Iterons: the emergent coherent structures of IAMs

Paweł Siwak
Poznań University of Technology
60-965 Poznań, Poland
siwak@sk-kari.put.poznan.pl

1.1 Introduction

Iterons of automata [15, 18] are periodic coherent propagating structures (substrings of symbols) that emerge in cellular nets of automata. They are like fractal objects; they owe their existence to iterated automata maps (IAMs) performed over strings. This suggests that the iterating of (automata) maps is a fundamental mechanism that creates localized persistent structures in complex systems.

Coherent objects appear in the literature under various names; there are waves, building blocks, particles, signals, discrete solitons, defects, gliders, localized moving structures, light bullets, propagating fronts, and many other entities including those with the characteristic suffix –on, like fluxons, cavitons, excitons, explosons, pulsons, virons, magnons, phonons, oscillons, peakons, compactons, etc., etc.

In this paper we present a unified automaton approach to the processing mechanisms capable of supporting such coherent entities in evolving strings.

The iterons comprise of particles and filtrons. The particles, or signals, are well known [2, 3, 6, 10, 12] in cellular automata (CAs) where iterated *parallel* processing of strings occurs. They spread and carry local results, synchronize various events, combine information, encode and transform data, and carry out many other actions necessary to perform a computation, to complete a global pattern formation process in extended dynamical systems, or simply to assure stability of a complex system.

The filtrons form another new [14-18] class of coherent objects supported by IAMs. They emerge in iterated *serial* string processing which is a sort of recursive digital filtering (IIR filtering) [13]. In many aspects the filtrons are like solitons known from nonlinear physics; e.g. they pass through one another, demonstrate elastic collisions, undergo fusion, fission and annihilation, and form breathers as well as other complex entities. The first observation of filtron type binary objects has been done by Park, Steiglitz and Thurston [13]. They introduced the model called parity rule filter CA, and showed that it is capable of supporting coherent periodic substrings with soliton-like behavior. Now, a number of particular models exist that support filtrons. These are iterating automata nets [4], filter CAs [1, 8, 11], soliton CAs [7, 11, 21], higher order CAs [2], sequentially updated CAs [4], integrable CAs [5], iterated arrays [6], IIR digital filters or filter automata [14-18], discrete versions of classical soliton equations (KdV, KP, L-V) [5, 20, 22], and fast rules [1, 11]. Some new models like box-ball systems [20, 22] and crystal systems [9] were introduced quite recently.

All these models and their coherent structures can be described by automata and their iterons [14-18]. In our approach the automata are a sort of medium (or complex system) and the passing strings resemble disturbances that propagate throughout this medium (or system). We identify the iterons by particular sequences of automaton operations. One could call these sequences an active mode of automaton medium.

For the filtrons we consider one-way 1-d homogeneous net with an automaton M. The symbols of evolving strings are distinguished from the states of automata. In such nets the strings flow throughout automata and evolve, the IAM is performed over a string in a natural way. The filtrons are special M-segments that involve certain sequences of operations of automaton M. These sequences are related to a class of paths on the automaton state diagram.

For the particles we consider de Bruijn graph G of their CA. This graph represents the constraints on possible sequences of elementary rules (ERs) used to update any segment of symbols. Again, the particles (as G-segments) are related to special sequences of ERs, which form paths on de Bruijn graph.

The characteristic sequences of operations associated with iterons lead to analytical tools in the analysis (and synthesis) of coherent structures in complex systems. So-called ring computation [14, 16, 17] has been already proposed to this end.

We present some examples of various phenomena of interacting filtrons, like multifiltron collisions, fusion, fission, and spontaneous decay or quasi-filtrons, and also some automata capable of supporting these events.

1.2 Automata and filtrons

Automata maps can be used to perform the processing of strings either in serial or in parallel manner. In both cases we characterize automata maps by some elementary operations; state-implied functions in serial IAMs and ERs in parallel IAMs.

1.2.1 Automaton and its state-implied functions

A Mealy type automaton M with outputs and an initial state is defined to be a system $M = (S, \Sigma, \Omega, \delta, \beta, s_0)$, where S, Σ and Ω are nonempty, finite sets of—respectively—states, inputs and outputs, $\delta: S \times \Sigma \rightarrow S$ is called the next state (or transition) function of M, and $\beta: S \times \Sigma \rightarrow \Omega$ is called the output function of M. Symbol $s_0 \in S$ denotes the initial state of M.

