurations diverge marking a "chaotic phase" with the precise value $K = 2$ identifying a "critical regime". Great overviews of the abundant results regarding random Boolean networks are for example [47, 71].

Informally speaking, the annealed approximations for RBNs give good results precisely because their architecture is "very random". Once studying discrete systems with less randomized architectures, the application of more sophisticated tools are necessary. This is exactly the case for graph CAs that we study in Chapter 3: the only difference between CAs and GCAs is that whereas the connectivity topology of the former is given by a regular grid, for the latter this is given by a random regular graph.

Graph Cellular Automata. Let S be a finite set of states. A Graph Cellular Automaton (GCA) is a discrete dynamical system that operates on configurations of some d-regular graph with n nodes. In this work, we only consider the case when the graph is sampled uniformly randomly from the set of all d-regular graphs. The state of each node gets updated synchronously, depending on its own state and the state of its neighbours; each node $i \in V$ uses an identical local update rule $f : S \times S^d \mapsto S$ and has a fixed ordering of its neighbours: $(i_1, \dots, i_d)$. This gives rise to a global mapping $F : S^n \to S^n$ governing the dynamics of the system. For a configuration $\mathbf{x} \in S^n$, the i-th node gets updated according to:

$$F(\mathbf{x})_i = f(x_i; x_{i_1}, \dots, x_{i_d}).$$

We write a semicolon to highlight that the first entry of f is always the state of the node being updated.

Fig 1.7 illustrates the difference between the architectures of cellular automata, graph cellular automata, and random Boolean networks.

Below, we give a very brief overview of the method we use in Chapter 3 to analyse the dynamical phase transitions of certain graph cellular automata.

1.6.2 Backtracking Dynamical Cavity Method

The Backtracking Dynamical Cavity Method (BDCM) was introduced only very recently by Freya Behrens and Lenka Zdeborová in [15]. It is a versatile approach based on the cavity method from statistical physics [100, 101] that allows us to analyse the dynamics of complex systems out of equilibrium. It can be seen as an extension of a previously introduced *Dynamical*

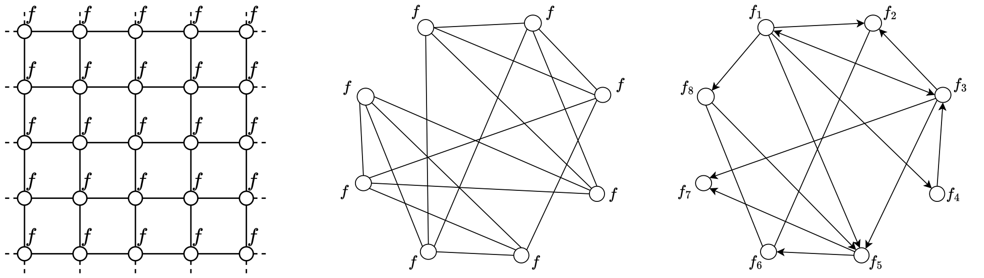


Figure 1.7: Figure illustrates architectures of three discrete dynamical systems. *(Left)* Cellular automata. *(Middle)* Graph cellular automata. *(Right)* Random Boolean networks.

Cavity Method (DCM) [62, 73, 88]. Given a dynamical system suitable for the application of BDCM (more details on this can be found in Chapter 3), the method allows us to answer questions such as:

- How many attractors of size 1 does the system have?
- Given initial configurations with a fixed density ρ_{init}, how many of them converge to attractors of size 1 in a fixed number of time-steps?

We illustrate the core idea of the method very coarsely on one of the simplest examples: determining the number of size 1 attractors of a particular GCA.

Let us consider a GCA of size n on a random d-regular graph of size $n \in \mathbb{N}$ with states $S = \{0,1\}$ and a local update rule $f : \{0,1\} \times \{0,1\}^d \to \{0,1\}$. We denote the ordered neighbourhood of node $i \in \mathbb{N}$ as $\partial_i = (i_1, \dots, i_d)$. We introduce a probability distribution over all configurations such that all size 1 attractors have equal probability, and any other configuration has probability zero; for $\mathbf{x} \in \{0,1\}^n$ we define:

$$P(\mathbf{x}) = \frac{1}{Z} \prod_{i=1}^{n} \mathbb{1}\left[f(\mathbf{x}_i, \mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_d}) = \mathbf{x}_i\right] \qquad \text{where}$$

$$Z = \sum_{\mathbf{x} \in \{0,1\}^n} \prod_{i=1}^{n} \mathbb{1}\left[f(\mathbf{x}_i, \mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_d}) = \mathbf{x}_i\right].$$

Here, $\mathbb{1}(\cdot)$ is the indicator function yielding a 1 on a true Boolean statement and 0 otherwise.

If we had access to the value Z, we would immediately know the number of size 1 attractors of the system. However, computing it exactly is intractable for large n since the sum goes over exponentially many summands. We typically assume that with $n \mapsto \infty$, Z grows approximately as e^{sn} for some $s \in \mathbb{R}^+$. In this case, we call s the *entropy* of the size 1 attractors. In this scenario, the BDCM employs the *belief propagation algorithm* that, under certain assumptions, efficiently recovers the value of s that is asymptotically exact in the limit when $n \mapsto \infty$.

Explaining the belief propagation formally is out of the scope of this introduction; its detailed description can be found in the textbook [99]. Informally speaking, the belief propagation algorithm works as follows: each node in the graph gets assigned certain “messages” that represent partial information about s. Such messages are passed between neighbouring nodes iteratively: each node assembles the messages from its neighbours to update its own message values. The message updates can be done efficiently only under the assumption that the messages travelling to the node are uncorrelated – that they come from branches that do not interact, in other words that the graph is a tree. Once the algorithm converges and the messages do not change their values after an update, the value of s can be reconstructed from the messages. When the underlying graph contains loops, the same algorithm can be used heuristically; however, there is no guarantee in general that the retrieved solution is correct.

For a random d-regular graph, it is a famous fact that as the size of the graph grows to infinity, the graph looks locally tree like: specifically, for a random d-regular graph of size n, the length of the shortest loop going through a typical node grows approximately as $\log(n)$. Nevertheless, in order to have a guarantee that the belief propagation recovers an asymptotically exact solution, further assumptions (called the replica symmetry [99]) need to be checked.

In Chapter 3, we use this method to answer more sophisticated questions about graph CAs. Specifically, we identify certain *dynamical phases* of a GCA that depend on:

- Type of the system's attractor. E.g.: Is it a size 1 attractor? If yes, does it consist of a homogeneous configuration or not?
- Speed of convergence to the attractor. E.g.: Does the system of size n reach the attractor within $\log(n)$ steps? Or does it converge in more than $\exp(n)$ steps?

Subsequently, for a particular $\rho_{\text{init}} \in [0, 1]$, we measure the entropy of each dynamical phase the system exhibits when initiated with configurations with the given ρ_{init}. This allows us to assess for each $\rho_{\text{init}} \in [0, 1]$ what is the most typical phase of the system. We systematically do this for increasing values of ρ_{init} and this allows us to spot abrupt changes in the type of typical dynamical phase the system exhibits: we call this the *dynamical phase transition.* The DCM and BDCM allow us to identify exact values of ρ_{init} where such phase transitions occur. We give an example of one of our results from Chapter 3 below.

1.6.3 Dynamical Phase Transitions in an Anti-Conformist GCA

Specifically, in Chapter 3, we analyse a family of GCAs called *conforming non-conformist GCAs.* We now showcase the results on one particular example.

We fix a degree $d = 5$. The *anti-conformist GCA* given by a random 5-regular graph of size n with states $\{0, 1\}$ updates each node $i \in \{1, 2, \dots, n\}$ in the following way:

- Majority of bits in the neighbourhood of i is computed.
- If at least four neighbours are in the majority state, node i is assigned the majority state.
- Otherwise, if only three neighbours are in the majority state, node i is assigned the minority state.

After a careful numerical analysis of the anti-conformist GCA's behaviour, we identified two dynamical phases of the system.

- In the *rapid phase* the system converges fast (approximately within $\log(n)$ steps) to attractors of size 1 or 2.
- In the *chaotic phase* the system converges slowly (approximately in $\exp(n)$ steps) to attractors of size 1 or 2.

For the anti-conformist GCA described above, we used DCM to identify a precise value of ρ_{init} when the system abruptly changes its typical dynamical phase from rapid to chaotic. The results showing such a dynamical phase transition are described in detail in Figure 1.8. Therein, we also demonstrate that the analytical result matches well with the results of our numerical experiments.

This concludes the introduction to Part I of this thesis. Below, we proceed with the introduction to Part II.

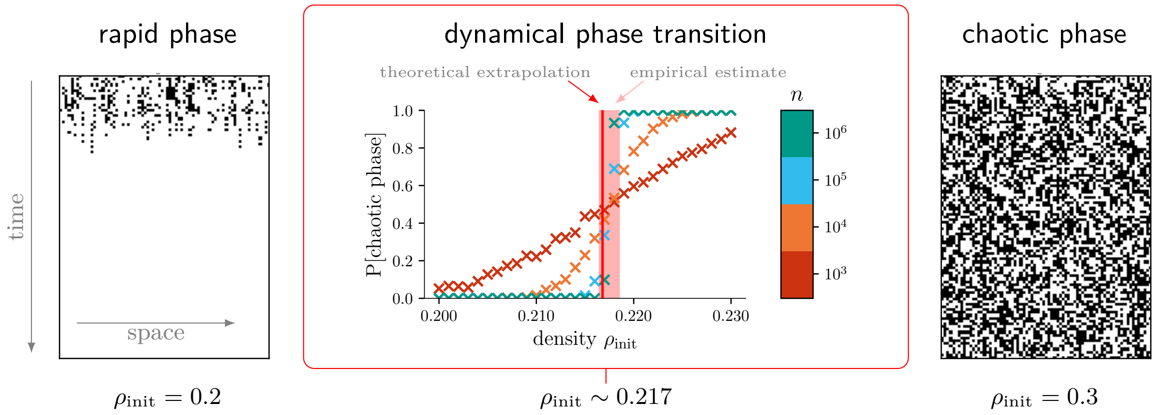


Figure 1.8: **A phase transition diagram for a particular instance of a 5-regular GCA with a an anti-conformist rule.** An illustration of the system's two phases that depend on the density (i.e., the average number of black-coloured nodes) in the initial configuration. The phases are illustrated by space-time diagrams for a system of size $n = 1000$ nodes, though only a window of 75 nodes is shown. *(Left)* Rapid phase: Fast convergence to the all-0 attractor. *(Right)* Chaotic Phase: Apparent randomness in the node's state, convergence takes longer. *(Middle)* In the large system limit, when $n \to \infty$, there is a dynamical phase transition. At a particular initial density value ρ_{init}, the typical behaviour of the system abruptly switches from the rapid to the chaotic phase. For each ρ_{init} and each system size n we sampled 1024 initial configurations with the given ρ_{init} and computed how often the system enters a chaotic phase. For practical purposes, we conclude the system is in a chaotic phase if it does not converge within $100 * \log_2(n)$ time-steps. The resulting frequency exhibits a sharp phase transition between 0.217 and 0.218, where the solid red line is our prediction from the DCM and the shaded red area comes from an empirical approximation. This transition separates the behaviour on the left and the right.

1.7 Cellular Automata as Models of Computation

Ever since the initial seminal papers on cellular automata [84, 112, 158] there has been a lot of interest in the connection between a system's dynamics and its ability to compute. As an example, we remind the bold assertion by Wolfram that all class 4 automata are capable of universal computation [158]. Showing that a given CA can solve a challenging computational task is indeed a convincing way to demonstrate that the system is complex. Part 2 of this thesis thus explores the complexity of cellular automata from the perspective of their computational capacity. We first give a brief overview on the literature studying cellular automata as models of computation. Subsequently, we summarize the results in Chapters 4 and 5 of the thesis.

The lines of work on CA computation naturally split into two branches: a practical one exploring how to automatically find automata that can solve tasks efficiently and a theoretical one formally studying computational capacity of CAs as well as their limitations.

1.7.1 Cellular Automata as Efficient Computers

When studying cellular automata as practical, fast models of parallel computation, it typically entails tackling the following questions:

1. What is a good set of challenging computational tasks that are suitable for the parallel nature of cellular automata?

2. Given such a task, what is a suitable way to encode the input into the CA's configurations? How to define the CA's halting state? And how to decode the CA's output?

3. Given a task and an “encoding and decoding scheme”, what is a good optimization method that would automatically search for automata capable of solving the task efficiently?

Such questions have been extensively explored and we give a brief overview of some of the works below.

Tasks for Cellular Automata

One classical line of work explores the ability of CA to self-organize. This includes tasks such as the famous “firing squad synchronization problem”. In this scenario, the objective is to design a CA with the following property: given an initial configuration on an arbitrary finite grid where all cells are in a stable state except for a single cell representing the “command to fire”, the CA should evolve in such a way that all cells eventually synchronize into a uniform “firing signal” state. Importantly, this synchronization should emerge without any prior occurrence of the “firing signal” state in the CA’s trajectory [152]. The original motivation behind this problem stems from the challenge of simultaneously activating all cells comprising a self-replicating CA structure, such as the one designed by von Neumann, when only local interactions are possible.

Subsequently, there have been multiple works exploring CAs as models of morphology. They typically aim to find (mostly 2D) CAs that can self-organize into various patterns from a given initial configuration: such as countries’ flags [113] or emojis resilient to perturbation [107]. Another simple, yet well-known task of CA synchronization is the majority task [22, 32]. Here, the goal is to design a CA that converges to a homogeneous configuration comprising the state that occurred most frequently in the initial configuration.

Other lines of work explore the capability of CAs to solve classical benchmark tasks from the machine learning field. This includes simple tasks such as memorizing 5 bits of information across prolonged time while the system is perturbed [163], balancing a cart-pole [149], or classifying MNIST digits [127]. Furthermore, other works explore the potential of CAs as language models [25]. A great overview of computation in CAs was written by Mitchell in [103].

Encoding and Decoding Information

The major challenge lies in understanding how a given CA processes information and in choosing suitable methods of encoding the input into the CA’s configurations as well as decoding the CA’s output. In practical scenarios, these choices can be made arbitrarily, or the encoding and decoding mappings can be optimized through some training process. A popular scheme employing such optimal searches is called reservoir computing [90, 163]. In this approach, the encoding is chosen randomly, and the decoding mapping is represented by a simple neural network layer trained to maximize the CA’s performance on a specific task. Notably, the CA reservoir approach stands out for its significantly greater energy efficiency when compared to the conventional training methods used for neural networks.

Evolving Cellular Automata

As opposed to carefully hand-designing automata capable of computing a given task, many works explore optimization methods for their automated discovery. One classical approach employs genetic algorithms, drawing inspiration from biological evolution [105, 120].

Assume a fixed way to evaluate CAs’ performance on a given computational task. In a typical setting, each CA is encoded as a bit string. Initially, the algorithm chooses a population of such strings at random. At each iteration, the population evolves by undergoing “mutations” – the strings are randomly perturbed – and “crossover” – the strings are combined to form new ones. Once a new population is obtained, it is subject to “selection” – the performance of each CA is evaluated on the given task and the “fittest candidates” are chosen for the next iteration. As the process is iterated, the fitness of the population should increase, with the hope of eventually containing well-performing candidates. This method has been used, e.g., to find CAs solving the majority task [106].

Despite the abundant work on exploring CAs as practical, fast computers, it still remains a great challenge to make their efficiency comparable to other state of the art models of computation. Furthermore, there is a considerable gap in our understanding of which automata are most suitable for efficient computing. This motivates the theoretical research on CA computation that we discuss below.

1.7.2 Theory of CA Computation

The body of work described above aims to showcase that cellular automata are fast, parallel computers. In contrast, the theoretical line of work focuses on answering questions such as:

- Given a particular class of CAs, does it contain one capable of universal computation?
- Given a particular CA, is it computationally universal?
- What CA classes are limited in terms of what they can compute?

Universal Computation in Cellular Automata The classical way to formally demonstrate a CA's complexity is to prove its Turing completeness. This is typically done by considering a suitable computationally universal system (a universal Turing machine, universal tag system, etc.) and embedding its computations into a CA's space-time diagrams. In this way, there have been constructed universal cellular automata with various properties; such as a 1D CA with a totalistic local rule (i.e., the local rule does not depend on the order of the cells in the CA neighbourhood) [1], a reversible CA (i.e., a CA whose global rule is an injective mapping) [146] or a 1D number conserving CA (a CA whose states form a subset of the integers that satisfies the following property: given an arbitrary finite cyclic grid, the CA global rule operating on it preserves the sum of states in each configuration) [108].

Another impressive line of work studies the universality of a particular given CA. It has been of great interest whether complex CAs with fascinating visualisations of their dynamics are Turing complete. This has indeed been proven e.g., for Game of Life or ECA 110 [18, 27]. Such proofs are typically very elaborate and require identifying various local structures that propagate through space in the CA's space-time diagrams (such as a glider in Game of Life). Subsequently, the interactions of such structures are shown to encode "basic logic gates" of the universal system being embedded into the CA's dynamics.

Intuitively, the mapping that embeds the universal system's computations into the CA's space-time diagrams should be "simple enough" to ensure that it is not performing the computation instead of the automaton at hand. However, there is no consensus on the precise definition of such embeddings, which makes it extremely hard to prove convincing negative results about the computational capacity of a CA. A great overview on universality in cellular automata was written by Ollinger in [118].

Cellular Automata Relative Simulation A different approach to formally assessing a CA's computational capacity is through the notion of *CA relative simulation.* Informally, we say that CA $\mathcal{A}$ is simulated by $\mathcal{B}$ if each space-time diagram of $\mathcal{A}$ can be, after suitable transformations, reproduced by $\mathcal{B}$. We argue that comparing two cellular automata is much more natural than comparing a CA with a Turing machine, since in the latter case, the architectures of the systems differ substantially. Past works have explored various notions of CA simulations, typically focusing on positive results: for a fixed family of CAs and a fixed CA simulation definition, authors construct *intrinsically universal CAs*; i.e., cellular automata that are able to simulate any other CA within the fixed family [1, 39, 40, 41, 49, 116, 117].

In contrast, the complementary work focuses on the negative results. For various notions of CA simulations, the goal is to show that particular types of automata are limited in terms of what they can simulate. Such results are scarce, yet they bring a valuable insight into the structure of the CA space imposed by the simulation relation. For certain CA simulation definitions, negative results have been shown about various classes of CAs such as nilpotent CAs or particular additive automata [35, 97, 98].

Generally, each CA relative simulation definition considers a certain class of CA transformations T that map each automaton $\mathcal{B}$ into a class of related automata $\mathrm{T}(\mathcal{B})$. Then, we say that $\mathcal{A}$ can be simulated by $\mathcal{B}$ if $\mathcal{A} \in \mathrm{T}(\mathcal{B})$. We propose an informal classification of the previously studied transformations into:

- algebraic: transformations of the CA's local rule; e.g.: products, sub-automata, quotients, iterations
- geometric: transformations of the CA's grid structure; e.g.: tiling of the grid space and grouping of multiple cells, shifts, reflections

In Chapter 4 of this thesis, we propose a definition of CA simulation that is, to the best of our knowledge, the most general algebraic one so far. We briefly summarize the results of Chapter 4 below.

1.8 Simulation Limitations of Affine Cellular Automata

In this section, we focus solely on one-dimensional cellular automata to simplify the notation. Each 1D CA $\mathcal{A} = (S^{\mathbb{Z}}, F)$ with radius r is fully determined by its local algebra $\mathbb{A} = (S, f)$, $f : S^{2r+1} \to S$. We sometimes write $\mathbb{A} = (S, f)_r$ to highlight the CA's radius. We first concisely introduce the definition of CA simulation.

1.8.1 Defining CA Simulation

The CA simulation we introduce is based on combining four notions: *sub-automata*, *quotient automata*, *CA products*, and *CA iterative powers*. We briefly explain them below.

Let $\varphi : S \to T$ be a mapping between finite sets. We define its *canonical extension* $\overline{\varphi} : S^{\mathbb{Z}} \to T^{\mathbb{Z}}$ simply as $\overline{\varphi}(c)_i = \varphi(c_i)$ for each $c \in S^{\mathbb{Z}}$ and each $i \in \mathbb{Z}$.

Definition 5. Let $\mathcal{A} = (S^{\mathbb{Z}}, F)$ and $\mathcal{B} = (T^{\mathbb{Z}}, G)$ be CAs with local algebras $\mathbb{A} = (S, f)_r$ and $\mathbb{B} = (T, g)_r$ respectively. We say that:

1. $\mathcal{A}$ is a sub-automaton of $\mathcal{B}$ (up to isomorphism) if there exists an injective mapping $\iota : S \to T$ such that $\overline{\iota} \circ F = G \circ \overline{\iota}$. This is illustrated in Figure 1.9.

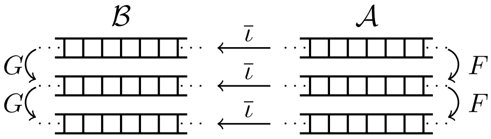


Figure 1.9: Diagram depicts $\mathcal{A}$ a sub-automaton of $\mathcal{B}$.

2. $\mathcal{A}$ a quotient automaton of $\mathbb{B}$ (up to isomorphism) if there exists a surjective mapping $\pi : T \to S$ such that $F \circ \overline{\pi} = \overline{\pi} \circ G$. This is illustrated in Figure 1.10.

3. Let $k \in \mathbb{N}$ and let $\mathbb{B}_1 = (T_1, g_1)_r, \ldots, \mathbb{B}_k = (T_k, g_k)_r$ be local algebras of some automata $\mathcal{B}_1, \ldots \mathcal{B}_k$ with radius r. We say that $\mathcal{A}$ is a product of $\mathcal{B}_1, \ldots, \mathcal{B}_k$ if $\mathbb{A} = \mathbb{B}_1 \times \cdots \times \mathbb{B}_k$.

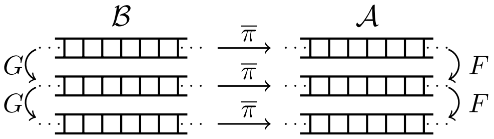


Figure 1.10: Diagram depicts $\mathcal{A}$ a quotient automaton of $\mathcal{B}$.

The mappings $\overline{\iota}$, and $\overline{\pi}$ provide means of "translating" between the space-time diagrams of $\mathcal{A}$ and $\mathcal{B}$. Specifically, they witness that whenever $\mathcal{A}$ is a sub-automaton or quotient automaton of $\mathcal{B}$, each space-time diagram of $\mathcal{A}$ can be reconstructed from a suitable space-time diagram of $\mathcal{B}$. This is illustrated in Example 6 for the case of a quotient automaton. Similarly, given automata $\mathcal{B}_1, \ldots, \mathcal{B}_k$ with the same radius such that $\mathcal{A}$ equals to their product, it is clear that together, the CAs $\mathcal{B}_1, \ldots, \mathcal{B}_k$ can reconstruct any space-time diagram of $\mathcal{A}$. It is crucial that the mappings $\overline{\iota}$, and $\overline{\pi}$ can be efficiently implemented by a computer program as they are extensions of mappings on finite sets. Moreover, the simplicity of the mappings guarantees that they do not process the information contained in the space-time diagrams in any non-trivial way. This is particularly important since, e.g., whenever $\mathcal{A}$ is a sub-automaton of $\mathcal{B}$, we would like to conclude that $\mathcal{B}$ is computationally stronger or equal to $\mathcal{A}$.

We now state an important observation while omitting the technical details: When assessing whether a given automaton is a sub-automaton or a quotient automaton of some CA, it suffices to study their local algebras. This is a crucial observation as it roots the theory of CA simulation into algebraic language.

Example 6. Let $\mathcal{B} = (T^{\mathbb{Z}}, G)$ be the CA with local algebra $\mathbb{B} = (\{0,1,2,3\}, g)_1$ defined as $g(x,y,z) = (x+z) \bmod 4$. Let $\mathcal{A} = (S^{\mathbb{Z}}, F)$ be the CA with local algebra $\mathbb{A} = (\{0,1\}, f)_1$ where $f(x,y,z) = (x+z) \bmod 2$ ($\mathcal{A}$ is the ECA 90). We define the mapping:

$$\begin{aligned} \pi : \quad \mathbb{B} &\longrightarrow \mathbb{A} \\ 0,2 &\longmapsto 0 \\ 1,3 &\longmapsto 1 \end{aligned}$$

Then, one can easily check that the canonical extension of π satisfies $F \circ \overline{\pi} = \overline{\pi} \circ G$. Figure 1.11 illustrates how any space-time diagram of $\mathcal{A}$ can be obtained from a suitable space-time diagram of $\mathcal{B}$ using $\overline{\pi}$ as the "translation mapping".

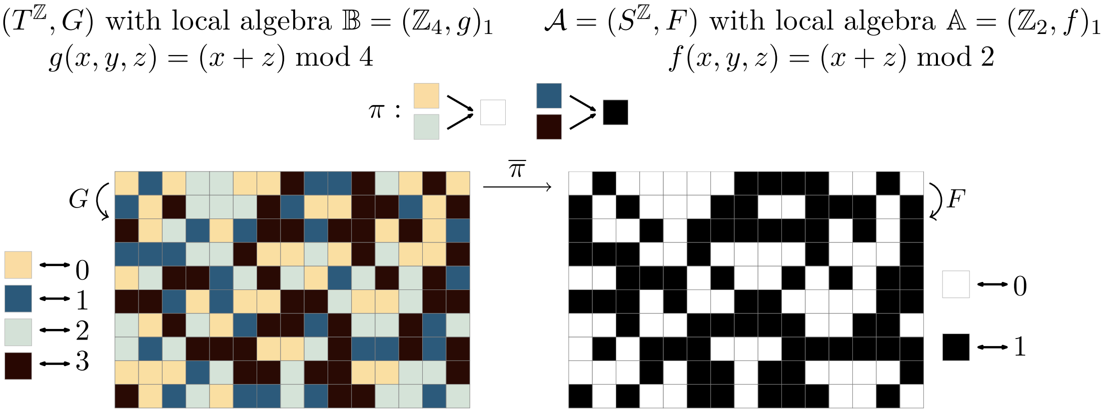

$\mathcal{B} = (T^{\mathbb{Z}}, G)$ with local algebra $\mathbb{B} = (\mathbb{Z}_4, g)_1$
$g(x,y,z) = (x+z) \bmod 4$
$\mathcal{A} = (S^{\mathbb{Z}}, F)$ with local algebra $\mathbb{A} = (\mathbb{Z}_2, f)_1$
$f(x,y,z) = (x+z) \bmod 2$
π :
$\overline{\pi}$
G
F
0
1
2
3
0
1

Figure 1.11: Illustration of Example 6. The figure shows the canonical extension $\overline{\pi}$ which effectively "translates" space-time diagrams of $\mathcal{B}$ to any given diagram of $\mathcal{A}$.

The notions of sub-automata, quotient automata and CA products do not take into account the most important aspect of cellular automata: the iterative application of their local rules. Thus, we describe below the established notion of *CA iterative powers*.

Let $\mathcal{B} = (T^{\mathbb{Z}}, F)$ be a CA with local algebra $\mathbb{B} = (T, g)_r$ and let $n \in \mathbb{N}$. It is natural to iterate the global map and obtain a new discrete system $(T^{\mathbb{Z}}, F^n)$. The aim is to construct a local algebra that would correspond to the CA $(T^{\mathbb{Z}}, F^n)$ while preserving the CA's radius; this is important as the sub-automata, quotients, and products only allow us to compare CAs with the same radius.

Without going into too many technical detials, the "trick" when constructing the local algebra corresponding to $(T^{\mathbb{Z}}, F^n)$ while preserving the CA's radius is to group together sequences of n consecutive cells – this is illustrated in Figure 1.12.

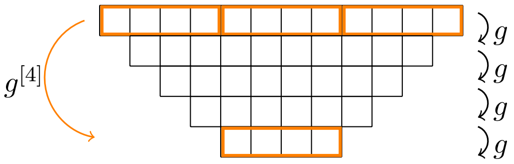


Figure 1.12: Illustration of $g^{[4]}$ for a ternary function g.

Definition 7. Let $\mathcal{B} = (T^{\mathbb{Z}}, F)$ be a CA with local algebra $\mathbb{B} = (T, g)_r$. We define its n-th iterative power to be the CA with local algebra $\mathbb{B}^{[n]} = (T^n, g^{[n]})_r$.

Thus each CA with local algebra $\mathbb{B} = (T, g)_r$ gives rise to a whole series of CAs with local algebras:

$$\mathbb{B} = (T, g)_r,\ \mathbb{B}^{[2]} = (T^2, g^{[2]})_r,\ \mathbb{B}^{[3]} = (T^3, g^{[3]})_r,\ \dots$$

that have identical radius but operate on "larger scales" in both state space and time.

Definition 8 (CA simulation)**.** Let $\mathcal{A} = (S^{\mathbb{Z}}, F)$ and $\mathcal{B} = (T^{\mathbb{Z}}, G)$ be CAs with local algebras $\mathbb{A} = (S, f)_r$ and $\mathbb{B} = (T, g)_r$ respectively. We say that $\mathcal{B}$ *can directly simulate* $\mathcal{A}$ if (at least) one of the following cases holds:

1. $\mathcal{A}$ is a sub-automaton of $\mathcal{B}$ (up to isomorphism)
2. $\mathcal{A}$ is a quotient automaton of $\mathcal{B}$ (up to isomorphism)
3. $\mathcal{A}$ is a product of some iterated powers of $\mathcal{B}$ (up to isomorphism).

We define the simulation relation $\mathcal{A} \preceq \mathcal{B}$ (we also write $\mathbb{A} \preceq \mathbb{B}$) to be the transitive closure of the direct simulation. If $\mathcal{A} \preceq \mathcal{B}$ we say that $\mathcal{B}$ *can simulate* $\mathcal{A}$.

In the following subsection, we briefly summarize the main result of Chapter 4. Therein, we show that certain classes of affine CAs are limited in terms of what they can simulate.

1.8.2 Simulation Limitations of Affine Cellular Automata

Additive automata are a much-studied class of CAs – in fact, they form one of the few classes of cellular automata that are amenable to algebraic analysis which yields rigorous results about their global dynamics [3, 54, 70, 96, 145]. Studying the sub-automata of additive CAs naturally leads to a broader class of *affine automata*, which we now introduce.

Definition 9 (affine CA, additive CA)**.** Let $\mathbb{F}$ be a finite field, V a finite-dimensional vector space over $\mathbb{F}$, and let $\mathcal{A} = (V^{\mathbb{Z}}, F)$ be a CA with local algebra $(V, f)_r$. We say that $\mathbb{A}$ (or $\mathcal{A}$) is *affine over* $\mathbb{F}$ if $f : V^{2r+1} \to V$ is an affine mapping between vector spaces over $\mathbb{F}$. In such a case, we can write f in the following form:

$$f(\mathbf{x}_{-r}, \dots, \mathbf{x}_r) = f_{-r}(\mathbf{x}_{-r}) + \cdots + f_r(\mathbf{x}_r) + \mathbf{c}, \tag{1.3}$$

where $f_i : V \to V$ is a linear mapping for each $-r \le i \le r$ and $\mathbf{c} \in V$ is a constant vector. The mapping f_i is called the *i-th component of f*. The class of all local algebras isomorphic to some affine local algebra over $\mathbb{F}$ with radius r is denoted as $\mathrm{AFF}_r^{\mathbb{F}}$.

In the special case when $f : V^{2r+1} \to V$ is a linear mapping between vector spaces over $\mathbb{F}$, we say that $\mathbb{A}$ (or $\mathcal{A}$) is *additive over* $\mathbb{F}$. In such a case, we can write f as in (1.3) with $\mathbf{c} = \mathbf{0}$. We denote the class of all local algebras isomorphic to some local algebra additive over $\mathbb{F}$ with radius r as $\mathrm{ADD}_r^{\mathbb{F}}$.

Example 10. Let $(\mathbb{F}_2^{\mathbb{Z}}, F)$ be the elementary CA 150 with local algebra $\mathbb{A} = (\mathbb{F}_2, f)_1$ where $f : \mathbb{F}_2^3 \to \mathbb{F}_2$ is defined as $f(x, y, z) = x + y + z$. Then, $\mathbb{A}$ is a CA additive over $\mathbb{F}_2$.

The results presented in Chapter 4 hold for affine CAs whose "outer" components are bijections – this is defined below.

Definition 11. Let $\mathbb{A} = (V, f)_r$ be an affine local algebra of a CA with radius r whose local rule f has components $f_{-r}, \dots, f_r$. We say that $\mathbb{A}$ is *left-permutive, witnessed by i,* if there exists $-r \le i \le r$ such that f_i is a bijection and f_k is the constant $\mathbf{0}$ mapping for all $k < i$. Similarly, $\mathbb{A}$ is *right-permutive, witnessed by j,* if there exists $-r \le j \le r$ such that f_j is a bijection and f_k is the constant $\mathbf{0}$ mapping for all $k > j$.

We write that $\mathbb{A} \in \mathrm{AFF}_{r;i,j}^{\mathbb{F}}$ if $\mathbb{A} \in \mathrm{AFF}_r^{\mathbb{F}}$ and at the same time $\mathbb{A}$ is left-permutive witnessed by i and right-permutive witnessed by j, $i < j$.

Now we can state the main result of Chapter 4:

Theorem 12. *Let p be a prime, $r \in \mathbb{N}$, and $-r \le i < j \le r$. Let $\mathbb{A}$, $\mathbb{B}$ be local algebras of cellular automata with radius r such that $\mathbb{B} \in \mathrm{AFF}_{r;i,j}^{\mathbb{F}_p}$. If $\mathbb{A} \preceq \mathbb{B}$, then $\mathbb{A} \in \mathrm{AFF}_{r;i,j}^{\mathbb{F}_p}$.*

This is a negative result that in its generality widely surpasses previous works [34, 35, 97] that studied simulation capacities of a highly specific class of additive CAs.

An important contribution of Chapter 4 is that therein, we propose a definition of CA simulation that is, to the best of our knowledge, the most general algebraic one so far. Moreover, we formalize all the notions regarding CA simulation in abstract algebraic language. This allowed us to see new connections to well-established algebraic fields that can provide powerful tools for analysing the CA simulation capacities. Whereas the proofs provided in this chapter do not rely on any sophisticated algebraic concepts, we remark that, as an example, Lemma 48 and Corollary 49 are a direct consequence of a deeper theorem by Smith [138] and Gumm [55] about Abelian algebras with a Maltsev term.

In what follows, we summarize the results of the last chapter of this thesis.

1.9 Simulation Capacities of Canonical Additive Automata

In Chapter 4, we have studied the simulation limitations of general classes of affine automata. As a consequence, automata with "bijective outer components" that are affine over different prime fields $\mathbb{F}_p$ are incomparable with respect to the simulation relation $\preceq$. However, this gives us no information about the relation of two particular CAs within the same class $\mathrm{AFF}_{r;i,j}^{\mathbb{F}_p}$. In Chapter 5, we complement the previous results by analysing the simulation capacity of individual additive automata. More precisely, given an additive CA $(\mathbb{F}_p, f)_r$, p prime, that satisfies certain conditions, we give an explicit characterization of all the automata it can simulate. As a special case, we describe the simulation capacities of any additive CA with radius $r = 1$ and local algebra of the form $(\mathbb{F}_p, f)_1$ (over an arbitrary field $\mathbb{F}_p$).

Rather than describing the result in its full generality here, we state a corollary where most technical details can be omitted.

Corollary 13. *Let p be a prime and consider the class* CA_p *of all CAs with p states and radius $r = 1$. Let $\mathbb{B} = (\mathbb{F}_p, f)_1$ be the local algebra of an additive CA; $f(x, y, z) = a_{-1}x + a_0y + a_1z$ with at least two of the coefficients $a_{-1}, a_0, a_1 \in \mathbb{F}_p$ non-zero. Then, one of the following cases holds:*

(1) $a_0 \neq 0$. Then within the class CA_p*, $\mathbb{B}$ can only simulate itself (up to isomorphism).*

(2) $a_0 = 0$. Then, within the class CA_p*, $\mathbb{B}$ can simulate (up to isomorphism) exactly itself and the automaton with local algebra $\mathbb{B}' = (\mathbb{F}_p, f')_1$ where $f'(x, y, z) = a_{-1}^2x + 2a_{-1}a_1y + a_1^2z$. Note that $\mathbb{B} \cong \mathbb{B}'$ if and only if $p = 2$.*

The corollary implies new results about simulation limitations even for the very well-studied case of elementary CAs. We describe one such result in the following example.

Example 14. We consider elementary CA 90 that is defined as $\mathrm{ECA}_{90} = (\mathbb{F}_2, f)_1$ where $f(x, y, z) = x + z$ and ECA 165 defined as $\mathrm{ECA}_{165} = (\mathbb{F}_2, g)_1$ where $g(x, y, z) = x + z + 1$. Clearly, $\mathrm{ECA}_{90} \cong \mathrm{ECA}_{165}$ via the mapping that exchanges 0 and 1. Then, Corollary 13 implies that within the class of elementary CAs, ECA 90 can only simulate itself and ECA 165.

Proof. Clearly, ECA 90 satisfies the assumptions of Corollary 13 (2). In the case of a CA additive over $\mathbb{F}_2$, the local algebras $\mathbb{B}$ and $\mathbb{B}'$ coincide (using the notation in the corollary). Thus, the corollary yields that ECA 90 can only simulate elementary CAs whose local algebras are isomorphic to ECA_{90}. In this case, we have only one bijection on $\mathbb{F}_2$ that is not identity, and this yields the ECA 165. □

This concludes the introduction to the thesis where we gave an outline of the thesis and briefly explained the results of each chapter while also providing a concise overview of previous work to motivate our results. In what follows, we present the four chapters.

2. Classification of Discrete Dynamical Systems Based on Transients

2.1 Introduction

There are many approaches to searching for systems capable of open-ended evolution. One option is to carefully design a model and observe its dynamics. Iconic examples include [24, 115, 128, 141]. However, as we lack any universally accepted formal definition of open-endedness or complexity, there is no formal method of proving the system is indeed "interesting". Conversely, lacking definitions of such key terms, it seems extremely difficult to design such models systematically.

Approaching the problem of searching for open-endedness bottom up, we can define a suitable classification of dynamical systems that would help us identify a region of complexity. An ideal classification would be based on a formally defined property, be effectively computable, and help us automatically search for complex systems possibly capable of modeling artificial evolution.

Over the years, many different metrics have been introduced to study systems' dynamics. As an example, cellular automata were studied in terms of their space-time dynamics observations [158], their space-time compression sizes [165], via their actions on probability measures [56], the Z-parameter [162], or the lambda parameter [83]. Most of such approaches show that the complex region of systems lies somewhere "in between" the ordered and chaotic phase.

In this paper, we introduce a novel method of classifying complex systems based on estimating their asymptotic average computation time with increasing space size. The key result is that the classification identifies a region of systems that seem to be at a phase transition between ordered and chaotic behavior. Across various classes of discrete systems, we demonstrate that complex systems such as cellular automata computing nontrivial tasks, universal Turing machines, or random Boolean networks at a critical phase belong to this region. Even though we are far from characterizing complexity, we hope this method helps us understand which formally defined properties correlate with it.

2.2 Transient Classification: A General Method

We first introduce the basic principle of the classification based on transients, which can be applied to any deterministic discrete space and time dynamical system (DDDS). In subsequent sections, we describe the results of the classification applied to cellular automata, Turing machines, and random Boolean networks to demonstrate its use across different classes of discrete dynamical systems.

Basic Notions

Let us consider a generic deterministic discrete system D operating on finite space, characterized by a tuple $D = (S, F)$ where S is a finite set of configurations and $F : S \to S$ is a global transition function governing the dynamics of the system. We define the *trajectory of a configuration* $u \in S$ as the sequence

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

As S is finite, every trajectory eventually becomes periodic. We call the preperiod of this sequence the *transient of initial configuration* u and denote its length by t_u. More formally, we define t_u to be the smallest positive integer i, for which there exist $j \in \mathbb{N}$, $j > i$, such that $F^i(u) = F^j(u)$. The periodic part of the sequence is called an *attractor*. The *phase-space* of $D = (S, F)$ is an oriented graph with vertices $V = S$ and edges $E = \{(u, F(u)), u \in S\}$. Such a graph is composed of components, each containing one attractor and multiple transient paths leading to the attractor. The phase-space completely characterizes the dynamics of the system. However, it is infeasible to describe when the configuration space S is large. Given a DDDS D, we will focus on studying its *average transient length*

$$T(D) = \frac{1}{|S|} \sum_{u \in S} t_u.$$

We describe the method of estimating a system's average transient length together with the error analysis in section Average Transients: Error Estimate.

The Main Principle

Suppose we have a sequence of DDDSs

$$D_1 = (S_1, F_1), D_2 = (S_2, F_2), D_3 = (S_3, F_3), \ldots$$

operating on configuration spaces of growing size. That is, $F_i : S_i \to S_i$ and $|S_i| < |S_{i+1}|$ for each i. For instance, the sequence can be given by a cellular automaton with a fixed local rule, operating on a finite cyclic grid of growing size.

Our goal is to estimate the asymptotic growth of the systems' average transient lengths, as shown in Figure 2.1.

Discrete system	**Average transient length**	**asymptotic growth** ↓
$D_1 = (S_1, F_1)$	$T(D_1)$	
$D_2 = (S_2, F_2)$	$T(D_2)$	
$D_3 = (S_3, F_3)$	$T(D_3)$	
$D_4 = (S_4, F_4)$	$T(D_4)$	
$\vdots$	$\vdots$	

Figure 2.1: Diagram depicting the asymptotic growth of average transient lengths of a sequence of discrete systems.

In practice, we generate a finite part of the sequence $\big(|S_i|, T(D_i)\big)_{i=1}^{B}$ where B is an upper bound imposed by our computational limitations and examine different regression fits of the data. Specifically, we evaluate the fit to constant, logarithmic, linear, polynomial, and exponential functions. We pick the best fit with respect to the R^2 score and obtain the classes: Bounded, Log, Lin, Poly, and Exp. If the score of the fit to all such functions is low (i.e., $R^2 < 85\%$), we say the system is Unclassified. Surprisingly, we found a very good fit to one of the classes with $R^2 > 90\%$ for most DDDSs we examined. The trend we have observed, which seems to hold across various families of DDDSs, is shown in Figure 2.2. We describe it in more detail for each family in the subsequent sections.

We do not claim our method determines the true asymptotic behavior of a system; it is merely a possible interpretation of the method. For some systems, the transient growth might correspond to more complicated functions but we have deliberately chosen the classes to be quite robust and coarse to have clearer boundaries between them. The uncertainty of the true asymptotic growth is especially relevant for the Lin and Poly Classes which identify the critical

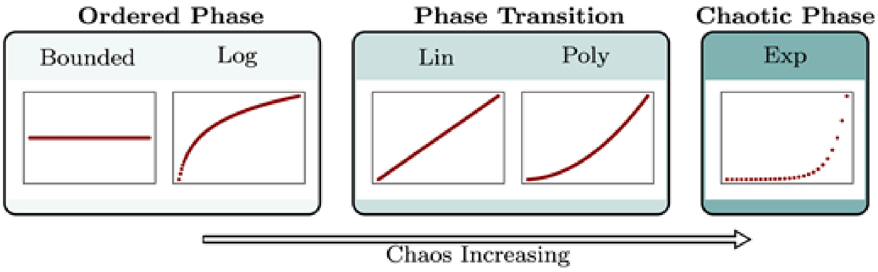


Figure 2.2: General trend of the transient classification results.

phase transition region. Such systems might turn out to be logarithmic or exponential, and it might be the case that we have not detected this due to our limited data. However, in such a case, such systems would exhibit significantly slower convergence to their asymptotic behavior than systems in other classes which is a typical property of a system at a phase transition.

Computational Interpretation

In non-classical models of computation [142], the process of traversing a discrete system'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. For cellular automata (CAs), this is explored in [72]. The average transient growth then corresponds to the average computation time of the system[1]. Therefore, we can interpret our goal as trying to estimate systems' **asymptotic average computation time**. DDDSs with bounded transient lengths can only perform trivial computation. On the other hand, DDDSs with exponential transient growth can be interpreted as inefficient computation models.

In the context of artificial evolution, we can view the global transition rule of a DDDS 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 DDDSs 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 configuration space tends to be very large. This motivates our study of the growth of average transient lengths rather than the maximum ones.

Average Transients: Error Estimate

Let us fix a DDDS $D = (S, F)$ operating on a large configuration space, e.g., $|S| \gg 2^{100}$. In such case, computing the average transient length μ is infeasible. Thus, we uniformly randomly sample initial configurations $u_1, u_2, \ldots, u_m$ and estimate μ by $\frac{1}{m}\sum_{i=1}^{m} t_{u_i}$. It remains to estimate the number of samples m so that the error $|\frac{1}{m}\sum_{i=1}^{m} t_{u_i} - \mu|$ is reasonably small.

More formally, for $D = (S, F)$, let (S, P) be a discrete probability space where S is the set of all configurations and P is a uniform distribution. Let $X : S \to \mathbb{N}$ 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 $var(X)$. It can be easily shown that $E(X) = \mu$. Our goal is to obtain a good estimate of $E(X)$ by the Monte Carlo method ([119]).

Let $(X_1, X_2, \ldots, X_m)$ be a random sample of iid random variables, $X_i \overset{d}{=} X$ for all i. Let $\mu^{(m)} = \frac{1}{m}\sum_{i=1}^{m} X_i$ be the sample mean and $\sigma^{(m)} = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}(X_i - \mu^{(m)})^2}$ the sample

[1] Here, the computation time is understood in the abstract sense; as the number of iterations of the transition function.

standard deviation. As X is a mapping from a finite set, $var(X) < \infty$, and thus we have by the Central limit theorem the convergence to a normal distribution. The interval

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

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

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

From the nature of our data, both the values $E(X) = \mu$ and $var(X)$ tend to grow with increasing size of the configuration space. Therefore, to employ a general method of estimating the number of samples, we normalize the error by the sample mean and consider $\frac{|\mu-\mu^{(m)}|}{\mu^{(m)}}$. Therefore for m sufficiently large such that

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

we have that $\mu^{(m)}$ differs from μ 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 most elementary CA this was satisfied typically after 400 data points were sampled). We approximate the uniform random sampling of initial configurations using Python's numpy.random library.

2.3 Cellular Automata

2.3.1 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 local update function depending only on the states of automata in their local neighborhood. A formal definition can be found in [76].

CA were first studied as models of self-replicating structures [82, 112, 129]. Subsequently, they were examined as dynamical systems [58, 63, 150], or as models of computation [103, 146]. Being so simple to simulate, yet capable of complex behavior and emergent phenomena [28, 59], CA provide a convenient tool to examine the key, yet undefined notions of complexity and emergence.

Basic Notions

We study the simple class of *elementary cellular automata* (ECAs), which are one-dimensional CAs with two states $\{0,1\}$ and neighborhood of size 3. Each ECA is given by a local transition function $f : \{0,1\}^3 \to \{0,1\}$. Hence, there are only 256 of them. The size of this CA class is the reason to make it our first case of study. One can simply explore it by studying the dynamics of every single ECA.

