On Soliton Collisions between Localizations in Complex Elementary Cellular Automata: Rules 54 and 110 and Beyond

Genaro J. Martínez

Departamento de Ciencias e Ingeniería de la Computación
Escuela Superior de Cómputo, Instituto Politécnico Nacional, México
and
Unconventional Computing Center, Bristol Institute of Technology
University of the West of England, Bristol BS16 1QY, United Kingdom
genaro.martinez@uwe.ac.uk

Andrew Adamatzky

Unconventional Computing Center, Bristol Institute of Technology
University of the West of England, Bristol BS16 1QY, United Kingdom
andrew.adamatzky@uwe.ac.uk

Fangyue Chen

School of Sciences, Hangzhou Dianzi University
Hangzhou, Zhejiang 310018, P. R. China
fychen@hdu.edu.cn

Leon Chua

Electrical Engineering and Computer Sciences Department
University of California at Berkeley, California, United States of America
chua@eecs.berkeley.edu

In this paper, a single-soliton two-component cellular automaton (CA) model of waves is presented as mobile self-localizations, also known as particles, waves, or gliders, in addition to its version with memory. The model is based on coding sets of strings where each chain represents a unique mobile self-localization. The original soliton models in CAs proposed with filter automata are briefly discussed, followed by solutions in elementary CAs (ECAs) domain with the famous universal ECA rule 110, and reporting a number of new solitonic collisions in ECA rule 54. A mobile self-localization in this study is equivalent to a single soliton because the collisions of the mobile self-localizations studied in this paper satisfy the property of solitonic collisions. A specific ECA with memory (ECAM), the ECAM rule $\phi_{\text{R9maj:4}}$, is also presented; it displays single-soliton solutions from any initial codification (including random initial conditions) for a kind of mobile self-localization because such an automaton is able to adjust any initial condition to soliton structures.

Complex Systems, **21** © 2012 Complex Systems Publications, Inc.
https://doi.org/10.25088/ComplexSystems.21.2.117

1. Introduction

A *soliton* can be defined informally as follows: when two solitary waves travel in opposite directions and collide, they emerge after collision with the same shape and velocity asymptotically. The phenomenon of the solitary wave was first recognized by English engineer John Scott Russell [1] and first formalized by Diederick J. Korteweg and Gustav de Vries in 1895 [2]. (Read Russell's original papers at http://www.ma.hw.ac.uk/~chris/scott_russell.html.) However, in 1965 the physician Martin Kruskal coined the term "soliton" to describe the phenomenon of the solitary wave.

Solitons in one-dimensional (1D) cellular automata (CAs) have their own interest and history; they have been extensively studied since 1986 by Kennet Steiglitz and colleagues [3–6]. These studies have been based on a variant of classic CAs, known as parity rule filter automata (PRFA). PRFA mainly use newly computed site values as soon as they are available; they are also analogous to infinite impulse response (IIR) digital filters, while conventional CAs correspond to finite impulse response (FIR) [3]. Incidentally, even more sophisticated solitons with PRFA were obtained by Siwak in [7], showing large and multiple simultaneous solitonic sequential collisions (i.e., not parallel mapping). Turbulence solitons in 1D CAs were explored by Aizawa, Nishikawa, and Kaneko in [8].

Studying 1D soliton CAs is important because it allows for fast prototyping of soliton logic. For practical implementations of soliton logic, see the overview developed by Blair and Wagner in [9], which led to novel designs of optical parallel computers. An interesting implementation showing the wave propagation equation in lattice gas simulated with a partitioned CA was developed by Margolus, Toffoli, and Vichniac in [10, 11]. Solitons have found numerous relevant applications, including in fiber optics, breather waves, the nonlinear Schrödinger equation, magnets, and recently in proteins and DNA, known as bio-solitons [12–16].

Historically, complex CAs have been related to the presence of mobile self-localizations (also called gliders, particles, or waves). The most famous CA is the two-dimensional (2D) CA in Conway's Game of Life [17], but we can also find a number of samples in 1D supporting mobile self-localizations, as in [18–23]. Some of them explicitly process signals (not mobile self-localizations) as found by Delorme and Mozayer in [24] or solve the firing squad synchronization problem as found by Umeo in [25]. Indeed, a number of CAs have been exploited as physical models in [26–30].

This paper is organized as follows. Section 2 gives a general introduction on CAs and basic notation. Section 3 presents experimental soliton solutions in CAs including solitons in complex elementary CA (ECA) rules 54 and 110, and we report in this paper the soliton reactions emerging in rule 54 from multiple collisions. In Section 3.3, we display a new ECA with memory (ECAM) able to solve experimen-

https://doi.org/10.25088/ComplexSystems.21.2.117

tally the most simple single-soliton two-component solution from any initial configuration. In Section 3.4, we discuss some computing capacities based on solitons. In Section 4, we will discuss final remarks.

2. One-Dimensional Cellular Automata

2.1 Elementary Cellular Automata

A CA is a quadruple $\langle \Sigma, \varphi, \mu, c_0 \rangle$ evolving on a specific d-dimensional lattice, where each cell x_i, $i \in N$ takes a state from a finite alphabet Σ such as $x \in \Sigma$. A sequence $s \in \Sigma^n$ of n cell-states represents a string or a global configuration c on Σ. We write a set of finite configurations as Σ^n. Cells update their states via an evolution rule $\varphi : \Sigma^\mu \to \Sigma$, such that μ represents a cell neighborhood that consists of a central cell and a number of neighbors connected locally. There are $|\Sigma|^\mu$ different neighborhoods and if $k = |\Sigma|$, then we have k^{k^n} different 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 $\Phi\left(c^t\right) \to c^{t+1}$. Thus c_0 is the initial configuration. Cell states of a configuration c^t are updated simultaneously by the evolution rule as

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

where i indicates cell position and r is the radius of neighborhood μ. Thus, the ECA class represents a system of order ($k = 2$, $r = 1$) in Wolfram's notation [28]. To represent a specific evolution rule we will write the evolution rule in a decimal notation, for example, φ_{R110}. Thus Figure 1 illustrates how evolution dynamics work in one dimension for an ECA.

2.2 Elementary Cellular Automata with Memory

Conventional CAs are memoryless: the new state of a cell depends on the neighborhood configuration solely at the preceding time step of φ. CAs with memory are an extension of CAs in such a way that every cell x_i is allowed to remember its states during some fixed period of its evolution. CAs with memory were originally proposed by Alonso-Sanz in [31–34].

Hence we implement a memory function ϕ, as follows:

$$\phi\left(x_i^{t-\tau}, \ldots, x_i^{t-1}, x_i^t\right) \to s_i, \tag{2}$$

https://doi.org/10.25088/ComplexSystems.21.2.117

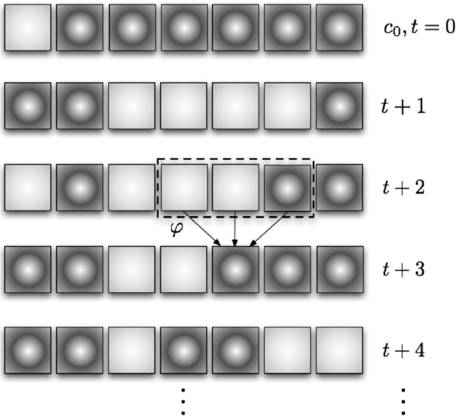


Figure 1. Dynamic in ECA on an arbitrary 1D array and hypothetical evolution rule φ.

where $\tau < t$ determines the *degree of memory* and each cell $s_i \in \Sigma$ is a state function of the series of states of the cell x_i with memory backward up to a specific value τ. To execute the evolution, we apply the original rule on the cells s as

$$\varphi(\ldots, s_{i-1}, s_i, s_{i+1}, \ldots) \to x_i^{t+1} \tag{3}$$

to get an evolution with memory. Thus in CAs 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 canalize memory to the map φ [33].

Let us consider the *memory function* ϕ as a *majority memory*,

$$\phi_{\text{maj}} \to s_i,$$

where in case of a tie given by $\Sigma_1 = \Sigma_0$ from ϕ we take the last value x_i. Thus, ϕ_{maj} function represents the classic majority function (for three values [35]). Then we have

$$\phi_{\text{maj}}(a, b, c) : (a \wedge b) \vee (b \wedge c) \vee (c \wedge a), \tag{4}$$

which represents the cells $\left(x_i^{t-\tau}, \ldots, x_i^{t-1}, x_i^t\right)$ and defines a temporal ring s before getting the next global configuration c. Of course, this evaluation can be for any number of values of τ. In this way, a number of functional memories may be used and not only the majority, including the minority, parity, alpha, and more [33, 34].

The evolution rules representation for ECAM is given in [36–38], as follows:

$$\phi_{CAR\,m:\tau} \tag{5}$$

https://doi.org/10.25088/ComplexSystems.21.2.117

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 $\phi_{\text{R30maj:5}}$. The memory is functional, as is the CA itself; see a schematic explanation in Figure 2. However, computationally a memory function has a quadratic complexity calculating its evolution space.

