Preprint. This paper has been accepted for publication in: **International Journal of General Systems**.
Special Issue on General Physical Systems and the Emergence of Physical Structure from Information Theory. Summer/Autumn 1996.

Filtrons and Their Associated Ring Computations

Pawel Siwak (e-mail: psiw@agat.sk-kari.poz.edu.pl)
Poznan University of Technology, 60-965 Poznan, Poland

Certain special computational models called the filter automata (FA) are presented. We use them in iterative, pipelined processing of discrete sequences. Such computations reveal characteristic moving, periodic objects, with soliton-like properties referred to as the filtrons. Filter automata play the role of a transmitting medium or a field, while filtrons behave like waves or particles.

We characterise FA by showing their distinguishing features, and filtrons - by introducing some parameters, such as edge numbers, number of FA cycles involved, distribution of internal idle pulses, etc. We show that filtrons may be generated in parallel by a multiprocessor ring net. This means that each filtron has its associated ring computation. If such a computation succeeds then its companion filtron exists, too. The link between filtrons and their ring computations is established simply by the filtron edge numbers. This gives a method of calculating a whole filtron if its edge number and supporting FA are known, and of looking for possible filter automata capable of supporting a prescribed filtron.

A number of FA computational phenomena, recently discovered, are also presented. Some of them call for physical experiments to verify to what extent the filter automata may model the physical reality. We believe that such experiments could be performed in optical fibres using the solitons of light.

Keywords: solitons, filter automata, filtrons, breathers, cellular automata, Mealy arrays, iterative processing, discrete particles, multiprocessor computations

1. Introduction

The paper presents the idea of filtrons and their connections with special ring computations. Both of them appear in a computational model called filter automata (FA) [Siwak, 1992], [Siwak, 1993], [Siwak, 1994], [Siwak, 1995], and [Puchalka and Siwak, 1995]. The term parity rule filter automaton (PRFA) was used for the first time in 1986 by K. Steiglitz and his co-workers in [Park, *et. al.*, 1986] to denote one-dimensional, binary, parity-rule recursive filtering (PRR filtering). This model was defined by a formula executed sequentially. In the domain of signal processing such operators are classified as Infinite Impulse Response (IIR) digital filters [Fitch, *et. al.*, 1984]. Steiglitz showed that his PRFA generate moving, periodic substrings or patterns. He called these structures particles. They were exciting because, travelling with different speeds, they were able to interact, exhibiting soliton-like properties. Soon [Steiglitz, *et. al.*, 1988], it was shown that the interacting particle substrings may perform meaningful computations.

Here, we use the term filter automaton in a wider, new meaning. It is a formal computational model operating over an arbitrary, finite alphabet X. This model makes use of activating and extinction conditions, as well as a processing program. Our definition of filter automata allows to analyse and explain how the transmitting medium and propagating signals interact, supporting the soliton-like phenomena. The generalised filter automata not only cover a wider family of 'particles' but also reveal many new objects and their surprising properties. Some FA objects are presented further, and it will become clear that a new term is necessary to identify them and to relate them to our filter automata model. The term 'particle' is associated rather with the objects of nuclear physics. The term 'discrete solitons' could suggest a numerical approximation of the objects from nonlinear physics. So our proposition to introduce and use the new term 'filtrons' seems to be justified.

The filter automaton F is a sequential transducer. We use it for iterative processing of sequences of symbols. This computation may be equivalent to a pipeline of copies of F, or to a one-way linear cellular automaton [Ibarra, *et. al.*, 1992]. The sites in our one-way cellular automata follow Mealy model properties, in contrast to Moore type machines typically employed as cells [Albert and Culik II, 1987], [Lindgren and Nordal, 1990], [Martin, 1992], [Smith, 1992]. Such an approach is crucial to support the moving periodic substrings referred to as the filtrons.

The filtrons are visible on the space-time (ST) diagrams of iterative computations of FA. They seem to mimic the phenomenon of solitary waves or solitons. On the ST diagrams of the automata one can observe a variety of solitonic behaviour: simple filtrons, periodic or higher order filtrons, solitonic collisions with non-demolition interaction and mutual attraction, and even jumps over the space. We also observed annihilating filtrons, fusion of filtrons, decay of filtrons and other fascinating phenomena. So far as I know, not all of these have been observed in physical experiments. All these computational phenomena may call for an experimental verification.

The primary feature of FA is that they describe the processing by a triple of simple components: an activation function, operations and extinction conditions. No filtering formulae are in use, no wave equations, not even states. The lines of such FA are like a physical medium and strings of symbols are like disturbances. So, the computation mimics a field or interaction between a signal and transmitting medium. Their second feature is the ability to support a huge variety of filtrons. Many of them remind one of soliton-like phenomena and the particles of nuclear physics.

There are strong motives to promote filter automata. It is my belief that FA is a research idea offering a significant technological and theoretical impact. However, there is still a fundamental question as to what extent the model of filter automata is capable of describing physical reality. Let us mention some arguments supporting this belief.

The properties of FA filtrons justify using them as perfect models of discrete solitary waves. It is well known [Drazin and Johnson, 1989], [Majewski, 1993], [Newell, 1985], [Taylor, 1992] that solitons represent waves (e.g. pulses of light) with extraordinary properties. As they propagate, the solitons are extremely stable, and particularly resistant to disruptions. Propagation distances of more than one million kilometres have been demonstrated for light pulses in optical fibres. Solitons only exist in nonlinear media. It is quite easy to generate short solitary pulses. For optical solitons times below 10 fs have been achieved, which equals only a couple of wave lengths of light (1.55 μm is the 3rd window). What is even more remarkable is the interaction of solitons. They collide without losing their identity, changing their shape, intensity or energy. They change only their orbital and translational phases. When they approach, forces act and solitons attract one another. The associated information is not destroyed, instead it is transformed. In short, solitons behave more like particles than waves. All these properties are very attractive in the field of telecommunication.

Two quite new possibilities have appeared. First, the data rate of information transmission can be extended dramatically; note that a 10 fs pulse implies 100 TeraHz frequency, which is about 5 orders better than that in conventional lines. Secondly, information could be processed during the transmission of colliding streams of solitons, which means that optical fibre lines could play the role of an extended solitonic processor.

These two immediate technical applications demonstrate that research of models of solitons is relevant and important. We believe that there are also many other connections of FA theory with quantum discrete models, dynamics of nonlinear systems, algebraic equations, signal processing, parallel computation and software engineering. Some of these connections will be mentioned.

2. Cellular processing

In this paper we consider computations iterated over an initial string by a single machine of finite state type. It is widely known that Moore and Mealy automata are equivalent when treated as transducers. However, in computational nets they exhibit completely different behaviour [Buning and Priese, 1980]. The iterative computations may be performed by appropriate iterative or cellular nets. We want to show that the proper net's node may facilitate the analysis of computational dynamics of such nets. Particularly, so called solitonic collisions may occur only in cellular processing performed by nets with signals separated from states. It is easier to use Mealy type machines for this purpose.

Let us begin with fundamental models [Mikolajczak, 1991].

A Mealy type automaton M with outputs and an initial state is defined to be an algebraic system $M = (S, X, Y, \delta, \beta, s_0)$, where: S is a non-empty, finite set of states, X is a non-empty, finite input alphabet, Y is a non-empty, finite output alphabet, δ is a next state (or transition) function $\delta : S \times X \rightarrow S$, β is an output function $\beta : S \times X \rightarrow Y$, and s_0 is an initial state, $s_0 \in S$.

The operation of an automaton M is illustrated in Fig. 1. For each letter x_i read from an input string it responds with an associated output letter y_i. The input string is read from left to right, one symbol at each instant τ of time, in such a way that:

$$\delta(s(\tau), x(\tau)) = s(\tau+1) \text{ and } \beta(s(\tau), x(\tau)) = y(\tau).$$

When the conversion of the whole string is finished we say that a single step of processing has been completed. The steps are indexed by t.

A Moore type machine is defined as an algebraic system $M = (S, X, Y, \delta, \lambda, s_0)$, which differs from the above only in output function $\lambda : S \rightarrow Y$ or $\lambda(s(\tau)) = y(\tau)$.

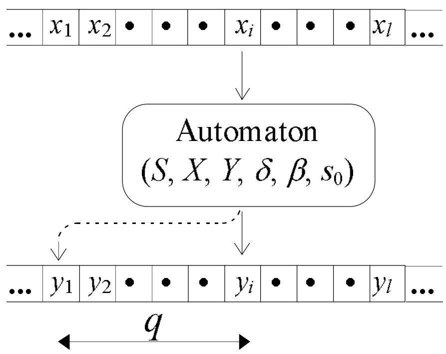

Fig. 1. Automaton M "running" to the right; q is a range of processing.

2.1. Iterative processing

We consider here iterative computations, assuming $X = Y$. This implies that an input string x^0 is given at time $t = 0$. After each iteration over x^t the next string (or configuration) x^{t+1} is computed over the succession of steps $t = 1, 2, \ldots$. It is easy to arrange such a computation using a pipeline-like structure of identical copies of an automaton F. This is shown in Fig. 2. Such linear structure of automata is known as the OLIA model [Ibarra, *et. al.*, 1992]. This is a linear cellular automaton with only one (left-hand) neighbour [Martin, 1992], [Garzon, 1995].

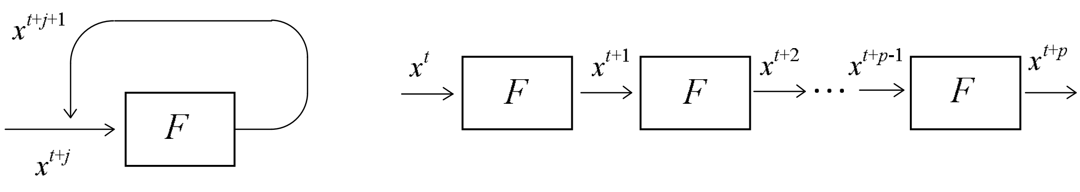

Fig. 2. Iterative processing of a discrete string x^t and an equivalent processing line (OLIA).

In general a cellular automaton is understood to be a lattice, each site or 'cell' of which operates like a copy of a single finite-state automaton. Cells interact according to a local function changing their states synchronously. If they are Moore type machines, there is a delay between any input and its resulting output. This is because in the Moore model the output can change only if the state has been changed. Such cellular automata (CA) always exhibit stable images of signals (configurations of symbols).

Most of the cellular automata, considered in literature, use Moore type cells [Smith, 1992]. Moreover, it is usually assumed that some working alphabet Σ (called a set of site values) represents states, as well as inputs and outputs of the cells; $\Sigma = S = X = Y$. In this Moore type automaton without outputs, the states and input/output signals are not distinguished. The output coding $\lambda : S \rightarrow Y$ is the identity function. This is an assumption with relevant consequences, nevertheless very rarely explicitly stated. It must be emphasised that in such cellular automata the real phenomenon of solitonic interactions can not occur, simply because the fixed neighbourhood's size limits the distance beyond which the sites cannot influence one another.