We identify each local rule f determining an ECA with the *Wolfram number* of the ECA defined as:

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

We will refer to each ECA as a "rule k" where k is the corresponding Wolfram number of its underlying local rule.

We will consider the CA to operate on finite grids with periodic boundary conditions. Hence, given a local rule f and a grid size n, we obtain a configuration space $\{0,1\}^n$ and a global update function $F : \{0,1\}^n \to \{0,1\}^n$.

Let $(\{0,1\}^n, F)$ be an ECA operating on a grid of size n and $(u, F(u), F^2(u), \ldots)$ a trajectory of a configuration $u \in \{0,1\}^n$. The space-time diagram of such a simulation is obtained by plotting the configurations as horizontal rows of black and white squares (corresponding to states 1 and 0) with a vertical axis determining the time, which is progressing downwards.

We note that properties of CA phase-spaces were examined among others by Wuensche in [162]. Precisely for this purpose, a software was designed by Wuensche in [161].

2.3.2 History of CA Classifications

An ideal classification would be based on a rigorously defined and efficiently measurable property, identifying a region of systems with interesting behavior. In this section, we describe three qualitatively different classifications of ECAs, and subsequently, we will compare our results to them.

Wolfram's Classification

The most intuitive and simple approach to examining the dynamics of CAs is to observe their space-time diagrams. This method was particularly proclaimed by Wolfram, e.g., in [158]. Therein, he established an informal classification of CA dynamics based on such diagrams. He distinguishes the following classes, which are shown in Figure 2.3.

Class 1	... quickly resolves to a homogenous state
Class 2	... exhibits simple periodic behavior
Class 3	... exhibits chaotic or random behavior
Class 4	... produces localized structures that interact with each other in complicated ways

The main issue is that we have no formal method of classifying CAs in this way. In fact, this problem is in general undecidable [29]. Moreover, the behavior of some CAs can vary with different initial configurations. An example being rule 126 which oscillates between Class 2 and Class 3 behavior, as shown in Figure 2.4. The transient classification we present in this paper deals with both these issues.

Zenil's Classification

In the first part of his paper [165], Zenil studied the compression size of the space-time diagrams of each ECA simulated for a fixed amount of steps. For the classification, he examines the simulations from a particular initial configuration (a single one surrounded by zeros). Using a clustering technique, he obtained two classes distinguishing between Wolfram's simple classes 1 and 2 and complex classes 3 and 4. We show our reproduction of Zenil's results in Figure 2.5.

His method nicely formalizes Wolfram's observations of the space-time diagrams. However, the results depend on the choice of initial conditions as well as the grid size, data representation, and the compression algorithm. We conducted multiple experiments presented in Figure 2.6, which suggest that Zenil's results might be sensitive to the choice of such parameters. We note that he addresses the sensitivity to the choice of the initial configurations in the second part of his paper [165].

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. Moreover, the data representation causes the extension of this method to more general dynamical

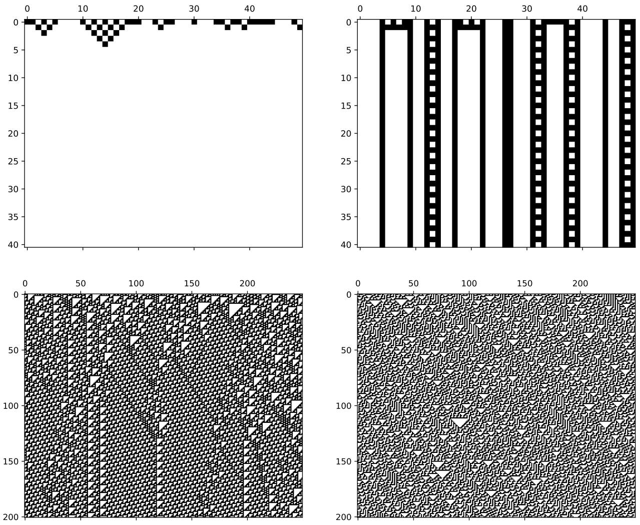


Figure 2.3: Space-time diagrams of rules from each Wolfram's class. Class 1 rule 32 is on top left, Class 2 rule 108 on top right, Class 4 rule 110 on the bottom left, and Class 3 rule 30 on the bottom right.

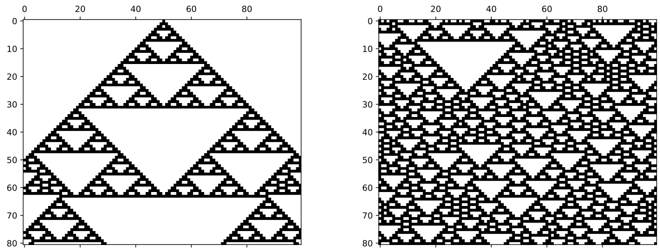


Figure 2.4: On the left, rule 126 is simulated with an initial condition consisting of a single 1 bit padded with 0's. On the right, the same rule is simulated with a random initial configuration.

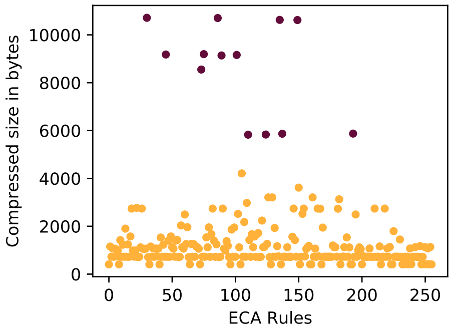


Figure 2.5: Reproduction of Zenil's results in [165]. The purple cluster corresponds to the interesting Class 3 and 4 rules, the yellow cluster to the rest.

systems to be problematic; for example, using gzip to compress space-time diagrams of a 2D cellular automaton is suboptimal.

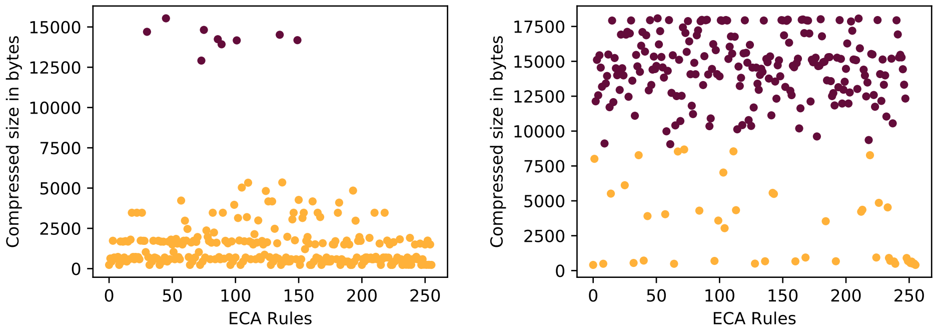


Figure 2.6: Graphs representing the results of Zenil's method when different parameter values were used. They demonstrate the possible sensitivity of the results are. On the left, the ECAs were simulated for longer time, which caused complex rules 110, 124, 137, and 193 to no longer belong to the "interesting" purple cluster. On the right, the ECAs were simulated from a fixed, randomly chosen initial condition. In such case, we obtain entirely different clusters.

Wuensche's Z-parameter

In [162], Wuensche chose an interesting approach by studying the ECA's behavior when reversing the simulations and computing the preimages of each configuration. He introduces the Z-parameter, representing the probability that a partial preimage can be uniquely prolonged by one symbol, and suggests that Class 4 CAs typically occurs at Z ≈ 0.75. However, no clear classification is formed. The crucial advantage is that the Z-parameter depends only on the CA's local rule and can be computed effectively. It is, however, questionable whether studying only the local rule could describe the overall dynamics of a system sufficiently well.

We note that transients of CAs have been examined, as in [162] or [57]. However, we are not aware of an attempt to compare the asymptotic growth of transients for different ECA.

2.3.3 Transient Classification of ECA

For each ECA given by a local rule f, we consider the sequence of systems

$$D_3 = (\{0,1\}^3, F_3), D_4 = (\{0,1\}^4, F_4), \ldots$$

which represent the ECAs operating on grids of growing size. We can apply the transient classification to this sequence, as described in Section Transient Classification: A General Method to estimate the asymptotic growth of the average transient lengths for each ECA.

We consider all 256 ECAs up to equivalence classes obtained by changing the role of "left" and "right" neighbor, the role of 0 and 1 state, or both. It can be easily shown that automata in the same equivalence class have isomorphic phase spaces for any grid size. Thus, they perform the same computation. This yields 88 effectively different ECAs, each being a representative with the minimum Wolfram number from its corresponding equivalence class. In this section, we present the classification of the 88 unique ECAs based on their asymptotic transient growth.

Results

We obtained a surprisingly clear classification of all the 88 unique ECAs with four major classes corresponding to the bounded, logarithmic, linear, and exponential growth of average transients. 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. See Figure 2.7.

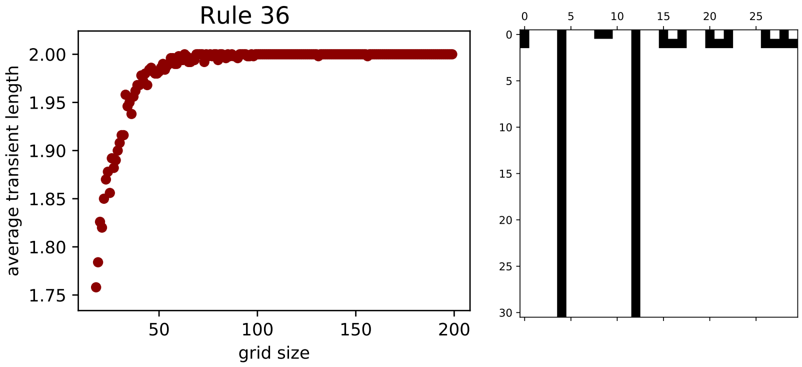


Figure 2.7: Bounded Class rule 36. The average transient plot is on the left, the space-time diagram on the right.

Log Class: 39/88 rules (44.32%). The largest ECA class exhibits logarithmic average transient growth. The event of two cells at a large distance "communicating" is improbable for this class. See Figure 2.8.

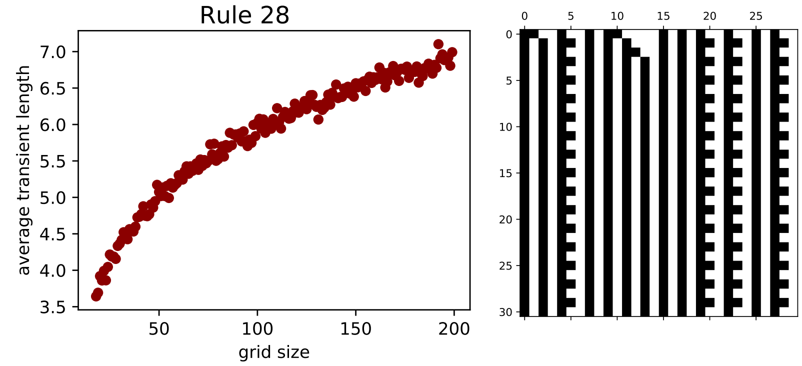


Figure 2.8: Log Class rule 28. The average transient plot is on the left, the space-time diagram on the right.

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 this class contains two rules known to have a nontrivial computational capacity: rule 184, which computes the majority of black and white cells, and rule 110, which is the only ECA so far proven to be Turing complete ([27]).

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 had only a very gradual incline. In fact, the only two rules with such slope greater than 1, rules 110 and 62, seem to be the ones with the most interesting space-time diagrams. See Figure 2.9.

We are aware that average transients of rules in Lin Class might turn out to grow logarithmically or exponentially given enough data samples. In such a case, the rules in Lin Class show a significantly slower convergence to their asymptotic behavior, which supports the hypothesis that they belong to a phase transition region.

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.

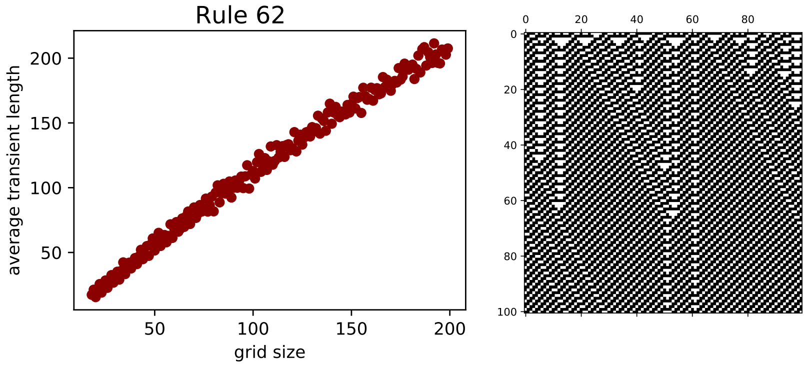


Figure 2.9: Lin Class rule 62. The average transient plot is on the left, the space-time diagram on the right.

The rules with the fastest growing transients are 45, 30 and 106. See Figure 2.10.

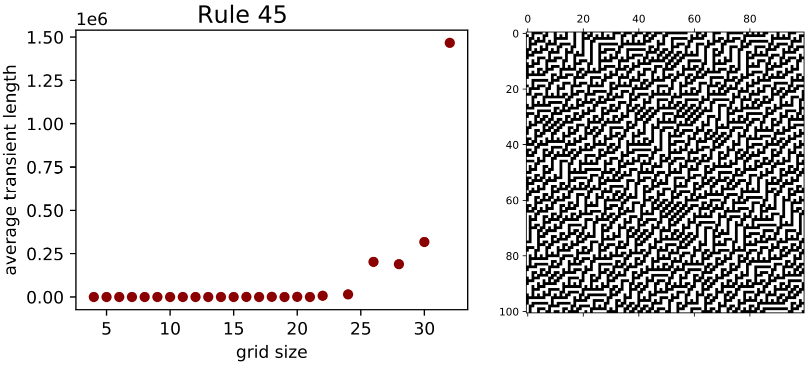


Figure 2.10: Exp Class rule 45. The average transient plot is on the left, the space-time diagram on the right.

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 [96] that the transient lengths of rule 90 depend on the largest power of 2, which divides the grid size. Therefore, the measured data did not fit any of the functions above but formed a rather specific pattern. See Figure 2.11.

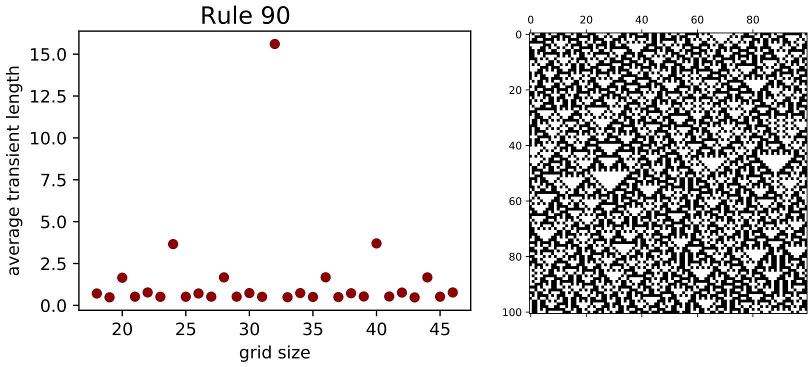


Figure 2.11: Affine Class rule 90. The average transient plot is on the left, the space-time diagram on the right.

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 structures. We could say such rules oscillate between easily predictable behavior and chaotic behavior. Their average transients and periods grow quite fast. See Figure 2.12.

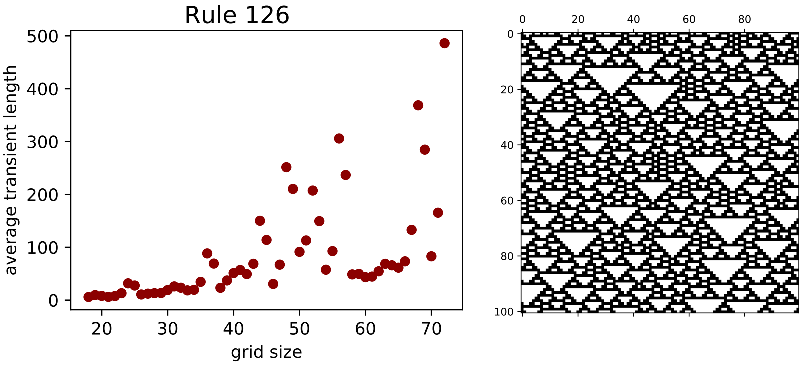


Figure 2.12: Fractal Class rule 126. The average transient plot is on the left, the space-time diagram on the right.

2.3.4 Discussion

We have also tried to measure the asymptotic growth of the average attractor size a_u, $u \in \{0,1\}^n$ as well as the average rho value defined as $\rho_u = t_u + a_u$. This, however, produced data points, which 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, and this causes oscillations. We conclude that such phase-space properties are not suitable for this classification method.

Exhaustive comparison for each ECA is presented in Table 2.13.

Wolfram's Classification – Discussion The significance of our results for ECAs 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 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 roughly correspond to it. Compared to the transient classification, it is, however, less fine-grained. Moreover, it contains some arbitrary parameters, such as the data representation and compression algorithm used. In addition, it uses a clustering technique, which 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.

Another important difference is that Zenil observed the simulations from a fixed initial configuration; i.e., he examined the local dynamics of ECA. In contrast, the transient classification is studying their global dynamics.

Classification Comparison				
ECA	**Transient**	**Wolfram**	**Zenil**	**Wuensche**
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
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 2.13: Comparing classifications of the 88 unique ECA.

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 (rule 110, 62, and 25) have this Z value precisely. 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 CA

So far, we have examined the toy model of ECA. Transient classification's true usefulness would stem from its application to more complex CAs, where it could be used to discover automata with interesting behavior.

Therefore, we applied the classification on a subset of two-dimensional CAs with a 3×3

neighborhood and three states to see whether 2D automata would still exhibit such clear transient growths.

We work with 2D CAs operating on a finite square grid of size $n \times n$. We consider the topology of the grid to be that of a torus for each cell to have a uniform neighborhood. We estimated the average transient length and measured the asymptotic growth with respect to n (i.e., the size of the square grid's side). 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 a square. As there are still 3^{2861} such symmetrical 2D CAs, we randomly sampled 10 000 of them.

For such a large space, we cannot examine each CA individually. Therefore, we fit the average transient growth to bounded, logarithmic, linear, polynomial, and exponential functions to obtain the classes Bounded, Log, Lin, Poly, and Exp. If none of the fits gives a good enough score (i.e., $R^2 > 85\%$), then we mark the corresponding CAs as unclassified. 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 resources restriction or rather strict conditions we imposed on a good regression fit. In this large space of 2D CAs, the Exp Class seems to dominate the rule space. Another interesting aspect in which 2D CAs differ from the ECAs is the emergence of rules in the Poly Class; the transients of such rules grow approximately quadratically. Moreover, our results suggest that the occurrence of Bounded Class CAs in 2D is much scarcer as we found no such CA in our sample. See Table 2.1.

Classification of 2D 3-state CAs (10 000 samples)	
Transient Class	Percentage of CA
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 CA.

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 exhibits the emergence of various local structures. For automata with a more gradual incline, such structures seem to die out quite fast. 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[2].

This suggests 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 the method designed by [26] where interesting novel automata were discovered by measuring growth of structured complexity using a data compression approach.

[2] `http://bit.ly/trans_class`

2.3.6 Transients Classification of Other Well Known CA

We were interested whether some well-known complex automata from larger CA spaces would conform to the transient classification as well. As we show in this section, the result is positive.

Game of Life As the left plot in Figure 2.14 suggests, the Turing complete Game of Life ([46]) seems to fit the Lin Class. This is confirmed by the linear regression fit with $R^2 \approx 98.4\%$.

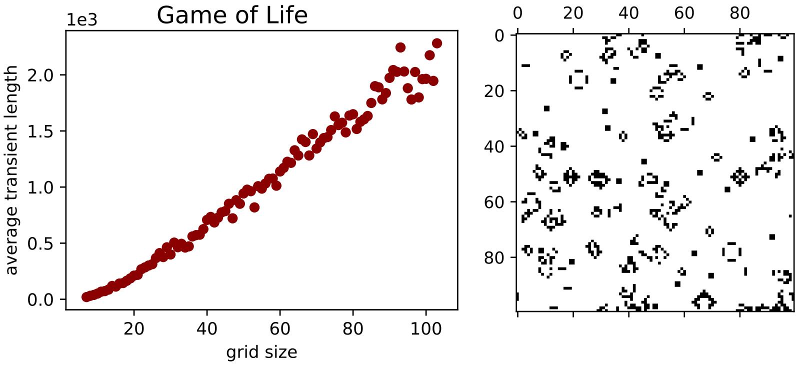


Figure 2.14: Game of Life. The average transient growth plot is on the left. On the right, we show a space-time diagram at time $t = 200$ started from a random initial configuration.

Genetically Evolved Majority CA In [105], Mitchell et al. 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 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. See Figure 2.15.

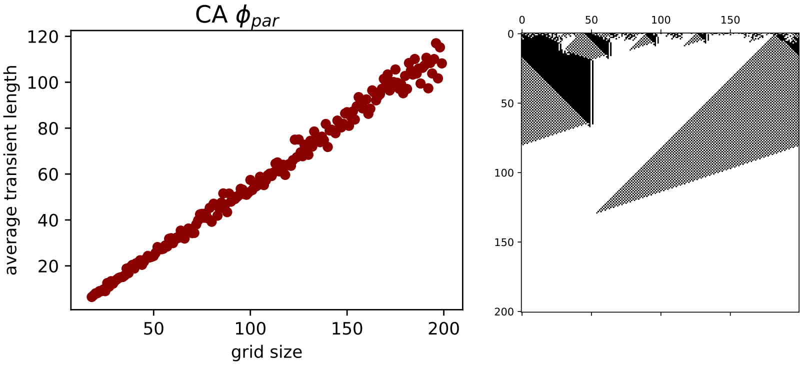


Figure 2.15: Cellular automaton ϕ_{par}. The average transient growth plot is on the left. On the right, we show a space-time diagram simulated from a random initial configuration.

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 $S = \{0, 1, 2\}$.

In [159], Wolfram presents a list of possibly complex CAs from this class. We applied the transient classification to such CA 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 2.16 where the linear regression fit has $R^2 \approx 97.63\%$.

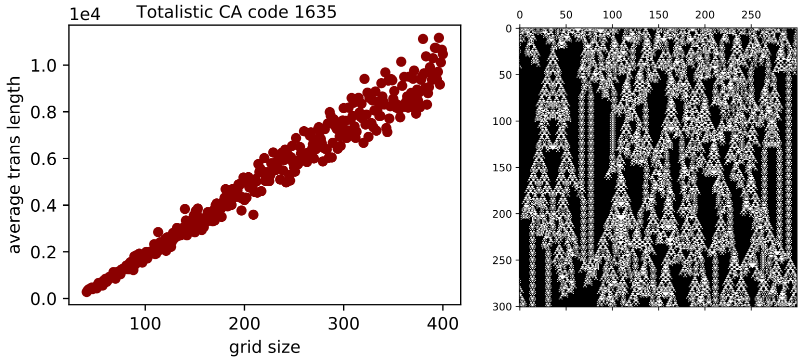


Figure 2.16: Totalistic cellular automaton with code 1635. The average transient growth plot is on the left. On the right, we show a space-time diagram of the evolution from a random initial configuration.

2.4 Turing Machines

In order to demonstrate the generality of the transient classification method, we further used it to examine the dynamics of Turing machines. In this section, we present the classification results.

2.4.1 Introducing Turing Machines

Informally, a Turing machine (TM) consists of an infinite tape divided into cells and a movable reading head that scans one cell of the tape at a time. Every cell contains a symbol from some finite alphabet A, and the Turing machine is at an internal state from a finite set S. Depending on the symbol the head is reading and on its internal state, the Turing machine changes its internal state, rewrites the symbol on the tape, and either moves one cell to the left, right or stays in place. Turing machines represent the most classical model of computation; the Church-Turing thesis states that "effectively calculable functions" are exactly those that can be realized by a Turing Machine [148]. For a formal definition of Turing machines as well as a great introduction to computability theory, see [139].

In this paper, we will consider deterministic Turing machines with one tape. S will always denote the finite set of internal states, A will denote the finite set of tape symbols. To ensure the Turing machine operates on a finite grid, as in the case of CAs, we will consider a tape of finite size with periodic boundary conditions. Therefore, each Turing machine with S and A operating on a tape of size n gives rise to a global update function

$$F : A^n \times \{1, 2, \dots, n\} \times S \to A^n \times \{1, 2, \dots, n\} \times S,$$

where each configuration specifies the content of the tape, the position of the head, and the internal state. Thus, we can apply the transient classification to it. We emphasize the non-traditional notion of the halting computation that we consider here. Classically, a Turing machine is considered to halt when it enters an attractor of size 1; that is when it does not change the tape's content, the head's position, or its internal state anymore. In our case, using the interpretation

$$\begin{aligned} \text{transients} &\approx \text{computation} \\ \text{attractors} &\approx \text{memory} \end{aligned}$$

we consider a Turing machine to halt whenever it enters **any** attractor. This is a much weaker notion of halting.

We will depict the space-time diagrams of TM computation as a matrix, each row corresponds to the content of the tape at subsequent time steps, and time is progressing downwards. As opposed to CAs with their inherently parallel nature, TMs are sequential computational models. Thus, at each time step, only one symbol on the tape is changed. To produce space-time diagrams comparable to those produced by CAs, we only depict tape contents at every n-th step where n is the size of the tape. This helps us to intuitively recognize the chaotic dynamics of TMs, an example is in Figure 2.17.

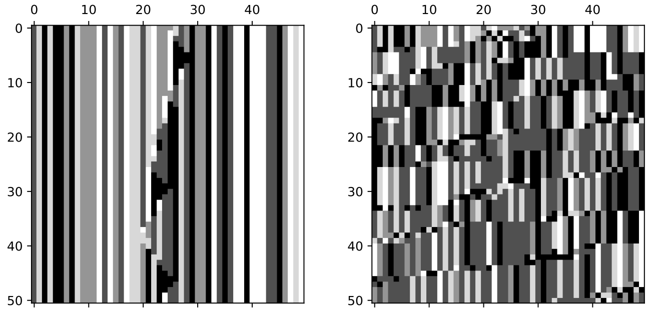


Figure 2.17: Space-time diagrams of a 6 symbol, 5 state TM in the Exp Class. Classical space-time diagram is shown on the left. On the right we show the space-time diagram depicting the content of the tape after every 50 steps of computation on a tape of size 50.

To the best of our knowledge, we know of no prior work examining the transients of Turing machines operating on cyclic tapes.

2.4.2 Transient Classification of Turing Machines

We have studied "small" TMs with the number of states $|S|$ ranging from 4 to 8 and the number of alphabet symbols $|A|$ ranging from 2 to 5. For every such combination of values $|S|$ and $|A|$, we have randomly generated 100 transition functions of TMs and computed each of the TM's average transient length estimate for cyclic tapes of sizes ranging from 20 to 400.

Results

For all the considered values of $|S|$ and $|A|$, more than 90% of the TMs were successfully classified using the transient classification method. We discuss a particular example in more detail below.

TMs with 7 states and 4 symbols As an example, we present classification results of 100 Turing machines with 7 states and 4 tape symbols. The results are summarized in Table 2.2.

Bounded Class (41/100 TMs). In this space of rather "small" Turing machines, the Bounded Class seems to dominate the space. TMs in the Bounded Class halt in time independent on the tape size. Therefore, for such TMs it seems improbable to perform any nontrivial computation on both the finite and infinite tape.

Log Class (2/100 TMs) The Log Class seems to be relatively small across all the TM classes we have examined. Here, the event that a TM head will read the whole input from the tape is improbable for large tape sizes.

Classification of TMs with 7 states and 4 symbols	
Transient Class	Percentage of TMs
Bounded Class	41%
Log Class	2%
Lin Class	28%
Poly Class	13%
Exp Class	15%
Unclassified	1%

Table 2.2: Classification of 100 randomly sampled TMs with 7 states and 4 symbols.

Lin Class (28/100 TMs.) Let us consider a TM with trivial dynamics, which, given any input configuration, traverses each cell one by one and changes the state of each cell to the state $0 \in S$. After all cells enter this state, the computation halts. Such trivial behavior could be realized in constant time by a CA, though for a TM it takes at least n steps where n is the size of the tape. Hence, some Turing machines in the Lin Class exhibit periodic or simple dynamics. This emphasizes the fact that being contained in a Lin or Poly Class seems to be a necessary condition for complexity, not a sufficient one. See Figure 2.18.

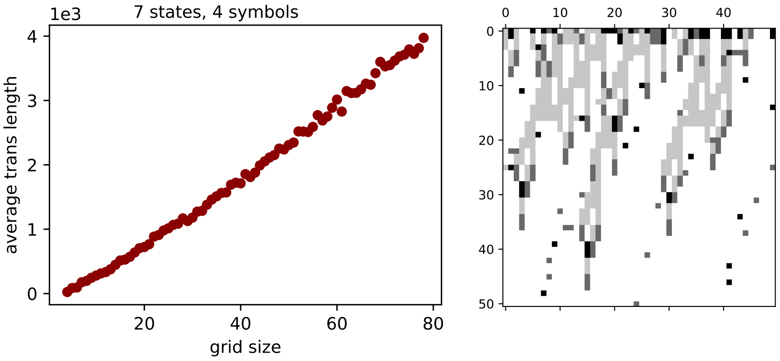


Figure 2.18: Example of a Turing machine with 7 states, 4 symbols in the Lin Class. Its space-time diagram seems to exhibit nontrivial behavior.

Nevertheless, we have observed TMs in the Lin Class whose space-time diagrams seem to contain some higher-level structures.

Poly Class (13/100 TMs.) In the Poly Class we have also observed TMs producing some higher-order structures.

Exp Class (15/100 TMs.) We find it interesting that once only every n-th row of the space-time diagram (n being the tape size) is depicted, the space-time diagrams of TMs in the Exp Class resemble the space-time diagrams of chaotic CAs. See Figure 2.19.

As in the case of CAs, we are aware of the fact that the true asymptotic behavior of TMs in Lin or Poly Class might turn out to be logarithmic or exponential. In such a case, the systems in these classes would need significantly longer time to converge to their typical long-term behavior, which is a typical property of systems at a phase-transition.

2.4.3 Transient Classification of Universal TMs

Without much doubt, universal Turing machines are considered complex. We have estimated the asymptotic average computation time of 7 universal Turing machines with a small number

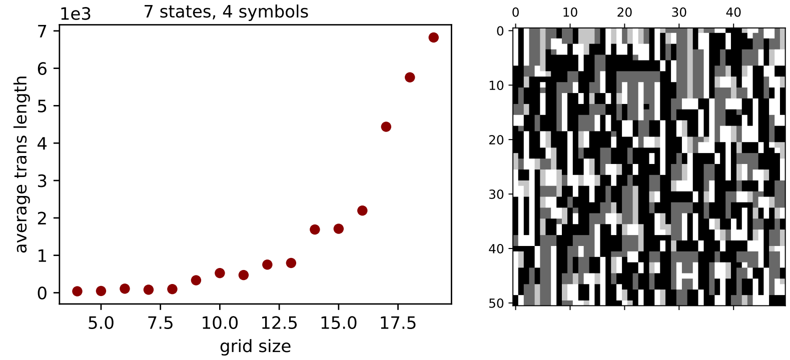


Figure 2.19: Example of a Turing machine with 7 states, 4 symbols in the Exp Class.

of states and symbols constructed by Rogozhin in [131]. All of them were successfully classified, 6 belonging to the Poly class and 1 to the Lin Class. An example is shown in Figure 2.20.

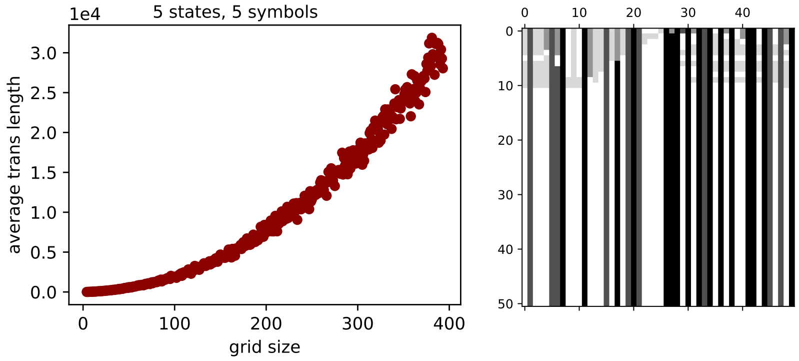


Figure 2.20: Universal TM with 10 states and 3 symbols belonging to the Poly Class. Again, every 50th step of the computation is shown in the space-time diagram.

Such results agree with the ones obtained for CAs, and support the hypothesis that complex dynamical systems belong to the Lin or Poly class.

2.5 Random Boolean Networks

Random Boolean networks form a very wide class of discrete dynamical systems that contains both CAs and TMs. In this section, we show that dynamical systems from this general class also conformed to our classification method.

2.5.1 Introducing Random Boolean Networks

The classical $N - K$ random Boolean network (RBN) is given by an oriented graph with N nodes, each one of them having exactly K edges pointing toward it. In addition, each node is equipped with a Boolean function of K variables. Every node can have the value of either 0 or 1, therefore the configuration space is exactly $\{0,1\}^N$. To update a particular configuration of the network, the values of all nodes are changed in parallel, according to the outputs of their corresponding Boolean functions. This gives rise to a global update rule $F : \{0,1\}^N \to \{0,1\}^N$. For a concise introduction to RBNs see [47].

RBNs were first introduced by Kaufmann [79] as models of gene regulatory networks. Classically, the nodes are interpreted as genes of a particular organism; their value represents whether the gene is “turned on” or “off”. In this setting, the attractors of the network represent different cell types of the organism. Over the years, the networks have been widely studied as models of cell differentiation [64], immune response [80], or neural networks [160]. The measures of criticality in RBNs were studied in [92, 123, 154]. A great overview on RBNs is for example [71].

Critical Behavior in RBNs

RBNs are generic in the sense that both the connections of nodes and the Boolean functions are chosen uniformly at random. This makes it possible to analytically study the properties of a typical $N-K$ network. Indeed, different approaches [38, 91]) lead to the same description of phase transitions in RBNs. We describe the results briefly below, as we will use them in our experiments.

We will describe a slightly more general model of RBNs. We consider a non-uniform connectivity of the nodes – for a network of size N, we will assign to each node $i \in \{1, \dots, N\}$ the connectivity $K_i \in \mathbb{N}$ and a Boolean function f_i of arity K_i. Such a network is parametrized by the mean connectivity $\langle K \rangle = \frac{1}{N} \sum_{i=1}^{N} K_i$. We also introduce the Boolean function sampling bias $p \in (0, 1)$. That is, we will sample the Boolean functions so that for all i the probability that $f_i(x_1, \dots x_{K_i}) = 1$ is p. In [38], the authors have analytically determined the edge of chaos for RBNs depending on the mean connectivity parameter $\langle K \rangle$ and Boolean function bias p. By studying the evolution of the distance between two randomly generated initial configurations over time, they have shown that the critical values of $\langle K \rangle$ and p are exactly those satisfying

$$\langle K \rangle = \frac{1}{2p(1-p)}. \tag{2.2}$$

They obtain the following phases of RBN behavior.

$$\begin{aligned}
&\text{Ordered Phase } \dots \text{RBNs with } \langle K \rangle < \frac{1}{2p(1-p)} \\
&\text{Critical Phase } \dots \text{ RBNs with } \langle K \rangle = \frac{1}{2p(1-p)} \\
&\text{Chaotic Phase } \dots \text{ RBNs with } \langle K \rangle > \frac{1}{2p(1-p)}
\end{aligned}$$

The curve given by (1) is shown in Figure 2.21.

In the next section, we support the analytical results by showing that the transient classification clearly distinguishes between the ordered, critical, and chaotic regions.

Phase-Space Properties of RBNs

The generic nature of $N-K$ RBNs makes it possible to analytically study their global dynamics. This is a key difference between CAs and RBNs: CAs have a very particular architecture with only local connections and a uniform local transition rule. Therefore, the mean-field approximation methods of phase-space properties used for a “typical” $N-K$ RBN would not be as easy to apply to CAs.

With the classical interpretation of RBNs as gene regulatory networks where attractors represent different cell types, most of the focus has been on analyzing the number and size of the attractors for different values of N and K. We briefly summarize some results related to our experiments below. For a more detailed discussion see [71].

Ordered Phase In the ordered phase, when $K = 1$, it has been shown that a probability of having an attractor of size l falls exponentially with l [43]. For a subset of $K = 2$ RBNs with ordered behavior, it has been shown in [93] that their average transient time grows at most logarithmically with the network size N.

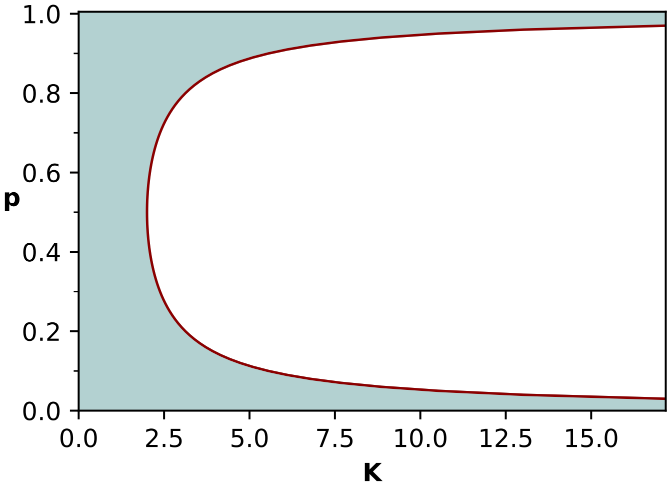


Figure 2.21: Red curve depicts the critical values of RBN mean connectivity $\langle K \rangle$ and Boolean function bias p. The blue area denotes the region of ordered behavior, white area denotes the chaotic region.

Chaotic Phase In the case when $p = \frac{1}{2}$ and $K \geq N$, the RBN is essentially a random mapping whose phase-space properties have been studied extensively. It has been shown that both the average attractor and transient lengths of such RBNs grow exponentially with increasing N [37, 61].

We note that some previous work examining the transients of RBNs was conducted in [19] and [160] but we are not aware of any studies, which would use the asymptotic transient growth to describe the behavior of RBNs at the critical region.

2.5.2 Transient Classification of RBNs

We have sampled RBNs parametrized by the mean connectivity $\langle K \rangle$ and the Boolean function bias p. In this section we show that the results of transient classification clearly distinguish the ordered, critical, and chaotic phase of RBNs.

Details of the Experiment

Our goal is to estimate the average transient length of a "typical" RBN of size N, with mean connectivity $\langle K \rangle$ and Boolean function bias p. We would do so for increasing N to observe the asymptotic behavior. We proceed as follows:

1. Given N, $\langle K \rangle$, and p, we generate a RBN $R(N, \langle K \rangle, p)$ with the corresponding parameters. We estimate the average transient length $T(R(N, \langle K \rangle, p))$ of $R(N, \langle K \rangle, p)$ using the approach described in Section Average Transients: Error Estimate.

2. We repeat step 1. and generate a sequence of RBNs

$$R_1(N, \langle K \rangle, p),\ R_2(N, \langle K \rangle, p),\ \dots,\ R_m(N, \langle K \rangle, p)$$

and their average transient lengths

$$T(R_1(N, \langle K \rangle, p)),\ \dots,\ T(R_m(N, \langle K \rangle, p))$$

to ensure that we are close to the the average transient length of an average RBN with parameters N, $\langle K \rangle$, and p. We determine the number of sampled RBNs needed to get sufficiently close to the true average behavior by method analogous to the one described in Average Transients: Error Estimate. Finally, we obtain the typical average transient length as

$$T(N, \langle K \rangle, p) = \frac{1}{m} \sum_{i=1}^{m} T(R_i(N, \langle K \rangle, p)).$$

3. We try to approximate the sequence $(T(N, \langle K \rangle, p))_{N=1}^{\infty}$ by generating a finite part of it. We typically compute $(T(N, \langle K \rangle, p))_{N=5}^{200}$, the upper bound being either $N = 200$ or the limit imposed by the computation time of the transient lengths.

2.5.3 Results

Ordered Phase We have computed K_c, p along the curve given by (1) for $p = 0.1, 0.2, \ldots, 0.9$ and sampled RBNs with parameters $\langle K_c - 1 \rangle$, p to ensure we are in the ordered region. See Figure 2.22.

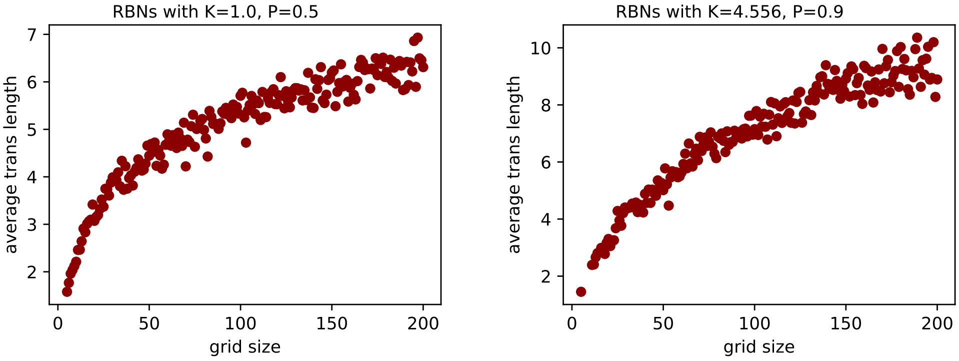


Figure 2.22: Growth of typical average transient lengths for RBNs in the ordered region. RBN with mean connectivity $\langle K \rangle = 1$ and Boolean function bias $p = 0.5$ on the left, RBN with $\langle K \rangle = 4.556$ and $p = 0.9$ on the right. The best fit for both was logarithmic, with R^2 score over 90%.

For all such ensembles of RBNs the best fit for the typical average transient asymptotic growth was logarithmic. This supports the analytical results proven for special cases of K and p values.

Critical Phase We have sampled RBNs with parameters $\langle K_c \rangle$, p along the curve given by (1) for $p = 0.1, 0.2, 0.3, \ldots, 0.9$. In all the sampled cases, the best fit for the typical average transient growth was linear. As in the case for CAs, we are aware that the asymptotic behavior of such RBNs can turn out to be logarithmic or exponential and that we just might not have sampled large enough networks. In such a case, we can interpret the Lin Class as a region of RBNs that take significantly longer to converge to their asymptotic behavior. See Figure 2.23.

Chaotic Phase We have computed the critical values K_c, p along the curve given by (1) for values $p = 0.2, 0.3, \ldots, 0.7, 0.8$ and sampled RBNs with parameters $\langle K_c + 2 \rangle$, p to ensure we are in the chaotic region. For all such ensembles of RBNs, the best fit for the typical average transient asymptotic growth was exponential, which again agrees with the analytic results. See Figure 2.24.

These experiments support the results obtained for CAs and TMs indicating that ordered discrete systems belong to the Bounded or Log Class, chaotic systems correspond to the Exp Class, and complex systems lie in the region “in between”, corresponding to the Lin and Poly Class.

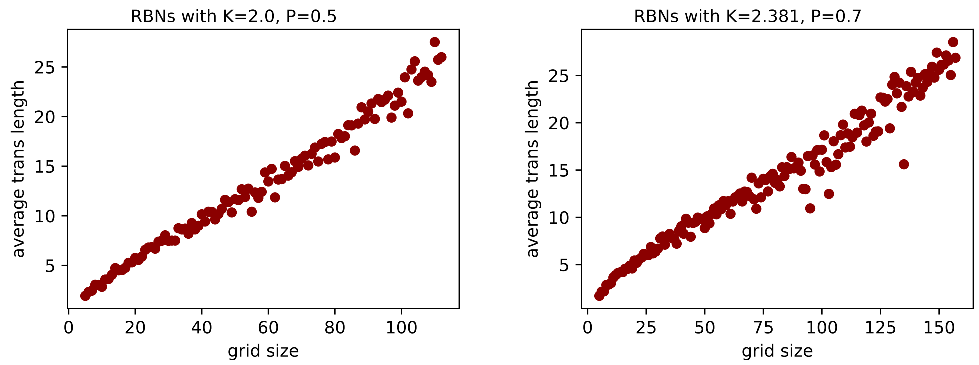


Figure 2.23: Growth of typical average transient lengths for RBNs in the critical region. RBN with mean connectivity $\langle K \rangle = 2$ and Boolean function bias $p = 0.5$ on the left, RBN with $\langle K \rangle = 2.381$ and $p = 0.7$ on the right. The best fit for both was linear, with R^2 score over 95%.

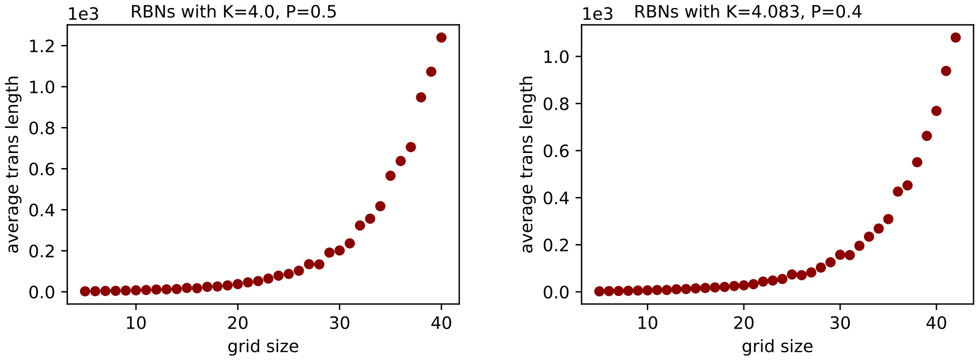


Figure 2.24: Growth of typical average transient lengths for RBNs in the chaotic region. RBN with mean connectivity $K = 4$ and Boolean function bias $p = 0.5$ on the left, RBN with $K = 4.083$ and $p = 0.4$ on the right. The best fit for both was exponential, with R^2 score over 99%.

2.6 Conclusion

We presented a classification method based on the asymptotic growth of average computation time. It is applicable to any deterministic discrete space and time dynamical system. We did present a good correspondence between the transient and Wolfram's classification in the case of ECAs. Further, we did show that the classification works for 2D CAs, Turing machines, and random Boolean networks, and we used it to discover 2D CAs capable of emergent phenomena. By demonstrating that complex discrete systems such as Game of Life, rule 110, several universal TMs, or RBNs with critical parameter values belong to the Lin or Poly Class, we believe that linear and polynomial transient growth navigates us toward a region of complex discrete systems.

Another elegant alternative would be to merge the Bounded and Log Class representing the ordered phase, and the Lin and Poly Class corresponding to the critical phase. In this way, we would obtain the traditional three phases of dynamics. This is entirely possible; we have kept the five classes to respect our initial experiments on elementary CAs where we obtained a much finer classification scheme.

The classification is based on a very simple idea and can be implemented with a few lines of code. In the field of ALife where novel discrete dynamical systems are designed as possible models of artificial evolution, the method we presented can be used to check whether such systems belong to the Lin or Poly Class. This might support the claim that such systems are capable of complex dynamics and emergent phenomena.

