

A Structurally Dynamic Cellular Automaton with Memory in the Triangular Tessellation

Ramón Alonso-Sanz*

ETSI Agrónomos (Estadística),
C. Universitaria. 28040, Madrid, Spain

The major features of conventional cellular automata include the inalterability of topology and the absence of memory. The effect of simple memory (memory in cells and links) on a particular reversible, structurally dynamic cellular automaton in the triangular tessellation is explored in this paper.

1. A triangular cellular automaton

Cellular automata (CAs) are discrete, spatially explicit extended dynamic systems. A CA system is composed of adjacent cells or sites arranged as a regular lattice, which evolves in discrete time steps. Each cell is characterized by an internal state whose value belongs to a finite set. The updating of these states is made simultaneously according to a common local transition rule involving only the neighborhood of each cell. Thus, if $\sigma_i^{(T)}$ is taken to denote the value of cell i at time step T , the site values evolve by iteration of the mapping: $\sigma_i^{(T+1)} = \phi(\sigma_j^{(T)} \in \mathcal{N}_i)$, where ϕ is an arbitrary function which specifies the CA rule operating on the neighborhood \mathcal{N} of the cell i [1].

This paper deals with a particular two-dimensional totalistic CA rule, the *parity* rule: $\sigma_i^{(T+1)} = \sum_{j \in \mathcal{N}_i} \sigma_j^{(T)} \bmod 2$, acting on cells with two possible state values (0 and 1). Despite its formal simplicity, the parity rule exhibits complex behavior [2]. We restrict ourselves here to the triangular tessellation, again a simple scenario that allows for complex behavior [3, 4]. Figure 1 shows an example of the parity rule operating on the triangular tessellation starting from an active cell with its three neighbors also active. This is the initial configuration used throughout this paper.

2. Cellular automata with memory

Standard CA are ahistoric (memoryless): the transition function depends on the neighborhood configuration of the cells only at the preceding

*Electronic mail address: ramon.alonso@upm.es.

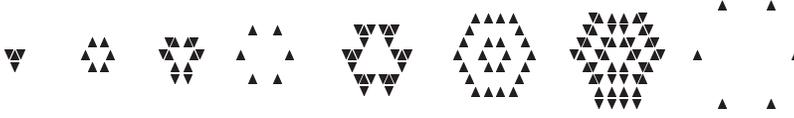


Figure 1. A triangular CA with the parity rule. Evolving patterns up to $T = 8$.

time step. Historic memory can be embedded in the CA dynamics by featuring every cell with a mapping of its states in the previous time steps. Thus, what is here proposed is to maintain the transition rules (ϕ) unaltered, but make them act on the cells featured by a function of their previous states: $\sigma_i^{(T+1)} = \phi(s_i^{(T)} \in \mathcal{N}_i)$, with $s_i^{(T)}$ being a state function of the series of states of the cell i up to time step T .

Thus, cells can be featured by a weighted mean value of their previous states:

$$m_i^{(T)}(\sigma_i^{(1)}, \sigma_i^{(2)}, \dots, \sigma_i^{(T)}) = \frac{\sigma_i^{(T)} + \sum_{t=1}^{T-1} \alpha^{T-t} \sigma_i^{(t)}}{1 + \sum_{t=1}^{T-1} \alpha^{T-t}} \equiv \frac{\omega_i^{(T)}}{\Omega(T)}, \quad (1)$$

and the s values are obtained by rounding the m values: $s_i^{(T)} = \text{round}(m_i^{(T)})$, with $s_i^{(T)} = \sigma_i^{(T)}$ if $m_i^{(T)} = 0.5$. Memory becomes operative after $T = 3$, with the initial assignments $s_i^{(1)} = \sigma_i^{(1)}$, $s_i^{(2)} = \sigma_i^{(2)}$.

In the two-state scenario, geometrically discounted memory does not have an effect if $\alpha \leq 0.5$, but if $\alpha \geq 0.61805$, cells with state history 001 or 110 will be featured after $T = 3$ as 0 and 1 respectively instead of 1 and 0 (last states), and the patterns of the ahistoric and historic models typically diverge from $T = 4$. This is so in Figure 2, in which case the pattern of s state values after $T = 3$ coincides with that at $T = 2$, and consequently the pattern at $T = 3$ is repeated at $T = 4$. In Figure 2, $\alpha = 0.6$ memory leads to extinction at $T = 5$, and $\alpha = 0.7$ memory at $T = 8$, but if $\alpha \geq 0.8$, a period-two oscillator appears as early as at $T = 4$. This is also true in the particular case of full memory: $\alpha = 1.0$, in which case $s_i^{(T)} = \text{mode}(\sigma_i^{(1)}, \dots, \sigma_i^{(T)})$.

The effect of memory in cells on CA has been studied in the references by Alonso-Sanz *et al.* As a general rule, geometrically discounted memory tends to truncate the expansive evolution of the parity rule, particularly at high values of the memory factor α in which case small oscillators tend to appear. But the effect is also dramatic at low values of α : the progression in size turns out to be restrained (leading to extinction in some cases) and the aspect of the patterns differs notably from that of the ahistoric ones.

