Fundamenta Informaticae 138 (2015) 1–16
DOI 10.3233/FI-2015-1194
IOS Press

On the Dynamics of Cellular Automata with Memory

Genaro J. Martínez*
Escuela Superior de Cómputo, Instituto Politécnico Nacional, México D.F., México
International Centre of Unconventional Computing, University of the West of England
BS16 1QY Bristol, United Kingdom
Genaro.Martinez@uwe.ac.uk

Andrew Adamatzky
International Centre of Unconventional Computing, University of the West of England
BS16 1QY Bristol, United Kingdom
Andrew.Adamatzky@uwe.ac.uk

Ramon Alonso-Sanz
Universidad Politecnica de Madrid. ETSIA (Estadistica,GSC)
Madrid, Spain
Ramon.Alonso@upm.es

Abstract. Elementary cellular automata (ECA) are linear arrays of finite-state machines (cells) which take binary states, and update their states simultaneously depending on states of their closest neighbours. We design and study ECA with memory (ECAM), where every cell remembers its states during some fixed period of evolution. We characterize complexity of ECAM in a case study of rule 126, and then provide detailed behavioural classification of ECAM. We show that by enriching ECA with memory we can achieve transitions between the classes of behavioural complexity. We also show that memory helps to 'discover' hidden information and behaviour on trivial (uniform, periodic), and non-trivial (chaotic, complex) dynamical systems.

Keywords: elementary cellular automata, classification, memory, computability, gliders, collisions, complex systems

*Address for correspondence: Escuela Superior de Cómputo, Instituto Politécnico Nacional, México D.F., México, International Centre of Unconventional Computing, University of the West of England

1. Introduction

A complexity theory emerged from studies of computable problems in computer science and mathematical foundations of computation, when a need came to compare performance and resource-efficiency of algorithms. Typically time complexity (number of steps) and space complexity (memory of a single processor and number of processors) are expressed in terms of a Turing machine or an equivalent mathematical device. Each specific kind of a Turing machine represents a certain class of complexity [27, 9, 15, 29]. When related to complex systems meaning of the word 'complexity' is different and heavily depends on its context. Complexity of a system is almost never quantified but often related to unpredictability. Theory of cellular automata (CA) deal with complexity its entire life [34, 2, 11, 12, 14, 26, 28, 39, 22]. Due to the transparency of CA structures, their complexity can be measured and analyzed [38, 13]. An elementary cellular automaton (ECA) is a one-dimensional array of finite automata, each automaton takes two states and updates its state in discrete times according to its own state and states of its two closest neighbours, all cells update their state synchronously. Thus in 1980s Wolfram subdivided ECA onto four complexity classes [38]:

- class I. CA evolving uniformly. Evolution is dominated by a unique state of alphabet from any random initial condition.
- class II. CA evolving periodically. Evolution is dominated by blocks of cells which are periodically repeated from any random initial condition.
- class III. CA evolving chaotically. evolution is dominated by sets of cells without some defined pattern for a long time from any random initial condition.
- class IV. Include all previous cases, known as the class *complex*. Evolution is dominated by non-trivial structures emerging and travelling along of the evolution space where also uniform, periodic, or chaotic regions can coexist with these structures.

2. Elementary cellular automata (ECA)

A CA is a tuple $\langle \Sigma, \varphi, \mu, c_0, d \rangle$ where d is a dimensional lattice and each cell x_i, $i \in N$ (where N belongs to natural numbers set), takes a state from a finite alphabet Σ such that $x \in \Sigma$. A sequence $c \in \Sigma^n$ (where $n \in N$), represents a string or a global configuration c on Σ of length of n cell-states. We write a set of finite configurations as Σ^n. Cells update their states by an evolution rule $\varphi : \Sigma^\mu \to \Sigma$, such that $\mu = 2r + 1$ represents a cell neighbourhood that consists of a central cell and a number of r-neighbours connected locally. If $k = |\Sigma|$ hence there are k^{2r+1} neighbourhoods and $k^{k^{2r+1}}$ evolution rules. An evolution diagram for a CA is represented by a sequence of configurations $\{c_i\}$ generated by the global mapping $\Phi : \Sigma^n \to \Sigma^n$, where a global relation is given as $\Phi(c^t) \to c^{t+1}$. Thus c_0 is the initial configuration. Cell states of a configuration c^t are updated simultaneously by the local rule, as follows: $\varphi(x^t_{i-r}, \ldots, x^t_i, \ldots, x^t_{i+r}) \to x^{t+1}_i$ where i indicates cell position and r is the radius of neighbourhood in μ. Thus, ECA represents a system of order $(k = 2,\ r = 1)$ (in Wolfram's notation [36]) . To represent a specific ECA evolution rule we will write the evolution rule in a decimal notation, e.g. as rule 54 or φ_{R54}.

3. Elementary cellular automata with memory (ECAM)

Conventional CA are memoryless: The new state of a cell depends on the neighbourhood configuration solely at the preceding time-step configuration. CA with memory are an extension of CA in such a way that every cell x_i is allowed to remember its states during some fixed period of its evolution. CA with memory have been proposed originally by Alonso-Sanz in [3, 4, 5, 6]. Hence we implement a memory function ϕ, as follows: $s_i^{(t)} = \phi(x_i^{t-\tau+1}, \ldots, x_i^{t-1}, x_i^t)$, where $1 \leq \tau \leq t$ determines the *degree of memory*. Thus, $\tau = 1$ means no memory (or conventional evolution), whereas $\tau = t$ means unlimited trailing memory. Each cell trait $s_i \in \Sigma$ is a state function of the series of states of the cell i with memory backward up to a specific value τ. In the memory implementations run here, commences to act as soon as t reaches the τ time-step. Initially, i.e., $t < \tau$, the automaton evolves in the conventional way. Later, to proceed in the dynamics, the original rule is applied on the cell states s as: $\varphi(\ldots, s_{i-1}^{(t)}, s_i^{(t)}, s_{i+1}^{(t)}, \ldots) \rightarrow x_i^{t+1}$ to get an evolution with memory. Thus in CA with memory, while the mapping φ remains unaltered, historic memory of all past iterations is retained by featuring each cell as a summary of its past states from ϕ. We can say that cells canalise memory to the map φ [4, 5].