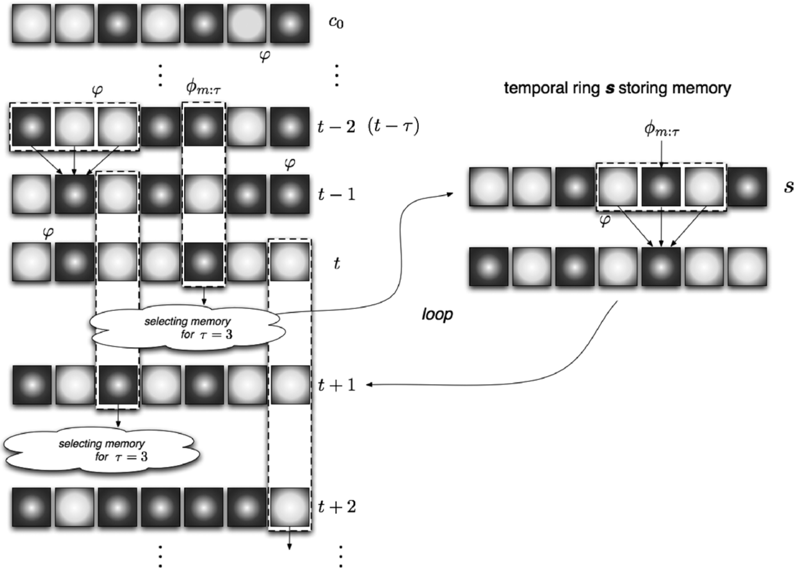


Figure 2. Dynamics in ECAM on an arbitrary 1D array and hypothetical evolution rule φ and memory function ϕ_m with $\tau = 3$.

3. Solitons in One-Dimensional Cellular Automata

A soliton is a solitary wave with nonlinear behavior that preserves its form and speed, interacting with some kind of perturbation. The latter can be another wave or some obstacle, continuing its travel affecting only its phase and position since each collision. One example is a water wave traveling and interacting with other waves; they can be found also in optics, sound, and molecules [12].

The solitary wave described by Scott become formally represented by the Korteweg–de Vries equation [2] as

$$u_t + u_{xxxx} + u\,u_x = 0, \tag{6}$$

where the function u measures high-wave and x-position at time t, and every subindex represents partial differences. The second term represents scattering-wave and the last term is the nonlinear term [5].

https://doi.org/10.25088/ComplexSystems.21.2.117

However, we will indicate that soliton models related to CAs do not find some direct relations matching some differential equation solutions. Nevertheless, Steiglitz has displayed some properties with Manakov systems and PRFA in [6, 39] in the search for a computable system collision-based soliton [4]. In addition, Adamatzky in [40] has designed a way to manipulate solitons to implement logic gates. On the other hand, Chua has explicitly developed an extended analysis on how ECAs can be described precisely as differential equations and cellular complex networks (CCN) in [41].

Although many studies were done on ECAs, we cannot find much about the soliton phenomena for each rule. Complex ECAs are direct candidates to explore such reactions from the interaction of their mobile self-localizations. Some explorations were described in [8, 19, 20, 22, 27, 29, 42, 43]. Solitons in CAs are characterized as a set of self-organized cells emerging on the evolution space; such complex patterns have a form, volume, velocity, phase, period, mass, and shift. Of course, not all these mobile self-localizations may work as solitons because they depend on their interaction with other structures. Consequently, a classification is necessary from the evolution space because they cannot be inferred from the local rule.

While a PRFA was designed to yield solitonic collisions calculating the new values as soon as they are available, their mobile self-localizations present a strong orientation to the left. This is a natural consequence of their function to calculate the next cell, which evaluates the $(i - r)^{t+1}$ cells [3]. The main and most important difference from conventional ECAs is that those mobile self-localizations working as solitons shall be searched explicitly and cannot be deduced to evolve the system. Thus, not all complex ECAs are able to produce collisions as solitons, although they could evolve some kind of mobile self-localizations. Steiglitz has amply researched the PRFA with the goal of reaching unconventional computing devices based on soliton collision [4–6, 44].

In Sections 3.1 through 3.3, we will discuss particular cases with complex ECAs and ECAM, displaying exact codifications to get soliton collisions between mobile self-localizations. We will also present some computable capacities. In the ECA domain, we have selected and researched only complex rules 54 and 110, because no other ECA rules present a universe with such diversity of mobile self-localizations and consequently an ample diversity of collisions. In the ECAM domain, we will present a single case that experimentally solves the most simple single-soliton two-component solution from any initial configuration, the evolution rule $\phi_{\mathrm{R9maj:4}}$.

3.1 Solitons in Elementary Cellular Automaton Rule 110

ECA rule 110 is a complex CA evolving with a complicated system of mobile self-localizations. Its local function is defined as follows:

https://doi.org/10.25088/ComplexSystems.21.2.117

$$\varphi_{\text{R110}} = \begin{cases} 1 & \text{if} \quad 001, 010, 011, 101, 011 \\ 0 & \text{if} \quad 000, 100, 111. \end{cases} \tag{7}$$

Figure 3 illustrates the complex dynamics from a typical random initial condition selecting the evolution rule φ_{R110}. A number of mobile self-localizations emerge on its evolution space and how a number of them collide.

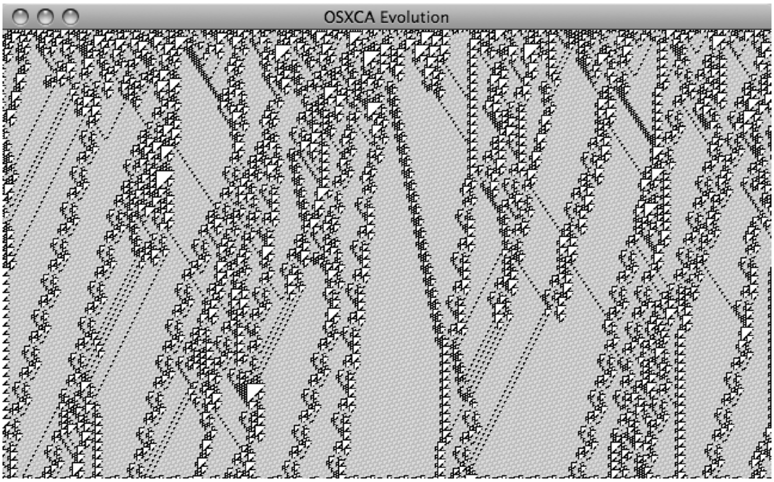


Figure 3. Random evolution in rule 110 on a ring of 644 cells to 375 generations. White cells represent state 0 and black cells state 1 starting on 50% of density. A filter is selected to get a better view of mobile self-localizations on its periodic background.

See detailed studies of rule 110 and mobile self-localizations in the glider system [21, 45, 46], universality [38, 47–52], and collisions and rule 110 objects [53, 54]. So generalities can be explored from the rule 110 repository. (For information on gliders in rule 110, visit http://uncomp.uwe.ac.uk/genaro/rule110/glidersRule110.html. For information on the rule 110 repository, visit http://uncomp.uwe.ac.uk/genaro/Rule110.html.)

We will focus on mobile self-localizations that present solitonic reactions. Localizations that have such a property are classified in Figure 4, following Cook's notation [47]. Here we can observe stationary, shift-right, and shift-left (displacements) localizations.

Rule 110 has an unlimited number of collisions as a consequence of some extendible mobile self-localizations [21, 53]. In this way, we have first constructed a set of configurations c coding each localization and yielding the solitonic reaction desired.

https://doi.org/10.25088/ComplexSystems.21.2.117

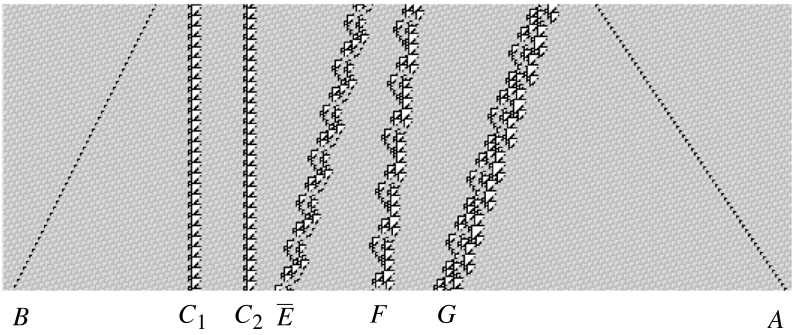


Figure 4. Set of mobile self-localizations with solitonic properties in rule 110.

To drive collisions and localizations, we will use the set of regular expressions f_1_1 localization-based to code initial configurations in rule 110; for full details, see [45]. (For information on glider-based regular language in rule 110, visit http://uncomp.uwe.ac.uk/genaro/rule110/listPhasesR110.txt.)

