Complex dynamics of elementary cellular automata emerging from chaotic rules

Genaro J. Martínez

Instituto de Ciencias Nucleares, and Centro de Ciencias de la Complejidad
Universidad Nacional Autónoma de México, México
genaro.martinez@uwe.ac.uk

Andrew Adamatzky

Bristol Institute of Technology, University of the West of England, Bristol, United Kingdom.
andrew.adamatzky@uwe.ac.uk

Ramon Alonso-Sanz

ETSI Agrónomos, Polytechnic University of Madrid, Madrid, Spain.
ramon.alonso@upm.es

We show techniques of analyzing complex dynamics of cellular automata (CA) with chaotic behaviour. CA are well known computational substrates for studying emergent collective behaviour, complexity, randomness and interaction between order and chaotic systems. A number of attempts have been made to classify CA functions on their space-time dynamics and to predict behaviour of any given function. Examples include mechanical computation, λ and Z-parameters, mean field theory, differential equations and number conserving features. We aim to classify CA based on their behaviour when they act in a historical mode, i.e. as CA with *memory*. We demonstrate that cell-state transition rules enriched with memory quickly transform a chaotic system converging to a complex global behaviour from almost any initial condition. Thus just in few steps we can select chaotic rules without exhaustive computational experiments or recurring to additional parameters. We provide analysis of well-known chaotic functions in one-dimensional CA, and decompose dynamics of the automata using majority memory exploring glider dynamics and reactions.

Keywords: Cellular automata, memory, complex dynamics, chaos, self-organization and filters

1. Introduction

In this paper we consider a simple tool to extract complex systems from a family of chaotic discrete dynamical system. We will employ a technique — *memory* based rule analysis [Alonso-Sanz & Martin, 2003], [Alonso-Sanz, 2008], [Alonso-Sanz, 2009], [Alonso-Sanz, 2009a], of using past history of a system to construct its present state and to manipulate its future.

We focus on one-dimensional CA. CA are well known computational substrates for studying emergent collective behaviour, complexity, randomness and interaction between order and chaos. A number of efforts have been made to classify CA functions on their space-time dynamics and to predict behaviour of any given function. Examples include mechanical computation, λ and Z-parameters, mean field theory, differential

equations and number conserving features. We aim to base CA classification on their behaviour in a historical mode, i.e. as *CA with memory* [Alonso-Sanz, 2008].

We study elementary CA (ECA) where each function evaluates a central cell with their two neighbourhoods (left and right) and every cell takes a value of its binary alphabet. ECA are introduced and extensively studied by [Wolfram, 1984], [Wolfram, 2002]. In ECA there is a set of functions determining global chaotic behaviour where global configurations are chaotic, many configurations have many ancestors, and attractors are dense [Wuensche & Lesser, 1992].

ECA is a one-dimensional array of finite automata, each automaton takes two states and updates its state in discrete time depending on its own state and states of its two closest neighbours, all cells update their state synchronously. A general classification of ECA was introduced in [Wolfram, 1994], as follows:

- class I. CA evolving to a homogeneous state.
- class II. CA evolving periodically.
- class III. CA evolving chaotically.
- class IV. Include all previous cases, as well known as class of *complex rules.*

In this classification class IV is of particular interest because the rules of the class exhibit non-trivial behaviour with rich diversity of patterns emerging and non-trivial interactions between travelling localizations, or gliders, e.g. ECA Rule 54 [Martínez *et al.*, 2006].

In present paper we aim to transform a chaotic evolution rule to a complex system by using memory

$$\text{chaotic ECA} \xrightarrow{\text{memory}} \text{complex ECA}$$

and derive a new classes of CA functions with historic evolution.

We believe that by employing historic evolution we are able to explore hidden properties of chaotic systems, and select chaotic rules with homogeneous dynamics.

2. Basic notation

2.1. *One-dimensional cellular automata*

One-dimensional CA is represented by an array of *cells* x_i where $i \in Z$ (integer set) and each x takes a value from a finite alphabet Σ. Thus, a sequence of cells $\{x_i\}$ of finite length n represents a string or *global configuration* c on Σ. This way, the set of finite configurations will be represented as Σ^n. An evolution is represented by a sequence of configurations $\{c_i\}$ given by the mapping $\Phi : \Sigma^n \to \Sigma^n$; thus their global relation is following

$$\Phi(c^t) \to c^{t+1} \tag{1}$$

where t is time steps and every global state of c is defined by a sequence of cell states. Also the cell states in configuration c^t are updated at the next configuration c^{t+1} simultaneously by a local function φ as follows

$$\varphi(x_{i-r}^t, \ldots, x_i^t, \ldots, x_{i+r}^t) \to x_i^{t+1}. \tag{2}$$

Following [Wolfram, 1984], [Wolfram, 2002] one can represents any CA with two parameters (k, r). Where $k = |\Sigma|$ is a number of states, and r is a radius of neighbourhood. Thus ECA are defined by parameters $(2, 1)$. There are Σ^n different neighbourhoods (where $n = 2r + 1$) and k^{k^n} different evolution rules.

In computer experiments we are using automata with periodic boundary conditions.

2.2. *Cellular automata with memory*

Conventional cellular automata are ahistoric (memoryless). A new state of a cell depends on the neighbourhood configuration solely at the preceding time step of φ (see Eq. 2).

CA with *memory* extends standard framework of CA by allowing every cell x_i to remember some period of its previous evolution [Alonso-Sanz, 2008].

Thus to implement a memory we design a memory function ϕ, as follows:

$$\phi(x_i^{t-\tau}, \ldots, x_i^{t-1}, x_i^t) \rightarrow s_i \tag{3}$$

such that $\tau < t$ determines the degree of memory backwards and each cell $s_i \in \Sigma$ is a state function of the series of states of the cell x_i with memory up to time-step. To execute the evolution we apply the original rule as follows:

$$\varphi(\ldots, s_{i-1}^t, s_i^t, s_{i+1}^t, \ldots) \rightarrow x_i^{t+1}.$$

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. Therefore cells *canalize* memory to the map φ.