In ECA with memory (ECAM) treated here, we consider the *memory function* ϕ in a form of *majority memory*, $\phi_{maj} \rightarrow s_i$, where in case of a tie, i.e., same number of 1s and 0s in past configurations, the last value x_i^t is to be adopted as $s_i^{(t)}$, which implies no memory effect. These $\#1 = \#0$ ties are only feasible when τ is even, in which case the effect of memory may appear as somehow *weaker*, or simply *different*, compared to the effect of the odd $\tau - 1$ or $\tau + 1$ close lengths of memory. Thus, ϕ_{maj} function represents the classic majority function. For three values [27], then we have that: $\phi_{maj}(a, b, c) : (a \wedge b) \vee (b \wedge c) \vee (c \wedge a)$. Any map of previous states may act as memory (not only majority). Thus, minority, parity, alpha, ..., or any CA rule acting as memory, weighted memory, ..., etc [5, 6]. Evolution rules representation for ECAM in this paper is given in [16, 17, 19] as follows: $\phi_{CARm:\tau}$ where *CAR* is the decimal notation of a particular ECA rule and m is the kind of memory used with a specific value of τ. This way, for example, the majority memory (maj) incorporated in ECA rule 30 employing five steps of a cell's history ($\tau = 5$) is denoted simply as: $\phi_{R30maj:5}$.

4. Rule 126 with memory: case study of complexity

Here we consider a particular case to illustrate the effect of memory, deriving in complex dynamics from a chaotic rule [23]. We deal with the chaotic ECA (class III) rule 126. This is a special chaotic rule because its evolution yields sets of regular languages [38, 26]. We can deduce from previous analysis that ECA rule 126 could contain another kind of interesting information. Selecting a kind of memory we will see that particularly ECAM $\phi_{R126maj:4}$ displays a large number of glider guns emerging from random initial conditions, and emergence of a number of non-trivial patterns colliding constantly [23].

The local-state transition function φ corresponding to ECA rule 126 is represented as follows:

$$\varphi_{R126} = \begin{cases} 1 & \text{if} \quad 110, 101, 100, 011, 010, 001 \\ 0 & \text{if} \quad 111, 000 \end{cases}$$

ECA rule 126 shows a chaotic global behaviour typical from Class III in Wolfram's classification [37]. In φ_{R126} we can easily recognize an initial high probability of alive cells, i.e. cells in state '1'; with a 75% to appear in the next time-step, and the complement of only 25% to get state 0. It will be always a new alive cell iff φ_{R126} has one or two alive cells so that the equilibrium is reached when there is an overpopulation condition.

4.1. Mean field approximation

Mean field theory is a technique for discovering statistical properties of CA without analysing evolution spaces of individual rules [26]. The method assumes that states in Σ are independent and do not correlate with each other in the local function φ_{R126}. Thus we can study probabilities of states in a neighbourhood in terms of the probability of a single state (the state in which the neighbourhood evolves), and probability of the neighbourhood as a product of the probabilities of each cell in it. McIntosh in [25] presents an explanation of Wolfram's classes with a mixture of probability theory and de Bruijn diagrams, resulting in a classification based on mean field theory curve, as follows: (a) class I: monotonic, entirely on one side of diagonal; (b) class II: horizontal tangency, never reaches diagonal; (c) class IV: horizontal plus diagonal tangency, no crossing; (d) class III: no tangencies, curve crosses diagonal.

For the one dimensional case, all neighbourhoods are considered as follows:

$$p_{t+1} = \sum_{j=0}^{k^{2r+1}-1} \varphi_j(X) p_t^v (1-p_t)^{n-v}$$

such that j is an index relating each neighbourhood composed by X cells $x_{i-r}, \ldots, x_i, \ldots, x_{i+r}$. Thus n is the number of cells into every neighbourhood, v indicates how often state '1' occurs in X, $n-v$ shows how often state '0' occurs in the neighbourhood X, p_t is the probability of cell being in state '1' while q_t is the probability of cell being in state '0', i.e., $q = 1 - p$. The polynomial for ECA rule 126 is defined as follows: $p_{t+1} = 3p_t q_t$. Because φ_{R126} is classified as a chaotic rule, we expect no tangencies and its curve must cross the identity; recall that φ_{R126} has a 75% of probability to produce a state one.

Mean field curve confirms that probability of state '1' in space-time configurations of φ_{R126} is 0.75 for high densities related to big populations of 1's. The curve demonstrates also that φ_{R126} is chaotic because the curve cross the identity with a first fixed point at the origin $f = 0$ and the absence of unstable fixed points inducing non stable regions in the evolution. Nevertheless, the stable fixed point is $f = 0.6683$, which represents a 'concentration' of '1's diminishing during the automaton evolution. So the initial inspection indicates no evidence of complex behaviour emerging in φ_{R126}. Of course a deeper analysis is necessary for obtaining more features from a chaotic rule, so the next sections explain other techniques to study in particular periodic structures.

4.2. De Bruijn diagrams

De Bruijn diagrams [26, 32] are proven to be an adequate tool for describing evolution rules in one dimension CA, although originally they were used in shift-register theory (the treatment of sequences where their elements overlap each other). Paths in a de Bruijn diagram may represent chains, configurations or classes of configurations in the evolution space.

For a one-dimensional CA of order (k, r), the de Bruijn diagram is defined as a directed graph with k^{2r} vertices and k^{2r+1} edges. The vertices are labeled with elements of an alphabet of length $2r$. An edge is directed from vertex i to vertex j, if and only if, the $2r-1$ final symbols of i are the same that the $2r-1$ initial ones in j form a neighbourhood of $2r+1$ states represented by $i \diamond j$. In this case, the edge connecting i to j is labeled with $\varphi(i \diamond j)$ (the value of the neighbourhood defined by the local function) [33].

The extended de Bruijn diagrams [26] are useful for calculating all periodic sequences by means of the cycles defined in the diagram. These ones also show the *shift* of a sequence for a certain number of *generations*. Thus we can get de Bruijn diagrams describing periodic sequences for ECA rule 126.