The automaton converts sequences of symbols (finite or infinite words). For each symbol σ_i read from an input string it responds with an associated output symbol ω_i which is a consecutive element of the resulting string. The input string is read sequentially from left to right, one symbol at each instant τ of time, in such a way that $\delta(s(\tau), \sigma(\tau)) = s(\tau+1)$ and $\beta(s(\tau), \sigma(\tau)) = \omega(\tau)$ for all $\tau = 1, 2, \ldots$.

Next state and output functions of automata are presented in tables or in a graph form that is called the state diagram of automaton. For any $s \in S$ and $\sigma \in \Sigma$ that imply $t = \delta(s, \sigma)$ and $\omega = \beta(s, \sigma)$ in Mealy model, there is a directed edge on the graph going from node s to node t, and labeled by σ/ω. In Moore model the output function is defined by $\lambda(s(\tau)) = \omega$ thus the outputs $\lambda(s)$ are attached to the nodes.

To allow the iterations we apply a unified set of symbols $A = \Sigma = \Omega = \{0, 1, ..., m\}$. Converted strings, when listed one under another, form an ST (space-time) diagram. It is often convenient to shift each output string by q positions to the left with respect to its input string. We say in such a case that the shift q has been applied to a diagram.

We also describe the automaton's operation by (state-implied) functions $f_s : A \rightarrow A$. They depend on states and are such that $f_s(a_i) = \beta(s, a_i)$ for all $s \in S$ and $a_i \in A$. The succession of outputs of the automaton are then: *next* $[f_s(a_i)] = f_{\delta(s, a_i)}(a_{i+1})$.

It is clear that the labeled path on state diagram of the automaton implied by any input string can be viewed as a sequence of operations f_s. Any input string determines the sequence of automaton operations; in a sense, the string and automaton interact.

1.2.2 Filtrons

We treat the automata lines as a medium or system and assume that its mode (idle or excited) decides on the existence or absence of coherent structures. Thus we use automata in the role of substring recognizers. The idea is as follows. Suppose that the automaton M reads a string $\ldots a_1 \ldots a_L \ldots$. Each time when M leaves a fixed (starting) state s under a symbol a_1 this transition is treated as the beginning of a substring, and activation of M. Also, each time when M enters some fixed state t (lets call it final state) under a symbol a_L we say that the end of the substring $a_1 \ldots a_L$ is recognized, and M is extinguished. The substring $a_1 \ldots a_L$ is said to be the M-segment.

Consider now some special M-segments. We assume strings $\ldots 0 a_1 \ldots a_L 0 \ldots$ where symbol 0 represents a background; $\delta(s_0, 0) = s_0$. For given M we choose initial state s_0 to be the starting state as well as the final state. In general case one can use another more complicated selection; e.g. the subsets of automaton states can be chosen as starting states and/or as final states, or even these sets can evolve in time.

Our basic coherent structure, the filtron is defined as follows [14, 15]. By a p-periodic filtron $a^t\_$ of an automaton M we understand a string $a_1^t\, a_2^t \ldots a_{L_t}^t$ of symbols from A with $a_1^t \neq 0$, such that during the iterated processing of configuration $a^t = \ldots 0 a^t\_ 0 \ldots$ by the automaton M the following conditions are satisfied for all $t = 0, 1, \ldots$:

- the string $a^t\_$ occurs in p different forms (filtron's orbital states), with $0 < L_t < \infty$,
- the string $a^t\_$ is an M-segment.

When a number of extinctions of given M still occurs before the last element of the string segment $a^t\_$ is read by M, we say that $a^t\_$ is a multi-M-segment string. Multi-M-segment strings lead to complex filtrons.

1.2.3 The models that support filtrons

The first model shown to be capable of supporting coherent periodic substrings with soliton-like behavior was parity rule filter CA [13]. The PST (Park-Steiglitz-Thurston) model consists of a special ST-window (called here FCA window) and a parity update function. The string processing, $a^t \to a^{t+1}$, proceeds as follows.