For example, let us consider memory function ϕ as a *majority memory*:

$$\phi_{maj} \rightarrow s_i \tag{4}$$

where in case of a tie given by $\Sigma_1 = \Sigma_0$ in ϕ, we shall take the last value x_i. So ϕ_{maj} represents the classic majority function for three variables [Minsky, 1967], hence we have:

$$\phi_{maj} : (x_1 \wedge x_2) \vee (x_2 \wedge x_3) \vee (x_3 \wedge x_1) \rightarrow x$$

on cells $(x_i^{t-\tau}, \ldots, x_i^{t-1}, x_i^t)$ and define a temporal ring before calculating the next global configuration c. The representation of a ECA with memory [Martínez *et al.*, 2010] is given as follows:

$$\phi_{CARm:\tau} \tag{5}$$

where CAR is a decimal notation of a particular ECA rule and m the kind of memory given with a specific value of τ. Thus the majority memory (maj) working in ECA Rule 86 checking tree cells $(\tau = 3)$ of history is simply denoted as: $\phi_{R86maj:3}$. Figure 1 depicts in detail the memory effect working on ECA.
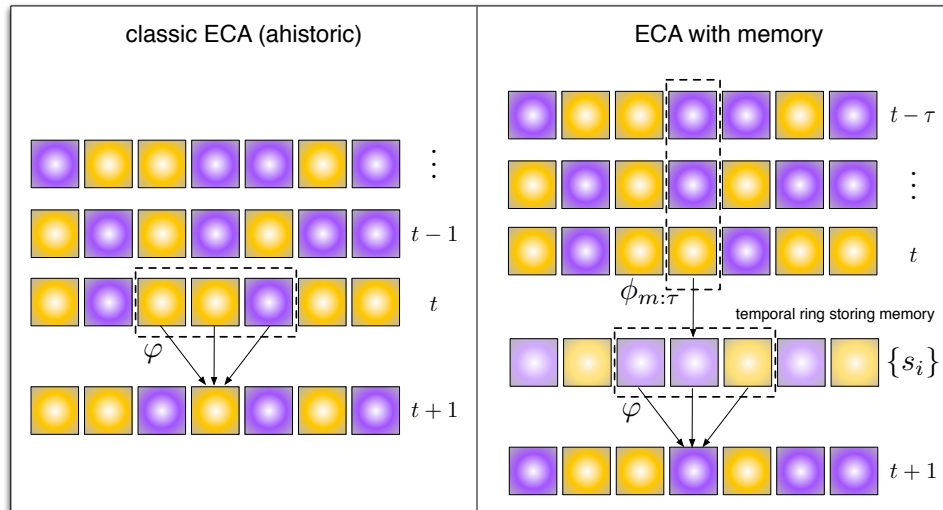

Fig. 1.   Cellular automata with memory in cells.

Note that memory is a simple function but its global behaviour Φ can be predicted from its local function ϕ and φ.

3. Classes of ECA by polynomials

3.1. *Mean filed approximation*

Mean field theory is a proven technique for discovering general statistical properties of CA without analyzing evolution spaces of individual rules [McIntosh, 2009].

The method assumes that elements of the set of states Σ are independent, uncorrelated between each other in the rule's evolution space φ. Therefore we can study probabilities of states in neighbourhood in terms of probability of a single state (the state in which the neighbourhood evolves), thus probability of a neighbourhood is the product of the probabilities of each cell in the neighbourhood.

In this way, it was proposed to explain Wolfram's classes by a mixture of probability theory and de Bruijn diagrams in [McIntosh, 1990], resulting in a classification based on mean field theory curve:

- class I: monotonic, entirely on one side of diagonal;
- class II: horizontal tangency, never reaches diagonal;
- class IV: horizontal plus diagonal tangency, no crossing;
- class III: no tangencies, curve crosses diagonal.

Thus for one dimension all cell neighbourhoods must be considered as:

$$p_{t+1} = \sum_{j=0}^{k^{2r+1}-1} \varphi_j(X) p_t^v (1 - p_t)^{n-v} \tag{6}$$

such that j is a number of relations from neighbourhoods and $X \in \Sigma$ represent of cells $x_{i-r}, \ldots, x_i, \ldots, x_{i+r}$. Thus n represents the number of cells in neighborhood, v indicates how often state one occurs in the neighborhood, $n-v$ shows how often state zero occurs in the neighborhood, p_t is a probability of cell being in state one, q_t is a probability of cell being in state zero (such that $q = 1 - p$).

4. Complex dynamics emerging from chaotic ECA

4.1. *Chaotic ECA*

Let us consider two cases of classic ECA with chaotic behaviour to demonstrate our results: the evolution rules 86 and 101.

We need to provide their mean filed approximation to verify that both function have a chaotic global behaviour before selecting the memory.

The local rule φ corresponding to rule 86 is following:

$$\varphi_{R86} = \begin{cases} 1 \text{ if } 110, 100, 010, 001 \\ 0 \text{ if } 111, 101, 011, 000 \end{cases}.$$

Initially φ_{R86} has produces states zero and one with the same probability. There is an equilibrium of states in Φ. On the other hand, φ_{R86} determines a surjective correspondence and therefore every configuration has at least one ancestor and no Garden of Eden configurations [Amoroso & Cooper, 1970]. Of course this rule is the reflection of well-known ECA rule 30 [Wuensche & Lesser, 1992].

The local function for rule 101 is following:

$$\varphi_{R101} = \begin{cases} 1 \text{ if } 110, 101, 010, 000 \\ 0 \text{ if } 111, 100, 011, 001 \end{cases}.$$

In this case, φ_{R101} has the same probability as φ_{R86} to produce states one and zero. However φ_{R101} is not a surjective rule and therefore has the Garden of Eden configurations, i.e., not all configurations have ancestors.