2.7 Future Work

We are interested in examining the transient growth of recurrent neural networks (RNNs). In the simplest case, we can add a “rounding off” output layer to discretize the configuration space. Then, the transient classification could be used to study the dynamics of RNNs, possibly guiding us towards appropriate network initializations and overall architectures yielding complex dynamics.

It would also be interesting to examine Busy Beaver Turing machines. Those are such TMs which take the longest time to halt (in the classical sense) among all TMs with the same number of states and tape symbols when run from an empty tape. It is interesting to observe how such machines behave when run from a randomly sampled initial configuration and whether they would exhibit complex dynamics, possibly being computationally universal [166].

Lastly, we could examine the dynamics of systems when simulated from a special region of its configuration space. The initial configurations could be generated by a designated algorithm, possibly discovering completely different dynamics of the system, as opposed to its average behavior.

3. Dynamical Phase Transitions in Graph Cellular Automata

3.1 Introduction

Dynamical systems can produce complex behaviour by iterating very simple local rules [104]. One of the simplest classes of such systems are Cellular Automata (CAs) [76, 134, 157]. They are a popular model system due to the fascinating structures produced in their dynamics' visualizations [162]. Analysing the global dynamics of CAs is, however, notoriously difficult and many such problems are in fact proven to be undecidable [29, 74]. One aspect of the hardness comes from the fact that the regular connectivity grid of CAs imposes significant correlations between the cells.

There are numerous ways the CA regular grid structure can be relaxed to obtain a system amenable to analysis by statistical physics. For example, the cell (or node) connectivity can be given by a random directed graph; and a (possibly different) update rule can be randomly generated for each node. This architecture gives the synchronous, deterministic, discrete dynamical systems called Random Boolean Networks (RBNs) [2]. Such a significant relaxation famously allows the RBNs' global dynamics to be analysed using mean field calculations and annealed approximations [38, 91, 140].

In this work, we study a more subtle relaxation of the CA structure. We consider systems where the connectivity of the nodes is determined by a random regular graph. All nodes in this network are updated synchronously by a fixed, identical local update rule. It is natural to call such systems Graph Cellular Automata (GCAs), although variations are known as Network Automata [38]. GCAs are very close to the CA architecture, and as such, it is still a challenge to study their dynamics analytically. Even the annealed calculation of the number of point attractors is non-trivial compared to the RBNs due to the non-directed nature of the interactions, see e.g. [31]. The main goal of this paper is to showcase a set of statistical physics tools and demonstrate that they are powerful enough to give asymptotically exact analytical results about the global dynamics of these discrete dynamical systems. Concretely, we use the *dynamical cavity method* (DCM) [69, 73, 78, 88, 102, 111] and its *backtracking* version (BDCM) [15] to give new results about the global dynamics of a specific subclass of GCAs. This class can be intuitively understood using the terminology of opinion dynamics.

Specifically, we study GCAs with *conforming non-conformist update rules.* They have binary states $\{0, 1\}$ and each node is updated in the following manner:

- if the states in a node's neighbourhood are strongly aligned (i.e., the majority wins by at least 2θ of neighbours being in the same state), the node follows the majority state in its neighbourhood
- otherwise, if the majority only has a slim lead over the minority (i.e., the majority wins by less than 2θ neighbours being in the same state), the node gets updated in one of the following non-conformist ways:
  - **type 1.** independent stubborn: the node keeps its state
  - **type 2.** independent volatile: the node changes its state
  - **type 3.** anti-conformist: the node follows the minority

All nodes are updated synchronously and deterministically, using the same update rule, either of type 1., 2., or 3 for a given value of $\theta \in \mathbb{N}_0$. The relevance of the conforming non-conformist rules stems from the fact that their dynamics can be interpreted as an opinion-formation process.

We note that the literature on opinion dynamics and its analysis through statistical physics is abundant [23, 52]. There is a plethora of connectivity topologies and update schemes that have been studied [6, 44, 144]. Some are particularly relevant to our work, as they study the co-existence of conformist and anti-conformist behaviour [45, 51, 110, 114].

The type of dynamical analysis that is of relevance in the context of opinion formation dynamics, is usually related to the dependence between the initial configurations and a type of attractor the system converges to. Some exemplary questions are:

- Which initial configurations can lead to consensus, and how fast?
- Which initial bias allows all opinions to prevail on the graph for a prolonged period of time?

To answer these questions, we consider the *density* or *bias* of a binary configuration, which is its average number of 1s, and we show that various conforming non-conformist GCAs converge to qualitatively different types of attractors depending on the density of their initial configuration. When we consider graphs with many vertices n, in the large system size limit, the transitions between these regions of different behaviours (e.g. finding consensus or having disagreement) become sharp: The probability to sample initial configurations that exhibit any other behaviour than what is typical for the region is going to zero. Because the behaviours we distinguish relate to the system's dynamics, such a sharp transition is called a *dynamical phase transition*.

The DCM and BDCM allow us to analytically identify values of initial densities where such a phase-transition occurs. This can be confirmed by numerical experiments which show that around the phase-transition, the system takes longer to converge to its typical attractor; a form of critical slowing down. Some of the results presented here have previously been used to illustrate the BDCM in the paper that introduced the backtracking version [15]. We expand on them, by discussing their relevance in the context of cellular automata and opinion dynamics, and add results for new classes of such dynamical systems.

Concretely, we show that for multiple GCAs with conforming non-conformist rules, configurations with low initial density values almost always converge to the homogeneous attractor of only 0s fast (consensus). However, above a certain initial density threshold, the systems instead exhibit more complex behaviour, which will be the object of our analysis with the (B)DCM. For example, in the case of a rule always following the majority, above a certain initial density threshold the system instead converges to an attractor oscillating between two configurations of mixed states. Another interesting type of phase-transition occurs for the anti-conformist rules of type 3 (following the minority instead of the majority when the race is tight). There, as shown in Fig. 3.1, for low values of initial configuration densities, the system converges to an all-0 consensus in time proportional to the logarithm of the network size. However, above a certain initial density threshold, the system instead takes an exponentially long time to converge.

These observations bring us back to the notoriously hard-to-analyse CAs discussed at the beginning: It is a long-standing challenge in the area of discrete systems to precise the emergence of complexity [87] and to identify a region of systems with complex behaviour [84]. In multiple works on classifying dynamics of cellular automata, the typical behaviour of the system is assessed by averaging over randomly sampled initial configurations [12, 38, 67, 86]. Specific analyses with respect to the initial configuration are the exception [7]. Our results emphasize that for certain systems, averaging the system's behaviour over initial configurations might be a coarse process, insensitive to the particularities of different initial configuration regions. For the anti-conformist rule we investigate, it is indeed the case that depending on the choice of initial configurations, the system either converges fast to a homogeneous attractor (simple regime), or it enters a chaotic regime. The qualitative difference in the rule's behaviour in the two phases is significant, see Fig. 3.1.

To summarize, in this work we show that the DCM and BDCM methods are powerful tools for analysing discrete dynamical systems. We demonstrate the existence of systems with

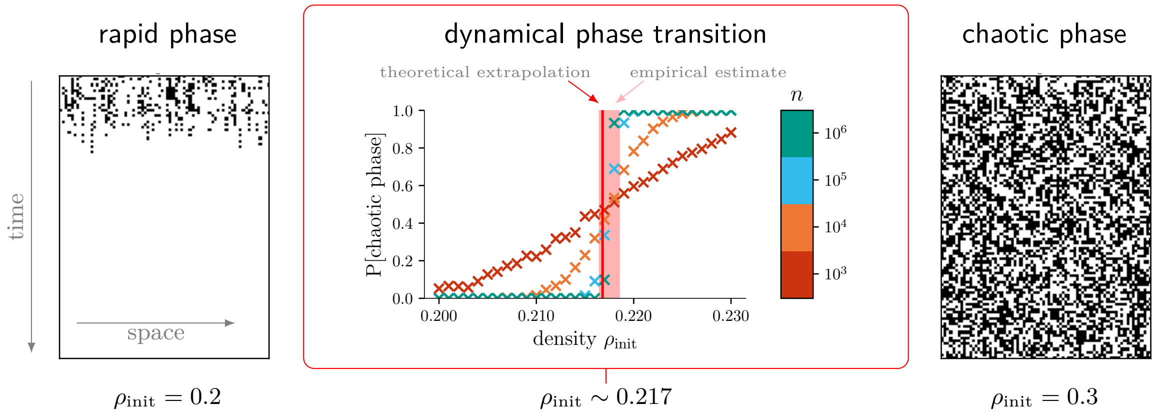


Figure 3.1: **A phase transition diagram for a particular instance of a 5-regular GCA with a conforming anti-conformist rule** 001011**.** An illustration of the system's two phases that depend on the density (i.e., the average number of black-coloured nodes) in the initial configuration. The phases are illustrated by space-time diagrams for a system of size $n = 1000$ nodes, though only a window of 75 nodes is shown. *(Left)* Rapid phase: Fast convergence to the all-0 attractor. *(Right)* Chaotic Phase: Apparent randomness in the nodes state, convergence takes longer. *(Middle)* In the large system limit, when $n \to \infty$, there is a dynamical phase transition. At a particular initial density value ρ_{init}, the typical behaviour of the system abruptly switches from the rapid to the chaotic phase. For each ρ_{init} and each system size n we sampled 1024 initial configurations with the given ρ_{init} and computed how often the system enters a chaotic phase. For practical purposes, we conclude the system is in a chaotic phase if it does not converge within $100 * \log_2(n)$ time-steps. The resulting frequency exhibits a sharp phase transition between 0.217 and 0.218, where the solid red line is our prediction from the DCM and the shaded red area comes from an empirical approximation. This transition separates the behaviour on the left and the right.

dynamical phase transitions between ordered and chaotic behaviour, and provide an analytical approach to identifying the transition between the two phases. From the perspective of opinion dynamics, we introduce a new twist on the majority dynamics where nodes are non-conforming when the majority only has a slim lead. Our analysis then shows how an initial bias affects the prevalence of both opinions and the time to reach a consensus or stable configuration. From the perspective of cellular automata, we narrow the gap between the popular systems on the grid and those amenable to statistical physics.

Note that the results presented in this paper have a certain overlap with the results presented in [15] by the same authors. The paper [15] was focused on the backtracking DCM that was introduced there and some of the GCAs that correspond to zero temperature dynamics in spin systems were discussed to illustrate the power of the method. The present paper is focused on a more generic class of cellular automata and their behaviour and the BDCM together with DCM are used as methods known from the existing literature.

3.2 Terminology and Notation

By an *undirected graph* of size n we understand the tuple $G = (V, E)$ where $V = \{1, \ldots, n\}$ is the set of nodes and $E = \{\{i, j\} \mid i, j \in V\}$ is the set of edges. For each node $i \in V$ we define the *neighbourhood of* i to be the set $\partial_i = \{j \mid \{i, j\} \in E\} \subseteq V$; and we define the degree of i as $d(i) = |\partial_i|$. We say an undirected graph is *d-regular* if each node has degree d.

Let G be a graph with n nodes and let S be a finite set of *states*. Each node i can be assigned a state $x_i \in S$; we represent such an assignment by the sequence $\mathbf{x} = x_1 \ldots x_n \in S^n$

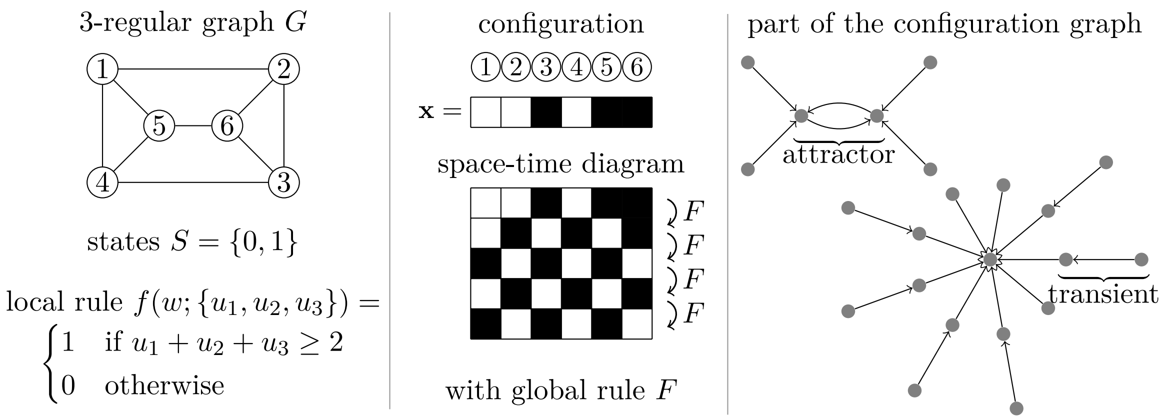

3-regular graph G
1
2
5
6
4
3
states $S = \{0, 1\}$
local rule $f(w; \{u_1, u_2, u_3\}) = \begin{cases} 1 & \text{if } u_1 + u_2 + u_3 \geq 2 \\ 0 & \text{otherwise} \end{cases}$
configuration
123456
$\mathbf{x} =$
space-time diagram
F
F
F
F
with global rule F
part of the configuration graph
attractor
transient

Figure 3.2: **Example of a GCA and its dynamics.** *(Left)* Defining the GCA from a 3-regular graph G, state set S, and local rule f following the majority. *(Middle)* A configuration $\mathbf{x}$ (0 is white and 1 is black) and the GCA's space-time diagram starting from $\mathbf{x}$. *(Right)* For the majority GCA defined on the left, we show a part of its configuration graph.

and call it a *configuration.*

Graph Cellular Automata. Let S be a finite set of states. A Graph Cellular Automaton (GCA) is a discrete dynamical system that operates on configurations of some graph with n nodes. In this work we only consider the case of random d-regular graphs. The state of each node gets updated synchronously, depending on its own state and the state of its neighbours; each node uses an identical local update rule $f : S \times S^d \to S$. This gives rise to a global mapping $F : S^n \to S^n$ governing the dynamics of the system. For a configuration $\mathbf{x} \in S^n$, the i-th node with neighbourhood $\partial_i = (i_1, \ldots, i_d)$ gets updated according to

$$F(\mathbf{x})_i = f(x_i; x_{i_1}, \ldots, x_{i_d}).$$

We write a semicolon to highlight that the first entry of f is always the state of the node being updated.

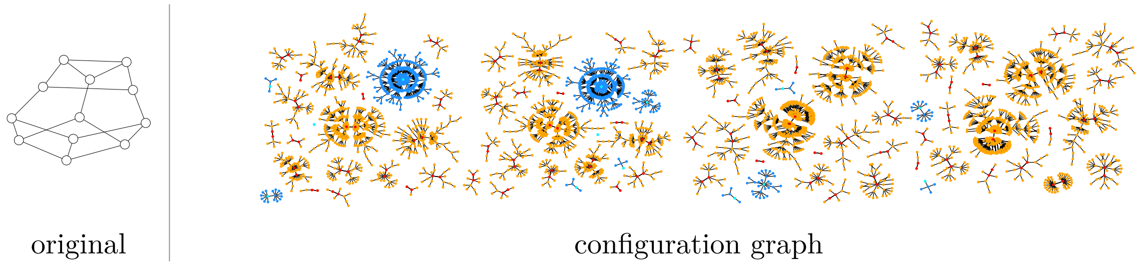
Global Dynamics. Let $F : S^n \to S^n$ be the global rule of some GCA. We will use the symbol $\underline{\mathbf{x}}$ to denote a sequence of configurations from S^n; i.e., $\underline{\mathbf{x}} = (\mathbf{x}^1, ..., \mathbf{x}^t)$ for some $t \in \mathbb{N}$. If $\underline{\mathbf{x}}$ satisfies that $\mathbf{x}^{i+1} = F(\mathbf{x}^i)$ for each i we call it the GCA's *trajectory of length t starting from the initial configuration* $\mathbf{x}^1$. We call a matrix whose rows are configurations of a GCA at consecutive times its *space-time diagram.*

Since the configuration space is finite, each long enough trajectory becomes eventually periodic. We call the preperiod of the sequence the *transient* and its periodic part the *attractor* or *limit cycle.* For an attractor, the set of configurations converging to it is called its *basin of attraction.*

We define the *configuration graph* (also called the *phase-space*) as an oriented graph whose vertices are the configurations from S^n with edges of the form $(\mathbf{x}, F(\mathbf{x}))$, $\mathbf{x} \in S^n$. The notions we defined are illustrated in Figure 3.2 and an example of the complete configuration graph for the majority rule on a graph with 12 nodes is shown in Figure 3.3.

Outer Totalistic GCAs. A GCA is outer totalistic if its local update function "does not distinguish between node's neighbours". A local rule f of an outer totalistic GCA is thus a function of a node's state and the *set of states* of its neighbours (oblivious to the ordering of the neighbours). We highlight this by writing the global dynamics in the form:

$$F(\mathbf{x})_i = f(x_i; \{x_j\}_{j \in \partial_i}).$$



original

configuration graph

Figure 3.3: **Complete configuration graph of a majority GCA.** Every node is a unique configuration of the system. The edges show how the dynamics evolve from one configuration into another. The different colours distinguish configurations that eventually evolve into different types of attractors. The orange colour marks configurations leading to cyclic attractors of size 2 marked red, the blue configurations converge to point attractors in cyan. This is the absolute majority rule, the GCA with code 0011 on a 3-regular graph with $n = 12$ nodes.

For example, $f : \{0,1\}^4 \to \{0,1\}$ given by $f(w; u_1, u_2, u_3) = (u_1 + u_2 + u_3) \bmod 2$ gives rise to a totalistic GCA whereas copying neighbour u_2's opinion given by $g(w; u_1, u_2, u_3) = u_2$ does not.

Outer Totalistic GCA Codes We restrict our study to outer totalistic GCAs with states $S = \{0,1\}$. In such a case, the local rule $f : \{0,1\} \times \{0,1\}^d \to \{0,1\}$ is characterized by a sequence of unary Boolean functions $(f_0, f_1, \dots, f_d)$, where for each $0 \le k \le d$, the function $f_k : \{0,1\} \to \{0,1\}$ dictates how a node changes its state if exactly k of its neighbours are in state 1. We further introduce a symbol for each unary Boolean function:

0	...	constant 0 function
$+$	...	identity function
1	...	constant 1 function
$-$	...	negation

Thus, each local rule f of an outer totalistic GCA on a d-regular graph is characterized by a $d+1$-tuple of symbols $(s_0, s_1, \dots, s_d) \in \{0, 1, +, -\}^{d+1}$. We will call this symbol sequence the *code* of the rule.

For example, for $d = 3$, the code $++++$ denotes the local update rule that preserves the state of each node; and the code 0011 represents a rule that updates each node based on the majority state of its neighbours. We note that an analogous representation has been introduced for example in [95].

3.3 Conforming Non-Conformist GCAs

In this paper, we study a class of outer totalistic GCAs with *conforming non-conformist update rules* (CNC). An update rule of an outer totalistic GCA on a d-regular graph with states $S = \{0,1\}$ is CNC with threshold $\theta \in \mathbb{N}_0$ if it updates each node in one of the following ways:

- **strong agreement region:** if the majority wins by at least 2θ of nodes; i.e., $|\sum_{j\in\partial i} x_j - \frac{d}{2}| \ge \theta$; the node conforms to the majority of its neighbours
- **weak agreement region:** if the majority wins by less than 2θ of nodes; i.e., if $|\sum_{j\in\partial i} x_j - \frac{d}{2}| < \theta$; the node gets updated in a non-conformist way:
  - stubborn independent: the node keeps its state; code type "**0+1**"
  - volatile independent: the node changes its state; code type "**0−1**"

- anti-conformist: the node follows the minority of its neighbours; code type "**0101**"

All the nodes in the network get updated synchronously, using the same update rule, either of type **0+1**, **0−1**, or **0101**. As an example, for $d = 5$, the anti-conformist GCA with threshold $\theta = 1$ corresponds to the rule with code 001011, and $\theta = 2$ gives the rule with code 011001.

We note that an odd connectivity d and $\theta = 0$ imply that all neighbourhood configurations result in a strong agreement region. In such a case, a node always conforms to the majority and this gives the well-studied case of absolute majority rules with code type "**01**". Whenever $\theta \geq 1$, some neighbourhood configurations result in a weak agreement region where the rules **0+1**, **0−1**, or **0101** demonstrate different forms of non-conformist behaviour.

The anti-conformist case of CNC rules has a particularly interesting interpretation in the context of opinion making: if the agreement of one's neighbours is weak, one has enough "courage" to demonstrate an attitude different from the majority. However, once the neighbours' opinion alignment is too strong, one conforms to the opinion of the majority.

Short Attractors. An important property of the CNC GCAs is that for an arbitrary system size, they only seem to have short attractors. As we will see, this is a crucial property that allows us to apply the BDCM method and analyse properties of the most typical attractor of the CNC GCAs.

Specifically, the absolute majority rules, together with the stubborn and volatile independent rules belong to a wider class of *majority threshold rules* which, irrespective of the system size, only have attractors of size 1 and 2. This applies to an arbitrary topology of the connectivity network, as long as it has undirected edges. This has been proved in [50] using an elegant argument by introducing a decreasing energy function for such systems.

For the case of anti-conformist CNC rules, we so far lack a proof of such a property. However, the numerical results suggest that attractors larger than 2 are not typical for anti-conformist GCAs of large size, as we only rarely sampled them (Appendix, Fig. 3.14). We note that the topology of a random regular graph seems crucial here as for preliminary experiments on a regular grid we encountered attractors larger than 2.

Related Work. The class of CNC rules, seemingly simple, contains systems with a wide variety of behaviour that have received a lot of attention in the literature, although not always in exactly the synchronous setting on random regular graphs. The interest is due to the rules' relevance in different application fields. For cellular automata, typically on lattices, density classification is used as a vehicle for reasoning about their computational capabilities [21, 135]. Bootstrap-percolation [164] or the zero-temperature Glauber dynamics [30, 109] can also be modelled with CNC rules and are approached on various types of graphs.

The CNC rules also play a prominent role in modelling opinion spreading. The co-existence of conformist and anti-conformist dynamics has been studied in models of collective behaviour [51, 114]. However, the co-existence is typically introduced in one of the two following ways:

1. The network consists of two types of nodes, conformist ones that always follow the majority and anti-conformist ones always following the minority.
2. With probability p a node gets updated using a majority rule, and with probability $1 - p$ it gets updated in an anti-conformist way.

In contrast, for the conforming non-conformist rules as considered in the present paper, the behaviour of a node is entirely determined by the nodes in its neighbourhood, not by external probabilities. We show two examples of such dynamic behaviour in Figure 3.4, where for the anti-conformist GCA 001011 and the volatile independent GCA $00-11$ we show three different initializations and their long-time behaviour in space-time diagrams.

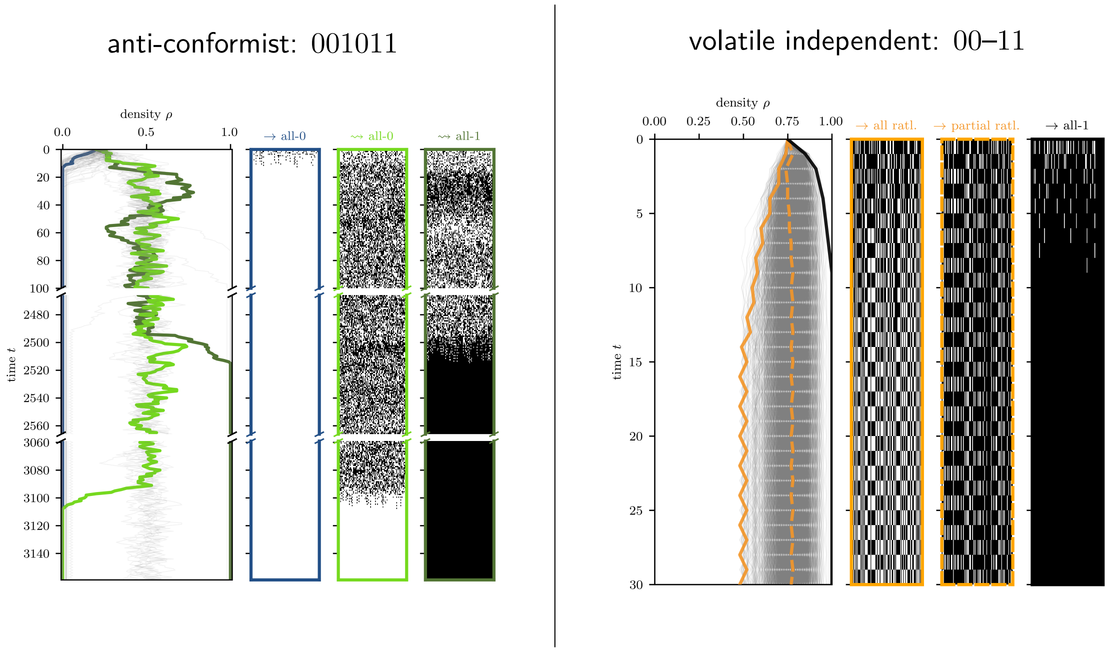


Figure 3.4: **Space-time diagrams.** We show two examples for conforming non-conformist rule dynamics on small random regular graphs with $n = 100$ nodes. *(Left)* The anti-conformist GCA 001011, in this case, the time axis is broken for visualization purposes, as for some samples the time to an attractor is extremely long. *(Right)* The volatile independent GCA $00 - 11$.

The connection between the CAs and opinion dynamics on graphs is discussed in [7]. All the mentioned applications directly raise relevant questions on the dynamics, e.g. how quickly or if at all one can reach consensus given an initial configuration [73, 121]. In the following, we show how to answer such questions for these seemingly simple but ubiquitous rules.

3.3.1 Types of Dynamical Phases

For conforming non-conformist GCAs we identify a number of qualitatively different phases the system exhibits when varying the density of 1s in the initial configuration. For the transients, we distinguish phases of slow and fast convergence. For the attractors, we distinguish between attractors of size 1 and 2, between the density of 1's in the attractor's configurations and the portion of nodes that are changing their state in a cyclic attractor. We call a specific combination of a transient and attractor type a *dynamical phase.* A *dynamical phase transition* is an abrupt, non-analytic change from one dynamical phase to another. It is the critical point where the system exhibits different qualitative behaviours on either side of the transition. This is defined in the large n limit, when the system has many interacting nodes.

To define this formally, let $\underline{\mathbf{x}} = (\mathbf{x}^1, ..., \mathbf{x}^p, ..., \mathbf{x}^{p+c})$ be a trajectory of a threshold GCA with a transient of length p leading into an attractor of length c.

Initial configuration. We define the *density* or *bias* of a configuration $\mathbf{x} \in \{0, 1\}^n$, $\mathbf{x} = (x_1, \ldots, x_n)$, as:

$$\rho(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} x_i. \tag{3.1}$$

The *initial density* for the trajectory $\underline{\mathbf{x}}$ is $\rho_{\text{init}}(\underline{\mathbf{x}}) = \rho(\mathbf{x}^1)$. We will show that as we vary ρ_{init}, the system exhibits changes in the phase it converges to that become more and more abrupt as the system size grows $n \to \infty$.

icon	attractor	description	parameters
□ ■	*homogeneous stable*	almost only point attractors with almost all nodes in state 0 or almost all nodes in state 1	$c = 1$ $\rho_{\text{attr}} \in \{0, 1\}$
▩	*mixed-colour stable*	almost only point attractors where at least a constant fraction of both 0's and 1's is present	$c = 1$ $\rho_{\text{attr}} \in (0, 1)$
▨	*partially rattling*	almost only 2-cycles with at least a constant fraction of both rattling and stable nodes	$c = 2$ $\alpha \in (0, 1)$
🟧	*all-rattling*	almost only 2-cycles with almost all rattling nodes	$c = 2$ $\alpha = 1$

Table 3.1: Four types of attractors, marking different destinations of their dynamical behaviour. We emphasize that our definition makes the distinction for α and ρ_{attr} only up to to a finite fraction $\Theta(n)$ of the nodes. This disregards a subleading number $o(n)$ of nodes that might have a different state in the homogeneous stable attractor, or $o(n)$ nodes that are not rattling in the all-rattling attractor. Likewise, the phases ignore $o(n)$ of transients which converge to attractors with limit cycle lengths with $c \notin \{1, 2\}$. (Informally, $g(n) \in \Theta(f(n))$ if g grows with the same order as f and $g(n) \in o(f(n))$ if g grows slower than f.)

Transient types. We say that the convergence to an attractor is rapid (ordered), if the transient length p as a function of the system size n grows in $O(\log n)$. Similarly, convergence is chaotic, if it takes a long time and p grows in $\Theta(\exp n)$. We conjecture from the numerical investigations that intermediate transient lengths do not appear in the systems considered here.

Attractor types. In general, we define the density of a limit cycle/attractor of length c as the average density over all its configurations:

$$\rho_{\text{attr}}(\underline{\mathbf{x}}) = \frac{1}{c} \sum_{t=1}^{c} \rho(\mathbf{x}^{p+t}). \tag{3.2}$$

For all attractors of size $c > 1$, we say that the i-th node is a *rattler* if it changes its state at least once in the limit cycle. Otherwise, we say that the i-th node is stable. We define the *activity* of a limit cycle as the average number of its rattlers, formally:

$$\alpha(\underline{\mathbf{x}}) = \frac{1}{n} \sum_{i \in V} \mathbb{1} \left[1 \leq \sum_{t=p+1}^{p+c-1} \mathbb{1}[x_i^t \neq x_i^{t+1}] \right] \tag{3.3}$$

With these definitions, we distinguish the four attractor types in Table 3.1.

Empirical Locations of Dynamical Phases. On finite systems, we can empirically measure all the previously defined properties and their scaling in the graph size n. For now, we explore GCAs with four rules: The absolute majority rule, and three rules with the different possible non-conforming behaviours under weak agreement (stubborn, volatile independent, and anti-conformist). Fig. 3.5 shows the transient length scaling in n and the attractor's density ρ_{attr} and activity α in terms of initial density ρ_{init}.

Clearly, for all four rules the homogeneous all-zero and all-one state is an attractor of the dynamics. We observe that when the initial bias is close to such a homogeneous attractor the convergence to it is rapid for all four rules. All the studied rules undergo a phase transition for some value ρ_{init} which jointly occurs with a slowing down of the convergence (an increase in transient lengths).

The anti-conformist rule's behaviour stands out, where the exponentially long transients lead to the all zero or all one attractor with equal probability. For the other three rules (majority,

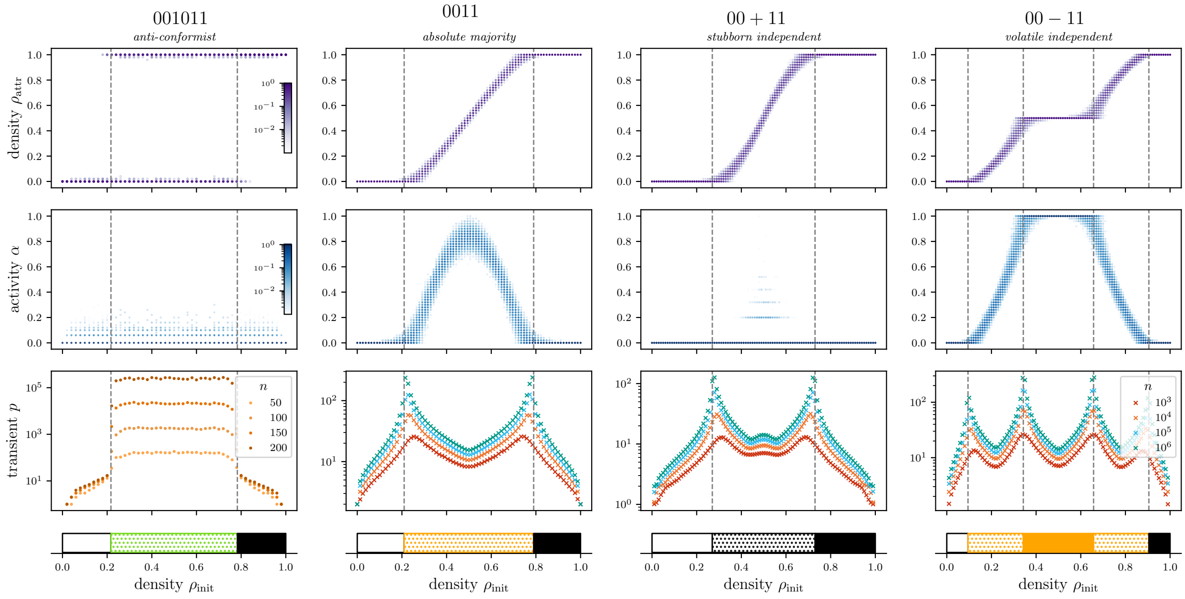


Figure 3.5: **Numerical experiments for four types of CNC rules for $d = 3, 4, 5$.** For rules 0011, $00 + 11$ and 00–11 we sampled 1024 graphs for every $n \in \{10^3, 10^4, 10^5, 10^6\}$ and every initial density $\rho_{\text{init}} = \frac{k}{100}$, $k \in \{0, 1, \ldots, 100\}$. For 001011, due to the exponential explosion of the transient length in $\sim (0.2, 0.8)$, we used $n \in \{50, 100, 150, 200\}$. *(First and second row)* Histograms of the properties of attractors: their density and the fraction of rattlers. With the exception of GCA 001011, they were computed for $n = 10^3$ with a binning on the y-axis for both ρ_{attr} and α and 101. For the GCA 001011 graphs of size $n = 200$ and 51 bins were used. *(Third row)* Average transient length p for 0011, 00+11 and 00–11; median transient length for 001011. We observe behaviour consistent with either exponential or logarithmic growth of the transient lengths as a function of the system size. *(Last row)* Diagram showing the dynamical phases corresponding to each of the attractor and transient type. Transitions between the phases correspond to peaks in the transient length or a change in the scaling of the transient regime. Note that the same data for the rules 00+11 and 00–11 were already used in [15] to illustrate the results that the BDCM can obtain.

stubborn/volatile independence) the slowing down is within the $O(\log n)$ regime, just with a larger growing prefactor in the vicinity of the phase transition. For the volatile independence rule, there are even two such transitions. The transient behaviour for anti-conformist rule is very different. It switches from the short $O(\log n)$ to the long $\Theta(\exp n)$ transients around the critical point close to ≈ 0.2 and ≈ 0.8. The long transients are maintained throughout the dynamical phase.

The different phases confined by those transitions occur as follows: The absolute majority rule on the 3-regular graphs phase rapidly converges to the partially rattling state, where a core is stable, and some nodes are changing their opinion at every iteration. The stubborn independent rule on the 4-regular graph produces an attractor with mixed (i.e. 0 and 1) stable opinions which is reached rapidly. The volatile rule, coming from the homogeneous all-zero attractor and increasing ρ_{init}, first goes through a dynamical phase of rapid convergence towards a partially rattling state, similar to the majority rule before. For very weak initial bias, eventually all nodes keep switching their colors – the all-rattling attractor.

Previous work identified similar dynamical phases for the threshold q-voter model [151], although that work considers a thresholded, noisy version of the majority rule.

We further highlight that the transitions only become sharp for large n. For smaller finite systems and particular initial density values ρ_{init}, we can observe the co-existence of phases at both sides of the transitions. For example, this happens for the two GCAs on small graphs with $n = 100$ nodes that are shown in Fig. 3.4, Section 3.3.

In Appendix 3.7 we provide some empirical results for examples with larger degrees. For rules which belong either to the absolute majority, stubborn and volatile independent rules, scaling the threshold θ as $O(1/\sqrt{d})$ exhibits the same transitions as the degree d grows, consistent with the type of large d behaviour observed in [14]. For the anti-conformist the picture is less clear as new types of behaviour emerge that are different from what we observed for the GCA 001011. Overall, we leave thorough empirical and theoretical investigations of larger degrees and their appropriate parameterizations to future work.

In the remaining Sections, we supplement our empirical results with a theoretical analysis of the precise positions of the phase transitions. For this, we first introduce the (backtracking) dynamical cavity method in Section 3.4, and then present the derived analytical dynamical phase transitions in Section 3.5.

3.4 Dynamical Cavity Methods

To analyse the dynamics of the previously introduced family of CNC GCAs, we use the *dynamical cavity method* (DCM) [69, 73, 78, 88, 102, 111] and its extension, the *backtracking dynamical cavity method* (BDCM) [15]. These methods are inspired by the cavity method from statistical physics which has proven its success in the analysis of static systems [101]. While their results hold for the thermodynamic limit, i.e. when the number of nodes n tends to infinity, we will see that the behaviour of systems with relatively small n already corresponds well to the theoretical predictions for large n.

Both methods consider motifs from the configuration graph (Fig. 3.3) that represent dynamical phenomena as the static element of a cavity analysis. The idea of the DCM is to take finite trajectories from the configuration graph. Similarly, the BDCM considers finite trajectories that lead into cycles of a fixed length. A general motif that encompasses both ideas is the *backtracking attractor*, defined as

$$\underline{\mathbf{x}} = (\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^p, \mathbf{x}^{p+1}, \dots, \mathbf{x}^{p+c}) \in (S^n)^{p+c}\,,$$

for $p, c \in \mathbb{N}$ where the first p configurations compose a transient and the last c configurations a limit cycle. Therefore, $c = 0$ gives the trajectories without attractors for the DCM and $c > 0$ gives the BDCM. Despite the static methodology, the backtracking attractor is inherently dynamic, so the static analysis allows one to infer back results about the dynamics. In order to identify the dynamical phase transitions from the previous section, it suffices to answer the following question: What are the average properties of the typical (= most numerous) backtracking attractor for a fixed ρ_{init} when $p \to \infty$?

Introduction to (B)DCM. Before we answer this question precisely for the conforming non-conformist GCAs we give a brief overview to the (B)DCM, to make clear how it works - and why this approximation is valid for the conforming non-conformist rules on random regular graphs. In this introduction, we want to give a good understanding of the method. However, we refer the reader to [15] for the original derivation.

The main ingredient to the (B)DCM is a probability distribution over all possible sequences of configurations $(S^n)^{p+c}$. In the simple case, the probability assigns a uniform value to all (p/c) backtracking attractors $\underline{\mathbf{x}}$ that occur in the configuration graph of the dynamics, and a zero measure to any other sequence:

$$P(\underline{\mathbf{x}}) = \frac{1}{Z} \mathbb{1}\left[F(\mathbf{x}^{p+c}) = \mathbf{x}^{p+1}\right] \prod_{t=1}^{p+c-1} \mathbb{1}\left[F(\mathbf{x}^t) = \mathbf{x}^{t+1}\right]. \tag{3.4}$$

Here, $\mathbb{1}(\cdot)$ is the indicator function on a Boolean statement where a true statement yields 1 and 0 otherwise. If $c = 0$ and therefore x^{p+1} is undefined, we drop the first factor where it appears. The normalization constant Z of this distribution is then equivalent to the number of valid backtracking attractors. Since this number Z is extensive in the system size n, we measure it in terms of the free entropy density $\Phi = \frac{1}{n}\log(Z)$. However, computing Z and therefore the entropy directly is intractable, due to the high-dimensional integral over all possible configurations. To solve this issue, analogous to the classical cavity method for static analysis, we use the Bethe Peierls approximation to compute its leading exponential factor using Belief Propagation (BP) on its factor graph. This approach is exact for factor graphs that are trees and, in many cases, leads to asymptotically exact results for sparse locally tree-like factor graphs. In the literature, the cases where the BP provides asymptotically exact results on sparse random graphs are called replica symmetric and [15] observed that it indeed plausibly provides asymptotically exact results for the cases studied there.

Eventually, this approach leads to a lower dimensional fixed point equation which is amenable to numerical solutions. In addition to the approximation of the free entropy density Φ_{BP}, this approximation conveniently admits a means of computing its marginals and expectations for observables[1] of the system, e.g. the density of the attractor. By additionally introducing re-weighting of the backtracking attractors in the probability distribution according to some external potential we can also 'fix' some of their properties to a prescribed constraint, and extract for example only backtracking attractors with a fixed initial density ρ_{init}.

Equations for random regular graphs. For random d-regular graphs this strategy admits a particularly simple analysis: Under the assumption that all neighbourhoods are locally the same, solving the BP on the factor graph corresponding to eq. (3.5) is equivalent to solving a fixed point equation for only one neighbourhood. Then, the message on the factor graph $\chi^{\rightarrow}_{\underline{x},\underline{y}} \in \mathbb{R}^{4(p+c)}$ from the center node x to its neighbour y is defined in terms of all possible values

[1] This is only possible when the observable factorizes over the nodes.

that its other $d-1$ neighbours $\mathbf{y}$ can take. It is re-weighted by $\chi^{\rightarrow}$ itself [2]:

$$\chi^{\rightarrow}_{\underline{x},\underline{y}} = \frac{1}{Z^{\rightarrow}} \underbrace{e^{-\lambda\tilde{\Xi}(\underline{x})}}_{\substack{a(\underline{x})\\ \text{observable/}\\ \text{constraint}}} \sum_{\underline{x},\underline{\mathbf{y}}_{[d-1]}} \left(\underbrace{\mathbb{1}\left[f(x^{p+c};\mathbf{y}^{p+c}_{[d]}) = x^{p+1}\right] \prod_{t=1}^{p+c-1} \mathbb{1}\left[f(x^{t};\mathbf{y}^{t}_{[d]}) = x^{t+1}\right]}_{\substack{\mathcal{A}(\underline{x},\underline{\mathbf{y}}_{[d]})\\ \text{valid } (p/c)\text{-backtracking attractor}}} \prod_{\underline{z}\in\underline{\mathbf{y}}_{[d-1]}} \chi^{\rightarrow}_{\underline{z},\underline{x}} \right) \tag{3.5}$$

Here, the inner constraint assures that we only consider valid backtracking attractors. The $Z^{\rightarrow}$ is again the normalization constant, the interval is $[k] = 1, ..., k$ and $\tilde{\Xi}$ is the factorized observable of the global extensive variable of interest $\Xi(\underline{\mathbf{x}}) = \frac{1}{n}\sum_{i=1}^{n}\tilde{\Xi}(\underline{x})$. This localized observable $\tilde{\Xi}$ with the factor λ allows for the previously mentioned re-weighting and constraining. As an example, take the initial density, for which we define the terms of the summand $\tilde{\Xi}(\underline{x}) = x^1$, so that the intensive global variant is $\frac{1}{n}\Xi(\underline{\mathbf{x}}) = \frac{1}{n}\sum_{i=1}^{n}\tilde{\Xi}(\underline{x}) = \rho_{\text{init}}(\underline{\mathbf{x}})$.

To obtain the BP approximation of the entropy density, it suffices to compute the following at the fixed point of eq.(3.5):

$$\Phi_{\text{BP}} = \log(Z^{\text{fac}}) - \tfrac{d}{2}\log(Z^{\text{var}}) \,, \tag{3.6}$$

$$Z^{\text{fac}} = \textstyle\sum_{\underline{x},\underline{\mathbf{y}}_{[d]}} \mathcal{A}(\underline{x},\underline{\mathbf{y}}_{[d]}) \prod_{\underline{y}\in\underline{\mathbf{y}}_{[d]}} \chi^{\rightarrow}_{\underline{x},\underline{y}} \,, \tag{3.7}$$

$$Z^{\text{var}} = \textstyle\sum_{\underline{x},\underline{y}} a(\underline{x}) \chi^{\rightarrow}_{\underline{y},\underline{x}} \chi^{\rightarrow}_{\underline{x},\underline{y}} \,. \tag{3.8}$$

Implementing and finding a solution to (3.5) can be intricate due to numerical instabilities. The solver used for our analysis is available on github[3].

Since the strength of the reweighting λ which we fix during the iteration of the fixed point, acts only as the Lagrangian multiplier, it has no immediate correspondence to the value of the constraint (e.g. ρ_{init}). To find the concrete value, we use that at a fixed point $\chi^{\rightarrow}$ it holds that

$$\frac{\partial\Phi_{BP}(\lambda)}{\partial\lambda} = -\frac{1}{n}\left\langle\tilde{\Xi}\right\rangle_{BP} = -\frac{\sum_{\underline{x},\underline{y}}\tilde{\Xi}(\underline{x})e^{-\lambda\tilde{\Xi}(\underline{x})}\chi^{\rightarrow}_{\underline{y},\underline{x}}\chi^{\rightarrow}_{\underline{x},\underline{y}}}{\sum_{\underline{x},\underline{y}} e^{-\lambda\tilde{\Xi}(\underline{x})}\chi^{\rightarrow}_{\underline{y},\underline{x}}\chi^{\rightarrow}_{\underline{x},\underline{y}}}. \tag{3.9}$$

We can measure the activity α or the density in the attractor ρ_{attr} by adjusting the function $\tilde{\Xi}$ correctly. This allows us to obtain their marginals even when we did not reweight the distribution, as this corresponds to the setting where the corresponding $\lambda = 0$.

Notice that the assumption of all the neighbourhoods being described by eq. (3.5) is equivalent to the replica symmetric assumption which in turn on random regular graphs without another source of disorder is equivalent to the annealed calculation of the free entropy. In the present systems, the annealed calculation is non-trivial, see e.g. [31] and writing the BP equations (3.5) is the most efficient way to obtain it we know of.

Application to conforming non-conformist GCAs. Notice that above we wrote the equations for dynamical systems that are updated in parallel, are deterministic and run in discrete time. The update function does not distinguish between particular neighbours of a node and the connectivity graph of the neighbouring nodes is locally tree-like in the large size limit. Finally, the size of the system's attractors has to stay constant as the system's size increases. Since from our definitions and our empirical observations all these properties hold for the conforming non-conformist rules on random regular graphs, the (B)DCM is perfectly suitable for an analysis of the CNC rules.

[2] This equation is equivalent to (17) from [15], and the derivation and factor graph is described therein using the same notation.

[3] `github.com/SPOC-group/dynamical-phase-transitions-GCAs`

Recall that we want to answer "What are the average properties of the typical backtracking attractor for a fixed ρ_{init} when $p \to \infty$?". One can take two approaches to this question, either by answering it starting from the initial or final configuration of the backtracking attractor.

To answer "What are the properties later in the dynamics given that the starting point is fixed?", we use the DCM. This means setting $c = 0$ in the backtracking attractors, we are only looking at paths. As we increase the trajectory length p we can observe how the density on the last configuration $\rho_p = \rho(\mathbf{x}^p)$ evolves.

To answer "How large is the basin of attraction of a specific type of attractor?", we use the BDCM. We can fix properties of the attractor, e.g. $c = 2$ and $\alpha = 0.5$ to identify a specific partially rattling attractor, and then increase p to measure the evolution of the size of the basin of attraction in terms of its entropy density. As one increases the length of the incoming path p, the analysis incorporates a growing fraction of the attractors' basin. Comparing this entropy between different types of attractors allows us to determine which is the most numerous and typical behaviour that is observed in the large n limit.

We will use these two general principles to identify analytically the dynamical phase transitions we empirically observed in Section 3.3.