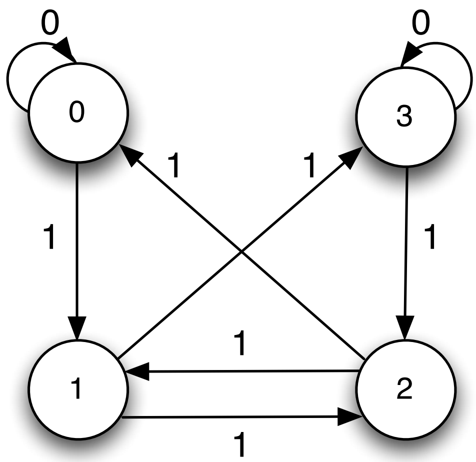
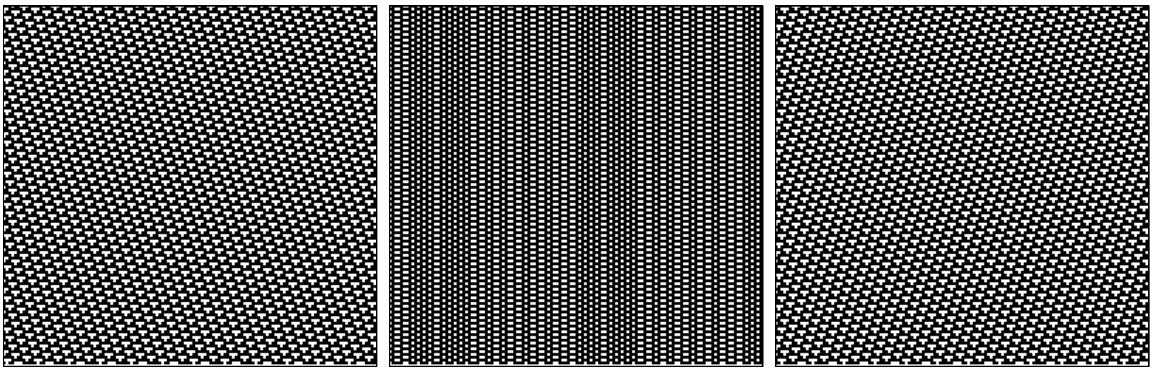


Figure 1. De Bruijn diagram for the ECA rule 126.

De Bruijn diagram associated to ECA rule 126 is shown in Fig. 1.[1] Figure 1 shows that there are two neighbourhoods evolving into 0 and six neighbourhoods into 1. State 1 has higher frequency. This indicates a possibility that the local transition function is injective and *Garden of Eden* configurations [1] exist. These are configurations that cannot be constructed from other configurations, i.e., configurations without ancestors. In one dimension, the *subset* diagram can calculate in two transformation the Garden of Eden configurations, and the *pair diagram* can calculate configurations with multiple ancestors [25]. Let us take the equivalent construction of a de Bruijn diagram in order to describe the evolution in two steps of ECA rule 126 (having now nodes composed by sequences of four symbols); the cycles of this new diagram are presented in Fig. 2.

Cycles inside de Bruijn diagrams can be used for obtaining regular expressions representing a periodic pattern. Figure 2 displays three patterns calculated as: (a) shift -3 in 2 generations representing a pattern with displacement to the left, (b) shift 0 in 2 generations describing a static pattern travelling without displacement, and (c) shift $+3$ in 2 generations is exactly the symmetric pattern given in the first evolution. So, we can also see in Fig. 2 that it is possible to find patterns travelling in both directions, as gliders or mobile structures. But generally these constructions (strings) cannot live in combination with others structures and therefore it is really hard to have this kind of objects with such characteristics. Although, ECA rule 126 has at least one glider! This will be explained in the next section.

[1]De Bruijn diagrams were calculated using NXLCAU21 designed by McIntosh; available in `http://delta.cs.cinvestav.mx/~mcintosh/cellularautomata/SOFTWARE.html`



(a) (b) (c)

Figure 2. Patterns calculated with extended de Bruijn diagrams, in particular from cycles of order $(x, 2)$ (that means x-shift in 2-generations).

4.3. Filters help for discovering hidden dynamics

Filters are essential tools for discovering hidden order in chaotic or complex rules. Filters were introduced in CA studies by Wuensche who employed them to automatically classify cell-state transition functions, see [42]. Also filters related to tiles were successfully applied and deduced in analysing space-time behaviour of ECA governed by rules 110 and 54 [24, 20, 21].

This way, we have found that ECA rule 126 has two types of two dimension tiles (which together work as filters over φ_{R126}): the tile $t_1 = \begin{bmatrix} 1111 \\ 1001 \end{bmatrix}$, and the tile $t_2 = \begin{bmatrix} 0000000 \\ 0111110 \\ 1100011 \\ 0110110 \\ 1111111 \end{bmatrix}$. Filter t_1 works on configurations generated by φ_{R126}. Filter t_2 is exploited when ECA rule 126 is enriched with memory.

The application of the filter t_1 is effective to discover gaps with little patterns travelling on triangles of '1' states in the evolution space. Although even in this case it may be unclear how a dynamics would be interpreted, a careful inspection on the evolution brings to light very small localizations (as still life), as shown in Fig. 3. This localization emerging in ECA rule 126 and pinpointed by a filter is the periodic pattern calculated the de Bruijn diagram (Fig. 2b). The last one offers more information because such cycles allow to classify the whole phases when this glider is coded in the initial condition. Circles in Fig. 3 show some interesting regions that now are clearer with filters working. Some of them display very simple gliders (stationary), periodic meshes, and non-periodic structures emerging and existing inside chaotic patterns in several generations.

4.4. Dynamics emerging in ECA rule 126 with memory

CA with memory opened a new family of evolution rules with different and interesting dynamics [5, 6]. We explore three types of memory: minority, majority, and parity. In the latter case, $s_i^{(t)} = x_i^{t-\tau+1} \oplus \ldots \oplus x_i^{t-1} \oplus x_i^t$. Figure 4 illustrates three kinds of dynamics emerging in ECAM rule 126, for some values

