Computing with Probabilistic Cellular Automata*

Martin Schüle[1], Thomas Ott[2], and Ruedi Stoop[1]

[1] Institute of Neuroinformatics, ETH and University of Zurich, 8057 Zurich, Switzerland
[2] Institute of Applied Simulation, ZHAW Zurich, 8820 Wädenswil, Switzerland

Abstract. We investigate the computational capabilities of probabilistic cellular automata by means of the density classification problem. We find that a specific probabilistic cellular automata rule is able to solve the density classification problem, i.e. classifies binary input strings according to the number of 1's and 0's in the string, and show that its computational abilities are related to critical behaviour at a phase transition.

1 Preliminaries

Cellular automata (CA) models have been widely studied and applied in physics, biology and computer science. They are among the simplest mathematical systems which exhibit self-organisation, complex patterning and capability of universal (Turing) computation [1,2,3]. Various authors have suggested CA as the generic model for parallel, biologically inspired computing [1,4,5]. As such they are closely related to neural networks [5]. In this contribution we investigate claims regarding the computational abilities of elementary probabilistic cellular automata.

1.1 Deterministic Cellular Automata

A *deterministic* cellular automaton (DCA) is specified by a d-dimensional regular discrete lattice L with given boundary conditions, a finite set Σ of states x_i assigned to each node or cell i of the lattice and a local rule f acting on the states in the range k of the neighbourhood N_k^i of each cell i in discrete time steps. Given some initial configuration of states, the local rule completely determines the dynamics of the cellular automaton. In this paper we deal with *finite* DCA, that is DCA with a finite number N of cells, and *elementary* DCA, that is DCA with $d = 1$, $\Sigma = \{0, 1\}$ and nearest neighbourhood $k = 3$. In this case, there are 256 different possible local rules ${x_i}^{t+1} = f({x_{i-1}}^t, {x_i}^t, {x_{i+1}}^t)$. The N cells are subject to periodic boundary conditions and their states x_i are updated synchronously by the local rule. Local rules are given by a rule table.

* This work was supported by ETH Research Grant TH-04 07-2.

C. Alippi et al. (Eds.): ICANN 2009, Part II, LNCS 5769, pp. 525–533, 2009.

Example 1 (Rule 232). The rule table of the so-called *majority* DCA rule 232 is $(f(111) = 1, f(110) = 1, f(101) = 1, f(100) = 0, f(011) = 1, f(010) = 0, f(001) = 0, f(000) = 0)$.
It is customary to assign a decimal number to such rule tables. One speaks of *rule 232* as the binary expansion of the decimal number 232 ($232 = 11101000$) which encodes the rule table when read from left to right.

A configuration or *global state* $\mathbf{x}^t$ of a CA is the string of the states of the N cells at the time t, that is $\mathbf{x}^t = (x_0^t, x_1^t, ..., x_{N-1}^t)$. Starting from an initial configuration $\mathbf{x}^0$, the global function or map F then maps configuration $\mathbf{x}^t$ to $\mathbf{x}^{t+1} = F(\mathbf{x}^t)$, thereby generating a space-time pattern. The global map F is only indirectly given through the local rule f. A *quiescent* or *stationary* global state $\mathbf{x}^*$ is defined as $\mathbf{x}^* = F(\mathbf{x}^*)$.

Any DCA rule can be represented as a *Boolean function*, which is expressible as a *disjunctive normal form* (DNF) [1]. The DNF is a disjunction of clauses, where a clause is a conjunction of Boolean variables.

Example 2 (DNF of rule 232). DCA rule 232 written as a DNF is $(X_{i-1} \wedge X_i \wedge X_{i+1}) \vee (X_{i-1} \wedge X_i \wedge \neg X_{i+1}) \vee (X_{i-1} \wedge \neg X_i \wedge X_{i+1}) \vee (\neg X_{i-1} \wedge X_i \wedge X_{i+1})$ with the Boolean variables X_i, the disjunction denoted by $\vee$, the conjunction by $\wedge$ and the negation by $\neg$.

As outlined in [6] the DNF of CA rules can then be rewritten as algebraic expressions which represents CA dynamics in a concise form.

Example 3 (Algebraic expression of rule 232). The algebraic expression for DCA rule 232 is

$$x_i{}^{t+1} = x_{i-1}{}^t \cdot x_i{}^t + x_i{}^t \cdot x_{i+1}{}^t + x_{i-1}{}^t \cdot x_{i+1}{}^t - 2x_{i-1}{}^t \cdot x_i{}^t \cdot x_{i+1}{}^t \qquad (1)$$

We now turn to a stochastic generalisation of deterministic CA, that is probabilistic CA.