Limitations and Alternative Methods. A significant limitation of the (B)DCM is that solving the previously mentioned fixed point equations numerically requires a computational budget which grows exponentially in $d(p + c)$ when considering a d-regular graph. While the dependence on d can be alleviated via dynamical programming [147], it is prohibitive to analyse very long paths p or large cycles c. This means that applying the method directly is only possible for small dynamic motifs which yield interesting results only for rapidly relaxing properties at the start or end of the dynamics. However, this is exactly what we observe for the conforming non-conformist rules and which makes the analysis with the (B)DCM feasible.

It is worth noting that by making additional assumptions, such as the one-time approximation, longer dynamics become amenable to the method. However, this is at the cost of further uncontrolled approximations [5, 11, 33]. Alternative methods of analysis from statistical physics give results for simpler dynamics; examples include but are not limited to the random functions in RBNs [38, 91, 140] or unidirectional dynamics with absorbing states [4, 88]. Another helpful feature is the relaxation of the topology, for example oriented graphs [111], graphs with asymmetrically weighted edges [102] for straightforward use with the DCM or independently re-sampled neighbourhoods at every iteration [110, 151] which are amenable to mean field methods. However, to the best of our knowledge the (B)DCM as we use it comes closest to the very difficult case of understanding cellular automata with its rigid and deterministic architecture.

3.5 Dynamical Phase Transitions for Conforming Non-Conformist GCAs

In the following, we detail how we apply the DCM and BDCM to the examples we investigated empirically in Section 3.3, Fig. 3.5. Recall that for all GCAs seen previously, when ρ_{init} is close enough to either 0 or 1, the dynamics rapidly falls into one of the homogeneous attractors, while the region in between exhibits more complex dynamics. This region differs for every rule type. The goal is to analytically identify these phase transitions between the regions precisely. Some of these results have previously been used to demonstrate the BDCM in [15].

Anti-conformist GCA: 001011. Recall that the anti-conformist GCA 001011 exhibits both chaotic and ordered behaviour for different values of ρ_{init}, but always converges to the all-1 or all-0 attractor eventually. The dynamics of this GCA is fully deterministic, yet the configurations of trajectories in the chaotic phase look random with respect to the density

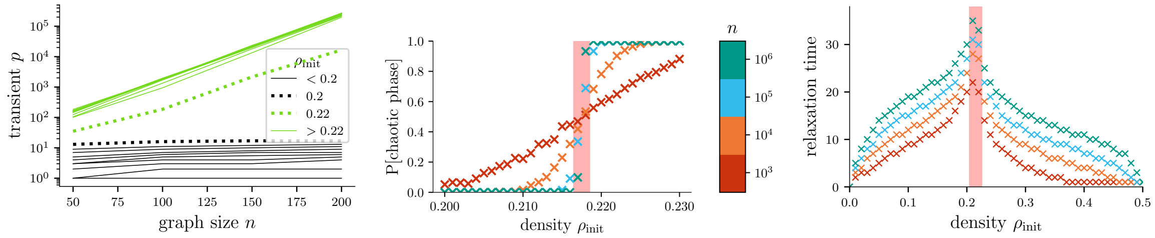
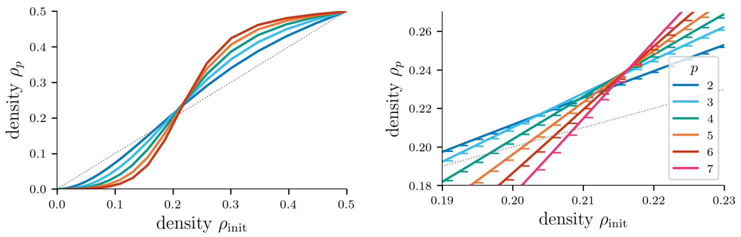


Figure 3.6: **Transient growth, chaotic phase classification and relaxation time for the anti-conformist GCA 001011.** *(Left)* For $\rho_{\text{init}} < 0.5$ we display the transient growth for graphs of size $n \in \{50, 100, 150, 200\}$, generated as in Fig. 3.5. The resolution of ρ_{init} is limited by $n = 50$, a stepsize of 0.02. A transition between an exponential (straight line in the log-linear plot) and a much slower transient growth between $\rho_{\text{init}} = 0.2$ and 0.22 is clearly visible. *(Middle)* Empirical phase transition for the onset of a chaotic phase, which in this case is defined as the attractor taking more (chaotic) or less (homogeneous stable) than $\log(n) * 100$ time steps to reach an attractor. The resolution of ρ_{init} is 0.001 and narrows the interval of the dynamical phase transition down to $[0.2165, 0.2185]$, the interval for $n = 10^6$ between which no samples out of 1024 exhibit a behaviour that is not consistent with their phase. *(Right)* The relaxation time describes the number of time steps required until either the chaotic regime or an attractor is reached. We empirically conclude the system is in the chaotic regime if the densities of 100 consecutive configurations remain in the interval $(0.5 - \frac{3}{\sqrt{n}}, 0.5 + \frac{3}{\sqrt{n}})$. For all values of n the maximal length of the two largest ρ_{init} we observe are $\rho_{\text{init}} = 0.21$ and 0.22.

ρ (see e.g. Fig. 3.4), hence the name. The difference in behaviour between the chaotic and ordered phase clearly shows in Fig. 3.6 (left), where the transient length grows exponentially in the graph size n for $\rho_{\text{init}} \geq 0.22$. However, running larger system sizes than $n = 200$ until convergence is prohibitively expansive, so the resolution of the transition we can obtain from this method is limited.

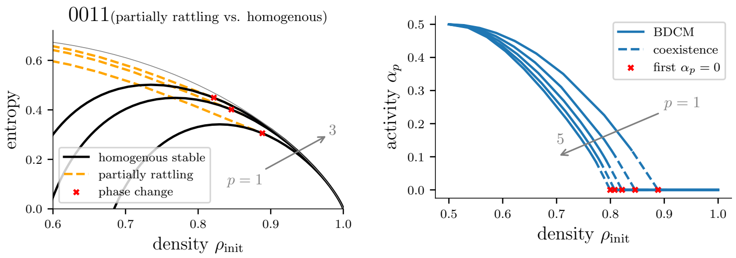
Therefore, as an additional criterion for identifying the chaotic phase for anti-conformist GCA, we check when the convergence time exceeds a threshold of $100 * \log_2(n)$. At this point, the simulation is stopped and trajectories that have not yet converged are classified as chaotic. Even though this heuristic is robust to changes of the factor 100 to 50 or 1000, we confirm the results with another method.

Inspecting the trajectories of the density ρ in Fig. 3.4, we observe that the density of configurations in a chaotic phase is oscillating around $\rho = 0.5$; more precisely it seems to



p	DCM prediction intersection of ρ_p and ρ_{p+1}
2	0.2039
3	0.2142
4	0.2158
5	0.2165
6	0.2167
$\to \infty$	0.2168 ± 0.0001
empiric	0.2175 ± 0.001

Figure 3.7: **DCM prediction of dynamical phase transition for the anti-conformist GCA** 001011. *(Left)* The prediction of the DCM for the density ρ_p after p steps, for different initial configurations ρ_{init}. *(Middle)* Zoom into the region of the phase transition, with data for $p = 7$ added. *(Right)* Table of the crossover points between the different lines. The curves in the middle zoom were fitted with a linear regression and then the intersection was computed. Extrapolating $p \to \infty$ gives a transition at $\rho_{\text{init}} = 0.2168 \pm 0.0001$ (see Appendix Fig. 3.17).



BDCM prediction 0011 for first $\alpha(\rho_{\text{init}}^{*,p}) = 0$		
p	$\rho_{\text{init}}^{*,p}$	$s_p^*/H(\rho_{\text{init}}^{*,p})$
1	0.8885	0.873
2	0.8459	0.933
3	0.8216	0.958
4	0.8080	0.971
5	0.7996	0.979
$\rightarrow \infty$	0.7875 ± 0.005	
empiric	0.785 ± 0.005	
[73]	> 0.7865	

Figure 3.8: **BDCM prediction for the absolute majority GCA** 0011**.** *(Left)* Entropy of the basin of attraction for the homogeneous and partially rattling attractors, for increasing path lengths $p = 1, 2, 3$. *(Middle)* The activity in the limit cycle for fixed points for $(p/c = 2)$ backtracking attractors. The dashed line shows the range of ρ_{init} for which our numerics did not find any fixed points. *(Right)* The table shows the values of the smallest $\rho_{\text{init}}^{*,p} > 0.5$ for which $\alpha(\rho_{\text{init}}^{*,p}) = 0.0$ together with the normalized entropy at the corresponding given ρ_{init}. It shows the extrapolation $p \to \infty$ and compares it with the numerical results and related work (see Appendix 3.9).

remain in the interval of densities $(0.5 - \frac{c}{\sqrt{n}}, 0.5 + \frac{c}{\sqrt{n}})$ where n is the system size and c is a constant (see Appendix 3.9 for details). We use this observation as a heuristic criterion for assessing whether a trajectory has entered the chaotic phase: Once a trajectory's densities stay in $(0.5 - \frac{3}{\sqrt{n}}, 0.5 + \frac{3}{\sqrt{n}})$ for a sufficient amount of time (100 time-steps), we conclude the trajectory is in the chaotic phase. The time it takes to either reach this chaotic phase or an attractor is shown in Fig. 3.6 (right), it peaks around the approximate location of the dynamical phase transition.

With these three numerical experiments from Fig. 3.6, we have a good agreement to identify a phase transition to be between $\rho_{\text{init}} = 0.217$ and 0.218. We proceed by obtaining it analytically using the DCM.

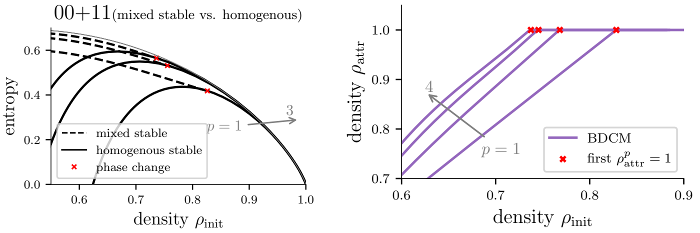
Recall that the DCM is limited to small lengths p of the trajectory for which we can solve the fixed point iterations efficiently. The question is then, how can we distinguish whether the dynamics converges fast or slow when we can look ahead only a finite number of steps p?

To answer this, observe that the relaxation time is extremely fast for any ρ_{init}, even for the ones that go on to stay in the chaotic region for an exponentially long time. Further, we observed that on average, during the chaotic phase, the density is 0.5. The appropriate question is then: After p steps of the DCM, what is the density of the last configuration ρ_p in the large n limit? At the infliction point for growing p, we expect to find the dynamical phase transition. In Fig. 3.7, we show an overview and zoom-in for the relationship between ρ_{init} and ρ_p, for p up to 7. We compare the extrapolated value for $p \to \infty$, assuming exponential convergence (see Appendix 3.9), and our empirical extrapolation. Indeed, the correspondence theory and empirics is very good, at a theoretically predicted transition around $\rho_{\text{init}} \sim 0.2168$.

Since both the chaotic and ordered dynamics for the anti-conformist GCA 001011 have attractors of the same type, the backtracking approach of the BDCM is not very insightful for this specific transition. However, it is useful to inspect the other CNC rules in the following.

Absolute Majority GCA: 0011**.** For the absolute majority GCA 0011, the convergence is logarithmic independently of ρ_{init} and the system's phases differ only in the type of attractor they converge to.

In Fig. 3.8 we show the entropy of backtracking attractors with a path length $p = 1, 2, 3$ obtained via the BDCM. Here, the entropy represents the size of its basin of attraction when stepping back p steps from the attractor, for a specific ρ_{init}. Each differently styled line represents a single type of attractor. Their entropy was obtained by solving the BDCM fixed point iteration



BDCM prediction 00+11 for first $\rho_{\text{attr}}(\rho^{*,p}_{\text{init}}) = 1$		
p	$\rho^{*,p}_{\text{init}}$	$s^*_p/H(\rho^{*,p}_{\text{init}})$
1	0.828	0.909
2	0.768	0.965
3	0.745	0.983
4	0.737	0.990
5	0.733	0.994
$\to \infty$	0.73 ± 0.005	
empiric [15]	0.731 ± 0.002	

Figure 3.9: **BDCM prediction for the stubborn independent rule 00+11.** *(Left)* Entropy of the basin of attraction for the all-1 and mixed stable attractors respectively, for increasing steps into the basin of attraction $p = 1, 2, 3$. The dynamical phase transition is marked in red. *(Middle)* The density of 1's in the attractor, ρ_{attr}, as a function of ρ_{init} for attractors with $c = 1$. *(Right)* The transitions is the first ρ_{init} for which the $\rho_{\text{attr}} = 1$. These values are recorded in the table together with the entropy of the basin of attraction at that point. The extrapolation agrees well with the empirical estimate of the transition from the maximal slowing down (see Appendix 3.9).

under the constraint matching the respective attractor properties, i.e. $c = 1, 2$. In addition, the value of ρ_{init} was constrained, giving the final result. In the large n limit only the types of attractors with the maximum entropy are expressed. Therefore, the correct way to interpret the plots is to check which attractor type has the maximum entropy for every ρ_{init} — this phase will be the one which is typically observed in large systems.

At $p = 0$, we would only count the attractors, without their basin, essentially using the method from [69]. However, only as we increase p and incorporate the basin of attraction, we observe that the overall picture from the empirics Fig. 3.5 is reproduced qualitatively by the BDCM: For large ρ_{init}, it shows the all-one attractor. Decreasing ρ_{init} around 0.5, one finds the partially rattling attractor.

Since a $(p/c = 1)$ backtracking attractor is also a $(p/c = 2)$ backtracking attractor, the two entropy curves naturally merge when the $(p/c = 2)$ backtracking attractors reduce to attractors that are of length $c = 1$. This merge between the two curves is the dynamical phase transition at a given fixed p, see Fig. 3.8 (left). Inspecting the fixed point for $c = 2$, we indeed find for large enough ρ_{init} that the activity α, the fraction of rattling nodes in the limit cycle, becomes essentially zero (Fig. 3.8 (middle)). This indicates that the number of such rattling nodes no longer scales in $O(n)$ and that the fixed point only considers limit cycles of length $c = 1$. Recording the switch from $\alpha = 0$ to $\alpha > 0$ gives the dynamical phase transition, as shown in the Table on the right in Fig.3.8. Even though we did not compute values larger than $p = 5$, we extrapolate the BDCM result to $p \to \infty$ to make our theoretical prediction. This agrees well with the empirical prediction (Appendix 3.9).

Stubborn Independent GCA: 00+11. We can do a similar type of analysis for the stubborn independent GCA $00 + 11$. This is the GCA where the node in the weak agreement region is stubborn, i.e. it sticks with its own opinion. In this analysis, we distinguish between two types of attractors that go either to the homogeneous all-1 or mixed stable state, which both have a limit cycle length of $c = 1$. Recall that the mixed stable state is defined to be an attractor where the density in the attractor ρ_{attr} is not 0 or 1 (see Table 3.1). In Fig. 3.9 (left) we show the entropy for small p in terms of ρ_{init}. To identify the dynamical phase transition we again track the spot where the attractor type with the maximum entropy switches over. Here, this is a merge of the two curves again. While we can restrict the fixed point iteration to variables that always end up in the all-1 attractor, for mixed stable attractors there is so far no technical means of constraining it to a non-zero ρ_{attr}. This is why we track the ρ_{attr} as a function of ρ_{init} in Fig. 3.9 (middle) and record when this property becomes close enough to 1.0, giving us the

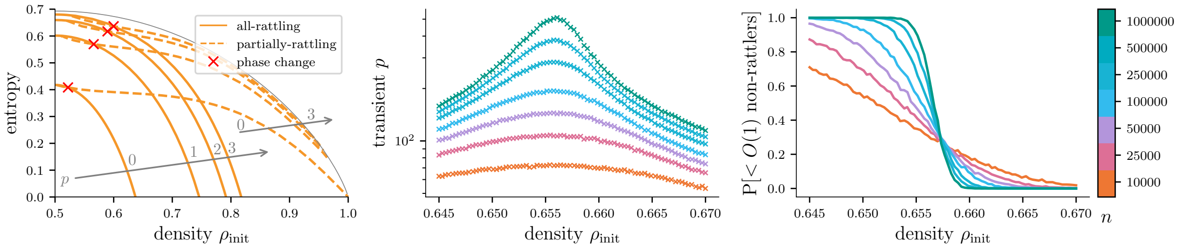


Figure 3.10: **Dynamical phase transition for the volatile independent GCA** $00-11$. Comparison of the analytical and empirical prediction of a dynamical phase transition for the volatile independent GCA $00-11$. We examine the transition between the all and partially rattling 2-cycles. *(Left)* Analytical prediction of the entropy for each ρ_{init} for the two different types of attractors basin of attraction for increasing transient lengths p. The intersection of the two entropy marks the phase transition for a given p and is marked in red. Because the computed entropy is not close enough to the maximal entropy, as shown by the grey line, the approximation of the transition is not very conclusive and extrapolating the four data points would result in very high uncertainty. *(Middle)* Zoom in on the average transient length around the phase transition from Fig. 3.5. *(Right)* Probability of obtaining a smaller than $o(n)$ fraction of rattlers, i.e. the fraction of nodes in the attractor. To determine a reasonable threshold for having a constant $o(n)$ fraction of rattlers, when n is finite, we analysed the scaling of the rattler fraction as a function of n, which resulted in an attractor having no more than 0.07% of non-rattlers to be classified as a partially-rattling attractor (Appendix 3.9). While the thresholds agree roughly, the accuracy is worse than for the GCAs discussed previously.

value of change. The table on the right of Fig. 3.9 records these values. The extrapolation to $p \to \infty$ matches well with the empirical result.

Volatile Independent GCA: $00-11$. The volatile independent GCA $00-11$ is slightly more complex, as it has more phases than the GCAs discussed previously, and four dynamical phase transitions (Fig. 3.5). Since the transition between partially rattling and homogeneous phase is similar to the GCA 0011, we discuss only the transition between all-rattling and partially-rattling, i.e. the change between attractors of length $c = 2$ where either all nodes change (activity $\alpha = 1.0$) or some of them are fixed ($\alpha < 1$).

In Fig. 3.10 the entropy of the two phases is shown on the left. This time, the two fixed points intersect, and do not merge. However, the fraction of the basin of attraction covered by the $p = 4$ steps that are taken back, is smaller than in the other examples. This can be viewed as a reason for which this dynamical transition is correct qualitatively, but the approximation is not precise. The fact that for this GCA, more steps back are necessary, reflects the common observation that close to phase transitions the convergence time increases, which makes the use of the BDCM more challenging computationally in its vicinity by default.

3.6 Conclusion and Open Questions

In this work, we use tools from statistical physics – the dynamical cavity method and its backtracking version – to demonstrate that they are powerful for deriving analytical results on the global dynamics of discrete dynamical systems in the large system size limit.

Concretely, we study a class of graph cellular automata called the conforming non-conformist GCAs that can be interpreted as various models of opinion formation dynamics. We argue that such systems exhibit a rich set of dynamical phases defined by their different transient and attractor properties, and we show the existence of sharp transitions between such phases in terms of the initial configuration density.

For two specific examples with small degrees, we showed how the (B)DCM methods are applied and predict the phase transitions. We show that our analytic predictions agree well with numerical estimates for reasonably large systems.

Such results enforce the narrative that for discrete dynamical systems, different choices of initial configurations can lead to qualitatively different regimes of the system's behaviour.

Relationship between CAs and GCAs. In its formulation, the graph cellular automata are extremely close to classical cellular automata – they only differ in how their nodes are connected. Whereas for CAs, the connectivity network is given by a regular grid, the GCAs' connections are defined by a random regular graph. As such, deriving analytical results about their global dynamics is challenging and our work shows a variety of new results about such systems.

It is not yet clear in how far our results for the random regular graphs (GCAs) transfer to the regular lattice (CAs). Even though classical CAs are not amenable to the analysis via (B)DCM, a numerical investigation is still possible. Previous work has shown that similar types of attractors and phases do occur on the lattice [7], but our own preliminary empirical investigations did not show an immediate and unambiguous connection. We leave a thorough investigation of these empirics for future work.

Clearly, an analytic method capable of directly handling deterministic CAs directly rather than extrapolating behaviour from the regular GCAs or probabilistic cellular automata [7, 122] is a challenging goal.

Limitations of the (B)DCM. A major drawback of the DCM and BDCM is the exponential computational barrier which depends on the length of the analysed motif $p + c$. Even when the system typically relaxes fast, as previously noted, this limitation may lead to less accurate estimates of the transition [15] as around phase transitions the transient length may increase due to critical slowing down. Therefore, it seems worth investigating if and how approximations to the DCM [5, 11, 33, 68, 147, 167] would give new insights into longer time scales, and if they remain accurate around phase transitions or suffer from similar limitations. Moreover, it is an open task to adapt such approximations to the backtracking version of the DCM.

Large degrees. For very large degrees d, which scale in the size of the graph n, we empirically extrapolate our results. We deduce from the numerics that a scaling of the weak agreement threshold θ approximately as $\sqrt{d}$ maintains the dynamical phase transitions we showed for the small degrees. For the absolute majority and stubborn/volatile independent GCAs we conjecture based on our numerical experiments that only the behaviour that we showed previously will occur. However, preliminary results for the anti-conformist GCAs showed that new types of behaviour emerge when we increase d, hinting at further dynamical phase transitions that require a higher resolution in d to manifest. We leave a thorough investigation of this rule space and its peculiarities for future work.

Short attractors. While for the absolute majority and volatile/stubborn independent GCAs only short attractors of length 1 and 2 can occur [48], we showed that empirically the same holds true for the anti-conformist GCAs on sparse random regular graphs of large size. Based on this evidence, we conjecture that in the large n limit such GCAs typically only has short attractors for finite d. This statement remains to be proven.

Phase transitions and complexity. There has been a plethora of works on dynamics of discrete systems that focus on their complex behaviour – this is typically associated with intriguing visualizations of the systems' space-time diagrams or with the capacity to compute challenging tasks [18, 72, 158]. Many attempts at formalizing the notion of complexity have been given with the general belief that the region of complex behaviour is located at a phase transition between "ordered" and "chaotic" systems [38, 84].

In our work, we do not explore the phase transition in the space of systems. Rather, for a fixed GCA, we describe the phase transition in the space of its initial configurations. This transition becomes particularly interesting for the anti-conformist GCAs that, near the transition, abruptly switch from logarithmic convergence to attractors (associated with simple behaviour) to an exponential one (interpreted as chaotic behaviour) [67]. As such, it becomes very interesting to ask: Is the behaviour of the system near the phase transition qualitatively different? Does it show some signs of "complexity"? From Fig. 3.6, middle, it is apparent that as we increase the system's size, near the phase-transition the system converges to its typical behaviour much more slowly than away from the transition. Thus, in our case, the complexity arises from deciding what type of behaviour the system will settle to near the transition. However, assessing the system's complexity near the transition would require carefully choosing a formal metric of complexity. Therefore, we leave such investigations for possible future work.

Opinion Dynamics. As a side product, we investigated our version of a popular framework from opinion dynamics [52] on a sparse graph. It encompasses a local update rule that seems anecdotally ubiquitous in popular culture: The conforming anti-conformist. This is an agent who only acts in favour of the minority when this minority is not too small, i.e. when the race between the majority and minority is tight. Our analysis showed that such behaviour allows for two opinions to co-exist for a prolonged period of time in the system and thereby maintains a diversity of opinions.

3.7 Larger degree behaviour

We studied the dynamics of all conforming non-conformist rules for connectivity $d = 3$ and $d = 4$ and observed the following general trend shown in Fig. 3.11.

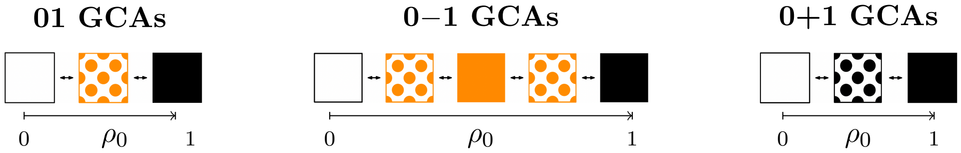


Figure 3.11: Phase diagram scheme for GCAs with rules of type **01**, **0−1** and **0+1**. While for all GCAs the homogeneous stable all-white and all-black phases are at the end of the ρ_{init} spectrum, the intermediate behaviour is qualitatively different. We argue that for each rule type, when increasing ρ_{init} from 0 to 1, the phases occurring always obey the order illustrated in the diagram, though, for degenerate cases, some of the phases might be missing (e.g., the constant 0 GCA only has the homogeneous all-0 phase).

Only the volatile independent rule types **0–1** exhibit both the all-rattling and partially-rattling phases; whereas the stubborn independent rule types only exhibit stable phases. One interesting question is: "How does the phase transition behaviour scale for larger values of d?" Fig. 3.13 illustrates that if the threshold θ remains constant as we increase the connectivity d, the interesting region of ρ_{init} shrinks and almost all initial densities exhibit fast convergence to either the all-0 or all-1 attractor. Eventually, as the degree grows all these rules behave as the majority rule.

Let $k \in \mathbb{N}$ and $\theta \in \mathbb{N}$. We can parameterize the conforming non-conformist rules with connectivity d odd in the following way:

$$\text{stubborn independent: } 0^k +^{2\theta} 1^k$$
$$\text{volatile independent: } 0^k -^{2\theta} 1^k$$
$$\text{anti-conformist: } 0^k 1^\theta 0^\theta 1^k$$

with $d = 2k + 2\theta - 1$. We note that the parameter θ indeed corresponds to the threshold parameter from the definition of CNC rules in Section 3.3. A few examples of the dynamical behaviour for conforming non-conformist rules with larger d and θ are shown in Fig. 3.12. For the stubborn/volatile independent GCAs we observe that if θ scales approximately as $\sqrt{d}$, the phase transitions are preserved (Fig. 3.12, left). Once θ deviates from $\sqrt{d}$ the transitions may collapse and certain phases are no longer present (Fig. 3.13). The situation, however, looks more complicated for the anti-conformist GCAs. In Fig. 3.12, right, we picked very specific values of k and θ, for which the general phase transition trend with the interesting region of alternate behaviour around $\rho_{\text{init}} = 0.5$ is present. We highlight that different values of k and θ yielded new types of behaviour for the anti-conformist GCAs that need further investigation and are left for future work.

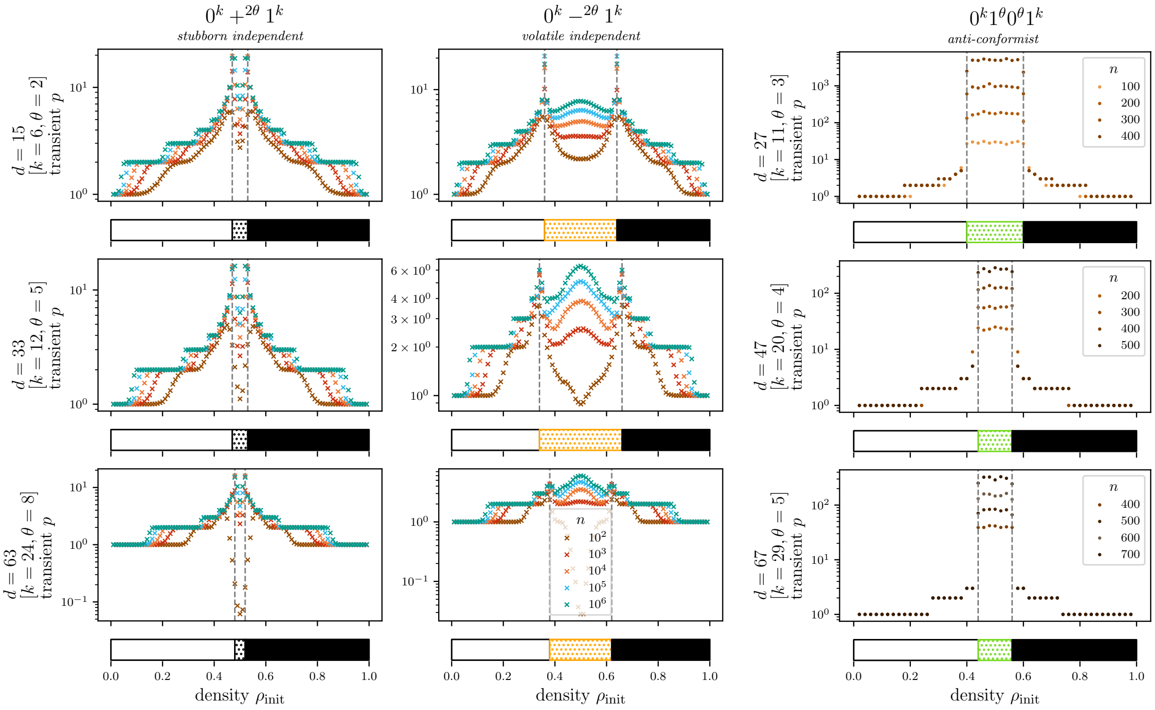


Figure 3.12: **Weak agreement region with scaling (very roughly) in $\theta \sim \sqrt{d}$.** Increasing the degree d for the stubborn/volatile independent GCAs and the anti-conformist GCA, while scaling the weak agreement region (very roughly) as $\sqrt{d}$. Samples were obtained as described in Fig. 3.5.

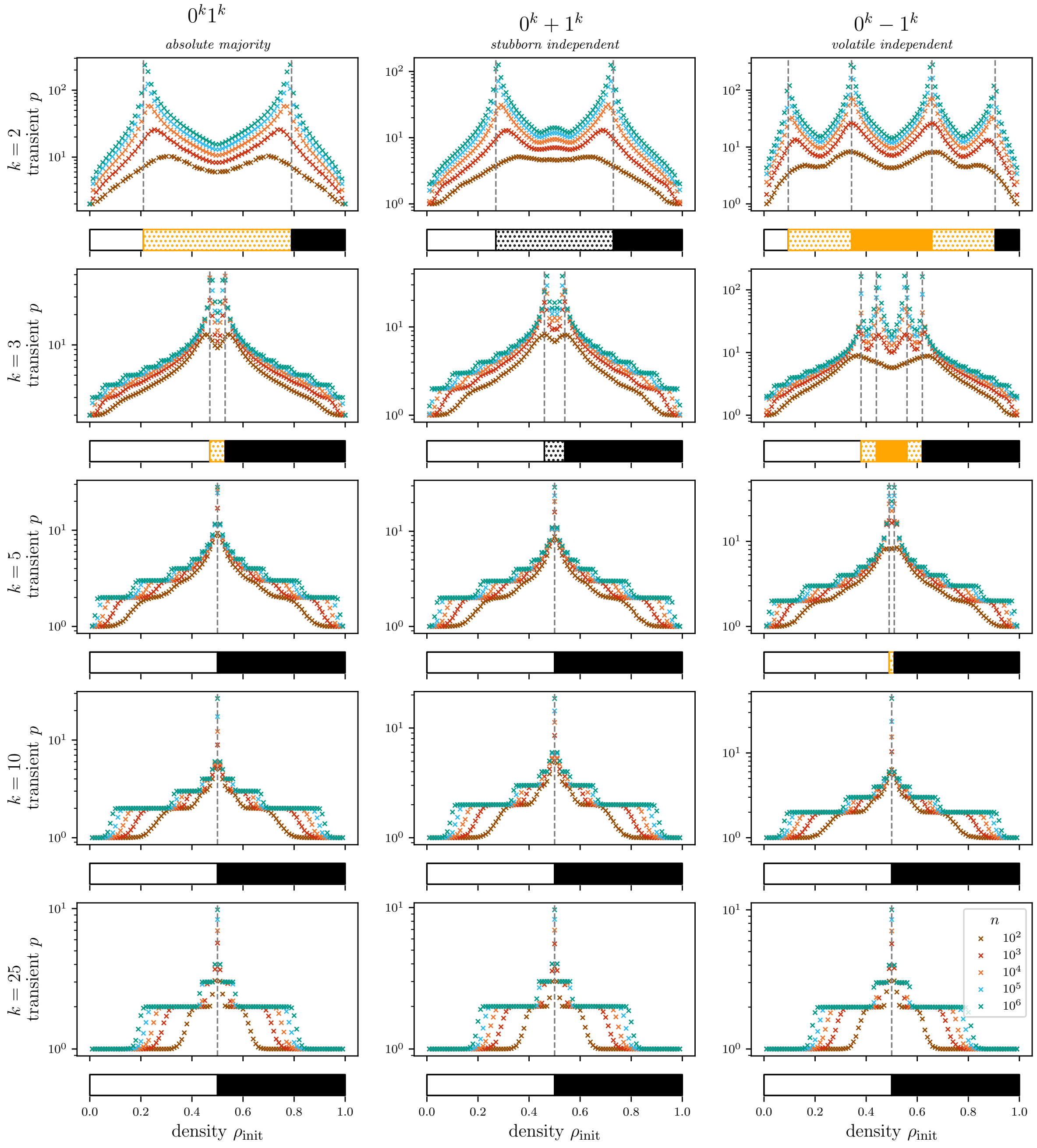


Figure 3.13: **Weak agreement region with** $\theta \in \{0, 1\}$**.** Increasing the degree d for the absolute majority and stubborn/volatile independent GCAs, while keeping the weak agreement region constant. Eventually, all GCAs behave like the absolute majority GCA. Samples were obtained as described in Fig. 3.5.

3.8 Supporting Empirics for Phase Characterization

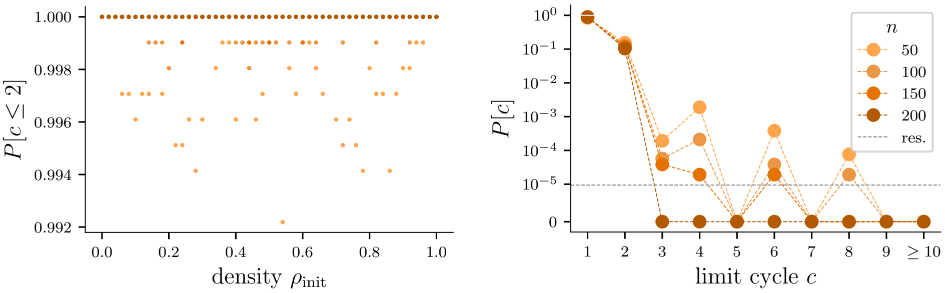


Figure 3.14: **Length of limit cycles for the anti-conformist GCA (code 001011).** We show the length of the limit cycles for the data collected for Fig. 3.5, i.e. 4-regular graphs and different initial densities. *(Left)* Probability that a sample has a limit cycles length $c \leq 2$. Since we only sample few such long limit cycles, we combine the data for the different initializations ρ_{init} on the *(right)*. The dashed grey line shows the minimal resolution we are limited to due to our sample size, which was 1024 for each of the 100 different initial densities.

In Fig. 3.14 we investigate the lengths c of the limit cycles for the anti-conformist GCA (code 001011). There, we almost always sample short attractors. As n grows, the number of large limit cycles drops rapidly. This leads us to the conjecture, that in the large n limit, for the anti-conformist GCAs, any attractors with sizes $c > 2$ will become vanishingly unlikely.

3.9 Supporting Material for Dynamical Phase Transition Predictions using the (B)DCM and Empirical Methods

In Fig. 3.15 we show yet another property of the chaotic phase of anti-conformist GCAs. The graph suggests the evolution of distances of two close-by initial configurations follows a (pseudo) random walk.

In Fig. 3.16 we demonstrate that the value of the densities observed during the chaotic phase to an attractor, fall into the interval between the dynamical transition lines.

In Fig. 3.17 we show how we extrapolated the $p = \infty$ behaviour from the first 7 time steps using the DCM for the anti-conformist GCA 001011.

We do the same for the absolute majority GCA 0011, in addition to precise numerics in Fig. 3.17.

Similarly, the BDCM results for the stubborn independent GCA 00+11 are extrapolated in Fig. 3.19.

In Fig. 3.20 we base our selection of the threshold to distinguish the all and partially rattling phase on numerical evidence that shows how the scaling of the activity α behaves differently for each ρ_{init}.

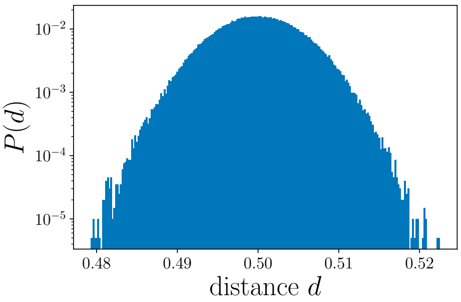


Figure 3.15: **Illustration of chaotic phase for anti-conformist GCA** 001011**: evolution of distances of two close-by trajectories.** For GCA 001011 we randomly generated an initial configuration with length $n = 10000$, and density $\rho_{\text{init}} = 0.5$ and a close-by configuration with $\epsilon \cdot n$ different bits; $\epsilon = 0.01$. While both configurations' trajectories remain in the chaotic regime, we measure their average Hamming distance and plot the probabilities for 100 such experiments averaged.

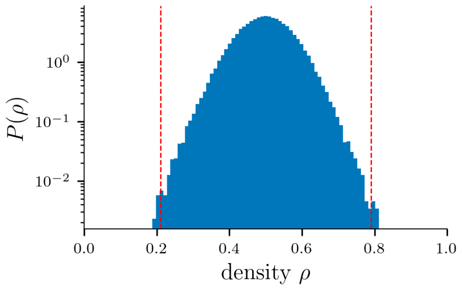


Figure 3.16: **Histogram of the density ρ for transients with a chaotic behaviour (after relaxation and before convergence to an attractor) for the anti-conformist GCA** 001011**.** We sample 1024 graphs of size $n = 100$ with the dynamics of the rule 001011 started at $\rho_{\text{init}} = 0.2$. The first 100 time steps after the start of the dynamics and last 100 time steps before reaching the attractor are removed for every sampled trajectory. Since the convergence is fast in the phase where an attractor is reached rapidly (see Fig.3.5), such a cut-off effectively removes all transients that converge rapidly (less than 200 time steps long). The histogram shows the density of the remaining trajectory which exhibits a chaotic behaviour. The red lines mark the phase transition measured empirically for rule 001011 between the phase of rapid convergence to the homogeneous state (left and right side) and the chaotic phase (between the red lines). Note that during this time almost all samples lie within the regime of the chaotic phase.

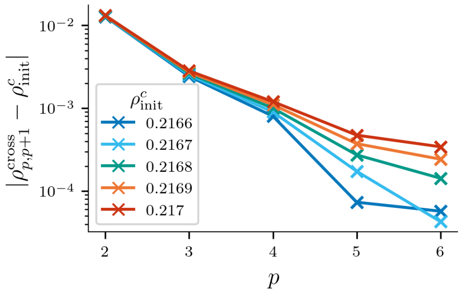


Figure 3.17: **Anti-conformist GCA** 001011 – **Extrapolation.** Extrapolation of the crossover points from Fig. 3.7 to $p \to \infty$. The plot shows the distance to the critical ρ^{c}_{init}, the possible location of phase transition points. Under the assumption of an exponentially fast convergence in p, the best fit seems to lie at roughly $\rho^{c}_{\text{init}} = 0.2168 \pm 0.0001$.

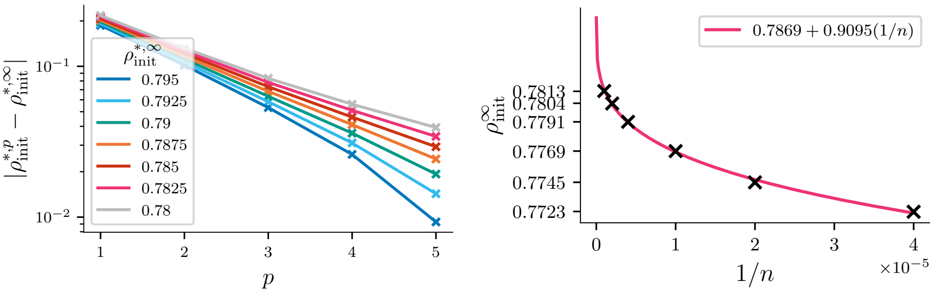


Figure 3.18: **Absolute majority rule** 0011 **– Extrapolation.** Extrapolation of the theoretical BDCM and empirical experiments to $n \to \infty$ and $p \to \infty$ respectively. *(Left)* Scaling of the distance to different values of extrapolated $\rho_{\text{init}}^{*,\infty}$ for data for $p = 1, ..., 5$ for the BDCM. Assuming exponentially fast convergence in p, a $\rho_{\text{init}}^{*,\infty} \sim 0.7875 \pm 0.005$ is reasonable. *(Right)* The location ρ_{init} of the slowest average convergence for experiments over $4,096$ samples of random regular graphs and initial configurations for a given graph size n are recorded, and then extrapolated to $n \to \infty$. With this we estimate that in the large system limit, the transition happens at $\sim 0.785 \pm 0.005$.

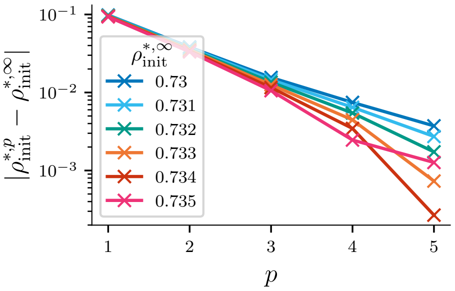


Figure 3.19: **Stubborn independent GCA** 00+11 **– Extrapolation.** As in Fig.3.18 we show different extrapolations of the BDCM's predictions for dynamical phase transitions for fixed p and extrapolate them to $p \to \infty$, concluding that convergence within $\sim 0.73 \pm 0.005$. The extrapolation for the empirics is taken from [15].

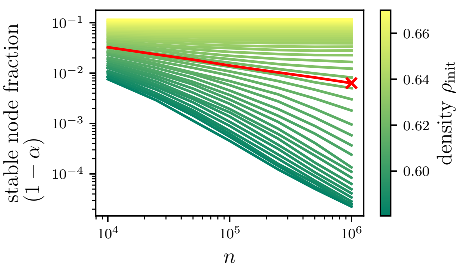


Figure 3.20: **Scaling of the non-rattling nodes for different ρ_{init}.** For some ρ_{init} the non-rattling nodes scales logarithmic in $o(n)$, for others they are a constant fraction in $\Theta(n)$. The red value marks the intermediate threshold we selected to classify the all-rattling and partially rattling phases in Fig. 3.10 (right), which had 0.7% of stable nodes.

4. Simulation Limitations of Affine Cellular Automata

4.1 Introduction

Cellular Automata (CAs) are a famous model of computation which demonstrates that very simple local rules can produce complex behaviour. In general, they have a great potential to solve challenging tasks efficiently due to their massively parallel nature. However, it is a long-standing challenge to formally assess the computational capacity of a given CA.

The usual formal method of demonstrating a CA's computational power is to prove its Turing completeness. Typically, this is done by taking a classical universal model of computation (Turing machine, tag system, etc.) and embedding its computations into the space-time diagrams of the given CA. These embeddings serve as impressive demonstrations that cellular automata are indeed complex systems. Moreover, the embeddings serve as powerful tools for constructing very compact computationally universal systems. However, there are several drawbacks to this approach.

1. Cellular automata are highly parallel systems, thus, it is ineffective to simulate a sequential Turing machine in its space-time diagrams to perform computations [118].

2. Given a CA, showing that it is Turing complete is very laborious. Most proofs of this sort rely on noticing localized structures in the CAs' space-time diagrams whose interactions are typically encoded as basic logic gates [27, 130].

3. Despite several attempts [36, 39, 143], there is no single generally accepted formal definition of simulation so far [118]. Thus, it is extremely difficult to prove convincing negative results.

A different approach to formally assessing a CA's computational capacity is through the notion of *CA relative simulation*. Informally, we say that CA $\mathcal{A}$ is simulated by $\mathcal{B}$ if each space-time diagram of $\mathcal{A}$ can be, after suitable transformations, reproduced by $\mathcal{B}$. We argue that comparing two cellular automata is much more natural than comparing a CA with a Turing machine, since in the latter case, the architectures of the systems differ substantially. Past works have explored various notions of CA simulations, typically focusing on positive results: for a fixed family of CAs and a fixed CA simulation definition, authors construct *intrinsically universal CAs*; i.e., cellular automata that are able to simulate any other CA within the fixed family [1, 39, 40, 41, 49, 116, 117].

In contrast, the complementary work focuses on the negative results. For various notions of CA simulations, the goal is to show that particular types of automata are limited in terms of what they can simulate. Such results are scarce, yet they bring a valuable insight into the structure of the CA space imposed by the simulation relation. For certain CA simulation definitions, negative results have been shown about various classes of CAs such as nilpotent CAs or particular additive automata [35, 97, 98].

Generally, each CA relative simulation definition considers a certain class of CA transformations T that map each automaton $\mathcal{B}$ into a class of related automata $\mathrm{T}(\mathcal{B})$. Then, we say that $\mathcal{A}$ can be simulated by $\mathcal{B}$ if $\mathcal{A} \in \mathrm{T}(\mathcal{B})$. We propose an informal classification of the previously studied transformations into:

- algebraic: transformations of the CA's local rule; e.g.: products, sub-automata, quotiens, iterations

- geometric: transformations of the CA's grid structure; e.g.: tiling of the grid space and grouping of multiple cells, shifts, reflections

In this work, we propose a definition of CA simulation that is, to the best of our knowledge, the most general algebraic one so far. We define the CA simulation and study its properties in Section 4.2. The importance of this section lies in formalizing all the CA notions in abstract algebraic language. This will allow us to see connections to well-established algebraic fields that can provide powerful tools for analysing the CA simulation capacities.

In Section 4.3, we introduce the class of affine CAs – automata whose local rules are affine mappings of vector spaces. This general class contains the much studied additive CAs [54, 70, 96]. Therein, we present Theorem 36 – the main result of this paper, paraphrased below:

Let $\mathcal{B}$ be a cellular automaton affine over a finite field $\mathbb{F}_p$. If $\mathcal{B}$ satisfies a certain mild condition ("the outer components of its local rule are bijective"), then all CAs simulated by $\mathcal{B}$ are also affine over $\mathbb{F}_p$ (and satisfy the same condition).

Thus, we show that affine CAs have very limited computational capacity. We note that the mild condition does not impose any important limitations as it is satisfied by almost all previously studied, non-trivial cases of affine CAs. We also note that our main result already implies the negative results about additive CAs proved previously in the literature [35, 97]. Even when applied to some of the best known additive automata, our result gives considerably more than was known before. To give one concrete example, it yields that ECA 60 [85] cannot simulate any other elementary cellular automaton (up to isomorphism); this is discussed in Example 54.

Section 4.4 contains elementary proofs of the results leading up to Theorem 36. In Section 4.5, we point towards a connection to deeper results from universal algebra. We believe that further study of connections between general algebra and CAs could be fruitful and provide insightful results about the computational limitations of various cellular automata classes.

