Applications of cellular automata in description of complex physical and biological systems

František Batysta
Supervisor: Petr Jizba

Faculty of Nuclear Sciences and Physical Engineering

Czech Technical University

Declaration

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady uvedené v přiloženém seznamu. Nemám žádný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

I declare that I wrote my bachelor thesis independently and exclusively with the use of cited bibliography. I agree with the usage of this thesis in a purport of the Act 121/2000 (Copyright Act).

Praha, July 24, 2011 František Batysta

Acknowledgements

And I would like to extend a particularly deep thank to Dr. Petr Jizba, who was my thesis advisor, who have supported me all along. Especially, I am grateful to him for the care with which he reviewed the original manuscript; and for conversations that clarified my thinking on this and other matters.

Also, I would like to thank my parents for supporting me and providing me with the opportunity to be where I am.

Abstract

In this thesis we study cellular automata (CA) – discrete dynamic systems on a regular lattice, which evolve synchronously in discrete time steps according to prescribed evolution rules. In our work we briefly introduce a concept of CA and discuss some CA properties such as an ability of CA to perform universal computation.

In the second part of our work we have demonstrated a wide applicability of CA in numerous systems. These included biological systems, such as ants complex behavior or elementary reversible CA systems (ERCA). Particularly for ERCA we have shown that they are very rich in their long-time statistical structure. An important result of this work is, that we have identified a proper (but non-classical) probability space for each ERCA in which the evolution rule is ergodic.

Key words Cellular automaton, ergodicity, entropy, universal Turing machine, self organization, ant trails.

Tato práce se zabývá aplikacemi celulárních automatů (CA) - diskrétních dynamických systémů definovaných na pravidelné mřížce, které se synchronně vyvíjí v diskrétních časových krocích na základě předem definovaných pravidel. V rámci rešeršní části této práce jsou zavedeny příslušné definice a uvedeny základní vlastnosti CA, například schopnost některých CA provádět libovolné početní úkony.

Ve druhé části této práce jsou předvedeny možnosti aplikace CA při popisu různých fyzikálních a biologických systémů, jako např. komplexní chování mravenců. Dále se zabýváme statistickými vlastnostmi tak zvaných elementárních reverzibilních celulárních automatů (ERCA). Důležitý výsledek této práce je, že jsme našli pro každý ERCA takový pravděpodobnostní prostor, ve kterém se evoluční pravidlo chová ergodicky.

Klíčová slova Celulární automat, ergodicita, entropie, univerzální Turingův stroj, samoorganizace, mravenčí cesty.

Contents

List of Figures

Chapter 1

Introduction

Cellular automata (CA) were invented by John von Neumann and his close friend Stanisław Ulam in the early 1950, when the former outlined a plan for create self-replicating machines [1]. These machines were referred by von Neumann as the automaton [2].

In this first chapter, we briefly describe what is a cellular automaton and give arguments why it is useful to study them. In the second chapter an abstract CA is defined mathematically and some classical results about CA are given. The chapters three and four contains two CA applications in describing complex systems: Ergodicity of the deterministic universe and the ant trail simulation. The last, fifth chapter concludes this thesis.

1.1 What is cellular automaton

In order to provide some insight into cellular automata, we will specify a rough list of properties that common CA has in most cases [3]. Abstract cellular automaton is a mathematical object with following properties:

- Discrete lattice of cells: Cellular automaton consist of finite (or infinite) number of elementary units called *cells*, that form in the lattice. All cells are supposed to be equivalent.

- Discrete states: Each cell takes one of finite number of possible discrete states.

- All cells are updates synchronously in a discrete time steps according to a local transition rule.
- Transition rule is deterministic. It provides a state of each individual cell in the next time step taking into account the states of cells in a neighborhood of the evaluated cell. The local transition rule should be same for all cells in the lattice.

In this thesis we will call this CA rules as *generic CA*. Although CA described by these properties often perform very rich and complex behavior, various generalizations and extensions are studied not less frequently. Such a generalizations are usually obtained by omitting any of the aforementioned properties. In other words, above defining rules serves as base for further generalizations.

1.2 Why study cellular automata

There are several motivations for study CA. In order to demonstrate usefulness of CA, let us give various examples of CA applications.

1.2.1 Powerful, inherently parallel computation

Individual cells can be constructed in form of *hardware*, which would considerably improve the speed of parallel computation. The fluid flow near and around aeroplane wings modeled by lattice gases CA [4, 5] is an example of such an application. This concept was suggested by Tommaso Toffoli in his article about *Programmable Matter* [6]. Toffoli and MIT's Information mechanics Group have build several *Cellular Automata Machines* (CAM), which work inherently parallel. Other promising way is the model of Asynchronous Logic Automata (ALA), which is derived by adding the temporal locality, i.e., the cells are updated asynchronously. The logical circuit for such a machine were proposed by Kailiang Chen [7].

1.2.2 Fundamental study of complex systems

Behavior of some complex systems, both abstract and real, is irreducible, i.e. each attempt to predict long term behavior by some simplifying like, say, solution of motion for 2-body problem, will fail [8]. These systems therefore define the most efficient way to simulate its own behavior [3]. CA are the most simple example of dynamical system, which are capable to exhibit complex behavior. They were, as simple models of complexity in nature, systematically investigated by Stephen Wolfram [9, 10, 11].

1.2.3 Discrete approximation to dynamical systems behavior

Wherever a lot of equivalent objects interact with each other, CA may be good approach to simulate their dynamic. We will give a few examples: CA can help in physics with simulation of Road Traffic, Lattice Gas, Diffusion phenomena, Non-equilibrium Phase Transitions or Modeling a Snow Transport by Wind (chapters in the book [12]). Chemical systems were simulated using cellular automata [13], as well as biological ones [14].

1.2.4 Models of fundamental physics

John Bell proved in 1964 [15], that any physical hidden variable theory is incompatible with statistical predictions of quantum mechanics. However, recent works [16] shows that quantum mechanics may be an emergent phenomenon in sense of t' Hooft's pre-quantization [17].

In principle, CA may represent appropriate underlying structure of quantum theory. Till now, partial progress has been achieved in this direction. Cellular automata with additional algebraic structure equivalent to the that of quantum mechanic harmonic oscillator were constructed [18, 19]. Moreover, an interesting consideration about CA model that could describe quantum mechanics has been made by G. t' Hooft [20].

Chapter 2

Definition, classical results

In this chapter we would like to formulate CA algebraically and give several examples of the most common CA. Then, we will briefly review the most interesting CA properties and classifications.

2.1 Abstract definition of generic cellular automata

The definition of generic cellular automata will follow the terminology of the Ilachinski's monograph [3] and is also inspired by the article [21].

Definition 1 (CA space). *Let $\Sigma = \{0, 1, 2 \ldots, k-1\}$ be a finite set of k cell-states. The space, where cellular automaton lives is a $d-$dimensional discrete Euclidean space. The configuration of a cellular automaton at a time t can be than represented by a map $\sigma(t) : \mathbb{Z}^d \to \Sigma$, where $t \in \mathbb{N}_0$. The set of all CA configurations σ is denoted as P.*

Definition 2 (CA evolution rule). *An evolution step from time t to $(t+1)$ is specified by local evolution rule $\varphi : \underbrace{\Sigma \times \Sigma \cdots \times \Sigma}_{n} \to \Sigma$, where n denotes a size of cell's neighborhood. The neighborhood itself can be identified with set of indices*

$$\mathcal{N}_{\vec{i}} = \mathcal{N}_{\vec{0}} + \vec{i},$$

where $\mathcal{N}_{\vec{0}}$ is the neighborhood of the origin.

Corresponding global evolution rule which updates all cells is called Φ, and maps $\sigma(t) \mapsto \sigma(t+1)$. The value of the cell at position $\vec{i}$ and time t can be expressed as $\sigma(\vec{i};t)$, or simply $\sigma_{\vec{i}}$. Using this notation, the state of the $i-th$ cell in the next time step is given by following equation

$$\sigma_{\vec{i}}(t+1) = \varphi\left(\sigma_{\vec{j}}(t)\right), \qquad \vec{j} \in \mathcal{N}_{\vec{i}} \tag{2.1}$$

Remark 1. In general, a lattice of CA is supposed to be infinite. However, when performing computer simulations, one can work only with a finite lattice. Therefore, it is convenient to introduce periodic boundary condition (Born–von Carman boundary conditions). For example, boundary conditions of two dimensional CA are

$$(\forall t \in \mathbb{N})\,(\forall i,j \in \mathbb{Z})\,(\sigma_{i+M,j} = \sigma_{i,j} \wedge \sigma_{i,j+N} = \sigma_{i,j})\,, \tag{2.2}$$

where $M, N \in \mathbb{N}$ are the periods of the lattice.

Remark 2. Let n be a given size of neighborhood and k a number of cell states. The question is how many possible distinct evolution rules are there for given n and$k$. A local evolution rule is uniquely defined when a certain value of a cell is assigned for all k^n states of neighborhood. Hence, the local evolution rule is defined by an k^n-digit number in a base-k number system, which gives us k^{k^n} distinct evolution rules for given n and k.

This makes systematic study of all rules (with given n and k) quite difficult, unless both n and k are *sufficiently small* natural numbers.

2.2 CA examples

2.2.1 One-dimensional Cellular automata

We can assume a symmetric neighborhood of *range* r. Thus, the general local evolution rule of 1-dimensional CA has a form

$$\sigma_i(t+1) = \varphi\left(\sigma_{i-r}(t), \sigma_{i-r+1}(t), \ldots, \sigma_{i+r}(t)\right). \tag{2.3}$$

The simplest non-trivial class of CA is *range* $r = 1$ and $\Sigma = \{0, 1\}$. The evolution rule of such CAs is given by

$$\sigma_i(t+1) = \varphi\left(\sigma_{i-1}(t), \sigma_i(t), \sigma_{i+1}(t)\right). \tag{2.4}$$

Equation (2.4) is commonly represented graphically (see Fig. 2.1). Since there are only $2^{2^3} = 256$ distinct local evolution rules of the form (2.4), all rules have been studied systematically by Stephen Wolfram [11].

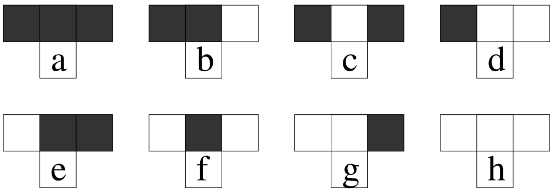
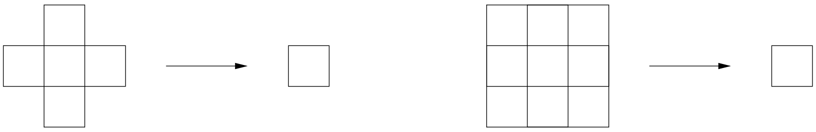


Figure 2.1: General one-dimensional ($r = 1$,$k = 2$) CA with two states: $\Sigma = \{0, 1\}$. The black square $\equiv 1$, white square $\equiv 0$. All letters $a, b, \ldots h$ take values from the set Σ, according to particular rule.

Wolfram introduced (today standard) naming system of such CA [22]. Each rule is associated with an eight digit binary number ($abcdefgh$) and with its corresponding decimal form.

2.2.2 Two-dimensional CA

In this section we explain the notation commonly used in description of two dimensional CA. In two dimensions one have more possibilities in choosing a neighborhood shape. CAs of our interest live in regular Euclidean lattice and two the most common types of neighborhoods are considered (see Fig. 2.2).



(a) Five cell neighbourhood (b) Nine cell neighborhood

Figure 2.2: Examples of neighborhoods in two dimensions.

Having 2-dimensional lattice, the equation (2.1) is usually expressed with the help of two indices

$$\sigma_{ij}(t+1) = \varphi\left(\sigma_{kl}(t)\right), \qquad (k,l) \in \mathcal{N}_{ij}. \tag{2.5}$$

Example 1 (Conway's Game of Life)**.** Life was invented by John Conway in 1970 and is indisputably the most famous 2–dimensional CA at all. It is two–dimensional, two–state CA defined on a lattice equipped with nine-point neighborhood. The local evolution rule is described in s following way [23]:

1. Any live cell with fewer than two live neighbors dies, as if by needs caused by under-population.
2. Any live cell with more than three live neighbors dies, as if by overcrowding.
3. Any live cell with two or three live neighbors lives, unchanged, to the next generation.
4. Any dead cell with exactly three live neighbors cells will come to life.

The popularity of the "Life" comes from its theoretical significance as well as from a beauty of its evolution. The behavior of the Life appears like evolution of living organism. Often, the evolution is quite surprising; one never knows, how will the pattern develop. The rule is chosen carefully so that small structures (gliders and spaceships) can propagate across the lattice while remaining local. Interactions of these structures allow complex patterns to emerge. Among many interesting properties we review here the most important theoretical results.

1. Game of Life is universal CA [24] [1]
2. Self replicating machines were explicitly constructed [25].
3. Life obey so called Halting theorem [8]. In general, one can not decide, whether the computation terminates after finite amount of time or continues forever [2].

[1] The universality of CA is described in the section 2.3.2

[2] In fact, the items 2 an 3 are the results of the universality, which is a very strong property

Further information, Java applets and articles on Conway's Game of Life can be found in [23, 26, 27]

2.2.3 Reversible CA

Physical phenomena at a microscopic level are typically considered as being time reversible, i.e., backward deterministic[1]. This is a motivation for study of CAs, which are supposed to be reversible at their primordial (i.e., microscopic) level. For CA can be the definition of reversibility formulated more precisely:

Definition 3. *A CA with the global evolution rule Φ is called reversible, if exists an inverse global evolution rule Ψ, such that $\Phi \circ \Psi = \mathrm{Id}$.*

We will mention several reversible CA examples, that are used in literature. In order to simplify the notation, we will often use 1-dimensional (r, k) CA for illustration of CA analysis. Generalization for other CA types is in most cases straightforward. In the next chapter, we will analyze some of them using various methods.