Assume a configuration at time t, $a^t = \ldots\ a_i^t \ldots\ = \ldots 0\, a_1^t \ldots a_{L_t}^t\ 0\ldots$, of elements from $A = \{0, 1\}$ such that: $0 \le t < \infty$, $-\infty < i < \infty$, $1 \le L_t < \infty$ and $a_1^t \ne 0$. The model (f_{PST}, r), with $r \ge 1$, computes the next configuration a^{t+1} at all positions i ($-\infty < i < \infty$) in such a way that: $a_i^{t+1} = f_{\mathrm{PST}}\,(a_i^t,\ a_{i+1}^t, \ldots, a_{i+r}^t,\ a_{i-r}^{t+1},\ a_{i-r+1}^{t+1}, \ldots, a_{i-1}^{t+1}) = 1$ if $S_{i,t}$ is even but not zero, and otherwise $a_i^{t+1} = 0$; $S_{i,t}$ is the sum of all arguments (window elements). Zero boundary conditions are assumed, which means that the segment $a^t\_$ is always preceded in configuration $a^t = \ldots 0 a^t\_0\ldots$ at the left side by enough zeros.

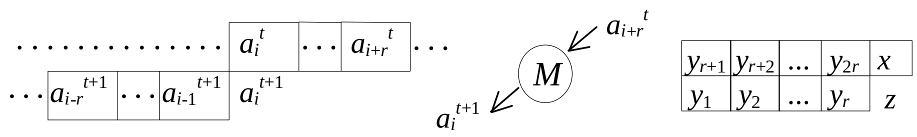

Figure 1. FCA window, and its automaton view: input (x), state (y_i) and output (z) variables.

The map $f_{\mathrm{PST}} : A^{2r+1} \to A$ is a Boolean function. With the variables shown on the right in figure 1, the function f_{PST} is $z = f\,(y_{r+1}, y_{r+2}, \ldots, y_{2r}, x, y_1, y_2, \ldots, y_r)$ and is given by: $z = y_1 \oplus y_2 \oplus \ldots \oplus y_{2r} \oplus x \oplus b$ where $b = \bar{y}_1 \cdot \bar{y}_2 \cdot \ldots \cdot \bar{y}_{2r} \cdot \bar{x}$ and $\oplus$ is *XOR* operation. The FCA window slides to the right, and f_{PST} is a nonlinear function.

The PST model is an automaton [14, 16]. Processing of strings is performed in the cycles of operations (N, A, ..., A) or NA^r where N is the negate operation and A is the accept operation. The processing starts with the first nonzero symbol $a_i = 1$ entering the window, and stops when the substring $*0^r$ (* is an arbitrary symbol) coincides with the cycle. Such set $\{*0^r\}$ is called a reset condition. The example of IAM for r = 3 is shown below. The applied shift is $q = 0$. The collision is nondestructive.

```
 0   ..1001----...111-----1111----...................
 1   .....11------.1101----1111----..................
 2   ......1001----.1011----1111----.................
 3   .........11------11100001111-----...............
 4   ..........1001----110100101101----..............
 5   .............1100000101101001011------..........
 6   ..............10010100101101001001----..........
 7   .................11100001111-----11------.......
 8   ..................110100101101----1001----......
 9   ...................101101001011----..11------...
10   .....................1111----111-----.1001----..
11   ......................1111----1101----...11----..
```

We show the filtrons on ST diagrams using a special convention. The symbol $0 \in A$ in a string represents a quiescent signal or a background, but sometimes it belongs to an active part of a string (M-segment). Thus, we use three different characters to present it on the ST diagrams. A dot ″.″ denotes zeros read by the automaton, which is inactive. A dash ″-″ represents tail zeros of an M-segment, that is all consecutive zeros preceding immediately the extinction of the automaton. Remaining zeros are shown as the digit 0. Moreover, all those symbols that activate the automaton are printed in bold. This convention helps one to recognize whether any two filtrons are distant, adjacent or overlap and form a complex object.

In the last few years, there has been an increasing interest in looking for models that support filtrons. We mentioned them in introduction. For all of them the equivalent automata have been found [14-18]. Some examples of automata that represent the models which were based on FCA window are:

$z = y_1 \oplus ... \oplus y_{2r} \oplus \overline{x} \oplus \overline{y}_1 ... \overline{y}_r \overline{y}_{r+2} ... \overline{y}_{2r} \overline{x}$; $(r \geq 0)$, for Ablowitz model [1],
$z = y_1 \oplus ... \oplus y_r \oplus y_{r+2} \oplus ... \oplus y_{2r+1} \oplus x \oplus y_1 ... y_{2r+1}\ x$; $(r > 0)$, for Jiang model [11],
$z = (y_{r+1} = 0) \wedge (y_{r+1} + ... + y_{\infty} > y_1 + ... + y_r)$; $(r > 0)$, for TS model [21],
$z = y_3 \oplus y_1 y_4 \oplus y_2 x$; $(r = 2)$, for BSR-1 model given by formula 1 in [5], and
$z = y_4 \oplus y_1 y_2 \oplus y_2 y_3 \oplus y_5 y_6 \oplus y_6 x$; $(r = 3)$, for BSR-23 model (formula 23 in [5]).
Some other automata for more recent models are given further.