This is in contrast to Mealy type CA which allow instantaneous (combinational) output from an input at each cell. Note, that for each Mealy type cell we can define a gate $f_s : X \rightarrow Y$ such that $\beta(s, x) = f_s(x) = y$ for any state $s \in S$. This means that the whole processing line of Mealy machines, such as shown in Fig. 2, operates like a single resulting gate, thanks to which the signals may propagate throughout such CA immediately. The solitonic interactions can then be supported. However, there are some well known problems with such nets. For example in nets with loops certain unstable cycles of signals [Buning and Priese, 1980] may occur. This is the main reason why this model was rather difficult to analyse and considered to be not very useful in most applications. This is also why the so called systolic nets and cellular automata are usually 'equipped' with Moore type cells.

Nevertheless, we have found simple constraints that allow use of Mealy type pipelines to perform computations. What is most important, such nets mimic the interplay between a field represented by all machines and particle-like objects represented by the non-zero substrings of symbols passing throughout the net. Also, this approach offers a simple analytical framework to explain the phenomenon of solitary waves and reveals new connections between computations and the behaviour of dynamical systems.

There are various speculations about the direction of the development of future computer architectures. With our model we advocate the idea of arranging computations by means of pipelines of filter automata which are Mealy-like machines. It is very probable that optical fibre will be able to play not only its traditional role of transmitting medium but simultaneously the role of processing line capable of performing the parallel computations.

Let us recall [Albert and Culik II, 1987], [Lindgren and Nordal, 1990], [Martin, 1992] that computations done with the help of the Turing machine can be easily converted to the cellular automaton with Moore sites and the nearest neighbourhood. A cellular automaton is a triple $CA = (C, \nu, \varphi)$ where $C = \{c_i\}$ is a set of cells, $\nu : C \rightarrow C^n$ is a neighbourhood function and $\varphi : S^n \rightarrow S$ acts as the local function of CA. Each cell is a Moore type automaton $c = (S, X, \varphi)$ without outputs, where $\varphi : S \times X \rightarrow S$.

It is assumed that $S \times X = S^n$ where an integer $n > 1$ determines a scope or index of neighbourhood of CA. In d-dimensional cellular automata the cells $c_i \in C$ are ordered according to their positions $i \in \boldsymbol{Z}^d$. In the 1-dimensional case the nearest neighbours are usually assumed, which means that $n = 3$ and $\nu(c_i) = (c_{i-1}, c_i, c_{i+1})$ for each cell c_i.

A configuration of CA is defined to be the function $f : C \rightarrow S$. S^C denotes the set of all configurations. Then, $f(\nu(c)) \in S \times X$ is the state of the neighbourhood of a given cell c (including itself). A global transition function of CA is such a function $\gamma : S^C \rightarrow S^C$ that $\gamma(f) =$

g implies $\gamma(f(c_i)) = \varphi(f(\nu(c_i))) = g(c_i)$ for all $i \in \mathbf{Z}$. A quiescent state $s_0 \in S$ of CA has such a property that $\varphi(s_0, s_0, s_0) = s_0$.

The conversion from any universal Turing machine to linear cellular automata was proposed for the first time by A. R. Smith (see [1992]) and is used in [Lindgren and Nordal, 1990]. Linear cellular automata may simulate any universal Turing machine $(S, T, \delta, \beta, \rho)$, where $\rho: S \times T \rightarrow \{\mathrm{L, R, H}\}$ determines the moves of the machine's head, and T is a set of tape symbols ($T = X = Y$). The local function of an appropriate cellular automaton is given by:

$\ldots x_0x_1\mathbf{s}{\rightarrow}\mathbf{x_2}x_3\ldots$	$\ldots x_0x_1\mathbf{s}{\rightarrow}\mathbf{x_2}x_3\ldots$
$\ldots x_0x_1\mathbf{y_2}\mathbf{t}{\rightarrow}x_3\ldots$	$\ldots x_0x_1{\leftarrow}\mathbf{t}\mathbf{y_2}x_3\ldots$
$\ldots x_0\mathbf{x_1}{\leftarrow}\mathbf{s}x_2x_3\ldots$	$\ldots x_0\mathbf{x_1}{\leftarrow}\mathbf{s}x_2x_3\ldots$
$\ldots x_0{\leftarrow}\mathbf{t}\mathbf{y_1}x_2x_3\ldots$	$\ldots x_0\mathbf{y_1}\mathbf{t}{\rightarrow}x_2x_3\ldots$

The set of states of CA is composed of tape symbols T, and $\{(\leftarrow s), (s\rightarrow)\}$ - all combinations of L/R moves and states $s \in S$ of the Turing machine.

Note also the simple conversion from linear cellular model to a pipeline of Mealy machines. If $\varphi(x_1, x_2, x_3) = x_2'$ holds for $CA = (C, \nu, \varphi)$, where $c = (X, X^2, \varphi)$, then a Mealy type machine $M = (S, X, X, \delta, \beta)$ can be defined as:

$$S = X \times X, \quad \delta((x_1, x_2), x_3) = (x_2, x_3) \ \text{ and } \ \beta((x_1, x_2), x_3) = \varphi(x_1, x_2, x_3) = x_2'.$$

The following relevant conclusion emerges from the above:

- the theory offers universal computational linear structures with Mealy-like nodes; they are pipelined, simple, homogeneous and operate in parallel,
- they are among the first candidates to play the role of future multiprocessing computational structures.

We shall use space-time (ST) diagrams (i, t) to present the results of iterative computations performed with filter automata. An initial configuration x^t will be placed in its first line, and the subsequent lines will correspond to the subsequent configurations x^{t+1}, x^{t+2}, ... The index i increases to the right, and denotes the position of a symbol x_i. The index t denotes iterations or steps of time. It is shown in Fig. 3. Note, that these ST diagrams show the input/output symbols instead of states of underlying model, which is usual in cellular automata.

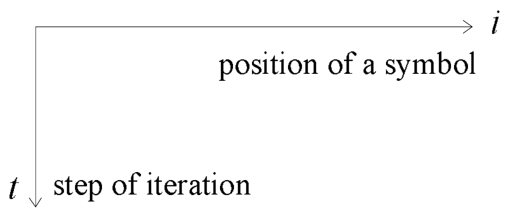

Fig. 3. Space-time diagram (i, t) of iterative processing.

Sometimes it is convenient to shift each subsequent configuration to the left by the constant value q. This parameter will be called the range of a processing automaton. The range is shown in Fig. 1.

2.2. Recursive filtering

The phenomenon of a particle substring and its solitonic collision was reported for the first time in [Park, *et. al.*, 1986], in the context of one-dimensional, binary, parity-rule recursive filtering. Such transducers belong to recursive filters and are classified as IIR filters [Fitch, *et. al.*, 1984]. Later, in [Steiglitz, *et. al.*, 1988], the particle substrings of PRR filtering were even shown to be able to perform meaningful computations. Then, the Fast Rule Theorem - an equivalent expression of PRR filtering formula was introduced [Papatheodorou, *et. al.*, 1988], [Papatheodorou and Tsantanis, 1989], and some properties of particle substrings were analysed [Fokas, *et. al*., 1989a], [Fokas, *et. al*., 1989b], [Papatheodorou and Fokas, 1989], [Fokas, *et. al*., 1990a]. Recently, certain generalisations of PRR filtering formula were added. In [Ablowitz, *et. al*., 1991] a non-binary alphabet X was included and an enlargement of the supported particle substrings was achieved. An elegant extension of the PRR filtering model towards group structures was presented in the paper [Fokas, *et. al*., 1990b]. We will refer to their model here as FPS filtering. The new idea of adding a memory variable to the generalised PRR filtering formula was also introduced in FPS filtering. Moreover, in the same paper two dimensional space was considered for the first time.

To show precisely the ideas which stand behind filter automata we first recall the two models of recursive filtering mentioned above. The PRR filtering is determined by an update function Φ and by r - the radius of its window. These are defined as follows [Park, *et. al*., 1986].

Let x^t denote a sequence $...0\, x_1^t \dots x_i^t \dots x_l^t\, 0...$ of elements from X, given at a moment t, where: $0 \le t < \infty$, $-\infty < i < \infty$, $l < \infty$, and $x_1^t, x_l^t \neq 0$. According to the parity rule Φ the next configuration x^{t+1} is calculated sequentially from left to right by:

$$x_i^{t+1} = \Phi\,(x_{i-r}^{t+1},\ x_{i-r+1}^{t+1}, \dots, x_{i-1}^{t+1},\ x_i^t,\ x_{i+1}^t, \dots, x_{i+r-1}^t,\ x_{i+r}^t),$$

where the parity rule Φ is a totalistic [Albert and Culik II, 1987] function $X^{2r+1} \to X$ such that

$$x_i^{t+1} = \begin{cases} 1 & \text{if } S_i \text{ is even but not zero,} \\ 0 & \text{otherwise,} \end{cases}$$

where $S_i = \sum_{j=-r}^{-1} x_{i+j}^{t+1} + \sum_{j=0}^{r} x_{i+j}^t$.

This model was called originally a parity rule filter automaton (PRFA). It has been pointed out that it is equivalent to processing sequences by the window shown in Fig. 4.

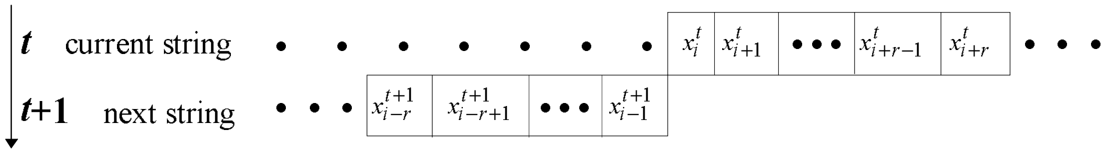

Fig. 4. Window of PRR filtering. Sliding to the right it operates as a sequential transducer.

FPS filtering is defined as follows [Fokas, *et. al*., 1990b]. Let $X = \{0, x_1, \dots, x_n\}$ be a finite group with its associated operation denoted by $\otimes$. Let 0 be the identity element, so $0 \otimes x = x \otimes 0 = x$. Let $\overline{x}$ be the inverse of x, i.e. $\overline{x} \otimes x = x \otimes \overline{x} = 0$. For X any finite group let us take a sequence x^t at the moment t : $...0\, x_1^t \dots x_i^t \dots x_l^t\, 0...$, where $l < \infty$ and $x_1^t \neq 0$, $x_l^t \neq 0$. Then a sequence at the moment t+1 is determined by:

$$x_i^{t+1} = \begin{cases} 0, \text{ if all factors } x_i \text{ of } S_i \text{ are } 0, \text{ or} \\ \overline{S_i \otimes z \otimes \overline{x}_{i+r}^t}, \text{ otherwise} \end{cases}$$

where:

$$S_i = \sum_{j=i}^{i+r-1}\!\otimes\, (x_{j-r}^{t+1} \otimes \overline{x}_j^t) \ \text{ and } \ z := \begin{cases} x_{i+r}^t, \text{ if all factors } x_i \text{ of } S_i \text{ are } 0, \text{ or} \\ z, \text{ otherwise} \end{cases},$$

and it is assumed that for $i \to -\infty$ all factors x_i of S_i are all zero. In the case of the Abelian group (Z_n, $\oplus$), where $Z_n = \{0, 1, ..., n\text{-}1\}$ and the symbol $\oplus$ denotes modulus n addition, the above definition reduces to the classical PRR filtering.

It is seen that any finite group is allowed instead of the binary alphabet X. Moreover, an auxiliary memory variable denoted by z is in use. The idea of FPS filtering was to generalise PRR filtering and retain (or even strengthen) its capability of supporting soliton-like particle substrings.