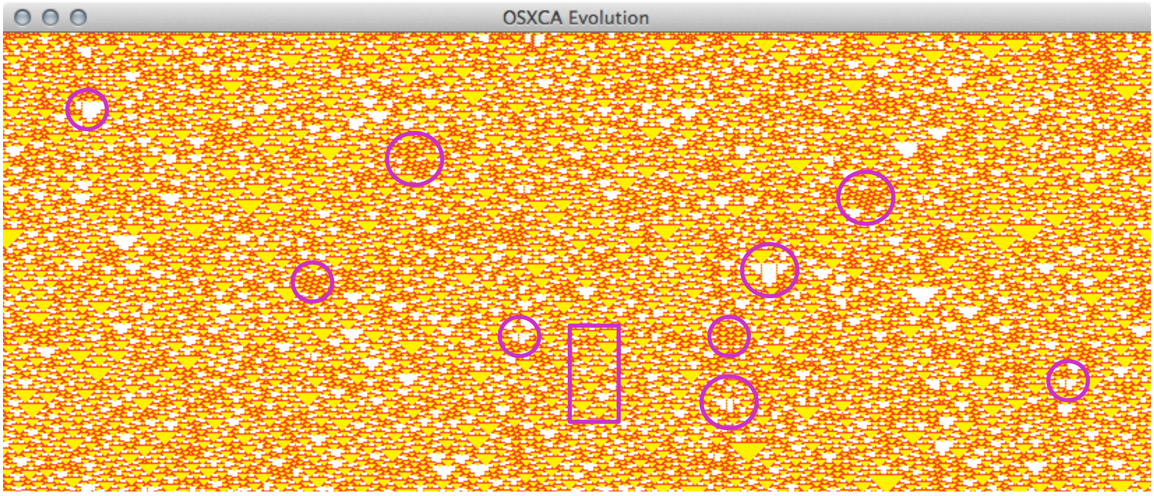


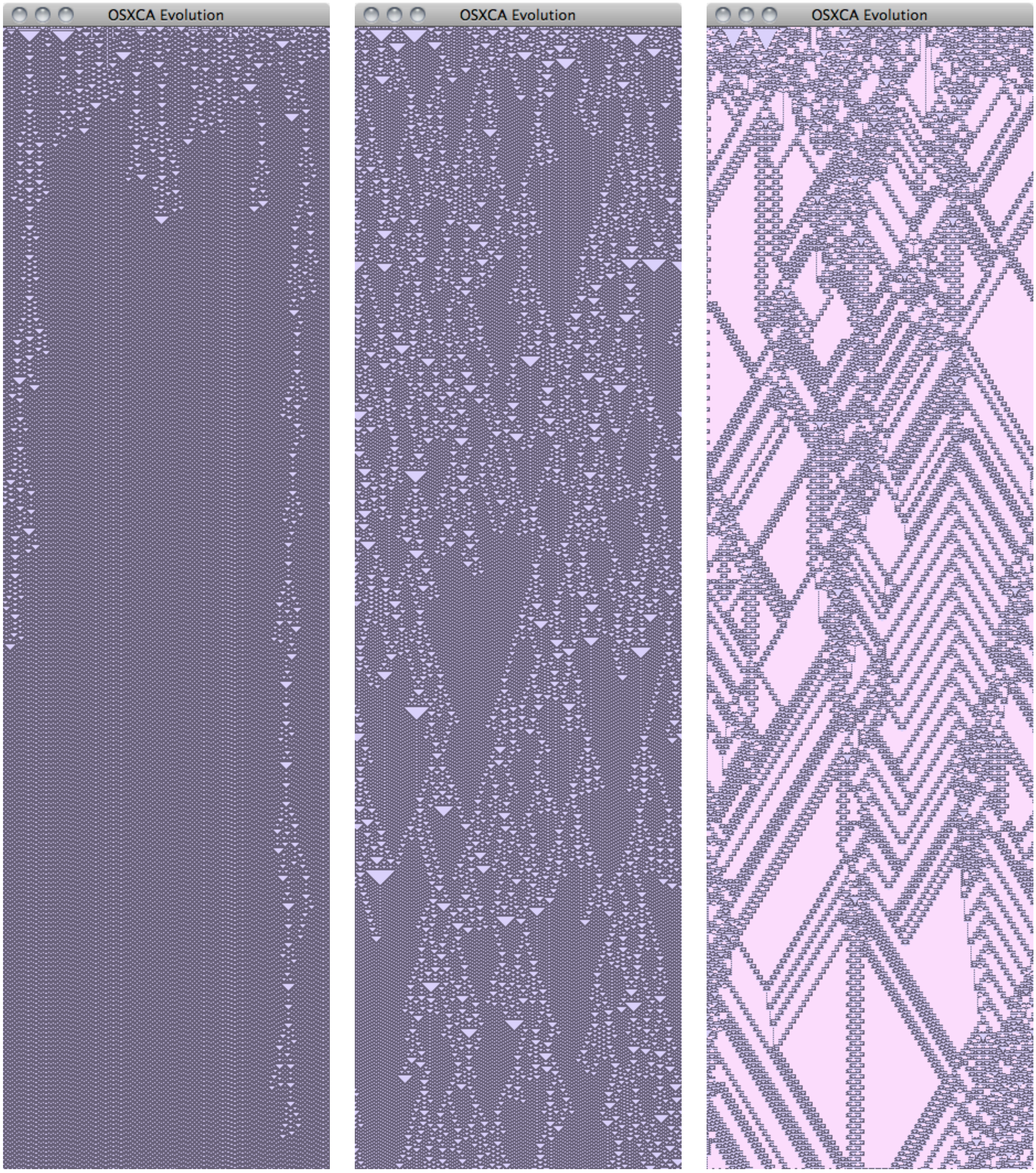
Figure 3. Filtered space-time configuration in ECA rule 126.

of τ.[2] Exploring different values of τ, we found that large odd values of τ tend to define *macrocells*-like patterns [40, 26], while even values are responsible for a mixture of periodic and chaotic dynamics. Figure 4(a) illustrates large periodic regions with few complex patterns travelling isolation developed by function $\phi_{R126min:3}$. Figure 4(b) shows the function $\phi_{R126par:2}$, its evolution is more interesting because we can see the emergence of some complex patterns than also interact producing other types of complex structures, including mobile self-localizations or gliders. By exploring systematically distinct values of τ, we found that $\phi_{R126maj:4}$ produces an ample diversity of non-trivial emergence of patterns travelling and colliding. Figure 4(c) shows the most interesting evolution with well defined complex patterns, not just mobile self-localisations but also the emergence of glider guns, they are complex patterns which travel on the evolution space emitting periodically another kind of gliders.

An interesting evolution is that starting with a single non-quiescent cell. Particularly, $\phi_{R126maj:4}$ displays a growth complex behaviour. An example of this space-time configuration is given in Fig. 5 , showing the first 1152 steps, where in this case the automaton needed other 30,000 steps to reach a stationary configuration. Filtering is convenient to eliminate the non relevant information about gliders. In the same figure, we can see a number of gliders, glider guns, still-life configurations, and a wide number of combinations of such patterns colliding and travelling with different velocities and densities. Consequently, we can classify a number of periodic structures, objects, and interesting reactions. By selecting a majority memory function on the chaotic ECA rule 126 we can transform its dynamics to complex dynamics. Thus, for some CA rules, with a memory m function ϕ and value τ we can derive a complex system from a chaotic system or vice versa, transform a chaotic system to complex.