1. 3 Cellular automata and particles

Now we will present the iterons that emerge in iterated parallel string processing performed in cellular automata. These are widely known [2, 6, 10] and called particles or signals. They are associated with some segments of strings. Usually, these are periodic objects which can be localized on some chosen area of ST diagram. In the simplest cases, when the background is steady, it is not difficult to extract and identify these objects. In the computing theory such particles are considered frequently. They are treated as functional components in the hierarchical description of a computation. The geometrical analysis of possible results of the interactions of particles dominates in this approach [6]. Another technique aimed at dealing with particles on periodic background was based on a filtering of ST diagrams by a sequential transducer [10]. Also, statistical analysis of ST diagram segments has been proposed [3]. However, there are many more complicated periodic entities. Some particles are like defects on a spatially periodic background or boundaries separating two phases of it. The defects can join into complexes. The background of what one would call particle can be periodically impure, and the particle itself may even contain some insertions. Such complicated boundary areas can be distant, adjacent or may overlap. Even more difficult is the case when neighborhood window is not connected (there are “holes” in a processing window). Moreover, in many cases the complexes of various entangled particles occur. This is why the description of particles and the complete analysis of their collisions results is not a trivial task [2, 3, 6, 10, 12].

1.3.1 Cellular automaton and its elementary rules

Cellular automata are defined by $CA = (A, f)$ where A is a set of symbols called the states of cells, $f : A^n \rightarrow A$ is a map called the local function or rule of CA, and $n = 2r + 1$ is the size of neighborhood (or processing window) with r left and r right neighbors. Typically, especially when $|A| = 2$ and r is small, the rule is determined by the number $\sum_{j=0}^{j=2^n-1} f(w_j) \cdot 2^j$; w_j denotes the neighborhood state (contents of the window).

The 1-d CA model converts the strings of symbols. We denote them by $a^\tau = ..., a_i^\tau, a_{i+1}^\tau, a_{i+2}^\tau, ...$, and call the current configuration of a CA. The next configuration $a^{\tau+1}$ is a result of updating simultaneously all the symbols from a^τ; for all $-\infty < i < +\infty$ we have $a_i^{\tau+1} = f(a_{i-r}^\tau, a_{i-r+1}^\tau, ..., a_i^\tau, ..., a_{i+r}^\tau)$. The resulting global CA map $a^\tau \to a^{\tau+1}$ is denoted by $\gamma(a^\tau) = a^{\tau+1}$.

The function f can be specified as the set of all $(n+1)$-tuples $(a_1, a_2, \ldots, a_{n+1}) \in A^{n+1}$, with $f(a_1, a_2, \ldots, a_n) = a_{n+1}$; these represent simply the single values of function f. Any such $(n+1)$-tuple is called here the elementary rule (ER) of the CA model. The sequences of ERs will be used to recognize the particles of CAs.

Note that we have a hierarchy of processing in CAs; there are ERs, sets of ERs (level of particles), local function f, global function γ, and iterations of function γ.

1.3.2 De Bruijn graph

De Bruijn graphs are used to build the automata that scan sequentially a string to detect some specific substrings of symbols. For a CA with n-wide window ($n = 2r+1$), the Moore automaton $G_n = (A^n, A, A, \delta, \lambda, s_0)$ that mimics the sliding window is defined by the next state function $\delta((a_1, ..., a_n), a_{n+1}) = (a_2, ..., a_{n+1})$, and the output function $\lambda(a_1, ..., a_n) = f(a_1, ..., a_n)$ identical with the rule f of CA.