To classify global behaviour properly of φ_{R86} and φ_{R101} we should calculate their mean field polynomials. Mean field polynomial for φ_{R86} is:
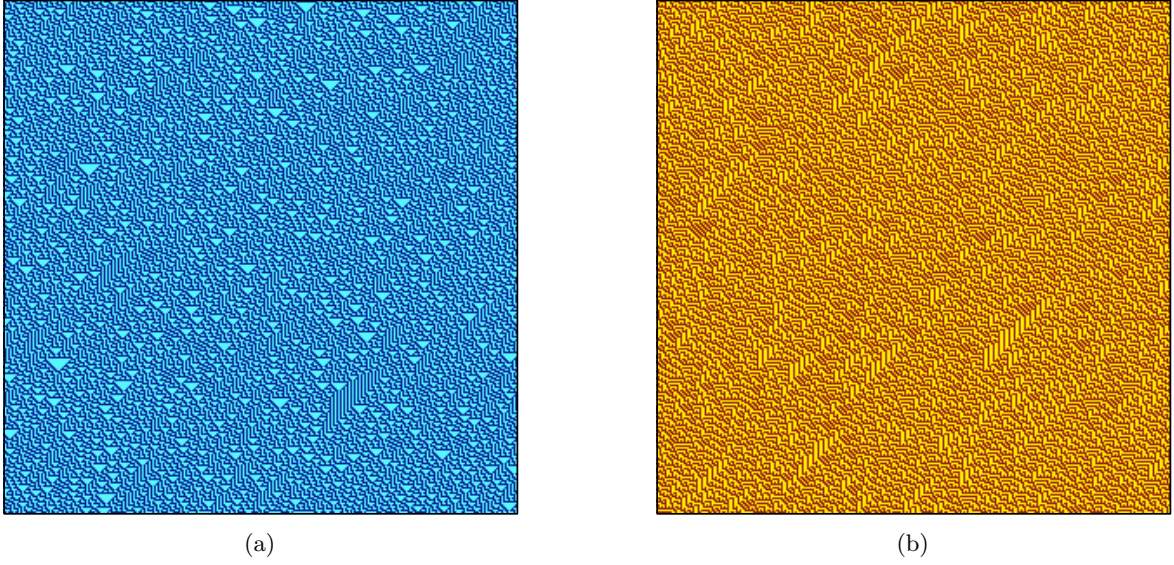
Fig. 2. Chaotic global behaviour in ECA evolution rules (a) φ_{R86} and (b) φ_{R101} evolving over an array of 295 cells in 295 generations. Both evolutions start in configurations with the same random initial density of 50%.

$$p_{t+1} = 3p_t q_t^2 + p_t^2 q_t \tag{7}$$

and for φ_{R101} we have:

$$p_{t+1} = 2p_t^2 q_t + p_t q_t^2 + q_t^3. \tag{8}$$

The polynomial for φ_{R86} satisfies the mean field classification (Sect. 3). Where rules in CA class III do not have tangencies and therefore the curve crosses the identity. Consequently, φ_{R86} evolves with a chaotic global behaviour (see Fig. 3(a)).
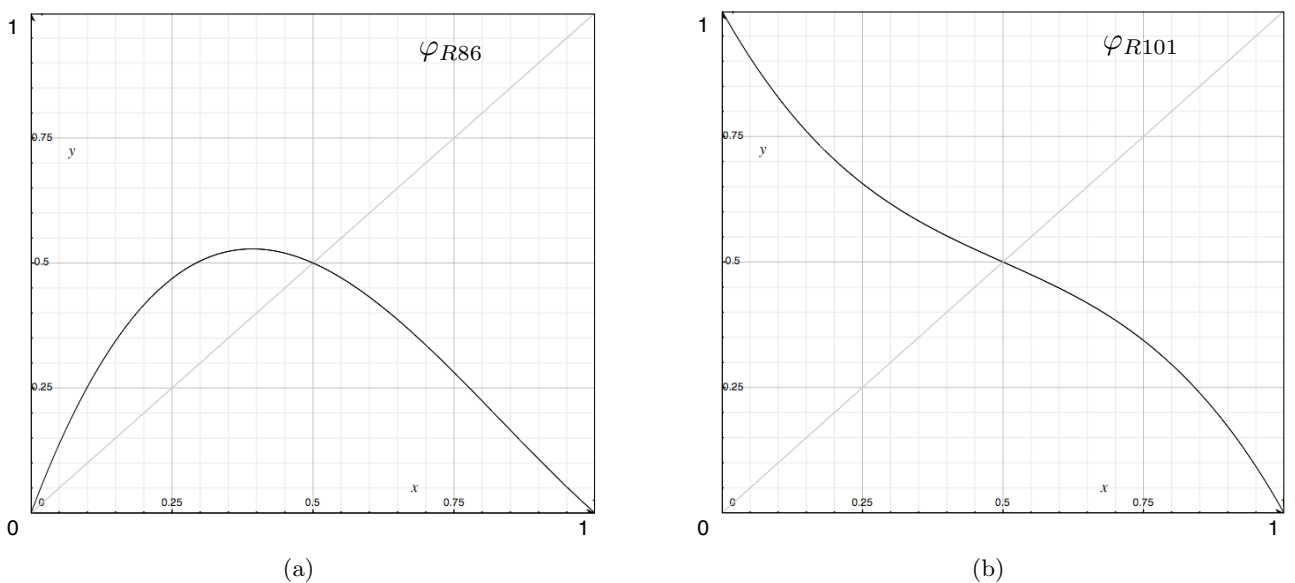

Fig. 3. Mean field curves for (a) φ_{R86} and (b) φ_{R101} respectively.

This mean field polynomial has an stable fixed point when Eq. 7 is $f = 0.5$. This value relate the existence of densities where the population of cells in state one is preserved with few changes. Also such fixed point confirm its initial probability since φ_{R86}. Of course, if there are extreme densities of zeros and ones then next time the configuration will be filled of states zeros only, a homogeneous global state.

Mean field curve for φ_{R101} (see Fig. 3(b)) presents another characteristic. Again the curve does not cross the identity and its global behaviour Φ should then be chaotic. Its stable fixed point $f = 0.5$ relates to the initial probability estimated since φ_{R101}. The curve displays what would happen if some initial configuration c_0 is dominated by state one, at the next step will be dominated by states zero and therefore this behaviour should repeat periodically. Such phenomenon also is balanced with its 50% of density to each step.

Finally Fig. 2 displays two evolutions with typical chaotic behaviour in ECA. First evolution (a) displays the chaotic global evolution of φ_{R86} since a random initial condition with a 50% of density. That confirm an evolution without some order or pattern defined. Second evolution (b) displays the chaotic global behaviour for φ_{R101} with the same parameters.