Table 1 shows a number of properties for each mobile self-localization (Figure 4), such as shift, period, speed, and volume (which can be related to its mass as well). All of them shall help us to synchronize collisions given a specific phase, where each mobile self-localization may present different *contact points* and collide with other mobile self-localizations. To produce a specific collision between mobile self-localizations at a given point, we must have full control over initial conditions, including distance between gliders and their phases at the moment of collision.

Mobile Self-Localization	Shift	Period	Speed	Volume
A	2	3	$2/3 \approx 0.666666$	6
B	2	4	$-1/2 = -0.5$	8
C_1	0	7	$0/7 = 0$	9–23
C_2	0	7	$0/7 = 0$	17
$\overline{E}$	8	30	$-4/15 \approx -0.266666$	21
F	4	36	$-1/9 \approx -0.111111$	15–29
G^n	14	42	$-1/3 \approx -0.333333$	24–38

Table 1. Mobile self-localizations properties such as solitons.

The notation proposed to codify initial conditions in rule 110 by phases is as follows:

$$\sharp_1\,(\sharp_2,\, f_i\_1), \tag{8}$$

https://doi.org/10.25088/ComplexSystems.21.2.117

where $\#_1$ represents a particular mobile self-localization (given in Table 1) and $\#_2$ represents its phase if it has a period greater than four (for full details, see [45]). Variable f_i indicates the phase currently used and the second subscript j (forming notation f_{i}_j) indicates the selected master set of regular expressions.

In [53], we have calculated experimentally the whole set of binary collisions between mobile self-localizations in rule 110, colliding all 1-1 mobile self-localizations. Thus in [55, 56], we have reported all soliton reactions in rule 110.

This way, 18 solitons (between two mobile self-localizations, i.e., binary) in rule 110 can be coded in phases, as follows.

(a) Soliton 1: $A(f_1\_1)$-6 e-$G(C,f_1\_1) \longrightarrow \{G, A\}$

(b) Soliton 2: $C_1(A,f_1\_1)$-3 e-$\overline{E}(B,f_1\_1) \longrightarrow \{\overline{E}, C_1\}$

(c) Soliton 3: $C_1(A,f_1\_1)$-3 e-$\overline{E}(C,f_1\_1) \longrightarrow \{\overline{E}, C_1\}$

(d) Soliton 4: $F(A,f_1\_1)$-3 e-$B(f_4\_1) \longrightarrow \{B, F\}$

(e) Soliton 5: $C_2(A,f_1\_1)$-3 e-$\overline{E}(C,f_1\_1) \longrightarrow \{\overline{E}, C_2\}$

(f) Soliton 6: $C_1(A,f_1\_1)$-2 e-$F(B,f_1\_1) \longrightarrow \{F, C_1\}$

(g) Soliton 7: $C_2(A,f_1\_1)$-2 e-$F(A,f_1\_1) \longrightarrow \{F, C_2\}$

(h) Soliton 8: $A(f_1\_1)$-4 e-$\overline{E}(A,f_1\_1) \longrightarrow \{\overline{E}, A\}$

(i) Soliton 9: $A(f_1\_1)$-4 e-$\overline{E}(B,f_1\_1) \longrightarrow \{\overline{E}, A\}$

(j) Soliton 10: $A(f_1\_1)$-4 e-$\overline{E}(C,f_1\_1) \longrightarrow \{\overline{E}, A\}$

(k) Soliton 11: $A(f_1\_1)$-4 e-$\overline{E}(H,f_1\_1) \longrightarrow \{\overline{E}, A\}$

(l) Soliton 12: $F(A,f_1\_1)$-e-$\overline{E}(A,f_1\_1) \longrightarrow \{\overline{E}, F\}$

(m) Soliton 13: $F(A,f_1\_1)$-e-$\overline{E}(C,f_1\_1) \longrightarrow \{\overline{E}, F\}$

(n) Soliton 14: $F(A,f_1\_1)$-e-$\overline{E}(D,f_1\_1) \longrightarrow \{\overline{E}, F\}$

(o) Soliton 15: $F(A,f_1\_1)$-e-$\overline{E}(E,f_1\_1) \longrightarrow \{\overline{E}, F\}$

(p) Soliton 16: $F(G,f_1\_1)$-e-$\overline{E}(A,f_1\_1) \longrightarrow \{\overline{E}, F\}$

(q) Soliton 17: $F(G,f_1\_1)$-e-$\overline{E}(B,f_1\_1) \longrightarrow \{\overline{E}, F\}$

(r) Soliton 18: $F(G,f_1\_1)$-e-$\overline{E}(H,f_1\_1) \longrightarrow \{\overline{E}, F\}$

Of course, from these solitonic binary collisions we can codify and synchronize most structures and therefore get multiple solitonic reac-

https://doi.org/10.25088/ComplexSystems.21.2.117

tions increasing its complexity. For example, we have the next codification:

(s) Multiple soliton:
$C_1(\mathrm{B},\mathrm{f}_1\_1)$-$e$-$C_1(\mathrm{A},\mathrm{f}_1\_1)$-$2\,e$-$C_2(\mathrm{A},\mathrm{f}_1\_1)$-$e$-$F(\mathrm{A},\mathrm{f}_1\_1)$-$e$-$\overline{E}(\mathrm{A},\mathrm{f}_1\_1)$-$3\,e$-$\overline{E}(\mathrm{C},\mathrm{f}_2\_1) \longrightarrow \{\overline{E}, \overline{E}, F, C_1, C_1, C_2\}$.

All these solitons in rule 110 are displayed in Figure 5. Each codification of (a) to (r) presents the binary case, and (s) presents a multiple solitonic collision with six localizations, where each is synchronized to produce the soliton reaction.

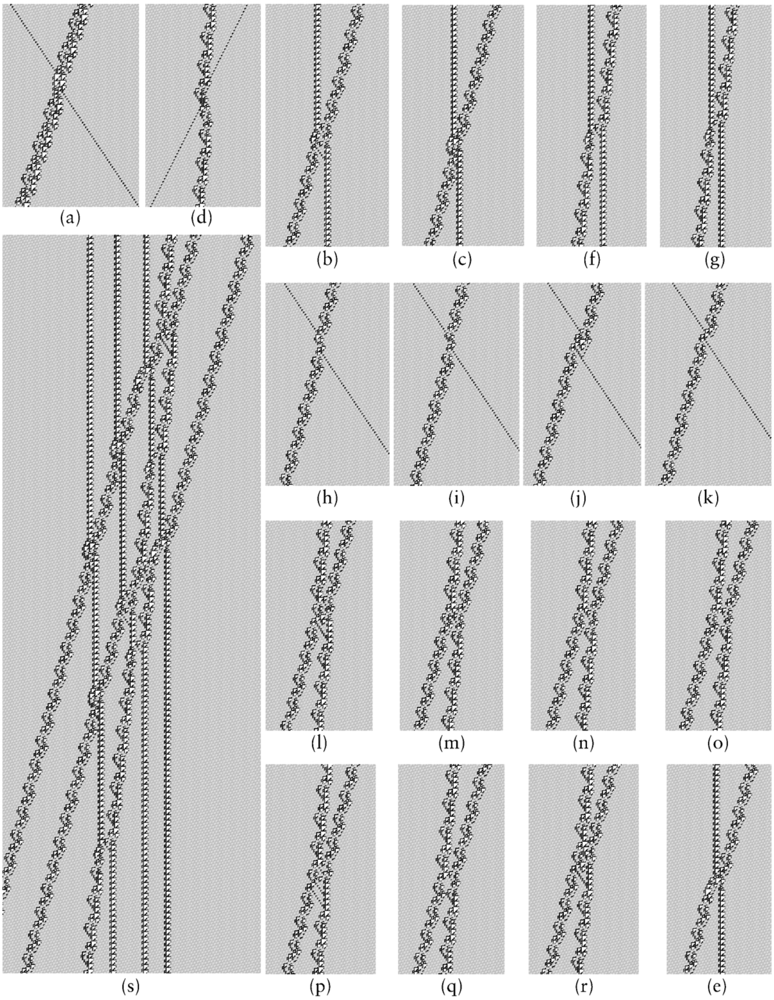


Figure 5. (a–r) Binary solitons in rule 110, and one case (s) illustrating multiple solitonic collision with six mobile self-localizations, synchronized and evolving in 964 generations.

https://doi.org/10.25088/ComplexSystems.21.2.117

Yet as a special case in rule 110, we can find a collision called a *pseudo-soliton* [55] that works recovering the original localization after two collisions. This is performed with B, $\overline{B}$, and F localizations. Localizations B and $\overline{B}$ have the same period and speed, but their volumes are different.