Example 2. elementary reversible CA

N. Markolus introduced in his article [29] a straightforward way to write down a reversible rule with help of irreversible one. For φ - arbitrary (r, k) CA, the reversible rule is defined as

$$\sigma_i(t+1) = \varphi\left[\sigma_{i-r}(t), \ldots, \sigma_{i+r}(t)\right] \ominus_k \sigma_i(t-1), \tag{2.6}$$

where $\ominus_k$ denotes subtraction *modulo-k*. The rules obtained from this process are frequently called *elementary–reversible CA* (ERCA), see Ref. [3]. □

This class of rules is not only reversible, but it is also invariant under time reversal in sense of following equation, which is a simple rearrangement of equation (2.6).

$$\sigma_i(t-1) = \varphi\left[\sigma_{i-r}(t), \ldots, \sigma_{i+r}(t)\right] \ominus_k \sigma_i(t+1). \tag{2.7}$$

[1] However, kaons, B-meson etc. violate this symmetry [28].

For general d-dimensional Euclidean space take equations (2.6) and (2.7) form

$$\sigma_{\vec{i}}(t+1) = \varphi\left(\sigma_{\vec{j}}(t)\right) \ominus_k \sigma_{\vec{i}}(t-1), \qquad \vec{j} \in \mathcal{N}_{\vec{i}}, \tag{2.8}$$

$$\sigma_{\vec{i}}(t-1) = \varphi\left(\sigma_{\vec{j}}(t)\right) \ominus_k \sigma_{\vec{i}}(t+1), \qquad \vec{j} \in \mathcal{N}_{\vec{i}}, \tag{2.9}$$

where $\mathcal{N}_{\vec{i}}$ is a neighborhood of the $\vec{i} - th$ cell.

Obviously, rule (2.6) is *second order* in time[1], which makes ERCA rules generalized ones. However, the following theorem shows, that each second-order rule can be viewed as a generic one.

Theorem 1. *For each CA (not necessary reversible) second order in time with local evolution rule*

$$\varphi: \quad \sigma_i(t+2) = \varphi\left[\sigma_{i+j}(t+1), \sigma_{i+j}(t)\right], \quad j \in \{-r, \ldots, r\}. \tag{2.10}$$

exists an equivalent first order (r, k^2) generic reversible rule, such that cells of the first order rule consist of two consecutive cells of the original second order rule.

Proof. We merge two consecutive cells together, so that $\forall i \in \mathbb{Z}$ the new cells consist of doublets:

$$\begin{aligned} \tilde{\sigma}_i(t) &\equiv [\sigma_i(t), \sigma_i(t+1)] \\ \tilde{\sigma}_i(t+1) &\equiv [\sigma_i(t+1), \sigma_i(t+2)] \\ &\ldots \end{aligned}$$

The situation is illustrated in the figure 2.3.

[1] two consecutive states are necessary to predict new state.

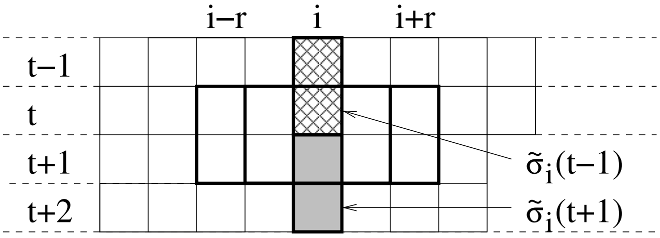


Figure 2.3: The merged state $\tilde{\sigma}_i(t{+}1)$ (gray cells) can be expressed as a function of merged cells $\tilde{\sigma}$ at time t and at coordinates $i-r, \dots, i+r$ (highlighted rectangle). In this picture: $r = 2$.

It is not difficult to see, that for $\forall i$ the cell $\tilde{\sigma}_i(t+1)$ can be expressed as a function of cell states $\sigma_j(\tau)$, where $j \in \{i-r, \dots, i+r\}$ and $\tau \in \{t, t+1\}$. We will show this for both parts of the doublet $\tilde{\sigma}_i(t+1) \equiv [\sigma_i(t+1), \sigma_i(t+2)]$.

1. $\sigma_i(t+1)$ is part of both merged cells $\tilde{\sigma}_i(t)$ and $\tilde{\sigma}_i(t+1)$. Hence, statement of the previous paragraph holds.

2. $\sigma_i(t{+}2)$: According equation (2.10) is $\sigma_i(t{+}2)$ function of $\sigma_i(t)$ and $\sigma_j(t{+}1)$, $j \in \{i-r, \dots, i+r\}$, what was to be proved.

Thus, we can write an evolution formula for $\tilde{\sigma}_i(t+1)$:

$$\tilde{\sigma}_i(t+1) = \tilde{\varphi}[\tilde{\sigma}_{i-r}(t), \dots, \tilde{\sigma}_{i+r}(t)],$$

which means, that $\tilde{\varphi}$ is *first order* generic (r, k^2) reversible rule. Rules Φ and $\tilde{\Phi}$ differ only in the definition of *cell*; but geometrical pattern remains identical. Therefore, we can say that rules Φ and $\tilde{\Phi}$ are equivalent. □

2.3 Behavioral classification and universality

This section introduces commonly used behavioral classification of CA according to Stephen Wolfram. Then, computational universality of class c4 CA is discussed.

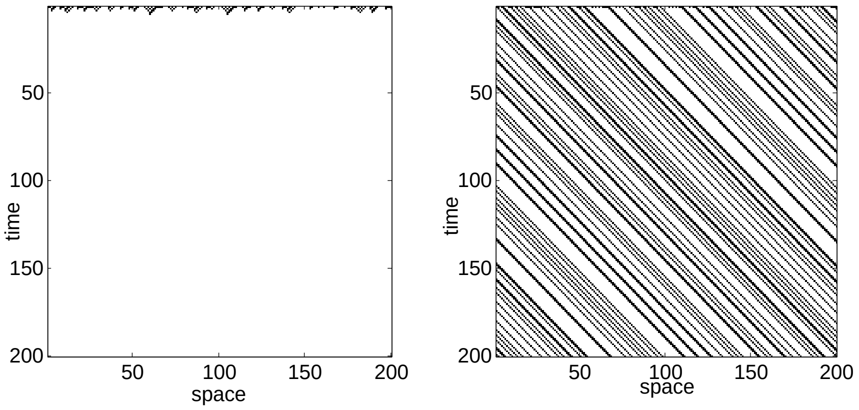
2.3.1 Wolfram's behavioral classes

The simplest way how to characterize CA dynamics is just have a look at the figure of given CA. Stephen Wolfram introduced four behavioral classes which enable us to make quick characterization of CA [22].

- Class c1: All cells eventually die out after short period of time.
- Class c2: Simple stable states or periodic and separated structures emerge.
- Class c3: Chaotic non-periodic patterns are generated.
- Class c4: Localized complex patterns are formed; they propagate and interact with each other.

The rules of the last class **c4** are probably the most exciting ones. They may be able to simulate logical operations and therefore, some of them, could serve as an universal computer, like a Turing machine. The universality of CA is a subject of the next part of this chapter.

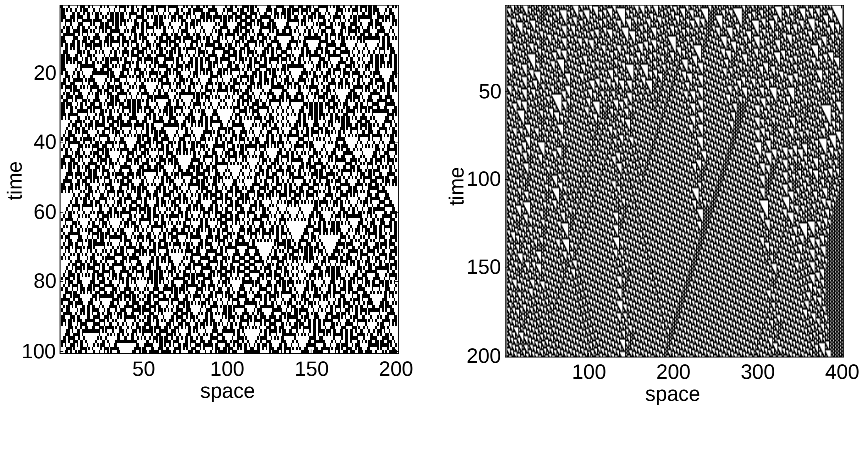
For a better insight into Wolfram's classes we put some representative pictures for each class in the Fig. 2.4 and 2.5.



(a) Rule R96

(b) Rule R10

Figure 2.4: Examples of CA from various behavioral classes introduced by S. Wolfram. (a) Trivial evolution of the class **c1** rule. (b) after a few steps became the pattern quite trivial and easy to predict. This CA belongs to the class **c2.**



(a) Rule R90

(b) R110

Figure 2.5: Examples of CA behavioral classes (a) Pattern appears highly chaotic; it is almost impossible to predict the CA state without explicitly running the simulation. Therefore, CA in is of the class **c3**. (b) example of the **c4** class: evolution is hard to predict. However, locally some structures propagate across the lattice and interact with each other.

2.3.2 Computational universality of CA

As we have already noted, there exist CAs capable of *universal computing.* For a proper explanation of this term we need to introduce the concept of Turing machine.

Turing's machine (TM) is a theoretical (though practically realizable) device that manipulates symbols on a strip of tape according to a prescribed rules [30]. The latter are specified in the so-called "Rules table". Despite its simplicity, TM can be made so that it simulates the logic of any computer algorithm – proof of this is not easy but it was proved in 1936 by Alan Turing [8]. TM is particularly useful in explaining the functions of a central processor inside computers. There are, however, two critical assumptions in TM, namely that both the time of calculation and the storage space for data are not limited in their size.

Since TM itself is a computer, it can be simulated by another TM. If TM is able to simulate any other Turing machine, it is called a **universal Turing machine** – UTM. Also the proof of UTM existence is quite complicated; firstly it was proved by Alonzo Church [31] in 1936.

In fact, all modern computers are based on the notion of UTM. Given the right instructions, and sufficient memory, any present computer (e.g., the one found in your fridge) could, for example, simulate a whole operating system. The fact that it might be ridiculous to waste time using fridge computers to do anything other than what their were designed for is irrelevant – the point is that they obey the same model of computation as every other computer and can therefore achieve the same result.

Since CA is a logical system, which may simulate other logical systems, it could happen, that there exists a CA which is able to simulate UTM. Such a CA is called **universal cellular automaton**. First universal CA was formulated by John von Neumann. He constructed (see, e.g., [32]) a two-dimensional CA with four cell neighborhood (Fig. 2.2a) and with cells of 29 cell states, that would be capable of simulating a UTM for certain configuration of about 200000 non-zero cells.

Today, a lot of CA rules are known to be universal. The most famous are, of course, the Game of Life, but the one with the most simple evolution rule is (according to Wolfram's notation) the $(r = 1, k = 2)$ rule $R110$ (see Fig. 2.5b).

2.4 Methods of CA analysis

This section describes various statistical methods of CA analysis. Substantially more extensive review of such methods can be found e.g., in Illachinsky's monograph [3]. Here, we recall only a necessary notation that will be used in the next chapter.

2.4.1 Density

A *density* is the simplest quantity characterizing configuration of CA at a time t. Assuming periodic lattice with finite number of unique cells, the density is defined as a fraction

$$\rho\left[\sigma\right](t) = \frac{\sum_{\vec{i}} \sigma_i(t)}{\sum_{\vec{i}} 1}, \tag{2.11}$$

where $\vec{i}$ goes over the period. The denominator stands for the number of all cells in one period.

Though an apparent simplicity, the *density* is an important quantity describing a CA thermodynamic behavior. If the density of cells at a long time scales tends to different value than 1/2 (assuming CA with two possible cell states, $\Sigma = \{0, 1\}$), than entropy tends to a non-extremal value and the system can not be ergodic (for equiprobable measure).

2.4.2 Two point correlation

Correlation between individual point is particularly important and easily measurable statistical quantity. Let us first define the two point correlations for two – states CA. Usually, *spin–like* cell ζ states are used instead of σ; where

$$\zeta_i = \begin{cases} +1 & \text{if} \quad \sigma_i = 1 \\ -1 & \text{if} \quad \sigma_i = 0 \end{cases}$$

Definition 4. *A two point space correlation in a given time is defined by averaging over the space lattice.*

$$C_r = \langle \zeta_i(t)\zeta_{i+r}(t)\rangle_i - \langle \zeta_i(t)\rangle_i \langle \zeta_{i+r}(t)\rangle_i, \tag{2.12}$$

where $r \in \mathbb{N}^d$ is an index shift.

Similarly is defined a time correlation:

Definition 5. *A k – order time correlation of the i-th cell is defined by averaging over the time.*

$$C_r = \langle \zeta_i(t)\zeta_i(t)\rangle_t - \langle \zeta_i(t)\rangle_t \langle \zeta_{i+r}(t)\rangle_t, \tag{2.13}$$

where $k \in \mathbb{N}^d$ is an order of the time correlation.

The presence of both time and space correlations is the first indication of complex behavior. The time correlation between two states is of particular importance when studying a CA dynamics. In general, it contains information about Markovianity of the system (i.e., memory properties) [34]. CA exhibiting long time local pattern formation, or CA which are not ergodic, must have time–correlated steps (does not exhibit a pure random walk through all possible states). Time correlation can be therefore used as a simple indicator of suspicion of complex behavior.

2.4.3 Entropy

In what follows we will describe entropic methods as a mean of CA analysis. At first, we introduce here a concept of Shannon entropy.

Definition 6 (Shannon entropy). *Let* (Ω, p) *be a discrete probability space, where* $\Omega = \{\omega_1, \ldots, \omega_n\}$ *is a finite set of possible events, and each element has a probability* p_i*. The Shannon entropy of* Ω *is*

$$H(A) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{2.14}$$

A probability distribution p can be obtained from CA simulations in three possible ways.

1. Averaging over sufficiently long time - gives a probability distribution of each single cell.
2. Averaging over sufficiently large *area* - results in time-dependent probability distribution.
3. Averaging both over space and over time - now the probability distribution depends only on CA itself and on the initial conditions.

However, Shannon's entropy is syntactic, not semantic entropy. It measures only overall randomness and does not take into account further structures and patterns of CA. Therefore, it is helpful to introduce other kinds of entropies and complexity measures, which will manage to do this.