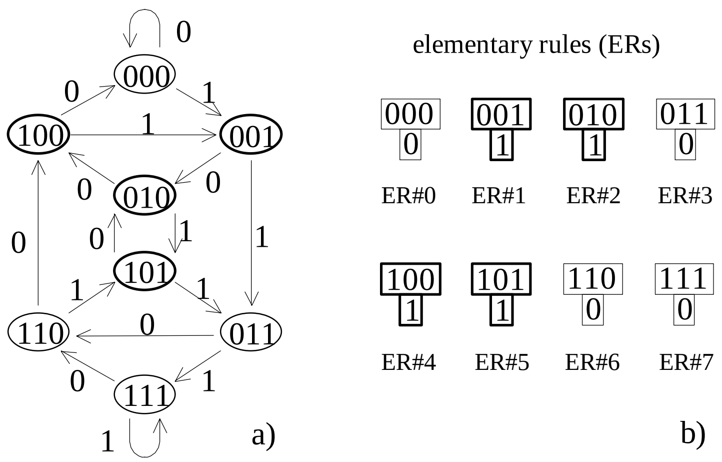

Figure 2. CA of rule 54; (a) de Bruijn graph as possible sequences of ERs, (b) the set of ERs.

We will use G_n to detect the possible strings of ERs associated with particles. By a p-periodic particle $a^\tau\_$ of an automaton CA we understand a string $a_1^\tau a_2^\tau ... a_{L_\tau}^\tau$ of symbols from A such that during the iterated CA processing the configuration $a^\tau = ...ua^\tau\_v...$ occurs in p different forms. The strings $\ldots u$ and $v\ldots$ represent regular (spatially periodic) areas. The starting states of G_n related with area $\ldots u$, and its final states with area $v\ldots$ (the roles of these sets can interchange) define the G-segments similarly to *M*-segments. However, all outputs of automaton G_n can be determined simultaneously for the entire configuration. G_n expresses the constraints on possible sequences of ERs and does not imply sequential detection of substrings of symbols or ERs involved in CA processing. Thus, for *G*-segments we use undirected graphs G_n.

Consider the CA with the rule 54 [3, 12]. Its local function $a_2' = f(a_1, a_2, a_3) = f(w_j)$ is given by $a_2' = 1 \Leftrightarrow w_j \in \{001, 010, 100, 101\}$. Below, we show three ST diagrams of CA processing. The second ST diagram is a recoded (in parallel, not sequentially) version of the first, and shows all ERs which are involved in processing. The position of each ER# is at its resulting symbol. We assumed four spatially periodic segments in the strings: (0111), (0001), (0) and (1). These are identified by the following sequences of ERs: $z = (5,3,7,6)$, $x = (4,0,1,2)$, o = (0) and | = (7), respectively. Thus we have four regular ER areas {*z*, *x*, o, |}. The boundaries between these areas are shown in the third diagram. They are recognized as the paths between starting and final sets of states of the de Bruijn graph; these sets correspond to regular areas {*z*, *x*, o, |}. These paths form *G*-segments on the undirected de Bruijn graph.

```
  1234567890123456789012345678 1234567890123456789012345678 1234567890123456789012345678

0 11.111...1...1..1...1...1... 2401253765376400137653765376 xxxx25zzzzzzz4oo1zzzzzzzzzzz
1 ..1...1.111.111111.111.111.1 7653764012401241240124012401 zzzzz64xxxxxxx41xxxxxxxxxxxx
2 .111.111...1......1...1...1. 0124012537653777765376537653 xxxxxx25zzzzz7||7zzzzzzzzzzz
3 1...1...1.111....111.111.111 5376537640124000012401240124 zzzzzzz64xxx40oo01xxxxxxxxxx
4 11.111.111...1..1...1...1... 2401240125376400137653765376 xxxxxxxx25zzz4oo1zzzzzzzzzzz
5 ..1...1...1.111111.111.111.1 7653765376401241240124012401 zzzzzzzzz64xxx41xxxxxxxxxxxx
6 .111.111.111......1...1...1. 0124012401253777765376537653 xxxxxxxxxx25z7||7zzzzzzzzzzz
7 1...1...1...1....111.111.111 5376537653764000012401240124 zzzzzzzzzzz640oo01xxxxxxxxxx
8 11.111.111.111..1...1...1... 2401240124012400137653765376 xxxxxxxxxxxxx4oo1zzzzzzzzzzz
9 ..1...1...1...1111.111.111.1 7653765376537641240124012401 zzzzzzzzzzzzz641xxxxxxxxxxxx
0 .111.111.111.1....1...1...1. 0124012401240137765376537653 xxxxxxxxxxxxx1377zzzzzzzzzzz
1 1...1...1...111..111.111.111 5376537653765240012401240124 zzzzzzzzzzzz524001xxxxxxxxxx
2 11.111.111.1...11...1...1... 2401240124013764137653765376 xxxxxxxxxxx13z6413zzzzzzzzzz
3 ..1...1...111.1..1.111.111.1 7653765376524013640124012401 zzzzzzzzzz52xx1364xxxxxxxxxx
4 .111.111.1...111111...1...1. 0124012401376524125376537653 xxxxxxxxx13zz524125zzzzzzzzz
5 1...1...111.1......1.111.111 5376537652401377776401240124 zzzzzzzz52xx137||764xxxxxxxx
6 11.111.1...111....111...1... 2401240137652400001253765376 xxxxxxx13zz5240oo0125zzzzzzz
7 ..1...111.1...1..1...1.111.1 7653765240137640013764012401 zzzzzz52xx13zz4oo1zz64xxxxxx
8 .111.1...111.111111.111...1. 0124013765240124124012537653 xxxxx13zz52xxxx41xxxx25zzzzz
9 1...111.1...1......1...1.111 5376524013765377776537640124 zzzz52xx13zzzz7||7zzzz64xxxx
0 11.1...111.111....111.111... 2401376524012400001240125376 xxx13zz52xxxx40oo01xxxx25zzz
1 ..111.1...1...1..1...1...1.1 7652401376537640013765376401 zz52xx13zzzzzz4oo1zzzzzz64xx
```