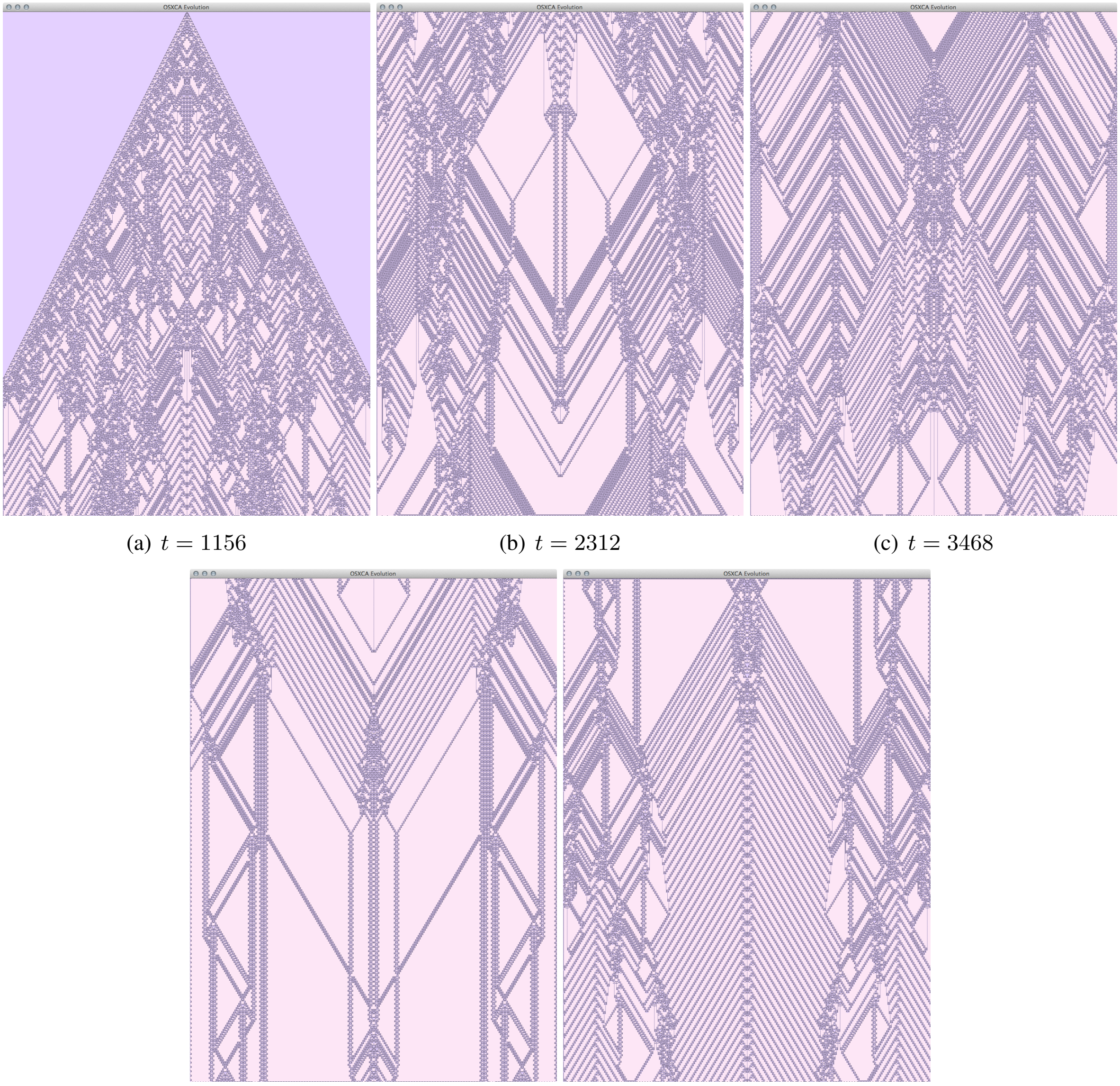
Further, we explore systematically — 88 representatives of equivalence classes of ECA rules [22] — if memory functions are able to implement transformations between Wolfram's classes. Thus, we prove experimentally that each class may be converted to another class with a particular kind of memory.

[2]Evolutions of $\phi_{R126maj:\tau}$ were calculated with *OSXLCAU21 system* available in `http://uncomp.uwe.ac.uk/genaro/OSXCASystems.html`



(a) (b) (c)

Figure 4. (a) $\phi_{R126min:3}$ displays a typical evolution of ECAM rule 126 with minority memory $\tau = 3$, (b) $\phi_{R126par:2}$ displays an evolution but now evolving with parity memory, and (c) the most interesting evolution with ECAM rule $\phi_{R126maj:4}$, where we can see the emergence of complex patterns as gliders and glider guns. In this case a filter is selected for a best view of complex patterns and their interactions. Snapshots start with the same random initial conditions on a ring of 296 cells evolving in 1036 generations.



(a) $t = 1156$ (b) $t = 2312$ (c) $t = 3468$

(d) $t = 4624$ (e) $t = 5780$

Figure 5. Filtered space-time configuration of ECAM $\phi_{R126maj:4}$ evolving with a ring of 843 cells, periodic boundaries, starting just from one non-quiescent cell and running up to $t = 5780$. The evolution is shown in five increasing t-intervals as labeled in the snapshots

.

5. ECAM classification

To derive a new rule from a basic ECA rule one should select an ECA rule and compose this rule with a memory function. The memory function will determine if the original ECA rule preserves the same class (respective to Wolfram's classes) or if it changes to another class. The parameter τ is updated from 2 to 20 to trace migration between classes. The values of τ exceeding 10 do not change rules behaviour, i.e., not more changes are reported. An exhaustive description of this analysis is provided in [18]. Using this technique we classified ECA rules as follows. From a ECA rule given we have that:

> ***Strong***, the memory functions **do most transformations** and the rule changes to another different class for any values of τ. Dynamics of CA from a 'strong class' can be changed by **any** memory function.
>
> ***Moderate***, the memory functions **can transform** the rule to another class **and conserve** the same class as well. Dynamics of CA from 'moderate class' can be changed by **at least one** or more memory functions.
>
> ***Weak***, the memory functions **are unable** to transform one class to another. Dynamics of CA from 'weak class' cannot be affected by **any** of the kind of memories studied in the present paper.

Table 1. ECAM classification.

TYPE	NUMBER	RULES
strong	39	2, 7, 9, 10, 11, 15, 18, 22, 24, 25, 26, 30, 34, 35, 41, 42, 45, 46, 54, 56, 57, 58, 62, 94, 106, 108, 110, 122, 126, 128, 130, 138, 146, 152, 154, 162, 170, 178, 184.
moderate	34	1, 3, 4, 5, 6, 8, 13, 14, 27, 28, 29, 32, 33, 37, 38, 40, 43, 44, 72, 73, 74, 77, 78, 104, 132, 134, 136, 140, 142, 156, 160, 164, 168, 172.
weak	15	0, 12, 19, 23, 36, 50, 51, 60, 76, 90, 105, 150, 200, 204, 232.