1.2 Probabilistic Cellular Automata

Probabilistic cellular automata (PCA) are generalized DCA in the regard that the states x_i are stochastically updated, that is by some local probability transition function. In the case of elementary PCA this means that the probability of having cell i the value $\tilde{x}_i = 1$ at the time $t+1$ is given by $p[\tilde{x}_i^{t+1}|(x_i^t)]$, where (x_i^t) are the states of the cells in the next-nearest-neighbourhood of cell i, i.e. $(x_i^t) = (x_{i-1}^t, x_i^t, x_{i+1}^t)$. The local probability transition function is subject to the normalisation condition $\sum_{\tilde{x}_i^{t+1}=\{0,1\}} p[\tilde{x}_i^{t+1}|(x_i^t)] = 1$. The *probabilistic majority rule* we will work with exemplifies the notion of elementary PCA.

Example 4 (Majority PCA rule). The rule table of the majority PCA rule is $(p(111) = 1, p(110) = \epsilon, p(101) = \epsilon, p(100) = 1-\epsilon, p(011) = \epsilon, p(010) = 1-\epsilon, p(001) = 1-\epsilon, p(000) = 0)$.

That is the probability for $\tilde{x}_i^{t+1} = 1$ given $(x_{i-1}^t, x_i^t, x_{i+1}^t)$ is $(1, \epsilon, \epsilon, 1-\epsilon, \epsilon, 1-\epsilon, 1-\epsilon, 0)$.

The properties of PCA from a statistical mechanics viewpoint have been widely discussed [7,8]. The general dynamics of PCA is given by a *master equation* [7,9] which reads to

$$P[\tilde{\mathbf{x}}_i^{t+1}] = \sum_{(\mathbf{x}_i)} P[(\mathbf{x}_i^t)] \prod_i p[\tilde{x}_i^{t+1}|(x_i^t)] \tag{2}$$

where $\sum_{(\mathbf{x}_i)} P[(\mathbf{x}_i^t)]$ is the sum (over all global states $(\mathbf{x}_i^t)$) of the probabilities to find the PCA in some state $\mathbf{x}_i$ at the time t and $p[\tilde{x}_i^{t+1}|(x_i^t)]$ is the local probability transition function.

Following the algebraic approach outlined above, the local probability transition function of the majority PCA rule can be written as

$$\begin{aligned} p[\tilde{x}_i^{t+1}|(x_i^t)] = {} & x_{i-1}{}^t + x_i{}^t + x_{i+1}{}^t - 2x_{i-1}{}^t x_i{}^t - 2x_{i-1}{}^t x_{i+1}{}^t - 2x_i{}^t x_{i+1}{}^t + 4x_{i-1}{}^t x_i{}^t x_{i+1}{}^t \\ & -\epsilon(x_{i-1}{}^t + x_i{}^t + x_{i+1}{}^t - 3x_{i-1}{}^t x_i{}^t - 3x_{i-1}{}^t x_{i+1}{}^t - 3x_i{}^t x_{i+1}{}^t + 6x_{i-1}{}^t x_i{}^t x_{i+1}{}^t). \end{aligned}$$

Accordingly, the dynamics of the majority PCA rule can be written as

$$\begin{aligned} \tilde{x}_i^{t+1} = {} & x_{i-1} + x_i + x_{i+1} - 2x_{i-1}x_i - 2x_{i-1}x_{i+1} - 2x_i x_{i+1} + 4x_{i-1}x_i x_{i+1} \\ & -Y_i(x_{i-1} + x_i + x_{i+1} - 3x_{i-1}x_i - 3x_{i-1}x_{i+1} - 3x_i x_{i+1} + 6x_{i-1}x_i x_{i+1}) \end{aligned} \tag{3}$$

where $\{Y_i\}_{i=0}^{N-1}$ is a set of iid random variables with probability distribution $p[Y_i = 1] = \epsilon, p[Y_i = 0] = 1 - \epsilon$ (and x_i^t shortened to x_i).