4.2 Defining Simulation of Cellular Automata

Definition 15 (Cellular automaton)**.** Let S be a finite set, $r \in \mathbb{N}$, and $f : S^{2r+1} \to S$ a function. Let $S^{\mathbb{Z}}$ denote the space of all bi-infinite sequences $c = \cdots c_{-1}c_0c_1\cdots$, $c_i \in S$ for each $i \in \mathbb{Z}$. A *one-dimensional cellular automaton (CA) with set of states S, radius r, and local rule f* is a dynamical system $\mathcal{A} = (S^{\mathbb{Z}}, F)$ where $F : S^{\mathbb{Z}} \to S^{\mathbb{Z}}$ is defined for any $c \in S^{\mathbb{Z}}$ and any position $i \in \mathbb{Z}$ as:

$$F(c)_i = f(c_{i-r}, \ldots, c_{i-1}, c_i, c_{i+1}, \ldots, c_{i+r}). \tag{4.1}$$

We call $S^{\mathbb{Z}}$ the *configuration space of $\mathcal{A}$* and F its *global rule.* We also call $(S^{\mathbb{Z}}, F)$ the *global algebra.* Each 1D CA is determined by its *local algebra* $\mathbb{A} = (S, f)$. Sometimes, we also write $\mathbb{A} = (S, f)_r$ to highlight the CA's radius. A *space-time diagram* of $\mathcal{A}$ with initial configuration $c \in S^{\mathbb{Z}}$ and t time-steps is a matrix whose rows are exactly $c, F(c), \ldots, F^t(c)$. Visualizations of CA space-time diagrams give valuable insights into the CA's dynamics. When depicting a space-time diagram, we only show a finite part of each row.

In this paper, we focus on 1D cellular automata. Thus, whenever we talk about a CA, we implicitly mean a 1-dimensional one. The definitions as well as most of the results of this paper can be generalised to higher dimensions quite straightforwardly, but the technical details would be tedious. Before we proceed with the definitions of CA simulation, we briefly review some simple algebraic concepts.

Let S be a set, $k \in \mathbb{N}$, and $f : S^k \to S$ a function. We call the tuple $A = (S, f)$ *an algebra of type k*. Let $B = (T, g)$ be an algebra also of type k. We say that:

- A is *isomorphic* to B if there exists a bijection $\varphi : S \to T$ such that for each $s_1, \ldots, s_k \in S$:
$$\varphi(f(s_1, \ldots, s_k)) = g(\varphi(s_1), \ldots, \varphi(s_k)).$$
We write $A \cong B$.
- A is a *subalgebra* of B if $S \subseteq T$ and $g\big|_{S^k} = f$.
- Let $\sim \subseteq T \times T$ be an equivalence relation. We call $\sim$ a *congruence* on B if:
$$g(t_1, \ldots, t_k) \sim g(t'_1, \ldots, t'_k) \text{ whenever } t_1 \sim t'_1, \ldots, t_k \sim t'_k \text{ for any } t_i, t'_i \in T, 1 \leq i \leq k.$$
We denote by $[t]_\sim = \{t' \in T \mid t' \sim t\}$ the *equivalence class* of $t \in T$. There is a well-defined algebra $B\big/_\sim = (U, h)$ where $U = \{[t]_\sim \mid t \in T\}$ and $h : U^k \to U$ is defined as $h([t_1]_\sim, \ldots, [t_k]_\sim) \coloneqq [g(t_1, \ldots, t_k)]_\sim$. We say that $\mathcal{A}$ is a *quotient algebra* of B if $\mathcal{A} = B\big/_\sim$ for some congruence $\sim$ on B.
- Let $B_1 = (T_1, g_1), \ldots, B_n = (T_n, g_n)$ all be algebras of type k. We define their *product* $B_1 \times \cdots \times B_n$ as the algebra $(T_1 \times \cdots \times T_n, h)$ where $h(\mathbf{u}_1, \ldots, \mathbf{u}_k)_i = g_i(u^i_1, \ldots, u^i_k)$ for each $1 \leq i \leq k$, $\mathbf{u}_i \in T_1 \times \cdots \times T_n$, $\mathbf{u}_i = u^1_i \cdots u^n_i$.

4.2.1 CA Canonical Relations

We start by defining canonical relations between CAs. We first do this in a purely algebraic manner and subsequently motivate the definitions by relating them to the CAs' global dynamics.

Definition 16 (CA canonical relations)**.** Let $\mathcal{A} = (S^{\mathbb{Z}}, F)$ and $\mathcal{B} = (T^{\mathbb{Z}}, G)$ be CAs with local algebras $\mathbb{A} = (S, f)_r$ and $\mathbb{B} = (T, g)_r$ respectively. We say that:

- *$\mathcal{A}$ is automaton-isomorphic to $\mathcal{B}$* if $\mathbb{A}$ is isomorphic to $\mathbb{B}$.
- *$\mathcal{A}$ is a sub-automaton of $\mathcal{B}$* if $\mathbb{A}$ is a subalgebra of $\mathbb{B}$.
- *$\mathcal{A}$ is a quotient automaton of $\mathcal{B}$* if $\mathbb{A}$ is a quotient algebra of $\mathbb{B}$.
- Let $\mathcal{A}_1, \ldots, \mathcal{A}_k$ be CAs with local algebras $\mathbb{A}_1 = (S_1, f_1)_r, \ldots, \mathbb{A}_k = (S_k, f_k)_r$ respectively. We define their *product* $\mathcal{A}_1 \times \cdots \times \mathcal{A}_k$ to be the CA given by the local algebra $\mathbb{A}_1 \times \cdots \times \mathbb{A}_k$.

It is natural to consider automata up to isomorphism. This gives rise to the natural algebraic operators we define below for a class $\mathbf{K}$ consisting of local algebras of some family of cellular automata with radius $r \in \mathbb{N}$.

$$\begin{aligned} \mathrm{S}(\mathbf{K}) &= \{\mathbb{A} \mid \exists \mathbb{B} \in \mathbf{K} \text{ and } \mathbb{C} \text{ subalgebra of } \mathbb{B} \text{ such that } \mathbb{A} \cong \mathbb{C}\} \\ \mathrm{H}(\mathbf{K}) &= \{\mathbb{A} \mid \exists \mathbb{B} \in \mathbf{K} \text{ and } \mathbb{C} \text{ quotient algebra of } \mathbb{B} \text{ such that } \mathbb{A} \cong \mathbb{C}\} \\ \mathrm{P}_{\mathrm{fin}}(\mathbf{K}) &= \{\mathbb{A} \mid \exists \mathbb{B}_1, \ldots, \mathbb{B}_k \in \mathbf{K} \text{ such that } \mathbb{A} \cong \mathbb{B}_1 \times \cdots \times \mathbb{B}_k\}. \end{aligned}$$

The operators S, H, and $\mathrm{P}_{\mathrm{fin}}$ transform a single local algebra $\mathbb{A}$ into classes of algebras $\mathrm{S}(\mathbb{A})$, $\mathrm{H}(\mathbb{A})$, and $\mathrm{P}_{\mathrm{fin}}(\mathbb{A})$. Subsequently, we will compose such operators and therefore, we generally define them on a class of local algebras $\mathbf{K}$ rather than on a single local algebra $\mathbb{A}$.

The CA canonical relations of Definition 16 relate the local algebras of cellular automata. As such, they do not explicitly describe how this relationship translates to the CAs' global dynamics. The crucial observation is the following: Suppose that $\mathcal{A}$ is a CA isomorphic to $\mathcal{B}$, is a sub-automaton of $\mathcal{B}$ or is a quotient automaton of $\mathcal{B}$. Then, for any space-time diagram $\mathbf{c}$ of $\mathcal{A}$, there exists a space-time diagram $\mathbf{d}$ of $\mathcal{B}$ which can be "translated" to $\mathbf{c}$ via a very simple mapping. Thus, $\mathcal{B}$ can effectively reproduce any dynamics of $\mathcal{A}$. Before we formalize this observation, we introduce some further terminology.

Let $\varphi : S \to T$ be a mapping between finite sets. We define its *canonical extension* $\overline{\varphi} : S^{\mathbb{Z}} \to T^{\mathbb{Z}}$ simply as $\overline{\varphi}(c)_i = \varphi(c_i)$ for each $c \in S^{\mathbb{Z}}$ and each $i \in \mathbb{Z}$.

Observation 17. *Let $\mathcal{A} = (S^{\mathbb{Z}}, F)$ and $\mathcal{B} = (T^{\mathbb{Z}}, G)$ be CAs with local algebras $\mathbb{A} = (S, f)_r$ and $\mathbb{B} = (T, g)_r$ respectively. Then, it holds that:*

1. *$\mathbb{A} \cong \mathbb{B}$ if and only if there exists a bijection $\varphi : S \to T$ such that $\overline{\varphi} \circ F = G \circ \overline{\varphi}$.*

2. *$\mathbb{A} \in \mathrm{S}(\mathbb{B})$ if and only if there exists an injective mapping $\iota : S \to T$ such that $\overline{\iota} \circ F = G \circ \overline{\iota}$.*

3. *$\mathbb{A} \in \mathrm{H}(\mathbb{B})$ if and only if there exists a surjective mapping $\pi : T \to S$ such that $F \circ \overline{\pi} = \overline{\pi} \circ G$.*

Below, we show an example illustrating the notion of a quotient automaton.

Example 18. Let $\mathcal{B} = (T^{\mathbb{Z}}, G)$ be the CA with local algebra $\mathbb{B} = (\{0,1,2,3\}, g)_1$ defined as $g(x, y, z) = (x + z) \bmod 4$. We consider $\sim \subseteq \mathbb{Z}_4 \times \mathbb{Z}_4$ defined as $x_1 \sim x_2$ if and only if x_1, x_2 have the same parity. It is clear that whenever $x_1 \sim x_2$, $y_1 \sim y_2$, and $z_1 \sim z_2$ for any $x_1, x_2, y_1, y_2, z_1, z_2 \in \{0,1,2,3\}$, then $g(x_1, y_1, z_1) \sim g(x_2, y_2, z_2)$. Thus, $\sim$ is a congruence on $\mathbb{B}$ and we can study $\mathbb{B}/\sim$.

Let $\mathcal{A} = (S^{\mathbb{Z}}, F)$ be the CA with local algebra $\mathbb{A} = (\{0,1\}, f)_1$ where $f(x, y, z) = (x + z) \bmod 2$ ($\mathcal{A}$ is the ECA 90). We define two mappings:

$$\begin{array}{rcl} \varphi : \ \mathbb{B}/\sim & \longrightarrow & \mathbb{A} \\ & [0]_\sim = \{0,2\} \mapsto 0 & \\ & [1]_\sim = \{1,3\} \mapsto 1 & \end{array} \qquad \begin{array}{rcl} \pi : \ \mathbb{B} & \longrightarrow & \mathbb{A} \\ & 0,2 \mapsto 0 & \\ & 1,3 \mapsto 1 & \end{array}$$

It is straightforward to verify that φ is an isomorphism between $\mathbb{B}/\sim$ and $\mathbb{A}$. Thus, it witnesses that $\mathbb{A} \in \mathrm{H}(\mathbb{B})$. Moreover, one can easily check that the canonical extension of π satisfies $F \circ \overline{\pi} = \overline{\pi} \circ G$ (thus, we have explicitly found the map π from Observation 17, part 3.). Figure 4.1 illustrates how any space-time diagram of $\mathcal{A}$ can be obtained from a suitable space-time diagram of $\mathcal{B}$ using $\overline{\pi}$ as the "translation mapping".

Let us fix the notation from Observation 17. The mappings $\overline{\varphi}, \overline{\iota}$, and $\overline{\pi}$ provide means of "translating" between the space-time diagrams of $\mathcal{A}$ and $\mathcal{B}$. It is crucial that the mappings can be efficiently implemented by a computer program as they are extensions of mappings on finite sets. Moreover, the simplicity of the mappings guarantees that they do not process the information contained in the space-time diagrams in any non-trivial way. This is particularly important since, e.g., whenever $\mathbb{A} \in \mathrm{S}(\mathbb{B})$, we would like to conclude that $\mathcal{B}$ is computationally stronger or equal to $\mathcal{A}$.

In contrast to the CA isomorphism, we can consider a general isomorphism between the algebras $(S^{\mathbb{Z}}, F)$ and $(T^{\mathbb{Z}}, G)$ that can be witnessed by an arbitrary mapping $\psi : S^{\mathbb{Z}} \to T^{\mathbb{Z}}$ (e.g., a non-recursive one). In such a case, we say that $\mathcal{A}$ and $\mathcal{B}$ are *isomorphic as dynamical systems.*

The operators S, H, and $\mathrm{P}_{\mathrm{fin}}$ allow us to compare CAs' local rules. However, they do not take into account the most important aspect of cellular automata: the iterative application of the local rules. Thus, we describe below the established notion of *CA iterative powers* also called *grouping* [98]. This notion naturally extends the possible relationships we can study between two automata and, together with the S, H, and $\mathrm{P}_{\mathrm{fin}}$ operators, will lead to the definition of CA simulation.

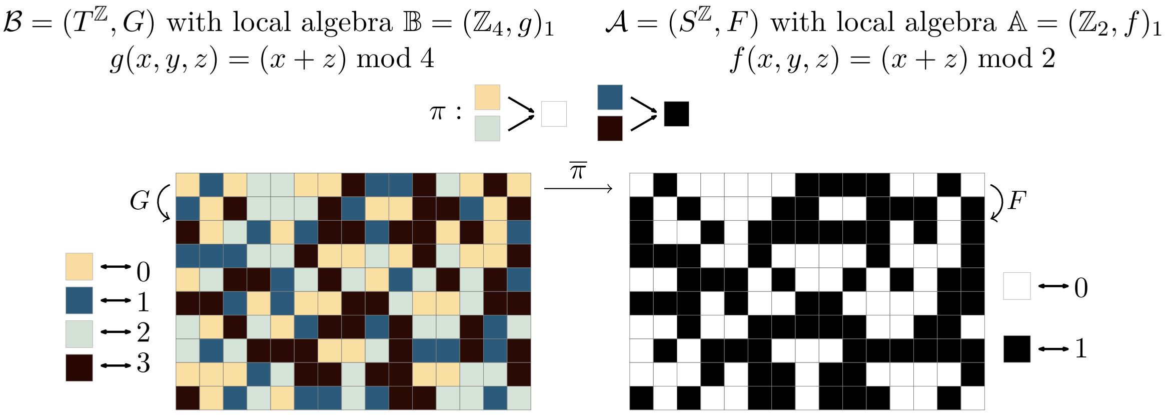

$\mathcal{B} = (T^{\mathbb{Z}}, G)$ with local algebra $\mathbb{B} = (\mathbb{Z}_4, g)_1$
$g(x, y, z) = (x + z) \bmod 4$
$\mathcal{A} = (S^{\mathbb{Z}}, F)$ with local algebra $\mathbb{A} = (\mathbb{Z}_2, f)_1$
$f(x, y, z) = (x + z) \bmod 2$
π :
$\overline{\pi}$
G
F
0
1
2
3
0
1

Figure 4.1: Illustration of Example 18. The figure shows that when $\mathbb{A} \in \mathrm{H}(\mathbb{B})$, there exists a canonical extension $\overline{\pi}$ which effectively "translates" space-time diagrams of $\mathcal{B}$ to any given diagram of $\mathcal{A}$.

4.2.2 Iterative Powers of CAs

Let $\mathcal{B} = (T^{\mathbb{Z}}, G)$ be a CA with local algebra $\mathbb{B} = (T, g)_r$. It is natural to iterate the global rule G to obtain its powers G^n for each $n \in \mathbb{N}$. This yields the "iterated" automaton $(T^{\mathbb{Z}}, G^n)$. The goal of this section is to describe the construction of the local algebra $\mathbb{B}^{[n]}$ corresponding to the iterated automaton $(T^{\mathbb{Z}}, G^n)$ in such a way that the type of $\mathbb{B}$ and $\mathbb{B}^{[n]}$ remains the same.

We start by defining a function $\widetilde{g}$ which can be seen as an intermediate step between a CA's local map g (with inputs of fixed length $2r + 1$) and its global map G (acting on infinite sequences).

Definition 19. (Unravelling a local function) Let T be a finite set and $g : T^{2r+1} \to T$ a function. We extend g to a mapping $\widetilde{g} : \bigcup_{k=2r+1}^{\infty} T^k \to \bigcup_{k=1}^{\infty} T^k$ defined as:

$$\widetilde{g}(u_1 \cdots u_k) = g(u_1, \ldots, u_{2r+1}) g(u_2, \ldots, u_{2r+2}) \cdots g(u_{k-2r}, \ldots, u_k)$$

for any $u_1 \cdots u_k \in T^k$, $k \geq 2r + 1$. By $\widetilde{g}^n$ we simply mean the composition $\widetilde{g}^n = \underbrace{\widetilde{g} \circ \widetilde{g} \circ \cdots \circ \widetilde{g}}_{n\times}$.

Each application of $\widetilde{g}$ shortens the input sequence by $2r$. Therefore, the n-th iteration shortens the input by $2nr$ and $\widetilde{g}^n$ can be seen as a $(2nr{+}1)$-ary function function into T, formally $\widetilde{g}^n|_{T^{2nr+1}} : T^{2nr+1} \to T$. This is illustrated in Figure 4.2.

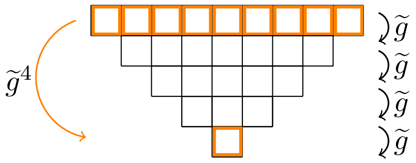


Figure 4.2: Illustration of $\widetilde{g}$ and $\widetilde{g}^n$ for a ternary $(r = 1)$ function g and $n = 4$.

It is not hard to verify that the iterated CA $(T^{\mathbb{Z}}, G^n)$ has the local algebra $(T, \widetilde{g}^n|_{T^{2nr+1}})_{nr}$ with type $2nr + 1$. Its type is different from $(T, g)_r$ whenever $n > 1$. Our goal is to introduce a simulation relation between two automata via the notions of automata quotients, sub-automata, finite products, and iterative powers. Therefore, we need all operators to preserve the type of the CAs' local algebras. For iterative powers, this can be achieved by "grouping together sequences of n consecutive states". This is formally defined below.

Definition 20. Let $\mathcal{B} = (T^{\mathbb{Z}}, G)$ be a CA with local algebra $\mathbb{B} = (T, g)_r$ and let $n \in \mathbb{N}$. We define the *unpacking map* $o_n : (T^n)^{\mathbb{Z}} \to T^{\mathbb{Z}}$ for any configuration $c \in (T^n)^{\mathbb{Z}}$ and any $j \in \mathbb{Z}$, $j = nq + r$, $r < n$, as $\left(o_n\left(c\right)\right)_j = (c_q)_r$.

This is illustrated in Figure 4.3.

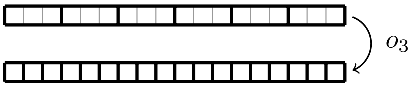


Figure 4.3: Diagram of the unpacking map o_3.

Definition 21. Let $\mathcal{B} = (T^{\mathbb{Z}}, G)$ be a CA with local algebra $\mathbb{B} = (T, g)_r$ and let $n \in \mathbb{N}$. We define the global map $G^{[n]} : (T^n)^{\mathbb{Z}} \to (T^n)^{\mathbb{Z}}$ as $G^{[n]} = o_n^{-1} \circ G^n \circ o_n$. This yields a new dynamical system $\left((T^n)^{\mathbb{Z}}, G^{[n]}\right)$. Further, we define the function $g^{[n]} : \underbrace{T^n \times \cdots \times T^n}_{(2r+1)\times} \to T^n$ as:

$$g^{[n]}(\mathbf{x}_{-r}, \ldots, \mathbf{x}_r) \coloneqq \widetilde{g}^n(x^1_{-r} \cdots x^n_{-r} x^1_{-r+1} \cdots x^n_{-r+1} \cdots x^1_r \cdots x^n_r)$$

where for each i, $\mathbf{x}_i = x^1_i \cdots x^n_i \in T^n$. The function $g^{[n]}$ has arity $2r + 1$ and is illustrated in Figure 4.4.

Observation 22. *Let $\mathcal{B} = (T^{\mathbb{Z}}, G)$ be a CA with local algebra $\mathbb{B} = (T, g)_r$ and let $n \in \mathbb{N}$. Then, $\left((T^n)^{\mathbb{Z}}, G^{[n]}\right)$ is a CA with local algebra $(T^n, g^{[n]})$ and radius r.*

Clearly, there is no automaton-isomorphism between $(T^{\mathbb{Z}}, G^n)$ and $\left((T^n)^{\mathbb{Z}}, G^{[n]}\right)$ since their state spaces have different sizes. However, they are isomorphic as dynamical systems via the unpacking mapping $o_n : (T^n)^{\mathbb{Z}} \to T^{\mathbb{Z}}$. Thus, the two systems do have equivalent dynamics.

Definition 23 (Iterative power of an automaton)**.** Let $\mathcal{B} = (T^{\mathbb{Z}}, G)$ be a CA with local algebra $\mathbb{B} = (T, g)_r$. For $n \in \mathbb{N}$, we define the n-th *iterative power* of $\mathcal{B}$ as the automaton $\mathcal{B}^{[n]} = \left((T^n)^{\mathbb{Z}}, G^{[n]}\right)$ with local algebra $\mathbb{B}^{[n]} = (T^n, g^{[n]})_r$.

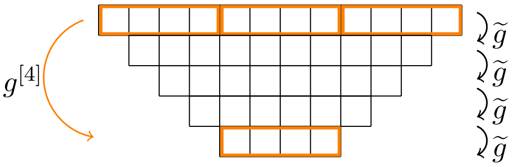


Figure 4.4: Illustration of $g^{[4]}$ for a ternary function g.

In this way, the local algebra $\mathbb{B}^{[n]}$ describes exactly n iterations of the CA's local rule while preserving the type of $\mathbb{B}$. Therefore, each local algebra $\mathbb{B}$ of a CA gives rise to a series of algebras of the same type, but operating on “larger scales” in both state space and time:

$$\mathbb{B} = (T, g)_r,\ \mathbb{B}^{[2]} = (T^2, g^{[2]})_r,\ \mathbb{B}^{[3]} = (T^3, g^{[3]})_r,\ \ldots$$

Definition 24. Let $\mathbf{K}$ be a class of local algebras of a family of CAs with radius r. We define the *iterative power* operator:

$$\Xi(\mathbf{K}) = \{\mathbb{A} \mid \exists \mathbb{B} \in \mathbf{K} \text{ and } n \in \mathbb{N} \text{ such that } \mathbb{A} \cong \mathbb{B}^{[n]}\}.$$

In contrast to the operators H, S, and $\mathrm{P}_{\mathrm{fin}}$ that are algebraic in their nature, Ξ can be viewed as a geometrical transformation of a CA's space-time diagrams. Indeed, the n-th iterative power essentially "selects" every n-th row of the space-time diagram, and groups the "cells" together into packages of n consecutive cells via the o_n^{-1} operator. We can finally proceed to introducing the general algebraic definition of CA simulation.

Definition 25. [CA simulation] Let $\mathcal{A}$ and $\mathcal{B}$ be cellular automata with local algebras $\mathbb{A} = (S, f)_r$ and $\mathbb{B} = (T, g)_r$ respectively. We say that $\mathcal{B}$ *simulates* $\mathcal{A}$ if $\mathbb{A} \in \mathrm{HSP}_{\mathrm{fin}}\Xi(\mathbb{B})$; and we write $\mathbb{A} \preceq \mathbb{B}$ or $\mathcal{A} \preceq \mathcal{B}$.

Note that the simulation relation is well-defined only for cellular automata with the same radius. However, whenever we consider an automaton $\mathcal{A}$ with radius r, we can also interpret it as a CA with radius $r' \in \mathbb{N}$ for any $r' > r$ in a natural way. This allows us to compare automata with different radii afterall.

A perhaps more natural definition of simulation would be to require the following properties: First, $\mathcal{A} \preceq \mathcal{B}$ if at least one of the following properties hold:

1. $\mathbb{A} \in \mathrm{S}(\mathbb{B})$; $\mathcal{A}$ is (isomorphic to) a sub-automaton of $\mathcal{B}$.

2. $\mathbb{A} \in \mathrm{H}(\mathbb{B})$; $\mathcal{A}$ is (isomorphic to) a quotient automaton of $\mathcal{B}$.

3. $\mathbb{A} \in \Xi(\mathbb{B})$; $\mathcal{A}$ is (isomorphic to) an iterative power of $\mathcal{B}$.

4. $\mathbb{A} \in \mathrm{P}_{\mathrm{fin}}\left(\mathrm{H}(\mathbb{B}) \cup \mathrm{S}(\mathbb{B}) \cup \Xi(\mathbb{B})\right)$; $\mathcal{A}$ is (isomorphic to) a finite product of automata which are themselves "simulated" by $\mathcal{B}$.

This is well-motivated, since in all these cases, any computation in $\mathcal{A}$ can be easily recovered from a computation in $\mathcal{B}$. But any reasonable definition of simulation also must be transitive. In the following subsection we prove that the transitive closure of the requirements above exactly yields the CA simulation in Definition 25.

4.2.3 Elementary Properties of CA Simulation

Lemma 26. *Let $\mathbb{A}$ be a local algebra of a CA $\mathcal{A}$ and let $m, n \in \mathbb{N}$. Then, $(\mathbb{A}^{[m]})^{[n]} \cong \mathbb{A}^{[mn]}$.*

Proof. Let $\mathcal{A} = (S^{\mathbb{Z}}, F)$. Then by definition $\mathcal{A}^{[mn]} = \left((S^{mn})^{\mathbb{Z}}, F^{[mn]}\right)$; $F^{[mn]} = o_{mn}^{-1} \circ F^{mn} \circ o_{mn}$, where $o_{mn} : (S^{mn})^{\mathbb{Z}} \to S^{\mathbb{Z}}$ is the unpacking map. Similarly, $(\mathcal{A}^{[m]})^{[n]} = \left(((S^m)^n)^{\mathbb{Z}}, (F^{[m]})^{[n]}\right)$; $(F^{[m]})^{[n]} = o_n^{-1} \circ (o_m^{-1} \circ F^m \circ o_m)^n \circ o_n$, where $o_m : S^m \to S$ and $o_n : (S^m)^n \to S^m$. Let $\varphi : S^{mn} \to (S^m)^n$ be the natural bijection. We will show that its extension $\overline{\varphi}$ satisfies $\overline{\varphi}^{-1} \circ (F^{[m]})^{[n]} \circ \overline{\varphi} = F^{[mn]}$. Then, Observation 17 already implies that the corresponding local algebras are isomorphic. It is easy to verify that $o_m \circ o_n \circ \overline{\varphi} = o_{mn}$. Then:

$$\begin{aligned}
\overline{\varphi}^{-1} \circ (F^{[m]})^{[n]} \circ \overline{\varphi} &= \overline{\varphi}^{-1} \circ o_n^{-1} \circ \left(o_m^{-1} \circ F^m \circ o_m\right)^n \circ o_n \circ \overline{\varphi} \\
&= \overline{\varphi}^{-1} \circ o_n^{-1} \circ o_m^{-1} \circ F^{mn} \circ o_m \circ o_n \circ \overline{\varphi} \\
&= o_{mn}^{-1} \circ F^{mn} \circ o_{mn} = F^{[mn]}.
\end{aligned}$$

□

Lemma 27. *Let $\mathbb{A}_1, \dots, \mathbb{A}_k$ be local algebras of some CAs with radius r and let $n \in \mathbb{N}$. Then $(\mathbb{A}_1 \times \cdots \times \mathbb{A}_k)^{[n]} \cong \mathbb{A}_1^{[n]} \times \cdots \times \mathbb{A}_k^{[n]}$.*

Proof. Let $\mathbb{A}_i = (S_i, f_i)_r$ for $1 \leq i \leq k$. Similarly to Lemma 26, it is a straightforward, yet technical verification that the natural bijection $\varphi : (S_1 \times \cdots \times S_k)^n \to S_1^n \times \cdots \times S_k^n$ is an isomorphism witnessing that $(\mathbb{A}_1 \times \cdots \times \mathbb{A}_k)^{[n]} \cong \mathbb{A}_1^{[n]} \times \cdots \times \mathbb{A}_k^{[n]}$. □

Proposition 28. *Let* $\mathbf{K}$ *be the class of local algebras of some class of CAs with radius r. Then,* $\mathrm{HSP}_{\mathrm{fin}}\Xi(\mathbf{K})$ *is already closed under the operators* $\mathrm{H}, \mathrm{S}, \mathrm{P}_{\mathrm{fin}}$*, and* Ξ.

Proof. Let $\mathbf{K}'$ be an arbitrary class of local algebras of the same type. It is a well-known result from universal algebra [17] that:

$$\begin{array}{lll} \mathrm{HH}(\mathbf{K}') = \mathrm{H}(\mathbf{K}'), & \mathrm{SS}(\mathbf{K}') \quad = \mathrm{S}(\mathbf{K}'), & \mathrm{P_{fin}P_{fin}}(\mathbf{K}') \quad = \mathrm{P_{fin}}(\mathbf{K}') \\ \mathrm{SH}(\mathbf{K}') \subseteq \mathrm{HS}(\mathbf{K}'), & \mathrm{P_{fin}H}(\mathbf{K}') \subseteq \mathrm{HP_{fin}}(\mathbf{K}'), & \mathrm{P_{fin}S}(\mathbf{K}') \subseteq \mathrm{SP_{fin}}(\mathbf{K}'). \end{array}$$

Moreover, Lemma 26 implies $\Xi\Xi(\mathbf{K}') = \Xi(\mathbf{K}')$ and Lemma 27 yields $\Xi\mathrm{P_{fin}}(\mathbf{K}') \subseteq \mathrm{P_{fin}}\Xi(\mathbf{K}')$. Further, in [98, Lemma 2] the authors showed: If $\mathbb{A} \in \mathrm{S}(\mathbb{B})$ for any local algebras $\mathbb{A}$, $\mathbb{B}$ of the same type, then $\mathbb{A}^{[n]} \in \mathrm{S}(\mathbb{B}^{[n]})$ for every $n \in \mathbb{N}$. Thus, $\Xi\mathrm{S}(\mathbf{K}') \subseteq \mathrm{S}\Xi(\mathbf{K}')$. In [35], the authors outline the proof of $\Xi\mathrm{H}(\mathbf{K}') \subseteq \mathrm{H}\Xi(\mathbf{K}')$. We give an explicit proof here.

Let $\mathbb{A} = (S, f)_r$ and $\mathbb{B} = (T, g)_r$ be such that $\mathbb{A} \in \mathrm{H}(\mathbb{B})$ and let $n \in \mathbb{N}$. We show that $\mathbb{A}^{[n]} \in \mathrm{H}(\mathbb{B}^{[n]})$. Since $\mathbb{A} \in \mathrm{H}(\mathbb{B})$, there is some surjective homomorphism $\varphi : T \twoheadrightarrow S$ of the local algebras. We extend it to a surjective mapping $\psi : T^n \twoheadrightarrow S^n$ simply by putting $\psi(\mathbf{x})_i = \varphi(x_i)$ for any $\mathbf{x} \in T^n$ and $i \in \{1, \dots, n\}$. Now it remains to verify that $\psi : \mathbb{A}^{[n]} \to \mathbb{B}^{[n]}$ is a surjective homomorphism of the algebras. First, let $m \geq 2r + 1$ and $x_1, \dots, x_m \in T$. We show that $\widetilde{f}(\widetilde{\varphi}(x_1 \cdots x_m)) = \widetilde{\varphi}(\widetilde{g}(x_1 \cdots x_m))$. Clearly, for every $r \leq i \leq m - r$, it holds that $\widetilde{f}(\widetilde{\varphi}(x_1 \cdots x_m))_i = f(\varphi(x_{i-r}) \cdots \varphi(x_{i+r})) = \varphi(g(x_{i-r}, \dots, x_{i+r})) = \widetilde{\varphi}(\widetilde{g}(x_1 \cdots x_m))_i$. Further, by induction we obtain that for every $n \in \mathbb{N}$ and every sequence $x_1, \dots, x_m \in T$, $m \geq 2nr + 1$, it holds that $\widetilde{f}^n(\widetilde{\varphi}(x_1 \cdots x_m)) = \widetilde{\varphi}(\widetilde{g}^n(x_1 \cdots x_m))$. Next, let $\mathbf{x}_{-r}, \dots, \mathbf{x}_r \in T^n$. We have:

$$f^{[n]}(\psi(\mathbf{x}_{-r}), \dots, \psi(\mathbf{x}_r)) = \widetilde{f}^n(\widetilde{\varphi}(\mathbf{x}_{-r} \cdots \mathbf{x}_r)) = \widetilde{\varphi}(\widetilde{g}^n(\mathbf{x}_{-r} \cdots \mathbf{x}_r)) = \psi(g^{[n]}(\mathbf{x}_{-r}, \dots, \mathbf{x}_r)).$$

Thus, indeed, $\psi : T^n \twoheadrightarrow S^n$ is a surjective homomorphism of algebras $\mathbb{A}^{[n]}$ and $\mathbb{B}^{[n]}$, and $\mathbb{A}^{[n]} \in \mathrm{H}(\mathbb{B}^{[n]})$. This already implies that $\Xi\mathrm{H}(\mathbf{K}') \subseteq \mathrm{H}\Xi(\mathbf{K}')$.

Now we are ready to finish the proof that $\mathrm{HSP_{fin}}\Xi(\mathbf{K})$ is already closed under $\mathrm{H}, \mathrm{S}, \mathrm{P_{fin}}$, and Ξ. We show this for the operator Ξ:

$$\Xi\mathrm{HSP_{fin}}\Xi(\mathbf{K}) \subseteq \mathrm{H}\Xi\mathrm{SP_{fin}}\Xi(\mathbf{K}) \subseteq \mathrm{HS}\Xi\mathrm{P_{fin}}\Xi(\mathbf{K}) \subseteq \mathrm{HSP_{fin}}\Xi\Xi(\mathbf{K}) = \mathrm{HSP_{fin}}\Xi(\mathbf{K})$$

where the first inclusion uses that $\Xi\mathrm{H}(\mathbf{K}') \subseteq \mathrm{H}\Xi(\mathbf{K}')$ for $\mathbf{K}' = \mathrm{SP_{fin}}\Xi(\mathbf{K})$ and the subsequent inclusions follow similar logic. For the other operators, the proof is analogous and takes less than one line, using the results summarized above. □

Corollary 29. *The CA simulation relation $\preceq$ is reflexive and transitive; i.e., it forms a preorder.*

Proof. The reflexivity of $\preceq$ is clear. We briefly discuss the transitivity. Let $\mathbb{A}$, $\mathbb{B}$, $\mathbb{C}$ be local algebras of CAs all with radius $r \in \mathbb{N}$ such that $\mathbb{A} \preceq \mathbb{B}$ and $\mathbb{B} \preceq \mathbb{C}$. By definition, $\mathbb{A} \in \mathrm{HSP_{fin}}\Xi(\mathbb{B})$ and $\mathbb{B} \in \mathrm{HSP_{fin}}\Xi(\mathbb{C})$. Thus, $\mathbb{A} \in \mathrm{HSP_{fin}}\Xi(\mathbb{B}) \subseteq \mathrm{HSP_{fin}}\Xi\mathrm{HSP_{fin}}\Xi(\mathbb{C}) = \mathrm{HSP_{fin}}\Xi(\mathbb{C})$ where the last equality holds due to Proposition 28. Hence, $\mathbb{A} \preceq \mathbb{C}$. □

Definition 30. Let $r \in \mathbb{N}$. We define SING_r to be the class of local algebras of the form $(S, f)_r$ where $|S| = 1$.

Clearly, any two local algebras in SING_r are isomorphic. Moreover, let $\mathbb{A} \in \mathrm{SING}_r$, and consider any local algebra $\mathbb{B} = (S, f)_r$ of a CA. Then, $\mathbb{A} \preceq \mathbb{B}$ since $\mathbb{A} \in \mathrm{H}(\mathbb{B})$. Thus, SING_r forms the minimum element within the class of all 1D CAs with radius r.

We note that our definition of CA simulation combines all classical algebraic operators that preserve the finiteness of the algebras. This has the following advantage, which emphasizes the connection to the field of universal algebra: For any class of local algebras $\mathbf{K}$ of CAs with radius r, $\mathrm{HSP_{fin}}\Xi(\mathbf{K})$ forms a so-called *pseudovariety*. Thus, as shown in [42], $\mathrm{HSP_{fin}}\Xi(\mathbf{K})$ can be characterized by a sequence of equations. Finding such equation sequences for different classes of CAs can provide important insight into the structure of the simulation relation and is an interesting line of future work.

4.3 Introducing Additive and Affine Automata

Additive automata are a much-studied class of CAs. Studying the automata simulated by them leads naturally to a broader class of *affine automata*, which we now introduce.

Definition 31 (affine CA, additive CA)**.** Let $\mathbb{F}$ be a finite field, V a finite-dimensional vector space over $\mathbb{F}$, and let $\mathcal{A} = (V^{\mathbb{Z}}, F)$ be a CA with local algebra $(V, f)_r$. We say that $\mathbb{A}$ (or $\mathcal{A}$) is *affine over* $\mathbb{F}$ if $f : V^{2r+1} \to V$ is an affine mapping between vector spaces over $\mathbb{F}$. In such a case, we can write f in the following form:

$$f(\mathbf{x}_{-r}, \dots, \mathbf{x}_r) = f_{-r}(\mathbf{x}_{-r}) + \cdots + f_r(\mathbf{x}_r) + \mathbf{c}, \tag{4.2}$$

where $f_i : V \to V$ is a linear mapping for each $-r \le i \le r$ and $\mathbf{c} \in V$ is a constant vector. The mapping f_i is called the *i-th component of f*. The class of all local algebras isomorphic to some affine local algebra over $\mathbb{F}$ with radius r is denoted as $\mathrm{AFF}_r^{\mathbb{F}}$.

In the special case when $f : V^{2r+1} \to V$ is a linear mapping between vector spaces over $\mathbb{F}$, we say that $\mathbb{A}$ (or $\mathcal{A}$) is *additive over* $\mathbb{F}$. In such a case, we can write f as in (4.2) with $\mathbf{c} = \mathbf{0}$. We denote the class of all local algebras isomorphic to some local algebra additive over $\mathbb{F}$ with radius r as $\mathrm{ADD}_r^{\mathbb{F}}$. We say that $\mathbb{A}$ is *canonical additive (or affine)* if it is additive (or affine) over $\mathbb{F}_p$ for some prime p and $V = \mathbb{F}_p$.

Example 32. Let $(\mathbb{F}_2^{\mathbb{Z}}, F)$ be the elementary CA 150 with local algebra $\mathbb{A} = (\mathbb{F}_2, f)_1$ where $f : \mathbb{F}_2^3 \to \mathbb{F}_2$ is defined as $f(x, y, z) = x + y + z \bmod 2$. Then, $\mathbb{A}$ is a CA additive over $\mathbb{F}_2$; in fact, it is canonical additive.

Automata additive over finite fields of the form $\mathbb{F}_p$, p prime, have been widely studied. In fact, they form one of the few classes of cellular automata that are amenable to algebraic analysis which yields rigorous results about their global dynamics. To name a few important results not directly related to CA simulation, the properties of the additive CAs' global dynamics have been carefully analysed in the seminal work [96] and consequently in [3, 54, 70, 145]. We remark that additive automata are sometimes also called *linear* in the literature.

The class of affine automata is a natural generalization of additive CAs that occurs when we study sub-automata of additive CAs. As shown further in the next section, it is the case that sub-automata of additive CAs need not be additive but are in general affine. To get acquainted with the two notions, we start with a simple lemma and an example.

Lemma 33. *Let $\mathbb{F}$ be a finite field and $r \in \mathbb{N}$. Let $\mathbb{A} = (V, f)_r$ be such that $\mathbb{A} \in \mathrm{AFF}_r^{\mathbb{F}}$. Then, $\mathbb{A} \in \mathrm{ADD}_r^{\mathbb{F}}$ if and only if there exists an idempotent element of f; i.e., there exists $\mathbf{v} \in V$ such that $f(\mathbf{v}, \dots, \mathbf{v}) = \mathbf{v}$.*

Proof. Each additive CA has $\mathbf{0}$ as an idempotent element. In the other direction, let $\mathbb{A} = (V, f)_r$ be such that $\mathbb{A} \in \mathrm{AFF}_r^{\mathbb{F}}$, and suppose that $\mathbf{v} \in V$ is an idempotent of f. Let $f_{-r}, \dots, f_r$ be the components of f. We define the bijection $\varphi : V \to V$ as $\varphi(\mathbf{v} + \mathbf{x}) = \mathbf{x}$ for every $\mathbf{x} \in V$. Next, we define $g : V^{2r+1} \to V$ as $g(\mathbf{x}_{-r}, \dots, \mathbf{x}_r) \coloneqq \sum_{i=-r}^{r} f_i(\mathbf{x}_i)$. Then, the CA with local algebra $(V, g)_r$ is additive, and it is straightforward to verify that $(V, f)_r \cong (V, g)_r$ via φ; indeed, for any $\mathbf{x}_{-r}, \dots, \mathbf{x}_r$ we have:

$$\varphi(f(\mathbf{x}_{-r} + \mathbf{v}, \dots, \mathbf{x}_r + \mathbf{v})) = \varphi(f(\mathbf{v}, \dots, \mathbf{v}) + \sum_{i=-r}^{r} f_i(\mathbf{x}_i)) = \sum_{i=-r}^{r} f_i(\mathbf{x}_i) =$$
$$= \sum_{i=-r}^{r} f_i(\varphi(\mathbf{x}_i + \mathbf{v})) = g(\varphi(\mathbf{x}_{-r} + \mathbf{v}), \dots, \varphi(\mathbf{x}_r + \mathbf{v})). \qquad \square$$

Example 34. $\mathrm{ADD}_1^{\mathbb{F}_2} \subsetneq \mathrm{AFF}_1^{\mathbb{F}_2}$. An example of an affine CA that is not isomorphic to any additive one is the elementary CA 105 with local algebra $(\mathbb{F}_2, f)_1$ where $f(x, y, z) = (x + y + z + 1) \bmod 2$ for any $x, y, z \in \mathbb{F}_2$. It is clear that neither 0 nor 1 are idempotent elements of f.

Many of the results presented in this paper hold for affine CAs with at least two components that are bijections. A special case of such rules are the *left- or right-permutive CAs* that are well studied in the literature [81]. Below, we define the left- and right-permutivity for affine automata in a slightly more general way compared to the classical definition to account for the fact that we only study cellular automata with a symmetrical neighbourhood given by a particular radius.

Definition 35. Let $\mathbb{A} = (V, f)_r$ be an affine local algebra of a CA with radius r whose local rule f has components $f_{-r}, \dots, f_r$. We say that $\mathbb{A}$ satisfies the *bijective condition* if at least two of the components are bijections. We say that $\mathbb{A}$ is *left-permutive, witnessed by i,* if there exists $-r \leq i \leq r$ such that f_i is a bijection and f_k is the constant $\mathbf{0}$ mapping for all $k < i$. Similarly, $\mathbb{A}$ is *right-permutive, witnessed by j,* if there exists $-r \leq j \leq r$ such that f_j is a bijection and f_k is the constant $\mathbf{0}$ mapping for all $k > j$.

We write that $\mathbb{A} \in \mathrm{AFF}^{\mathbb{F}}_{r;i,j}$ if $\mathbb{A} \in \mathrm{AFF}^{\mathbb{F}}_{r}$ and at the same time $\mathbb{A}$ is left-permutive witnessed by i and right-permutive witnessed by j, $i < j$. Clearly, all such algebras satisfy the bijective condition. Analogously, we define the subclass $\mathrm{ADD}^{\mathbb{F}}_{r;i,j}$.

Now we can state the main result of this paper that we prove in Section 4.4:

Theorem 36. *Let p be a prime, $r \in \mathbb{N}$, and $-r \leq i < j \leq r$. Let $\mathbb{A}$, $\mathbb{B}$ be local algebras of cellular automata with radius r such that $\mathbb{B} \in \mathrm{AFF}^{\mathbb{F}_p}_{r;i,j}$. If $\mathbb{A} \preceq \mathbb{B}$, then $\mathbb{A} \in \mathrm{AFF}^{\mathbb{F}_p}_{r;i,j}$.*

4.3.1 Related Work on CA Simulation

In their seminal work [98], Mazoyer and Rapaport study the properties of a strongly related notion of CA simulation based on iterative powers and sub-automata. In [97], they specifically focus on showing that a particular class of canonical additive CAs is limited in terms of what it can simulate. Subsequently, their work was continued in [34], where the authors introduce a generalized version of the iterative powers of CAs that we denote here by $\widetilde{\Xi}$ for clarity. Informally, $\widetilde{\Xi}$ allows for much more general geometrical transformations of the CA space-time diagrams. Whereas Ξ is based on grouping together blocks of consecutive "cells" via the operators o_n, $\widetilde{\Xi}$ allows for groupings of much more general patterns, as well as, e.g., shifts of the CA's configuration space. In their subsequent work [35], the same authors introduce various notions of CA simulation; the most general one is roughly defined below.

Definition 37 (CA Simulation: Delorme, Mazoyer, Ollinger, Theyssier [35])**.** Let $\mathcal{A}$ and $\mathcal{B}$ be two 1D CAs both with radius $r \in \mathbb{N}$. For their local algebras, we write $\mathbb{A} \preceq_{\mathrm{m}} \mathbb{B}$ if $\widetilde{\Xi}(\mathbb{A}) \cap \mathrm{HS}\widetilde{\Xi}(\mathbb{B}) \neq \emptyset$.

Both lines of work ([97, 98] and [34, 35]) analyse various important properties of the different CA simulations. This includes studying the simulation limitations of a special family of additive automata, namely those with local algebras of the form $\mathcal{Z}_p = (\mathbb{F}_p, f)_1$ where $f(x, y, z) = (x + y + z) \bmod p$. Their negative result is that whenever $p \neq q$, one has $\mathcal{Z}_p \not\preceq_{\mathrm{m}} \mathcal{Z}_q$. The authors use two observations:

1. Let $\mathbb{A}$, $\mathbb{B}$ be local algebras of CAs with radius r. Then, $\widetilde{\Xi}(\mathbb{A}) \cap \mathrm{HS}\widetilde{\Xi}(\mathbb{B}) \neq \emptyset$ if and only if $\widetilde{\Xi}(\mathbb{A}) \cap \mathrm{HS}\Xi(\mathbb{B}) \neq \emptyset$.

2. Let $\mathbb{A}$ be a local algebra of a CA with p states. Then the algebras in $\widetilde{\Xi}(\mathbb{A})$ have p^k states, $k \geq 1$.

Given the observations, the authors reduce the problem of showing $\mathcal{Z}_p \not\preceq_{\mathrm{m}} \mathcal{Z}_q$ for $p \neq q$ to proving that any algebra in $\mathrm{HS}\Xi(\mathcal{Z}_q)$ has q^l elements for some $l \geq 0$. Assuming this extra piece of knowledge, comparing the sizes of algebras in $\mathrm{HS}\Xi(\mathcal{Z}_q)$ and in $\widetilde{\Xi}(\mathcal{Z}_p)$ already implies the negative result, as $p^k \neq q^l$ for $k \geq 1$, $l \geq 0$.