Definition 7 (Local Shannon entropy)**.** *Let us consider an arbitrary CA. Let $\mathcal{N}_i$ be a neighborhood of the i-th cell (distinguished from domain of definition of the local evolution rule). The local Shannon's entropy $H_{loc}(i)$ of the i-th cell is also evaluated according to equation* (2.14)*, with probabilities redefined as*

$$p_i = \frac{\textit{number of cells } \sigma \in \mathcal{N}_i \textit{ in the i-th state}}{\textit{number of cells in } \mathcal{N}}. \tag{2.15}$$

The last entropic measure which we will use in the next chapter is a $k-$order conditional Shannon entropy. It quantifies the uncertainty of the state at the time t, which remains when the state of the system at the time $t-k$ is known. This measure is useful when studying non-Markovian processes.

Definition 8 (Conditional *k-order* Shannon entropy)**.** *Conditional k-order Shannon entropy of given CA at time t is [35]*

$$H(\sigma(t)|\sigma(t-k)) = -\sum_{i=1}^{n} p(i,t-k) \sum_{j=1}^{n} p(j,t|i,t-k) \log_2 p(j,t|i,t-k). \tag{2.16}$$

The meaning of this entropic measure (already mentioned above – remaining entropy in case that the state at the time $t-k$ is known) may be clarified by a following equation, which can be derived directly from the definition 2.16 (see e.g. [35]).

$$H(\sigma(t)|\sigma(t-k)) = H(\sigma(t), \sigma(t-k)) - H(\sigma(t-k)),$$

where $H(\sigma(t), \sigma(t-k))$ is the joint entropy of the two states $\sigma(t)$ and $\sigma(t-k)$.

Chapter 3

Statistical properties of ERCA

In this chapter, we analyze statistical properties of selected 2-dimensional ERCA with the help of methods described in the previous chapter. We will also mention some of their theoretical properties.

3.1 Ergodic theory of cellular automata

Since CA simulation is nothing than an interaction of a huge amount of interacting cells, our attention may be turned to statistical properties of CA evolution, namely, we will concern ourselves with ergodicity of elementary reversible rules (defined on the page 8).

While in a conventional statistical physics there is a natural state-ensemble measure – phase-space measure (also known as Liouville measure), no such natural measure exists in general for CA. Hence, we have to recall an appropriate notion of probability space and give some definitions allowing us to study the ergodicity with means of mathematics.

Probability space $(P, \mathcal{F}, \mu)$ consists of

$P \ldots$ the set of all CA states (set of elementary events),

$\mathcal{F} \ldots$ is a $\sigma-$algebra, where $\mathcal{F} \subset 2^P$,

$\mu \ldots$ is a probability measure $\mu : \mathcal{F} \to [0, 1]$.

Definition 9. *Let $(P, \mathcal{F}, \mu)$ is a probability space. F is a measurable map $\Phi : P \to P$. Than Φ is measure-preserving, if* $(\forall A \in \mathcal{F})\,(\mu(\Phi^{-1}(A)) = \mu(A))$.

Finding an appropriate measure in whose framework we can ensure ergodicity of a transition rule Φ is an important result of this chapter. In fact, generally can exist (and in mathematical ergodic theory invariably do exist) various different measures that can do the job. We will stress this point again at the later stage of the present chapter.

Let us first define ergodicity of a transition rule Φ. Intuitively, CA are said to be ergodic if they can pass in the course of their time evolution (which might be very long) through all possible feasible configurations. This definition deserves some specification. Mainly, the definition should include dependence of ergodicity on the probability measure:

Definition 10. *Let Φ be measure preserving transformation on P. Than Φ is ergodic if*

$$(\forall A)\left(\mathcal{F}\Phi^{-1}(A) = A\right) \Rightarrow (\mu(A) = 1 \vee \mu(A) = 0)\,.$$

The most important is the fact, that ergodicity is supposed to identify large time averages with state ensemble averages. This is a tenor of the Ergodic theorem:

Theorem 2 (Ergodic theorem [36])**.** *Let Φ be measure-preserving and ergodic on P. Then for $\forall g : P \to \mathbb{R}$ satisfying*

$$\int |g(x)| \mathrm{d}\mu(x) < +\infty\,,$$

we have

$$\lim_{n\to\infty} \frac{\sum\limits_{k=0}^{n-1} g\left(\Phi^k \omega\right)}{n} = \int g(x) \mathrm{d}\mu(x)$$

for all ω except a set of measure zero.

The following theorem gives a necessary and sufficient condition to ensure that Φ is not ergodic. This theorem is taken from [37]. While majority CAs are not ergodic, this theorem can serve as simple way how to show it.

Theorem 3. *Let* $(P, \mathcal{F}, \mu)$ *is a probability space, and* Φ *is a transformation preserving the measure* μ*. Than, the transformation* Φ *is not ergodic if and only if there exist an invariant set* S*, invariant measures* μ_1, μ_2*, such that for certain* $\lambda \in (0, 1)$

$$\mu = \lambda\mu_1 + (1 - \lambda)\mu_2,$$

and

$$\mu_1(S) = 1, \quad \mu_2(S) = 0,$$

Proof. See for example [37]. □

Generally, despite simplicity of CA, the ergodicity of an ergodic CA is not easy to prove. There exists no general recipe how to do it. Nevertheless, some theoretical results concerning special CA classes can be found in literature, e.g., in Refs. [38, 36, 39].

3.2 Ergodicity of ERCA

Let us now turn our attention to the ergodic properties of ERCA defined by the equation 2.8. The most important object of the ergodic theory is Ergodic transformation

$$\Phi : P \to P$$

(see definition 10). However, ERCA are of second order in time, which means that the global evolution rule is

$$\Phi : P \times P \to P.$$

Hence, strictly speaking, the global evolution rule of ERCA can not be considered as an ergodic rule. However, in sense of the Theorem 1 the ERCA can be equivalently viewed as a transition rule between two-states

$$\tilde{\Phi} : P \times P \to P \times P.$$

Let us denote $\tilde{P} \equiv P \times P$. Now, it makes sense to find (for given $\tilde{\Phi} : \tilde{P} \to \tilde{P}$) a measure μ and the sigma algebra $\mathcal{F}$, such that the mapping $\tilde{\Phi}$ is ergodic transformation.

Before analyzing ergodic properties of ERCA we will remark on some elementary properties of global transformation $\tilde{\Phi}$.

Remark 3. Let $\tilde{\Phi}$ is a global evolution rule of ERCA. Than, $\tilde{\Phi}$ is a bijection.

Proof. Inverse mapping of $\tilde{\Phi}$ can be derived using the equation (2.9). Hence, $\tilde{\Phi}$ is bijective mapping. □

Remark 4. Assuming Born–von Carman periodic boundary conditions on, WLOG, two-dimensional lattice, the number of possible distinct states of CA is necessarily finite and is equal to the number 2^{MN}, where M, N are the periods of the lattice. Therefore, an arbitrary initial conditions result into a periodic evolution.

Remark 5. There is no *Garden of Eden*, i.e. each two-cell $\tilde{\sigma}$ has its predecessor.

Proof. The existence of a predecessor can be proved with the help of the equation (2.9), or using following argument: Let us suppose, on contrary, that there is a state $\tilde{\sigma}$, such that for $\forall \tilde{\sigma}^1 \in P, \tilde{\Phi}\tilde{\sigma}^1 \neq \tilde{\sigma}$. From the last Remark 4 follows, that for each initial condition there exist a evolution period N, such that

$$\tilde{\Phi}^N \tilde{\sigma} = \tilde{\sigma}.$$

Thus, the predecessor of the state $\tilde{\sigma}$ is $(\tilde{\Phi}^{N-1}\tilde{\sigma})$. □

From the previous remarks follows that the state-space $\tilde{P}$ of each ERCA is divided into several cycles - set of states that are periodically visited during the simulation. Depending on initial conditions, the simulation starts in a certain cycle and never reaches a state of a different cycle.

Definition 11 (ERCA cycle)**.** *A set $C \subset \tilde{P}$ is called a cycle of ERCA $\tilde{\Phi}$ if there exist a finite period N, so that and*

$$(\forall \tilde{\sigma} \in C)(\tilde{\Phi}^N \tilde{\sigma} = \tilde{\sigma})$$
$$(\forall \tilde{\sigma}_1 \tilde{\sigma}_2 \in C)(\exists M \leq N)(\tilde{\Phi}^M \tilde{\sigma}_1 = \tilde{\sigma}_2)$$

Two distinct probability spaces $(\tilde{P}, \mu_1, \mathcal{F}_1)$, resp. $(\tilde{P}, \mu_2, \mathcal{F}_2)$, are considered, where $\mathcal{F}_1 = 2^{\tilde{P}}$ –set of all subsets of $\tilde{P}$; and $\mu_1(\sigma) = \frac{1}{n_1}$, where n_1 is cardinality of $\tilde{P}$. Let $C \subset \tilde{P}$ be a cycle of cardinality n_2. The letter sigma algebra is also $\mathcal{F}_2 = 2^{\tilde{P}}$; the measure μ_2 is non-zero only for $\tilde{\sigma} \in C$:

$$\mu_2(\tilde{\sigma}) = \begin{cases} 1/n_2 & \text{if } \tilde{\sigma} \in C \\ 0 & \text{if } \tilde{\sigma} \in P \setminus C \end{cases}$$

Remark 6. If given ERCA consists of more than one cycle, automatically, it is not ergodic with respect to the probability space $(\tilde{P}, \mu_1, \mathcal{F}_1)$.

Proof. To prove this, let us suppose, on contrary, that $\tilde{\Phi}$ preserves the measure μ_1 and is ergodic. The probability space can be decomposed into two sets

$$\tilde{P} = C \cap (\tilde{P} \setminus C)\,,$$

where $C \subset \tilde{P}$ is a cycle. Similarly, the probability measure μ_1 can be decomposed as

$$\mu_1 = \frac{n_2}{n_1}\mu_2 + \left(1 - \frac{n_2}{n_1}\right)\left(\frac{n_1}{n_1 - n_2}\left[\mu_1 - \frac{n_2}{n_1}\mu_2\right]\right)\,. \tag{3.1}$$

The above equation can be verified by algebraical modifications of the right side. Since the set $\tilde{P} \setminus C$ is non-empty, an expression

$$\left(\frac{n_1}{n_1 - n_2}\left[\mu_1 - \frac{n_2}{n_1}\mu_2\right]\right)$$

is a *complementary* measure of μ_2, i.e., it is an uniform measure on $\tilde{P} \setminus C$ and is also invariant under $\tilde{\Phi}$. The equation 3.1 represents a convex decomposition of μ_1 for $\lambda = n_2/n_1$. Hence, according to the Theorem 3, the measure μ_1 is not ergodic. □

Theorem 4. *Each ERCA transformation $\tilde{\Phi}$ is measure preserving and ergodic with respect to the probability space $(\tilde{P}, \mu_2, \mathcal{F}_2)$.*

Proof. This can be proved directly from the definition 10. Suppose, that A is an invariant set, i.e. $\tilde{\Phi}^{-1}(A) = A$. Clearly, A can be written as an union of cycles C_i

$$A = \bigcup_i C_i$$

Using the fact that mu_2 is non-zero only on the cycle C, we obtain a relation for $\mu_2(A)$:

$$\mu_2(A) = \begin{cases} 1 & \dots \quad \text{if } C \subset \bigcup_i C_i \\ 0 & \dots \quad \text{if } C \subset P \setminus \bigcup_i C_i, \end{cases}$$

And therefore, $\tilde{\Phi}$ is ergodic. □

Note, that there are as many similar (and ergodic) measures as is the number of distinct cycles. However, there is an important drawback of the measure μ_2. The μ_2 is non-zero on the cycle C, which means that knowledge of ERCA trajectory is necessary when evaluating the measure. Hence, in fact, both sides of the equation (2) from the Ergodic theorem depend explicitly on the trajectory in time. This, unfortunately, substantially reduces the practical usefulness of ERCA ergodicity with respect to the measure μ_2.

3.3 Definition of simulated rules

Let us define three selected two–dimensional ERCA rules. In the text below, we will denote them **A, B** and **C**. In order to obtain greater variety of their behavior, we derive these ERCA rules from generic evolution rules of different Wolfram's behavioral classes.

Definition 12 (A). *A reversible **rule A** on Euclidean lattice with five cells neighborhood (see the Fig. 2.2a) is defined by the equation* (2.8)*, where local evolution rule φ is graphically represented in the Fig. 3.1.*

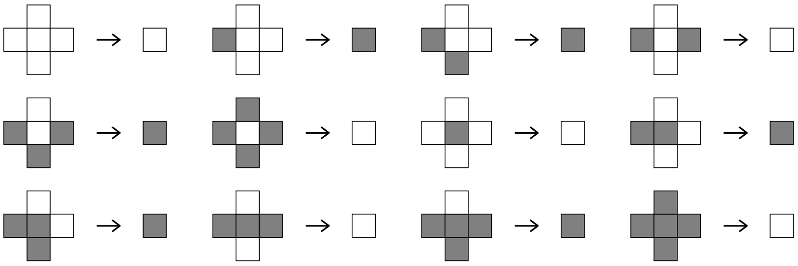


Figure 3.1: The local evolution rule A. This rule is invariant under 90-degrees rotation. This means, for instant, that the second diagram describes in fact evolution of four distinct neighborhood states. No matter whether the black is on the left, on the right, at the top or below; always, in that cases, the resulting cell will become black.

Definition 13 (B)**.** *Rule B is defined in the same way as A, but the local evolution rule φ is different – see the Fig. 3.2*

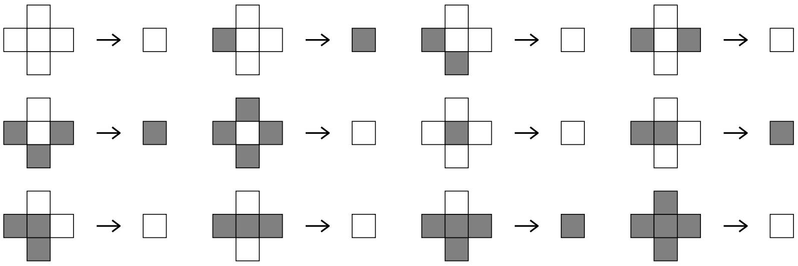


Figure 3.2: Local evolution rule of B. The symmetry of the local evolution rule is described in the caption of Fig. 3.1.