3. Definition of filter automata

Recursive filtering performs a sequential processing of strings, and thus can be easily converted to a finite-state machine. An analysis of Mealy automata equivalent to PRR filtering and experiments with machines that support particle substrings lead us to further generalisation of the previous models. The definition is as follows [Puchalka and Siwak, 1995].

The filter automaton is a sequential transducer $F = (X, \iota, \mu, \kappa)$, where:

$X = \{0, 1, ..., m\}$ is a finite set, called the alphabet of F, with $m > 0$,

$\iota : X \to \{0, 1\}$ is a mapping, called the activating function of F,

$\mu : X^{\leq r+1} \to X$ is a partial mapping (the control function of F, $r > 0$),

$\kappa = (h_0, h_1, ..., h_m, f_1, ..., f_r)$ is a sequence of mappings $X \to X$, called the program of F, which generates outputs in discrete time steps τ according to the following, cyclic order:

$$-\operatorname{next}(h_0(x_\tau)) = \begin{cases} h_0(x_{\tau+1}) & \text{if } \iota(x_\tau) = 0 \\ f_1(x_{\tau+1}) & \text{if } \iota(x_\tau) \neq 0 \end{cases}, \qquad \text{and } z := x_\tau$$

$$-\operatorname{next}(h_i(x_\tau)) = \begin{cases} h_j(x_{\tau+1}) & \text{if } \mu(x_\tau) \text{ is defined} \\ f_1(x_{\tau+1}) & \text{otherwise} \end{cases}, \text{ where } j = \mu(x_\tau) \text{ and } 0 < i \leq m,$$

$$-\operatorname{next}(f_i(x_\tau)) = \begin{cases} h_j(x_{\tau+1}) & \text{if } \mu(x_{i\div\tau}) \text{ is defined} \\ f_{i+1}(x_{\tau+1}) & \text{otherwise} \end{cases}, \text{ where } j = \mu(x_{i\div\tau}) \text{ and } 0 < i < r,$$

$$-\operatorname{next}(f_r(x_\tau)) = \begin{cases} h_j(x_{\tau+1}) & \text{if } \mu(x_{r\div\tau}) \text{ is defined} \\ h_z(x_{\tau+1}) & \text{otherwise} \end{cases}, \text{ where } j = \mu(x_{r\div\tau}).$$

In the above $x_{i\div\tau} = (x_{\tau-i}, x_{\tau-i+1}, ..., x_\tau)$ denotes a $(i+1)$-segment which coincides with a current cycle of automaton's activity, and $X^{\leq r+1}$ is used instead of $X \cup X^2 \cup ... \cup X^{r+1}$. The operation of the filter automata is illustrated in Fig. 5.

The filter automaton operates in two modes, it is either inactive or active. The inactive filter automaton waits for the occurrence of an activating element (such as $\iota(x) = 1$) while the read symbols are output as $h_0(x)$. We assume $0 \in X$ to be the quiescent symbol, which means that $h_0(0) = 0$.

When the filter automaton is activated, successive operations of its program κ are executed; we say that the automaton performs its cycle of activity. The possible cycles are: $(h_i, f_1, ..., f_l)$, where $l \leq r$. Each cycle, followed by the mapping h_0, is said to be an extinction cycle and each cycle, followed by h_i for $i \neq 0$, is said to be an upholding cycle. We say that the filter automaton performs an idle step at the moment τ if it performs an h_0 operation at the same time.

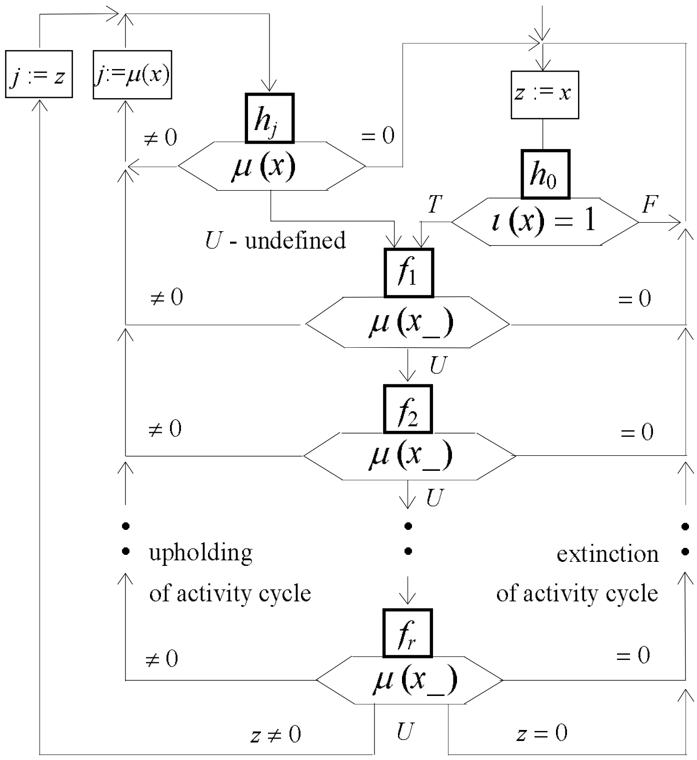

Fig. 5. A flow chart of processing performed by $F = (X, \iota, \mu, \kappa)$,
$x\_ \in X^{\leq r+1}$ is the current segment $x_{i \div \tau}$ of a string, which coincides with the activity cycle of F.

Any l-tuple $x\_ \in X^l$, $0 < l \leq r$, such that $\mu(x\_) = 0$ will be called the reset condition of a filter automaton or an extinguishing segment. An activated filter automaton remains active until some reset condition coincides with a cycle of its activity.

From the above, it is seen that filter automata constitute a new, well defined, special class of sequential transducers. They feature the following properties.

1. The states are eliminated; instead two modes of operation are in use: a filter automaton is either inactive (extinguished or idle) or active.
2. Three notions describe how a filter automaton works:
   - activation (conditions which force its activity),
   - processing (operations which convert read symbols), and
   - extinction (reset conditions which cause the automaton to be idle).
3. The activity of filter automata is a cyclic process; a sequence of operations (program κ) is performed one at each instant τ of time. The cycle of operations can be shortened, and run again or extinguished. A number of distinct cycles is possible.
4. The responses of filter automata are available by the same principle as in a Mealy machine M from Fig. 1. At each moment τ - when the state s of the machine is fixed - each output symbol y_τ depends on the input letter x_τ which is read by the machine at this moment; so the responses are determined by the operation $f_s : X \rightarrow Y$.
5. The inactive mode is stable, which means that there is an input symbol $0 \in X$ (the so called quiescent or zero symbol) which rests a filter automaton in inactive mode until an activating symbol is read. The output returned by the inactive automaton is also assumed to be quiescent.

The model of filter automata is a generalisation of transducers based on recursive filtering formulae in the following sense:

- any finite alphabet $|X| = m+1$ can be used, not only the binary one $\{0,1\}$, or a group type,
- any l-tuple is allowed as the reset condition ($0 < l \leq r+1$), not only strings of the form $*0^r$ which is the case in PRR filtering and in FPS filtering, and
- any (r+1)-tuple of mappings $X \rightarrow X$ may constitute the cycle of automaton's activity. Note that in PRR filtering only two types of (r+1)-cycles are employed as the parity update rule: either $\mathbf{0}A^r$ or NA^r, where A denotes the accept operation (*id.*), and N the negate operation,
- the cycles of automaton's activity may be shortened.

Moreover, any range q can be used to visualise computations on ST diagrams since it is now independent of the processing automaton; note that in PRR filtering it was actually $q = r$.

The idea of filter automata resolves a number of problems. In this model no filters with their analytical formulae are used any longer, neither are typical automata with their states, nor cellular automata with Moore type cells. Instead, in a filter automaton $F = (X, \iota, \mu, \kappa)$, the processing is governed by an activation function ι, operations κ, and a control function μ.

The consequences of such an approach are very relevant. The pipelined nets of filter automata are very easily tuned as well as interpreted as a sort of 'medium'. Note, that they are similar to Mealy nets, so - if one is interested in propagation of signals - many 'strange' phenomena may be depicted: there are plenty of periodic and moving objects. Sometimes signals disappear for some time, sometimes signals jump over fragments of space (if these are represented by activated automata). In this way the essence of FA computations has become visible and may be exploited; FA simulate mutual interaction between an environment and its disturbance. The following duality has become clear: there are no particle-like objects without their associated processing automata, and only the automata may serve as classifiers of such objects. In our space of filter automata the operations and symbols interact.
The following example illustrates how the filter automata operate.

Example 1. Let us consider the filter automaton $F_1 = (X, \iota, \mu, \kappa)$. Let $X = \{0,1\}$, $\iota(0) = 0$ and $\iota(1) = 1$. The function μ is represented by the two reset conditions: $\mu^{-1}(0) = \{*00\} = \{000, 100\}$. The program $\kappa = (h_0, h_1, f_1, f_2)$ is such that $h_0 = \mathbf{0}$, $h_1 = N$ (negate), $f_1 = f_2 = A$ (accept). In Table 1 we show processing of a binary sequence. The chosen filter automaton F is equivalent to PRR filtering with $r = 2$.

Table 1. Processing of a binary sequence by the filter automaton F_1.

discrete time pulses τ	…	1	**2**	3	4	5	6	7	8	9	10	**11**	12	13	14	15	**16**	17	18	19	20	21	22	…
input sequence x^t	…	0	1	0	1	0	1	1	1	0	0	1	0	0	0	0	1	1	1	0	0	0	0	…
operations performed by F_1	…	h_0	h_0	f_1	f_2	h_1	f_1	f_2	h_1	f_1	f_2	h_0	f_1	f_2	h_0	h_0	h_0	f_1	f_2	h_1	f_1	f_2	h_0	…
pulse type (a - active, i - idle)	…	i	i	a	a	a	a	a	a	a	a	i	a	a	i	i	i	a	a	a	a	a	i	…
automaton's activity cycle #	…			←1→			←2→			←3→			←4→					←5→			←6→			…
output sequence x^{t+1}	…	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	…

The symbols should be read from left to right in all sequences. Six cycles of activity are forced on the automaton F_1, however it is activated three times (at $\tau = 2$, 11 and 16). The cycles #1 and #2 are upholding cycles. There are three extinction cycles: #3, #4 and #6 (associated extinguishing segments are printed on the grey background). Note also that cycles #3 and #4 are adjacent, however they are separated by one idle pulse. The cycles #4 and #5 are separated by three such pulses. We shall say that there are two series of internal idle pulses (IIPs).

The ST diagram below shows 23 steps of processing of the same initial configuration. Note a solitonic collision of substrings 11 and 111; they interact only in steps 18 and 19.

```
t=00:  ...01010111001000011100000000000000000000000000000...
t=01:  ...00011110000000001110000000000000000000000000000...
t=02:  ...00001100000000000111000000000000000000000000000...
t=03:  ...00000101000000000011100000000000000000000000000...
t=04:  ...00000001100000000001110000000000000000000000000...
t=05:  ...00000000101000000000111000000000000000000000000...
t=06:  ...00000000001100000000011100000000000000000000000...
t=07:  ...00000000000101000000001110000000000000000000000...
                              .   .   .
t=16:  ...00000000000000000000000001100001110000000000000...
t=17:  ...00000000000000000000000000101000111000000000000...
t=18:  ...00000000000000000000000000001100011100000000000...
t=19:  ...00000000000000000000000000000101010101000000000...
t=20:  ...00000000000000000000000000000001110001100000000...
t=21:  ...00000000000000000000000000000000111000101000000...
t=22:  ...00000000000000000000000000000000011100001100000...
t=23:  ...00000000000000000000000000000000001110000101000...
```

4. Filtrons

ST diagrams of the filter automaton model exhibit a highly nontrivial and surprisingly rich set of particular substrings. We are especially interested in the periodic and quasi-periodic ones.