The crucial information about possible sizes of algebras in $\mathrm{HS}\Xi(\mathcal{Z}_q)$ follows immediately from our results. (This is not surprising, since Theorem 36 provides a lot of information about the structure of these algebras.) Indeed: For every prime q, clearly $\mathcal{Z}_q \in \mathrm{ADD}^{\mathbb{F}_q}_{1;-1,1}$. By Theorem 36, any local algebra in $\mathrm{HS}\Xi(\mathcal{Z}_q)$ belongs to $\mathrm{AFF}^{\mathbb{F}_q}_{1;-1,1}$; therefore it has an underlying structure of a vector space over $\mathbb{F}_q$ and hence, it has q^l elements.

4.4 Simulation Limitations of Additive and Affine Automata

In this section, we prove Theorem 36. In more detail, we will show that for any finite field $\mathbb{F}_p$, p prime, any $r \in \mathbb{N}$, and any $-r \leq i < j \leq r$, it holds that:

$$\mathrm{HSP}_{\mathrm{fin}}\Xi(\mathrm{AFF}^{\mathbb{F}_p}_{r;i,j}) = \mathrm{AFF}^{\mathbb{F}_p}_{r;i,j}, \tag{4.3}$$

$$\mathrm{HSP}_{\mathrm{fin}}\Xi(\mathrm{ADD}^{\mathbb{F}_p}_{r;i,j}) = \mathrm{AFF}^{\mathbb{F}_p}_{r;i,j}. \tag{4.4}$$

Theorem 36 is then a direct consequence of (4.3). Concretely, we will show this result by studying how the operators Ξ, $\mathrm{P}_{\mathrm{fin}}$, S, and H change the sets $\mathrm{AFF}^{\mathbb{F}_p}_{r;i,j}$ and $\mathrm{ADD}^{\mathbb{F}_p}_{r;i,j}$.

Lemma 38. *Let $\mathbb{A}$ be a local algebra of a CA that is affine (or additive) over a finite field $\mathbb{F}$ and let $n \in \mathbb{N}$. Then $\mathbb{A}^{[n]}$ is again affine (or additive) over $\mathbb{F}$.*

Proof. Let $\mathbb{A} = (V, f)_r$ where f has linear components $f_{-r}, \ldots, f_r$ and $f(\mathbf{x}_{-r}, \ldots, \mathbf{x}_r) = f_{-r}(\mathbf{x}_{-r}) + \cdots + f_r(\mathbf{x}_r) + \mathbf{c}$ for some constant $\mathbf{c} \in V$; $\mathbf{x}_{-r}, \ldots, \mathbf{x}_r \in V$. We put $g(\mathbf{x}_{-r}, \ldots, \mathbf{x}_r) \coloneqq f(\mathbf{x}_{-r}, \ldots, \mathbf{x}_r) - \mathbf{c}$ and $\mathbb{B} \coloneqq (V, g)_r$. Clearly, $\mathbb{B}$ is additive over $\mathbb{F}$. Let $n \in \mathbb{N}$; we first show that $\mathbb{B}^{[n]}$ is also additive. It is straightforward to verify that for each $k \geq 2r+1$, $\widetilde{g}|_{V^k} : V^k \to V^{k-2r}$ is a linear mapping of vector spaces. Thus, since composition of linear mappings is linear, we have that $\widetilde{g}^n|_{V^{n(2r+1)}} : V^{n(2r+1)} \to V^n$ is also linear. It is then a straightforward yet slightly technical step to verify that $g^{[n]}$ is a linear mapping and thus, that $\mathbb{B}^{[n]}$ is additive over $\mathbb{F}$.

Next, we observe that $\widetilde{f}(\mathbf{x}_1 \cdots \mathbf{x}_k) = \widetilde{g}(\mathbf{x}_1 \cdots \mathbf{x}_k) + \underbrace{\mathbf{c}\,\mathbf{c} \cdots \mathbf{c}\,\mathbf{c}}_{k-2r\,\times}$ and that $\widetilde{f}((\mathbf{x}_1 \cdots \mathbf{x}_k) + (\mathbf{y}_1 \cdots \mathbf{y}_k)) = \widetilde{g}(\mathbf{x}_1 \cdots \mathbf{x}_k) + \widetilde{f}(\mathbf{y}_1 \cdots \mathbf{y}_k)$ for any $k \geq 2r+1$ and any $\mathbf{x}_1, \ldots, \mathbf{x}_k, \mathbf{y}_1, \ldots, \mathbf{y}_k \in V$. Combining these two facts, it is straightforward to verify by induction on n that for any $k \geq 2nr+1$

$$\widetilde{f}^n(\mathbf{x}_1 \cdots \mathbf{x}_k) = \widetilde{g}^n(\mathbf{x}_1 \cdots \mathbf{x}_k) + \widetilde{f}^{n-1}(\underbrace{\mathbf{c}\,\mathbf{c} \cdots \mathbf{c}\,\mathbf{c}}_{k-2r\,\times})$$

for each $\mathbf{x}_1, \ldots, \mathbf{x}_k \in V$. For $k = n(2r+1)$ we get that $f^{[n]}(\cdot) = g^{[n]}(\cdot) + \mathbf{d}$ for a constant $\mathbf{d} \in V^n$ and thus, that $\mathbb{A}^{[n]}$ is affine over $\mathbb{F}$. □

Lemma 39. *Let $\mathbb{A}$ be a local algebra of a CA affine over $\mathbb{F}$. If $\mathbb{A}$ is left-permutive witnessed by i then for each $n \in \mathbb{N}$, $\mathbb{A}^{[n]}$ is also left-permutive witnessed by i. Analogous property holds for right-permutive local algebras.*

Proof. This is a straightforward generalization of Lemma 3 from [97]. □

We note that the notion of left and right-permutivity can be defined for arbitrary CAs, not just for affine ones. In such a case, the statement in Lemma 39 can be generalized to the broader definition as the proof does not rely in any fundamental way on the fact that the CA's local rule is affine.

Corollary 40. *Let $\mathbb{F}$ be an arbitrary finite field, $r \in \mathbb{N}$, and $-r \leq i < j \leq r$. Then:*

$$\Xi(\mathrm{AFF}^{\mathbb{F}}_{r}) = \mathrm{AFF}^{\mathbb{F}}_{r}, \qquad \Xi(\mathrm{ADD}^{\mathbb{F}}_{r}) = \mathrm{ADD}^{\mathbb{F}}_{r},$$
$$\Xi(\mathrm{AFF}^{\mathbb{F}}_{r;i,j}) = \mathrm{AFF}^{\mathbb{F}}_{r;i,j}, \qquad \Xi(\mathrm{ADD}^{\mathbb{F}}_{r;i,j}) = \mathrm{ADD}^{\mathbb{F}}_{r;i,j}.$$

Observation 41. *Let $\mathbb{A}_1, \dots, \mathbb{A}_k$ be local algebras of CAs with radius r that are affine (or additive) over a finite field $\mathbb{F}$. Then, $\mathbb{A} = \mathbb{A}_1 \times \cdots \times \mathbb{A}_k$ is again affine (or additive) over $\mathbb{F}$. Moreover, let $-r \leq i < j \leq r$. If the algebras $\mathbb{A}_1, \dots, \mathbb{A}_k$ all have their i-th and j-th components bijective, then so does $\mathbb{A}$.*

Corollary 42. *Let $\mathbb{F}$ be an arbitrary finite field, $r \in \mathbb{N}$, and $-r \leq i < j \leq r$. It holds that:*

$$\mathrm{P_{fin}}(\mathrm{AFF}^{\mathbb{F}}_r) = \mathrm{AFF}^{\mathbb{F}}_r, \qquad \mathrm{P_{fin}}(\mathrm{ADD}^{\mathbb{F}}_r) = \mathrm{ADD}^{\mathbb{F}}_r,$$
$$\mathrm{P_{fin}}(\mathrm{AFF}^{\mathbb{F}}_{r;i,j}) = \mathrm{AFF}^{\mathbb{F}}_{r;i,j}, \qquad \mathrm{P_{fin}}(\mathrm{ADD}^{\mathbb{F}}_{r;i,j}) = \mathrm{ADD}^{\mathbb{F}}_{r;i,j}.$$

Thus, we have shown that both the operators Ξ and $\mathrm{P_{fin}}$ preserve the class of $\mathrm{AFF}^{\mathbb{F}}_{r;i,j}$ for each finite field $\mathbb{F}$, each radius $r \in \mathbb{N}$, and each $-r \leq i < j \leq r$. Below, we show similar results for the operators S and H under the assumption that the affine automata satisfy the bijective condition. In what follows, p always denotes a prime number.

4.4.1 Sub-automata of affine CAs

In this section, we study the operator S on the class of affine automata. Compared to the part about $\mathrm{P_{fin}}$ and Ξ, our results need more assumptions: We only work over the prime fields $\mathbb{F}_p$ and we require the bijective condition (see Definition 35). The importance of the assumptions is illustrated in Example 51 where we exhibit an affine CA violating the bijective condition that contains a sub-automaton that is not affine.

We start by noticing that certain invariant subspaces produce a natural family of sub-automata.

Observation 43. *Let $\mathbb{B} = (V, f)_r$ be the local algebra of a CA affine over $\mathbb{F}_p$. Suppose that $W \leq V$ is a subspace invariant under all components of f and that $\mathbf{v} \in V$ satisfies $f(\mathbf{v}, \dots, \mathbf{v}) \in \mathbf{v} + W$. Then $\mathbb{A} = (\mathbf{v} + W, f|_{(\mathbf{v}+W)^{2r+1}})_r$ belongs to $S(\mathbb{B})$.*

Assuming the bijective condition, the observation can be turned into an equivalence – every sub-automaton is of the simple form described above.

Proposition 44. *Let $\mathcal{B}$ be a CA with local algebra $\mathbb{B} = (V, f)_r$ that is affine over $\mathbb{F}_p$ and satisfies the bijective condition. Let $\mathcal{A}$ be a sub-automaton of $\mathcal{B}$ with the local algebra $\mathbb{A} = (U, f|_{U^{2r+1}})_r$. Then, $U = \mathbf{v} + W$ for a subspace $W \leq V$ invariant under all components of f and $\mathbf{v} \in \mathbf{V}$ such that $f(\mathbf{v}, \dots, \mathbf{v}) \in \mathbf{v} + W$.*

Proof. Let $f_{-r}, \dots, f_r$ be the components of f and $f(\mathbf{x}_{-r}, \dots, \mathbf{x}_r) = \sum_{i=-r}^{r} f_i(\mathbf{x}_i) + \mathbf{c}$ for some $\mathbf{c} \in V$. Let us fix $-r \leq i < j \leq r$ such that f_i and f_j are bijections. We put $U_i := f_i(U) \subseteq V$ and $U_j := f_j(U) \subseteq V$ and we note that $|U_i| = |U_j| = |U| =: k$. We first show that both U_i and U_j are affine subspaces of V. We have that:

$$k = |U| \geq |f(U, \dots, U)| = |f_{-r}(U) + \cdots + f_r(U)| \geq |f_i(U) + f_j(U)| = |U_i + U_j| \geq |U_i| = k.$$

Thus, we also have $|U_i + U_j| = k$. Let $\mathbf{u}_i \in U_i$ and $\mathbf{u}_j \in U_j$. We put $W_i := U_i - \mathbf{u}_i$ and $W_j := U_j - \mathbf{u}_j$. Then clearly $\mathbf{0} \in W_i$ and $\mathbf{0} \in W_j$ and moreover, since translations do not change the set sizes, $|W_i| = |W_j| = |W_i + W_j| = k$. Hence, $W_i + W_j$ is equal to some set W and $W \supseteq W_i + \mathbf{0} = W_i$. Since W_i has the same size as W, this yields $W_i = W$. Analogously, we have $W_j = W$ and thus $W = W_i = W_j = W_i + W_j = W + W$. This proves that W is closed under addition, which is over $\mathbb{F}_p$ sufficient to make it a vector subspace of V. Finally, $U_i = \mathbf{u}_i + W$ and $U_j = \mathbf{u}_j + W$ are both affine subspaces of V, one being just a translation of the other.

Now, pick any $\mathbf{u} \in U$ and put $\mathbf{v} := \left(\sum_{l=-r}^{r} f_l(\mathbf{u})\right) - f_i(\mathbf{u}) + \mathbf{u}_i + \mathbf{c}$. Then:

$$U \supseteq f(U, \dots, U) \supseteq f_{-r}(\mathbf{u}) + \cdots + f_{i-1}(\mathbf{u}) + U_i + f_{i+1}(\mathbf{u}) + \cdots + f_r(\mathbf{u}) + \mathbf{c} = \mathbf{v} + W.$$

Since $|U| = |W|$, we get $U = \mathbf{v} + W$.

It is left to show that the subspace W is invariant under all components of f. For any fixed k with $-r \leq k \leq r$, we show that $f_k(W) \leq W$:

$$\mathbf{v} + W \supseteq f(\mathbf{v} + W, \ldots, \mathbf{v} + W) = f(\mathbf{v}, \ldots, \mathbf{v}) + f_{-r}(W) + \cdots + f_r(W) \supseteq f(\mathbf{v}, \ldots, \mathbf{v}) + f_k(W).$$

Since $f_k(W)$ is a subspace, this immediately yields $f(\mathbf{v}, \ldots, \mathbf{v}) \in \mathbf{v} + W$ as well as $f_k(W) \leq W$, which concludes the proof. □

At this point, it remains to show that the sub-automaton $\mathbb{A} = (\mathbf{v} + W, f|_{(\mathbf{v}+W)^{2r+1}})_r$ from the previous proposition is indeed isomorphic to an affine automaton (whose states must form a vector space, not an affine subspace).

Corollary 45. *Let $r \in \mathbb{N}$, $-r \leq i < j \leq r$. It holds that* $\mathrm{S}(\mathrm{AFF}^{\mathbb{F}_p}_{r;i,j}) = \mathrm{AFF}^{\mathbb{F}_p}_{r;i,j}$.

Proof. Take a $\mathbb{B} = (V, f)_r \in \mathrm{AFF}^{\mathbb{F}_p}_{r;i,j}$, and consider any $\mathbb{A} \in \mathrm{S}(\mathbb{B})$. From Proposition 44, we already know that $\mathbb{A}$ is of the form $\mathbb{A} = (\mathbf{v} + W, f|_{(\mathbf{v}+W)^{2r+1}})_r$ where $W \leq V$ is a subspace invariant under all components of f, $\mathbf{v} \in V$, and $f(\mathbf{v}, \ldots, \mathbf{v}) \in \mathbf{v} + W$. We need to construct a local algebra $\mathbb{A}'$ affine over $\mathbb{F}_p$ such that $\mathbb{A} \cong \mathbb{A}'$.

There is a $\mathbf{w} \in W$ such that $f(\mathbf{v}, \ldots, \mathbf{v}) = \mathbf{v} + \mathbf{w}$. We define $\mathbb{A}' = (W, g)_r$ where for any $\mathbf{x}_{-r}, \ldots, \mathbf{x}_r \in W$, $g(\mathbf{x}_{-r}, \ldots, \mathbf{x}_r) \coloneqq \sum_{i=-r}^{r} f_i(\mathbf{x}_i) + \mathbf{w}$. Since W is invariant under components of f, indeed $g : W^{2r+1} \to W$. Thus, $(W, g)_r$ is affine over $\mathbb{F}_p$ with linear components $g_k = f_k|_W$; in particular, g_i and g_j are bijections. Let us define $\varphi : \mathbf{v} + W \to W$ by $\varphi(\mathbf{v} + \mathbf{x}) = \mathbf{x}$ for any $\mathbf{x} \in W$. Then, it is straightforward to verify that $\mathbb{A} \cong \mathbb{A}'$ via φ. Indeed, for any $\mathbf{x}_{-r}, \ldots, \mathbf{x}_r \in W$, we have:

$$\begin{aligned}\varphi(f(\mathbf{v} + \mathbf{x}_{-r}, \ldots, \mathbf{v} + \mathbf{x}_r)) &= \varphi(f(\mathbf{v}, \ldots, \mathbf{v}) + \sum_{i=-r}^{r} f_i(\mathbf{x}_i)) \\ &= \varphi(\mathbf{v} + \mathbf{w} + \sum_{i=-r}^{r} f_i(\mathbf{x}_i)) = \mathbf{w} + \sum_{i=-r}^{r} f_i(\mathbf{x}_i) = g(\mathbf{x}_{-r}, \ldots, \mathbf{x}_r).\end{aligned}$$ □

4.4.2 Sub-automata of additive CAs

This section is not part of the proof of Theorem 36; rather, it shows that the analogy for additive automata does not hold, as a sub-automaton of an additive automaton need not be additive. Rather, it turns out that affine automata can be introduced as sub-automata of additive ones.

Proposition 46. *Let $r \in \mathbb{N}$, $-r \leq i < j \leq r$. It holds that* $\mathrm{S}(\mathrm{ADD}^{\mathbb{F}_p}_{r;i,j}) = \mathrm{AFF}^{\mathbb{F}_p}_{r;i,j}$.

Proof. Corollary 45, together with the fact that additive automata are a subclass of affine ones, yields $\mathrm{S}(\mathrm{ADD}^{\mathbb{F}_p}_{r;i,j}) \subseteq \mathrm{S}(\mathrm{AFF}^{\mathbb{F}_p}_{r;i,j}) = \mathrm{AFF}^{\mathbb{F}_p}_{r;i,j}$. We complement the results by showing that $\mathrm{AFF}^{\mathbb{F}_p}_{r;i,j} \subseteq \mathrm{S}(\mathrm{ADD}^{\mathbb{F}_p}_{r;i,j})$. Let $\mathcal{A}$ be an affine CA with local algebra $\mathbb{A} = (V, f)_r$, $\mathbb{A} \in \mathrm{AFF}^{\mathbb{F}_p}_{r;i,j}$. We construct a CA $\mathcal{B}$ with local algebra $\mathbb{B} \in \mathrm{ADD}^{\mathbb{F}_p}_{r;i,j}$ such that $\mathbb{A} \in \mathrm{S}(\mathbb{B})$. As always, write $f : V^{2r+1} \to V$ as $f(\mathbf{x}_{-r}, \ldots, \mathbf{x}_r) = f_{-r}(\mathbf{x}_{-r}) + \cdots + f_r(\mathbf{x}_r) + \mathbf{c}$ where $f_i : V \to V$ are linear mappings and $\mathbf{c} \in V$ is a constant. We put $W = V \times \mathbb{F}_p$. We fix a basis $(\mathbf{v}_1, \ldots, \mathbf{v}_{k-1})$ of V; then, $B \coloneqq (\mathbf{w}_1, \ldots, \mathbf{w}_{k-1}, \mathbf{w}_k) \coloneqq ((\mathbf{v}_1, 0), \ldots, (\mathbf{v}_{k-1}, 0), (\mathbf{0}, 1))$ is a basis of W. We define linear mappings $g_i : W \to W$ for each $-r \leq i \leq r$ on the basis B as follows:

$$\begin{aligned} g_i(\mathbf{w}_j) &\coloneqq (f_i(\mathbf{v}_j), 0) && \text{for all } -r \leq i \leq r \text{ and } 1 \leq j \leq k-1, \\ g_i(\mathbf{w}_k) &\coloneqq \begin{cases} (\mathbf{0}, 1) & \text{for } -r \leq i < 0, \\ (\mathbf{0}, -1) & \text{for } 0 < i \leq r, \\ (-\mathbf{c}, 1) & \text{for } i = 0. \end{cases} \end{aligned}$$

We put $g(\mathbf{x}_{-r},\dots,\mathbf{x}_r) = g_{-r}(\mathbf{x}_{-r}) + \dots + g_r(\mathbf{x}_r) + (\mathbf{c}, 0)$ for any $\mathbf{x}_{-r},\dots,\mathbf{x}_r \in W$ and define the automaton $\mathcal{B}$ with local algebra $\mathbb{B} = (W, g)_r$. Clearly $\mathbb{A} \cong (V \times \{0\}, g|_{(V\times\{0\})^{2r+1}})$, so indeed $\mathbb{A} \in \mathrm{S}(\mathbb{B})$. From the construction, it is clear that for each $-r \le i \le r$, g_i is bijective whenever f_i is. Furthermore, $g(\mathbf{w}_k,\dots,\mathbf{w}_k) = \underbrace{(\mathbf{0},1) + \dots + (\mathbf{0},1)}_{r\times} + (-\mathbf{c},1) + \underbrace{(\mathbf{0},-1) + \dots + (\mathbf{0},-1)}_{r\times} + (\mathbf{c},0) = (\mathbf{0},1) = \mathbf{w}_k$. Thus, g has an idempotent element, and due to Lemma 33, $\mathbb{B} \in \mathrm{ADD}^{\mathbb{F}_p}_{r;i,j}$. □

4.4.3 Quotient automata of affine CAs

In this section, we study the operator H on the class of affine automata. Again, the bijective condition is required; in Example 52 we construct an affine CA violating the bijective condition that contains a quotient automaton that is not affine.

We start by a simple observation: Every invariant subspace gives rise to a congruence and thus, to a quotient automaton. Subsequently, we show the converse: If an affine CA $\mathbb{B}$ satisfies the bijective condition, then each congruence on $\mathbb{B}$ is already a congruence of the underlying vector space. To complement the result, in Example 53 we construct an affine CA violating the bijective condition with a congruence that is not a congruence of the vector space.

Observation 47. *Let $\mathcal{B}$ be an affine CA over a finite field $\mathbb{F}$ with local algebra $\mathbb{B} = (V, f)_r$. Let $W \le V$ be a subspace invariant under all components of f. We define $\sim \subseteq V \times V$ as follows: $\mathbf{u} \sim \mathbf{v}$ if and only if $\mathbf{u} - \mathbf{v} \in W$. Then, $\sim$ is a congruence on $\mathbb{B}$.*

In several steps, we now proceed to show that under the above-mentioned conditions, every quotient automaton is of the form described by Observation 47.

Lemma 48. *Let $\mathcal{B}$ be a CA with local algebra $\mathbb{B} = (V, f)_r$ that is affine over a finite field $\mathbb{F}$ and satisfies the bijective condition. Denote the linear components of f by $f_k : V \to V$; $-r \le k \le r$. For every congruence $\sim \subseteq V \times V$ on $\mathbb{B}$ and every $\mathbf{u}, \mathbf{u}', \mathbf{v}, \mathbf{v}' \in V$ it holds that:*

1. *If $\mathbf{u} \sim \mathbf{u}'$ then $f_k(\mathbf{u}) \sim f_k(\mathbf{u}')$ for all $-r \le k \le r$.*
2. *Moreover, if f_k is a bijection for some $-r \le k \le r$, then $f_k(\mathbf{u}) \sim f_k(\mathbf{u}')$ implies $\mathbf{u} \sim \mathbf{u}'$.*
3. *If $\mathbf{u} \sim \mathbf{u}'$ and $\mathbf{v} \sim \mathbf{v}'$ then $\mathbf{u} + \mathbf{v} \sim \mathbf{u}' + \mathbf{v}'$.*
4. *If $\mathbb{F} = \mathbb{F}_p$ for some prime p, then $[\mathbf{0}]_\sim = \{\mathbf{x} \in V \mid \mathbf{x} \sim \mathbf{0}\}$ is a subspace of V invariant under all components of f.*

Proof. Throughout the proof, we write $f(\mathbf{x}_{-r},\dots,\mathbf{x}_r) = \sum_{i=-r}^{r} f_i(\mathbf{x}_i) + \mathbf{c}$ with components $f_k : V \to V$, $-r \le k \le r$ and with $\mathbf{c} \in V$. We denote the two bijective components by f_i and f_j; note that this is more general than assuming $\mathbb{B} \in \mathrm{AFF}^{\mathbb{F}}_{r;i,j}$, as we do not require f_i and f_j to be the “outer” components.

1.: Let $-r \le k \le r$ such that $k \ne i$. Since f_i is a bijection, there exists $\mathbf{b} \in V$ such that $f_i(\mathbf{b}) = -\mathbf{c}$. Then:

$$\begin{aligned} f_k(\mathbf{u}) = f_k(\mathbf{u}) - \mathbf{c} + \mathbf{c} = f(\mathbf{0},\dots,\mathbf{0},\mathbf{u},\mathbf{0},\dots,\mathbf{0},\mathbf{b},\mathbf{0},\dots,\mathbf{0}) \sim \\ \sim f(\mathbf{0},\dots,\mathbf{0},\mathbf{u}',\mathbf{0},\dots,\mathbf{0},\mathbf{b},\mathbf{0},\dots,\mathbf{0}) = f_k(\mathbf{u}') - \mathbf{c} + \mathbf{c} = f_k(\mathbf{u}'), \end{aligned}$$

where $\mathbf{u}, \mathbf{u}'$ are on the k-th position and $\mathbf{b}$ is on the i-th. The proof is analogous if $k = i$ as we can use the bijective component f_j.

2.: If f_k is bijective and $f_k(\mathbf{u}) \sim f_k(\mathbf{u}')$, we can repeatedly apply part 1. to get $f_k^n(\mathbf{u}) \sim f_k^n(\mathbf{u}')$ for every $n \in \mathbb{N}$. Since $f_k : V \to V$ is an automorphism of a finite-dimensional vector space over a finite field, there exists n such that $f_k^n = \mathrm{id}$. Thus, $\mathbf{u} = f_k^n(\mathbf{u}) \sim f_k^n(\mathbf{u}') = \mathbf{u}'$.

3.: Let $\mathbf{u} \sim \mathbf{u}'$. We first show that $\mathbf{u} - \mathbf{c} \sim \mathbf{u}' - \mathbf{c}$. There exist some $\mathbf{a}, \mathbf{a}' \in V$ such that $f_i(\mathbf{a}) = \mathbf{u}$ and $f_i(\mathbf{a}') = \mathbf{u}'$ and from *2.*, it holds that $\mathbf{a} \sim \mathbf{a}'$. There also exists a $\mathbf{b} \in V$ such that $f_j(\mathbf{b}) = -\mathbf{c} - \mathbf{c}$. Then:

$$\begin{aligned}\mathbf{u} - \mathbf{c} = f_i(\mathbf{a}) + f_j(\mathbf{b}) + \mathbf{c} = f(\mathbf{0}, \dots, \mathbf{0}, \mathbf{a}, \mathbf{0}, \dots, \mathbf{0}, \mathbf{b}, \mathbf{0}, \dots, \mathbf{0}) \sim \\ \sim f(\mathbf{0}, \dots, \mathbf{0}, \mathbf{a}', \mathbf{0}, \dots, \mathbf{0}, \mathbf{b}, \mathbf{0}, \dots, \mathbf{0}) = f_i(\mathbf{a}') + f_j(\mathbf{b}) + \mathbf{c} = \mathbf{u}' - \mathbf{c},\end{aligned}$$

where $\mathbf{a}, \mathbf{a}'$ are on the i-th position and $\mathbf{b}$ is on the j-th. Next, let $\mathbf{v} \sim \mathbf{v}'$ and let $\mathbf{y}, \mathbf{y}' \in V$ be such that $f_j(\mathbf{y}) = \mathbf{v}$ and $f_j(\mathbf{y}') = \mathbf{v}'$. Since $\mathbf{u} - \mathbf{c} \sim \mathbf{u}' - \mathbf{c}$, we find $\mathbf{x}, \mathbf{x}' \in V$ such that $f_i(\mathbf{x}) = \mathbf{u} - \mathbf{c}$, $f_i(\mathbf{x}') = \mathbf{u}' - \mathbf{c}$, and from *2.*, we again have $\mathbf{x} \sim \mathbf{x}'$. Then:

$$\begin{aligned}\mathbf{u} + \mathbf{v} = (\mathbf{u} - \mathbf{c}) + \mathbf{v} + \mathbf{c} = f_i(\mathbf{x}) + f_j(\mathbf{y}) + \mathbf{c} = f(\mathbf{0}, \dots, \mathbf{0}, \mathbf{x}, \mathbf{0}, \dots, \mathbf{0}, \mathbf{y}, \mathbf{0}, \dots, \mathbf{0}) \sim \\ \sim f(\mathbf{0}, \dots, \mathbf{0}, \mathbf{x}', \mathbf{0}, \dots, \mathbf{0}, \mathbf{y}', \mathbf{0}, \dots, \mathbf{0}) = f_i(\mathbf{x}') + f_j(\mathbf{y}') + \mathbf{c} = (\mathbf{u}' - \mathbf{c}) + \mathbf{v}' + \mathbf{c} = \mathbf{u}' + \mathbf{v}',\end{aligned}$$

where $\mathbf{x}, \mathbf{x}'$ are on the i-th position and $\mathbf{y}, \mathbf{y}'$ are on the j-th.

4.: By *3.*, the congruence $\sim$ "respects addition" in V. If $\mathbb{F} = \mathbb{F}_p$, then multiplication of vectors from V by scalars from $\mathbb{F}$ is generated by addition in V, and therefore $\sim$ also "respects multiplication by scalars". Hence, $\sim$ is not only a congruence on the algebra but also a congruence on the vector space V, and thus $[\mathbf{0}]_\sim \leq V$. By *1.*, the subspace $[\mathbf{0}]_\sim$ is invariant under all components of f. □

We now need to exploit the information about $[\mathbf{0}]_\sim$ to infer knowledge about the quotient space.

Corollary 49. *Let $r \in \mathbb{N}$ and $-r \leq i < j \leq r$. Then $\mathrm{H}(\mathrm{AFF}^{\mathbb{F}_p}_{r;i,j}) = \mathrm{AFF}^{\mathbb{F}_p}_{r;i,j}$. Moreover, $\mathrm{H}(\mathrm{ADD}^{\mathbb{F}_p}_{r;i,j}) = \mathrm{ADD}^{\mathbb{F}_p}_{r;i,j}$.*

Proof. Let $\mathcal{B}$ be an affine CA with local algebra $\mathbb{B} = (V, f)_r \in \mathrm{AFF}^{\mathbb{F}_p}_{r;i,j}$, and consider a congruence $\sim$ on $\mathbb{B}$. Let $\mathbb{A} = \mathbb{B}/\sim = (V/\sim, h)_r$ be the quotient algebra of $\mathbb{B}$. By Lemma 48, part *4.*, $\sim$ is a congruence on the vector space V and thus, $V/\sim$ is again a vector space over $\mathbb{F}_p$. Let $W = [\mathbf{0}]_\sim$. Then $V/\sim = \{\mathbf{x} + W \mid \mathbf{x} \in V\}$. For each $-r \leq l \leq r$ we define $h_l(\mathbf{x} + W) := f_l(\mathbf{x}) + W$ for any $\mathbf{x} \in V$. Thanks to Lemma 48, *1.*, $h_l : V/\sim \to V/\sim$ is a well-defined mapping. Clearly, it is a linear mapping on $V/\sim$. By definition, $h(\mathbf{x}_{-r} + W, \dots, \mathbf{x}_r + W) = f(\mathbf{x}_{-r}, \dots, \mathbf{x}_r) + W = \sum_{l=-r}^{r} f_l(\mathbf{x}_l) + \mathbf{c} + W$. This is further equal to $\sum_{l=-r}^{r} (f_l(\mathbf{x}_l) + W) + (\mathbf{c} + W) = \sum_{l=-r}^{r} h_l(\mathbf{x}_l + W) + (\mathbf{c} + W)$. This shows, as expected, that h is indeed an affine mapping with components h_l, $-r \leq l \leq r$. Clearly, if f_l is a bijection for some $-r \leq l \leq r$, then h_l is surjective and therefore also a bijection. Thus, $\mathbb{A} = \mathbb{B}/\sim = (V/\sim, h)_r$ is a CA affine over $\mathbb{F}_p$ whose local rule has its i-th and j-th component bijective. This finishes the proof in the affine case.

To conclude the second part of the statement: If $\mathbb{B} \in \mathrm{ADD}^{\mathbb{F}_p}_{r;i,j}$, then $\mathbf{0} \in V$ is an idempotent of f, so it is easy to see that $[\mathbf{0}]_\sim$ is an idempotent for $\mathbb{A} = \mathbb{B}/\sim$. Hence $\mathbb{A} \in \mathrm{ADD}^{\mathbb{F}_p}_{r;i,j}$. □

4.4.4 Main Result and Examples

Combining Corollaries 40, 42, 45, 49, and Proposition 46 yields the following main result. Note that the announced Theorem 36 is just another formulation of (4.5).

Theorem 50. *For any $r \in \mathbb{N}$, $-r \leq i < j \leq r$, and any prime p it holds that*

$$\mathrm{HSP}_{\mathrm{fin}}\Xi(\mathrm{AFF}^{\mathbb{F}_p}_{r;i,j}) = \mathrm{AFF}^{\mathbb{F}_p}_{r;i,j}, \tag{4.5}$$

$$\mathrm{HSP}_{\mathrm{fin}}\Xi(\mathrm{ADD}^{\mathbb{F}_p}_{r;i,j}) = \mathrm{AFF}^{\mathbb{F}_p}_{r;i,j}. \tag{4.6}$$

When analysing the sub-automata and quotient automata of affine CAs, we relied on the assumption that the CA has its two components bijective. We now give two simple counterexamples of affine CAs violating this condition that contain a sub-automaton or a quotient automaton that is not affine.

Example 51. Let V be a finite-dimensional vector space over $\mathbb{F}_2$ and let $\mathcal{B}$ be an additive CA over $\mathbb{F}_2$ with local algebra $\mathbb{B} = (V, f)_r$ where f has components $f_i : V \to V$, $-r \leq i \leq r$. Suppose that there exist vectors $\mathbf{v}_1 \neq \mathbf{v}_2 \in V$ such that $f_i(\mathbf{v}_1) = f_i(\mathbf{v}_2) = \mathbf{0}$ for each $-r \leq i \leq r$. Let $S = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{0}\}$ and $\mathbb{A} = (S, f|_{S^{2r+1}})$. Then $\mathbb{A} \in \mathrm{S}(\mathbb{B})$, but clearly $\mathbb{A}$ is not affine over $\mathbb{F}_2$ because S cannot be isomorphic to any vector space over $\mathbb{F}_2$.

Example 52. Let V be a finite-dimensional vector space over $\mathbb{F}_2$ and let $\mathcal{B}$ be an additive CA over $\mathbb{F}_2$ with local algebra $\mathbb{B} = (V, f)_r$; $|V| = 2^k > 2$. Suppose that $\mathrm{Im}\,(f) \neq V$; $|\mathrm{Im}\,(f)| = 2^l \geq 2$. We define $\sim \subseteq V \times V$ as follows: $\mathbf{u} \sim \mathbf{u}$ for each $\mathbf{u} \in V$ and $\mathbf{u} \sim \mathbf{v}$ if and only if $\mathbf{u}, \mathbf{v} \in \mathrm{Im}\,(f)$. It is straightforward to verify that $\sim$ is a congruence on $\mathbb{B}$ that partitions V into the following equivalence classes: $\mathrm{Im}\,(f)$ and $2^k - 2^l$ classes each containing a single element. Thus, $2 \nmid |\mathbb{B}/\sim| = 2^k - 2^l + 1$ and therefore, $\mathbb{B}/\sim$ is not isomorphic to any vector space over $\mathbb{F}_2$.

Example 53. Let V be a finite-dimensional vector space over $\mathbb{F}_2$ and let $\mathcal{B}$ be an additive CA over $\mathbb{F}_2$ with local algebra $\mathbb{B} = (V, f)_r$ where f has components $f_i : V \to V$, $-r \leq i \leq r$. Suppose that the linear map $f_0 \neq \mathrm{id}$ is a bijection and all the other components are constant zero mappings. We define a relation $\sim \subseteq V \times V$ such that $\mathbf{u} \sim \mathbf{v}$ if and only if there exists some $k \geq 0$ such that $f_0^k(\mathbf{u}) = \mathbf{v}$. Since f_0 is bijective on a finite set V, $\sim$ is symmetrical and therefore an equivalence. Further, it is straightforward to verify that $\sim$ is a congruence on $\mathbb{B}$. Then $[\mathbf{0}]_\sim = \{\mathbf{0}\}$, but there exists a $\mathbf{v} \in V$ such that $|[\mathbf{v}]_\sim| > 1$. Thus, $\sim$ is not a congruence of the vector space V.

Example 54 (ECA 60)**.** Let us consider the class of elementary CAs; i.e., CAs with states $\mathbb{F}_2$ and radius $r = 1$. We consider elementary CA 60 that is defined as $\mathrm{ECA}_{60} = (\mathbb{F}_2, f)_1$ where $f(x, y, z) = x + y \bmod 2$ and ECA 195 defined as $\mathrm{ECA}_{195} = (\mathbb{F}_2, g)_1$ where $g(x, y, z) = x + y + 1 \bmod 2$. Clearly, $\mathrm{ECA}_{195} \cong \mathrm{ECA}_{60}$ via the bijection that exchanges 0 and 1. Further, $\mathrm{ECA}_{60}, \mathrm{ECA}_{195} \in \mathrm{AFF}^{\mathbb{F}_2}_{1;-1,0}$ and they are the only two elementary CAs that belong to this class. Thus, Theorem 50 implies that the only elementary CA that can be simulated by ECA 60 is itself (up to isomorphism).

Canonical Affine Cellular Automata Canonical affine CAs form the most studied subclass of affine automata. For them, analysing $\mathrm{HSP}_{\mathrm{fin}}\Xi(\mathbb{A})$ becomes simple again. Suppose $\mathcal{A}$ is affine over $\mathbb{F}_p$ with local algebra $\mathbb{A} = (\mathbb{F}_p, f)_r$ and has its local rule of the form $f(x_{-r}, \ldots, x_r) = a_{-r}x_{-r} + \cdots + a_r x_r + c$ for some coefficients $a_{-r}, \ldots, a_r \in \mathbb{F}_p$ and $c \in \mathbb{F}_p$. Then, we can distinguish the following cases:

1. $a_i = 0$ for all $-r \leq i \leq r$. In such a case, the CA is a constant zero mapping and studying $\mathrm{HSP}_{\mathrm{fin}}\Xi(\mathbb{A})$ is trivial.

2. $a_i \neq 0$ for some $-r \leq i \leq r$ and $a_j = 0$ for all $j \neq i$. Then, the CA is essentially a “shift operator” and again, studying $\mathrm{HSP}_{\mathrm{fin}}\Xi(\mathbb{A})$ is simple.

3. There are two non-zero coefficients a_i and a_j, $i < j$ (we can take i to be the smallest such coefficient and j the largest one). Hence, $\mathbb{A} \in \mathrm{AFF}^{\mathbb{F}}_{r;i,j}$ and the results of this section apply.

4.5 Concluding Remarks

We stress that an important part of the merit of this paper lies in formalizing the notion of CA simulation into algebraic language. This makes it possible to see new connections to well established fields of abstract algebra. Whereas the proofs provided in this paper do not rely on any sophisticated algebraic concepts, we remark that, as an example, Lemma 48 and Corollary 49 are a direct consequence of a deeper theorem by Smith [138] and Gumm [55] about Abelian algebras with a Maltsev term.

We believe that the connection with abstract algebra can provide powerful tools for deriving a plethora of both negative and positive results regarding the simulation capacity of various CA classes in the future.

5. Simulation Capacity of Canonical Additive Automata

In the previous chapter, we have shown that the class $\mathrm{AFF}^{\mathbb{F}_p}_{r;i,j}$ is closed under all the operators $\mathrm{H}, \mathrm{S}, \mathrm{P}_{\mathrm{fin}}, \Xi$ for any prime p, any radius r, and any $-r \leq i < j \leq r$. As a consequence, automata with "bijective outer components" that are affine over different prime fields $\mathbb{F}_p$ are incomparable with respect to the simulation relation $\preceq$. However, this gives us no information about the relation of two particular CAs within the same class $\mathrm{AFF}^{\mathbb{F}_p}_{r;i,j}$. In this chapter, we complement the previous results by analysing the simulation capacity of individual additive automata. Specifically, we will analyse $\mathrm{HSP}_{\mathrm{fin}}\Xi(\mathbb{A})$ for the local algebra $\mathbb{A}$ of any canonical additive CA satisfying certain conditions. As a special case, this will allow us to describe the simulation capacities of any canonical additive CA with radius $r = 1$ (over an arbitrary field $\mathbb{F}_p$).

Recall that canonical additive CAs have a local algebra of the form $(\mathbb{F}_p, f)_r$ for some prime $p \in \mathbb{P}$. Thus, the local map $f : \mathbb{F}_p^{2r+1} \to \mathbb{F}_p$ is defined as $f(x_{-r}, \dots, x_r) = a_{-r}x_{-r} + \dots + a_r x_r$ for some $a_{-r}, \dots, a_r \in \mathbb{F}_p$. We simply say that f is *given* by $(a_{-r}, \dots, a_r)$; and we call $a_{-r}, \dots, a_r$ its *coefficients*. We note that canonical CAs having at most one non-zero coefficient are trivial to analyse. Thus, in this chapter, we focus on studying CAs with at least two non-zero coefficients. Such CAs satisfy the bijective condition (see Definition 35), which allows us to use some results from the previous chapter.

Canonical additive CAs represent a highly specific subset within the broader category of additive automata. Nevertheless, they remain a focal point in the existing literature, underscoring their significance.

Now we define a stronger version of the useful bijective condition. Most of the results in this chapter are obtained assuming this property.

Definition 55. Let $\mathbb{B} = (\mathbb{F}_p, f)_r$ be a CA local algebra such that $\mathbb{B} \in \mathrm{ADD}^{\mathbb{F}_p}_{r;i,j}$ for some $-r \leq i < j \leq r$. Let $f : \mathbb{F}_p^{2r+1} \to \mathbb{F}_p$ be given by $(a_{-r}, \dots, a_r)$. We say that f (or $\mathbb{B}$) is *doubly bijective* if $a_{i+1} \neq 0$ and $a_{j-1} \neq 0$.

Example 56. Let $\mathbb{A} = (\mathbb{F}_p, f)_1$ such that $f(x, y, z) = x + y + z$ and let $\mathbb{B} = (\mathbb{F}_p, g)_1$ be such that $g(x, y, z) = x + z$. Then $\mathbb{A}$ is doubly bijective but $\mathbb{B}$ is not.

5.1 Results Summary

We summarize four auxiliary results that we prove throughout this chapter.

Proposition 57. *For any local algebra $\mathbb{B} = (\mathbb{F}_p, f)_r$ of a canonical additive CA it holds that:*

(1) $\mathbb{B}^{[p^k]} \cong \underbrace{\mathbb{B} \times \dots \times \mathbb{B}}_{p^k \times}$ *for any $k \in \mathbb{N}$.*

(2) More generally, $\mathbb{B}^{[p^k \cdot l]} \cong \underbrace{\mathbb{B}^{[l]} \times \dots \times \mathbb{B}^{[l]}}_{p^k \times}$ *for any $k, l \in \mathbb{N}$.*

(3) If $\mathbb{B}$ is doubly bijective then for $l \in \mathbb{N}$ such that $p \nmid l$, the algebra $\mathbb{B}^{[l]}$ has only trivial subalgebras and quotient algebras.

(4) If $\mathbb{B} = (\mathbb{F}_p, f)_1$ is given by $(a, 0, b)$, $a, b \neq 0$, then for $l \in \mathbb{N}$ odd such that $p \nmid l$, the algebra $\mathbb{B}^{[l]}$ has only trivial subalgebras and quotient algebras.

Proof. This is proven as Corollary 63, Corollary 64, Proposition 81 and Proposition 84, respectively. □

After some work (see Section 5.3) this yields the following two main results of this chapter:

Theorem 58. *Let* $\mathbb{B} = (\mathbb{F}_p, f)_r$ *be a doubly bijective local algebra of a canonical additive CA. Then:*

$$\mathrm{HSP}_{\mathrm{fin}}\Xi(\mathbb{B}) = \mathrm{P}_{\mathrm{fin}}\Xi(\mathbb{B}) \cup \mathrm{SING}_r.$$

In plain words, if $\mathbb{A}$ is simulated by $\mathbb{B}$ then $\mathbb{A}$ is isomorphic to a finite product of some iterative powers of $\mathbb{B}$. Moreover, as we will see, the exponents in the iterative powers can all be chosen to be coprime with p.

Theorem 59. *Let* $\mathbb{B} = (\mathbb{F}_p, f)_1$ *be a local algebra of a canonical additive CA with* $f(x, y, z) = ax + bz$*;* $a, b \neq 0$*. Let* $\mathbb{B}' = (\mathbb{F}_p, f')_1$ *where* $f'(x, y, z) = a^2x + 2aby + b^2z$*. Then:*

$$\mathrm{HSP}_{\mathrm{fin}}\Xi(\mathbb{B}) = \mathrm{P}_{\mathrm{fin}}\Xi(\{\mathbb{B}, \mathbb{B}'\}) \cup \mathrm{SING}_r.$$

The two theorems above already describe the computational capacity of any canonical additive CA with radius $r = 1$ that is non-trivial in the sense that at least two of its components are non-zero. As a special case, we obtain the following corollary.

Corollary 60. *Let* p *be a prime and consider the class* CA_p *of all CAs with* p *states and radius* $r = 1$*. Let* $\mathbb{B} = (\mathbb{F}_p, f)_1$ *be the local algebra of a canonical additive CA given by* (a_{-1}, a_0, a_1) *with at least two non-zero coefficients. Then, one of the following cases holds:*

(1) $a_0 \neq 0$*. Then* $\mathbb{B}$ *is doubly bijective and within the class* CA_p*,* $\mathbb{B}$ *can only simulate itself (up to isomorphism).*

(2) $a_0 = 0$*. Then, within the class of* CA_p*,* $\mathbb{B}$ *can simulate (up to isomorphism) exactly itself and the automaton with local algebra* $\mathbb{B}' = (\mathbb{F}_p, f')_1$ *given by* $(a_{-1}^2, 2a_{-1}a_1, a_1^2)$*. Note that* $\mathbb{B} \cong \mathbb{B}'$ *if and only if* $p = 2$*.*

The corollary implies new results about simulation limitations even for the very well-studied case of elementary CAs. We describe one such example below.

Example 61. We consider elementary CA 90 that is defined as $\mathrm{ECA}_{90} = (\mathbb{F}_2, f)_1$ where $f(x, y, z) = x + z \bmod 2$ and ECA 165 defined as $\mathrm{ECA}_{165} = (\mathbb{F}_2, g)_1$ where $g(x, y, z) = x + z + 1 \bmod 2$. Clearly, $\mathrm{ECA}_{90} \cong \mathrm{ECA}_{165}$ via the mapping that exchanges 0 and 1. Then, Corollary 60 implies that within the class of elementary CAs, ECA 90 can only simulate itself and ECA 165.

Proof. Clearly, ECA 90 satisfies the assumptions of Corollary 60 (2). In the case of a CA additive over $\mathbb{F}_2$, the local algebras $\mathbb{B}$ and $\mathbb{B}'$ coincide (using the notation in the corollary). Thus, the corollary yields that ECA 90 can only simulate elementary CAs whose local algebras are isomorphic to ECA_{90}. In this case, we have only one bijection on $\mathbb{F}_2$ that is not identity, and this yields the ECA 165. □

5.2 Iterative Powers of Canonical Additive CAs Split into Products

In this section, we prove the parts (1) and (2) of Proposition 57.