Note that the memory mechanism here adopted is accumulative in its demand of knowledge of past history: to calculate the memory *charge*

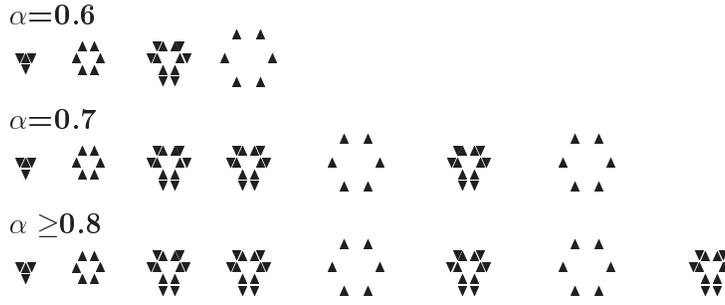


Figure 2. Effect of memory on the parity rule starting as in Figure 1 with memory factor α .

$\omega_i^{(T)}$ stated in [1], it is not necessary to know the whole $\{\sigma_i^{(t)}\}$ series, while it suffices to (sequentially) proceed as: $\omega_i^{(T)} = \alpha\omega_i^{(T-1)} + \sigma_i^{(T)}$.

Let us point out here that the implementation of memory adopted in this work, keeping the transition rule unaltered but applying it to a function of previous states, can be adopted in any dynamical system. In earlier work [8–10] we explored the effect of embedding this kind of memory into *discrete dynamical systems*: $x_{T+1} = f(x_T)$ by means of $x_{T+1} = f(m_T)$ with m_T being a mean value of past states. We have studied this approach in what is perhaps the canonical example: the logistic map, which becomes with memory $x_{T+1} = m_T + \lambda m_T(1 - m_T)$. In [7], we studied the effect of memory in a particular markovian stochastic process (the random walk), $\mathbf{p}'_{T+1} = \mathbf{p}'_T \mathbf{M}$ by means of $\mathbf{p}'_{T+1} = \boldsymbol{\pi}'_T \mathbf{M}$ with $\boldsymbol{\pi}'_T$ being a weighted mean of the probability distributions up to T . Returning to the CA scenario, memory can be embedded in continuous CA (or coupled map lattices), in which case the state variable ranges in \mathbb{R} , so $\sigma_i^{(T+1)} = \varphi(\sigma_j^{(T)} \in \mathcal{N}_i^{(T)})$, where φ is a continuous function. The formulation with memory will be: $\sigma_i^{(T+1)} = \varphi(m_j^{(T)} \in \mathcal{N}_i^{(T)})$, as illustrated in a one-dimensional example in [15].

3. Reversible cellular automata with memory

It should be emphasized that the memory mechanism considered here is different from that of other CA with memory reported in the literature. Typically, higher-order-in-time rules incorporate memory into the transition rule. Thus, in second-order-in-time rules, the transition rule operates as: $\sigma_i^{(T+1)} = \Phi(\sigma_j^{(T)} \in \mathcal{N}_i, \sigma_j^{(T-1)} \in \mathcal{N}_i)$. Particularly interesting is the reversible formulation based on the subtraction modulo two (noted \ominus): $\sigma_i^{(T+1)} = \phi(\sigma_j^{(T)} \in \mathcal{N}_i) \ominus \sigma_i^{(T-1)}$, reversed as $\sigma_i^{(T-1)} = \phi(\sigma_j^{(T)} \in \mathcal{N}_i) \ominus \sigma_i^{(T+1)}$. Figure 3 shows the evolving patterns from the reversible formulation of the example from Figure 1. As a rule, the pattern at $T = 0$ in the reversible simulations here is the same as that at $T = 1$.



Figure 3. A reversible triangular CA with the parity rule starting as in Figure 1.

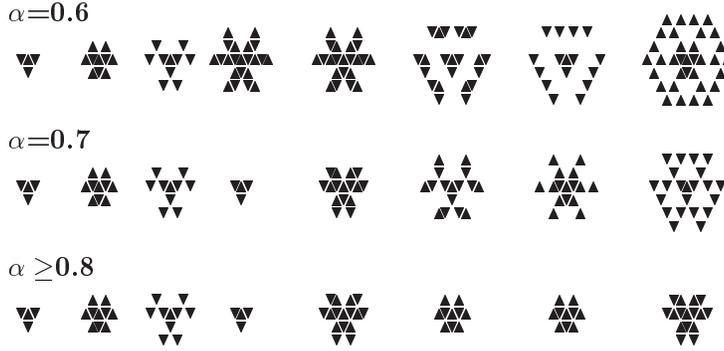


Figure 4. Effect of memory in the reversible triangular CA shown in Figure 3.

To preserve reversibility, the reversible formulation with memory must be: $\sigma_i^{(T+1)} = \phi(s_i^{(T)} \in \mathcal{N}_i) \ominus \sigma_i^{(T-1)}$ [12]. Figure 4 shows an example starting as in Figure 3. The general considerations regarding the inertial effect of memory in the irreversible scenario apply in the reversible implementation. Thus starting as in Figure 3 but with full memory, a period-four oscillator (that shown in Figure 4) appears at $T = 5$.