Solving the master equation for large N is usually not feasible. In order to simplify the treatment one works in a *mean field approximation* (MF). The MF approximation assumes that the values x_i^t are independent of each other and the probability of having cell i in state x_i^t is therefore given through the *global density* $\rho^t = \frac{1}{N}\sum_i x_i^t$. In the MF, the dynamics of the global density variable becomes (with ρ^t shortened to ρ)

$$\rho^{t+1} = 3\rho - 6\rho^2 + 4\rho^3 - \epsilon(3\rho - 9\rho^2 + 6\rho^3) \tag{4}$$

PCA are *finite Markov chains* [14]. In the case of the majority PCA rule defined above, we have an *absorbing* finite Markov chain. Again, as an array of N cells yields 2^N different configurations, i.e. global states, most techniques developed in the field of finite Markov chains are not practicable for large N. Theorems regarding general properties of absorbing finite Markov chains can however be of use.

2 Computing with Probabilistic Cellular Automata

The computational abilities of deterministic cellular automata (DCA) have been early recognized and discussed in Wolfram's contributions and by ensuing papers [10]. The basic idea is that some input string $\mathbf{x}^0$ at time $t = 0$ is processed to some output string $\mathbf{x}^T$ at time T through the DCA's time evolution. We therefore define *computation*, in this context, as the global map $G(\mathbf{x^0}) = \mathbf{x^T}$ with $G = F^T(\mathbf{x^0})$, that is the global map F is iterated T times. The *computation time* T is the number of the discrete time steps in the DCA evolution from the

input to the output string. The output string $\mathbf{x}^T$ is usually, but not always, some quiescent, i.e. stationary state.

Wolfram and other authors have described computing by DCA within formal language theory. We follow here a different approach by, generally, adhering to a dynamical system viewpoint on CA and, specifically, by discussing a particular exemplary problem, the so-called *density classification problem*. In fact, it can be questioned whether any global map G should be called "computational" irrespective of a specific, well-defined *computational problem* at hand.

The density classification problem is the computational task to classify input strings according to their densities of 1's and 0's. Usually this means that the CA should asymptotically evolve to either the 0- or 1-quiescent state, that is the global state with all states equal 0, or 1 respectively, depending on the initial densities. In the Markov chain liteature, this final, stationary global state is called the *absorbing* state. The density classification problem is an obvious, well-defined computational task which any basic computing device should be able to carry out. Land and Belew have however shown that there exists no elementary DCA able to solve the problem [11]. Later Fuks has demonstrated that a specific PCA rule can solve the problem in a stochastic sense [12]. We investigate a different PCA rule, e.g. the majority PCA rule introduced above, and discuss its computational capabilities in broader terms.

2.1 PCA and DCA

As we will see, the majority PCA rule solves the density classification problem in a stochastic sense. The basic reason for this is, that the PCA, unlike DCA, will not get stuck in certain non-intended periodic patterns, that is certain quiescent or periodic states. As pointed out before, a PCA rule is a stochastic combination of DCA rules [8]. In the case of the majority PCA rule this means that we have with a certain probability ϵ DCA rule 232 (the deterministic majority rule) and with probability $1-\epsilon$ DCA rule 150.

Example 5 (The majority PCA rule as a combination of DCA rules). With probability ϵ the majority PCA rule is equal to DCA rule 232

$$x_i^{t+1} = x_{i-1}{}^t x_i{}^t + x_{i-1}{}^t x_{i+1}{}^t + x_i{}^t x_{i+1}{}^t - 2x_{i-1}{}^t x_i{}^t x_{i+1}{}^t$$

and with probability $1-\epsilon$ DCA to rule 150

$$x_i^{t+1} = x_{i-1}{}^t + x_i{}^t + x_{i+1}{}^t - 2x_{i-1}{}^t x_i{}^t - 2x_{i-1}{}^t x_{i+1}{}^t - 2x_i{}^t x_{i+1}{}^t + 4x_{i-1}{}^t x_i{}^t x_{i+1}{}^t.$$

It has been shown [13] that a pair of elementary DCA, namely rules 184 and 232, can solve the density classification problem exactly. In view of the above considerations, this comes as no surprise as a pair of DCA rules is, in a certain sense, equivalent to a single PCA rule. In future work, we intend to study further which combinations of DCA are equivalent to which PCA and why.