4.1. Definition and parameters of a filtron

Any finite sequence x^t over X, starting and ending with a non-zero element will be called a string. Any string x^t (when the configuration $..0x^t0..$ is passed iteratively through a filter automaton) such that $x_{i+d}^{t+p} = x_i^t$ holds for some integer p, for all $-\infty < i < \infty$, and all $t \geq 0$, is said to be a periodic string. The smallest integer p with the above property is called the period of a string. The integer d is called the displacement of a periodic string. The ratio d/p was defined [Park, *et. al.*, 1986] to be the speed of a moving, periodic string.

There is a number of terms like discrete solitary waves, particles, solitons, basic strings, breathers, etc., used to describe various periodic strings appearing on ST diagrams. Some objects, such as the root signal, were already known in the theory of signal processing [Fitch, *et. al.*, 1984] and in cellular automata. Using the concept of filter automata we are able to define and classify uniformly all these objects. As it was mentioned before they will be called the filtrons.

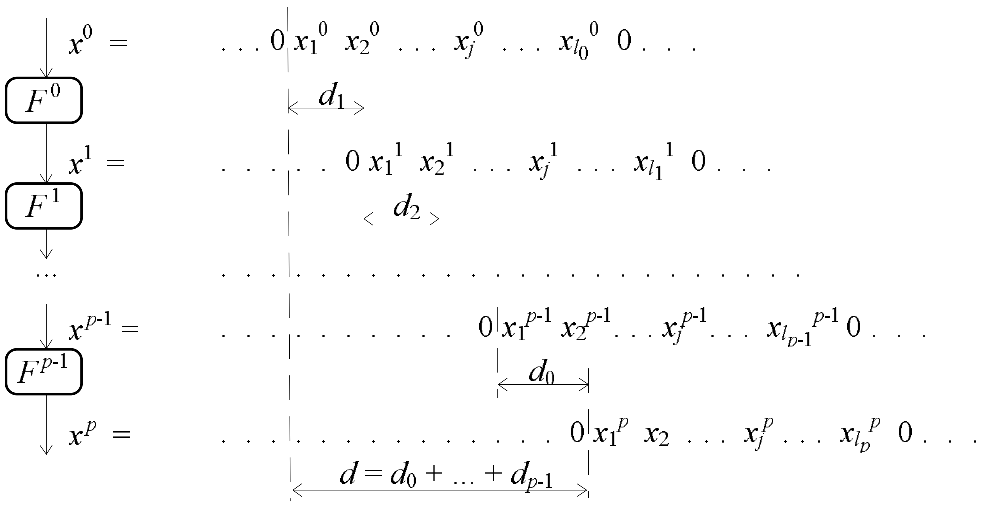

Fig. 6. Pipeline of automata and the filtron situated on ST diagram.

We define the p-periodic connected filtron x^t of a filter automaton F to be a p-periodic string, such that for each $t = 0, 1, ..., p$-1 the automaton F is activated by x^t but only once. When there are multiple activations we will use the general term 'complex filtron'. This definition is equivalent to that given in [Steiglitz, *et. al.*, 1988]. Another approach using the concept of splitting has been proposed in a series of papers by Fokas and his co-workers [Fokas, *et. al.*, 1989b], [Papatheodorou and Fokas, 1989], [Fokas, *et. al.*, 1990a], [Fokas, 1990].

After Steiglitz [1988] we shall refer to the succession of strings passed through by a filtron as the set of its orbital states; the sequence $(x^0, x^1, ..., x^{p\text{-}1})$ is called the orbit of the filtron. In Fig. 6 we show the ST diagram of a filtron and its relation to the pipeline of filter automata.

When a configuration ..0x^t0.. is read by a filter automaton F it may happen that F is activated only once. We say in this case that the string x^t is connected. But it may happen as well that F is activated a number of times. Then, after each extinction cycle (besides the last one) there is a series of intervals during which the automaton F remains inactive, but after which it is activated again. These intervals are called the internal idle pulses (IIPs). If there are h_t such series then the given configuration consists of 1+h_t connected components. The number of IIPs in each of h_t series will be denoted by $nip_t(i)$, for $i = 1, 2, ..., h_t$.

t:	••• $\mathbf{0}x_1^t$**xxx** ••• $e^t(1)$**xxx** ••• $e^t(2)$**xxx** • • • • $e^t(h_{t\text{-}1})$**xxx** ••• $e^t(h_t)$**xxx**x_l^t**0** •••
IIPs:	••• **0** —— $nip_t(1)$ —— $nip_t(2)$ —— • • • • —— $nip_t(h_t)$ —— **0** •••

Fig. 7. Series of internal idle pulses within an orbital state x^t.

For the periodic strings we may define $h = max\{h_t\}$, and for each $i = 1, 2, ..., h_t$ the sequence of the form $vip(i) = (nip_{p\text{-}1}(i), nip_{p\text{-}2}(i), ..., nip_0(i))$. We will call it a vector of i-th series of IIPs. IIPs allow us to distinguish multiple objects occurring in complex periodic strings. Note also that the separation of two strings by at least one IIP indicates that they do not interact with each other at a given moment.

Any specific complex filtron of a given filter automaton F can be described by its left edge symbols $\varepsilon = (x_1^{p\text{-}1}, x_1^{p\text{-}2}, ..., x_1^0)$, and a distribution of internal "holes" (idle pulses). Moreover, the inside (left) edges of symbols must be given too. The notions of series of internal pulses and of internal edge symbols $e^t(1), e^t(2), ..., e^t(h_{t\text{-}1})$ are illustrated in Fig. 7.

To complete the description of the filtron we add a sequence $\Delta = (d_{p\text{-}1}, d_{p\text{-}2}, ..., d_0)$ of its row displacements. The row displacement d_t of orbital state x^t of a given filtron is associated with a passage of its predecessor $x^{t\text{-}1}$ through the associated filter automaton. By definition, $d_t = j - i$ where j denotes the position of the first leftmost nonzero element of the orbital state x^t and i denotes the position of the first leftmost nonzero element of the state $x^{t\text{-}1}$, assuming that $q = 0$.

The sequence Δ plays a significant role in filter automata theory. We will call it the filtron edge number. Note, that for a fixed $d_{max} = r$, the sequence Δ can be expressed as a number:

$$\chi = \sum_{i=0}^{p-1} (r+1)^i d_i \,.$$

To summarise, we characterise the filtrons by the following parameters:

- period p and delay d $(d = d_{p\text{-}1} + ... + d_0)$,
- filtron edge number χ or row displacements $\Delta = (d_{p\text{-}1}, d_{p\text{-}2}, ..., d_0)$,
- orbit $(x^0, x^1, ..., x^{p\text{-}1})$ and orbital states (also translational and orbital phases [34]),
- filtron left edge (leftmost symbols); $\varepsilon = (x_1^{p\text{-}1}, x_1^{p\text{-}2}, ..., x_1^0)$,
- numbers h_t of series of internal idle pulses; $t = p$-1, ..., 0, and $h = max\{h_t\}$,
- distribution of IIPs; $vip(i) = (nip_{p\text{-}1}(i), nip_{p\text{-}2}(i), ..., nip_0(i))$, $i = 1, ..., h$,
- internal edges (of 'holes'); $vis(i) = (e^{p\text{-}1}(i), e^{p\text{-}2}(i), ..., e^0(i))$, $i = 1, ..., h$.

4.2. Some classes of filtrons

Now we may recall [Siwak, 1993] the classification of periodic strings. It is based on:
- the number c of filter automaton activity cycles involved,
- the number $1+h$ of connected components,
- the distribution of IIPs described by vectors $vip(i)$.

First we have three classes of periodic strings consisting of one component of connectedness. They have just been defined to be the connected filtrons.

- The fleeting filtrons. They are the connected filtrons which involve only one cycle of activity of the associated filter automaton; $c = 1$.
- The simple filtrons are those connected filtrons which imply exactly two cycles ($c = 2$) of activity of the associated filter automaton: the first one being the upholding cycle and the second one - the extinction cycle.
- The connected filtrons. According to the definition they have no IIPs; $h_t = 0$ for any t.

Secondly, we propose a number of classes of periodic objects which are not connected filtrons. They will be called non-connected filtrons or simply filtrons.

- The complex filtrons; they are such that $h \geq 2$.
- The consistent filtrons. This family contains non-connected filtrons which are described by the binary vectors $vip(i)$ of IIPs ($h_t(i) \leq 1$).
- The stuck filtrons. This is a subfamily of consistent filtrons which contains multiple copies of some connected filtron.
- The multiple filtrons. This is a subfamily of non-connected filtrons which consist of multiple copies of a certain connected filtron in each step t on the space-time diagram.

There is a huge family of complex periodic objects called the breathers. It seems that this name was used for the first time in the context of PRR filtering [Fokas, *et. al.*, 1990a]. Fokas defined a breather as a periodic string consisting of a number of connected particle substrings bounded together in such a way that they interact solitonically forever. We propose to call them the simple breathers when two interacting connected filtrons may be distinguished, and the complex breathers when at least three connected filtrons interact.

- The simple breathers. This family covers non-connected filtrons such that some orbital state x^t contains exactly two connected filtrons; additionally $h < 2$ and $h_t = 0$ for some $t \in \{0, 1, \ldots, p\text{-}1\}$.
- The complex breathers. We assume that at least three interacting connected filtrons must be distinguishable.
- The strange filtrons constitute quite a different family of filtrons; they are such that imply non-constant number c of cycles of activity of the associated filter automaton.
- One can distinguish also a class of strange breathers. These are complex filtrons, with multiple activations (or inside holes) but such that their connected components can not exist alone.

At the end of this classification let us mention the family which can be called the set of filtrons. The sets consist of a number of non-interacting filtrons. A typical example are the bundles of connected particle substrings which were used in [Steiglitz, *et. al.*, 1988] to perform some computations.

There is also a vast family of objects which are very interesting. At first glance they look like the breathers, but they are unstable and decay spontaneously after some time (number of iterations). These are not periodic objects, of course. We will call them the quasi-filtrons or in short q-filtrons.

It seems that much work still has to be done to clarify the relations between filter automata and their various filtrons.

4.3. How the filtrons are visualised

The filtrons may be visualised in a number of ways. Let us list here some methods we use.

1. Text ST diagrams. They show all rows of ST diagrams using symbols from X.

2. Symbolic ST diagrams. They were used in Fig. 8 and 15. In these diagrams the following special convention is used to make the filtrons more visible:

- natural symbols x from a working alphabet X,
- tail zeros denoted by "-",
- dot symbols " · ", and
- activating symbols printed in bold.