Figure 3. Three ST diagrams of rule 54 CA processing; string evolution, ERs involved in CA processing, and specific sequences of ERs (particles) that separate areas *x*, *z*, o and |.

Let us show some *G*-segments using the associated sequences of ERs. For a single particle we have: a^{τ} = ..xx**25**zzz.., $a^{\tau+1}$ = ..zzz**64**xx.., and for a complex particle: a^{τ} = ..xxx**41**xxx.., $a^{\tau+1}$ = ..zz**7||7**zz.., $a^{\tau+2}$ = ..x**40**oo**01**x.., $a^{\tau+3}$ = ..zz**4**oo**1**zz.. But other objects may be transient like ..zz**41**xx.. and ..zz**7|||**.. or irregular in some other way; e.g. they may annihilate or generate new entities.

In general, one has to determine how to filter out the particles, or which sequences of ERs are to be treated as regular areas to play the role of starting and final state sets (background) for *G*-segments. From the example it is seen that the central entity ..xxx**41**xxx.., which is known as g_e particle of the CA 54 [3, 12], should be rather considered as a complex object, since it is a kind of breather – arises from interaction between two simple particles represented by ERs ..xxx**40**ooo.. and ..ooo**01**xxx.. .

1.4 Some automata and iteron phenomena

In this section we present some filtrons phenomena and IAMs of some special automata. Let us start with multifiltron collisions.

In figure 4 we show solitonic collisions of filtrons that vibrate in a way. These are supported by automata that perform the cycles of operations. Similar cycles were applied in the PST model. Automaton M_{11} has the cycle (NNNNAAA) (N is the negate operation and A is the accept operation) and reset condition {***0000}, while automaton M_{12} has the cycle (NNNAAAA) and the same reset condition. The cyclic processing starts with the first encountered nonzero element $a_i = 1$.

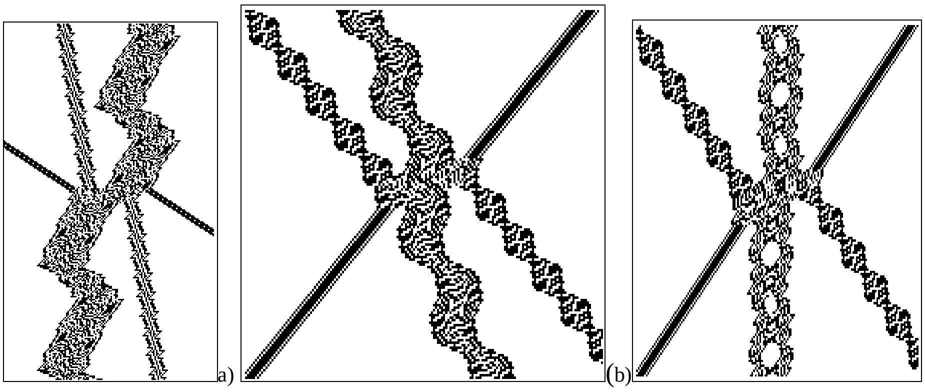

Figure 4. Colliding filtrons of automaton M_{11} (a), and of M_{12} – two ST diagrams (b); $q = 1$.