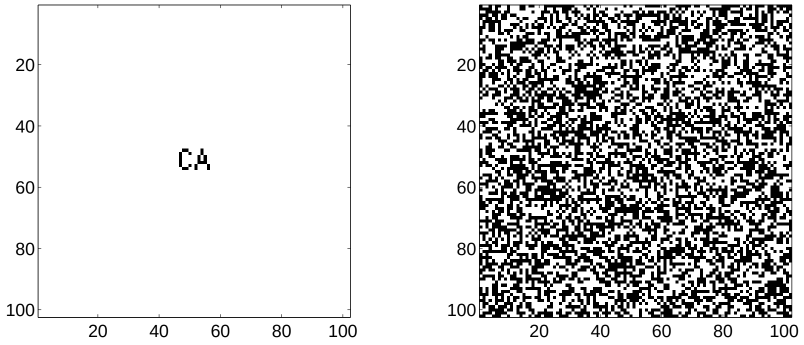
Definition 14 (C)**.** *The third rule is defined in the Euclidean space with a nine point neighborhood (see fig. 2.2b). We chose a local evolution rule φ as a well known Conway's Game of Life. The overall evolution is given by the equation* (2.8) *with the Conway's rule as the local rule φ.*

Each ERCA rule was derived from a generic rule of different behavioral class. The evolution of the rule 12 is of high randomness which place it into the class

c3. Rule 13 just replicate the original pattern, when the time is power of two [12], witch allows high predictability. Hence, **B** is class **c2**. At last, the Life is known to universal **c4** class.

3.3.1 Initial conditions

Here, we define initial conditions that will be mostly used in our simulations. The initial conditions consist of two consecutive states of CA, say $\sigma(0), \sigma(1)$. Note, that both slides $\sigma(0), \sigma(1)$ can be chosen completely arbitrarily. In our simulations, we always let the slide $\sigma(0)$ blank; as $\sigma(1)$ we choose one of the slides of the Fig. 3.3. The initial condition 3.3a is chosen so that the initial pattern is as simple as possible, but not symmetric. In the Fig. 3.3b, the initial condition is set as a state with maximum entropy, i.e. is completely random.



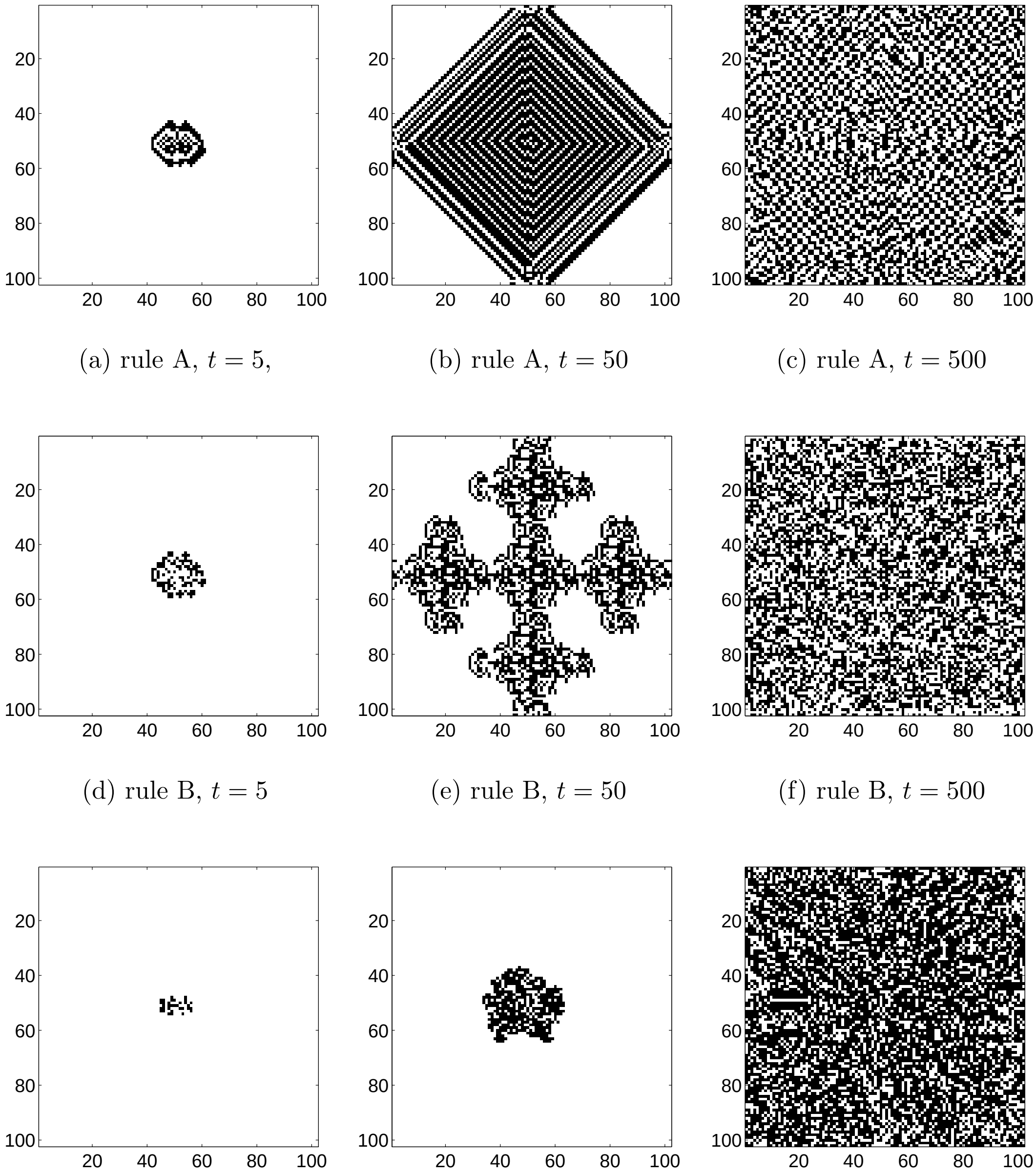
(a) Initial conditions "CA"

(b) Random initial conditions

Figure 3.3: Initial conditions. This Fig. shows only slides $\sigma(1)$, while $\sigma(0)$ are left blank.

3.4 Simulation results

The simplest analysis is just run and look at the ERCA evolution. Several snapshots evolved from the "CA" initial condition after, 5, 50 and 500 time steps are in the Fig. 3.4



(a) rule A, $t = 5$,

(b) rule A, $t = 50$

(c) rule A, $t = 500$

(d) rule B, $t = 5$

(e) rule B, $t = 50$

(f) rule B, $t = 500$

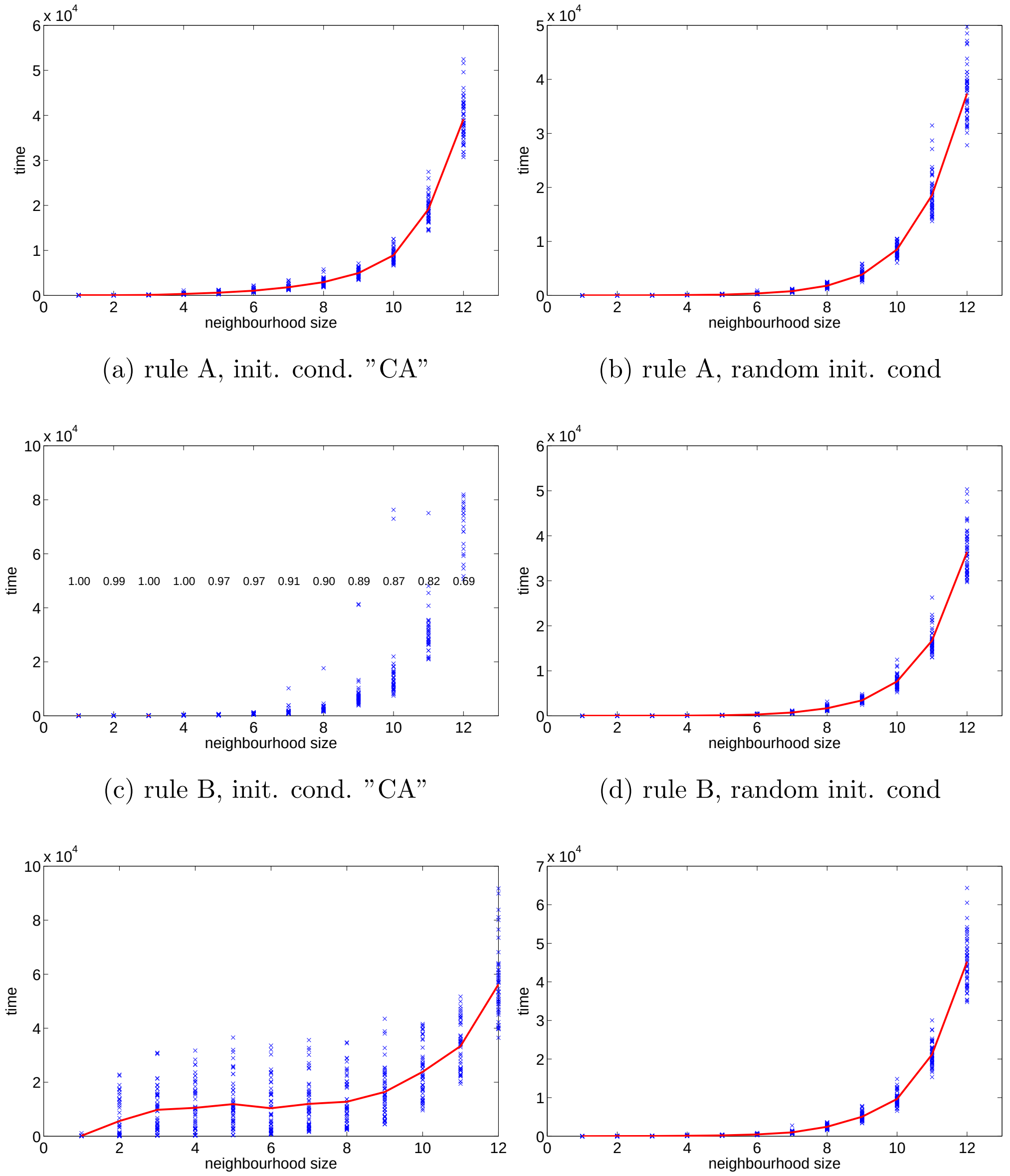
(g) rule C, $t = 5$

(h) rule C, $t = 50$

(i) rule C, $t = 500$

Figure 3.4: Evolution of rules A, B and C

The Fig. 3.4 shows, that all three create highly disordered pattern after some time. Nevertheless, all ERCA must after certain amount of time (probably very long) return to the initial state. We shall also note, that while the rule A perform high space correlation, the rule C is highly time correlated.



(a) rule A, init. cond. "CA"

(b) rule A, random init. cond

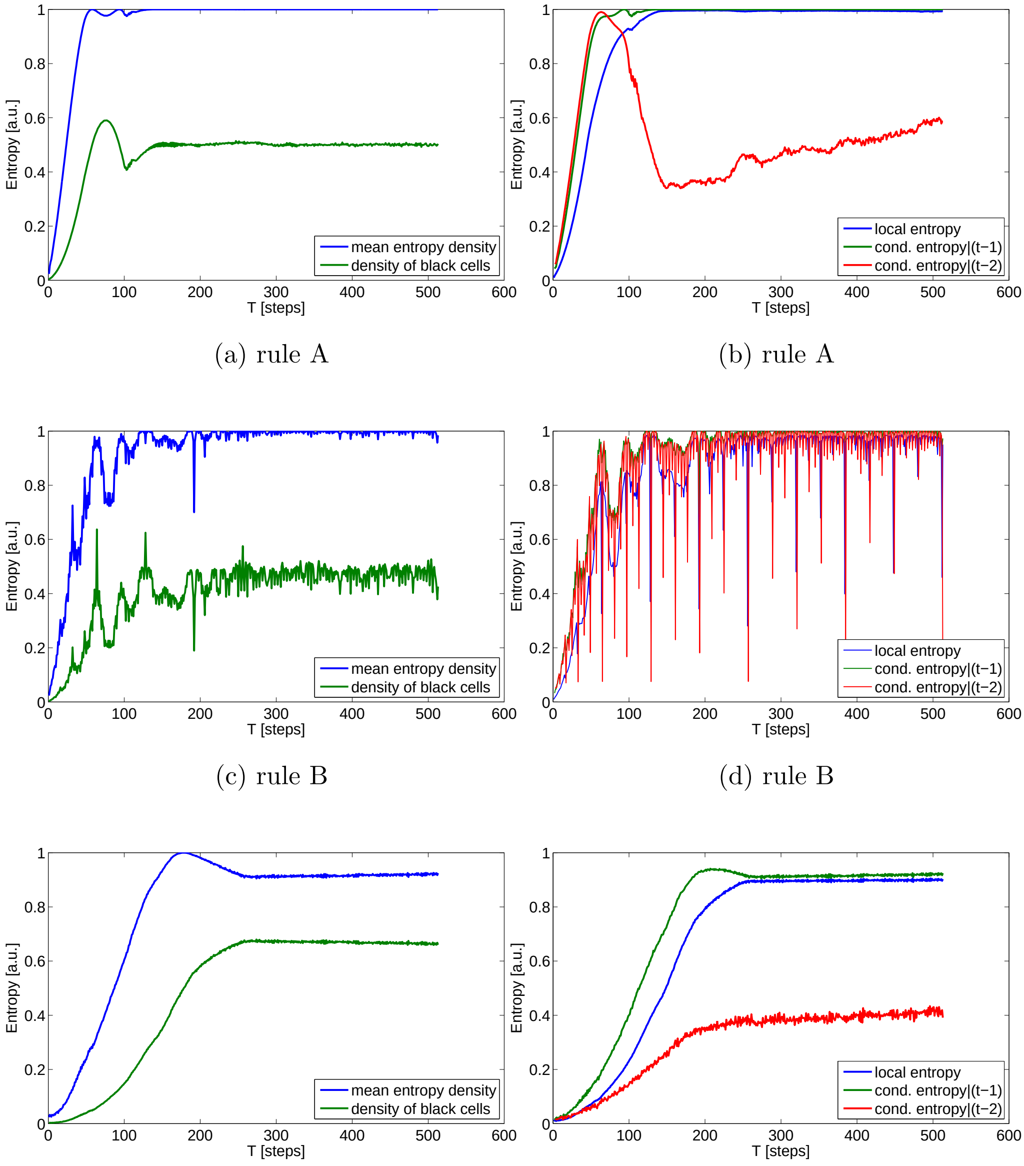
(c) rule B, init. cond. "CA"

(d) rule B, random init. cond

(e) rule C, init. cond. "CA"

(f) rule C, random init. cond

Figure 3.5: Analysis of local ergodicity. Number of time steps needed until CA goes through all states of the neighborhood of given size. 50 independent experiments were done for each neighborhood size. In case that some states were never reached, the average fraction of reached neighborhood states is displayed (picture (c)). Evolution of "CA" initial conditions (always on the left) is compared with that of random ones (on the right). Simulations ran on the 100×100 lattice. All simulation (except 3.5c) seem to confirm local ergodicity up to the size of neighborhood of 12.