Typically, any string is specified by its original symbols x_i from X, providing they are read by the associated filter automaton F during its activity cycle. The exception are the tail zeros, that is all consecutive zeros preceding immediately the extinction of the automaton F and read still during its activity. These are denoted by the dash symbol "-". All the zero symbols read by the automaton F in its inactive mode are replaced by the dot symbol. We print in bold all those symbols that activate the automaton F.

3. Graphical (pixel) ST diagrams. These show all rows of ST diagrams using pixel codes (if possible colour codes) of symbols from X. The examples are given further.

4. Wave-like planar diagrams. These use special graphical codes to visualise discrete strings in the form of waves. An example is shown in Fig. 11 b).

5. Wave-like pseudo 3-D diagrams. An example of such presentation is shown in Fig. 9.

6. Discrete ST diagrams of the filtrons. An example is shown in Fig. 10.

Let us now illustrate some notions introduced above.

Example 2. In Table 2 we show the symbolic ST diagram of a complex breather of the automaton $F_2 = (X, \iota, \mu, \kappa)$, where: $X = \{0,1\}$, $\iota(0) = 0$ and $\iota(1) = 1$, μ is represented by the reset conditions $\mu^{-1}(0) = \{**00\}$, and the program $\kappa = (h_0, h_1, f_1, f_2, f_3)$ is such that $h_0 = \mathbf{0}$, $h_1 = f_2 = N$ (negate), $f_1 = f_3 = A$ (accept). Note that $q = 3$.

Table 2. A complex breather of the automaton F_2.

t	orbital states of the complex breather	d_t	h_t	$vip(i)$		
				$i =$ 1	2	3
00	················**1**---·**1**10100010001----··	2	1	2	0	0
01	···············**1**---**1**1110111011101--····	2	1	1	0	0
02	··············**1**0010111011101111---·····	2	0	0	0	0
03	·············**1**1110111011101--**1**---······	2	1	1	0	0
04	···········**1**0111011101111--·**1**---·······	1	1	2	0	0
05	···········**1**---**1**---**1**011----**1**---········	3	3	1	1	1
06	··········**1**---**1**---·**1**010001----·········	2	2	1	2	0
07	·········**1**---**1**---**1**111011101--··········	2	2	1	1	0
08	········**1**---**1**001011101111---···········	2	1	1	0	0
09	·······**1**---**1**111011101--**1**---············	2	2	1	1	0
10	······**1**001011101111---**1**---·············	2	1	1	0	0
11	·····**1**111011101--**1**---**1**---··············	2	2	1	1	0
12	···**1**011101111--·**1**---**1**---···············	1	2	2	1	0
13	···**1**---**1**011----**1**---**1**---················	3	3	1	1	1
14	··**1**---·**1**10100010001----················	2	1	2	0	0

Example 3. Consider a filter automaton $F_3 = (X, \iota, \mu, \kappa)$ defined by: $X = \{0, 1, 2, 3\}$, the activating function ι such that $\iota(x) = 1$ for all $x \neq 0$, and the (only) extinction condition $\mu^{-1}(0) = \{0000\}$. Its program $\kappa = (h_0, h_1, h_2, h_3, f_1, f_2, f_3)$ is as follows:

$$h_0 = \begin{pmatrix} 0123 \\ 0000 \end{pmatrix},\ h_1 = \begin{pmatrix} 0123 \\ 3022 \end{pmatrix},\ h_2 = \begin{pmatrix} 0123 \\ 2301 \end{pmatrix},\ h_3 = \begin{pmatrix} 0123 \\ 0321 \end{pmatrix},\ f_1 = f_3 = \begin{pmatrix} 0123 \\ 0123 \end{pmatrix},\ f_2 = \begin{pmatrix} 0123 \\ 0132 \end{pmatrix}.$$

The ST diagram of a filtron of this automaton is shown below with $q = 0$. The filtron image is exactly the same as that presented in [Fokas, *et. al.*, 1990b] to illustrate FPS filtering. However, the processing performed by our automaton F_3 is quite different (it is seen from the collision of 4 filtrons presented in Fig. 8). The filtron is determined by: $p = 9$, $\Delta = (d_8, \ldots, d_0) = (2,1,1,1,1,2,1,2,1)$ and $\varepsilon = (3,3,3,2,2,1,3,2,1)$; $d = 12$.

```
t=0:  ··10231101----··············  d0 = 1
t=1:  ····2301013-----············  d1 = 2
t=2:  ·····30121302----···········  d2 = 1
t=3:  ·······1223021-----·········  d3 = 2
t=4:  ········22332103----········  d4 = 1
t=5:  ·········23301032----·······  d5 = 1
t=6:  ··········33030322----······  d6 = 1
t=7:  ···········30313221----·····  d7 = 1
t=8:  ·············3102211-----···  d8 = 2
t=9:  ··············10231101----··  d9 = 1
```

The same filter automaton F_3 exhibits the very special ST diagram shown on Plate 1. This is a solitonic collision of some four filtrons. We show it also in Fig. 8 using the symbolic ST diagram. One can note that interaction begins in step 07 with three connected strings. In steps 08 up to 13 there are two connected strings, in steps 14 up to 16 there are three such strings, and then all the four filtrons reappear again.

```
00: ·1----------·······10201---------·····2003-------····1112----·······················
01: ···3------------···3000003------------·32----------·1123----························
02: ·····1--------------·10201-----------·2001--------·1333----·························
03: ·······3------------·3000003----------·12--------·3233----··························
04: ·········1------------·10201---------·2003------·2231----···························
05: ···········3----------·3000003--------·32------·2212----····························
06: ·············1----------·10201--------2001----·2122----·····························
07: ···············3--------·3000003------·12000001322-----·····························
08: ·················1---------·1020100000200300122303------····························
09: ···················3--------30000030303300023333003-----····························
10: ·····················1--------102000231002032310201-------··························
11: ························30000030303300023333003-----3-------························
12: ··························102000231002032310201-------1------·······················
13: ··························30303300023333003000003-------·3--------··················
14: ··························200300122303-----·10201---------·1--------················
15: ···························3201203032------·3000003--------··3--------··············
16: ··························201302002001-----···10201--------····1--------············
17: ··························1322----·12--------·3000003---------···3-------------·····
18: ·························3323----·2003---------·10201-------------·1------------····
19: ························3331------·32----------·3000003-------------·3-----------···
20: ·······················3211------·2001-----------·10201---------------·1----------··
21: ······················2111--------·12-------------·3000003--------------·3---------·
22: ·····················1112--------·2003-------------·10201-----------------·1-------·
```

Fig. 8. Solitonic collision of the four filtrons. Range $q = 2$.

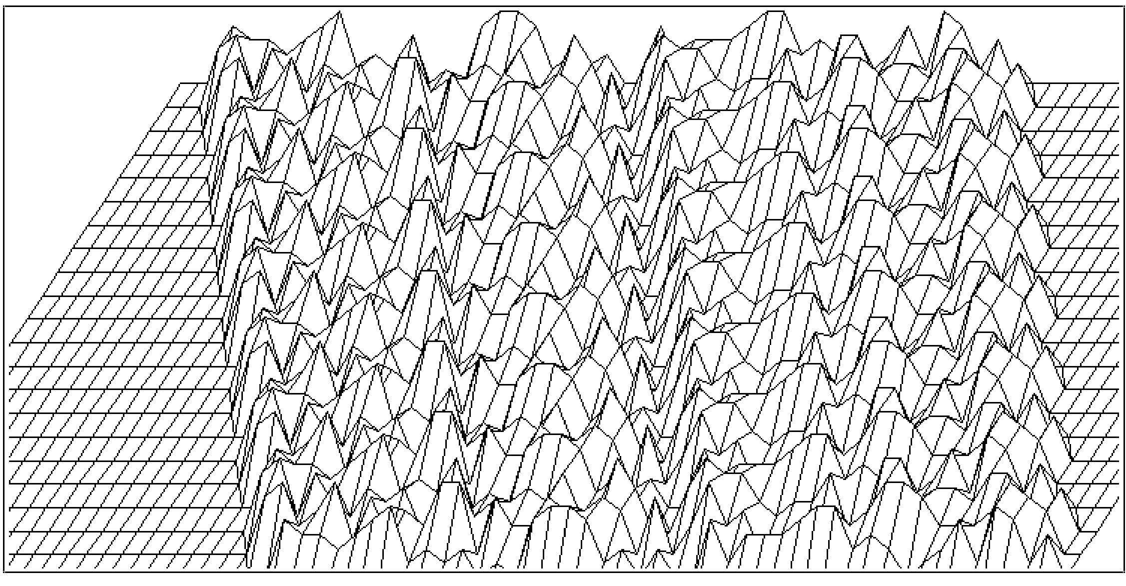

Fig. 9. "3-D" view at a multi-valued filtron.

Example 4. In this example we present Fig. 9 with the wave-like pseudo 3-D ST diagram of a filtron of some filter automaton.

...0 433234512120104554120200413334552320124154130000000000 0...
...0 045142132300512232411530035442122400552332441130000000 0...
...0 000452351413255042243343040252301013235542223543000000 0...
...0 000000433234512120104554120200413334552320124154130000 0...

Example 5. In Fig. 10 we show the discrete ST diagrams of a filtron of period $p = 3$ over $X = \{0, 1, 2, 3, 4\}$. The edge number is $\chi = 4^2 \cdot d_2 + 4^1 \cdot d_1 + 4^0 \cdot d_0 = 16 \cdot 3 + 4 \cdot 1 + 1 \cdot 2 = 54$. There are six cycles of activity of associated filter automaton. The distribution of cycles is shown below each diagram. From such diagrams one can easily determine the activating function ι, some functions of program κ of the automaton, and the control function. We have:

$$h_0 = \begin{pmatrix} 01234 \\ 0{*}0{*}{*} \end{pmatrix},\ h_1 = \begin{pmatrix} 01234 \\ 340{*}{*} \end{pmatrix},\ h_2 = \begin{pmatrix} 01234 \\ 34021 \end{pmatrix},\ f_1 = \begin{pmatrix} 01234 \\ 13420 \end{pmatrix},\ f_2 = \begin{pmatrix} 01234 \\ 12403 \end{pmatrix},\ f_3 = \begin{pmatrix} 01234 \\ 03142 \end{pmatrix}.$$

Note, that three reset conditions must be defined in associated filter automaton to support this filtron, too. These are: $\mu^{-1}(0) = \{0000, 1100, 1000\}$. The functions h_1 have been used also (four times), so $\mu^{-1}(1) = \{2322, 4411, 4400, 0310\}$.