Now we will select a kind of memory and uncover "hidden" properties of chaotic ECA's.

4.2. *Filtering evolutions*

Filters selected in CA are a useful tool for understand "hidden" properties of CA. This tool was amply developed by Wuensche in a context of automatic classification of CA. The filters were derived from mechanical computation techniques [Hanson & Crutchfield, 1997] and analysis of cell-state frequencies [Wuensche, 1999].

Others derivations deducing filters relate as tiling, were reported for ECA rule 110 [Martínez *et al.*, 2006], and rule 54 [Martínez *et al.*, 2006]. However in general such filters are not widely exploited in CA studies. We consider the tile representation to identify filters as block of cells in one or two dimensions. We explain each tile filtering φ_{R86} and φ_{R101} in the Sec. 4.3.

4.3. *Complex dynamics emerging from φ_{R86} and φ_{R101} with majority memory*

Firstly we should consider a kind of memory, in this case the majority memory ϕ_{maj} (see Eq. 4) and then a value for τ. This value represents the number of cells backward to consider in the memory (as we saw in Sec. 2.2).

Implementing the majority memory ϕ_{maj} we can select some ECA and experimentally explore its effect. Figures 4 and 5 show outcomes of selecting memory τ working on φ_{R86} and φ_{R101} respectively. The result is a new family of ECA but now with majority memory, they are the rules: $\phi_{R86maj:3}, \ldots, \phi_{R86maj:\infty}$, and $\phi_{R101maj:3}, \ldots, \phi_{R101maj:\infty}$.

As a characteristic while the memory is working on ϕ_{R86maj} and $\phi_{R101maj}$ a periodic background was more evident and it can be represented as a tile. These filters work as well on the original rules φ, see Fig. 4 and 5.

The memory effect produces an emergency of patterns. The patterns interact quickly (in time-scale of CA development) with each other. In fact, for ϕ_{R86maj} and some values of τ the original behaviour changes its dynamics dramatically. Following our previous findings [Martínez *et al.*, 2010], in press] we consider only even values that offer better global dynamics. Thus the new rule $\phi_{R86maj:8}$ displays particles travelling in different velocities on a periodic background (see Fig. 4).

The second case $\phi_{R101maj}$ displays more attractive result. These three rules support stationary and mobile particles, travelling and colliding, some collisions can be interpreted as solitonic reaction [Adamatzky, 2002] (see Fig. 5).

Also on all evolutions a filter was selected to clarify evolutions and patterns.[1] Filters really are useful to recognize periodic dominant patterns of objects moving into such local universes.