For reversing from T it is necessary to know not only $\sigma_i^{(T)}$ and $\sigma_i^{(T+1)}$ but also $\omega_i^{(T)}$ to be compared to $\Omega(T)$, to obtain:

$$s_i^{(T)} = \begin{cases} 0 & \text{if } 2\omega_i^{(T)} < \Omega(T) \\ \sigma_i^{(T+1)} & \text{if } 2\omega_i^{(T)} = \Omega(T) \\ 1 & \text{if } 2\omega_i^{(T)} > \Omega(T) \end{cases} .$$

Then to obtain $s_i^{(T-1)}$, it is necessary to obtain: $\omega_i^{(T-1)} = (\omega_i^{(T)} - \sigma_i^{(T)})/\alpha$. But in order to avoid dividing by the memory factor (recall that operations with real numbers are not exact in computer arithmetic), it is preferable to work with $\gamma_i^{(T-1)} = \omega_i^{(T)} - \sigma_i^{(T)}$, and to compare these values to $\Gamma(T-1) = \sum_{t=1}^{T-1} \alpha^{T-t}$. This leads to:

$$s_i^{(T-1)} = \begin{cases} 0 & \text{if } 2\gamma_i^{(T-1)} < \Gamma(T-1) \\ \sigma_i^{(T)} & \text{if } 2\gamma_i^{(T-1)} = \Gamma(T-1) \\ 1 & \text{if } 2\gamma_i^{(T-1)} > \Gamma(T-1) \end{cases} .$$

Continuing in the reversing process: $\gamma_i^{(T-2)} = \gamma_i^{(T-1)} - \alpha\sigma_i^{(T-1)}$ and $\Gamma(T-2) = \sum_{t=1}^{T-2} \alpha^{T-t}$. In general: $\gamma_i^{(T-\tau)} = \gamma_i^{(T-\tau+1)} - \alpha^{\tau-1}\sigma_i^{(T-\tau+1)}$ and $\Gamma(T-\tau) = \sum_{t=1}^{T-\tau} \alpha^{T-t}$, giving:

$$s_i^{(T-\tau)} = \begin{cases} 0 & \text{if } 2\gamma_i^{(T-\tau)} < \Gamma(T-\tau) \\ \sigma_i^{(T-\tau+1)} & \text{if } 2\gamma_i^{(T-\tau)} = \Gamma(T-\tau) \\ 1 & \text{if } 2\gamma_i^{(T-\tau)} > \Gamma(T-\tau) \end{cases} .$$

4. Reversible structurally dynamic cellular automata

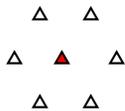
Structurally dynamic CA (SDCA) were suggested by Ilachinski and Halpern in [20]. The essential new feature of this model is that the connections between the cells are allowed to change according to rules similar in nature to the state transition rules associated with the conventional CA. This means that given certain conditions, specified by the link transition rules, links between rules may be created and destroyed. The neighborhood of each cell is now dynamic rather than fixed throughout the automaton, so state and link configurations of an SDCA are both dynamic and continually interacting.

In the Ilachinski and Halpern model, an SDCA consists of a finite set of binary-valued *cells* numbered 1 to N whose connectivity is specified by an $N \times N$ connectivity matrix in which $\lambda_{ij} = 1$ if cells i and j are connected; 0 otherwise. So, now: $\mathcal{N}_i^{(T)} = \{j/\lambda_{ij}^{(T)} = 1\}$ and $\sigma_i^{(T+1)} = \phi(\sigma_i^{(T)} \in \mathcal{N}_i^{(T)})$. The *distance* between two cells i and j , δ_{ij} , is defined as the number of links in the shortest path between i and j . We say that i and j are *direct* neighbors if $\delta_{ij} \leq 1$, and that i and j are *next-nearest* neighbors if $\delta_{ij} = 2$. There are two types of link transition functions in an SDCA: *couplers* and *decouplers*, the former add new links, the latter remove links. The coupler and decoupler set determines the link transition rule: $\lambda_{ij}^{(T+1)} = \psi(l_{ij}^{(T)}, \sigma_i^{(T)}, \sigma_j^{(T)})$.

Instead of introducing the formalism of the SDCA, we deal here with just one example in which the decoupler rule removes all links connected to cells in which both values are zero ($\lambda_{ij}^{(T)} = 1 \rightarrow \lambda_{ij}^{(T+1)} = 0$ iff $\sigma_i^{(T)} + \sigma_j^{(T)} = 0$) and the coupler rule adds links between all next-nearest neighbor sites in which both values are one ($\lambda_{ij}^{(T)} = 0 \rightarrow \lambda_{ij}^{(T+1)} = 1$ iff $\sigma_i^{(T)} + \sigma_j^{(T)} = 2$ and $\delta_{ij}^{(T)} = 2$). All of these totalistic rules are applied at the same time (in parallel).

Let us consider the case of Figure 5, in which, again, the initial triangular lattice¹ is seeded as in Figure 3. After the first iteration, at

¹With next-nearest neighborhood:



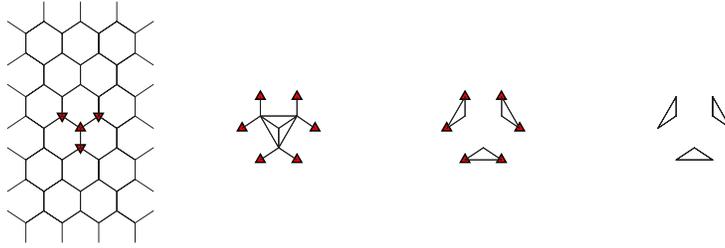


Figure 5. The ahistoric SDCA described in section 4 starting as in Figure 1.

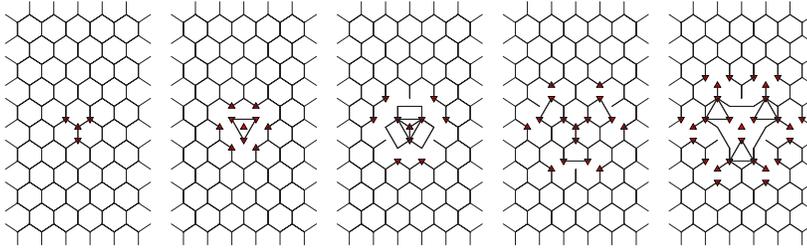


Figure 6. The reversible SDCA starting as in Figure 5, up to $T = 5$.

time step $T = 2$, most of the lattice structure has decayed as an effect of the decoupler rule, so that the active value cells (the next-nearest neighbors of the central cell) and links are confined into a small region, that dies off at $T = 5$.