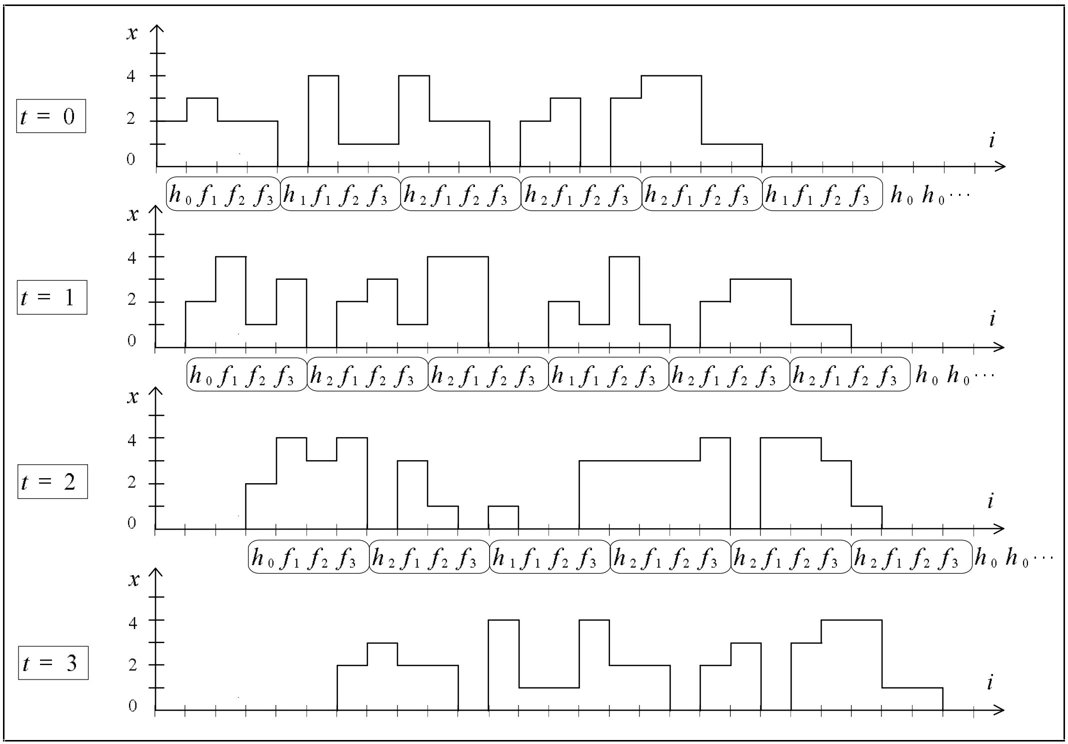

Fig. 10. Discrete ST diagrams of a filtron and the distribution of cycles of activity (h_i, f_1, f_2, f_3).

5. Solitons - their properties and a short history

The interaction of filtrons associated with automata, especially their collisions, bear a rather close resemblance to solitonic waves. They are well known in nonlinear transmitting media. However, it seems to be not easy to give a comprehensive and precise definition of solitons [Drazin and Johnson, 1989], [Newell, 1985], [Taylor, 1992], [Olver and Sattinger, 1990]. This entity is a local travelling wave pulse, a lump like, coherent structure - the solution of a field equation. It exhibits remarkable stability and particle-like properties in collisions.

We show an example of typical solitonic solution, the pulse of *sech* profile in Fig. 11 a), and solitonic collision of two filtrons of some simple filter automaton in Fig. 11 b). Solitonic collisions mean something special. First, when the objects approach closely enough they attract one another, just like nuclear particles do. Then, they interact and later reappear again and travel separately. Each object precisely regains its former identity. So their shapes, speeds, amplitudes and internal frequencies are preserved under collision. The only memory of the collision is a translational shift. Note that the spatial positions of filtrons presented in Fig. 11 b) are changed a little, just as if they were attracted in the first phase of collision. In general, to classify the collision to be a solitonic one, it is also required that such a nonlinear superposition of n colliding objects must occur for an arbitrary n. However, it seems that no solution presenting the solitonic collision of more than three solitons has been ever published. The reason lies mainly in extremely explosive numerical difficulties connected with the traditional approach, namely, when one simulates the behaviour of three or more colliding solitons by means of nonlinear wave equations using the inverse scattering method [Ablowitz, 1990].

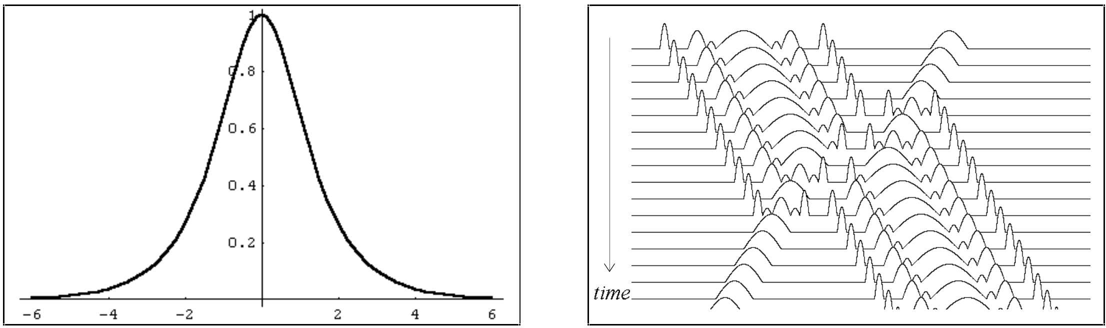

Fig. 11. a) Unitary pulse ***sech***(x) for $x \in [-6, +6]$. b) Solitonic collision of two filtrons of F.

In summary, it is widely accepted [Newell, 1985], [Majewski, 1993] that the soliton is a solution of a nonlinear differential (NLD) wave equation which owes its existence to a balance of two forces; linear one which acts to disperse the pulse (group velocity dispersion GVD), and nonlinear one which focuses it (e.g. Kerr effect), and features:

– remarkable stability against disturbances,
– periodicity and associated individual attributes (translational and orbital phases),
– particle-like properties (solitons attract one another),
– amplitude-dependent velocity (bigger waves travel faster),
– non-destructive collisions,
– nonlinear superposition of n waves.

As will be seen later, all these properties are also attributes of filtrons of filter automata, in particular solitonic collisions of n filtrons have been corroborated for n up to 7.

For the sake of completeness we present here an outlook on some discoveries connected with the phenomenon of the solitary wave.

1844	Scott Russell	- first description of the solitary wave (observation in 1834),
1895	Korteveg, deVries	- equation with the balance between nonlinearity and dispersion,
1940	Fermi, Pasta, Ulam	- unthermalized heat conductivity in solids [Newell, 1985],
1964	Zabusky, Kruskal	- solution to the Korteweg-deVries equations,
1971	Zakharov, Shabat	- inverse scattering method solution to NLS equation,
1973	Hasegawa, Tappert	- optical pulse solution of NL Schrödinger equation for optical fibres
1980	L. F. Mollenauer	- first soliton travels via an optical fibre [Taylor, 1992],
1992	Bell Labs	- first NOR gate and a net protocol performed by interacting solitons
1992	Adam Majewski	- Warsaw Univ. of Technology - lectures given on solitonics,
1993	Drummond et al.	- quantum nature of optical solitons [Drummond, *et. al.*, 1993],
1993	Hyman, Rosenau	- new types of solitons: compactons, peakons.

6. Parallel generation of filtrons

In [Steiglitz, *et. al.*, 1988] the observation has been made that, given the leftmost edge of a particle substring, the parity update rule of PRR filtering uniquely determines the entire object. The left edge of an object was understood to be the pattern formed by the leftmost nonzero bit of each row in the ST diagram of the object over one complete cycle. Steiglitz has proposed a method in which the succeeding layers (indexed by j) of symbols, adjacent to the particle edge, were computed by proceeding upward from the bottom row x_j^{p-1} up to the first one x_j^0. If there was no way to fill in the symbol to be consistent with the update rule, then the conclusion was that no particle substring with the current left edge exists. Consistency with the update rule has been also used in [Papatheodorou, *et. al.*, 1988], [Papatheodorou and Fokas, 1989] in some methods of looking for PRR filtering particle substrings of period $p = 1$.

6.1. The edge net of a filtron

In the case of filter automata a simple extension of the update rule consistency criteria is not possible. Instead, we found it useful [Siwak, 1993] to construct a number of circular nets of some automata G, being closely related to the given filter automata F. These were: a two layer ring net of inverse filter automata, an edge net, an edge register and an edge automaton of the filtron. All these ring models are based on the notion of filtron edge number, which represents the sequence $\Delta = (d_{p\text{-}1}, d_{p\text{-}2}, ..., d_0)$ of row displacements of the filtron. These row displacements play the role of inter-processor delays in our rings. It seems that this notion of communication delays, together with the filtron left edge symbols $\varepsilon = (x_1^{p\text{-}1}, x_1^{p\text{-}2}, ..., x_1^0)$, and the vectors $vip(i) = (nip_{p\text{-}1}(i), nip_{p\text{-}2}(i), ..., nip_0(i))$, and internal edges $vis(i) = (e^{p\text{-}1}(i), e^{p\text{-}2}(i), ..., e^0(i))$ for each $i = 1, ..., h$, open the possibility of an analytical treatment of filter automata and their filtrons.

The crucial step is to generate the filtrons in parallel. Their ST diagram images emerge layer by layer from left to right, instead of row by row which is the case during an iterative processing of the first orbital state x^0 down to the last one $x^{p\text{-}1}$ illustrated in Fig. 2.

The main advantage of this approach is unmasking a connection between any filtron of a given filter automaton F and a ring computation performed in an appropriate circular net. This fundamental connection is based on the notion of inverse filter automata.

An inverse filter automaton F^{-1}, of a given filter automaton F, will be denoted by G. By definition, if a filter automaton F performs the conversion of x^t into x^{t+1}, then its inverse automaton companion G performs the conversion of x^{t+1} into x^t. This means that G reconstructs the input string x^t of automaton F reading the output string $x^{t+1}$of F. Thus, when the automaton F uses the operations h and f, then during its active mode the automaton G must perform the inverse operations to h and f. These inverse operations will be denoted by $g = h^{-1}$ and $e = f^{-1}$.

Let us define now the following ring construction. Assume a filtron is described by its edge number $\Delta = (d_{p\text{-}1}, d_{p\text{-}2}, ..., d_0)$ and edge symbols $\varepsilon = (x_1^{p\text{-}1}, x_1^{p\text{-}2}, ..., x_1^0)$. Suppose that automata G^t, for $t = 0, 1, ..., p\text{-}1$, are placed in p rows along the left edge of a filtron each having one of the assumed edge symbols x_1^t at its output. The inputs of these automata will be taken from the same position but a row below, that is from $..0x^{t+1}0..$. The only exception is the last automaton, $G^{p\text{-}1}$, which reads a symbol from the first row as it were the row denoted by p. This is because the filtron is a periodic object. This results in a ring network of processors G^t shown in Fig. 12.

Now, we are ready to reconstruct the ST diagram of a filtron by a parallel computation. Assume, that all automata G^t are to be shifted in parallel to the right by one position with each clock pulse τ. At each position all automata write symbols to their rows. Note, that in each row t the symbol written to it by the automaton G^t is read by its predecessor in the ring exactly after

d_t instances of time τ. This means that the automata G^t are separated by the delays determined precisely by row displacements $\Delta = (d_{p\text{-}1}, d_{p\text{-}2}, ..., d_0)$ of the filtron. We have then a certain special kind of ring computation. The inter-processor delay lines (IDLs) act simply like shift registers RS_t, and automata G^t operate like combinational mappings $\varphi^t \in \{g_0, ..., g_m, e_1, ..., e_r\}$. The ring net resulting from the above construction (of registers RS_t and operations φ^t) will be called the edge net of a filtron. States of the computation of the edge net of a filtron are determined at any time $\tau = 0, 1, ..., n$ by the symbols $x_{\tau+1}^t$ at the outputs of processors G^t, by the contents s_τ^t of separating shift registers SR_t, and by operations $\varphi_{\tau+1}^t$ performed by all processors.