(a) rule A

(b) rule A

(c) rule B

(d) rule B

(e) rule C

(f) rule C

Figure 3.6: We analyzed a time dependence of Shannon entropy, density of black cells (on the left), local and conditional entropy. For all entropy analysis we used the "CA" initial conditions. Simulations ran on the 100×100 lattice.

Though none of ERCA rule is ergodic with respect to the measure μ_1, many ERCA rules reach thermodynamic equilibrium – the measures are stabilized in time (see Fig. 3.6a 3.6e, 3.6f). We can show (see Fig. 3.5) that the ERCA rule may become ergodic, if we introduce classes of equivalence for ERCA states:

Definition 15. *The ERCA state $\sigma^{(1)}$ is equivalent with the state $\sigma^{(2)}$ with respect to the neighborhood $\mathcal{N}$ (distinct from rule neighborhood) if and only if*

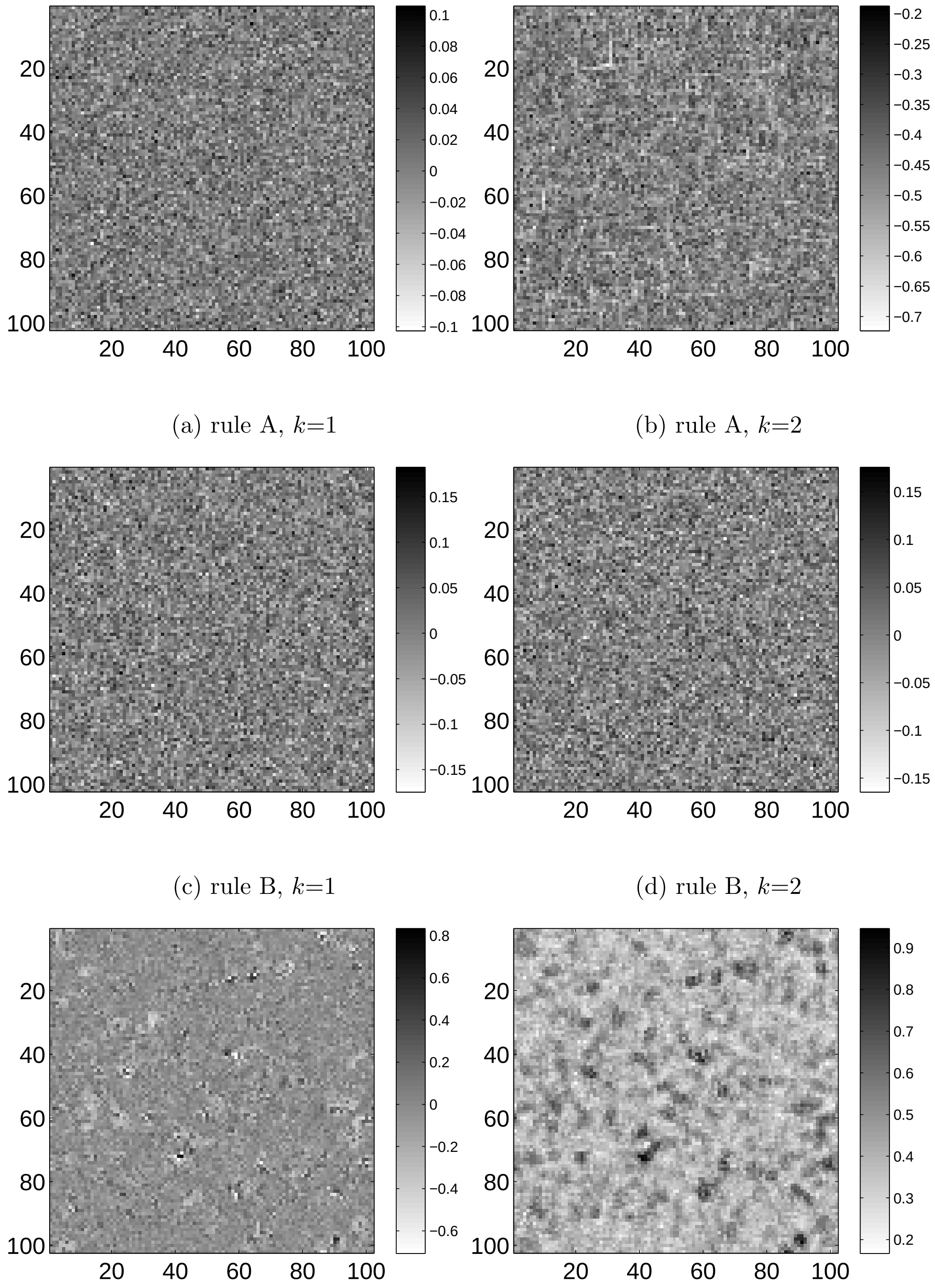
$$\left(\forall \vec{i} \in \mathcal{N}\right)\left(\sigma^{(1)}_{\vec{i}} = \sigma^{(2)}_{\vec{i}}\right)$$

We say, that the rule Φ is locally ergodic in the neighborhood $\mathcal{N}$, if it can pass in the course of their time evolution through all possible feasible classes of equivalence of $\mathcal{N}$. Simulations showed, that (except rule B with "CA" initial conditions) all rules are locally ergodic in neighborhoods up to the neighborhood size of 12 cells.

In the figure 3.6 is shown a time dependence of various entropic measures. All simulation started from "CA" initial condition and therefore have low initial entropy. After approximately 300 time steps the entropic measures often tends to stabilize at certain values (there are, however numerous exceptions).

Interesting behavior perform the rule B. When the time is equal to the power of two (and its multiple) the pattern jumps to a highly ordered state. In the next steps, it becomes highly chaotic again. This results in the *beats* of entropy dependence.

Finally the Figs. 3.7 and 3.8 show the time correlations of selected ERCA rules (averaged over 500 time steps). Note, that time correlations depend on both specific ERCA rule and initial conditions.



(a) rule A, k=1

(b) rule A, k=2

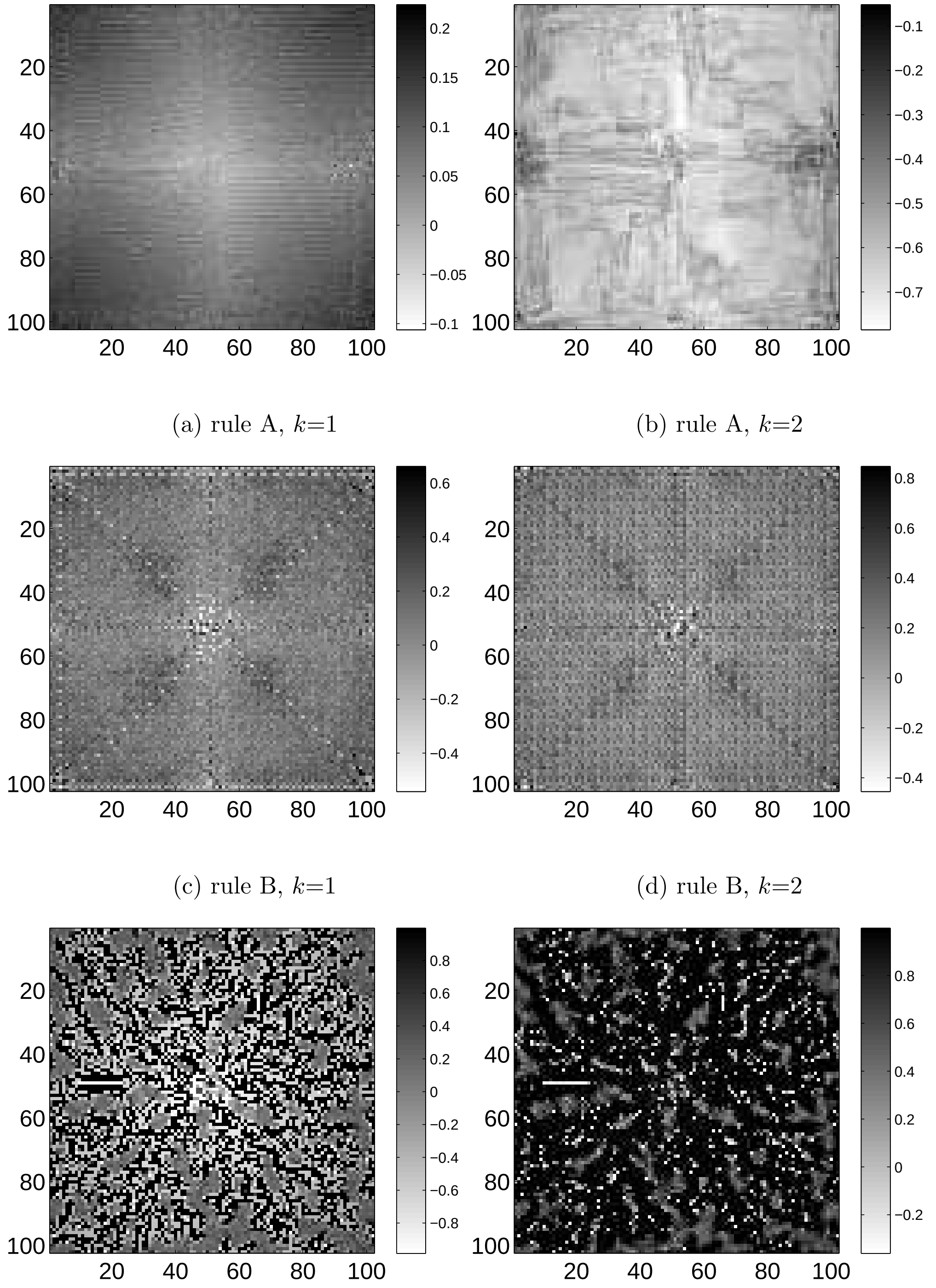
(c) rule B, k=1

(d) rule B, k=2

(e) rule C, k=1

(f) rule C, k=2

Figure 3.7: Time correlation analysis of rules A,B and C averaged over first 512 time steps, where k is the order of correlation. Starting form random initial condition. While rules A and B remain practically uncorrelated, the rule C creates local areas with high second order time correlation.



(a) rule A, k=1

(b) rule A, k=2

(c) rule B, k=1

(d) rule B, k=2

(e) rule C, k=1

(f) rule C, k=2

Figure 3.8: Time correlation analysis, same as in the Fig. 3.8, but starting with "CA" initial conditions.

Chapter 4

CA application in biology

CA can be used in simulation of biological system, provided that this system is made of large number organisms and behaviour of each organism is reducible to the finite number of possible responds. This assumption fulfilled if, for instant, observed object is rather simple (like unicellular organisms). However, more complex organisms could be simulated as well, if we consider only the most important behavioural characteristics. For example, the bird flocking was successfully simulated using CA [40].

In the natural world occur many forms of cooperation of live organisms. We will give several examples. Complex societies are created by number of insect species, among them the most remarkable are the societies of ant and bees [41].

In this chapter, we will describe simulations of ant movement and a spontaneous trail creation. In order to keep things as simple, as possible, only one ant caste (forager ants) is supposed. In other words we will simulate only ants, whose task is to find food an deliver it to the nest.

4.1 Ant trail simulation

Here, we will summary empirical evidences on ant's orientation and movement in the environment. Later, these facts will inspire us to formulate a simplified CA model. Forager ants searching for food usually lay down chemicals called *pheromones*, as they return to the nest [42, 43]. The pheromone trail is renewed as long as the supply hold out.

There are several mechanisms of ant orientation. It is suggested that ants remember number of steps they taken [44]. Also, they have a visual memory of surrounding objects if available [45]. The return to the nest is also simplified by position of the Sun [46] and gravity vector evaluation. Moreover, some species are able to orient to Earth's magnetic field [47]. It can be said, that any ant knows the direction and the distance to its nest.

4.2 One dimensional model

The ant trail simulation on one-dimensional lattice was published in detail in the twelfth chapter of the book [48]. Here, we will follow this definition.

Definition: The ants live on the one-dimensional ring of size N, i.e. in terms of CA, there is N different cells $\{\sigma_i\}_{i=1}^{N}$ with a boundary condition $\sigma_{N+1} = \sigma_1$. Each cell may, or may not be occupied by ant and by a pheromone droplet. Thus, it is convenient to write a cell state in the vector form

$$\sigma_i = \begin{pmatrix} a \\ p \end{pmatrix},$$

where

$$a = \begin{cases} 1 \text{ when ant is present} \\ 0 \text{ when ant is not present} \end{cases} \qquad p = \begin{cases} 1 \text{ when pheromone is present} \\ 0 \text{ when pheromone is not present} \end{cases}.$$

Evolution rule: We will follow the formalism of mentioned publication [48]. In this model, all ants are allowed to move in the same direction. The probabilistic evolution rule is illustrated in the figure 4.1.