The Fredkin's reversible construction is feasible in the SDCA scenario, extending the \ominus operation also to links:

$$\lambda_{ij}^{(T+1)} = \psi(\lambda_{ij}^{(T)}, \sigma_i^{(T)}, \sigma_j^{(T)}) \ominus \lambda_{ij}^{(T-1)}.$$

Figure 6 shows the evolution of the reversible formulation of the SDCA of Figure 5 up to $T = 5$. At variance with what happens in the irreversible formulation in Figure 5, the initial lattice structure does not decay at $T = 2$ (nor at posterior time steps) because of adding the structure at $T = 0$ (at $T - 1$), supposed to be the same as that at $T = 1$. Link transition rules do not alter auto-connections, but subtraction of patterns may do so. Thus, for example, in Figure 6 every cell is auto-connected at $T = 0$ and $T = 1$, but the subtraction of these patterns leads to the complete disappearance of auto-connections at $T = 2$. Auto-connections are not represented in the figures, but of course they affect the mass updating.

5. A reversible structurally dynamic cellular automaton with memory

Memory can be embedded in links in a manner similar to that in state values, so the link between any two cells is featured by a mapping of its previous values: $l_{ij}^{(T)} = \text{round}(m_{ij}^{(T)})$, $l_{ij}^{(T)} = \lambda_{ij}^{(T)}$ if $m_{ij}^{(T)} = 0.5$, after $m_{ij}^{(T)} = \omega_{ij}^{(T)}/\Omega(T)$ [2], with

$$\omega_{ij}^{(T)} = \lambda_{ij}^{(T)} + \sum_{t=1}^{T-1} \alpha^{T-t} \lambda_{ij}^{(t)} = \omega_{ij}^{(T-1)} + \alpha \lambda_{ij}^{(T)}.$$

The *distance* between two cells in the historic model (d_{ij}) is defined in terms of l instead of λ values, so that i and j are *direct* neighbors if $d_{ij} = 1$, and are *next-nearest* neighbors if $d_{ij} = 2$; $N_i^{(T)} = \{j/d_{ij}^{(T)} = 1\}$. Generalizing the approach to embedded memory introduced in section 2, the unchanged transition rules (ϕ and ψ) may operate on the featured link and mass values: $\sigma_i^{(T+1)} = \phi(s_j^{(T)} \in N_i)$, $\lambda_{ij}^{(T+1)} = \psi(l_{ij}^{(T)}, s_i^{(T)}, s_j^{(T)})$ [5]. An example is given in the SDCA with full memory in Figure 7, in which a period-two oscillator has an inactive component with no active cell. In Figure 7 the central cell is initially inactive in order to avoid extinction, allowing us to appreciate the effect of memory.

A generalization of the Fredkin’s reversible construction is feasible in the SDCA scenario endowed with memory as: $\sigma_i^{(T+1)} = \phi(s_j^{(T)} \in N_i^{(T)}) \ominus \sigma_i^{(T-1)}$, $\lambda_{ij}^{(T+1)} = \psi(l_{ij}^{(T)}, s_i^{(T)}, s_j^{(T)}) \ominus \lambda_{ij}^{(T-1)}$. Now, for reversing from T it is necessary to know not only $\sigma_i^{(T)}$, $\lambda_{ij}^{(T)}$, $\sigma_i^{(T+1)}$, and $\lambda_{ij}^{(T+1)}$, but also $\omega_i^{(T)}$ and $\omega_{ij}^{(T)}$, proceeding for reversing in connections as stated for reversing in mass values in section 3.

Figure 8 shows the effect of memory with $\alpha = 0.7$ in the initial scenario of Figure 6 from $T = 4$ to $T = 8$. If $\alpha \geq 0.61805$, the initial pattern is restored (as shown in Figure 8), which clearly differentiates the evolution from that of the ahistoric model. The evolving patterns in the reversible SDCA in the initial scenario of Figure 8 using full memory converge from $T = 7$ to a period-two oscillator, one of whose

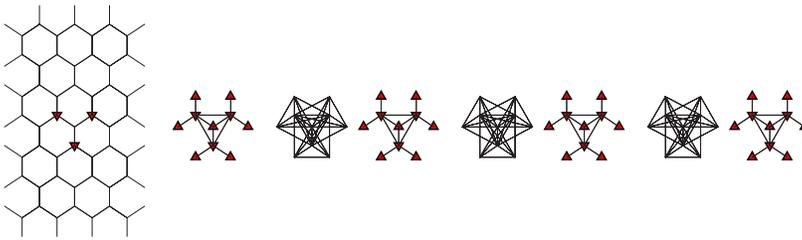


Figure 7. The SDCA with full memory starting as in Figure 5 but with the central cell inactive at $T = 1$.