In [53], we calculated the whole set of binary collisions between mobile self-localizations and summarized them in Table 2. We have placed particular attention on asterisk labels because they represent precisely the pseudo-soliton in rule 110. Hence we know that the reaction $F \to \overline{B} = \{B, F\}$ and also that $F \to B = \{\overline{B}, F\}$, thus a loop may be constructed to synchronize such collisions. Figure 6 displays such a construction, given its codification in phases as

$$F(\mathrm{G},\mathrm{f}_3\_1)\text{-}2\,e\text{-}F(\mathrm{A},\mathrm{f}_1\_1)\text{-}e\text{-}B(\mathrm{f}_1\_1)\text{-}5\,e\text{-}\overline{B}(\mathrm{B},\mathrm{f}_4\_1).$$

Collisions $F \to \overline{B}$	Collisions $F \to B$
$F(\mathrm{A},\mathrm{f}_1\_1)\text{-}e\text{-}\overline{B}(\mathrm{A},\mathrm{f}_1\_1) = \{A, B, \overline{B}, F\}$	$F(\mathrm{A},\mathrm{f}_1\_1)\text{-}e\text{-}B(\mathrm{f}_1\_1) = \{\overline{B}, F\}$ *
$F(\mathrm{A},\mathrm{f}_1\_1)\text{-}e\text{-}\overline{B}(\mathrm{B},\mathrm{f}_1\_1) = \{A, 2\,C_3, C_1\}$	$F(\mathrm{G},\mathrm{f}_1\_1)\text{-}e\text{-}B(\mathrm{f}_1\_1) = \{\overline{B}, F\}$ *
$F(\mathrm{A},\mathrm{f}_1\_1)\text{-}e\text{-}\overline{B}(\mathrm{C},\mathrm{f}_1\_1) = \{A, C_2\}$	$F(\mathrm{H},\mathrm{f}_1\_1)\text{-}e\text{-}B(\mathrm{f}_1\_1) = \{D_2, A^2\}$
$F(\mathrm{G},\mathrm{f}_1\_1)\text{-}e\text{-}\overline{B}(\mathrm{A},\mathrm{f}_1\_1) = \{C_2, A^2\}$	$F(\mathrm{A2})\text{-}e\text{-}B = \{B, F\}$ *(soliton)*
$F(\mathrm{G},\mathrm{f}_1\_1)\text{-}e\text{-}\overline{B}(\mathrm{B},\mathrm{f}_1\_1) = \{A, A^3, A, \overline{E}$	
$F(\mathrm{G},\mathrm{f}_1\_1)\text{-}e\text{-}\overline{B}(\mathrm{C},\mathrm{f}_1\_1) = \{B, F\}$ *	
$F(\mathrm{H},\mathrm{f}_1\_1)\text{-}e\text{-}\overline{B}(\mathrm{A},\mathrm{f}_1\_1) = \{A, C_2\}$	
$F(\mathrm{H},\mathrm{f}_1\_1)\text{-}e\text{-}\overline{B}(\mathrm{B},\mathrm{f}_1\_1) = \{\overline{E}, A^5\}$	
$F(\mathrm{H},\mathrm{f}_1\_1)\text{-}e\text{-}\overline{B}(\mathrm{C},\mathrm{f}_1\_1) = \{\overline{E}, A^5\}$	
$F(\mathrm{A2},\mathrm{f}_1\_1)\text{-}e\text{-}\overline{B}(\mathrm{A},\mathrm{f}_1\_1) = \{C_1\}$	
$F(\mathrm{A2},\mathrm{f}_1\_1)\text{-}e\text{-}\overline{B}(\mathrm{B},\mathrm{f}_1\_1) = \{A, B^3, \overline{E}\}$	

Table 2. Reactions relation between B, $\overline{B}$, and F mobile self-localizations in rule 110.

3.2 Solitons in Elementary Cellular Automaton Rule 54

ECA rule 54 is a complex CA evolving with an apparently simple system of mobile self-localizations. Its local function is defined as follows:

$$\varphi_{\mathrm{R54}} = \begin{cases} 1 & \text{if} \quad 001,\, 010,\, 100,\, 101 \\ 0 & \text{if} \quad 000,\, 011,\, 110,\, 111. \end{cases} \tag{9}$$

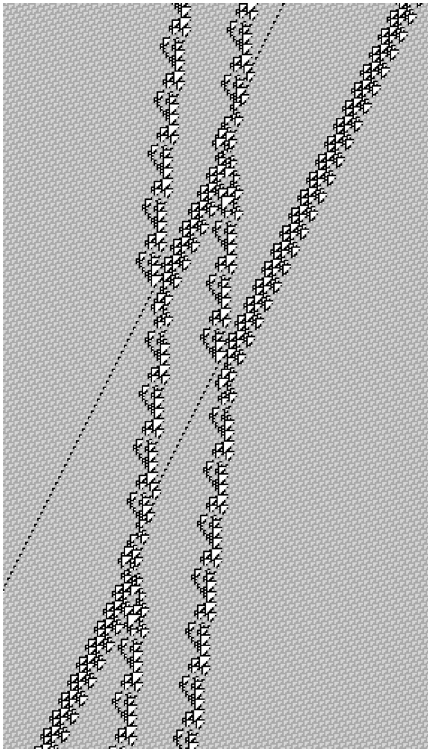


Figure 6. Pseudo-soliton in rule 110.

Figure 7 illustrates the complex dynamics from a typical random initial condition selecting the evolution rule φ_{R54}. A number of mobile self-localizations emerge on its evolution space and collide. Particularly, rule 54 is able to evolve with the emergence of glider gun patterns due to random initial conditions, a feature that cannot be found on any other ECAs. (A glider gun is a complex structure that periodically emits a localization, famously known from the Game of Life CA.)

Detailed analysis of various aspects of ECA rule 54 can be found in the localizations system [19, 57–59], computations [58], collisions [58], and algebraic properties [60, 61]. So generalities can be explored from the rule 54 repository. (For information on gliders in rule 54, visit http://uncomp.uwe.ac.uk/genaro/rule54/glidersRule54.html. For information on the rule 54 repository, visit http://uncomp.uwe.ac.uk/genaro/Rule54.html.)

The set of mobile self-localizations in rule 54 is significantly small compared with rule 110 (which has a base set of 12 mobile self-localizations). Rule 54 basically has four primitive or basic mobile self-

https://doi.org/10.25088/ComplexSystems.21.2.117

localizations (two stationary, one shift-right, and one shift-left displacements) and three kinds of glider guns [58]. This way, basic mobile self-localizations work to produce solitons from multiple collisions. In Figure 8, we present these basic mobile self-localizations following Boccara's notation [19].

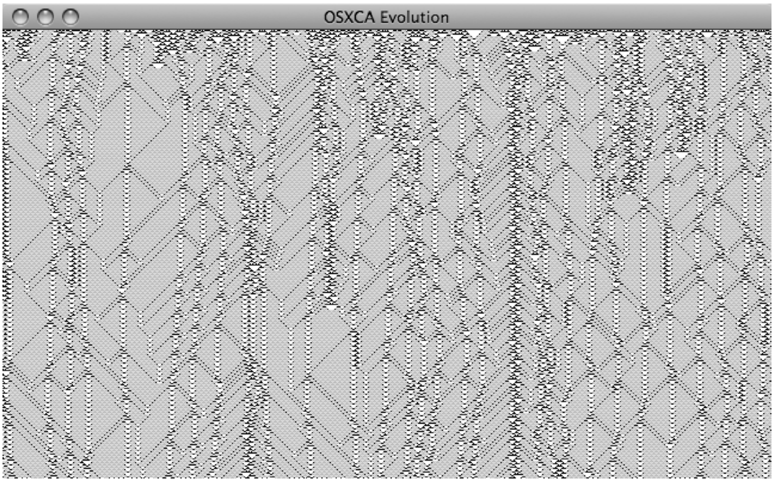


Figure 7. Random evolution in rule 54 on a ring of 644 cells to 375 generations. White cells represent the state 0 and black cells the state 1. The evolution starts with a density at 50%. A filter is applied to get a better view of mobile self-localizations and collisions on its periodic background.

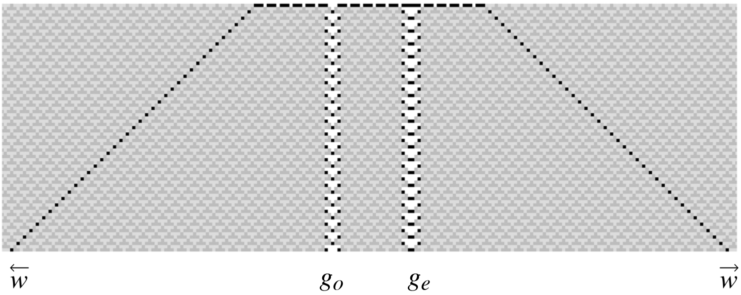


Figure 8. Set of mobile self-localizations with solitonic properties in rule 54.

Particularly, solitons in rule 54 cannot emerge from binary collisions; instead, they are found in multiple collisions. This way, solitons there are on the domain of triple collisions and beyond [58]. Properties for these mobile self-localizations are characterized in Table 3.

Mobile Self-Localization	Shift	Period	Speed	Volume
$\overrightarrow{w}$	2	2	1	2
$\overleftarrow{w}$	-2	2	-1	0–4
g_o	0	4	0	6–2
g_e	0	4	0	7–3

Table 3. Basic mobile self-localizations properties in rule 54.