2.2 Density Classification and Phase Transitions

We now discuss simulation results for the majority PCA and compare it with the MF predictions. As mentioned before, the majority PCA is a finite, elementary probabilistic cellular automata with periodic boundary conditions and synchronous updating. Because the majority PCA is a finite absorbing Markov chain, every individual majority PCA will end up in an absorbing state after a finite number of time steps [14], that is every input string will eventually be classified. An ensemble of equivalent majority PCA will however show a distinct behaviour which can be approximated by the MF approach. In the MF

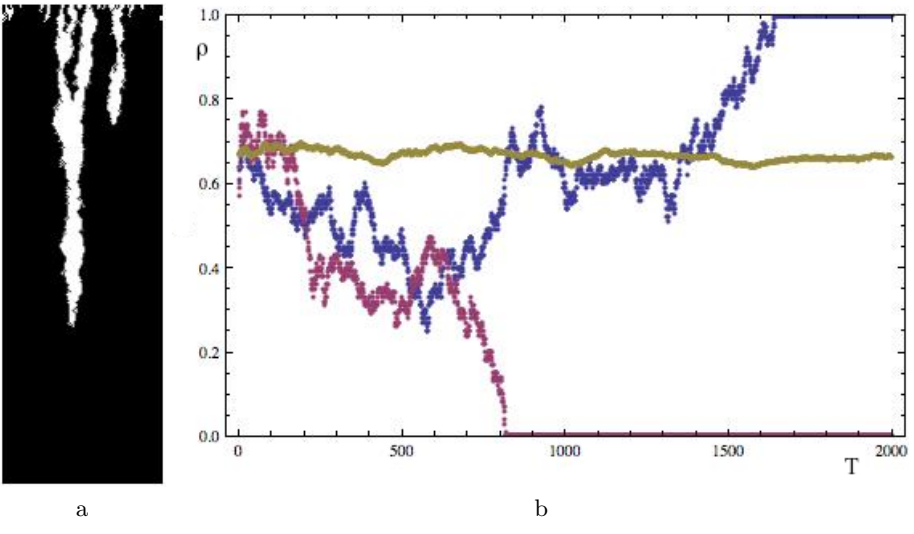

Fig. 1. a Space-time pattern of a single majority PCA with $\rho^0 = \frac{2}{3}$, $\epsilon = \frac{2}{3}$ and $N = 100$. Time axis is from top down. **b** Dynamics of the global density for two single majority PCA with $\rho^0 = \frac{2}{3}$, $\epsilon = \frac{2}{3}$ and $N = 100$ and of the mean global density of an ensemble of 100 majority PCA with $\rho^0 = \frac{2}{3}$, $\epsilon = \frac{2}{3}$ and $N = 100$. The mean global density fluctuates around $\rho^0 = \frac{2}{3}$.

approach the global density for $\epsilon = \frac{2}{3}$ is $\rho^{t+1} = \rho^t$, that is the global density for an ensemble of equivalent majority PCA is preserved. For $\epsilon = \frac{2}{3}$ this result holds also when using the exact local probability transition function, which can be seen by taking expectation values of both sides of equation (3) which yields $E[\rho^{t+1}] = \frac{1}{N}\sum_i E[\tilde{x}_i^{t+1}] = \frac{1}{N}\sum_i E[x_i^t] = E[\rho^t]$. For $\epsilon = \frac{2}{3}$ the majority PCA thus solves the density classification problem in the sense that input strings will be classified correctly with a probability equal to the initial density of the input string. For $\epsilon > \frac{2}{3}$ the MF approach predicts an unstable fixed point at $\rho = \frac{1}{2}$ and stable fixed points at $\epsilon = 0$, $\epsilon = 1$ respectively. For $\epsilon < \frac{2}{3}$ the MF approach predicts a stable fixed point at $\rho = \frac{1}{2}$ and unstable fixed points at $\epsilon = 0$, $\epsilon = 1$

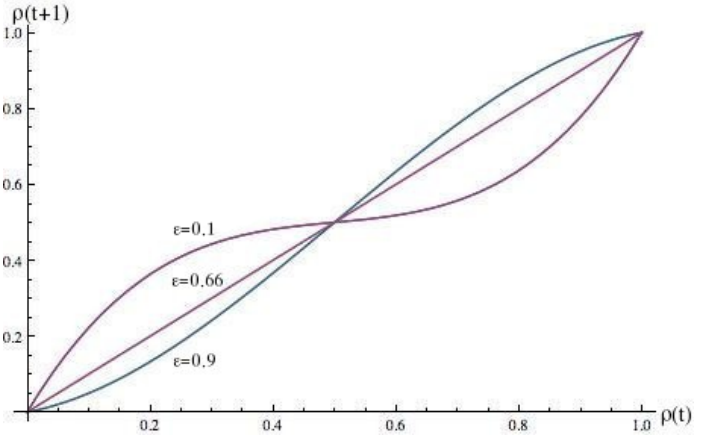

Fig. 2. Dynamics of the global density in the MF approach, i.e. ρ^{t+1} vs. ρ^t, for $\epsilon = 0.1, \frac{2}{3}$ and 0.9 respectively

respectively. The MF approach thus predicts a first-order phase transition at the critical point $\epsilon = \frac{2}{3}$ in the order parameter ρ. By "first-order phase transition" we mean, in this context, a discontinuous transition in the order parameter.