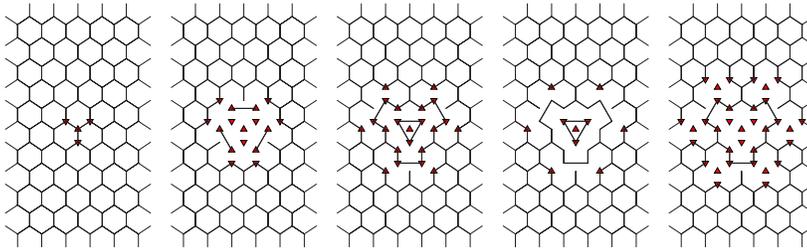


Figure 8. The reversible SDCA with $\alpha = 0.7$ memory starting as in Figure 6. Evolution from $T = 4$ up to $T = 8$.

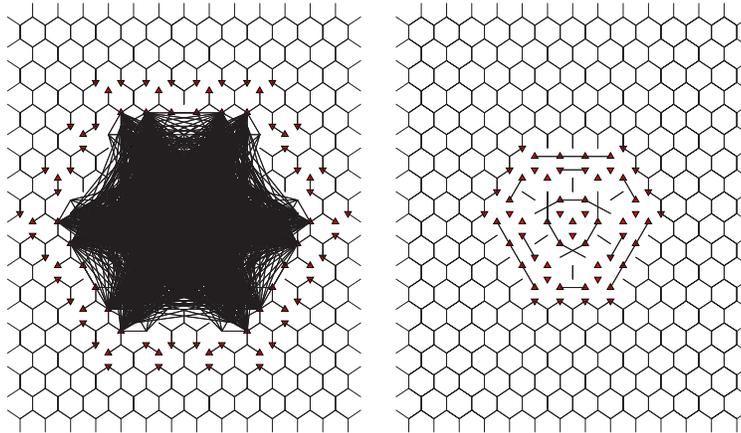


Figure 9. Patterns at $T = 13$ of the ahistoric and $\alpha = 0.7$ memory SDCAs.

components has no active mass cell. In the $\alpha = 0.9$ memory model a period-four oscillator is generated at $T = 11$.

Figure 9 shows the patterns at $T = 13$ for the ahistoric and $\alpha = 0.7$ reversible SDCA with the initial steps shown in Figures 6 and 8. In the ahistoric model, the web of connections is so dense in its central area that it is impossible to discern it. The web appears dramatically cleared in the historic model with memory factor $\alpha = 0.7$ (a likeable “face” seems to appear on it). The clearing of the web of connections together with a restraint in the advance of mass, mark the inertial effect of memory.

Figure 10 shows the evolution of mass density, the average number of nearest and next-nearest neighbors per site, and the effective dimension in the simulations of Figure 9. The *effective dimension* is the average ratio of the number of next-nearest to nearest neighbors per site,² which is a discrete analog to the continuous Hausdorff dimension. The

²In the triangular tessellation, the effective dimension is: $6/4 = 1.5$.

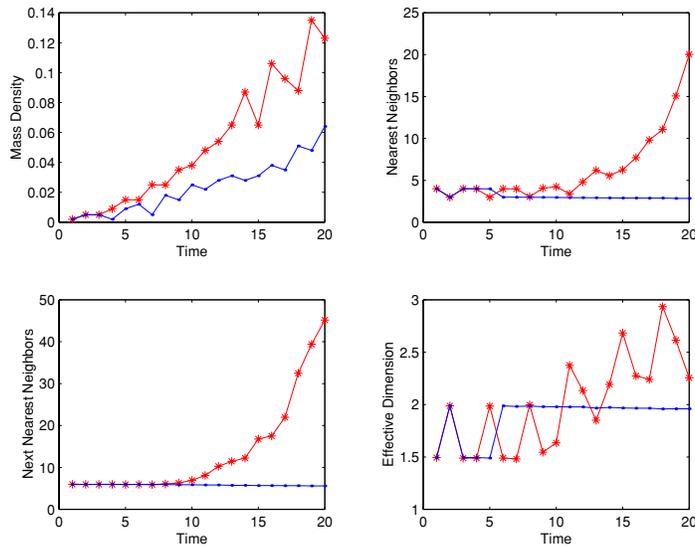


Figure 10. Evolution of mass density, average number of nearest and next-nearest neighbors per site, and effective dimension in the ahistoric (starred) and $\alpha = 0.7$ simulations of Figures 6 and 8 up to $T = 20$, implemented in a lattice of size 43×43 .

smoothing effect of memory is seen again in Figure 10: (i) the tendency of the mass density to grow is clearly restrained with memory, (ii) the evolution of the two-neighbor densities and of the effective dimension is soon fixed at around 3, 6, and 2 respectively.

As a glimpse of what happens when starting from arbitrary conditions, Figure 11 shows evolving patterns of the reversible cellular automaton considered here starting with random cell states. Periodic boundary conditions are imposed on the edges, but the wiring of border cells is omitted in Figure 11 to facilitate visualizing the web of connections. This would be masked if the links connecting opposite border cells were to appear. So,



After $T = 4$ the lattice has so many connections that it is difficult to discern which sites are connected, so only the wiring of the central cell (dramatically affected by memory) is shown. Figure 12 shows the evolution of densities and effective dimension in the scenario of Figure 11. The smoothing effect of memory is clearly appreciated in the nearest and