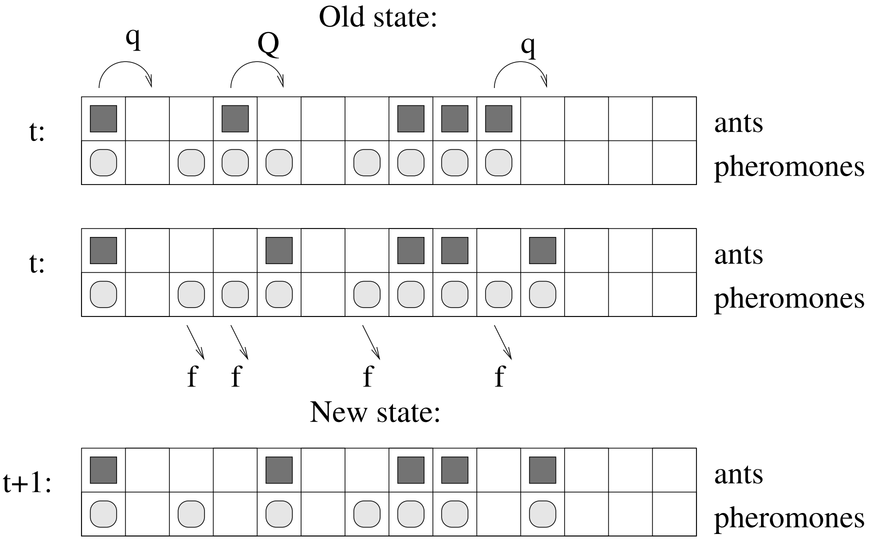


Figure 4.1: Ant trail in 1D: evolution rule.

Updating cells proceeds in two stages: 1) Motion of ants; 2) Evaporation of pheromones.

1. **Motion of ants** The ant at the $i - th$ position moves forward with probability p_i

$$p_i = \begin{cases} Q & \text{if} \quad \sigma_{i+1} = \binom{0}{1} \\ q & \text{if} \quad \sigma_{i+1} = \binom{0}{0} \\ 0 & \text{if} \quad \sigma_{i+1} = \binom{1}{1} \vee \sigma_{i+1} = \binom{1}{0}. \end{cases}$$

2. **Evaporation of pheromones** Each ant creates a pheromone in it's position after stage 1). This means:

$$\sigma_i(t) = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \vee \sigma_i(t) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \Rightarrow \sigma_i(t+1) \equiv \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Free pheromones (with no ant in same cell) will evaporate with probability f:

$$\sigma_i(t) = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \Rightarrow \sigma_i(t+1) \equiv \begin{cases} \binom{0}{0} \text{ with probability } f \\ \binom{0}{1} \text{ with probability } (1-f) \end{cases}$$

When starting from random initial condition (each cell with *ant probability* = *pheromone probability* = 1/3), for parameters $Q = 0.75, q = 0.25, f = 0.005$, thick clusters propagating with probability q emerges (see Fig. 4.2).

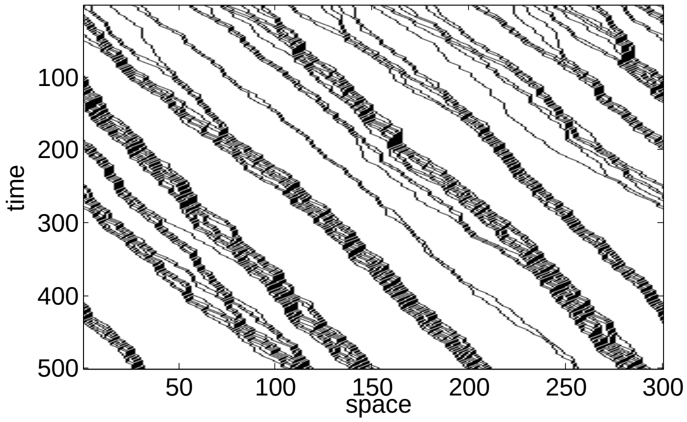
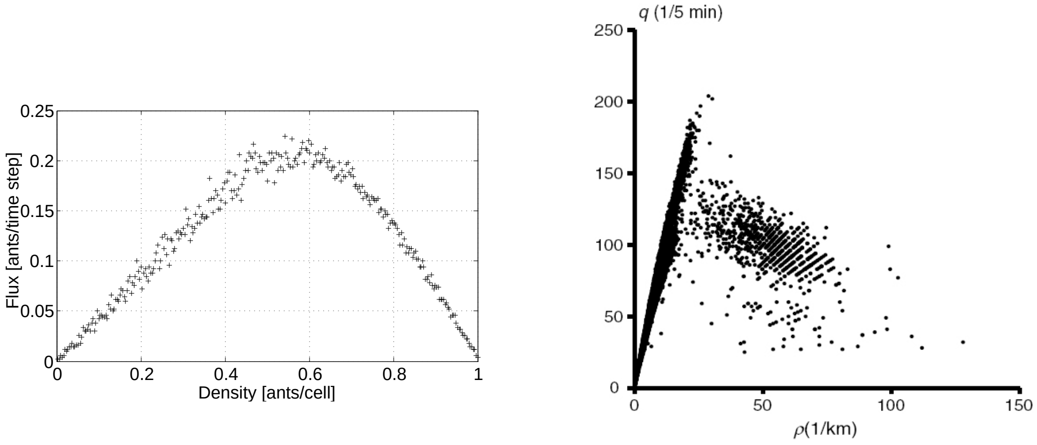


Figure 4.2: Fifty ants on a periodic one-dimensional trail starting from random initial condition. After a few steps clusters moving forward with an average speed q. All ants join together into one cluster after time of order 10^5 steps.



(a) ant trail

(b) road traffic (real experiment)

Figure 4.3: Fundamental diagram of ant movement on 1-dimensional trail. Model parameters were set to: $Q = 0.8, q = 0.4, f = 0.05$. Compared with the real road traffic experiment (taken from [49])

Fundamental diagram of unidirectional ant movement is very similar to that of highway transport [50].

4.3 Two dimensional model

In this section we would like to simulate the ant trail formation between a food source and an ant's nest. One dimensional model and it's behaviour inspired us to make an extension in two dimensions. This leads to additional problems which have to be solved. For instant, somehow, orientation of ants and the pheromone trail formation have to be defined.

Here, we give a list of requirements, that should our model obey. These assumptions are inspired by the empirical facts mentioned in the section 4.1.

1. Every ant knows the position of the nest.
2. Maximum one ant per cell is allowed.
3. The ants can carry an item of food.
4. Motion of ants is brownian.
5. Ants carrying food deliver it to the nest.
6. Ants searching for food prefer direction where pheromone is more concentrated. Also, the pheromone stimulate ants to go further from their nest.
7. Ants delivering food to the nest drop down pheromone.
8. A pheromone trail diffuse to the environment and slowly evaporate, unless renewed by other ants.

We will describe the model more technically.

4.3.1 Space definition

The environment is made of three distinct arrays of size $N \times N$; let denote them $\mathcal{A}, \mathcal{O}$ and $\mathcal{P}$, where $N \in \mathbb{N}$ is size of the arrays; $\mathcal{A}, \mathcal{O}$ and $\mathcal{P}$ contain information about ants position, objects position (like the nest and food) and pheromone distribution respectively. Following equations (4.1),(4.2) and (4.3) explain more exactly the meanings of arrays $\mathcal{A}, \mathcal{O}$ and $\mathcal{P}$.

$$\mathcal{A}_{i,j} = \begin{cases} -1 \ldots \text{ ant with food at the position } (i,j) \\ 0 \ldots \text{ no ant} \\ 1 \ldots \text{ ant without food} \end{cases} \tag{4.1}$$

$$\mathcal{O}_{i,j} = \begin{cases} -1 \ldots \text{ nest cells at the position } (i,j) \\ 0 \ldots \text{ no object at this position} \\ 1 \ldots \text{ food supply} \\ 2 \ldots \text{ forbidden area} \end{cases} \tag{4.2}$$

$$\mathcal{P}_{i,j} \in \{0, 1, 2, \ldots, 255\} \ldots \text{ represents pheromone concentration.} \tag{4.3}$$

There are fixed boundary conditions instead of periodic ones in this model. We consider such a choice as more natural world for ants.

4.3.2 Pheromone diffusion

A discrete approximation of continuous diffusion equation is used in description of pheromone spreading into the environment. The classical diffusion equation can be written as

$$\frac{\partial \varphi(r,t)}{\partial t} = K \Delta \varphi(r,t), \tag{4.4}$$

where K is a diffusion coefficient. For grid size h the discrete approximation of Laplace operator is

$$\Delta \varphi(x,y) \simeq \frac{\varphi(x+h,y) + \varphi(x-h,y) + \varphi(x,y-h) + \varphi(x,y+h) - 4\varphi(x,y)}{h^2}$$

For our purposes we take the grid size h equal to one. Therefore, we get for arbitrary discrete array $\mathcal{P}$ following definition of discrete Laplacian Δ_d:

$$\Delta_d \mathcal{P}_{i,j} \equiv \mathcal{P}_{i+1,j} + \mathcal{P}_{i-1,j} + \mathcal{P}_{i,j-1} + \mathcal{P}_{i,j+1} - 4\mathcal{P}_{i,j}$$

Using this, the diffusion rule including evaporation can be written in two steps (4.5) and (4.6)

$$\tilde{\mathcal{P}}(t) \equiv C(\mathcal{P}(t) + K\Delta_d \mathcal{P}(t)) \tag{4.5}$$

$$\mathcal{P}(t+1) \equiv \min(255, \max(0, \lfloor \tilde{\mathcal{P}}(t) \rfloor)), \tag{4.6}$$

where $C \in (0,1)$ is evaporation constant, and $\lfloor x \rfloor$ denotes the largest previous integer of x. Note that while equation (4.5) contains whole dynamics, the latter equation (4.6) only keeps the array $\mathcal{P}$ integer non-negative.

4.3.3 Ant motion

The motion of ants has two phases:

1. Decision where to jump
2. If chosen cell is free, jump there. Otherwise something else may happen.

Ad 1) Let an ant is at a position $\vec{x} \equiv (i,j)$. Then, the ant chooses a new position as a realization of a discrete random variable X.

$$X = \mathrm{round}(F), \tag{4.7}$$

where F is normally distributed variable in $\mathbb{R}^2$ with a mean value vector $\vec{x} + a\vec{n}$ and a diagonal covariance matrix $\sigma^2\mathbb{I}$. Here, $\vec{n}$ denotes a unit vector to the certain direction, a and $\sigma^2 \in \mathbb{R}$ are parameters. From a physical point of view the first parameter a is proportional to the external accelerating field, whereas σ^2 is proportional to the temperature of the Brownian particles ($m\langle \frac{\mathrm{d}x}{\mathrm{d}t}^2 \rangle = k_B T$).

$$F \sim \mathcal{N}(\vec{x} + a\vec{n}, \sigma^2\mathbb{I}) \tag{4.8}$$

Ants prefer cells with higher concentration of pheromone. Therefore, ants need to measure discrete gradient of pheromone concentration $\vec{\nabla}_d \mathcal{P}$, where

$$\begin{aligned}(\vec{\nabla}_d \mathcal{P}_{i,j})_x \equiv& \frac{1}{2}\left(\mathcal{P}_{i+1,j} + \mathcal{P}_{i-1,j}\right)\\ (\vec{\nabla}_d \mathcal{P}_{i,j})_y \equiv& \frac{1}{2}\left(\mathcal{P}_{i,j+1} + \mathcal{P}_{i,j-1}\right).\end{aligned}$$

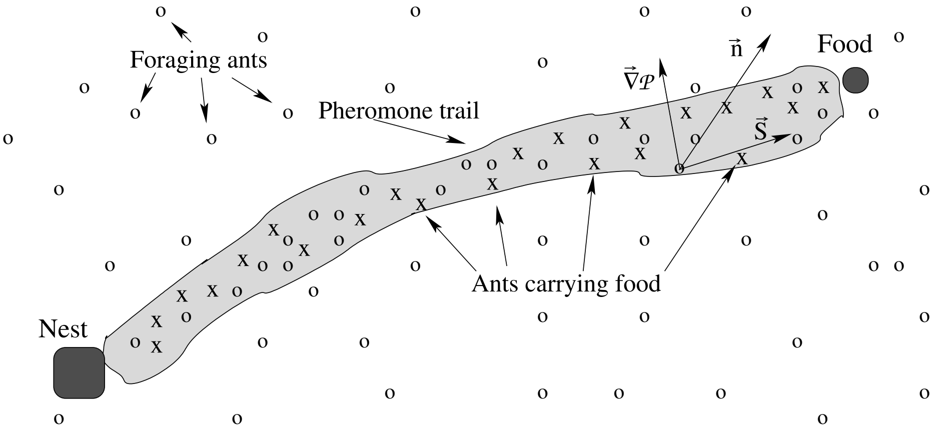


Figure 4.4: Scheme of 2D ant trail. Freely moving foragers search for food. At the same time, they are attracted by pheromone (gray), which help them to find the direction. Loaded ants (marked as x) are delivering the food to their nest.

Depending on whether an ant carry food or not is the direction vector $\vec{n}$ determined as

$$\vec{n} = \begin{cases} c_1(\vec{G} - \vec{S}) \dots \text{ if ant carry food} \\ c_2(\vec{G} + \vec{S}) \dots \text{ if search for food} \end{cases}, \tag{4.9}$$

where c_1, c_1 are multiplicative constants so that $\vec{n}$ is unit vector and

$$\begin{aligned}\vec{S} =& \frac{\vec{x} - \vec{x}_{nest}}{\|\vec{x} - \vec{x}_{nest}\|} = \text{ The unit radius vector of the current position.}\\ \vec{G} =& \begin{cases} \frac{\vec{\nabla}_d \mathcal{P}}{\|\vec{\nabla}_d \mathcal{P}\|} \dots \text{ if } \vec{\nabla}_d \mathcal{P} \neq 0 \\ \text{random unit vector} \dots \text{ else;} \end{cases}\end{aligned}$$

Remaining constant a and σ^2 can be chosen freely as a parameters of the model. Usually, we choose them close to one.

The whole situation is illustrated in the picture 4.4.