---

[1] All evolutions simulated to ECA and ECA with memory they are calculated with OSXLCAU21 system, available from http://uncomp.uwe.ac.uk/genaro/OSXCASystems.html
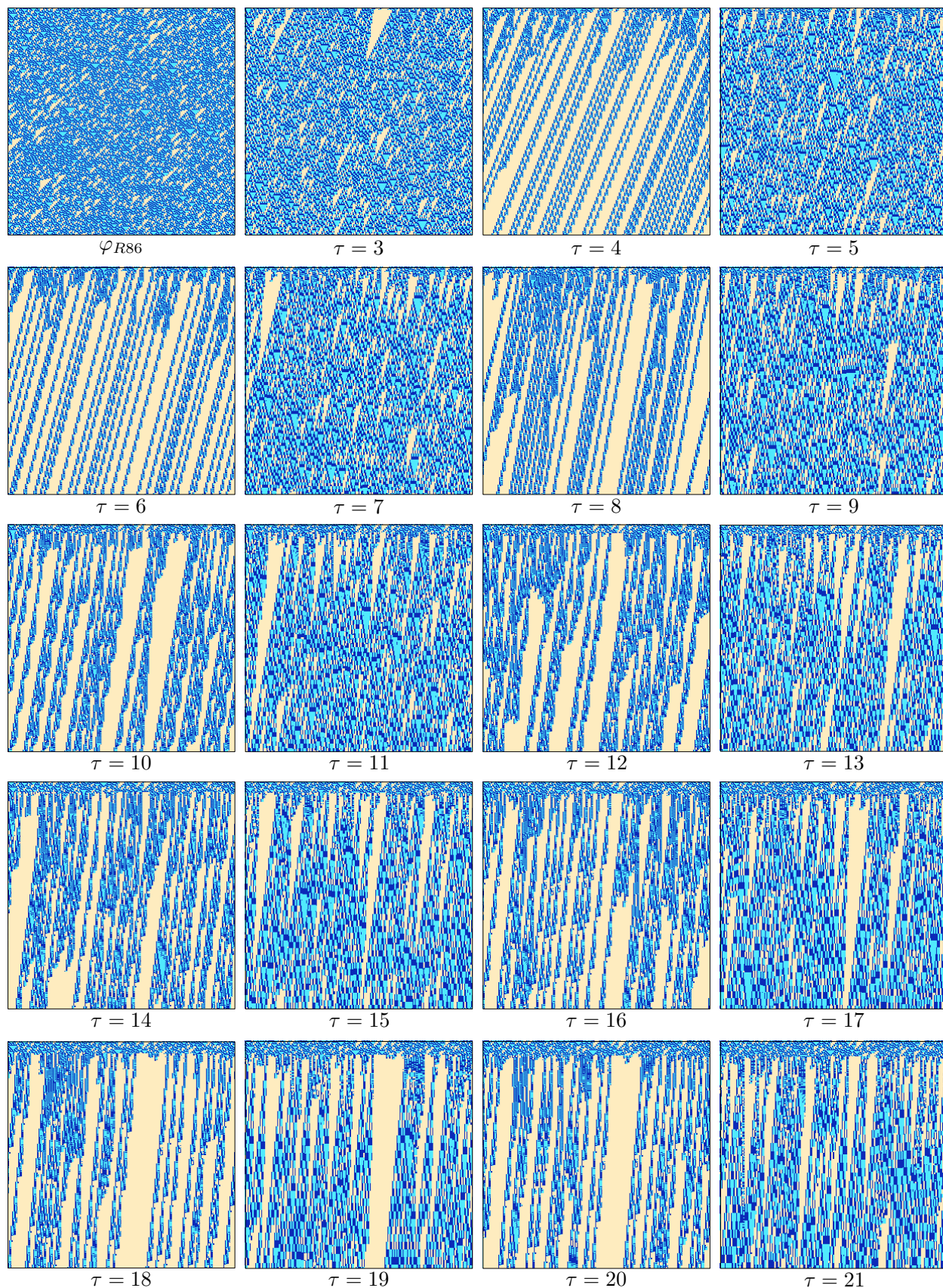
Fig. 4. Majority memory $\phi_{R86maj:\tau}$ working in φ_{R86} with τ values of 3 to 21, respectively. The first one is the original ECA rule 86 evolution. All snapshots evolve with the same random initial condition of 50% over an array of 300 cells to 300 generations, all evolutions are filtered.
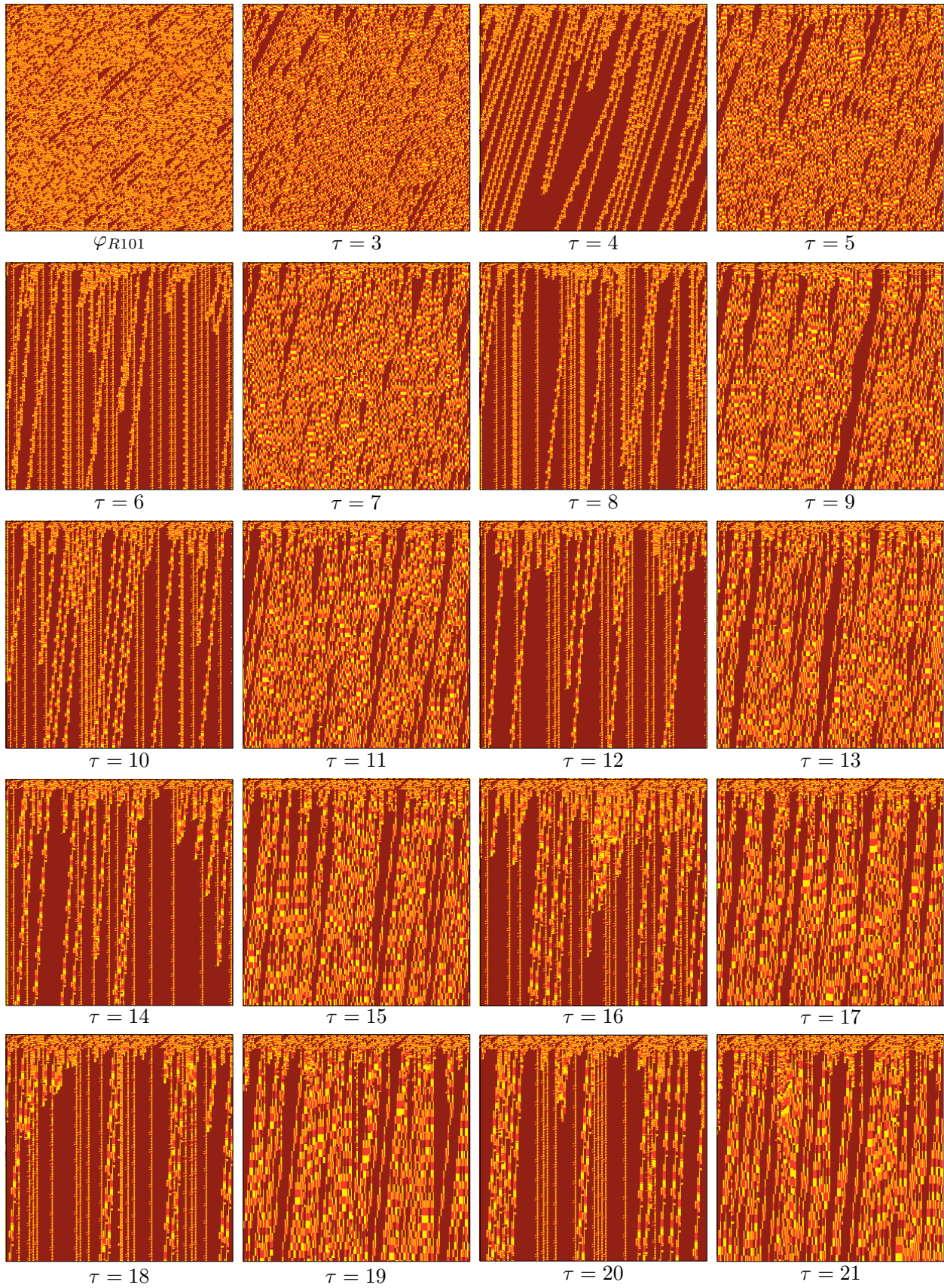
Fig. 5.   Majority memory $\phi_{R101maj:\tau}$ working in φ_{R101} with τ values of 3 to 21, respectively. The first one is the original ECA rule 101 evolution. All snapshots evolve with the same random initial condition to 50% over an array of 300 cells to 300 generations, all evolutions are filtered.

The first two-dimensional tile working in ϕ_{R86maj} is represented as $t_{\phi_{86}} = \begin{bmatrix} 101 \\ 101 \end{bmatrix}$. Also this tile works on the original evolution rule as shows the Fig. 4. Tile reported for $\phi_{R101maj}$ is determined for the two-dimensional tile $t_{\phi_{101}} = \begin{bmatrix} 100 \\ 100 \end{bmatrix}$. So this filter works on the original evolution rule as well as shows the Fig. 5.

The effect of memory producing new evolution rules is preserved in some way. Initially the existence of a filter that can evolve on all different function, that is not rare because the memory only read the history and process the new generation with the original rule.

4.4. *Coding particles*

4.4.1. *Self-organization by structure formation*

Patterns as particles and non-trivial behaviour emerging in these new CA with memory ϕ_{R86maj} and $\phi_{R101maj}$, naturally conduce to well-know problems as self-organization.
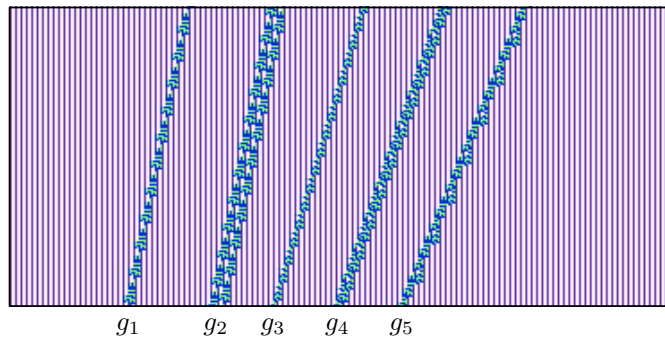

Fig. 6. Set of particles $\mathcal{G}$ emerging and living in $\phi_{R101maj:4}$.