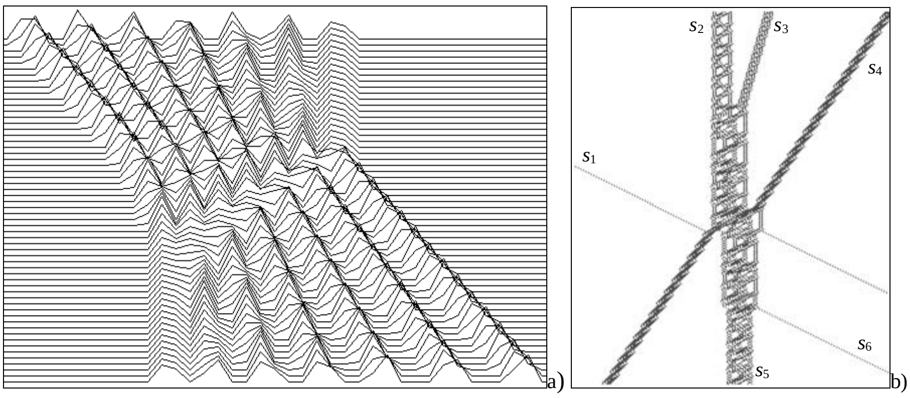

Figure 5. (a) Multi-object collision of filtrons of $M(16, 22)$; $q = 1$ [18]. (b) Quasi-filtron, $q = 2$.

In figure 5 (a) we show the multivalued filtrons over alphabet A with $|A| = 17$. These collision is supported by automaton $M(16, 22)$. $M(m, n)$ is given by: $s' = \delta(s, \sigma) = s + \min(n - s, \sigma) - \min(s, m - \sigma)$, $\omega = \beta(s, \sigma) = \sigma + \min(s, m - \sigma) - \min(n - s, \sigma)$. These automata are equivalent to a box-ball system with carrier [16, 20].

Figure 5 (b) shows fusion, fission and a quasi-coherent object. Such quasi-filtrons remain coherent for a long number of iterations, and at some moment they decay. Here, two filtrons s_2 and s_3 get into fusion into an unstable object. After 160 iterations it decays onto filtrons s_5 and s_6. In the meantime this quasi-filtron takes a part in solitonic collision with two other filtrons (s_1 and s_4) approaching from its both sides.

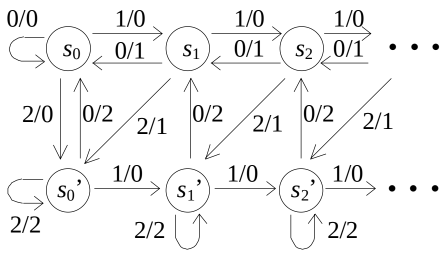

Figure 6. Automaton equivalent of a bozonic-fermionic crystal model; $A = \{0, 1, 2\}$[18].

In figure 6 we show the automaton that represents a *bozonic-fermionic* crystal [9]. Its counter memory is infinite for the symbol 1, and simultaneously is finite for the symbol 2. Filtrons supported by this model are shown in the ST diagram below.

```
0 .211---....1-.....211---....2-......111021---2-.......111---..21--2-................
1 ....211---..1-.......211---..2-........11021102--........111---.2102-...............
2 .......211---1-.........211---2-.........11--2121---........111---212--.............
3 ..........21101---.........21102--.........11--2-211---........11100221--...........
4 .............2-111---.........2121---........1102-..211---........1112021---........
5 ..............2-..111---........2-211---.......112--...211---........1210211---.....
6 ...............2-....111---......2-..211---......121--....211---......1-21--211---..
7 ................2-......111---....2-....211---....1-21--.....211---....1-.21--.211-.
```

1.5 Concluding remarks

In our approach, two issues occur. The first one is that automaton operations and converted strings interact like a medium (or field) and its disturbances. The other is that we identify the spatial extend of any coherent structure by the sequences of operations of the underlying automaton.

Automaton approach indicates at deep and relevant connections between the computational processes occurring within the nets of automata (given by IAMs) and the equations of motion of nonlinear dynamical systems or the behavior of discrete complex systems. The automaton iterating process over strings is crucial for the existence of coherent persistent structures. This is why the new term—the iterons of automata—has been proposed. The iterons seem to be as fundamental as are fractals.

The behavior of iterons is very rich and strongly depends on the underlying automaton. We have shown here only few examples. Others, like bouncing filtrons, trapped colliders, cool filtrons, repelling filtrons, annihilation, are shown in [14-18].