Ad 2) Let suppose, that an ant is at the position $\vec{x}$ and we got a vector $\vec{y}$: the realisation of the random variable X (from equation 4.7). We will describe all possibilities that may happen.

$$\begin{aligned}
\mathcal{A}(\vec{y}) = \mathcal{O}(\vec{y}) = \mathcal{P}(\vec{y}) = 0 \quad &\longrightarrow \begin{cases} \mathcal{A}(\vec{y}) \equiv \mathcal{A}(\vec{x}) \\ \mathcal{A}(\vec{x}) = 0 \\ \mathcal{P}(\vec{y}) = \max(\mathcal{P}(\vec{y}), -\operatorname{sgn}(\mathcal{A}(\vec{x}))) \end{cases} \\
\mathcal{O}(\vec{y}) = +1 \wedge \mathcal{A}(\vec{x}) = +1 &\longrightarrow \quad \mathcal{A}(\vec{x}) = -\mathcal{A}(\vec{x}) \\
\mathcal{O}(\vec{y}) = -1 \wedge \mathcal{A}(\vec{x}) = -1 &\longrightarrow \quad \mathcal{A}(\vec{x}) = -\mathcal{A}(\vec{x}) \\
\text{Otherwise (position } \vec{y} \text{ inaccessible)} &\longrightarrow \quad \text{No action of the ant.}
\end{aligned}$$

4.3.4 Formulation of the rule

Finally, we have to say is how to put previous sections together. Let do it.

1. **Initial condition.** We will sum up all objects to be define at the beginning of the simulation. These are arrays $\mathcal{O}, \mathcal{P}, \mathcal{A}$. Array $\mathcal{O}$ defining the object is static during the simulation. Initial ant position (encoded in $\mathcal{A}$) is usually random distribution of given number of ants without food. $\mathcal{P}$ is set to the zero. Furthermore, evaporation and diffusion coefficients C, K have to be set up.

2. **Iterative simulation** Similarly to the one-dimensional model proceeds the iterations in two stages in following order:

   (a) **Updating ant's positions** Described in the section 4.3.3

   (b) **Diffusion and evaporation of the pheromone** Described in the section 4.3.2

4.3.5 Results

In all simulations presented here are diffusion and evaporation coefficients C and K from equation 4.3.2 fixed to $C = 0.999$ and $K = 0.1$. The first figure shows the effect of pheromone (Fig. 4.5). Naturally, the pheromone trail significantly increases ants efficiency.

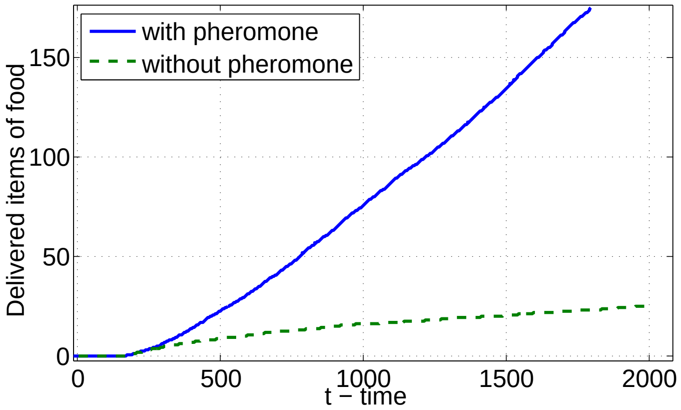
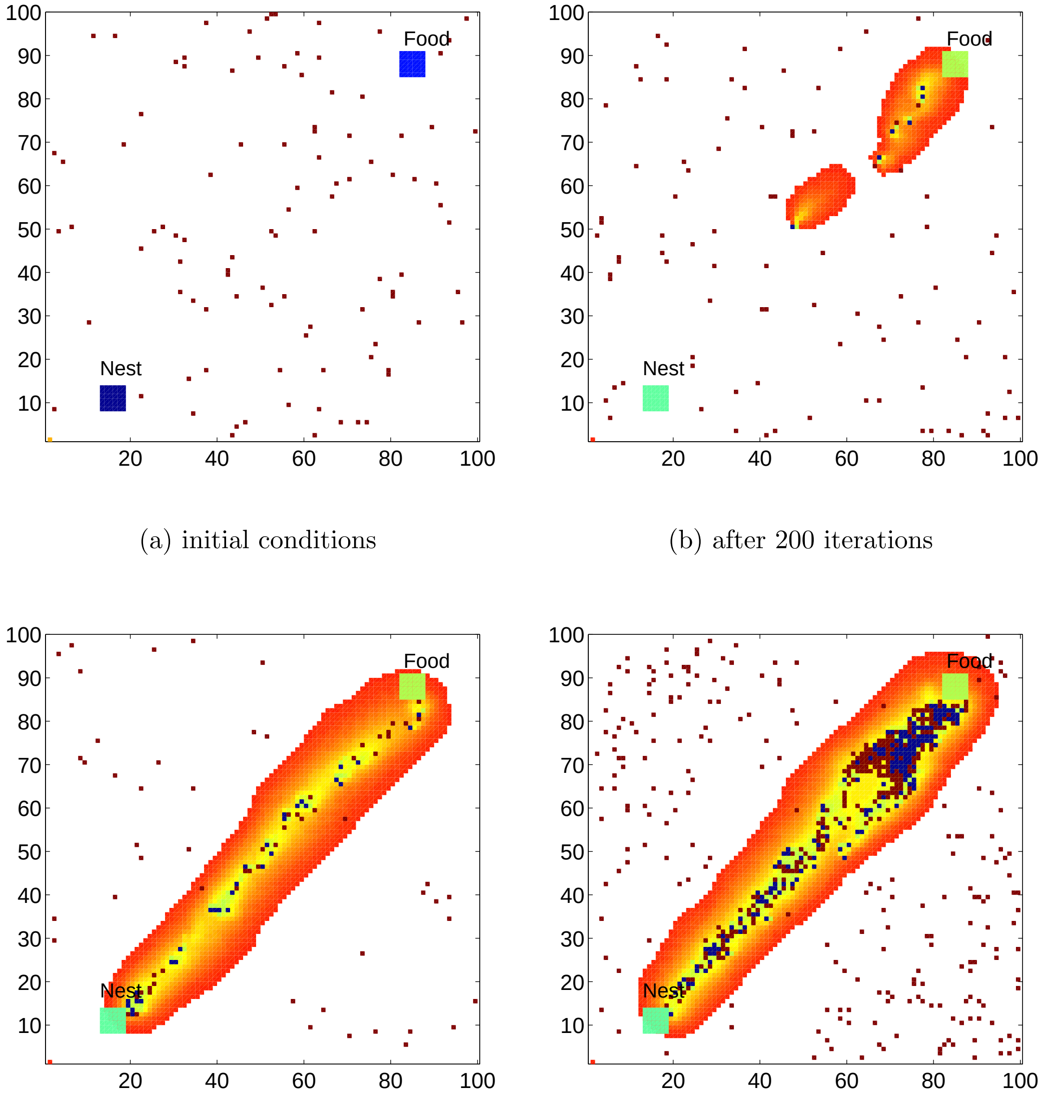


Figure 4.5: Dependence of delivered items of food on the time. Comparison of the model with pheromone with the same situation, but without pheromone. Simulation ran on in the grid of a size 100×100 points, with 200 ants; parameters $a = 0.6$, $\sigma^2 = 0.6$.

We executed our model in the grid of a size 100×100 points, with 100 ants; parameters $a = 0.6$, $\sigma^2 = 0.6$. Within several steps some ant found the source of food and creates pheromone trail (see figure 4.6).

In the Fig. 4.7 is shown ant trail formation in the environment with square obstacle placed symmetrically between the nest and the source of food. Ants randomly chose one of two equivalent paths. Sometimes (approximately each 1000 time steps) ants change their mind and by mutual agreement chose the opposite path.



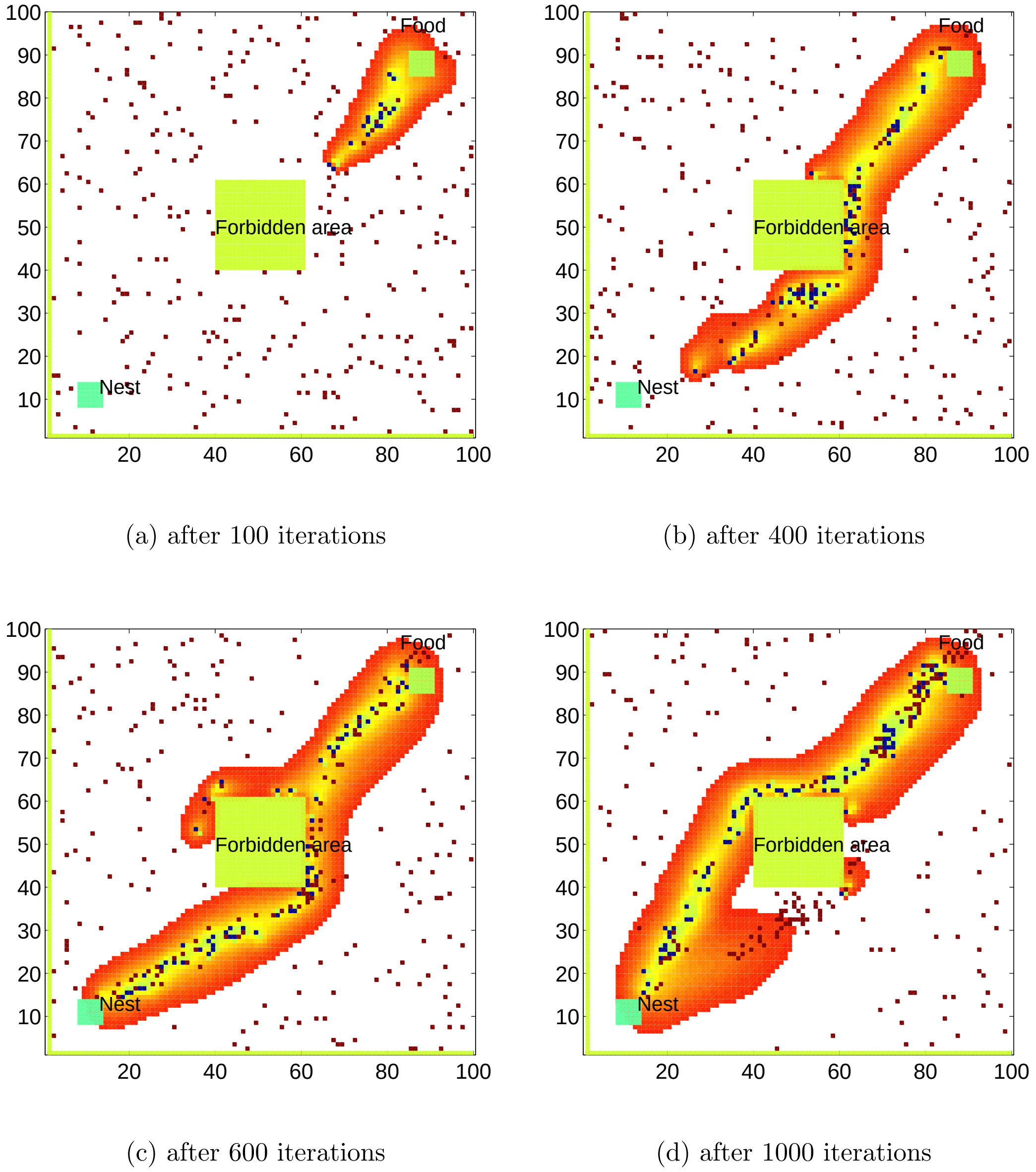
(a) initial conditions

(b) after 200 iterations

(c) after 1500 iterations

(d) traffic jam

Figure 4.6: Illustration of ant trail formation. Figures 4.6a, 4.6b, 4.6c simulate motion of 100 ants with model parameters $a = 0.6$, $\sigma^2 = 0.6$. In the figure 4.6b is remarkable a diffusing and evaporating pheromone trail of single ant (in the middle). In the figure 4.6c is trail fully established. The last figure 4.6d is taken from simulation of 600 ants with the same parameters $a = 0.6$, $\sigma^2 = 0.6$. Increased number of ants caused "traffic jam" on the ant trail.



(a) after 100 iterations

(b) after 400 iterations

(c) after 600 iterations

(d) after 1000 iterations

Figure 4.7: a) The first foragers decide, which path will be used. b) Most of ants use bottom path. c) Still, most of ants use bottom path, however, some foragers discovered second upper path. d) Upper path is fully developed. Original path is going to disappear.

Chapter 5

Conclusions

In our work we have reviewed basic properties of CA and demonstrated their wide applicability in numerous systems. These included biological systems, such as ants complex behaviour or reversible CA systems. Particularly for elementary reversible CA we have shown that they are very rich in their long-time statistical structure. In many case we were able to demonstrate that the latter CA are not ergodic (with respect to equiprobable measure) at any observable computer time scale. On the other hand, periodicity of such CA ensures that they must ultimately come back into a low entropy state. This might be of a particular conceptual importance in statistical physics where, as a rule, ergodicity is very difficult to analyse. Our results also show that while many systems in nature might seemingly follow the second law of thermodynamics (even on time scales that are much larger than typical time scales inherent to a dynamics) they ultimately conspire against it. On the same vein this is the scenario which is predicted in many cosmological scenarios (see, e.g., R. Penrose, Cycles of Time: An Extraordinary New View of the Universe [51]).

In the last section which is devoted to the CA application in biology we created a model that simulates self-organization of ants while creating ant trails. This specific application of CA nicely characterizes a complex behaviour which can be achieved through simple CA rules.