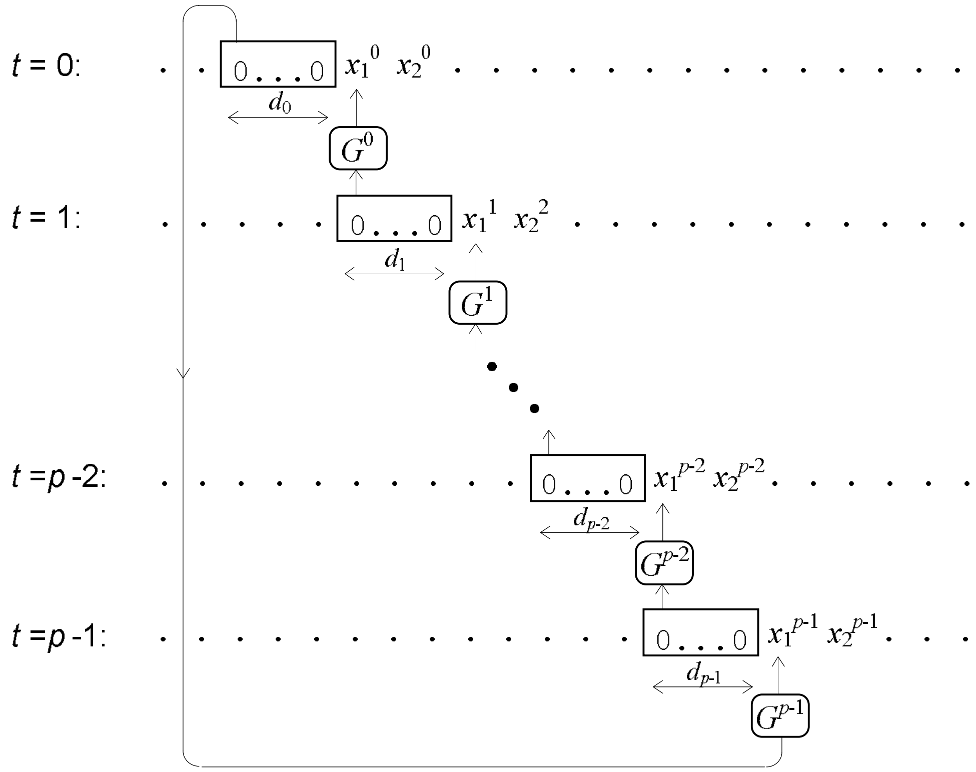

Fig. 12. Ring construction for a filtron with $\Delta = (d_{p\text{-}1}, ..., d_0)$ and $\varepsilon = (x_1^{p\text{-}1}, ..., x_1^0)$.

The ring computation starts at $\tau = 0$ with all IDLs containing only zeros, and with operations $\varphi_1^t = g_0$. Operations g_0 are performed by all processors G^t in such a way that they are activated returning the first (edge) layer $(x_1^0, x_1^1, ..., x_1^{p\text{-}1})$ of a filtron. The consecutive layers of the filtron are generated in succeeding pulses of time τ; the layer $(x_j^0, x_j^1, ..., x_j^{p\text{-}1})$ at time $\tau = j\text{-}1$. The computation is completed after some n pulses when all IDLs are cleared again, and all processors G^t are extinguished (all perform operations g_0). We say that the zero global state is reached by the edge net or simply that the ring computation is succeeded.

The edge net of a filtron is shown in Fig. 13. It is drawn at the moment $\tau = 0$ when all processors perform operations g_0 and the edge symbols x_1^t are expected at their outputs. The ring computation starts being specified by the edge of a filtron.

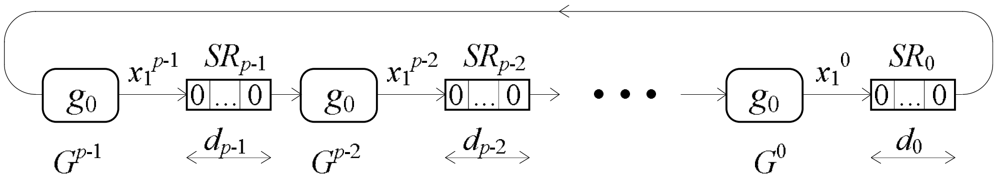

Fig. 13. Start of the edge net of a filtron: processors G^t return edge symbols and SR_t are empty.

<u>Example 6</u>. Let us consider the filter automaton $F_5 = (X, \iota, \mu, \kappa)$ with $X = \{0, 1, 2, 3\}$, $\iota^{-1}(1) = \{1, 2, 3\}$, $\mu^{-1}(0) = \{30{*}{*}, {*}000\}$ and $\kappa = (h_0, h_1, h_2, h_3, f_1, f_2, f_3)$, where:

$$h_0 = \begin{pmatrix} 0123 \\ 0031 \end{pmatrix},\ h_1 = h_2 = h_3 = \begin{pmatrix} 0123 \\ 1031 \end{pmatrix},\ f_1 = f_3 = id,\ f_2 = \begin{pmatrix} 0123 \\ 2301 \end{pmatrix}.$$

The ST diagram given below shows a filtron of period $p = 5$; ($x^5 = x^0$). The row displacements are $\Delta = (d_4, d_3, d_2, d_1, d_0,) = (2,0,2,2,0)$, so $\chi = 552$. The (left) edge symbols are $\varepsilon = (x_1^4, x_1^3, x_1^2, x_1^1, x_1^0) = (3,1,3,1,1)$.

```
t = 0: ···103--------·······     d0 = 0
t = 1: ·····10102--------····    d1 = 2
t = 2: ·······303-2--------···   d2 = 2
t = 3: ·······1010302-·····      d3 = 0
t = 4: ·········301-·······      d4 = 2
t = 5: ·········103-------···    d5 = 0
```

This diagram may be generated as a result of the ring computation of processors G^t (which are the inverse automata to F_5^t) and separating shift registers SR_t. The appropriate edge net of the filtron described by Δ and ε is shown in Fig. 14. We have also:

$$h_0^{-1} = g_0 = \begin{pmatrix} 0 & 1\,2\,3 \\ \{0,1\} & 3-2 \end{pmatrix},\ h_1^{-1} = g_1 = g_2 = g_3 = \begin{pmatrix} 0\ \ 1 & 2\,3 \\ 1\{0,3\} & -2 \end{pmatrix},\ e_1 = e_3 = id,\ f_2^{-1} = e_2 = \begin{pmatrix} 0123 \\ 2301 \end{pmatrix}.$$

From the operation g_0 it is seen that the edge symbols specified by ε are the only possible for the filtron with $\chi = 552$.

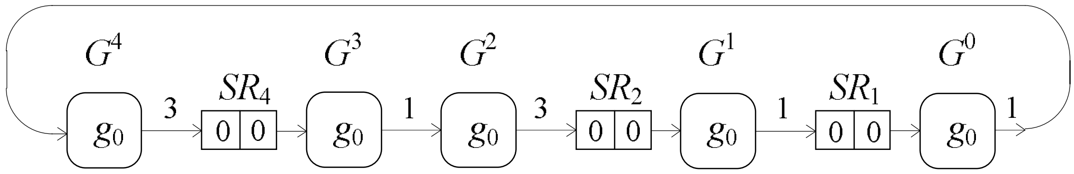

Fig. 14. Edge net of the filtron 103 of edge: $\Delta = (2,0,2,2,0)$ and $\varepsilon = (3,1,3,1,1)$.

The ring computation associated with this filtron is shown in the Table 3. The state of the whole computation at each instant of time τ is determined by the symbols at the outputs of processors G^t, by the contents of separating shift registers SR_t, and by the operations $\varphi_{\tau+1}{}^t$ (g_i or e_i) performed by the processors.

Table 3. Ring computation associated with the filtron of edge number $\chi = 552$.

τ	operations of	state of the ring computation							
	$G^4G^3G^2G^1G^0$	G^4	SR_4	G^3	G^2	SR_2	G^1	SR_1	G^0
0	$g_0g_0g_0g_0g_0$	3	00	1	3	00	1	00	1
1	$e_1e_1e_1e_1e_1$	0	30	0	0	30	0	10	0
2	$e_2e_2e_2e_2e_2$	1	03	1	3	03	1	01	3
3	$e_3e_3e_3e_3e_3$	-	10	0	-	30	0	10	-
4	$g_0g_1g_0g_1g_1$	·	01	3	2	03	2	01	-
5	$g_0e_1e_1e_1e_1$	·	00	0	-	20	-	20	-
6	$g_0e_2e_2e_2e_2$	·	00	2	-	02	-	02	-
7	$g_0e_3e_3e_3e_3$	·	00	-	-	00	-	00	-
8	$g_0g_0g_0g_0g_0$	·	00	·	·	00	·	00	·

There are six occurrences of extinction cycles in this ring computation. Extinguishing segments (reset conditions) are printed on the grey background in the Table 3. It is seen that the machine G^2 was extinguished twice, and its cycles of activity are adjacent. This filtron is a strange object; the number of cycles of automaton's activity varies with orbital states, and actually the state x^4 involves only one such cycle, while all other orbital states involve two cycles.

We present a sequence of pixel ST diagrams on Plate 1 (D:2-9) to show a variety of possible collisions supported by the automaton F_5. Initial configuration is of the form: $\cdot\cdot 01020^{128}(10300000)^5 0^{13} 103020 \cdot\cdot$. It contains three objects: the filtron 102, the bundle of five filtrons 103 and the steady filtron 10302. The separating block of 128 zeros was successively increased giving a number of results; some of them are presented. The range chosen on ST diagrams is $q = 2$.

6.2. Ring computations

The edge net of a filtron shown in Fig. 13 performs a multiprocessor computation. In general, any such computation can be characterised by four components: instruction of processors, inter-processor communication constraints (delays), data used to perform a computational task, and synchronisation or a control to co-ordinate the whole computational process.

All these components are clearly visible in the filtron ring computation. We have:

- instructions; they are represented by the operations of program $\kappa = (h_0, h_1,\ldots, h_m, f_1,\ldots,f_r)$,
- inter-processor delays $\Delta = (d_{p\text{-}1}, d_{p\text{-}2}, \ldots, d_0)$ (expressed as the filtron edge number),
- data represented by the filtron edge symbols $\varepsilon = (x_1^{p-1}, x_1^{p-2}, \ldots, x_1^0)$, and
- synchronisation; the internal synchronisation is determined by the function μ, and the external is imposed by the vectors $vip(i)$.

Note, that having any three out of these four ring computation components one may look for the remaining one. This fundamental fact gives a basis for algorithms aimed at predicting filtrons if the associated filter automaton is given, and for looking for the possible filter automata capable of supporting a prescribed filtron.

In the general case the inverse operations φ^t to the given functions h: $X \to X$ and f: $X \to X$ are the relations. So, there are three possible cases:

- the value $\varphi^t(x)$ is determined uniquely as a single symbol from X,
- $\varphi^t(x)$ is not determined, which will be denoted by $g(x)$ = - or by $e(x)$ = -, and
- $\varphi^t(x)$ represents at least two symbols from the set X.

These three cases imply three different events during the ring computation:

- the computation is performed uniquely,
- there is a deadlock during the ring computation (it is blocked), and
- the branching of computation occurs, which means that it may proceed in many ways.

When the ring computation succeeds, without a deadlock and branching a unique filtron with a given description exists for the chosen filter automaton.