In this paper, we report an unexplored set of solitonic collisions in rule 54. They are obtained for systematic analysis by reactions across multiple collisions.

Figure 9 presents 14 kinds of solitons constructed in rule 54. It is easy to recognize that you can derive 28 similar reactions in total, because rule 54 is a symmetric rule and therefore you can obtain the next 14 symmetric collisions.

Figure 9(a) and (b) display two pairs of mobile self-localizations producing the same soliton reaction; however, the collision is different because while in (a) the first pair of mobile self-localizations delay its trajectory, in (b) it advances for six cells. Thus these are possible controller intervals of mobile self-localizations trajectories. Similar cases are presented in Figure 9(c), (d), and (e), but these reactions are between three mobile self-localizations.

Figure 9(g) starts a solitonic reaction with more than four mobile self-localizations, but here we employ more space between intervals of mobile self-localizations (see (f) and (h) as well). Hence we can use several mobile self-localizations to preserve the soliton reaction. Noticeably, Figure 9(i), (j), and (k) display three different kinds of collisions to get soliton reactions, employing four mobile self-localizations.

The last set of collisions (Figure 9(l), (m), and (n)) display more large synchronizations of mobile self-localizations, with different intervals and numbers of them. Of course, it is possible to design more sophisticated collisions working with a diverse range of packages of mobile and stationary self-localizations.

3.3 Solitons in Elementary Cellular Automata with Memory

In this section, we will present the simple single-soliton two-component solution [44] for a specific ECAM. The main characteristic is that only one mobile self-localization is processed. Thus a mobile self-localization with shift-right and shift-left displacement always produces the same reaction.

An extensive and systematic analysis is done for ECAM in [62]. From here, we have selected the ECAM rule $\phi_{\text{R9maj:4}}$, because no other rule has the same features.

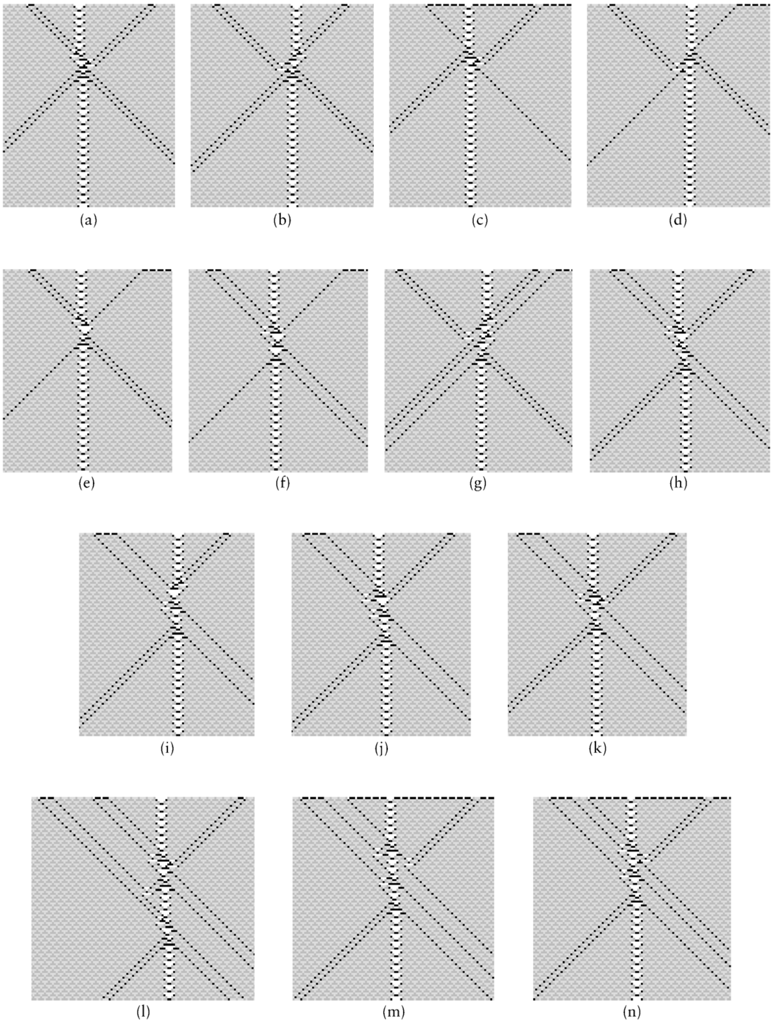


Figure 9. Catalog of soliton collisions in rule 54.

Figure 10 shows the ECA base that shall be enriched with *majority memory function* (see Section 2.2). We study ECA rule 9 because it displays basic interaction of solitons with simple collisions. In Figure 10(a), a single soliton travels along the evolution space, while in (b) a number of interactions occur during a short history, starting from a random initial condition. Extending the evolution space in (c), we can better observe how these solitons emerge in ECA rule 9 and how they collide inside a fast stationary periodic attractor.

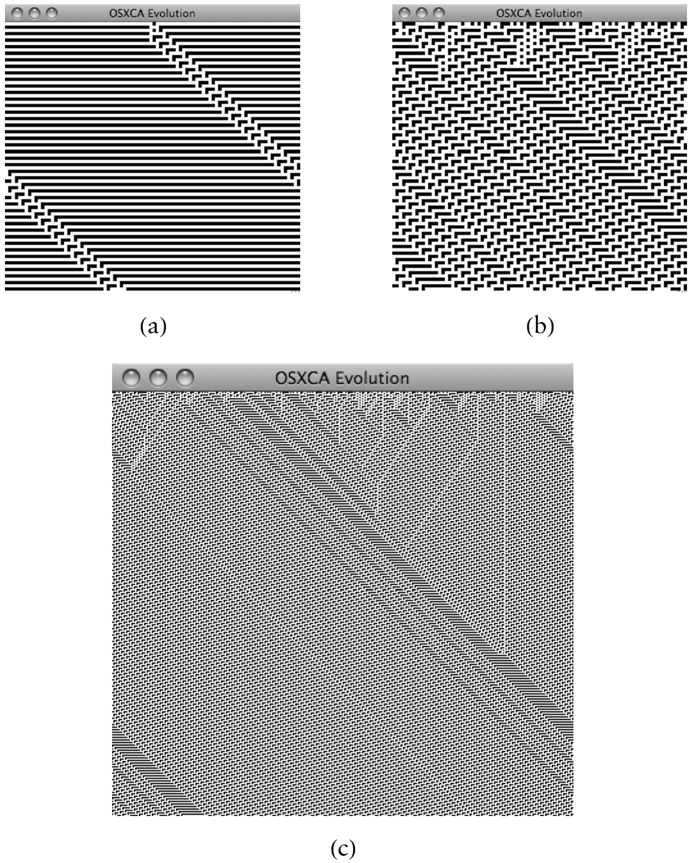


Figure 10. Typical snapshots of ECA rule 9. (a) Evolution starts with a single cell in state 1. (b) A random evolution is presented at 50%. (c) Other random evolutions with small pixels, 360 cells for 331 generations.

In [36, 37, 63], we have demonstrated how ECAs, when enriched with memory, produce different dynamics. Here we will exploit this tool to get simple solitonic reactions.

We use the majority memory with $\tau = 4$ in ECA rule 9. Obtain the ECAM rule $\phi_{\text{R9maj:4}}$, which evolves with two mobile self-localizations emerging on its evolution space: $\mathcal{G}_{\phi_{\text{R9maj:4}}} = \left\{\overrightarrow{p}, \overleftarrow{p}\right\}$. The localization's properties are easy to calculate. The $\overrightarrow{p}$ mobile self-localization has a volume of 5×6 cells, a mass of 12 cells, and moves two cells in five generations (shift-right displacement). The $\overleftarrow{p}$ mobile self-localization has a volume of 5×3 cells, a mass of seven cells, and moves two cells in five generations (shift-left displacement).