The simulation results displayed in Fig. 3 shows that there is indeed a phase transition at $\epsilon = \frac{2}{3}$, albeit of second order (i.e. a continuous phase transition). The global density at $\epsilon = \frac{2}{3}$ is, as predicted, preserved. For $\epsilon > \frac{2}{3}$ the final global density is below the MF prediction, but nevertheless classifies better than the rule in [12]. For $\epsilon < \frac{2}{3}$ the global density rapidly tends to $\rho^T = \frac{1}{2}$.

As stated before, every individual majority PCA will eventually end up in the absorbing state, that is in either the 0- or 1-quiescent state. We define the *accurancy* of the computation as the fraction of correct solutions by the majority PCA to the density classification problem, that is the final global density. The *efficiency* of the computation is then the ratio of time to absorption T to the accurancy. As can be inferred from Fig. 3 and 4 the computational efficiency is highest around the critical point $\epsilon = \frac{2}{3}$ for initial densities $\rho^0 = 0.5, \frac{1}{3}$, and $\frac{2}{3}$ and at around $\epsilon = 0.9$ for initial densities $\rho^0 = 0.1$ and 0.9. In ongoing work we study the possibilities to derive analytical expressions formalising these observations.

There has been much speculation in recent years about computation at the "edge of chaos" or near critical points of phase transitions, albeit no formal theory, as far as we know, has elaborated on these speculations. From the majority PCA, we first see that we have a phase transition at $\epsilon = \frac{2}{3}$ and an enhanced computational capacity in the sense that there is a transition from solving one problem, the density classification problem, to another which could be termed the "reshuffling problem" as an input string with an arbitrary initial density will be effectively reshuffled to a state with $\rho = \frac{1}{2}$. Secondly, we observe that the "optimal" computational capability of the majority PCA, that is its efficiency in the sense defined above, is in the vicinity of the critical point $\epsilon = \frac{2}{3}$.

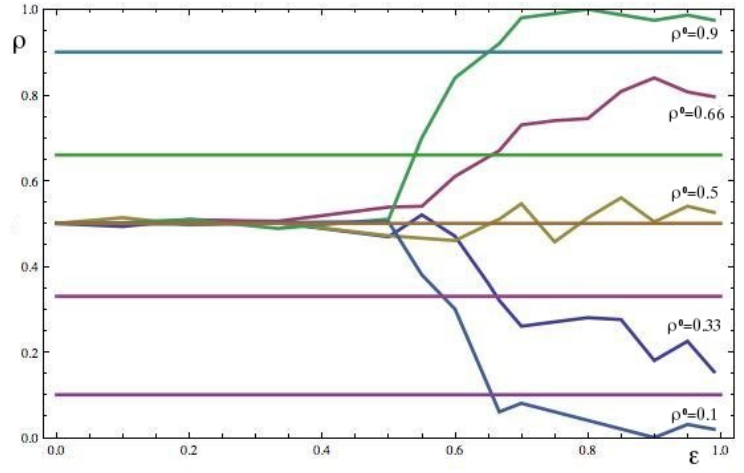

Fig. 3. The global density ρ at time $T = 10000$ for an ensemble of 100 identical majority PCA with $N = 100$ cells in dependence of the parameter ϵ for different initial global densities $\rho^0 = 0.9, \frac{2}{3}, 0.5, \frac{1}{3}, 0.1$