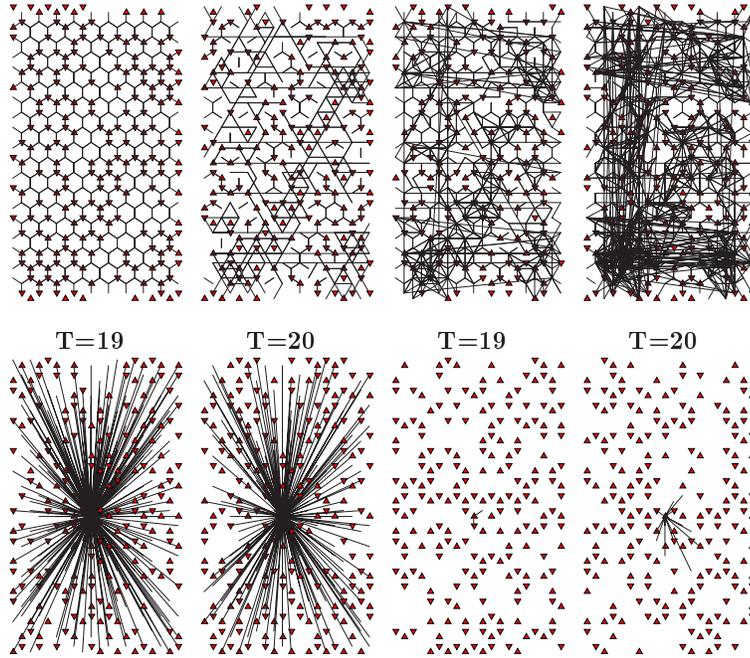


Figure 11. Patterns starting at random with mass in a triangular tessellation of size 20×20 . Reversible SDCA evolution without memory from $T = 1$ up to $T = 4$ and at $T = 19$ and $T = 20$. The last two patterns at $T = 19$ and at $T = 20$ correspond to the $\alpha = 0.7$ memory model. In the second row of patterns only the wiring of the central cell is shown.

next-nearest neighbor dynamics in Figure 12. The number of nearest neighbors oscillates around one and that of next-nearest neighbors in a (14,20) interval, much below the increasing values of these parameters in the ahistoric model. The erratic behavior of mass density seems to be very little affected by memory, whereas the evolution of the effective dimension is rather unexpected.

6. Other memories

Memory may be embedded either in cells but not in connections, or else only in connections. Figures 13 and 14 show the initial effect of such memory implementations on the example treated here.

Average-like memory models can readily be proposed by generalizing the memory charges as: $\omega_i^{(T)} = \sum_{t=1}^T \delta(t)\sigma_i^{(t)}$, $\omega_{ij}^{(T)} = \sum_{t=1}^T \delta(t)\lambda_{ij}^{(t)}$, with $\Omega(T) = \sum_{t=1}^T \delta(t)$. The geometric discount model considered until now ($\delta(t) = \alpha^{T-t}$) is just one of the many possible weighting functions.

Alternatively, previous states can be pondered with the weight: $\delta(t) = t^c$ [14]. Choosing integer c parameter values allows working only with

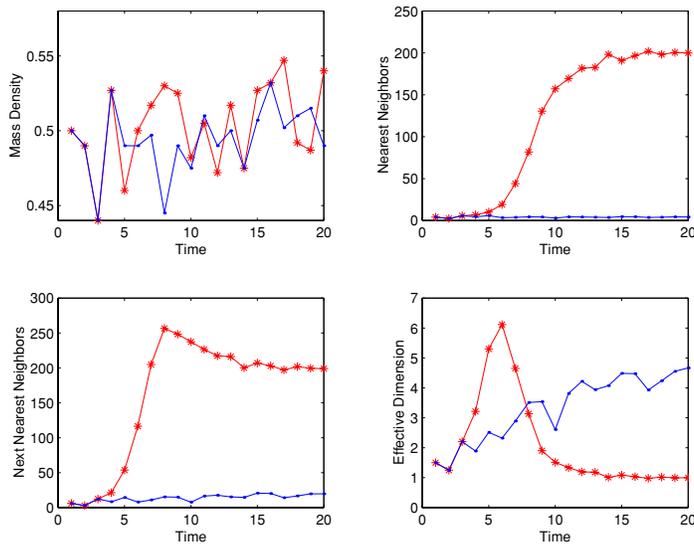


Figure 12. Evolution of mass density, average number of nearest and next-nearest neighbors per site, and effective dimension in the ahistoric (starred) and $\alpha = 0.7$ simulations of Figure 11.

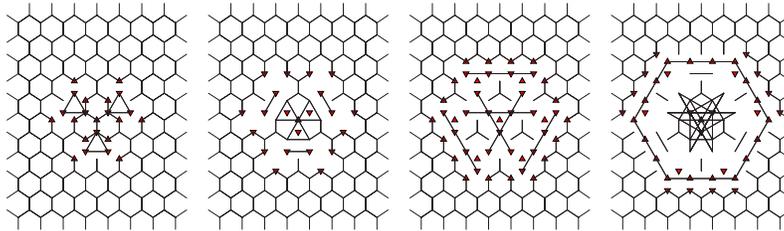


Figure 13. The evolving patterns of the SDCA described in text with $\alpha = 0.7$ memory only in connections. Evolution from $T = 4$ to $T = 7$ starting as in Figure 6.

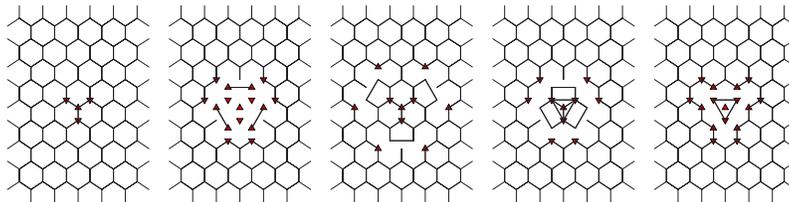


Figure 14. The evolving patterns of the SDCA described in text with $\alpha = 0.7$ memory only in mass values. Evolution from $T = 4$ to $T = 8$ starting as in Figure 6.