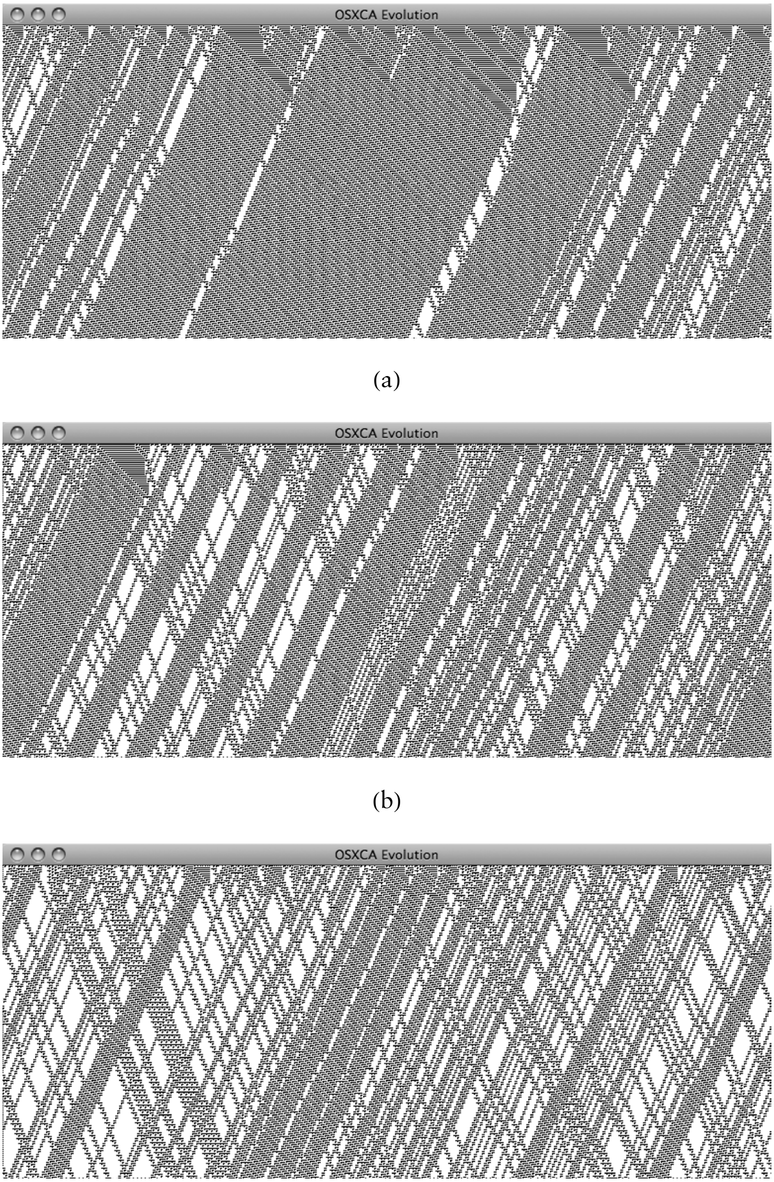
Mobile self-localizations emerging in ECAM $\phi_{\text{R9maj:4}}$ preserve the solitonic reaction after any collision. But there are really two different collisions (two contact points [3] or phases [45] in every mobile self-localization) between $\vec{p}$ and $\overleftarrow{p}$ mobile self-localizations. At the first collision, the soliton is preserved because the sequence is fused in a string of four cells in state one, while the second reaction fuses a string of eight cells in state one. Finally, they open in both mobile self-localizations after exactly seven generations.

Thus the automaton $\phi_{\text{R9maj:4}}$ adjusts every string to always evolve with the same mobile self-localization and soliton reactions, as follows in the next relation of collisions:

$$\vec{p} \to \overleftarrow{p} = \left\{\overleftarrow{p}, \vec{p}\right\}, \text{and} \vec{p} \leftarrow \overleftarrow{p} = \left\{\overleftarrow{p}, \vec{p}\right\}.$$

Figure 11 illustrates three different random initial conditions where the ECAM rule $\phi_{\text{R9maj:4}}$ always evolves in solitonic collisions. The first evolution (Figure 11(a)) starts with an initial density of 10% for state one. The result implies a high production of $\vec{p}$ mobile self-localizations with very few $\vec{p}$ mobile self-localizations, always preserving the solitonic collisions inside bigger fields of $\overleftarrow{p}$. In the opposite case, the second evolution (Figure 11(b)) has an initial density of 80% for state one and again produces high concentrations of $\overleftarrow{p}$ mobile self-localizations with some $\vec{p}$ mobile self-localizations; however, the new solitonic reaction is always preserved. The final evolution (Figure 11(c)) displays 50% of states one and zero, generating a similar distribution of both mobile self-localizations. In all cases, the ECAM rule $\phi_{\text{R9maj:4}}$ evolves any initial condition in solitons. Thus you can begin with any number of mobile self-localizations in $\phi_{\text{R9maj:4}}$ and the solitons are always produced—a characteristic that no conventional ECAs have. However, such behavior can be reproduced identically in other kinds of CAs, including the reversible *block CAs* (also known as partitioned 1D CAs) explored by Wolfram in [48, Chapter 9].

By the way, recently a soliton was discovered in ECA rule 26 (Figure 12).



(a)

(b)

(c)

Figure 11. Typical snapshots of ECAM rule $\phi_{\mathrm{R9maj:4}}$. (a) Starts with an initial density of 10%. (b) Presents an initial density of 80%. (c) Has an initial density of 50%. All evolutions are filtered for the best visualization of mobile self-localization interaction; the evolutions are on a ring of 776 cells for 315 generations.

https://doi.org/10.25088/ComplexSystems.21.2.117

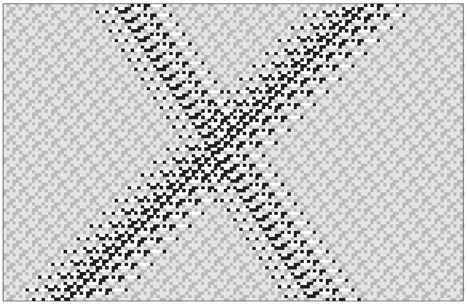


Figure 12. Soliton in class 2 ECA rule 26.

3.4 Computing with Cellular Automata Solitons

Solitons are useful for preserving information such as in the fiber-optic communications field. Of particular interest is whether such solitons could emulate an equivalent Turing machine. Steiglitz et al. have designed a number of results trying to reach this goal in [4–6, 44] and logic gates with solitons in [9, 40]. In [63], for example, the authors have developed a very simple substitution system as an implementation of the function addToHead(), based on soliton reactions, where such mechanisms can also be designed as simple colliders [38]. Figure 13 displays such operations between two mobile self-localizations in the ECAM $\phi_{\mathrm{R30maj:8}}$ [36, 63].

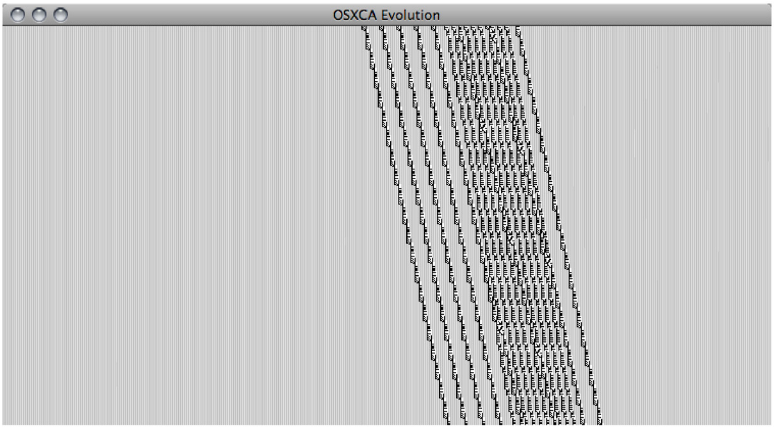


Figure 13. The ECAM $\phi_{\mathrm{R30maj:8}}$ presents a solitonic collision that can be coded for any n, $m \in \mathcal{Z}^{+}$, such that $p^{n}_{\phi_{\mathrm{R30maj:8}}} \to q^{m}_{\phi_{\mathrm{R30maj:8}}}$ is always a soliton.

https://doi.org/10.25088/ComplexSystems.21.2.117

During the last decade, we have seen a number of significant advances in work with solitons for modeling unconventional computing devices [5–7, 9, 40, 44, 63, 64, 65]. As a result, we can see how solitons could be important in developing computable devices in the construction of equivalent Turing machines. We also want to recall the results obtained in ECA rule 110, where a cyclic tag system was developed to perform computation-based collisions with a large number of mobile self-localizations on an incredible global synchronization in millions of cells. Solitonic reactions were very useful to write binary data and preserve information in the whole mechanism. For full details, see [47–49, 51, 52]. (Details and large snapshots about the cyclic tag system working in rule 110 can be found at http://uncomp.uwe.ac.uk/genaro/rule110/ctsRule110.html.)

Figure 14 illustrates how information, encoded in a configuration of solitons (localizations), may be conserved and recognized. For example, a collision between trains A^4 of gliders with glider $\overline{E}$ leads to formation of new traveling localizations, or bits. Moreover, when a bit is already on the tape—represented by a configuration of four stationary C_2 localizations—the train of gliders $\overline{E}$, traveling east, does not destroy the bit. The train just passes through the configuration of localizations (for full details, see [52]).

4. Conclusions

We have reported a complete number of solitons in elementary cellular automaton (ECA) rule 110 from binary collisions. For ECA rule 54, we have reported a new number of collisions that yield solitons, which could be manipulated to develop computable devices, or even further, complex constructions based on solitonic reactions. Finally, we have characterized a simple single-soliton two-component solution with a simple ECA with memory (ECAM) $\phi_{\text{R9maj:4}}$, where mobile self-localizations always work as solitons even starting from random initial conditions, because each soliton is always constructed from $\phi_{\text{R9maj:4}}$.

With regards to memory effect in CAs [33, 34], we recently have studied how a memory function helps to describe dynamics properties that are not evident at the first instance [36, 37, 63]. In the present paper, the majority memory selected in ECA rule 9 opens a new evolution rule ECAM $\phi_{\text{R9maj:4}}$ able to simulate solitons. Of course, fragments of the original evolution rule determine such dynamics.