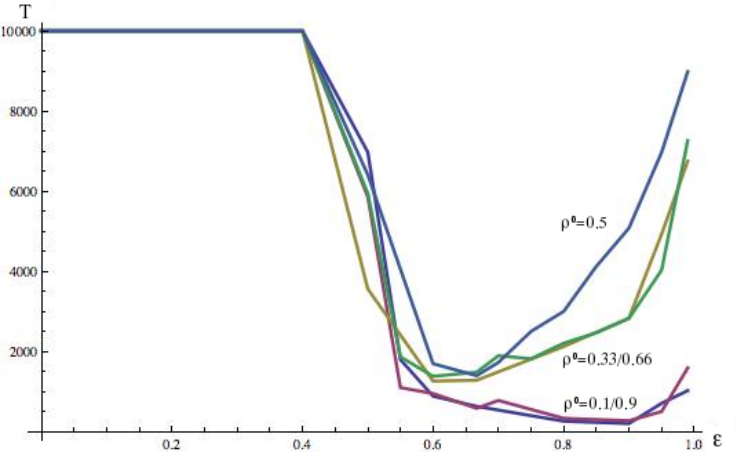

Fig. 4. The absorption time T for an ensemble of 100 identical majority PCA with $N = 100$ in dependence of parameter ϵ for different initial global densities ρ^0. The curves show the *minimum* average absorption time as the running time of the simulation was bounded by $T = 10000$.

3 Summary

In order to examine the computational abilities of probabilistic cellular automata we have investigated a specific well-defined computational task, the density classification problem, that is the classification of binary input strings according to the number of 1's and 0's in the input strings. In this contribution we showed:

1. that a simple stochastic generalisation of the deterministic majority rule 232, that is the majority PCA rule, can solve the density classification problem (which is not solvable by DCA) in a stochastic sense,
2. that, compared with the deterministic rule, the enhanced computational capability is due to a stochastic combination of deterministic CA rules,
3. that the majority PCA is classifying more strings correctly above a critical point than other rules proposed before
4. and that there is a second-order phase transition in the order parameter ρ which is related to the computational abilities of the majority PCA.

This preliminary contribution focuses on illustrating the computational abilities of probabilistic cellular automata within a specific computational task. Probabilistic cellular automata are simple computational systems which can closely model biological of physical systems. We believe that further investigation into probabilistic cellular automata will shed light on the connection between actual physical or biological systems and their computational abilities. Probabilistic cellular automata offer a unified approach combining the methods of statistical mechanics with the dynamical systems approach and notions of computation. As such, they can serve as a natural benchmark for novel measures of "natural computation" (e.g. [15]), which may eventually lead to a better understanding of what "computation by nature" is.

References

1. Wolfram, S.: A New Kind of Science. B&T (2002)
2. Cook, M.: Universality in Elementary Cellular Automata. Complex Systems 15, 1–40 (2004)
3. Deutsch, A., Dormann, S.: Cellular Automaton Modeling of Biological Pattern Formation, Birkhäuser (2004)
4. Kari, J.: Theory of Cellular Automata: A Survey. Theoretical Computer Science 334, 3–33 (2005)
5. Garzon, M.: Models of Massive Parallelism: Analysis of Cellular Automata and Neural Networks. Springer, Heidelberg (1995)
6. Schüle, M., Ott, T., Stoop, R.: Global dynamics of finite cellular automata. In: Kůrková, V., Neruda, R., Koutník, J. (eds.) ICANN 2008, Part I. LNCS, vol. 5163, pp. 71–78. Springer, Heidelberg (2008)
7. Grinstein, G., Jayaprakash, C.: Statistical Mechanics of Probabilistic Cellular Automata. Phys. Rev. Lett. 55, 2527–2530 (1985)
8. Lebowitz, J.L., Maes, C., Speer, E.R.: Statistical Mechanics of Probabilistic Cellular Automata. Journal of Statistical Physics 59, 117–170 (1990)

9. Van Kampen, N.G.: Stochastic Processes in Physics and Chemistry. North Holland Publishing Company, Amsterdam (1981)
10. Wolfram, S.: Computation Theory of Cellular Automata. Communications in Mathematical Physics 96, 15–57 (1985)
11. Land, M., Belew, R.K.: No Perfect Two-State Cellular Automata for Density Classification Exists. Phys. Rev. Lett. 74, 5148–5150 (1984)
12. Fuks, H.: Nondeterministic density classification with diffusive probabilistic cellular automata. Phys. Rev. E 66, 066106 (2002)
13. Fuks, H.: Solution of the density classification problem with two cellular automata rules. Phys. Rev. E 55, R2081 - R2084 (1997)
14. Kemeny, J.G., Snell, J.L.: Finite Markov Chains. D. Van Nostrand Co. (1960)
15. Stoop, R., Stoop, N.: Natural computation measured as a reduction of complexity. Chaos 14, 675–679 (2004)