Appendix A

This appendix contains the most important part of the MATLAB code (for version R2009b) simulating and analysing two dimensional reversible rules (ERCA).

```
function [T S R aveg period condprobabs]=rever2d(rule,c,c0,t,draw)

% INPUT
% c     - initial configuration at the time 1
% c0    - initial configuration at the time 0
% rule  - number of simulated rule
% t     - maximum time
% draw  - logical; if the evolution is showed or not
% OUTPUT
% T           - time vector
% S           - structure of entreopies evolution
% R           - space density evolution
% aveg        - mean density over time
% period      - period, if 0 than period is grater than t
% condprobabs - condition probabilities (odrder 1 and 2);

    % -> INITIALIZATION
rulebin=dec0bin(rule);
c=boundconds(c);
initstate=c; %initial condition for the period control
period=0;c0=0*c;aveg=c0;c3=c0;sc=c0;sc0=c0;sc3=c0;ck1=c0;ck2=c0;
T=1:t;R=0*T;S.loc=R;S.glob=R;S.glob1=R;S.glob2=R;
S.loc1=R;S.loc2=R;
    % <- INITIALIZATION

    % -> CA ANALYSIS
for t=T %time iteration
%     CA density
```

```
    R(t)=sum(sum(c))/n^2;
%     local entropy per point
    S.loc(t)=sum(sum(entropyfilt(c,true(7))/n^2));
%     global entropy per point
    S.glob(t)=entropy(c);
%     conditioned probabilities
    p1=sum(sum(c.*c0))/sum(sum(c0));
    p2=sum(sum(c.*(1-c0)))/sum(sum(1-c0));
    q=sum(sum(c0))/n^2;
    S.glob1(t)=entro2(p1,p2,q);
    p1=sum(sum(c.*c3))/sum(sum(c3));
    p2=sum(sum(c.*(1-c3)))/sum(sum(1-c3));
    q=sum(sum(c3))/n^2;
%     conditioned entropy of the second order
    S.glob2(t)=entro2(p1,p2,q);
%     calculation of conditioned probabilities
    if t>=3
%      cond.probabs. averaging
        sc=sc+c;sc2=sc2+c0;sc3=sc3+c3;
        ck1=ck1+c.*c0;
        ck2=ck2+c.*c3;
    end
    aveg=aveg+c;
    % <- CA ANALYSIS

    % -> CONTROL OF PERIOD
    if and(~period,t>1)
        if c==initstate
            period =t-1;
            disp(period);
            T(t+1:end)=[];
            S.loc(t+1:end)=[];
            R(t+1:end)=[];
            break;
        elseif c==c3
            period=2*t-2;
            T=1:2*t-3;
            S.loc(t+1:end)=[];
            R(t+1:end)=[];
            S.loc=[S.loc fliplr(S.loc(1:end-3))];
            R=[R fliplr(R(1:end-3))];
            break;
        end
    end
    % <- CONTROL OF PERIOD

    % -> DRAW THE WORLD
    if draw
showtheworld(c)
    end
    % <- DRAW THE WORLD

    % -> CA EVOLUTION
    d=0*c;
    for i=2:n-1
        for j=2:n-1
            if rule > 0
            x=neumann(c(i-1:i+1,j-1:j+1),rulebin);
            else
            x=life(c(i-1:i+1,j-1:j+1));
            end
            d(i,j)=mod(x-c0(i,j),2); %elementary reversible
        end
    end
    c3=c0;c0=c; %shift to the history
    c=boundconds(d);
    % <- CA EVOLUTION

    % -> SNAPSHOT
    if sum(t==[5 50 500])==1
  makeasnap(c);
    end
    % <- SNAPSHOT

end % time iteration

    % -> CONDITIONED PROBABILITIES
condprobabs.b1=ck1./sc2;
condprobabs.b2=ck2./sc3;
condprobabs.w1=(T(end)-sc-sc2+ck1)./(T(end)-sc2);
condprobabs.w2=(T(end)-sc-sc3+ck2)./(T(end)-sc3);
    % <- CONDITIONED PROBABILITIES
```

Appendix B

This appendix contains the most important part of the MATLAB code (for version R2009b), which simulate ant trail formation.

```
function [items]=Ants(Nants,jump,variance,show)

% INPUT
% Nants - number
% jump  - parameter a
% variance - parameter sigma^2
% swoh  - logical, decides, whether show the evolution
% OUTPUT
% items - number of items of delivered food

N=100;                  %Lattice size
T=1000;                 %Number of time steps;
pmax=255; %Maximum pheromone value
ant=zeros(N);           %Lattice of ants
p=zeros(N);             %Lattice of pheromones
obj=zeros(N);           %Lattice of objects
anttab=zeros(Nants,3);  %List of the ants - number of the ants is conserved.
obj(8:13,13:18)=-1;     %Nest cells
obj(85:90,82:87)=1;     %Food cells
% obj(40:60,40:60)=2;     %Forbidden area
obj(:,1)=2;obj(:,end)=2;%Borders
obj(1,:)=2;obj(end,:)=2;%Borders
nest=[10,15];           %Nest coordinates
food=[87,84];           %Food coordinates
items=zeros(1,T);
store =0;

% Direction from the nest; any ant knows this.
```

```
[x y]=meshgrid(1:N);
DirFromNestX=(x-nest(2));DirFromNestY=(y-nest(1));
ss=sqrt(DirFromNestX.^2+DirFromNestY.^2);
DirFromNestX=DirFromNestX./ss;DirFromNestY=DirFromNestY./ss;

% ant(17,50)=3;
% Ants generation
i=0;
while i<Nants
    x=randi(N-2)+1;y=randi(N-2)+1;
    if ~(obj(x,y) || ant(x,y))
        i=i+1;
        anttab(i,:)=[randi(8)-1 x y];
        ant(x,y)=1+anttab(i,1);
    end
end

sheet([400 400]); % figure initialization

for t=1:T  % time iteration

    % -> SET BUFFER VARIABLES
    nant=0*ant;
    nanttab=0*anttab;
    [gradpy gradpx] = gradient(p,1,1);
    ss=sqrt(gradpx.^2+gradpy.^2);
    ss(ss==0)=1;
    gradpx=gradpx./ss;gradpy=gradpy./ss;

    % <- SET BUFFER VARIABLES

    % -> DRAW A WORLD
    if show
        disp([t, store])
        Show(ant,p,obj);
        drawnow;
    end
    % <- DRAW A WORLD

    % -> PHEROMONE DIFFUSION
      D=0.1;
```

```
     koef=0.999;
     p(obj==2)=0;
     np=koef*min(pmax,max(0,floor(p+D*4*del2(p))));
   % <- PHEROMONE DIFFUSION

   % -> MOVE OF THE ANT
   for i=1:Nants
       % Move of the ant:
       % the ant try jump according to jump rule.
       % if the jump is unsuccessful, the ant stays still.
       % movement is not successful if the target cell is not empty.

       ii=anttab(i,2);jj=anttab(i,3);d=ant(ii,jj);

 % Movement of the ant with food.
       if d<0 || 0
           xd=+1*gradpx(ii,jj)-DirFromNestY(ii,jj);
           yd=+1*gradpy(ii,jj)-DirFromNestX(ii,jj);
% Movement of the ant without food.
% Prefers a direction from the nest if stays on a pheromone.
       elseif d>0 && p(ii,jj)>0;
           xd=+1*gradpx(ii,jj)+DirFromNestX(ii,jj);
           yd=+1*gradpy(ii,jj)+DirFromNestY(ii,jj);
       else
 %Brownian motion
           r=2*pi*rand;
           xd=sin(r);
           yd=cos(r);
       end

 % Normalisation of the ant direction
       if norm([xd yd])
           no=norm([xd yd]);xd=xd/no;yd=yd/no;
       end
 % Rule for ant jump
       x=round(ii+jump*xd+variance*randn);
       y=round(jj+jump*yd+variance*randn);
       x=min(N,max(1,x));
       y=min(N,max(1,y));

 % discussion of various situations
```

```
if ~edge(x,y,N) && nant(x,y)==0 && obj(x,y)==0 && ant(x,y)==0
      % sucessful jump
      nant(x,y)=d;
      nant(ii,jj) = 0;
      nanttab(i,:)=[d-1 x y];
      np(ii,jj)=max(p(ii,jj),-sign(d)*pmax);
        elseif d>0 && obj(x,y)==1
      % found food
            nant(ii,jj)=-d;
            nanttab(i,:)=[-(d-1) ii jj];
        elseif d<0 && obj(x,y)==-1
      % returned to the nest with food
            nant(ii,jj)=abs(d);
            nanttab(i,:)=[abs(d)-1 ii jj];
            store=store+1;
        else
      % unsuccesful jump
            nant(ii,jj)=d;
            nanttab(i,:)=anttab(i,:);
        end

    end %i=1:Nants
    % <- MOVE OF THE ANT

    % -> UPDATE VARIABLES BEFORE NEW CYCLE
    ant=nant;
    p=np;
    anttab=nanttab;
    % -< UPDATE VARIABLES BEFORE NEW CYCLE

    items(t)=store; % measure of delivered items of food
end % time iteration
```

Bibliography

[1] J. Neumann and A. Taub, *Collected works: Rings of operators* (Pergamon press, 1961) ISBN 9780080095691

[2] J. L. Schiff, *Cellular Automata: A Discrete View of the World* (Wiley & Sons, Inc., 2007) ISBN 047016879X

[3] A. Ilachinski, *Cellular automata: a discrete universe* (World Scientific Pub Co Inc, 2001) ISBN 981238183X

[4] G. Weisbuch and S. Ryckebusch, *Complex systems dynamics: An introduction to automata networks.* (Addison-Wesley, 1991) ISBN 9780201627329

[5] D. A. Wolf-Gladrow, *Lattice-gas cellular automata and lattice Boltzmann models: an introduction* (Springer, 2000) ISBN 9783540669739

[6] T. Toffoli and N. Margolus, Phys. D: Nonlinear Phenom. **47**, 263 (1991)

[7] K. Chen and N. Gershenfeld, *Circuit design for logic automata*, Master's thesis, MIT (2009), `http://hdl.handle.net/1721.1/52781`

[8] A. Turing, J. of Math. **58**, 345 (1938)

[9] S. Wolfram, Nature **311**, 419 (1984)

[10] S. Wolfram, *Cellular automata and complexity: collected papers* (Westview Press Boulder, CO:, 1994) ISBN 0201627167

[11] S. Wolfram, *A new kind of science*, Vol. 1 (Wolfram Media Champaign, IL, 2002) ISBN 1579550088

[12] B. Chopard and M. Droz, *Cellular Automata Modeling of Physical Systems* (1998) ISBN 0521461685

[13] L. Kier, P. Seybold, and C. Cheng, *Modeling chemical systems using cellular automata* (Springer, Dordrecht, 2005) ISBN 9781402036576

[14] G. Ermentrout and L. Edelstein-Keshet, J. Theor. Biol. **160**, 97 (1993)

[15] J. Bell, Physics **1**, 195 (1964)

[16] M. Blasone, P. Jizba, and F. Scardigli, J.Phys.Conf.Ser. **174**, 012034 (2009), `http://arxiv.org/abs/0901.3907`

[17] G. 'tHooft, Arxiv preprint hep-th/0003005(2000)

[18] G. 'tHooft, "Quo vadis quantum mechanics?." (Springer, 2005) Chap. Determinism beneath quantum mechanics, pp. 99–111, ISBN 9783540221883

[19] M. Blasone and P. Jizba, J. Phys.: Conf. Ser. **67**, 012046 (2007)

[20] G. 'tHooft, Class. Quantum Grav. **16**, 3263 (1999)

[21] J. Kari, *Developments in Language Theory*, Lecture Notes in Computer Science **3572**, 2 (2005)

[22] S. Wolfram, Rev. Mod. Phys. **55**, 601 (1983)

[23] "Lifewiki," (2011), website of a community for Conway's Game of Life, `http://www.conwaylife.com/wiki/Main_Page`

[24] "Universal turing machine (utm) implemented in conway's game of life," (2010), website of Paul Rendell, the author, `http://rendell-attic.org/gol/utm/index.htm`

[25] "Universal constructor based spaceship," (2010), announcement at the web forum, `http://conwaylife.com/forums/viewtopic.php?f=2&t=399&p=2327#p2323`

[26] “Open directory project,” (2011), conway’s Game of Life, `http://www.dmoz.org/Computers/Artificial_Life/Cellular_Automata/Conway%27s_Game_of_Life/`

[27] “Game of life news,” (2011), web page, `http://pentadecathlon.com/lifeNews/index.php`

[28] A. Carter and A. Sanda, Phys. Rev. D **23**, 1567 (1981)

[29] N. Margolus, Phys D Nonlinear Phenom **10**, 81 (1984)

[30] *The Universal Turing Machine: A Half-Century Survey*, edited by R. Herken (Springer, 1995) ISBN 3211826378

[31] A. Church, J. Symb. Log. **1**, 40 (1936)

[32] M. Gardner, “Wheels, life, and other mathematical amusements,” (W.H. Freeman, 1983) Chap. 20-22, ISBN 9780716715887

[33] M. Cook, Complex Systems **15**, 1 (2004)

[34] M. van Lieshout, “Markovianity in space and time,” (2005), `http://arxiv.org/pdf/math/0608242`

[35] R. M. Gray, *Entropy and information theory* (Springer-Verlag New York, Inc., 1990) ISBN 0387973710

[36] S. Willson, Theory Comput. Syst. **9**, 132 (1975)

[37] O. De Feo and I. Belykh, “Dynamical system theory for engineers,” (2006), available online, `http://lanoswww.epfl.ch/studinfo/courses/cours_dynsys/notes/ErgodicTheoryOfDiscreteDynamicalSystems.pdf`

[38] M. Pivato, “The ergodic theory of cellular automata,” (2007), `http://euclid.trentu.ca/pivato/Research/ergodic.CA.pdf`

[39] G. Cattaneo, E. Formenti, G. Manzini, and L. Margara, in *STACS 97* (Springer, 1997) pp. 427–438

[40] A. T. Sampson, J. M. Bjørndalen, and P. S. Andrews, Evolutionary Computation 2009 CEC 09 IEEE Congress on, 225(2009), `http://kar.kent.ac.uk/24108/`

[41] D. Gordon, Nature **380**, 121 (1996)

[42] H. Wassmann, Curr. Biol. **16**, R570 (1996)

[43] A. Attygalle and E. Morgan, Adv. Insect Physiol. **18**, 1 (1985)

[44] M. Wittlinger, R. Wehner, and H. Wolf, Science **312**, 1965 (2006)

[45] S. Åkesson and R. Wehner, J. Exp. Biol. **205**, 1971 (2002)

[46] D. Vowles, Nature **165**, 282 (1950)

[47] Y. Camlitepe and D. Stradling, Proc. R. Soc. B **261**, 37 (1995)

[48] A. Hoekstra, J. Kroc, and P. Sloot, *Simulating Complex Systems by Cellular Automata* (Springer, 2010) ISBN 9783642122026

[49] Y. Sugiyama, M. Fukui, M. Kikuchi, K. Hasebe, A. Nakayama, K. Nishinari, S. Tadaki, and S. Yukawa, New J. Phys. **10**, 033001 (2008)

[50] S. Benjamin, N. Johnson, and P. Hui, J. Phys. A: Math. Gen. **29**, 3119 (1996)

[51] R. Penrose, *Cycles of Time: An Extraordinary New View of the Universe* (Knopf Doubleday Publishing Group, 2011) ISBN 9780307265906