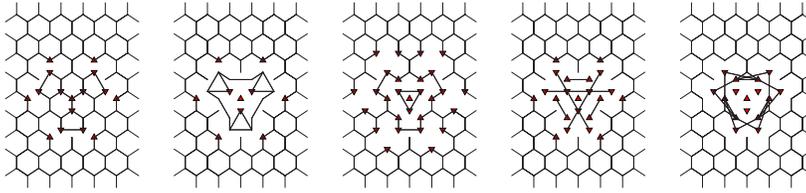


Figure 15. The evolving patterns of the SDCA described in text, with integer-based memory $\delta(t) = t$. Evolution from $T = 4$ to $T = 8$ starting as in Figure 6.

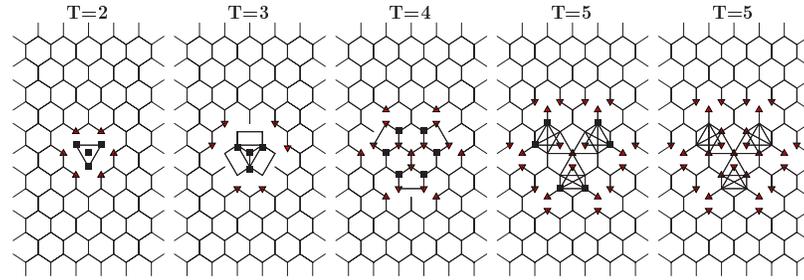


Figure 16. The evolving patterns of the three-states reversible SDCA described in the text. *Square* cells are at state 2. Ahistoric evolution from $T = 2$ up to $T = 5$ starting as in Figure 6. Pattern at $T = 5$ with $\delta(t) = 2^t$ memory in cells.

integers by comparing the $2\omega_i$ and $2\omega_{ij}$ figures across the lattice to the factor $\Omega(T)$. For $c = 0$, we have the full historic model; for $c = 1$ it is $\delta(t) = t$ with $\Omega(T) = T(T + 1)/2$. The larger the value of c , the more heavily the recent past is taken into account, and consequently the closer to the scenario of the ahistoric one. Figure 15 shows an example of the effect of $\delta(t) = t$ integer-based memory implementation.

Another weight with the same integer-based property is c^t (again provided that c is an integer). This memory weight is not operative with cells and links with two states, but it becomes operative when allowing three states (0, 1, 2), in which case the $m_i^{(T)}$ and $m_{ij}^{(T)}$ values are to be compared to the hallmarks 0.5 and 1.5, assigning the last state/link value in the case of an equality to any hallmark. In order to work with integers and save computing demands, instead of comparing the m values to the hallmarks 1/2 and 3/2, it is preferable to compare the 2ω values to the hallmarks 1 and 3 [11]. Figure 16 shows the effect of $\delta(t) = 2^t$ memory starting as in Figure 6 but allowing cells to have three states (links remain two-valued). When working with three states, the reversible mechanism is implemented with subtraction modulo 3, thus the initially $\sigma = 1$ cells reach state 2 (square cells) at $T = 2$ in Figure 16 because $0 \ominus_3 1 = 2$. The pattern with memory at $T = 5$ in Figure 16 coincides with that of the ahistoric model except as regards

the state-two cells in the ahistoric model, which are not active in the historic pattern.

Reversing is easier in the integer-based memory scenarios than in that of geometric discount as $\omega_i^{(T)} = \omega_i^{(T-1)} + \delta(T)\sigma_i^{(T)}$ and $\omega_{ij}^{(T)} = \omega_{ij}^{(T-1)} + \delta(T)\lambda_{ij}^{(T)}$ readily reverses, without the computational inconvenience of division by α . Working only with integers (*à la* CA) is a clear computational advantage. Nevertheless, the integer-based weights c^t and t^c share the same drawback: they “explode,” at high values of t , even for $c = 2$.

7. Discussion

The effect of memory embedded in cells and links on a particular reversible structurally dynamic cellular automaton (SDCA) (based on the parity rule) in the triangular tessellation is studied qualitatively (pictorially) in this work. As a rule, geometrically discounted memory has been shown to produce an inertial effect that tends to preserve the main features of initial conditions. This notably alters the ahistoric dynamics, even if a low level of memory is implemented.

A complete analysis of the effect of memory on reversible SDCA is left for future work which will develop a phenomenology of reversible SDCA with memory, that is, the full analysis of the rule space based on the morphological classification of patterns formed, the intrinsic parameters (e.g., Langton’s lambda, Wuensche’s Z parameter), the structure of global transition graphs (this would be feasible only in the one-dimensional case), the entropy, and other dynamics-related issues. Potential fractal features are also to come under scrutiny [23].

Some critics may argue that memory is not in the realm of CA, (or even of dynamic systems), but we believe that the subject is worth studying. At least CA with memory can be considered as a promising extension of the basic paradigm. A major impediment to modeling with CA stems from the difficulty of utilizing their complex behavior to exhibit a particular behavior or perform a particular function: embedding memory in states and links broadens the spectrum of CA as a tool for modeling. It is likely that in some contexts, a transition rule with memory could match the “correct” behavior of the CA system of a given complex system.