Lemma 62 (Martin, Odlyzko, Wolfram [96])**.** *Let* $\mathbb{B} \in \mathrm{ADD}_r^{\mathbb{F}_p}$ *be a local algebra of a canonical additive CA;* $\mathbb{B} = (\mathbb{F}_p, f)_r$ *where* f *is given by* $(a_{-r}, \ldots, a_r)$*. Then, for each* $k \in \mathbb{N}$*, it holds that* $f^{[p^k]}(\mathbf{x}_{-r}, \ldots, \mathbf{x}_r) = a_{-r}\,\mathbf{x}_{-r} + \cdots + a_r\,\mathbf{x}_r$ *for any* $\mathbf{x}_{-r}, \ldots, \mathbf{x}_r \in \mathbb{F}_p^{p^k}$*.*

Proof. The reader can find the detailed proof in the seminal paper on additive automata [96]. We only outline it here. Each configuration $c \in \mathbb{F}_p^{\mathbb{Z}}$ can be associated with a generating function $c(x) = \Sigma_{i=-\infty}^{\infty} c_i x^i$. Let $l(x) = a_r x^{-r} + a_{r-1} x^{-r+1} + \cdots + a_{-r} x^r$. Then, for the CA's global rule F it is easy to verify that the generating sequence for $F(c)$ corresponds to $l(x) \cdot c(x)$ where the coefficients of the polynomials are computed over $\mathbb{F}_p$. Then, $F^n(c)$ corresponds to $l(x)^n \cdot c(x)$ for each $n \in \mathbb{N}$. Since the Frobenius endomorphism $x \mapsto x^{p^k}$ acts as the identity on $\mathbb{F}_p$, we have: $l(x)^{p^k} = a_r^{p^k} x^{-rp^k} + a_{r-1}^{p^k} x^{(-r+1)p^k} + \cdots + a_{-r}^{p^k} x^{rp^k} = a_r x^{-rp^k} + a_{r-1} x^{(-r+1)p^k} + \cdots + a_{-r} x^{rp^k}$. This already implies $f^{[p^k]}(\mathbf{x}_{-r}, \ldots, \mathbf{x}_r) = a_{-r}\,\mathbf{x}_{-r} + \cdots + a_r\,\mathbf{x}_r$; $\mathbf{x}_{-r}, \ldots, \mathbf{x}_r \in \mathbb{F}_p^{p^k}$. □

Corollary 63. *Let $\mathbb{B} \in \mathrm{ADD}_r^{\mathbb{F}_p}$ be a local algebra of a canonical additive CA and let $k \in \mathbb{N}$. Then,*
$\mathbb{B}^{[p^k]} \cong \underbrace{\mathbb{B} \times \cdots \times \mathbb{B}}_{p^k \times}$.

Corollary 64. *Let $\mathbb{B} \in \mathrm{ADD}_r^{\mathbb{F}_p}$ be a local algebra of a canonical additive CA and let $k, l \in \mathbb{N}$. Then, $\mathbb{B}^{[p^k \cdot l]} \cong \underbrace{\mathbb{B}^{[l]} \times \cdots \times \mathbb{B}^{[l]}}_{p^k \times}$.*

Proof. We have:

$$\begin{aligned}
\mathbb{B}^{[p^k \cdot l]} &\cong \big(\mathbb{B}^{[p^k]}\big)^{[l]} && \text{by Lemma 26,}\\
&\cong (\underbrace{\mathbb{B} \times \cdots \times \mathbb{B}}_{p^k \times})^{[l]} && \text{by Corollary 63,}\\
&\cong \underbrace{\mathbb{B}^{[l]} \times \cdots \times \mathbb{B}^{[l]}}_{p^k \times} && \text{by Lemma 27.}
\end{aligned}$$

This concludes the proof. □

5.3 Characterizing the Simulation Capacity of Canonical Additive CAs

In this section, we prove Theorems 58 and 59 – the main results of this chapter. However, we postpone the proofs of Propositions 81 and 84 until the next section. We first recall that both sub-automata and quotient automata of additive CAs satisfying the bijective condition are determined by subspaces invariant under all the CA's local rule components. This is summarized in the following proposition.

Proposition 65. *Let $\mathbb{B} = (V, f)_r$ be a local algebra of an additive CA satisfying the bijective condition. Then:*

(1) Let $U \subseteq V$ and $\mathbb{A} = (U, f|_{U^{2r+1}})_r$. Then, $\mathbb{A}$ is a subalgebra of $\mathbb{B}$ if and only if $U = \mathbf{x} + W$ where $W \leq V$ is a subspace invariant under all components of f and $\mathbf{x} \in V$ satisfies $f(\mathbf{x}, \ldots, \mathbf{x}) \in \mathbf{x} + W$.

(2) Let $\sim \subseteq V \times V$ be a congruence on $\mathbb{B}$. Then, there exists a subspace $W \leq V$ invariant under all components of f such that for each $\mathbf{x}, \mathbf{y} \in V$ we have: $\mathbf{x} \sim \mathbf{y}$ if and only if $\mathbf{x} - \mathbf{y} \in W$. In fact, $W = [\mathbf{0}]_\sim$. (Conversely, each such invariant subspace defines a congruence on $\mathbb{B}$ in this way.)

Proof. Part (1) is essentially Lemma 44 from the previous chapter. Part (2) is implied by Observation 47 and Lemma 48 from the previous chapter. □

The relationship between a CA's sub-automata (or quotient automata) and its invariant subspaces motivates the following definition.

Definition 66. Let $\mathbb{B} = (V, f)_r$ be a local algebra of an additive CA. We say that $\mathbb{B}$ is *simple* if the only subspaces of V invariant under all components of f are the *trivial* subspaces $\{\mathbf{0}\}$ and V.

Note that thanks to Proposition 65, for an additive automaton $\mathcal{B}$ satisfying the bijective condition, the following three properties are equivalent:

- $\mathcal{B}$ is simple.
- $\mathcal{B}$ has no non-trivial sub-automata.
- $\mathcal{B}$ has no non-trivial quotient automata.

We have already shown that for a local algebra $\mathbb{B}$ that is canonical additive over $\mathbb{F}_p$, each iterative power $\mathbb{B}^{[p^k \cdot l]}$ splits as $\mathbb{B}^{[l]} \times \cdots \times \mathbb{B}^{[l]}$. In order to understand the sub-automata and quotient automata of $\mathbb{B}^{[p^k \cdot l]}$, we need to understand the invariant subspaces of $\mathbb{B}^{[l]}$, $p \nmid l$, and subsequently, the invariant subspaces of products of $\mathbb{B}^{[l]}$. In the next section, we will show that if $\mathbb{B}$ is doubly bijective then $\mathbb{B}^{[l]}$ is simple for each $p \nmid l$, as stated in Proposition 81. Now, we characterize all invariant subspaces of products of simple local algebras and subsequently, we combine this with Proposition 81 to prove Theorems 58 and 59.

Lemma 67. *Let $\mathbb{B}_1 = (U, f)_r$ and $\mathbb{B}_2 = (V, g)_r$ be local algebras of CAs additive over $\mathbb{F}_p$ that are both simple. Let $\mathbb{B} = \mathbb{B}_1 \times \mathbb{B}_2 = (U \times V, h)_r$ and let $W \leq U \times V$ be a non-trivial subspace invariant under all components of h. Then, either $W = \{\mathbf{0}\} \times V$, $W = U \times \{\mathbf{0}\}$, or $W = \{(\mathbf{u}, \psi(\mathbf{u})) \mid \mathbf{u} \in U\}$ where $\psi : U \to V$ is an isomorphism of vector spaces which satisfies $\psi \circ f_i = g_i \circ \psi$ for all $-r \leq i \leq r$.*

Proof. We distinguish the following cases:

a) There exists $\mathbf{u} \in U$, $\mathbf{u} \neq \mathbf{0}$ such that $(\mathbf{u}, \mathbf{0}) \in W$. Since $\mathbb{B}_1$ is simple we get $U \times \{\mathbf{0}\} \subseteq W$. Subsequently, since $\mathbb{B}_2$ is simple also, this implies either $W = U \times \{\mathbf{0}\}$ or $W = U \times V$.

b) There exists $\mathbf{v} \in V$, $\mathbf{v} \neq \mathbf{0}$ such that $(\mathbf{0}, \mathbf{v}) \in W$. This is analogous to case a).

c) For each $\mathbf{u} \in U$ there exists at most one $\mathbf{v} \in V$ such that $(\mathbf{u}, \mathbf{v}) \in W$; otherwise case b) would apply. We define $\pi_U : U \times V \to U$; $\pi((\mathbf{u}, \mathbf{v})) = \mathbf{u}$ and analogously, we define π_V. Clearly, $\{\mathbf{0}\} \neq \pi_U(W) \leq U$ is a subspace invariant under all components of f and thus, $\pi_U(W) = U$. Similarly, $\pi_V(W) = V$. Thus, for each $\mathbf{u} \in U$ there exists exactly one $\mathbf{v} \in V$ such that $(\mathbf{u}, \mathbf{v}) \in W$; we put $\psi(\mathbf{u}) = \mathbf{v}$. Hence, we get a bijective mapping $\psi : U \to V$. Clearly, $(\mathbf{u}, \mathbf{v}) \in W$ and $(\mathbf{u}', \mathbf{v}') \in W$ gives $(\mathbf{u} + \mathbf{u}', \mathbf{v} + \mathbf{v}') \in W$ and thus, $\psi(\mathbf{u} + \mathbf{u}') = \psi(\mathbf{u}) + \psi(\mathbf{u}')$. Similarly, since W is invariant under all components of h, we get that if $(\mathbf{u}, \mathbf{v}) \in W$ then $(f_i(\mathbf{u}), g_i(\mathbf{v})) \in W$ for all $-r \leq i \leq r$. Thus, $\psi(f_i(\mathbf{u})) = g_i(\psi(\mathbf{u}))$ for all $\mathbf{u} \in U$ and all $-r \leq i \leq r$. Hence, $\psi : U \to V$ is an isomorphism of vector spaces which satisfies $\psi \circ f_i = g_i \circ \psi$ for all $-r \leq i \leq r$. □

Note that in the previous lemma, W of the last form can obviously only occur if $\mathbb{B}_1 \cong \mathbb{B}_2$.

Proposition 68. *Let $\mathbb{B}_1, \mathbb{B}_2 \in \mathrm{ADD}^{\mathbb{F}_p}_{r;i,j}$ for some $-r \leq i < j \leq r$. Suppose that both $\mathbb{B}_1$ and $\mathbb{B}_2$ are simple. Let $\mathbb{A}$ be one of the following:*

(1) a subalgebra of $\mathbb{B}_1 \times \mathbb{B}_2$,

(2) a quotient algebra of $\mathbb{B}_1 \times \mathbb{B}_2$.

Then, $\mathbb{A} \in \mathrm{SING}_r$, $\mathbb{A} \cong \mathbb{B}_1$, $\mathbb{A} \cong \mathbb{B}_2$ or $\mathbb{A} \cong \mathbb{B}_1 \times \mathbb{B}_2$.

Proof. We denote $\mathbb{B}_1 = (U, f)_r$, $\mathbb{B}_2 = (V, g)_r$ and $\mathbb{B}_1 \times \mathbb{B}_2 = (U \times V, h)_r$. Clearly, $\mathbb{B}_1 \times \mathbb{B}_2 \in \mathrm{ADD}^{\mathbb{F}_p}_{r;i,j}$. We study the two cases separately.

(1) Proposition 65 (1) yields $\mathbb{A} = \left(\mathbf{x} + W, h|_{(\mathbf{x}+W)^{2r+1}}\right)$ for some subspace $W \leq U \times V$ invariant under all components of h and for some $\mathbf{x} \in U \times V$ satisfying $h(\mathbf{x}, \dots, \mathbf{x}) \in \mathbf{x} + W$. In the light of Lemma 67, we distinguish the following cases. If $W = \{(\mathbf{0}, \mathbf{0})\}$ then clearly $\mathbb{A} \in \mathrm{SING}_r$. If $W = U \times V$ then clearly $\mathbb{A} = \mathbb{B}_1 \times \mathbb{B}_2$.

We consider the case when $W = \{(\mathbf{u}, \psi(\mathbf{u})) \mid \mathbf{u} \in U\}$ where $\psi : U \to V$ is an isomorphism of vector spaces, or $W = U \times \{\mathbf{0}\}$. Then, clearly $\pi_U : \mathbb{A} \to \mathbb{B}_1$ defined as $\pi_U((\mathbf{u}, \mathbf{v})) = \mathbf{u}$ is a surjective homomorphism of the algebras, and since the cardinalities match, it is an isomorphism. (Note that we did not need the explicit knowledge that the underlying set of $\mathbb{A}$ has the form $\mathbf{x} + W$; for this proof, it sufficed to know its cardinality and the fact that π_U is surjective.)

The case when $W = \{\mathbf{0}\} \times V$ is analogous to the previous one.

(2) We put $\mathbb{B} = \mathbb{B}_1 \times \mathbb{B}_2$. Proposition 65 (2) gives us that $W := [\mathbf{0}]_\sim \leq U \times V$ is a subspace invariant under all components of h. Thanks to Lemma 67 it is enough to distinguish the following cases. For $W = \{(\mathbf{0}, \mathbf{0})\}$ it holds that $\mathbb{B}/_\sim \cong \mathbb{B}$. Similarly, $W = U \times V$ implies $\mathbb{B}/_\sim \in \mathrm{SING}_r$. $W = U \times \{\mathbf{0}\}$ implies $\mathbb{B}/_\sim \cong \mathbb{B}_2$. Similarly, $W = \{\mathbf{0}\} \times V$ implies $\mathbb{B}/_\sim \cong \mathbb{B}_1$.

We are left with the case when $W = \{(\mathbf{u}, \psi(\mathbf{u})) \mid \mathbf{u} \in U\}$ where $\psi : U \to V$ is an isomorphism of vector spaces. We define the mapping $\varphi : \mathbb{B}/_\sim \to \mathbb{B}_2$ as follows: $\varphi([(\mathbf{u}, \mathbf{v})]_\sim) := \mathbf{v} - \psi(\mathbf{u})$. We check φ is well-defined and injective: Let $(\mathbf{u}, \mathbf{v}), (\mathbf{u}', \mathbf{v}') \in U \times V$. Then:

$$\begin{aligned}
(\mathbf{u}, \mathbf{v}) \sim (\mathbf{u}', \mathbf{v}') &\iff (\mathbf{u} - \mathbf{u}', \mathbf{v} - \mathbf{v}') \in W \\
&\iff \mathbf{v} - \mathbf{v}' = \psi(\mathbf{u} - \mathbf{u}') \\
&\iff \mathbf{v} - \psi(\mathbf{u}) = \mathbf{v}' - \psi(\mathbf{u}') \\
&\iff \varphi([(\mathbf{u}, \mathbf{v})]_\sim) = \varphi([(\mathbf{u}', \mathbf{v}')]_\sim).
\end{aligned}$$

It is straightforward to see that φ is surjective and thus an isomorphism of the algebras. Therefore, $\mathbb{B}/_\sim \cong \mathbb{B}_2$. □

The previous proposition only analyses the product of two simple local algebras of CAs. However, the results can be generalized in a quite straightforward way. We present the general version below.

Lemma 69. *Let $k \in \mathbb{N}$ and let $\mathbb{B}_i = (V_i, f_i)_r$ be a simple local algebra of a CA additive over $\mathbb{F}_p$ for all $1 \leq i \leq k$. Let $\mathbb{B}_1 \times \cdots \times \mathbb{B}_k = (V_1 \times \cdots \times V_k, h)_r$. Let $W \leq V_1 \times \cdots \times V_k$ be a subspace invariant under all components of h. Then, there exists a partition of indices $I_1, \dots, I_l \subseteq \{1, \dots, k\}$, $\dot\bigcup_{j=1}^{l} I_j = \{1, \dots, k\}$; $I_j = \{i_{j,1}, \dots, i_{j,k_j}\}$ with the following property: For each $j \in \{1, \dots, l\}$ there exist isomorphisms $\psi^j_2 : V_{i_{j,1}} \to V_{i_{j,2}}, \dots, \psi^j_{k_j} : V_{i_{j,1}} \to V_{i_{j,k_j}}$ which commute with the corresponding local rule components such that (after suitable reordering of the factors) $W = W_1 \times \cdots \times W_l$ where for each $j \in \{1, \dots, l\}$ we put $W_j = \{(\mathbf{v}, \psi^j_2(\mathbf{v}), \dots, \psi^j_{k_j}(\mathbf{v})) \mid \mathbf{v} \in V_{i_{j,1}}\}$.*

Proof. This is a slightly technical, yet straightforward generalization of Lemma 67. □

Proposition 70. *Let $\mathbb{B}_1, \dots, \mathbb{B}_k \in \mathrm{ADD}^{\mathbb{F}_p}_{r;i,j}$ for some $-r \leq i < j \leq r$, all of them simple. Let $\mathbb{A}$ be one of the following:*

(1) a subalgebra of $\mathbb{B}_1 \times \cdots \times \mathbb{B}_k$,

(2) a quotient algebra of $\mathbb{B}_1 \times \cdots \times \mathbb{B}_k$.

Then, either $\mathbb{A} \in \mathrm{SING}_r$ or there exist indices $\{i_1, \dots i_l\} \subseteq \{1, \dots, k\}$ such that $\mathbb{A} \cong \mathbb{B}_{i_1} \times \cdots \times \mathbb{B}_{i_l}$.

Proof. This is a generalization of Proposition 68 using Lemma 69. □

Now we are ready to prove one of the main results of this chapter that considers the simulation limitations of any doubly bijective canonical additive CA.

Theorem 58. *Let $\mathbb{B} = (\mathbb{F}_p, f)_r$ be a doubly bijective local algebra of a canonical additive CA. Then:*

$$\mathrm{HSP}_{\mathrm{fin}}\Xi(\mathbb{B}) = \mathrm{P}_{\mathrm{fin}}\Xi(\mathbb{B}) \cup \mathrm{SING}_r.$$

Proof. Since $\mathbb{B}$ is doubly bijective, $\mathbb{B} \in \mathrm{ADD}^{\mathbb{F}_p}_{r;i,j}$ for some $-r \leq i < j \leq r$. Let $\mathbb{A} \in \mathrm{P}_{\mathrm{fin}}\Xi(\mathbb{B})$. Corollary 64 gives us that

$$\mathbb{A} \cong \mathbb{B}^{[l_1]} \times \cdots \times \mathbb{B}^{[l_m]}$$

where the exponents $l_1, \ldots, l_m$ are coprime with p (and not necessarily pairwise distinct). Proposition 81 that we prove in the next section gives us that each algebra $\mathbb{B}^{[l_i]}$, $1 \leq i \leq m$, is simple. Further, Corollary 40 yields that $\mathbb{B}^{[n]} \in \mathrm{ADD}^{\mathbb{F}_p}_{r;i,j}$ for each $n \in \mathbb{N}$. Thus, all the assumptions for Proposition 70 are met. Hence, Proposition 70 (1) gives that if $\mathbb{A}' \in \mathrm{S}(\mathbb{A})$ then $\mathbb{A}' \in \mathrm{P}_{\mathrm{fin}}\Xi(\mathbb{B})$ or $\mathbb{A}' \in \mathrm{SING}_r$. Similarly, Proposition 70 (2) gives that if $\mathbb{A}'' \in \mathrm{H}(\mathbb{A}')$ then $\mathbb{A}'' \in \mathrm{P}_{\mathrm{fin}}\Xi(\mathbb{B})$ or $\mathbb{A}'' \in \mathrm{SING}_r$. Thus, $\mathrm{HSP}_{\mathrm{fin}}\Xi(\mathbb{B}) = \mathrm{P}_{\mathrm{fin}}\Xi(\mathbb{B}) \cup \mathrm{SING}_r$. □

This concludes the characterization of the simulation capacity of any canonical additive CA that is doubly bijective. Let us now focus on the canonical additive CAs with radius $r = 1$. Let $\mathbb{B} = (\mathbb{F}_p, f)_1$ be given by $(a_{-1}, a_0, a_1) \in \mathbb{F}_p^3$. As already mentioned, if at most one of the coefficients is non-zero, analysing $\mathrm{HSP}_{\mathrm{fin}}\Xi(\mathbb{B})$ is straightforward. If at least two of the coefficients in (a_{-1}, a_0, a_1) are non-zero, we distinguish the following two cases:

1. $a_0 \neq 0$. Then, $\mathbb{B}$ is doubly bijective and Theorem 58 applies.

2. $a_0 = 0$. This is the case we analyse below.

Lemma 71. *Let $\mathbb{B} = (\mathbb{F}_p, f)_1$ be a local algebra with $f(x, y, z) = ax + bz$. Let $\mathbb{B}' = (\mathbb{F}_p, f')_1$ where $f'(x, y, z) = a^2x + 2aby + b^2z$. Then, $\mathbb{B}^{[2]} \cong \mathbb{B}' \times \mathbb{B}'$.*

Proof. Let $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{F}_p^2$; $\mathbf{x} = x_1x_2, \mathbf{y} = y_1y_2, \mathbf{z} = z_1z_2$. By definition

$$\begin{aligned} f^{[2]}(\mathbf{x}, \mathbf{y}, \mathbf{z}) &= \widetilde{f}^2(x_1, x_2, y_1, y_2, z_1, z_2) = \widetilde{f}(ax_1 + by_1, ax_2 + by_2, ay_1 + bz_1, ay_2 + bz_2) \\ &= a^2x_1 + 2aby_1 + b^2z_1, a^2x_2 + 2aby_2 + b^2z_2 = a^2\,\mathbf{x} + 2ab\,\mathbf{y} + b^2\,\mathbf{z}\,. \end{aligned}$$

Above, for clarity, we separate the elements of the sequences $\mathbf{u} \in \mathbb{F}_p^k$, $k \geq 1$ by a comma. This shows that $f^{[2]}$ indeed acts as "two independent copies of f". □

Note that in Lemma 71 it is useful to distinguish two cases. For $p \neq 2$, $\mathbb{B}'$ is a doubly bijective local algebra; moreover, it is clearly not isomorphic to $\mathbb{B}$ as the "middle coefficient" is non-zero. For $p = 2$, $\mathbb{B}' = \mathbb{B}$. (Over $\mathbb{F}_2$ there is only one algebra satisfying the assumptions, namely $\mathbb{B} = (\mathbb{F}_2, f)_1$ with $f(x, y, z) = x + z$; this corresponds to ECA 90.) This distinction plays a role in the following proof as well.

Theorem 59. *Let $\mathbb{B} = (\mathbb{F}_p, f)_1$ be a local algebra of a canonical additive CA with $f(x, y, z) = ax + bz$; $a, b \neq 0$. Let $\mathbb{B}' = (\mathbb{F}_p, f')_1$ where $f'(x, y, z) = a^2x + 2aby + b^2z$. Then:*

$$\mathrm{HSP}_{\mathrm{fin}}\Xi(\mathbb{B}) = \mathrm{P}_{\mathrm{fin}}\Xi(\{\mathbb{B}, \mathbb{B}'\}) \cup \mathrm{SING}_r.$$

Proof. First assume $p \neq 2$. By Lemma 71, $\mathbb{B}^{[2]} \cong \mathbb{B}' \times \mathbb{B}'$, and consequently, $\mathbb{B}^{[2l]} \cong \mathbb{B}'^{[l]} \times \mathbb{B}'^{[l]}$ for any $l \in \mathbb{N}$. Let $\mathbb{A} \in \mathrm{P}_{\mathrm{fin}}\Xi(\mathbb{B})$. Corollary 64 gives us

$$\begin{aligned} \mathbb{A} &\cong \mathbb{B}^{[l_1]} \times \cdots \times \mathbb{B}^{[l_m]} \times \mathbb{B}^{[2l'_1]} \times \cdots \times \mathbb{B}^{[2l'_{m'}]} \\ &\cong \mathbb{B}^{[l_1]} \times \cdots \times \mathbb{B}^{[l_m]} \times (\mathbb{B}'^{[l'_1]} \times \mathbb{B}'^{[l'_1]}) \times \cdots \times (\mathbb{B}'^{[l'_{m'}]} \times \mathbb{B}'^{[l'_{m'}]}) \end{aligned}$$

where all the exponents l_i, l'_i are coprime with p (and not necessarily pairwise distinct) and all the l_i are odd. Since $p \neq 2$, we know that $\mathbb{B}'$ is a doubly bijective local algebra. Hence, by Proposition 81, for any $l' \in \mathbb{N}$, $p \nmid l'$, $\mathbb{B}'^{[l']}$ is simple. Further, $\mathbb{B}$ satisfies the assumptions of Proposition 84 and thus, for any $l \in \mathbb{N}$ odd, $p \nmid l$, $\mathbb{B}^{[l]}$ is simple. (Note that Propositions 81 and 84 are proven in the next section.)

If $p = 2$, for $\mathbb{A} \in \mathrm{P}_{\mathrm{fin}}\Xi(\mathbb{B})$ it holds that $\mathbb{A} \cong \mathbb{B}^{[l_1]} \times \cdots \times \mathbb{B}^{[l_m]}$ where all the exponents l_i are odd. Thus, we can directly apply Proposition 84 to obtain that each $\mathbb{B}^{[l_i]}$ is a simple local algebra.

Thus, in either case, $\mathbb{A}$ is isomorphic to a product of simple local algebras, all belonging to $\mathrm{ADD}^{\mathbb{F}_p}_{1;-1,1}$. Hence, all the assumptions for Proposition 70 are met. Let $\mathbb{A}' \in \mathrm{S}(\mathbb{A})$. Proposition 70 (1) gives that $\mathbb{A}' \in \mathrm{P}_{\mathrm{fin}}\Xi(\{\mathbb{B}, \mathbb{B}'\})$ and is again a product of simple local algebras belonging to $\mathrm{ADD}^{\mathbb{F}_p}_{1;-1,1}$ or $\mathbb{A} \in \mathrm{SING}_r$. Similarly, Proposition 70 (2) gives that if $\mathbb{A}'' \in \mathrm{H}(\mathbb{A}')$ then $\mathbb{A}'' \in \mathrm{P}_{\mathrm{fin}}\Xi(\{\mathbb{B}, \mathbb{B}'\})$ or $\mathbb{A}'' \in \mathrm{SING}_r$. Thus, $\mathrm{HSP}_{\mathrm{fin}}\Xi(\mathbb{B}) = \mathrm{P}_{\mathrm{fin}}\Xi(\{\mathbb{B}, \mathbb{B}'\}) \cup \mathrm{SING}_r$. □

With this, we finish the analysis of the simulation capacity of all ternary canonical additive CAs, proving Corollary 60.

5.4 Invariant Subspaces of Iterated Powers

Let us fix a canonical additive CA with local algebra $\mathbb{B} = (\mathbb{F}_p, f)_r$ given by $(a_{-r}, \dots, a_r)$ and let F denote its global rule. The goal of this section is to understand subspaces $W \leq \mathbb{F}_p^l$ invariant under all components of $f^{[l]}$ for $p \nmid l$. To achieve this, we first describe some elementary properties of the components of $f^{[l]}$.

5.4.1 Components of CA Iterated Powers

In this subsection, we will study the general form of the components $f^{[n]}_{-r}, \dots, f^{[n]}_r$ for $n \in \mathbb{N}$. It will be useful to view the components $f^{[n]}_i : \mathbb{F}_p^n \to \mathbb{F}_p^n$ in their matrix form. Thus, we first introduce some notation.

Definition 72. Let $n \in \mathbb{N}$ and let p be a prime. We will denote the canonical basis of $\mathbb{F}_p^n$ by $K = (\mathbf{e}_1, \dots, \mathbf{e}_n)$. Let $(\mathbb{F}_p, f)_r$ be the local algebra of some canonical additive CA. We will denote the matrix corresponding to $f^{[n]}_i$ with respect to K as $A_{i,n} \in \mathbb{F}_p^{n \times n}$ for each $-r \leq i \leq r$.

In what follows, we will show that the matrices $A_{i,n}$, $-r \leq i \leq r$, have a very special form and are fully determined by the configuration $F^n(\cdots 00100 \cdots)$. To illustrate this, we first show two simple examples.

Example 73. Let $\mathbb{A} = (\mathbb{F}_p, f)_r$ be the local algebra of an additive CA given by $(a_{-r}, \dots, a_r)$ and let $n = p^k$ for some $k \in \mathbb{N}$. Then Lemma 62 gives us that for each $-r \leq i \leq r$, $A_{i,n} = a_i \cdot I \in \mathbb{F}_p^{n \times n}$.

Definition 74. For a finite set of states S, we define the shift map $\sigma : S^{\mathbb{Z}} \to S^{\mathbb{Z}}$ as: $\sigma(c)_i = c_{i+1}$ for each $c \in S^{\mathbb{Z}}$ and each $i \in \mathbb{Z}$. We define the reflection map $\rho : S^{\mathbb{Z}} \to S^{\mathbb{Z}}$ as: $\rho(c)_i = c_{-i}$ for each $c \in S^{\mathbb{Z}}$ and each $i \in \mathbb{Z}$. And for each $c \in S^{\mathbb{Z}}$ and each $i \leq j$, by $c_{[i,j]}$ we understand the sequence $c_i c_{i+1} \cdots c_j$. Lastly, for each $k \in \mathbb{Z}$, $e^k \in \mathbb{F}_p^{\mathbb{Z}}$ denotes the k-th “canonical” configuration: $e^k_i = \delta_{k,i}$.

Example 75 (Components of additive CAs have a special form)**.** Let $\mathbb{A} = (\mathbb{F}_3, f)_1$ be the local algebra of an additive CA given by $(2, 1, 1)$. Let F denote the CA’s global rule. We compute $F^4(e^0)$ in Figure 5.1.

Suppose we would like to find out what is the first column of the matrix $A_{0,4}$. It is straightforward that this is equal to $A_{0,4} \cdot \mathbf{e}_1 = f^{[4]}_0(\mathbf{e}_1) = f^{[4]}(\mathbf{0}, \mathbf{e}_1, \mathbf{0}) = F^4(e^0)_{[0,3]}$. This

$e^0 =$	···	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	···
$F(e^0) =$	···	0	0	0	0	0	0	1	1	2	0	0	0	0	0	0	···
$F^2(e^0) =$	···	0	0	0	0	0	1	2	2	1	1	0	0	0	0	0	···
$F^3(e^0) =$	···	0	0	0	0	1	0	0	1	0	0	2	0	0	0	0	···
$F^4(e^0) =$	···	0	0	0	1	1	2	1	1	2	2	2	1	0	0	0	···

Figure 5.1: The scheme shows the computation of $F^4(e^0)$.

corresponds to the subsequence $1, 2, 2, 2$ in the bottom line of Figure 5.1. In a similar fashion, all columns of all the matrices $A_{-1,4}, A_{0,4}, A_{1,4}$ can be obtained from the configuration $F^4(e^0)$. The form of the matrices is shown in Figure 5.2.

$$\begin{array}{c}
\left(\begin{array}{cccc|cccc|cccc}
1 & 2 & 2 & 2 & 1 & 1 & 2 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 2 & 2 & 2 & 1 & 1 & 2 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 2 & 2 & 2 & 1 & 1 & 2 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 2 & 2 & 2 & 1 & 1 & 2 & 1 & 1
\end{array}\right) \\
\begin{array}{ccc}
\qquad \| \qquad & \qquad \| \qquad & \qquad \| \qquad \\
A_{-1,4} & A_{0,4} & A_{1,4}
\end{array}
\end{array}$$

Figure 5.2: The scheme illustrates that the first row of the matrix $(A_{-1,4}|A_{0,4}|A_{1,4})$ contains the "non-zero part" of the reflected configuration $\rho(F^4(e^0))$. Further, each subsequent row of $(A_{-1,4}|A_{0,4}|A_{1,4})$ is simply obtained by shifting the previous row.

Definition 76. Let $\mathbb{F}$ be a field. We denote by $J_n \in \mathbb{F}^{n \times n}$ the matrix with a single "shifted" diagonal of 1's; $[J_n]_{i,j} = 1$ if $j = i + 1$ and $[J_n]_{i,j} = 0$ otherwise. This is illustrated in Figure 5.3.

It is straightforward to check that for $1 \leq k \leq n - 1$, $[J_n^k]_{i,j} = 1$ if $j = i + k$ and $[J_n^k]_{i,j} = 0$ otherwise. Also, that J_n^n is the zero matrix. Lastly, analogous properties hold for J_n^{T}, as illustrated in Figure 5.3.

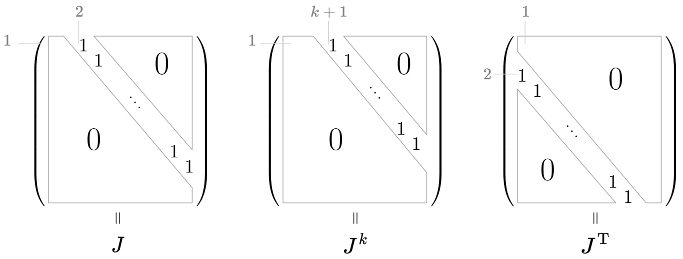


Figure 5.3: Illustration of J_n, its power and its transpose.

In Example 75, we have seen that the matrices $A_{-1,4}, A_{0,4}, A_{1,4}$ have a very special form and are fully determined by the configuration $F^4(e^0)$. Lemma 77 and Figure 5.4 show that this is the case in general.

Lemma 77. *Let $\mathbb{B} = (\mathbb{F}_p, f)_r$ be the local algebra of a canonical additive CA with global rule F. Denote $c = F^n(e^0)$. Then, for each $-r \leq i \leq r$ it holds that*

$$A_{i,n} = c_{-in} I + \sum_{k=1}^{n-1} c_{-in-k} J_n^k + \sum_{k=1}^{n-1} c_{-in+k} (J_n^{\mathrm{T}})^k. \tag{5.1}$$

This is illustrated in Figure 5.4.

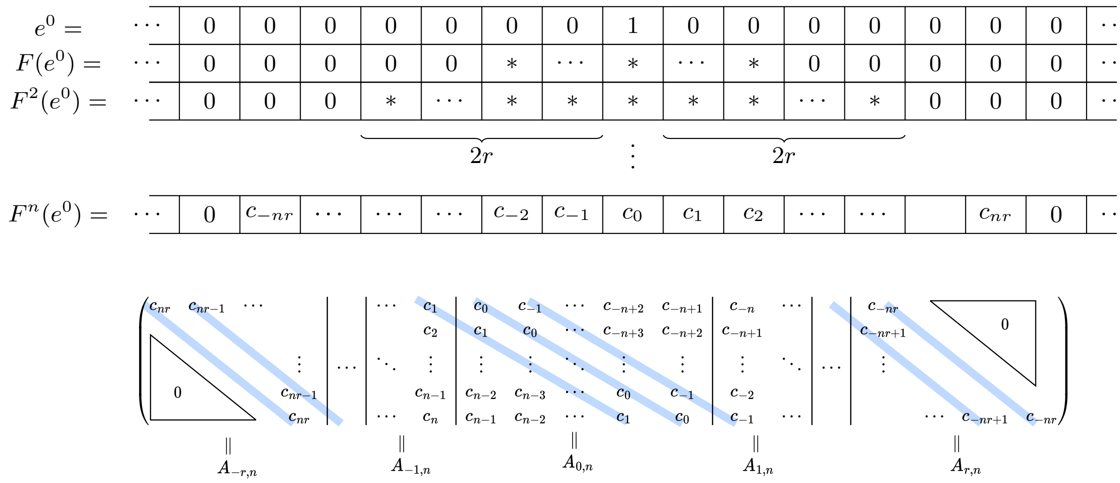


Figure 5.4: The diagram illustrates that the first row of the matrix $(A_{-r,n}| \cdots |A_{r,n})$ contains exactly the "non-zero part" of the reflected configuration $\rho(F^n(e^0))$; and, each subsequent row is obtained by shifting the previous row by one cell "to the right".

Proof. Let $-r \leq i \leq r$ and $1 \leq j \leq n$. In order to prove the statement, it suffices to show that $A_{i,n} \cdot \mathbf{e}_j = (c_{-in-j+1}, \dots, c_{-in-j+n})^{\mathrm{T}}$. And this is indeed the case, since directly by the definitions we have

$$A_{i,n} \cdot \mathbf{e}_j = f^{[n]}(\mathbf{0}, \dots, \mathbf{0}, \mathbf{e}_j, \mathbf{0}, \dots, \mathbf{0}) = \big(F^n(e^0)_{[-in-j+1,-in-j+n]}\big)^{\mathrm{T}} = (c_{-in-j+1}, \dots, c_{-in-j+n})^{\mathrm{T}}$$

where $\mathbf{e}_j$ is on the i-th position in the input of $f^{[n]}$ above. □

In what follows, we will study the subspaces $W \leq \mathbb{F}_p^n$ invariant under all the components $f_i^{[n]}$, $-r \leq i \leq r$. For that, we need more information about the matrices $A_{i,n}$. It seems challenging to fully describe such matrices for arbitrarily large n. However, Lemma 78 shows that if $\mathbb{B} \in \mathrm{ADD}_{r;i,j}^{\mathbb{F}_p}$, then we can easily describe the form of the CA's "outer components" represented by the matrices $A_{i,n}$ and $A_{j,n}$. For the case of doubly bijective CAs, this information will already be enough to conclude that for $p \nmid n$ there are no non-trivial subspaces of $\mathbb{F}_p^n$ invariant under both $f_i^{[n]}$ and $f_j^{[n]}$ and thus, that $\mathbb{B}^{[n]}$ is simple.

Lemma 78. *Let $\mathbb{B} = (\mathbb{F}_p, f)_r$ be the local algebra of an additive CA given by $(a_{-r}, \dots, a_r)$ such that $\mathbb{B} \in \mathrm{ADD}_{r;i,j}^{\mathbb{F}_p}$ for some $-r \leq i < j \leq r$. Then, for each $n \in \mathbb{N}$ it holds that $A_{-r,n} = \cdots = A_{i-1,n}$ are zero matrices and $A_{j+1,n} = \cdots = A_{r,n}$ as well. In addition, it holds that:*

$$A_{i,n} = a_i^n \cdot I + n a_i^{n-1} a_{i+1} \cdot J_n + \left(n a_i^{n-1} a_{i+2} + \binom{n}{2} a_i^{n-2} a_{i+1}^2\right) \cdot J_n^2 + \sum_{k=3}^{n-1} b_k \cdot J_n^k \qquad (5.2)$$

$$A_{j,n} = a_j^n \cdot I + n a_j^{n-1} a_{j-1} \cdot J_n^{\mathrm{T}} + \left(n a_j^{n-1} a_{j-2} + \binom{n}{2} a_j^{n-2} a_{j-1}^2\right) \cdot (J_n^{\mathrm{T}})^2 + \sum_{k=3}^{n-1} b'_k \cdot (J_n^{\mathrm{T}})^k \qquad (5.3)$$

for some $b_k, b'_k \in \mathbb{F}_p$, $3 \leq k \leq n-1$ where all the coefficients are computed in $\mathbb{F}_p$ and where $\binom{1}{2}$ is to be interpreted as 0.

Proof. We show the proof for matrix $A_{i,n}$ as the case for $A_{j,n}$ is analogous. Denote $c(n) := F^n(e^0)$ for each $n \in \mathbb{N}$. By induction on n, we will show that $c(n)_m = 0$ for $m \geq -in + 1$. Thanks to Lemma 77 this already implies that $A_{-r,n} = \cdots = A_{i-1,n}$ is the zero matrix and that $A_{i,n}$ is upper triangular. Further, we show by induction on n that $c(n)_{-in} = a_i^n$, $c(n)_{-in-1} = n a_i^{n-1} a_{i+1}$

and $c(n)_{-in-2} = na_i^{n-1}a_{i+2} + \binom{n}{2}a_i^{n-2}a_{i+1}^2$. Again, by Lemma 77, this will imply that $A_{i,n}$ has the form in (5.2).

It is straightforward to verify that the claims hold for $n = 1$. Let $n \geq 2$ and suppose the induction hypotheses hold for $n-1$, we show they hold for n. Let us denote $c = c(n-1)$ and $c' = c(n)$.

We first show that $c'_m = 0$ for $m \geq -in+1$: Let $k \in \{-in+1, \dots, nr\}$. Then, $c'_k = f(c_{k-r}, \dots, c_{k+r}) = \sum_{l=-r}^{r} a_l c_{k+l}$. Here, each summand is zero either because $k+l \geq -i(n-1)+1$ and $c_{k+l} = 0$ or because $k+l < -i(n-1)+1$, which implies that $l < i$ and in such case, $a_l = 0$.

Next, we deduce the form of $c'_{-in}, c'_{-in-1}, c'_{-in-2}$: By the induction hypothesis, we have $c_{-(n-1)i} = a_i^{n-1}$, $c_{-(n-1)i-1} = (n-1)a_i^{n-2}a_{i+1}$, and $c_{-i(n-1)-2} = (n-1)a_i^{n-2}a_{i+2} + \binom{n-1}{2}a_i^{n-3}a_{i+1}^2$. Then,

$$\begin{aligned}
c'_{-ni} &= f(c_{-ni-r}, \dots, c_{-ni+r}) \\
&= a_{-r}c_{-ni-r} + \dots + a_{i-1}c_{-ni+i-1} + a_i c_{-ni+i} + a_{i+1}c_{-ni+i+1} + \dots + a_r c_{-ni+r} \\
&= 0c_{-ni-r} + \dots + 0c_{-ni+i-1} + a_i c_{-ni+i} + a_{i+1}0 + \dots + a_r 0 = a_i a_i^{n-1} = a_i^n.
\end{aligned}$$

Similarly:

$$\begin{aligned}
c'_{-ni-1} &= f(c_{-ni-r-1}, \dots, c_{-ni+r-1}) \\
&= a_{-r}c_{-ni-r-1} + \dots + a_{i-1}c_{-ni+i-2} + a_i c_{-ni+i-1} + a_{i+1}c_{-ni+i} + \dots + a_r c_{-ni+r-1} \\
&= 0c_{-ni-r} + \dots + 0c_{-ni+i-2} + a_i c_{-ni+i-1} + a_{i+1}c_{-ni+i} + a_{i+2}0 + \dots + a_r 0 \\
&= a_i(n-1)a_i^{n-2}a_{i+1} + a_{i+1}a_i^{n-1} = na_i^{n-1}a_{i+1}.
\end{aligned}$$

And finally:

$$\begin{aligned}
c'_{-ni-2} &= f(c_{-ni-r-2}, \dots, c_{-ni+r-2}) \\
&= a_{-r}c_{-ni-r-2} + \dots + a_{i-1}c_{-ni+i-3} + a_i c_{-ni+i-2} + a_{i+1}c_{-ni+i-1} + \dots + a_r c_{-ni+r-2} \\
&= 0c_{-ni-r} + \dots + 0c_{-ni+i-2} + a_i c_{-ni+i-2} + a_{i+1}c_{-ni+i-1} + a_{i+2}c_{-ni+i} + \dots + a_r 0 \\
&= a_i\left((n-1)a_i^{n-2}a_{i+2} + \binom{n-1}{2}a_i^{n-3}a_{i+1}^2\right) + a_{i+1}\left((n-1)a_i^{n-2}a_{i+1}\right) + a_{i+2}a_i^{n-1} \\
&= (n-1)a_i^{n-1}a_{i+2} + \binom{n-1}{2}a_i^{n-2}a_{i+1}^2 + (n-1)a_i^{n-2}a_{i+1}^2 + a_i^{n-1}a_{i+2} \\
&= na_i^{n-1}a_{i+2} + \binom{n}{2}a_i^{n-2}a_{i+1}^2.
\end{aligned}$$

□

Corollary 79. *Let $\mathbb{B} = (\mathbb{F}_p, f)_r$ be the local algebra of a CA; $\mathbb{B} \in \mathrm{ADD}_{r;i,j}^{\mathbb{F}_p}$. Let $\mathbb{B}$ be given by $(a_{-r}, \dots, a_r)$. Then, for each $n \in \mathbb{N}$, $A_{i,n}$ is upper triangular with a_i^n on the diagonal and $A_{j,n}$ is lower triangular with a_j^n on the diagonal. Therefore, $A_{i,n}$ and $A_{j,n}$ are regular. In particular, this implies that for each n, $\mathbb{B}^{[n]} \in \mathrm{ADD}_{r;i,j}^{\mathbb{F}_p}$.*

Lemma 78 gives us even stronger consequences. In particular, for $\mathbb{B} = (\mathbb{F}_p, f)_r \in \mathrm{ADD}_{r;i,j}^{\mathbb{F}_p}$ we will use it in the next subsection to show that, under certain assumptions, there is no non-trivial subspace of $\mathbb{F}_p^n$ invariant under both $f_i^{[n]}$ and $f_j^{[n]}$ for $p \nmid n$.

5.4.2 Analysing Invariant Subspaces of CA Iterated Powers

Let us fix a local algebra $\mathbb{B} = (\mathbb{F}_p, f)_r$ of a canonical additive CA. In this subsection, we shall study two cases in which we are able to prove that $\mathbb{B}^{[n]}$, $p \nmid n$, is a simple local algebra. First, we study the case when $\mathbb{B}$ is doubly bijective. We start by proving a useful lemma.

Lemma 80. *Let p be a prime, $n \in \mathbb{N}$, and let V over $\mathbb{F}_p$ of dimension n. Let $f_L, f_R : V \to V$ be linear mappings which satisfy the following: There exists a basis $(\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n)$ of V such that*

$$\begin{aligned}
&\langle \mathbf{b}_1, \dots, \mathbf{b}_n \rangle \overset{f_L}{\twoheadrightarrow} \langle \mathbf{b}_1, \dots, \mathbf{b}_{n-1} \rangle \overset{f_L}{\twoheadrightarrow} \cdots \overset{f_L}{\twoheadrightarrow} \langle \mathbf{b}_1, \mathbf{b}_2 \rangle \overset{f_L}{\twoheadrightarrow} \langle \mathbf{b}_1 \rangle \overset{f_L}{\twoheadrightarrow} \{\mathbf{0}\},\\
&\langle \mathbf{b}_1, \dots, \mathbf{b}_n \rangle \overset{f_R}{\twoheadrightarrow} \langle \mathbf{b}_2, \dots, \mathbf{b}_n \rangle \overset{f_R}{\twoheadrightarrow} \cdots \overset{f_R}{\twoheadrightarrow} \langle \mathbf{b}_{n-1}, \mathbf{b}_n \rangle \overset{f_R}{\twoheadrightarrow} \langle \mathbf{b}_n \rangle \overset{f_R}{\twoheadrightarrow} \{\mathbf{0}\}
\end{aligned}$$

where $\twoheadrightarrow$ denotes surjectivity and $\langle \cdot \rangle$ denotes the linear span. Then, the only subspaces of V invariant under both f_L and f_R are $\{\mathbf{0}\}$ and V.

Proof. Let $\{\mathbf{0}\} \neq W \leq V$ be a subspace invariant under both f_L and f_R and let $\mathbf{0} \neq \mathbf{w} \in W$. Since f_L is nilpotent, there must exist some $1 \leq k \leq n$ such that $f_L^k(\mathbf{w}) = \mathbf{0}$ and $f_L^{k-1}(\mathbf{w}) \neq \mathbf{0}$ (we assert that $f_L^0(\mathbf{w}) = \mathbf{w}$). Thus, $f_L^{k-1}(\mathbf{w}) \in \operatorname{Ker}(f_L) = \langle \mathbf{b}_1 \rangle$ and hence, $\mathbf{b}_1 \in W$. An analogous argument for f_R yields $\mathbf{b}_n \in W$.