Table 1 presents the ECA classification based in memory functions. We have ECA rules which composed with a particular kind of memory are able to reach another class including the original dynamic. The main feature is that, at least, a rule with memory is able to reach all different class. Rules with this property are called *universal* ECAM (five rules). They are rules 22, 54, 130, 146, 152. We highlight that all the universal rules are classified as strong in ECAM's classification.

Also, we have ECA that when composed with memory are able to yield a complex ECAM but with elements of the original ECA rule. They are called *complex ECAM* (44 rules): 6, 9, 10, 11, 13, 15, 22, 24, 25, 26, 27, 30, 33, 35, 38, 40, 41, 42, 44, 46, 54, 57, 58, 62, 72, 77, 78, 106, 108, 110, 122, 126, 130, 132, 138, 142, 146, 152, 156, 162, 170, 172, 178, 184 (an ECAM is the complex in the same sense as complex CA, i.e, it exhibits behaviour matching Wolfram class of complexity [37, 42, 41, 22]). These rules can be detailed in terms of ECAM's classification, as follows:

strong:	9, 10, 11, 15, 22, 24, 25, 26, 30, 35, 41, 42, 46, 54, 57, 58, 62, 106, 108, 110, 122, 126, 130, 138, 146, 152, 162, 170, 178, 184.
moderate:	6, 13, 27, 33, 38, 40, 44, 72, 77, 78, 132, 142, 156, 172.
weak:	-

It is remarkable that none of the rules classified in the weak class is able to reach complex behaviour. These set of rules are robust to any perturbation in terms of ECAM's classification.

6. Relations in ECAM classes

We will deal here with the canonical representative rule of every one of the 88 equivalence classes [35], and not explicitly with the 256 ECA rules. We will enumerate the most important relations. Therefore, from the transitions in Tab 2 , we can reach a class from any other class with some kind of memory at least once. Particularly, ECA rule 130 exemplifies a universal rule that reaches any other class including itself. ECAM preserves the main characteristics of the original evolution rule and they can be found in both ECA and ECAM rules. As was detailed in ECA rule 126, a glider that is found in ECAM $\phi_{R126maj:4}$ already existed in the conventional ahistoric rule. Thus, the dynamics in ECAM also cannot be induced from some previous ECA.

Table 2. Transitions between ECA (column) and ECAM (row) classes governed by function $\phi_{CAm:\tau}$. An example for each class is shown. Particularly, ECA rule 130 is a universal class.

$\phi_{CAm:\tau}$	ECAM uniform	ECAM periodic	ECAM chaos	ECAM complex
ECA uniform	φ_{R32} to $\phi_{R32maj:3}$	φ_{R160} to $\phi_{R160par:5}$	φ_{R40} to $\phi_{R40par:2}$	φ_{R40} to $\phi_{R40par:4}$
ECA periodic	φ_{R130} to $\phi_{R130maj:4}$	φ_{R130} to $\phi_{R130maj:3}$	φ_{R130} to $\phi_{R130par:3}$	φ_{R130} to $\phi_{R130min:3}$
ECA chaos	φ_{R18} to $\phi_{R18maj:10}$	φ_{R30} to $\phi_{R30maj:4}$	φ_{R30} to $\phi_{R30par:2}$	φ_{R126} to $\phi_{R126maj:4}$
ECA complex	φ_{R54} to $\phi_{R54maj:6}$	φ_{R54} to $\phi_{R54par:2}$	φ_{R110} to $\phi_{R110min:3}$	φ_{R54} to $\phi_{R54maj:8}$

If you have selected an ECA class I, II, III, or IV; you could obtain an ECAM class I, II, III, or IV without some prefix which determines exactly the result. Diagrams displayed in Fig. 6 show how we use memory to move between classes of ECA and ECAM. Diagram in Fig. 7 (all memories) shows a directed graph strongly connected due to the transitions in Tab. 2 and Fig. 6. That means that you can reach any class from any class including themselves (loops). As outlined in [13] in the conventional ahistoric context, it is not possible to determine the behaviour of a ECAM from that of its conventional ahistoric ECA. This way, it is undecidable to determine a behaviour of a CAM from any CA. Incidentally, memory can be implemented on any dynamical system, with the aim of discovering hidden information, such as was studied in excitable CA [10].

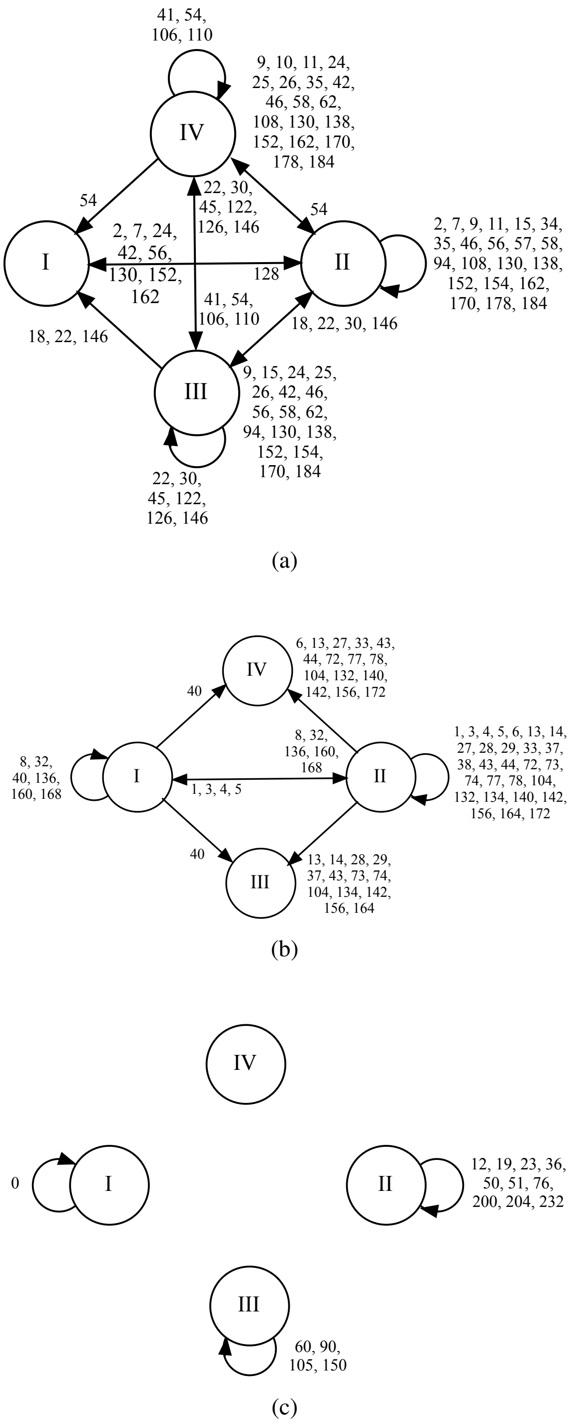


Figure 6. Reachability between ECAM classes: (a) “strong” class, (b) “moderate” class, (c) “weak” class.