Considering the evolution rule $\phi_{R101maj:4}$, we have done a classification of particles in this local universe (see Fig. 6). The universe is not bigger compared with other complex rules. However that all particles in $\phi_{R101maj:4}$ can be produced from other particles in binary collisions. Such self-organization by structure formation [Kauffman, 1993] is demonstrated in the following set of reactions between particles:

$$g_i \rightarrow g_j = g_k$$

such that $i \neq j \neq k$ and $i, j, k \in \mathcal{G}_{\phi_{R101maj:4}}$.

Figure 7 presents the set of reactions necessary to produce every particle:

a) $g_4 - b^3 - g_5 = g_1$
b) $g_1 - b^2 - g_4 = g_2$
c) $g_1 - b^4 - g_4 = g_3$
d) $g_1 - b^6 - g_5 = g_4$
e) $g_3 - b^3 - g_4 = g_5$
f) $g_3 - b^2 - g_4 = \emptyset$

Of course, they are not all possibilities to get every particle and a organization of several particles could be produce even more complex behaviour imitating physical, biological, chemical or computational phenomena: wave propagation, reaction-diffusion, morphogenesis, particle collision, fluid-dynamics, (tissue) grown, pattern formation, self-reproduction, self-assembly, artificial life, synthetic constructions (engineering), tessellation, differential equations, soliton solutions, formal languages, or unconventional computing [Adamatzky, 2002], [Bar-Yam, 1997], [Margenstern, 2007], [Mitchell, 2009], and [Morita, 1998].
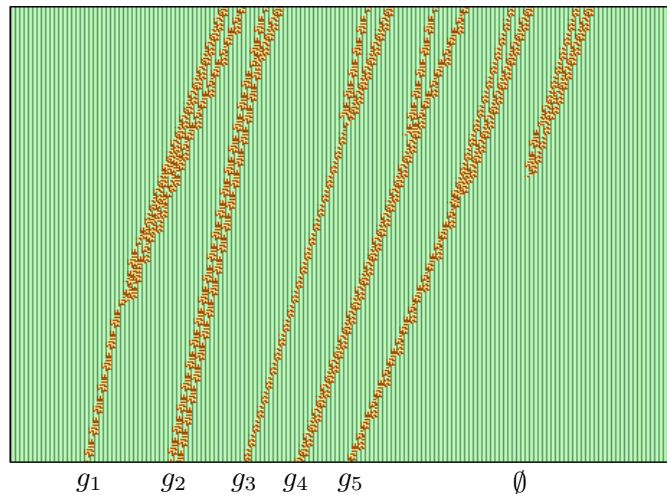
Fig. 7.   Self-organization by particle collisions. The set $\mathcal{G}_{\phi_{R101maj:4}}$ is produced. CA evolution is filtered.
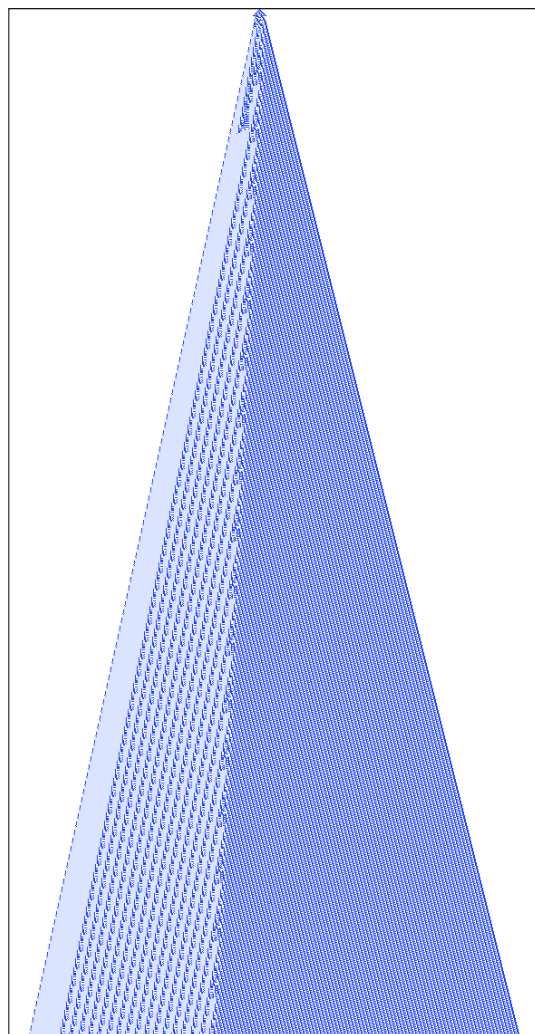