Next, let $1 < i \leq n$. Since $f_L(\langle \mathbf{b}_1, \dots, \mathbf{b}_i \rangle) = \langle \mathbf{b}_1, \dots, \mathbf{b}_{i-1} \rangle$ and $f_L(\langle \mathbf{b}_1, \dots, \mathbf{b}_{i-1} \rangle) = \langle \mathbf{b}_1, \dots, \mathbf{b}_{i-2} \rangle$, we have that $f_L(\mathbf{b}_i) = c \cdot \mathbf{b}_{i-1} + \mathbf{v}$ for some $c \neq 0$ and some $\mathbf{v} \in \langle \mathbf{b}_1, \dots, \mathbf{b}_{i-2} \rangle$. Iterating this argument gives $f_L^{n-i}(\mathbf{b}_n) = c \cdot \mathbf{b}_i + \mathbf{v} \in W$ for some $c \neq 0$ and some $\mathbf{v} \in \langle \mathbf{b}_1, \dots, \mathbf{b}_{i-1} \rangle$.

Recall that $\mathbf{b}_1, \mathbf{b}_n \in W$. Let $1 < i < n$ and suppose that $\mathbf{b}_1, \dots \mathbf{b}_{i-1} \in W$. We show that $\mathbf{b}_i \in W$. (Then the proof will be immediately finished by induction.) We have that $f_L^{n-i}(\mathbf{b}_n) = c \cdot \mathbf{b}_i + \mathbf{v} \in W$ for some $c \neq 0$ and some $\mathbf{v} \in \langle \mathbf{b}_1, \dots, \mathbf{b}_{i-1} \rangle$. The induction hypothesis gives that $\mathbf{v} \in W$ and thus, $\mathbf{b}_i \in W$. Hence, $W = V$. □

Proposition 81. *Let $\mathbb{B} = (\mathbb{F}_p, f)_r$ be the local algebra of a canonical additive CA that is doubly bijective. Let $n \in \mathbb{N}$ such that $p \nmid n$. Then, $\mathbb{B}^{[n]}$ is simple.*

Proof. Let $\mathbb{B}$ be given by $(a_{-r}, \dots, a_r)$; $\mathbb{B} \in \mathrm{ADD}^{\mathbb{F}_p}_{r;i,j}$. In fact, we will show that if $W \leq \mathbb{F}_p^n$ is invariant under both $A_{i,n}$ and $A_{j,n}$, it implies that either $W = \{\mathbf{0}\}$ or $W = \mathbb{F}_p^n$. From Lemma 78, we have that $A_{i,n}$ is upper triangular, $A_{i,n} = b \cdot I + b_1 \cdot J_n + \sum_{k=2}^{n-1} b_k \cdot J_n^k$ where $b = a_i^n \neq 0$ and $b_1 = n a_i^{n-1} a_{i+1} p \neq 0$ (where the coefficients are computed, of course, modulo p). Similarly, we have that $A_{j,n}$ is lower triangular, $A_{j,n} = b' \cdot I + b'_1 \cdot J_n^{\mathrm{T}} + \sum_{k=2}^{n-1} b'_k \cdot (J_n^{\mathrm{T}})^k$ where $b' = a_j^n \neq 0$ and $b'_1 = n a_j^{n-1} a_{j-1} \neq 0$.

Let $g : \mathbb{F}_p^n \to \mathbb{F}_p^n$ be an arbitrary linear mapping and $c, d \in \mathbb{F}_p$, $c \neq 0$, arbitrary constants. We observe that $W \leq \mathbb{F}_p^n$ is invariant under g if and only if it is invariant under $c(g - d\,\mathrm{id})$.

Thus, we consider mappings $f_L \coloneqq f_i^{[n]} - b\,\mathrm{id}$, $f_R \coloneqq f_j^{[n]} - b'\,\mathrm{id}$, and show that there are no non-trivial subspaces invariant under both f_L and f_R. The mappings have the following form with respect to the canonical basis:

$$[f_L]_K = \begin{pmatrix} 0 & 1 & * & \cdots & * & * \\ 0 & 0 & 1 & \cdots & * & * \\ 0 & 0 & 0 & \cdots & * & * \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{pmatrix} \text{ and } [f_R]_K = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ * & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ * & * & * & \cdots & 0 & 0 \\ * & * & * & \cdots & 1 & 0 \end{pmatrix}.$$

It is now easy to see that the maps f_L, f_R satisfy the assumptions of Lemma 80 where the witnessing basis is the canonical one. Therefore, the only subspaces of $\mathbb{F}_p^n$ invariant under both f_L and f_R are indeed the trivial ones. Hence, $\mathbb{B}^{[n]}$ is simple. □

We finish this section by analysing invariant spaces of iterated powers of CAs with radius $r = 1$ given by $(a_{-1}, 0, a_1)$. We first state a general lemma that we subsequently use to show that in such a case, $\mathbb{B}^{[n]}$ is simple for any $n \in \mathbb{N}$ odd, $p \nmid n$.

Lemma 82. *Let p be a prime, $n \in \mathbb{N}$ odd, and let V a vector space over $\mathbb{F}_p$ of dimension n. Let $f_L, f_R : V \to V$ be linear mappings which satisfy the following: There exists a basis $(\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n)$ of V such that:*

$$\langle \mathbf{b}_1, \mathbf{b}_3, \dots, \mathbf{b}_n \rangle \xrightarrow{f_L}\!\!\!\!\twoheadrightarrow \langle \mathbf{b}_1, \mathbf{b}_3, \dots, \mathbf{b}_{n-2} \rangle \xrightarrow{f_L}\!\!\!\!\twoheadrightarrow \cdots \xrightarrow{f_L}\!\!\!\!\twoheadrightarrow \langle \mathbf{b}_1, \mathbf{b}_3 \rangle \xrightarrow{f_L}\!\!\!\!\twoheadrightarrow \langle \mathbf{b}_1 \rangle \xrightarrow{f_L}\!\!\!\!\twoheadrightarrow \{\mathbf{0}\},$$
$$\langle \mathbf{b}_2, \mathbf{b}_4, \dots, \mathbf{b}_{n-1} \rangle \xrightarrow{f_L}\!\!\!\!\twoheadrightarrow \langle \mathbf{b}_2, \mathbf{b}_4, \dots, \mathbf{b}_{n-3} \rangle \xrightarrow{f_L}\!\!\!\!\twoheadrightarrow \cdots \xrightarrow{f_L}\!\!\!\!\twoheadrightarrow \langle \mathbf{b}_2, \mathbf{b}_4 \rangle \xrightarrow{f_L}\!\!\!\!\twoheadrightarrow \langle \mathbf{b}_2 \rangle \xrightarrow{f_L}\!\!\!\!\twoheadrightarrow \{\mathbf{0}\},$$

and

$$\langle \mathbf{b}_1, \mathbf{b}_3, \dots, \mathbf{b}_n \rangle \xrightarrow{f_R}\!\!\!\!\twoheadrightarrow \langle \mathbf{b}_3, \dots, \mathbf{b}_{n-2}, \mathbf{b}_n \rangle \xrightarrow{f_R}\!\!\!\!\twoheadrightarrow \cdots \xrightarrow{f_R}\!\!\!\!\twoheadrightarrow \langle \mathbf{b}_{n-2}, \mathbf{b}_n \rangle \xrightarrow{f_R}\!\!\!\!\twoheadrightarrow \langle \mathbf{b}_n \rangle \xrightarrow{f_R}\!\!\!\!\twoheadrightarrow \{\mathbf{0}\},$$
$$\langle \mathbf{b}_2, \mathbf{b}_4, \dots, \mathbf{b}_{n-1} \rangle \xrightarrow{f_R}\!\!\!\!\twoheadrightarrow \langle \mathbf{b}_4, \dots, \mathbf{b}_{n-3}, \mathbf{b}_{n-1} \rangle \xrightarrow{f_R}\!\!\!\!\twoheadrightarrow \cdots \xrightarrow{f_R}\!\!\!\!\twoheadrightarrow \langle \mathbf{b}_{n-3}, \mathbf{b}_{n-1} \rangle \xrightarrow{f_R}\!\!\!\!\twoheadrightarrow \langle \mathbf{b}_{n-1} \rangle \xrightarrow{f_R}\!\!\!\!\twoheadrightarrow \{\mathbf{0}\}$$

where $\twoheadrightarrow$ denotes surjectivity and $\langle \cdot \rangle$ denotes the linear span.

We define the subspaces $V_{\text{odd}} = \langle \{\mathbf{b}_i \mid i \text{ odd}\} \rangle \le V$ and $V_{\text{even}} = \langle \{\mathbf{b}_i \mid i \text{ even}\} \rangle \le V$. Let $W \le V$ be a non-trivial subspace invariant under both f_L and f_R. Then either $W = V_{\text{odd}}$ or $W = V_{\text{even}}$.

Proof.

a) Let $\{\mathbf{0}\} \neq W \le V_{\text{odd}}$ be a subspace invariant under both f_L and f_R. The mappings $f_L|_{V_{\text{odd}}}$ and $f_R|_{V_{\text{odd}}}$ satisfy the assumptions of Lemma 80 and thus, we have $W = V_{\text{odd}}$.

Analogously, we obtain that if $\{\mathbf{0}\} \neq W \le V_{\text{even}}$ is a subspace invariant under both f_L and f_R then $W = V_{\text{even}}$.

b) We have that $V = V_{\text{odd}} \oplus V_{\text{even}}$ and we denote by $\pi_{\text{odd}} : V \to V_{\text{odd}}$ and $\pi_{\text{even}} : V \to V_{\text{even}}$ the corresponding projections. We show that $\pi_{\text{odd}} \circ f_L = f_L \circ \pi_{\text{odd}}$.

Let $\mathbf{u} \in V_{\text{odd}}$ and $\mathbf{v} \in V_{\text{even}}$. Then $\pi_{\text{odd}}(f_L(\mathbf{u}+\mathbf{v})) = \pi_{\text{odd}}(f_L(\mathbf{u}) + f_L(\mathbf{v})) = f_L(\mathbf{u}) = f_L(\pi_{\text{odd}}(\mathbf{u}+\mathbf{v}))$. Analogously, one can verify that $\pi_{\text{odd}} \circ f_R = f_R \circ \pi_{\text{odd}}$, $\pi_{\text{even}} \circ f_L = f_L \circ \pi_{\text{even}}$, and $\pi_{\text{even}} \circ f_R = f_R \circ \pi_{\text{even}}$.

c) Let $\{\mathbf{0}\} \neq W \le V$ be invariant under both f_L and f_R. If $\pi_{\text{even}}(W) = \{\mathbf{0}\}$ then clearly, $W \le V_{\text{odd}}$ and from a) we know that $W = V_{\text{odd}}$. Analogously, $\pi_{\text{odd}}(W) = \{\mathbf{0}\}$ implies that $W = V_{\text{even}}$. Hence, let us assume that $\pi_{\text{odd}}(W) \neq \{\mathbf{0}\} \le V_{\text{odd}}$ and $\pi_{\text{even}}(W) \neq \{\mathbf{0}\} \le V_{\text{even}}$. Since the operators π_{odd}, π_{even} commute with the operators f_L, f_R, we have that $f_L(\pi_{\text{odd}}(W)) = \pi_{\text{odd}}(f_L(W)) \subseteq \pi_{\text{odd}}(W)$ and similarly $f_R(\pi_{\text{odd}}(W)) \subseteq \pi_{\text{odd}}(W)$. Hence, $\pi_{\text{odd}}(W) \le V_{\text{odd}}$ is invariant under both f_L and f_R and thus, by a) we have $\pi_{\text{odd}}(W) = V_{\text{odd}}$. Analogously, we obtain that $\pi_{\text{even}}(W) = V_{\text{even}}$.

Since n is odd, $|V_{\text{odd}}| > |V_{\text{even}}|$ and there exist $\mathbf{u} \in V_{\text{odd}}$ and $\mathbf{v}, \mathbf{v}' \in V_{\text{even}}$, $\mathbf{v} \neq \mathbf{v}'$, such that $\mathbf{u}+\mathbf{v} \in W$ and $\mathbf{u}+\mathbf{v}' \in W$. Then, $\mathbf{v} - \mathbf{v}' \in W$ and we have that $\{\mathbf{0}\} \neq W \cap V_{\text{even}} \le V_{\text{even}}$ is a subspace invariant under both f_L and f_R. Thus, from a) we have that $V_{\text{even}} \le W$ and consequently, $W = V_{\text{odd}} + V_{\text{even}} = V$. □

This finishes the proof of one of the most important results of this section, already outlined in Proposition 57 (3).

Lemma 83. *Let $\mathbb{B} = (\mathbb{F}_p, f)_1$ be the local algebra of a canonical additive CA where $f(x, y, z) = a_{-1}x + a_1 z$. Then, for each $n \in \mathbb{N}$ odd it holds that $F^n(e^0)_i = 0$ for each i even. Further, for each $n \in \mathbb{N}$ even it holds that $F^n(e^0)_i = 0$ for each i odd.*

Proof. This is clear, as in the computation of odd positions of $F(c)$ for any configuration c one only uses even positions of c and vice versa. □

Proposition 84. *Let $\mathbb{B} = (\mathbb{F}_p, f)_1$ be the local algebra of a canonical additive CA where $f(x, y, z) = a_{-1}x + a_1 z$ with $a_{-1}, a_1 \neq 0$. Let $n \in \mathbb{N}$ be odd such that $p \nmid n$. Then, $\mathbb{B}^{[n]}$ is simple.*

Proof. We will show that, after certain transformations that preserve the set of invariant subspaces, the mappings $f_{-1}^{[n]}$ and $f_1^{[n]}$ satisfy the assumptions of Lemma 82 with the witnessing basis being the canonical $(\mathbf{e}_1, \dots, \mathbf{e}_n)$. This will yield that the only possible non-trivial subspaces of $\mathbb{F}_p^n$ invariant under both $f_{-1}^{[n]}$ and $f_1^{[n]}$ are the spaces $V_{\text{odd}} = \langle\{\mathbf{e}_i \mid i \text{ odd}\}\rangle \leq V$ and $V_{\text{even}} = \langle\{\mathbf{e}_i \mid i \text{ even}\}\rangle \leq V$. We will finish the proof by showing that neither V_{odd} nor V_{even} is invariant under $f_0^{[n]}$.

Denote $c = F^n(e^0)$. Lemma 83 gives us that $c_i = 0$ whenever i is even. Lemma 77 thus gives us that the matrices $A_{-1,n}, A_{0,n}, A_{1,n}$ have the following form:

$$\begin{array}{ccc}
A_{-1,n} & A_{0,n} & A_{1,n} \\
\shortparallel & \shortparallel & \shortparallel \\
\begin{pmatrix}
c_n & 0 & c_{n-2} & \cdots & 0 & c_1 \\
0 & c_n & 0 & \cdots & c_3 & 0 \\
0 & 0 & c_n & \cdots & 0 & c_3 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & c_n & 0 \\
0 & 0 & 0 & \cdots & 0 & c_n
\end{pmatrix}
&
\begin{pmatrix}
0 & c_{-1} & \cdots & c_{-(n-2)} & 0 \\
c_1 & 0 & \cdots & 0 & c_{-(n-2)} \\
0 & c_1 & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
c_{n-2} & 0 & \cdots & 0 & c_{-1} \\
0 & c_{n-2} & \cdots & c_1 & 0
\end{pmatrix}
&
\begin{pmatrix}
c_{-n} & 0 & 0 & \cdots & 0 & 0 \\
0 & c_{-n} & 0 & \cdots & 0 & 0 \\
c_{-(n-2)} & 0 & c_{-n} & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & c_{-3} & 0 & \cdots & c_{-n} & 0 \\
c_{-1} & 0 & c_{-3} & \cdots & 0 & c_{-n}
\end{pmatrix}
\end{array} \tag{5.4}$$

We can see that the matrices $A_{-1,n}$ and $A_{1,n}$ have zeroes on "odd diagonals" and $A_{0,n}$ has zeroes on "even diagonals". Moreover, Lemma 78 gives that $c_n = a_{-1}^n \neq 0$ and $c_{n-2} = \left(na_{-1}^{n-1}a_1 + \binom{n}{2}a_{-1}^{n-2}a_0^2\right) = na_{-1}^{n-1}a_1 \neq 0$ (where the coefficients are computed, of course, modulo p). Similarly, $c_{-n} = a_1^n \neq 0$ and $c_{-(n-2)} = na_1^{n-1}a_0 \neq 0$.

Let $g : \mathbb{F}_p^n \to \mathbb{F}_p^n$ be an arbitrary linear mapping and $c, d \in \mathbb{F}_p$, $c \neq 0$, arbitrary constants. Again, we observe that $W \leq \mathbb{F}_p^n$ is invariant under g if and only if it is invariant under $c(g - d\,\text{id})$. We put $f_L := f_{-1}^{[n]} - c_n\,\text{id}$ and $f_R := f_1^{[n]} - c_{-n}\,\text{id}$. Now, it is easy to see that the maps f_L, f_R satisfy the assumptions of Lemma 82 where the witnessing basis is the canonical basis $(\mathbf{e}_1, \dots, \mathbf{e}_n)$. Therefore, the only subspaces of $\mathbb{F}_p^n$ invariant under both f_L and f_R are V_{even} and V_{odd}.

We finish the proof by showing that neither V_{odd} nor V_{even} is invariant under $f_0^{[n]}$. We know that $A_{0,n}$ has the form in (5.4) and that $c_{n-2} \neq 0$. Hence, $f_0^{[n]}(\mathbf{e}_1) = c_1\,\mathbf{e}_2 + c_3\,\mathbf{e}_4 + \cdots + c_{n-2}\,\mathbf{e}_{n-1} \notin V_{\text{odd}}$. Similarly, $f_0^{[n]}(\mathbf{e}_2) = c_{-1}\,\mathbf{e}_1 + c_1\,\mathbf{e}_3 + \cdots + c_{n-2}\,\mathbf{e}_n \notin V_{\text{even}}$. This finishes the proof that $\mathbb{B}^{[n]}$ is simple. □

5.5 Concluding Remarks

Again, the proofs in this chapter do not require the understanding of any deeper algebraic theorems. However, we again highlight a connection to some well-established results from universal algebra with the following example. It is a well-known result that for a congruence-permutable algebra that is a subdirect product of finitely many simple algebras, it holds that the algebra is isomorphic to a direct product of their subset (proof can be found e.g., in [124, Theorem 2.4.7]. Since any local algebra of a canonical additive CA satisfying the bijective condition is congruence-permutable, Proposition 70 (1) is a direct consequence of this result.

Bibliography

[1] Jürgen Albert and Karel Culik II. A simple universal cellular automaton and its one-way and totalistic version. *Complex Systems*, 1:1–16, 1987.

[2] Maximino Aldana, Susan Coppersmith, and Leo P Kadanoff. Boolean dynamics with random couplings. *Perspectives and Problems in Nolinear Science: A Celebratory Volume in Honor of Lawrence Sirovich*, pages 23–89, 2003.

[3] Jean-Paul Allouche, Friedrich von Haeseler, Heinz-Otto Peitgen, and Gencho Skordev. Linear cellular automata, finite automata and pascal's triangle. *Discrete Applied Mathematics*, 66(1):1–22, 1996.

[4] F Altarelli, A Braunstein, L Dall'Asta, and R Zecchina. Optimizing spread dynamics on graphs by message passing. *Journal of Statistical Mechanics: Theory and Experiment*, 2013(09):P09011, sep 2013.

[5] Erik Aurell and Hamed Mahmoudi. Dynamic mean-field and cavity methods for diluted Ising systems. *Physical Review E*, page 12, 2012.

[6] Robert Axelrod. The dissemination of culture: A model with local convergence and global polarization. *Journal of conflict resolution*, 41(2):203–226, 1997.

[7] Franco Bagnoli and Raul Rechtman. Phase Transitions of Cellular Automata. *arxiv*, 2014.

[8] Per Bak, Kan Chen, and Michael Creutz. Self-organized criticality in the'game of life. *Nature*, 342(6251):780–782, 1989.

[9] Per Bak, Chao Tang, and Kurt Wiesenfeld. Self-organized criticality: An explanation of the 1/f noise. *Physical review letters*, 59(4):381, 1987.

[10] Stefania Bandini, Giancarlo Mauri, and Roberto Serra. Cellular automata: From a theoretical parallel computational model to its application to complex systems. *Parallel Computing*, 27(5):539–553, 2001.

[11] Thomas Barthel. Matrix product algorithm for stochastic dynamics on networks applied to nonequilibrium Glauber dynamics. *Physial Review E*, page 6, 2018.

[12] John E. Bates and Harvey K. Shepard. Measuring complexity using information fluctuation. *Physics Letters A*, 172(6):416–425, 1993.

[13] C. Bédard, H. Kröger, and A. Destexhe. Does the $1/f$ frequency scaling of brain signals reflect self-organized critical states? *Phys. Rev. Lett.*, 97:118102, Sep 2006.

[14] Freya Behrens, Gabriel Arpino, Yaroslav Kivva, and Lenka Zdeborová. (Dis)assortative Partitions on Random Regular Graphs. *Journal of Physics A: Mathematical and Theoretical*, 55(39):395004, 2022.

[15] Freya Behrens, Barbora Hudcová, and Lenka Zdeborová. The backtracking dynamical cavity method. *Phys. Rev. X*, 13, Aug 2023.

[16] Freya Behrens, Barbora Hudcová, and Lenka Zdeborová. Dynamical phase transitions in graph cellular automata, 2023.

[17] Clifford Bergman. *Universal Algebra, Fundamentals and Selected Topics.* CRC Press, 2011.

[18] Elwyn R Berlekamp, John H Conway, and Richard K Guy. *Winning ways for your mathematical plays, volume 2.* AK Peters/CRC Press, 2004.

[19] Amartya Bhattacharjya and Shoudan Liang. Median attractor and transients in random boolean nets. *Physica D: Nonlinear Phenomena*, 95(1):29–34, 1996.

[20] Sydney Brenner. Life's code script. *Nature*, 482(7386):461–461, 2012.

[21] Ana Busic, Nazim Fates, Jean Mairesse, and Irene Marcovici. Density Classification on Infinite Lattices and Trees. *LATIN 2012: Theoretical Informatics*, page 12, 2012.

[22] Mathieu S Capcarrere, Moshe Sipper, and Marco Tomassini. Two-state, r= 1 cellular automaton that classifies density. *Physical review letters*, 77(24):4969, 1996.

[23] Claudio Castellano, Santo Fortunato, and Vittorio Loreto. Statistical physics of social dynamics. *Reviews of modern physics*, 81(2):591, 2009.

[24] Alastair Channon. Unbounded evolutionary dynamics in a system of agents that actively process and transform their environment. *Genetic Programming and Evolvable Machines*, 7:253–281, 09 2006.

[25] Hugo Cisneros, Tomas Mikolov, and Josef Sivic. Benchmarking learning efficiency in deep reservoir computing. In *Conference on Lifelong Learning Agents*, pages 532–547. PMLR, 2022.

[26] Hugo Cisneros, Josef Sivic, and Tomas Mikolov. Evolving structures in complex systems. *Proceedings of the 2019 IEEE Symposium Series on Computational Intelligence*, pages 230–238, 2019.

[27] Matthew Cook et al. Universality in elementary cellular automata. *Complex systems*, 15(1):1–40, 2004.

[28] James P. Crutchfield and James E. Hanson. Turbulent pattern bases for cellular automata. *Physica D: Nonlinear Phenomena*, 69(3):279 – 301, 1993.

[29] Karel Culik II and Shuangni Yu. Yu, s.: Undecidability of ca classification schemes. complex systems 2, 177-190. *Complex Systems*, 2, 01 1988.

[30] Michael Damron and Arnab Sen. Zero-temperature Glauber dynamics on the 3-regular tree and the median process. *Probability Theory and Related Fields*, 178(1):25–68, 2020.

[31] Yatin Dandi, David Gamarnik, and Lenka Zdeborová. Maximally-stable local optima in random graphs and spin glasses: Phase transitions and universality. *arXiv preprint arXiv:2305.03591*, 2023.

[32] Rajarshi Das, James P. Crutchfield, Melanie Mitchell, and James E. Hanson. Evolving globally synchronized cellular automata. In *Proceedings of the 6th International Conference on Genetic Algorithms*, page 336–343, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[33] Gino Del Ferraro and Erik Aurell. Dynamic message-passing approach for kinetic spin models with reversible dynamics. *Physical Review E*, 92(1):010102, 2015.

[34] Marianne Delorme, Jacques Mazoyer, Nicolas Ollinger, and Guillaume Theyssier. Bulking i: an abstract theory of bulking. *Theoretical Computer Science*, 412(30):3866–3880, 2011.

[35] Marianne Delorme, Jacques Mazoyer, Nicolas Ollinger, and Guillaume Theyssier. Bulking ii: Classifications of cellular automata. *Theoretical Computer Science*, 412(30):3881–3905, 2011.

[36] Jean-Charles Delvenne, Petr Kůrka, and Vincent Blondel. Decidability and universality in symbolic dynamical systems. *Fundamenta Informaticae*, 74(4):463–490, 2006.

[37] B. Derrida and H. Flyvbjerg. The random map model: A disordered model with deterministic dynamics. *Journal De Physique*, 48:971–978, 1987.

[38] B. Derrida and Y. Pomeau. Random networks of automata: A simple annealed approximation. *Europhys. Lett.*, 1:45–49, Jan 1986.

[39] Bruno Durand and Zsuzsanna Róka. The game of life: universality revisited. *Cellular Automata: a Parallel Model*, pages 51–74, 1999.

[40] Jérôme Olivier Durand-Lose. Intrinsic universality of a 1-dimensional reversible cellular automaton. In Rüdiger Reischuk and Michel Morvan, editors, *STACS 97*, pages 439–450, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.

[41] Jérôme Olivier Durand-Lose. Reversible space–time simulation of cellular automata. *Theoretical computer science*, 246(1-2):117–129, 2000.

[42] Samuel Eilenberg and Marcel-Paul Schützenberger. On pseudovarieties. *Advances in Mathematics*, 19(3):413–418, 1976.

[43] H. Flyvbjerg and N. J. Kjaer. Exact solution of kauffman's model with connectivity one. *Journal of Physics A: Mathematical and General*, 21(7):1695–1718, apr 1988.

[44] Serge Galam. Majority rule, hierarchical structures, and democratic totalitarianism: A statistical approach. *Journal of Mathematical Psychology*, 30(4):426–434, 1986.

[45] Serge Galam. Contrarian deterministic effects on opinion dynamics:"the hung elections scenario". *Physica A: Statistical Mechanics and its Applications*, 333:453–460, 2004.

[46] Martin Gardener. The fantastic combinations of John Conway's new solitaire game "life" by Martin Gardner. *Scientific American*, 223:120–123, 1970.

[47] Carlos Gershenson. Introduction to random boolean networks. *arXiv preprint nlin/0408006*, 2004.

[48] E. Goles and J. Olivos. Periodic behaviour of generalized threshold functions. *Discrete Mathematics*, 30(2):187–189, 1980.

[49] Eric Goles, Pierre-Etienne Meunier, Ivan Rapaport, and Guillaume Theyssier. Communication complexity and intrinsic universality in cellular automata. *Theoretical Computer Science*, 412(1-2):2–21, 2011.

[50] Eric Goles-Chacc, Françoise Fogelman-Soulie, and Didier Pellegrin. Decreasing energy functions as a tool for studying threshold networks. *Discrete Applied Mathematics*, 12(3):261–277, 1985.

[51] Michel Grabisch and Fen Li. Anti-conformism in the threshold model of collective behavior. *Dynamic Games and Applications*, 10(2):444–477, 2020.

[52] Michel Grabisch and Agnieszka Rusinowska. A survey on nonstrategic models of opinion dynamics. *Games*, 11(4):65, 2020.

[53] Pu-hua Guan. Cellular automaton public-key cryptosystem. *Complex Systems*, 1:51–57, 1987.

[54] Pu-hua Guan and Yu He. Exact results for deterministic cellular automata with additive rules. *Journal of Statistical Physics*, 43:463–478, 1986.

[55] Heinz Peter Gumm. Algebras in congruence permutable varieties: geometrical properties of affine algebras. *Algebra Universalis*, 9:8–34, 1979.

[56] Howard A. Gutowitz. A hierarchical classification of cellular automata. *Physica D: Nonlinear Phenomena*, 45(1):136–156, 1990.

[57] Howard A. Gutowitz. Transients, cycles, and complexity in cellular automata. *Physical Review A*, 44, 12 1994.

[58] Howard A. Gutowitz, Jonathan D. Victor, and Bruce W. Knight. Local structure theory for cellular automata. *Physica D: Nonlinear Phenomena*, 28(1):18 – 48, 1987.

[59] James E. Hanson. Emergent Phenomena in Cellular Automata. *Meyers R. (eds) Encyclopedia of Complexity and Systems Science*, 2009.

[60] James E. Hanson and James P. Crutchfield. Computational mechanics of cellular automata: An example. *Physica D: Nonlinear Phenomena*, 103(1-4):169–189, 1997.

[61] B. Harris. Probability distributions related to random mappings. *Annals of Mathematical Statistics*, 31:1045–1062, 1960.

[62] JPL Hatchett, B Wemmenhove, I Pérez Castillo, T Nikoletopoulos, NS Skantzos, and ACC Coolen. Parallel dynamics of disordered ising spin systems on finitely connected random graphs. *Journal of Physics A: Mathematical and General*, 37(24):6201, 2004.

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

[64] Sui Huang and Donald E. Ingber. Shape-dependent control of cell growth, differentiation, and apoptosis: Switching between attractors in cell regulatory networks. *Experimental Cell Research*, 261(1):91–103, 2000.

[65] Barbora Hudcová. Complexity in cellular automata. Master's thesis, Charles University, Faculty of Mathematics and Physics, 2020.

[66] Barbora Hudcová and Jakub Krásenský. Simulation limitations of affine cellular automata, 2023.

[67] Barbora Hudcová and Tomáš Mikolov. Classification of Discrete Dynamical Systems Based on Transients. *Artificial Life*, 27(3–4):220–245, 03 2022.

[68] Christian John Hurry, Alexander Mozeika, and Alessia Annibale. Dynamics of sparse boolean networks with multi-node and self-interactions. *Journal of Physics A: Mathematical and Theoretical*, 55(41):415003, 2022. Publisher: IOP Publishing.

[69] Sungmin Hwang, Enrico Lanza, Giorgio Parisi, Jacopo Rocchi, Giancarlo Ruocco, and Francesco Zamponi. On the number of limit cycles in diluted neural networks. *Journal of Statistical Physics*, 181(6):2304–2321, 2020.

[70] Erica Jen. Linear cellular automata and recurring sequences in finite fields. *Communications in Mathematical Physics*, 119:13–28, 1988.

[71] Leo Kadanoff, Susan Coppersmith, and Maximino Aldana. Boolean dynamics with random couplings. *Perspectives and Problems in Nonlinear Science*, 05 2002.

[72] Kunihiko Kaneko. Complexity in basin structures and information processing by the transition among attractors. *Theory and Applications of Cellular Automata*, pages 367–399, 01 1985.

[73] Yashodhan Kanoria and Andrea Montanari. Majority dynamics on trees and the dynamic cavity method. *The Annals of Applied Probability*, 21(5), 2011.

[74] Jarkko Kari. Reversibility of 2d cellular automata is undecidable. *Physica D: Nonlinear Phenomena*, 45(1-3):379–385, 1990.

[75] Jarkko Kari. Cryptosystems based on reversible cellular automata. *Manuscript, August*, 1992.

[76] Jarkko Kari. Theory of cellular automata: A survey. *Theoretical Computer Science*, 334(1):3 – 33, 2005.

[77] Jarkko Kari and Nicolas Ollinger. Periodicity and immortality in reversible computing. In *International Symposium on Mathematical Foundations of Computer Science*, pages 419–430. Springer, 2008.

[78] Brian Karrer and M. E. J. Newman. Message passing approach for general epidemic models. *Physical Review E*, 82(1):016101, 2012.

[79] S.A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22(3):437–467, 1969.

[80] S.A. Kauffman and E. D. Weinberger. The nk model of rugged fitness landscapes and its application to maturation od the immune response. *Journal of Theoretical Biology*, 141:211–245, 1989.

[81] Petr Kůrka. Topological dynamics of cellular automata. In *Codes, Systems, and Graphical Models*, pages 447–485. Springer, 2001.

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

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

[84] Christopher G. Langton. Computation at the edge of chaos: Phase transitions and emergent computation. *Physica D: nonlinear phenomena*, 42(1-3):12–37, 1990.

[85] Wentian Li, Norman H. Packard, et al. The structure of the elementary cellular automata rule space. *Complex systems*, 4(3):281–297, 1990.

[86] Wentian Li, Norman H. Packard, and Chris G. Langton. Transition phenomena in cellular automata rule space. *Physica D: Nonlinear Phenomena*, 45(1):77–94, 1990.

[87] Kristin Lindgren and Mats G. Nordahl. Complexity measures and cellular automata. *Complex Syst.*, 2(4):409–440, aug 1988.

[88] Andrey Y. Lokhov, Marc Mézard, and Lenka Zdeborová. Dynamic message-passing equations for models with unidirectional dynamics. *Physical Review E*, 91(1):012811, 2015.

[89] Amahury Jafet López-Díaz, Fernanda Sánchez-Puig, and Carlos Gershenson. Temporal, structural, and functional heterogeneities extend criticality and antifragility in random boolean networks. *Entropy*, 25(2):254, 2023.

[90] Mantas Lukoševičius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer science review*, 3(3):127–149, 2009.

[91] Bartolo Luque and Ricard V. Solé. Phase transitions in random networks: Simple analytic determination of critical points. *Phys. Rev. E*, 55:257–260, Jan 1997.

[92] Bartolo Luque and Ricard V. Solé. Lyapunov exponents in random boolean networks. *Physica A: Statistical Mechanics and its Applications*, 284(1):33–45, 2000.

[93] J. F. Lynch. A criterion for stability in random boolean cellular automata. *arXiv: Adaptation and Self-Organizing Systems*, 1993.

[94] Bruce D Malamud, Gleb Morein, and Donald L Turcotte. Forest fires: an example of self-organized critical behavior. *Science*, 281(5384):1840–1842, 1998.

[95] Carsten Marr and Marc-Thorsten Hütt. Outer-totalistic cellular automata on graphs. *Physics Letters A*, 373(5):546–549, 2009.

[96] Olivier Martin, Andrew Odlyzko, and Stephen Wolfram. Algebraic properties of cellular automata. *Communications in Mathematical Physics*, 93, 06 1984.

[97] Jacques Mazoyer and Ivan Rapaport. Additive cellular automata over $\mathbb{Z}_p$ and the bottom of $(CA, \leq)$. In *Mathematical Foundations of Computer Science 1998: 23rd International Symposium, MFCS'98 Brno, Czech Republic, August 24–28, 1998 Proceedings 23*, pages 834–843. Springer, 1998.

[98] Jacques Mazoyer and Ivan Rapaport. Inducing an order on cellular automata by a grouping operation. In *STACS 98: 15th Annual Symposium on Theoretical Aspects of Computer Science Paris, France, February 25–27, 1998 Proceedings*, pages 116–127. Springer, 2005.

[99] Marc Mézard and Andrea Montanari. *Information, physics, and computation.* Oxford University Press, 2009.

[100] Marc Mézard and Giorgio Parisi. The bethe lattice spin glass revisited. *The European Physical Journal B-Condensed Matter and Complex Systems*, 20:217–233, 2001.

[101] Marc Mézard and Giorgio Parisi. The cavity method at zero temperature. *Journal of Statistical Physics*, 111:1–34, 2003.

[102] Kazushi Mimura and A. C. C. Coolen. Parallel dynamics of disordered Ising spin systems on finitely connected directed random graphs with arbitrary degree distributions. *Journal of Physics A: Mathematical and Theoretical*, 42(41):415001, 2009.

[103] Melanie Mitchell. Computation in cellular automata: A selected review. *Non-Standard Computation*, pages 95–140, 1998.

[104] Melanie Mitchell. *Complexity: A guided tour.* Oxford university press, 2009.

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

[106] Melanie Mitchell, Peter Hraber, and James P. Crutchfield. Revisiting the edge of chaos: Evolving cellular automata to perform computations. *arXiv preprint adap-org/9303003*, 1993.

[107] Alexander Mordvintsev, Ettore Randazzo, Eyvind Niklasson, and Michael Levin. Growing neural cellular automata. *Distill*, 5(2):e23, 2020.

[108] Andrés Moreira. Universality and decidability of number-conserving cellular automata. *Theoretical computer science*, 292(3):711–721, 2003.

[109] Robert Morris. Zero-temperature Glauber dynamics on $\mathbb{Z}^d$. *Probability Theory and Related Fields*, 149(3):417–434, 2011.

[110] Roni Muslim, Sasfan A. Wella, and Ahmad R. T. Nugraha. Phase transition in the majority rule model with the nonconformist agents. *Physica A: Statistical Mechanics and its Applications*, 608:128307, 2022.

[111] I. Neri and D. Bollé. The cavity approach to parallel dynamics of Ising spins on a graph. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(08):P08009, 2009.

[112] John von Neumann and Arthur W. Burks. *Theory of Self-Reproducing Automata.* University of Illinois Press, Urbana, USA, 1966.

[113] Stefano Nichele, Mathias Berild Ose, Sebastian Risi, and Gunnar Tufte. Ca-neat: Evolved compositional pattern producing networks for cellular automata morphogenesis and replication. *IEEE Transactions on Cognitive and Developmental Systems*, 10(3):687–700, 2018.

[114] Bartłomiej Nowak and Katarzyna Sznajd-Weron. Homogeneous symmetrical threshold model with nonconformity: Independence versus anticonformity. *Complexity*, 2019, 2019.

[115] Charles Ofria and Claus O. Wilke. Avida: A software platform for research in computational evolutionary biology. *Artificial Life*, 10(2):191–229, 3 2004.

[116] Nicolas Ollinger. The quest for small universal cellular automata. In *Automata, Languages and Programming: 29th International Colloquium, ICALP 2002 Málaga, Spain, July 8–13, 2002 Proceedings 29*, pages 318–329. Springer, 2002.

[117] Nicolas Ollinger. The intrinsic universality problem of one-dimensional cellular automata. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 632–641. Springer, 2003.

[118] Nicolas Ollinger. *Universalities in Cellular Automata**, pages 189–229. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[119] Art B. Owen. *Monte Carlo theory, methods and examples.* `https://artowen.su.domains/mc/`, 2013.

[120] Norman H. Packard. Adaptation toward the edge of chaos. *Dynamic patterns in complex systems*, 212:293–301, 1988.

[121] David Peleg. Local majorities, coalitions and monopolies in graphs: A review. *Theoretical Computer Science*, 282(2):231–257, 2002.

[122] Niels K. Petersen and Preben Alstrøm. Phase transition in an elementary probabilistic cellular automaton. *Physica A: Statistical Mechanics and its Applications*, 235(3):473–485, 1997.

[123] Omar Pineda, Hyobin Kim, and Carlos Gershenson. A novel antifragility measure based on satisfaction and its application to random and biological boolean networks. *Complexity*, 2019:1–10, 05 2019.

[124] Michael Pinsker. Rosenberg's characterization of maximal clones. Master's thesis, Vienna University of Technology, 2002.

[125] Sidney Pontes-Filho, Pedro Lind, Anis Yazidi, Jianhua Zhang, Hugo Hammer, Gustavo BM Mello, Ioanna Sandvig, Gunnar Tufte, and Stefano Nichele. A neuro-inspired general framework for the evolution of stochastic dynamical systems: Cellular automata, random boolean networks and echo state networks towards criticality. *Cognitive Neurodynamics*, 14(5):657–674, 2020.

[126] Mikhail Prokopenko, Fabio Boschetti, and Alex J Ryan. An information-theoretic primer on complexity, self-organization, and emergence. *Complexity*, 15(1):11–28, 2009.

[127] Ettore Randazzo, Alexander Mordvintsev, Eyvind Niklasson, Michael Levin, and Sam Greydanus. Self-classifying mnist digits. *Distill*, 5(8):e00027–002, 2020.

[128] Thomas S. Ray. An approach to the synthesis of life. *Artificial Life II, Santa Fe Institute Studies in the Sciences of Complexity*, XI:371–408, 1991.

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

[130] Paul Rendell. Turing universality of the game of life. *Collision-based computing*, pages 513–539, 2002.

[131] Yurii Rogozhin. Small universal turing machines. *Theoretical Computer Science*, 168(2):215–240, 1996.

[132] Mathieu Sablik and Guillaume Theyssier. Topological dynamics of cellular automata: Dimension matters. *Theory of Computing Systems*, 48(3):693–714, 2011.

[133] Guillermo Santamaría-Bonfil, Carlos Gershenson, and Nelson Fernández. A package for measuring emergence, self-organization, and complexity based on shannon entropy. *Frontiers in Robotics and AI*, 4:10, 2017.

[134] Palash Sarkar. A brief history of cellular automata. *ACM Comput. Surv.*, 32(1):80–107, 2000.

[135] Roberto H. Schonmann. On the Behavior of Some Cellular Automata Related to Bootstrap Percolation. *The Annals of Probability*, 20(1):174–193, 1992.

[136] Cosma Rohilla Shalizi and James P. Crutchfield. Computational mechanics: Pattern and prediction, structure and simplicity. *Journal of statistical physics*, 104:817–879, 2001.

[137] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[138] Jonathan D.H. Smith. Mal'cev varieties. *Springer Lecture Notes*, 554, 1976.

[139] Robert I. Soare. *Turing Computability, Theory and Applications.* Springer-Verlag, 2016.

[140] Z Somogyvari and Sz Payrits. Length of state cycles of random boolean networks: an analytic study. *Journal of Physics A: Mathematical and General*, 33(38):6699, 2000.

[141] Lisa Soros and Kenneth Stanley. Identifying necessary conditions for open-ended evolution through the artificial life world of chromaria. *Artificial Life Conference Proceedings*, pages 793–800, 2014.

[142] Susan Stepney. *Nonclassical Computation — A Dynamical Systems Perspective.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[143] Klaus Sutner. Universality and cellular automata. In *International Conference on Machines, Computations, and Universality*, pages 50–59. Springer, 2004.

[144] Katarzyna Sznajd-Weron and Jozef Sznajd. Opinion evolution in closed community. *International Journal of Modern Physics C*, 11(06):1157–1165, 2000.

[145] Satoshi Takahashi. Self-similarity of linear cellular automata. *Journal of Computer and System Sciences*, 44(1):114–140, 1992.

[146] Tommaso Toffoli. Computation and construction universality of reversible cellular automata. *Journal of Computer and System Sciences*, 15(2):213 – 231, 1977.

[147] Giuseppe Torrisi, Reimer Kühn, and Alessia Annibale. Uncovering the non-equilibrium stationary properties in sparse Boolean networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2022(5):053303, May 2022.

[148] A. M. Turing. On computable numbers with as application to the entscheidungsproblem. *Proc. London Math. Soc.*, 2:230–265, 1936.

[149] Alexandre Variengien, Stefano Nichele, Tom Glover, and Sidney Pontes-Filho. Towards self-organized control: Using neural cellular automata to robustly control a cart-pole agent. *arXiv preprint arXiv:2106.15240*, 2021.

[150] Gérard Y. Vichniac. Simulating physics with cellular automata. *Physica D: Nonlinear Phenomena*, 10(1):96 – 116, 1984.

[151] Allan R. Vieira and Celia Anteneodo. Threshold q-voter model. *Physical Review E*, 97(5):052106, 2018.

[152] Abraham Waksman. An optimum solution to the firing squad synchronization problem. *Information and control*, 9(1):66–78, 1966.

[153] Mitchell M. Waldrop. *Complexity: The emerging science at the edge of order and chaos.* Simon and Schuster, 1993.

[154] X. Wang, Joseph Lizier, and Mikhail Prokopenko. Fisher information at the edge of chaos in random boolean networks. *Artificial life*, 17:315–29, 07 2011.

[155] Nicholas W. Watkins, Gunnar Pruessner, Sandra C. Chapman, Norma B Crosby, and Henrik J Jensen. 25 years of self-organized criticality: Concepts and controversies. *Space Science Reviews*, 198:3–44, 2016.

[156] Stephen J. Willson. On the ergodic theory of cellular automata. *Mathematical systems theory*, 9:132–141, 1975.

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

[158] Stephen Wolfram. Universality and complexity in cellular automata. *Physica D: Nonlinear Phenomena*, 10(1):1–35, 1984.

[159] Stephen Wolfram. *A New Kind of Science.* Wolfram Media, Champaign, USA, 2002.

[160] Andrew Wuensche. The emergence of memory categorisation far from equilibrium. *Towards a Science of Consciousness: The First Tuscon Discussions and Debates*, pages 383–392, 1996.

[161] Andrew Wuensche. *Exploring discrete dynamics - Second Edition. The DDLab manual.* Luniver Press, 2016.

[162] Andrew Wuensche and Mike 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.

[163] Ozgur Yilmaz. Reservoir computing using cellular automata, 2014.

[164] Ahad N. Zehmakan. Two Phase Transitions in Two-Way Bootstrap Percolation. In Pinyan Lu and Guochuan Zhang, editors, *30th International Symposium on Algorithms and Computation (ISAAC 2019)*, volume 149 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 5:1–5:21, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

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

[166] Hector Zenil. On the dynamic qualitative behavior of universal computation. *Complex Systems*, 20, 01 2012.

[167] Pan Zhang. Inference of kinetic Ising model on sparse graphs. *Journal of Statistical Physics*, 148(3):502–512, 2012.

List of publications

- B. Hudcová, T. Mikolov, *Classification of Complex Systems Based on Transients*, Artificial Life Conference Proceedings, MIT Press, 367–375 (2020).
- B. Hudcová, T. Mikolov, *Computational Hierarchy of Elementary Cellular Automata*, Artificial Life Conference Proceedings, MIT Press, 361–368 (2021).
- B. Hudcová, T. Mikolov, *Classification of Discrete Dynamical Systems Based on Transients*, Artificial Life, 27 (3-4), MIT Press, 220–245 (2022).
- T. Lindell, B. Hudcová, S. Nichele *Canonical Computations in Cellular Automata and Their Application for Reservoir Computing*, Artificial Life Conference Proceedings, MIT Press, 109–117 (2023).
- F. Behrens, B. Hudcová, L. Zdeborová, *The Backtracking Dynamical Cavity Method*, Phys. Rev. X, 13 (3), American Physical Society (2023).
- F. Behrens, B. Hudcová, L. Zdeborová, *Dynamical Phase Transitions in Graph Cellular Automata*, submitted (2023). Available at arXiv:2310.15894.
- B. Hudcová, S. Nichele, T. Mikolov *Studying Encoder-Decoder Relation between Cellular Automata to Uncover Their Computational Structure*, submitted (2023). Available at SSRN:4244761.
- B. Hudcová, J. Krásenský, *Simulation Limitations of Affine Cellular Automata*, submitted (2023). Available at arXiv:2311.14477.