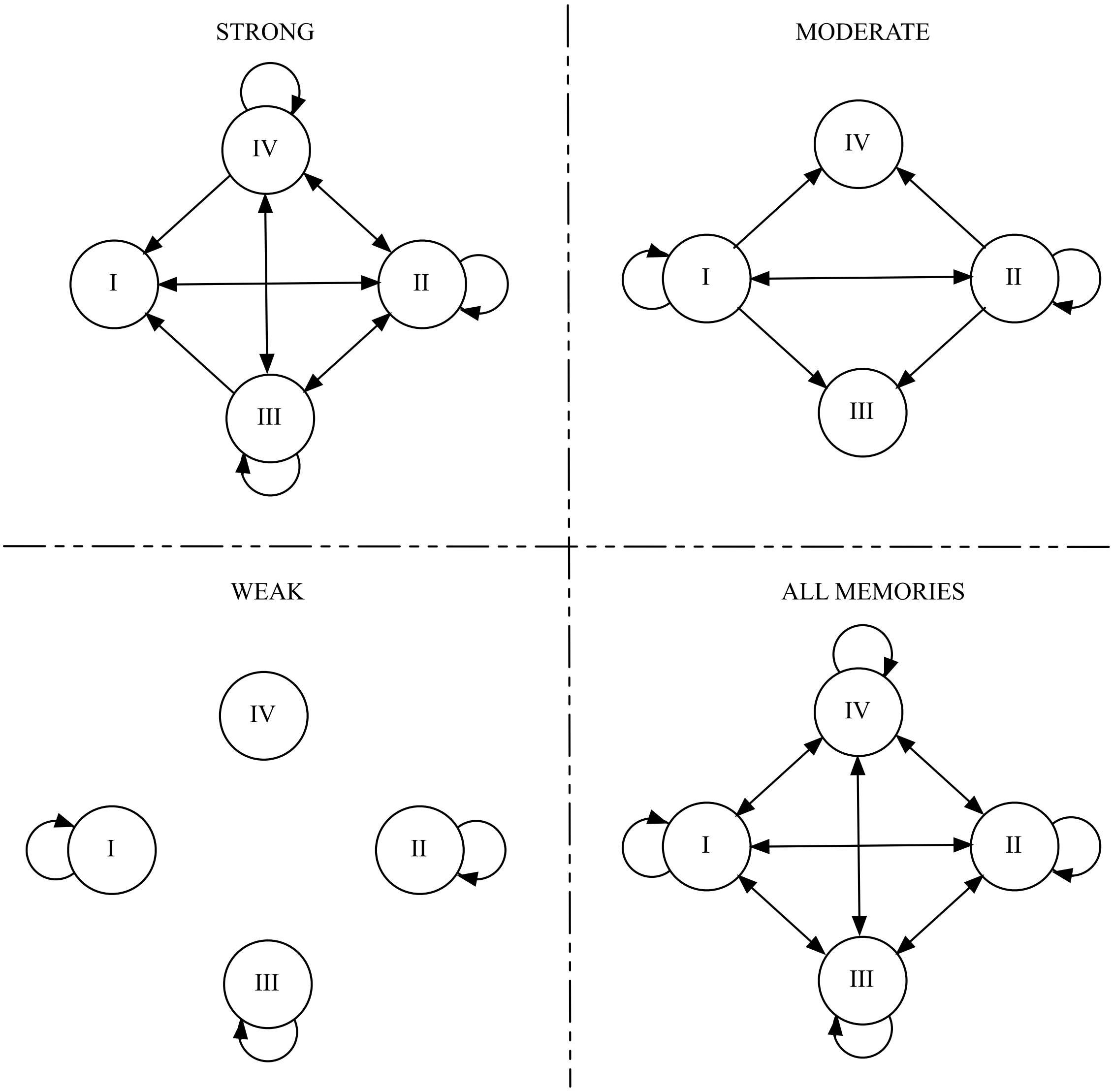


Figure 7. Every ECAM class has rules with behaviour class I, II, III, or IV. If you take one ECA rule with a kind of memory hence you can change to another class. "All memories" diagram show that it is possible to reach any class from some ECA enriched with memory, thus some ECAM is able to reach any class.

7. Discussion

We demonstrated that a memory is a 'universal' switch which allows us to change dynamics of a complex spatially extended systems and to guide the system in a 'labyrinth' of complexity classes. Memory allows us to make complex systems simple and to simple ones complex. The memory implementation mechanism studied here constitute an extension of the basic CA paradigm allowing for an easy systematic study of the effect of memory in CA. This may inspire some useful ideas in using CA as a tool for modelling phenomena with memory. This task has been traditionally achieved by means of differential, or finite-difference, equations, with some continuous component. Thus, it seems plausible that further study on CA with memory should prove profitable, and may be possible to paraphrase Toffoli [31] in presenting CA with memory *as an alternative to (rather than an approximation of)* integro-differential equations in modelling phenomena with memory. Besides their potential applications, CA with memory have an aesthetic and mathematical interest on their own, so that we believe that the subject is worth to studying. Last but not least, other memories are possible. In this study we have implemented an explicit dependence in the dynamics of the past states in the manner: first summary then rule. But the order summary-rule may be inverted, i.e., the rule is first applied and a summary is then presented as new state [7, 8]. This alternative memory implementation enriches the potential use of memory in discrete systems as a tool for modelling, and, again, in our opinion deserves attention on its own. So, CA with memory offer another possibilities to design novel and compact universal systems [30].

References

[1] Amoroso, S. and Cooper, G. [1970] "The Garden-of-Eden theorem for finite configurations," *Proc. Amer. Math. Soc.* **26(1)**, 158–164.

[2] Adamatzky, A. and Bull, L. [2009] "Are complex systems hard to evolve?" *Complexity* **14(6)**, 15–20.

[3] Alonso-Sanz, R. [2006] "Elementary cellular automata with elementary memory rules in cells: the case of linear rules," *J. Cellular Automata* **1(1)**, 71–87.