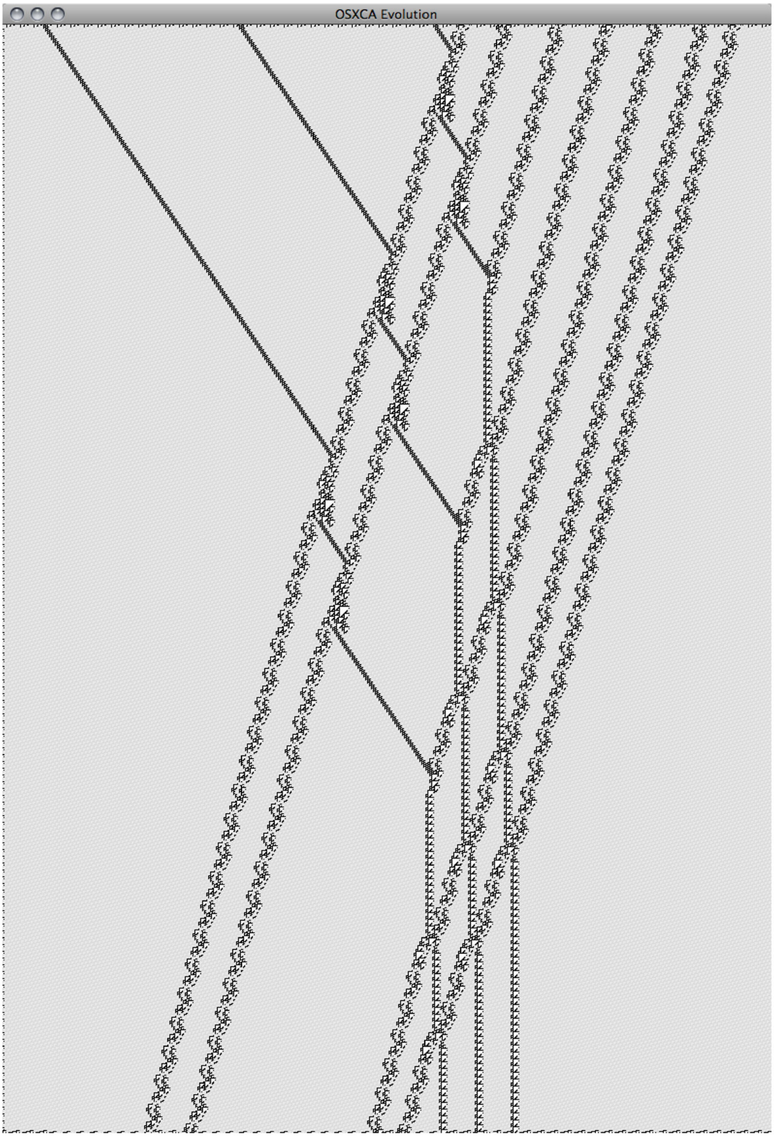


Figure 14. Solitons working to yield, handle, and control bits in one cyclic tag system in rule 110. This evolution begins with 793 cells to 1144 generations; the evolution is filtered, suppressing its ether (periodic background).

References

[1] J. S. Russell, "Report of Waves," in *Report of the British Association for the Advancement of Science*, London: Taylor and Francis, 1845 pp. 311–390.

[2] D. J. Korteweg and G. de Vries, "On the Change of Form of Long Waves Advancing in a Rectangular Canal, and on a New Type of Long Stationary Waves," *Philosophical Magazine*, **39**(5), 1895 pp. 422–443.

[3] J. K. Park, K. Steiglitz, and W. P. Thurston, "Soliton-Like Behavior in Automata," *Physica D: Nonlinear Phenomena*, **19**(3), 1986 pp. 423–432. doi:10.1016/0167-2789(86)90068-0.

[4] K. Steiglitz, I. Kamal, and A. Watson, "Embedding Computation in One-Dimensional Automata by Phase Coding Solitons," *IEEE Transactions on Computers*, **37**(2), 1988 pp. 138–145. doi:10.1109/12.2143.

[5] M. H. Jakubowski, K. Steiglitz, and R. Squier, "Computing with Solitons: A Review and Prospectus," *Multiple-Valued Logic*, **6**(5–6), 2001 pp. 439–462. http://www.cs.princeton.edu/~ken/mvl.pdf.

[6] D. Rand, K. Steiglitz, and P. R. Prucnal, "Signal Standardization in Collision-Based Soliton Computing," *International Journal of Unconventional Computing*, **1**(1), 2005 pp. 31–45. http://www.cs.princeton.edu/~ken/IJUC_final.pdf.

[7] P. Siwak, "Iterons of Automata," in *Collision-Based Computing* (A. Adamatzky, ed.), London: Springer, 2002 pp. 299–354.

[8] Y. Aizawa, I. Nishikawa, and K. Kaneko, "Soliton Turbulence in One-Dimensional Cellular Automata," *Physica D: Nonlinear Phenomena*, **45**(1–3), 1990 pp. 307–327. doi:10.1016/0167-2789(90)90191-Q.

[9] S. Blair and K. Wagner, "Gated Logic with Optical Solitons," in *Collision-Based Computing* (A. Adamatzky, ed.), London: Springer, 2002 pp. 355–380.

[10] N. Margolus, T. Toffoli, and G. Vichniac, "Cellular-Automata Supercomputers for Fluid-Dynamics Modeling," *Physical Review Letters*, **56**(16), 1986 pp. 1694–1696. doi:10.1103/PhysRevLett.56.1694.

[11] T. Toffoli and N. Margolus, *Cellular Automata Machines: A New Environment for Modeling*, Cambridge, MA: The MIT Press, 1987.

[12] A. S. Davydov, *Solitons in Molecular Systems*, 2nd ed., Boston: Kluwer Academic Publishers, 1991.

[13] N. Akhmediev and A. Ankiewicz, eds., *Dissipative Solitons*, Berlin: Springer, 2005.

[14] L. I. Manevich and V. V. Simmons, *Solitons in Macromolecular Systems*, New York: Nova Biomedical Books, 2008.

[15] T. Dauxois and M. Peyrard, *Physics of Solitons*, Cambridge, UK: Cambridge University Press, 2006.

[16] A. Hasegawa and M. Matsumoto, *Optical Solitons in Fibers*, 3rd ed., Springer Series in Photonics, New York: Springer-Verlag, 2003.

[17] M. Gardner, "Mathematical Games: The Fantastic Combinations of John H. Conway's New Solitaire Game Life," *Scientific American*, **223**, 1970 pp. 120–123.

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

[19] N. Boccara, J. Nasser, and M. Roger, "Particlelike Structures and Their Interactions in Spatiotemporal Patterns Generated by One-Dimensional Deterministic Cellular-Automaton Rules," *Physical Review A*, **44**(2), 1991 pp. 866–875. doi:10.1103/PhysRevA.44.866.

[20] J. T. Lizier, M. Prokopenko, and A. Y. Zomaya, "Information Transfer by Particles in Cellular Automata," in *Proceedings of the 3rd Australian Conference on Progress in Artificial Life (ACAL07)*, Gold Coast, Australia (M. Randall, H. A. Abbass, and J. Wiles, eds.), Berlin: Springer-Verlag, 2007 pp. 49–60.

[21] G. J. Martínez, H. V. McIntosh, and J. C. Seck-Tuoh-Mora, "Gliders in Rule 110," *International Journal of Unconventional Computing*, **2**(1), 2006 pp. 1–49. http://uncomp.uwe.ac.uk/genaro/Papers/ Papers_on_CA _files/MARTINEZ.pdf.

[22] A. Moreira, N. Boccara, and E. Goles, "On Conservative and Monotone One-Dimensional Cellular Automata and Their Particle Representation," *Theoretical Computer Science*, **352**(2), 2004 pp. 285–316. doi:10.1016/j.tcs.2004.06.010.

[23] M. Pivato, "Defect Particle Kinematics in One-Dimensional Cellular Automata," *Theoretical Computer Science*, **377**(1), 2007 pp. 205–228. doi:10.1016/j.tcs.2007.03.014.

[24] M. Delorme and J. Mazoyer, "Signals on Cellular Automata," in *Collision-Based Computing* (A. Adamatzky, ed.), London: Springer, 2002 pp. 231–276.

[25] H. Umeo and K. Kubo, "A Seven-State Time-Optimum Square Synchronizer," *Lecture Notes in Computer Science*, **6350**, 2010 pp. 219–230. doi:10.1007/978-3-642-15979-4_24.

[26] N. Margolus, "Physics-Like Models of Computation," *Physica D: Nonlinear Phenomena*, **10**(1–2), 1984 pp. 81–95. doi:10.1016/0167-2789(84)90252-5.

[27] S. Wolfram, "Cellular Automaton Fluids 1: Basic Theory," *Journal of Statistical Physics*, **45**(3–4), 1986 pp. 471–526. doi:10.1007/BF01021083.