Fig. 8.   Stream of particles and fuse patterns emerging from a single cell in state 1 with $\phi_{R86maj:8}$. These patterns exhibit unlimited growth.
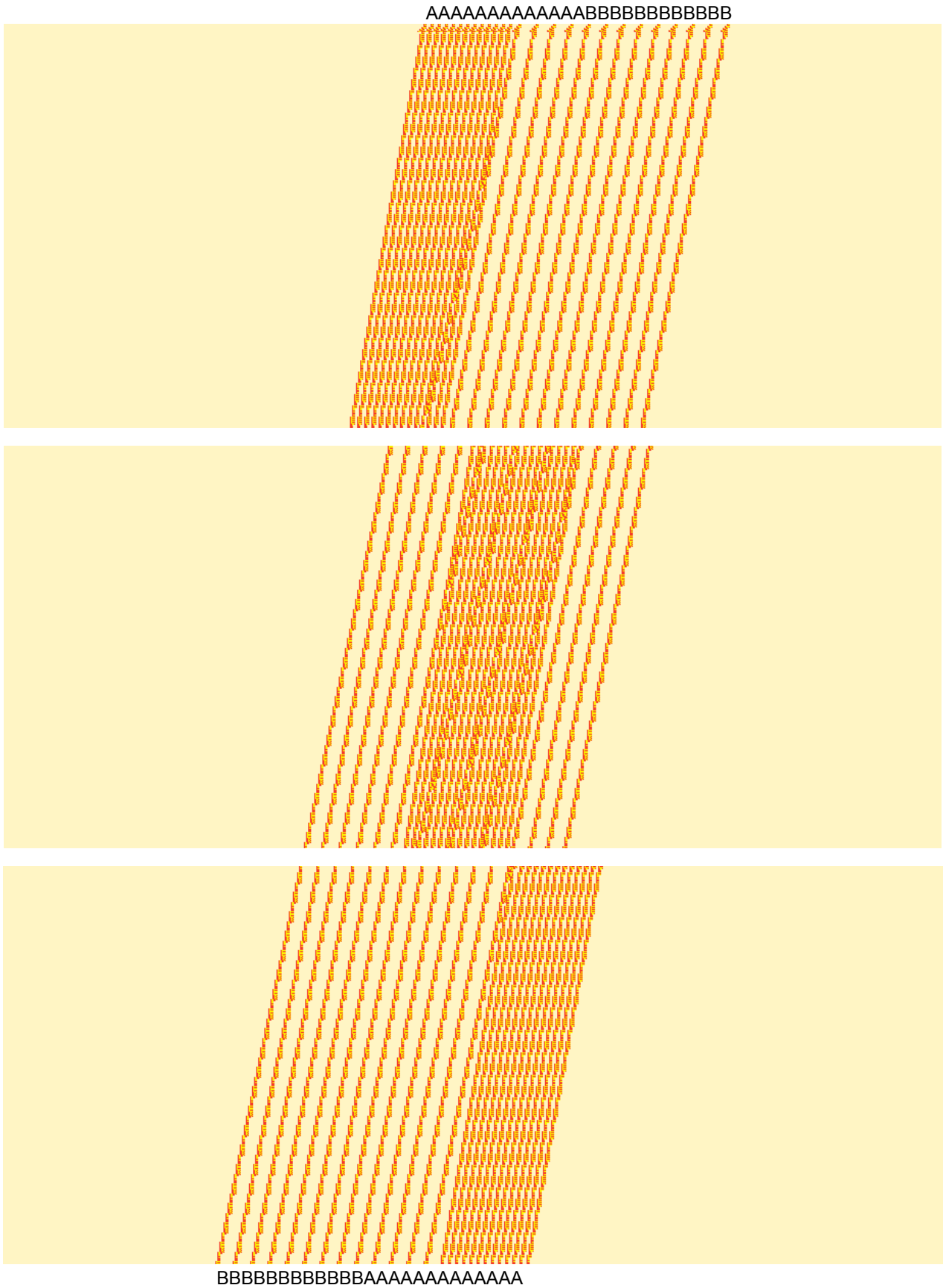
AAAAAAAAAAAAAABBBBBBBBBBBB

BBBBBBBBBBBBBAAAAAAAAAAAAA

Fig. 9.   A simple substitution system processing the word $A^{12}B^{12}$ to $B^{12}A^{12}$ with $\phi_{R86maj:8}$. The final production is reached on 6,888 generations by synchronization of multiple soliton reactions.

4.4.2. *Generator pattern*

Figure 8 shows the evolution of CA from a single cell in state 1, the automaton is governed by rule $\phi_{R86maj:8}$. A fuse pattern is organized by stream of gliders (left) emitted periodically every 62 steps and a fixed periodic pattern (right) growing with a velocity of $-\frac{1}{4}$.

The above examples of CA with memory are just two simple cases showing the memory effect on traditional chaotic functions. Another cases were developed for the ECA rules 30 and 126 in [Martínez *et al.*, 2010].

4.4.3. *Implementing basic computing functions*

We can employ the particles codification to represent solutions of some basic computing functions. Let us consider the rule $\phi_{R86maj:8}$. We want to implement a simple substitution function addToHead working on two strings $w_1 = A_1, \ldots, A_n$ and $w_2 = B_1, \ldots, B_m$, where $n, m \geq 1$. For example, if $w_1 = AAA$, $w_2 = BBB$ and $w_3 = w_1 w_2$ then the addToHead($|w_2|$) will yield: $w_3 = w_2 w_1$ (see schematic diagram of Fig. 10).

To implement such function in $\phi_{R86maj:8}$ we must represent every data 'quantum' as a particle. Gliders g_1 and g_2 are coded to reproduce a soliton reaction.[2] Another problem is synchronize several gliders and obtain the same result with multiple collisions.

The codification is not sophisticated however a systematic analysis of reactions is required. We known than a periodic gap and one fixed phase between particles is sufficient to reproduce the addToHead function for any string $A^n B^m$.
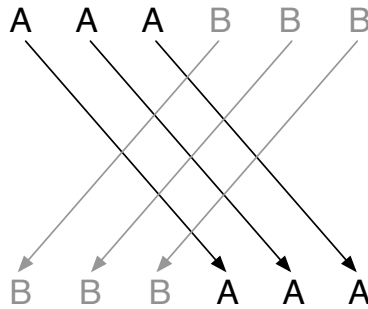

Fig. 10.   Schematic diagram adding the string w_2 to head of the list w_3.

Figure 9 shows fragments of evolutions of $\phi_{R86maj:8}$ from an initial condition coded by gliders, representing the string $AAAAAAAAAAAABBBBBBBBBBBB$. Using function addToHead we produce the final string $BBBBBBBBBBBBAAAAAAAAAAAA$ after 6,888 generations. The first snapshot in Fig. 9 shows its initial configuration and the first 400 steps, the middle snapshot mainly presents how the string w_1 across the string w_2 preserving the information (soliton reaction), and the third snapshot shows the final global configuration so given the string $w_2 w_1$ processed in parallel with $\phi_{R86maj:8}$.