[4] Alonso-Sanz, R. [2009] *Cellular Automata with Memory* (Old City Publishing).

[5] Alonso-Sanz, R. [2009] "Cellular automata with memory," *Encyclopedia of Complexity and Systems Science*, ed. Meyers, R. (Springer-Verlag, New York), pp. 823–848.

[6] Alonso-Sanz, R. [2011] *Discrete Systems with Memory* (World Scientific Series on Nonlinear Science, Series A, Vol. 75).

[7] Alonso-Sanz, R. [2013] "Elementary cellular automata with memory of delay type," Kari, J., Kutrib, M., Malcher, A. (Eds.), Lecture Notes in Computer Science **8155**, 67-83.

[8] Alonso-Sanz, R. [2014] "Reversible cellular automata with memory of delay type," *Complexity* **20(1)**, 49–56.

[9] Arbib, M. A. [1969] *Theories of Abstract Automata* (Prentice-Hall Series in Automatic Computation).

[10] Alonso-Sanz, R. and Adamatzky, A. [2008] "On memory and structural dynamism in excitable cellular automata with defensive inhibition," *Int. J. Bifurcation and Chaos* **18(2)**, 527–539.

[11] Boccara, N. [2004] *Modelling Complex Systems* (Springer-Verlag, New York).

[12] Chopard, B. and Droz, M. [1998] *Cellular Automata Modeling of Physical Systems* (Collection Aléa Saclay, Cambridge University Press).

[13] Culik II, K. and Yu, S. [1988] "Undecidability of CA Classification Schemes," *Complex Systems* **2**, 177–190.

[14] Hoekstra, A. G., Kroc, J. and Sloot, P. M. A. (eds.) [2010] *Simulating Complex Systems by Cellular Automata: Understanding Complex Systems* (Springer-Verlag, Berlin, Heidelberg).

[15] Hopcroft, J. E. and Ullman, J. D. [1987] *Introduction to Automata Theory Languages, and Computation* (Addison-Wesley Publishing Company).

[16] Martínez, G. J., Adamatzky, A., Alonso-Sanz, R. and Seck-Tuoh-Mora, J. C. [2010] "Complex dynamic emerging in Rule 30 with majority memory," *Complex Systems* **18(3)**, 345–365.

[17] Martínez, G. J., Adamatzky, A. and Alonso-Sanz, R. [2012] "Complex dynamics of elementary cellular automata emerging in chaotic rules," *Int. J. Bifurcation and Chaos* **22(2)**, 1250023-13.

[18] Martínez, G. J., Adamatzky, A. and Alonso-Sanz, R. [2013] "Designing Complex Dynamics in Cellular Automata with Memory," *Int. J. Bifurcation and Chaos* **23(10)**, 1330035-131.

[19] Martínez, G. J., Adamatzky, A. Chen, F. and Chua, L. [2012] "On Soliton Collisions between Localizations in Complex Elementary Cellular Automata: Rules 54 and 110 and Beyond," *Complex Systems* **21(2)**, 117–142.

[20] Martínez, G. J., Adamatzky, A. and McIntosh, H. V. [2006] "Phenomenology of glider collisions in cellular automaton Rule 54 and associated logical gates," *Chaos, Solitons and Fractals* **28**, 100–111.

[21] Martínez, G. J., Adamatzky, A. and McIntosh, H. V. [2014] "Complete Characterization of Structure of Rule 54," *Complex Systems* 32(3), 259–293.

[22] Martínez, G. J. [2013] "A Note on Elementary Cellular Automata Classification," *J. Cellular Automata* **8(3-4)**, 233–259.

[23] Martínez, G. J., Adamatzky, A., Seck-Tuoh-Mora, J. C. and Alonso-Sanz, R. [2010] "How to make dull cellular automata complex by adding memory: Rule 126 case study," *Complexity* **15(6)**, 34–49.

[24] Martínez, G. J., McIntosh, H. V. and Seck-Tuoh-Mora, J. C. [2006] "Gliders in Rule 110," *Int. J. Unconventional Computing* **2(1)**, 1–49.

[25] McIntosh, H. V. [1990] "Wolfram's Class IV and a Good Life," *Physica D* **45**, 105–121.

[26] McIntosh, H. V. [2009] *One Dimensional Cellular Automata* (Luniver Press).

[27] Minsky, M. [1967] *Computation: Finite and Infinite Machines* (Prentice Hall, Inc.).

[28] Morita, K. [1998] "Cellular Automata and Artificial Life," *Complex Systems*, eds. Goles, E. and Martinez, S. (Kluwer Academic Publishers), pp. 151–200.

[29] Rogozhin, Y. [1996] "Small Universal Turing Machines," *Theor. Comput. Sci.* **168(2)**, 215–240.

[30] Rogozhin, Y. and Verlan, S. [2006] "On the Rule Complexity of Universal Tissue P Systems," *Lecture Notes in Computer Science* **3850**, 356–362.

[31] Toffoli,T. [1984] "Cellular automata as an alternative to (rather than an approximation of) differential equations in modeling physics," *Physica D* **10**, 117–27.

[32] Voorhees, B. H. [1996] *Computational analysis of one-dimensional cellular automata* (World Scientific Series on Nonlinear Science, Series A, Vol. 15).

[33] Voorhees, B. H. [2008] "Remarks on Applications of De Bruijn Diagrams and Their Fragments," *J. Cellular Automata* **3(3)**, 187–204.

[34] von Neumann, J. [1966] *Theory of Self-reproducing Automata*, edited and completed by Burks, A. W., (University of Illinois Press, Urbana and London).

[35] Wuensche, A. and Lesser, M. [1992] *The Global Dynamics of Cellular Automata* (Addison-Wesley Publishing Company).

[36] Wolfram, S. [1983] "Statistical Mechanics of Cellular Automata," *Rev. Mod. Phys.* **55**, 601–644.

[37] Wolfram, S. [1984] "Universality and complexity in cellular automata," *Physica D* **10**, 1–35.

[38] Wolfram, S. [1984] "Computation Theory of Cellular Automata," *Commun. Math. Phys.* **96(1)**, 15–57.

[39] Wolfram, S. [1986] *Theory and Applications of Cellular Automata* (World Scientific Publishing Company, Singapore).

[40] Wolfram, S. [1994] *Cellular Automata and Complexity: Collected Papers* (Addison-Wesley Publishing Company).

[41] Wolfram, S. [2002] *A New Kind of Science* (Wolfram Media, Inc., Champaign, Illinois).

[42] Wuensche, A. [1999] "Classifying Cellular Automata Automatically," *Complexity* **4(3)**, 47–66.