[28] S. Wolfram, *Theory and Applications of Cellular Automata*, Singapore: World Scientific Press, 1986.

[29] B. Chopard and M. Droz, *Cellular Automata Modeling of Physical Systems*, Cambridge, UK: Cambridge University Press, 1998.

[30] N. Margolus, "Universal Cellular Automata Based on the Collisions of Soft Spheres," in *Collision-Based Computing* (A. Adamatzky, ed.), London: Springer, 2002 pp. 107–134.

[31] R. Alonso-Sanz and M. Martin, "Elementary Cellular Automata with Memory," *Complex Systems*, **14**(2), 2003 pp. 99–126. http://www.complex-systems.com/pdf/14-2-1.pdf.

[32] R. Alonso-Sanz and M. Martin, "Elementary Cellular Automata with Elementary Memory Rules in Cells: The Case of Linear Rules," *Journal of Cellular Automata*, **1**(1), 2006 pp. 71–87.

[33] R. Alonso-Sanz, *Cellular Automata with Memory*, Philadelphia: Old City Publishing, 2008.

[34] R. Alonso-Sanz, *Discrete Systems with Memory*, Hackensack, NJ: World Scientific, 2011.

[35] M. L. Minsky, *Computation: Finite and Infinite Machines*, Englewood Cliffs, NJ: Prentice Hall, 1967.

[36] G. J. Martínez, A. Adamatzky, R. Alonso-Sanz, and J. C. Seck-Tuoh-Mora, "Complex Dynamics Emerging in Rule 30 with Majority Memory," *Complex Systems* **18**(3), 2010 pp. 345–365. http://www.complex-systems.com/pdf/18-3-5.pdf.

[37] G. J. Martínez, A. Adamatzky, J. C. Seck-Tuoh-Mora, and R. Alonso-Sanz, "How to Make Dull Cellular Automata Complex by Adding Memory: Rule 126 Case Study," *Complexity*, **15**(6), 2010 pp. 34–49. doi:10.1002/cplx.20311.

[38] G. J. Martínez, A. Adamatzky, C. R. Stephens, and A. F. Hoeflich, "Cellular Automaton Supercolliders," *International Journal of Modern Physics C*, **22**(4), 2011 pp. 419–439.

[39] S. V. Manakov, "On the Theory of Two-Dimensional Stationary Self-Focusing of Electromagnetic Waves," *Soviet Physics JETP*, **38**(2), 1974 pp. 248–253.

[40] A. Adamatzky, "New Media for Collision-Based Computing," in *Collision-Based Computing* (A. Adamatzky, ed.), London: Springer, 2002 pp. 411–442.

[41] L. Chua, *A Nonlinear Dynamics Perspective of Wolfram's New Kind of Science,* Vol. 1–5, Hackensack, NJ: World Scientific, 2006–2012.

[42] A. Adamatzky and G. J. Martínez, "On Generative Morphological Diversity of Elementary Cellular Automata," *Kybernetes*, **39**(1), 2010 pp. 72–82. doi:10.1108/03684921011021282.

[43] F. Chen, L. Shi, G. Chen, and W. Jin, "Chaos and Gliders in Periodic Cellular Automaton Rule 62," *Journal of Cellular Automata*, forthcoming.

[44] K. Steiglitz, "Time-Gated Manakov Spatial Solitons Are Computationally Universal," *Physical Review E*, **63**(1), 2000 pp. 016608. doi:10.1103/PhysRevE.63.016608.

[45] G. J. Martínez, H. V. McIntosh, J. C. Seck-Tuoh-Mora, and S. V. Chapa-Vergara, "Determining a Regular Language by Glider-Based Structures Called Phases f_i_1 in Rule 110," *Journal of Cellular Automata*, **3**(3), 2008 pp. 231–270.

[46] H. V. McIntosh, *Rule 110 as It Relates to the Presence of Gliders*, Technical report, Departamento de Aplicación de Microcomputadoras, Instituto de Ciencias, Universidad Autónoma de Puebla, Mexico, 1999. http://delta.cs.cinvestav.mx/~mcintosh/cellularautomata/Papers_files/rule110.pdf.

[47] M. Cook, "Universality in Elementary Cellular Automata," *Complex Systems*, **15**(1), 2004 pp. 1–40. http://www.complex-systems.com/pdf/15-1-1.pdf.

[48] S. Wolfram, *A New Kind of Science*, Champaign, IL: Wolfram Media, Inc., 2002.

https://doi.org/10.25088/ComplexSystems.21.2.117

[49] H. V. McIntosh, *"Rule 110 Is Universal!" A Gamin's Guide To Goldbergean Gadgeteering*, Technical report, Departamento de Aplicación de Microcomputadoras, Instituto de Ciencias, Universidad Autónoma de Puebla, Mexico, 2002. http://delta.cs.cinvestav.mx/~mcintosh/cellularautomata/Papers_files/texlet.pdf.

[50] T. Neary and D. Woods, "P-Completeness of Cellular Automaton Rule 110," in *International Colloquium on Automata Languages and Programming (ICALP06)*, Venice, Italy (M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, eds.), London: Springer, 2006 pp. 132–143. http://www.dna.caltech.edu/~woods/download/NearyWoodsBCRI-04-06.pdf.

[51] M. Cook, "A Concrete View of Rule 110 Computation," in *Proceedings of the International Workshop on the Complexity of Simple Programs (CSP08)*, Cork, Ireland (T. Neary, D. Woods, A. K. Seda, and N. Murphy, eds.), Electronic Proceedings in Theoretical Computer Science, 2009 pp. 31–55. doi:http://dx.doi.org/10.4204/EPTCS.1.4.

[52] G. J. Martínez, H. V. McIntosh, J. C. Seck-Tuoh-Mora, and S. V. Chapa-Vergara, "Reproducing the Cyclic Tag System Developed by Matthew Cook with Rule 110 Using the Phases f_1_1," *Journal of Cellular Automata*, **6**(2–3), 2011 pp. 121–161.

[53] G. J. Martínez and H. V. McIntosh. "ATLAS: Collisions of Gliders Like Phases of Ether in Rule 110." (Aug 14, 2001) http://uncomp.uwe.ac.uk/genaro/Papers/Papers_on_CA_files/ATLAS/bookcollisions.html.

[54] G. J. Martínez, H. V. McIntosh, J. C. Seck-Tuoh-Mora, and S. V. Chapa-Vergara, "Rule 110 Objects and Other Constructions Based-Collisions," *Journal of Cellular Automata*, **2**(3), 2007 pp. 219–242. http://eprints.uwe.ac.uk/10579/1/JCA_0010_Martinez.pdf.

[55] G. J. Martínez. "Solitones en el autómata celular unidimensional regla 110." (Apr 8, 2002) http://uncomp.uwe.ac.uk/genaro/Papers/Papers_on_CA_files/solitonesR110/solitones-110.html.

[56] G. J. Martínez. "Encontrando solitones en el autómata celular regla 110." (Jan 25, 2002) http://www.tuobra.unam.mx/obrasPDF/publicadas/020802115837.html.

[57] J. E. Hanson and J. P. Crutchfield, "Computational Mechanics of Cellular Automata: An Example," *Physics D: Nonlinear Phenomena*, **103**(1–4), 1997 pp. 169–189. doi:10.1016/S0167-2789(96)00259-X.

[58] G. J. Martínez, A. Adamatzky, and H. V. McIntosh, "Phenomenology of Glider Collisions in Cellular Automaton Rule 54 and Associated Logical Gates," *Chaos, Solitons & Fractals*, **28**(1), 2006 pp. 100–111. doi:10.1016/j.chaos.2005.05.013.

[59] G. J. Martínez, A. Adamatzky, and H. V. McIntosh, "On the Representation of Gliders in Rule 54 by de Bruijn and Cycle Diagrams," *Lecture Notes in Computer Science*, **5191**, 2008 pp. 83–91. doi:10.1007/978-3-540-79992-4_11.

[60] M. Redeker, "Gliders and Ether in Rule 54." http://arxiv.org/abs/1007.2920v1.

[61] B. Martin, "A Group Interpretation of Particles Generated by One-Dimensional Cellular Automaton, Wolfram's Rule 54," *International Journal of Modern Physics C*, **11**(1), 2000 pp. 101–123.

[62] G. J. Martínez, A. Adamatzky, and R. Alonso-Sanz, "Memories Dilute Complex Systems, Knowledge Dilutes Extremism," forthcoming.

[63] G. J. Martínez, A. Adamatzky, and R. Alonso-Sanz, "Complex Dynamics of Elementary Cellular Automata Emerging from Chaotic Rules," *International Journal of Bifurcation and Chaos*, **22**(2), 2012 pp. 1250023-13.

[64] A. Adamatzky, *Computing in Nonlinear Media and Automata Collectives*, Bristol, UK: Institute of Physics, 2001.

[65] A. Adamatzky, ed., *Collision-Based Computing*, London: Springer, 2002.

https://doi.org/10.25088/ComplexSystems.21.2.117