5. Discussion

We have demonstrated that elementary cellular automata (ECA) with memory offer a powerful approach to discovering complex dynamics based on particles and non-trivial reactions between the particles. Such problem has been substantiated by a number of different techniques, e.g. number-conservation [Boccara & Fukś, 2002], [Imai *et al.*, 2004], exhaustive search [Eppstein, 2002], tiling [Martínez *et al.*, 2006], [Margenstern, 2007], de Bruijn diagrams [Martínez *et al.*, 2008], Z-parameter [Wuensche, 1999], genetic algorithms [Das *et al.*, 1994], mean field theory [McIntosh, 1990] or from a differential equations point view [Chua,

---

[2]These gliders are a reflection of $\phi_{R30maj:8}$, because ECA rule 86 is the reflection of rule 30, and consequently their gliders emerging with memory can be coded in a similar way [Martínez *et al.*, 2010].

2007]. Thus the memory function ϕ offers a more easy way to get similar and, in some cases, more strong results reporting new complex rules in ECA with memory.

We have enriched some classic chaotic ECA rules with majority memory and demonstrated that by applying certain filtering procedures we can extract rich dynamics of travelling localizations. Therefore, we can deduce a relation on chaotic systems decomposed in complex dynamics as a self-contained set. Generally a relation of sets of complex dynamics can be self-contained describing Φ as attractors, like a set diagram (Fig. 11 [Adamatzky *et al.*, 2006]).
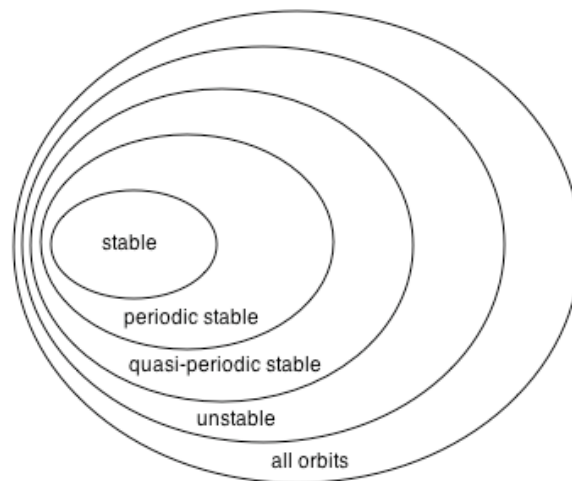

Fig. 11.   Classes of global behaviour.

This way, the most bigger set in Fig. 11 'all orbits' corresponds to complex dynamics and the 'unstable' set represents the chaotic systems. Indeed there is a number of properties between orbits and characteristics that cannot be inferred directly. However the memory plays a role of a powerful tool to discover such properties. Finally, the memory function ϕ can be applied to any CA or dynamical system.

Acknowledgments

References

Amoroso, S. & Cooper, G. [1970] "The Garden-of-Eden theorem for finite configurations," *Proceedings of the American Mathematical Society* **44**, pp. 189–197.

Adamatzky, A. [1994] *Identification of Cellular Automata*, (Taylor and Francis, London).

Adamatzky, A. (ed.) [2002] *Collision-Based Computing*, (Springer-Verlag London).

Alonso-Sanz, R. [2006] "Elementary rules with elementary memory rules: the case of linear rules," *Journal of Cellular Automata* **1**, pp. 71–87.

Alonso-Sanz, R. [2008] *Cellular Automata with Memory*, (Old City Publishing, Inc.).

Alonso-Sanz, R. [2009] "Cellular automata with memory," In *Encyclopedia of Complexity and Systems Science*, ed. Meyers, R., (Springer NY).

Alonso-Sanz, R. [2009] "Spatial order prevails over memory in boosting cooperation in the iterated prisoner's dilemma," *Chaos* **19**(2), p. 023102.

Alonso-Sanz, R. & Martin, M. [2003] "Elementary CA with memory," *Complex Systems* **14**, pp. 99–126.

Adamatzky, A., Martínez, G. J. & Seck-Tuoh-Mora, J. C. [2006] "Phenomenology of reaction-diffusion binary-state cellular automata," *Int. J. Bifurcation and Chaos* **16**(10), pp. 1–21.

Bar-Yam, Y. [1997] *Dynamics of Complex Systems*, (Perseus Books).

Boccara, N. & Fukś, H. [2002] "Number-Conserving Cellular Automaton Rules," *Fundamenta Informaticae* **52**, pp. 1–13.

Chua, L. O. [2007] *A Nonlinear Dynamics Perspective of Wolfram's New Kind of Science*, (World Scientific Publishing Company).

Das, R., Mitchell, M. & Crutchfield, J. P. [1994] "A genetic algorithm discovers particle-based computation in cellular automata," *Lecture Notes in Computer Science* **866**, pp. 344–353.

Eppstein, D. [2002] "Searching for spaceships," *MSRI Publications* **42**, pp. 433–452.

Hanson, J. E. & Crutchfield, J. P. [1997] "Computacional Mechanics of Cellular Automata: An Example," *Physics D* **103**, pp. 169–189.

Imai, K., Ikazaki, A., Iwamoto, C. & Morita, K. [2004] "A logically universal number-conserving cellular automaton with a unary table-lookup function," *IEICE Trans. on Information and Systems* **E87-D**, pp. 694–699.

Kauffman, S. A. [1993] *The Origins of Order: Self-Organization and Selection in Evolution*, (Oxford University Press, New York).

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

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

Martínez, G. J., Adamatzky, A. & McIntosh, H. V. [2008] "On the representation of gliders in Rule 54 by de Bruijn and cycle diagrams," *Lecture Notes in Computer Science* **5191**, pp. 83–91.

Margenstern, M. [2007] *Cellular Automata in Hyperbolic Spaces*, Old City Publishing, Inc.

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

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

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

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

Mitchell, M. [2009] *Complexity: A Guided Tour*, (Oxford University Press).

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

Mitchell, M. & Newman, M. [2002] "Complex Systems Theory and Evolution," In *Encyclopedia of Evolution*, ed. Pagel, M. (Oxford University Press).

Morita, K. [1998] "Cellular Automata and Artificial Life," *6th. Summer School on Complex Systems*.

Robinson, C. [1994] *Dynamical Systems: stability, symbolic dynamics and chaos*, (CRC Press).

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

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

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

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

Wuensche, A. [1994] "Complexity in one-d cellular automata," *Santa Fe Institute* working paper 94-04-025.

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