Potential applications of the theory of iterons cover, among others, simulating nonlinear physics phenomena, future solitonic computations [19], complex systems behavior analysis and synthesis, photonic transmission, and solitary waves prediction.

References

[1] M. J. Ablowitz, J. M. Keiser, L. A. Takhtajan. A class of stable multistate time-reversible cellular automata with rich particle content. *Physical Review* A, **44**, No. 10, (15 Nov. 1991), pp. 6909-6912.

[2] A. Adamatzky. *Computing in Nonlinear Media and Automata Collectives*. Institute of Physics Publishing, Bristol, 2001.

[3] N. Boccara, J. Nasser, M. Roger. Particle-like structures and their interactions in spatio-temporal patterns generated by one-dimensional deterministic cellular-automaton rules. *Physical Review* A, **44**, No. 2 (15 July), 1991, pp. 866-875.

[4] C. L. Barrett, C. M. Reidys. Elements of a theory of computer simulation I: sequential CA over random graphs. *Applied Mathematics and Computation*, **98**, 1999, pp. 241-259.

[5] M. Bruschi, P. M. Santini, O. Ragnisco. Integrable cellular automata. *Physics Letters* A, **169**, 1992, pp. 151-160.

[6] M. Delorme, J. Mazoyer (eds.). *Cellular automata*. A Parallel Model. Kluwer, 1999.

[7] A. S. Fokas, E. P. Papadopoulou, Y. G. Saridakis. Coherent structures in cellular automata. *Physics Letters A*, **147**, No. 7, 23 July 1990, pp. 369-379.

[8] A. H. Goldberg. Parity filter automata. *Complex Systems*, **2**, 1988, pp. 91-141.

[9] Hikami K., Inoue R. Supersymmetric extension of the integrable box-ball system. *Journal of Physics A: Mathematical and General*, **33**, No. 29 (9 June 2000), pp. 4081-4094.

[10] W. Hordijk, C. R. Shalizi, J. P. Crutchfield. Upper bound on the products of particle interactions in cellular automata. *Physica* D, **154**, 2001, pp. 240--258.

[11] Z. Jiang. An energy-conserved solitonic cellular automaton. *Journal of Physics A*: *Mathematical and General*, **25**, No.11, 1992, pp. 3369-3381.

[12] B. Martin. A group interpretation of particles generated by 1-d cellular automaton, 54 Wolfram's rule. LIP Report No. RR 1999-34, Ecole Normale Superieure de Lyon, pp. 1-24.

[13] J. K. Park, K. Steiglitz, W. P. Thurston. Soliton-like behavior in automata. *Physica* D **19**, 1986, pp. 423-432.

[14] P. Siwak. Filtrons and their associated ring computations. *International Journal of General Systems*, **27**, Nos. 1-3, 1998, pp. 181-229.

[15] P. Siwak. Iterons, fractals and computations of automata. In: D. M. Dubois (ed.), *CASYS'98* AIP Conference Proceedings **465**. Woodbury, New York, 1999, pp. 367-394.

[16] P. Siwak. Soliton-like dynamics of filtrons of cyclic automata, *Inverse Problems*, **17**, No. 4 (August 2001), pp. 897-918.

[17] P. Siwak. Anticipating the filtrons of automata by complex discrete systems analysis. In: D. M. Dubois (ed.), *CASYS'2001* AIP Conference Proceedings **627**. Melville, New York, 2002, pp. 206-217.

[18] P. Siwak. Iterons of automata. In: *Collision-based computing*. A. Adamatzky (ed.), Springer-Verlag, London. 2002, pp. 299-354.

[19] K. Steiglitz. Time-gated Manakov spatial solitons are computationally universal. *Physical Review* A, **63**, 2000.

[20] D. Takahashi, J. Matsukidaira. Box and ball system with carrier and ultradiscrete modified KdV equation. *Journal of Physics A*: *Mathematical and General*, **30**, No.21, 1997, pp. L733-L739.

[21] D. Takahashi, J. Satsuma. A soliton cellular automaton. *Journal of the Physical Society of Japan*, **59**, No. 10, October 1990, pp. 3514-3519.

[22] T. Tokihiro, A. Nagai, J. Satsuma. Proof of solitonical nature of box and ball systems by *means* of inverse ultra-discretization. *Inverse Problems*, **15**, 1999, pp. 1639-1662.