The SDCA seems to be particularly appropriate for modeling the human brain function (links/synapses connect cells/neurons) in which the relevant role of memory is apparent. Reversibility in this context would avoid the possibility of “reinventing history.” Models similar to SDCA have been adopted to build a dynamical network approach to quantum space-time physics [22]. Reversibility is an important issue at such a fundamental physics level.

Apart from their potential applications, SDCA with memory have an aesthetic and mathematical interest on their own. The study of the effect of memory on CA has been rather neglected and there have been only limited investigations of SDCA since its introduction in the late 1980s.³ Nevertheless, it seems plausible that further study on SDCA (and lattice gas automata with dynamical geometry [25]) with memory⁴ should turn out to be profitable.

Acknowledgments

This work was supported by MEC Grant PR2006-0081. Attendance to the NKS 2006 Conference was supported by the Universidad Politécnica de Madrid (Ayudas para la presentación de ponencias).

References

- [1] S. Wolfram, *A New Kind of Science* (Wolfram Media, Inc., Champaign, IL, 2002).
- [2] P. Julian and L. O. Chua, “Replication Properties of Parity Cellular Automata,” *International Journal of Bifurcation and Chaos*, **12**(3) (2002) 477–494.
- [3] C. Bays, “Cellular Automata in the Triangular Tessellation,” *Complex Systems*, **8** (1994) 127–150.
- [4] R. W. Gerling, “Classification of Triangular and Honeycomb Cellular Automata,” *Physica A*, **162** (1990) 196–209.
- [5] R. Alonso-Sanz, “A Structurally Dynamic Cellular Automaton with Memory,” *Chaos, Solitons and Fractals*, **32** (2006) 1285–1295.
- [6] A. Ilachinski, *Cellular Automata: A Discrete Universe* (World Scientific, Singapore, 2000).
- [7] R. Alonso-Sanz, (2006). “Elementary Rules with Elementary Memory Rules: The Case of Linear Rules,” *Journal of Cellular Automata*, **1** (2006) 70–86.
- [8] R. Alonso-Sanz and M. Martin, “One-dimensional Cellular Automata with Memory in Cells of the Most Frequent Recent Value,” *Complex Systems*, **15**(3) (2005) 203–236.
- [9] R. Alonso-Sanz, “Phase Transitions in an Elementary Probabilistic Cellular Automaton with Memory,” *Physica A*, **347** (2005) 383–401.

³To the best of our knowledge, the relevant references on SDCA are [17–21], together with a review chapter in [6] and a section in [24].

⁴Not only in the basic paradigm scenario, but also in SDCA with random but value-dependent rule transitions (which relates SDCA to genetic networks), and/or in SDCA with the extensions considered in [21], such as unidirectional links.

- [10] R. Alonso-Sanz, “One-dimensional, $r = 2$ Cellular Automata with Memory,” *International Journal of Bifurcation and Chaos*, **14** (2004) 3217–3248.
- [11] R. Alonso-Sanz and M. Martin, “Three-state One-dimensional Cellular Automata with Memory,” *Chaos, Solitons, and Fractals*, **21** (2004) 809–834.
- [12] R. Alonso-Sanz, “Reversible Cellular Automata with Memory,” *Physica D*, **175** (2003) 1–30.
- [13] R. Alonso-Sanz and M. Martin, “Elementary Cellular Automata with Memory,” *Complex Systems*, **14** (2003) 99–126.
- [14] R. Alonso-Sanz and M. Martin, “Cellular Automata with Accumulative Memory: Legal Rules Starting from a Single Site Seed,” *International Journal of Modern Physics C*, **14** (2003) 695–719.
- [15] R. Alonso-Sanz and M. Martin, “One-dimensional Cellular Automata with Memory: Patterns Starting with a Single Site Seed,” *International Journal of Bifurcation and Chaos*, **12** (2002) 205–226.
- [16] R. Alonso-Sanz and M. Martin, “Two-dimensional Cellular Automata with Memory: Patterns Starting with a Single Site Seed,” *International Journal of Modern Physics C*, **13** (2002) 49–65, and the references therein.
- [17] P. Halpern and G. Caltagirone, “Behavior of Topological Cellular Automata,” *Complex Systems*, **4** (1990) 623–651.
- [18] P. Halpern, “Sticks and Stones: A Guide to Structurally Dynamic Cellular Automata,” *American Journal of Physics*, **57** (1989) 405–408.
- [19] D. Hillman, “Combinatorial Spacetimes,” Ph.D. Thesis (mathematics) University of Pittsburgh, 1995; hep-th/9805066.
- [20] A. Ilachinsky and P. Halpern, “Structurally Dynamic Cellular Automata,” *Complex Systems*, **1** (1987) 503–527.
- [21] S. M. Majercik, “Structurally Dynamic Cellular Automata,” University of Southern Maine (MsC), 1994.
- [22] M. Requardt, (1998). “Cellular Networks as Models for Plank-scale Physics,” *Journal of Physics A: Mathematical and Theoretical*, **31** (1998) 7797–8021; arxiv.org/abs/gr-qc/0308089.
- [23] J. R. Sanchez and R. Alonso-Sanz, “Multifractal Properties of R90 Cellular Automaton with Memory,” *International Journal of Modern Physics C*, **15** (2004) 1461–1470.
- [24] A. Adamatzky, *Identification of Cellular Automata* (Taylor and Francis, Bristol, PA, 1994).
- [25] P. J. Love, B. M. Boghosian, and D. A. Meyer, “Lattice Gas Simulations of Dynamical Geometry in One Dimension,” *Philosophical Transactions of the Royal Society A*, **362** (2004) 1667–1675.