The second possible result of the ring computation is just deadlock. The deadlock may happen when we are looking for the filtron and at some pulse τ certain processors G^t are not able to compute their outputs (they can not return a value $g(x)$ = - or $e(x)$ = -). It is interesting that many filtrons don't exist; their edge numbers seem to be prohibited for a given filter automaton. Deadlock may occur also, if we are looking for the filter automaton which could support a given filtron. In this case, the extinction conditions of the desired filter automaton, which are necessary to stop the ring computation, may cause deadlock before the final zero global state is reached.

In the third case of branching of the ring computation, an external decision is needed to determine which particular result is looked for. One must choose the value from the set $g(x)$ or $e(x)$. In the general case, the vectors $vip(i) = (nip_{p\text{-}1}(i), nip_{p\text{-}2}(i), \ldots, nip_0(i))$, and internal edges $vis(i) = (e^{p-1}(i), e^{p-2}(i), \ldots, e^0(i))$ must be given for the computation associated with a complex filtron to succeed. If there are numerous successful continuations of the ring computation then many filtrons exist with the same given initial description.

It is quite natural to link these three cases of computational process in the edge net of a filtron with the possible results of solving the set of algebraic equations. Such a representative set of equations, for a given pair, filtron and filter automaton, may be deduced from the discrete ST diagrams (such as shown in Fig. 10) of the filtron. The set of filtron equations is obtained immediately from the columns of the ST diagram. The filtron equations are of the form:

$$\text{either } h\,(x_j^{t-1}) = x_{j-d_t}^{\,t} \text{ or } f\,(x_j^{t-1}) = x_{j-d_t}^{\,t}.$$

Example 7. Consider a binary filter automaton $F_6 = (X,\ \iota,\ \mu,\ \kappa)$, where $X = \{0,\ 1\}$, $\iota(0) = 0$, $\iota(1) = 1$, $\mu^{-1}(0) = \{*000\}$, and $\kappa = (h_0, h_1, f_1, f_2, f_3)$, where: $h_0 = 0$, $h_1 = N$, $f_1 = f_2 = f_3 = id$. This automaton is equivalent to PRR filtering with $r = 3$. Assume that we are looking for a filtron with $\Delta = (d_2, d_1, d_0) = (1, 2, 2)$; $\chi = 14$. The associated ring computation is shown in Table 4.

Table 4. Looking for a potential filtron with edge number $\chi = 14$; note a deadlock at $\tau = 12$.

τ	operations	contents of IDLs		
	$\varphi^2\ \varphi^1\ \varphi^0$	SR_2	SR_1	SR_0
0	$g_0\ g_0\ g_0$	0	0 0	00
1	$e_1\ e_1\ e_1$	1	1 0	10
2	$e_2\ e_2\ e_2$	0	1 1	01
3	$e_3\ e_3\ e_3$	1	0 1	10
4	$g_1\ g_1\ g_1$	0	1 0	11
5	$e_1\ e_1\ e_1$	0	1 1	11
6	$e_2\ e_2\ e_2$	1	0 1	11
7	$e_3\ e_3\ e_3$	1	1 0	11
8	$g_1\ g_1\ g_1$	1	1 1	01
9	$e_1\ e_1\ e_1$	0	0 1	00
10	$e_2\ e_2\ e_2$	0	0 0	10
11	$e_3\ e_3\ e_3$	0	0 0	01
12	$g_1\ g_0\ g_1$	1	0 0	00

The required filtron doesn't exist. Its associated ring computation is blocked at $\tau = 12$, because the automaton G^1 is extinguished (it has just introduced 4 consecutive zeros to the SR_1; they are printed on the grey background) and has to perform the operation $g_0(1) = $ -.

Example 8. Consider a binary filter automaton $F_7 = (X,\ \iota,\ \mu,\ \kappa)$, where $X = \{0,\ 1\}$, $\iota(0) = 0$, $\iota(1) = 1$, $\mu^{-1}(0) = \{0011,\ 0000\}$, and $\kappa = (h_0, h_1, f_1, f_2, f_3)$ with: $h_0 = 0$, $h_1 = f_1 = N$, $f_2 = f_3 = id$.

Table 5. Automaton F_7 and filtrons of $\Delta = (2, 3)$. Branching in the ring computation at $\tau = 8$.

τ	operations	IDLs	
	$\varphi^1\ \varphi^0$	SR_1	SR_0
0	$g_0\ g_0$	0 0	0 00
1	$e_1\ e_1$	1 0	1 00
2	$e_2\ e_2$	1 1	1 10
3	$e_3\ e_3$	0 1	1 11
4	$g_1\ g_1$	1 0	1 11
5	$e_1\ e_1$	0 1	1 11
6	$e_2\ e_2$	0 0	- 11
7	$e_3\ e_3$	1 0	- 01
8	$g_0\ g_1$	1 1	- 00
9	$g_0\ e_1$	• 1	- 00
10	$g_0\ e_2$	• 0	- 00
11	$g_0\ e_3$	• 0	- 00
12	$g_0\ g_0$	• 0	- 00

τ	operations	IDLs	
	$\varphi^1\ \varphi^0$	SR_1	SR_0
8	$g_0\ g_1$	1 1	0 00
9	$e_1\ e_1$	**1** 1	0 00
10	$e_2\ e_2$	1 1	0 00
11	$e_3\ e_3$	0 1	1 00
12	$g_1\ g_0$	0 0	1 10
13	$e_1\ e_1$	1 0	**1** 11
14	$e_2\ e_2$	0 1	1 11
15	$e_3\ e_3$	1 0	1 11
16	$g_1\ g_1$	1 1	- 11
17	$e_1\ e_1$	- 1	- 01
18	$e_2\ e_2$	- 0	- 00
19	$e_3\ e_3$	- 0	- 00
20	$g_0\ g_0$	- 0	- 00

Assume that we are looking for a filtron with $\Delta = (d_1, d_0) = (2, 3)$; $\chi = 11$. The associated ring computation is shown in Table 5. It branches at $\tau = 8$. In the left hand part of the Table, the computation has succeeded after two cycles of activity of G^1 and three cycles of activity of G^0. In the right hand part of the Table it has succeeded after five cycles of activity of both machines. The extinction conditions met during the computation are printed on the grey background. The conclusion is that there exist at least two filtrons described by $\chi = 11$, for F_7. Below, we show the ST diagrams of both filtrons. In the first case we have a strange filtron, in the second - a complex object.

```
t=0:   ..11111-------........   ..11111000000111111------......
t=1:   ....11010011..........   ....1101001111001011-----.....
t=2:   .......11111-------...   .......11111000000111111------..
```

7. New phenomena call for physical experiments

In the series of plates which follow, we present a number of phenomena which were observed on ST diagrams of processing performed by filter automata. Some of them can occur neither in PRR filtering [26] nor in FPS filtering, some seem to resemble the solitonic solutions of the wave equations, and some seem to be registered and published for the first time.

The important question as to what extent the model of filter automata may mimic physical fields is waiting for research. Nevertheless, the solitons presented here may provoke some simple physical experiments (like measurement of the minimum distance of solitons before they attract, etc.), possibly with pulses of light. They could focus our attention on the right choices of filter automata family representatives and of their parameters.

```
00: .2-------................22002-------.......20030002----...3002----.....3031----..3133----..
01: ..2-------.............20000002----........320022------...21------...211-----.1231----....
02: ...2-------............20022-------.....20032001----...1002----...1102----.2213----......
03: ....2-------...........200002------.....300012------...23------.1023----.2132----.......
04: .....2-------..........22002-------.....22001-------3002----..333-----1222----..........
05: ......2-------.........20000002----.....20030002----..21------3201----2323----...........
06: .......2-------........20022-------....320022------1002----2011----3332----.............
07: ........2-------.......200002------..20032001----..230000011200003221------.............
08: .........2-------......22002-------.300012------30020013200202201002----................
09: ..........2-------.....20000002----...22001000000210133002123020021-------...............
10: ...........2-------....20022-------.20030002002301310023220000230002----.................
11: ............2-------...200002------.320022033212120330200203320022------.................
12: .............2-------...220020000020032030212220300301203020032001------................
13: ..............2-------.20000002030233010310230100321301030233021002----..................
14: ...............2-------.20022302130123100301202233012302130230022-------.................
15: ................2-------203023301031023010032130103023302100200002------.................
16: .................200000200320302122203003012030200320001------22002-------...............
17: ..................203023301031023010032130103023302100200002------2-------...............
18: .................20032030212220300301203020032001----..22002-------2-------..............
19: .................30020013200202301002----300012------20000002----...2-------.............
20: .................210133002122020021------.22001-------20022-------...2-------............
21: ...............10113000132220001002----.20030002----..200002------....2-------...........
22: ..............11200003322-----.23------.320022------...22002-------....2-------..........
23: ............130300322303----.3002----.20032001----...20000002----.......2-------.........
24: ..........303302223033----...21------.300012------...20022-------......2-------.........
25: .........23123200231-----..1002----....22001-------...200002------.......2-------........
26: .......312120003102----....23------..20030002----......22002-------.......2-------.......
27: .....1313----1021----....3002----....320022------....20000002----..........2-------......
28: ...3133----.313----.....21------..20032001----.......20022-------..........2-------.....
29: .1231----.1201----.....1002----....300012------.......200002------...........2-------....
```

Fig. 15. Six filtrons of periods 1, 4, 5, 4, 18 and 12 collide solitonically. F_3 is the supporting filter automaton; $q = 3$.

Plates 1-10 with diagrams D:1 to D:72 present the following phenomena:

- solitonic collision of four filtrons; diagram D:1 (see also Fig. 8),
- solitonic collisions of six filtrons; diagrams D:10 (the same solitonic collision of six filtrons of the filter automaton F_3 is shown in Fig. 15 in the form of symbolic ST diagram), D:11,

diagram D:12 presents a complicated but solitonic collision of five filtrons and a breather, diagrams D:13 to D:16 present some non-solitonic collisions of six filtrons,

- solitonic collisions of seven filtrons; diagrams D:17, D:18, three other experiments of looking for solitonic collision are presented on D:19 to D:21,
- a collision of nine filtrons which is almost solitonic; diagram D:22,
- fusion of filtrons into breathers; diagrams D:15, D:23, D:26, D:27,
- fusion of a breather and a bundle of filtrons into a very complicated ('heavy') breather; diagrams D:41, D:42,
- fusion of a breather and a filtron into q-filtrons; diagrams D:24, D:27, D:37, D:43 to D:49,
- quasi-filtrons - a spontaneous decay; diagrams D:43 up to D:49, D:67, D:72 (the life times of presented q-filtrons are: 66, 212 and 160 steps, respectively),
- quasi-filtrons which are able to collide solitonically; the examples are shown on Plate 9. Each pair of diagrams (D:61-62, D:63-64, D:65-66) presents solitonic and non-solitonic collision of the unstable complex filtron with two other filtrons. Initial positions differ only with the number of separating zeros. The diagram D:61 starts with the configuration $A0^{350}B0^{25}C0^{87}D$ and the diagram D:62 starts with the configuration $A0^{339}B0^{25}C0^{87}D$. The configurations of diagrams D:63 and D:64 are: $A0^{308}B0^{25}C0^{87}D$ and $A0^{307}B0^{25}C0^{87}D$. Here, A, B, C and D denote the following filtrons: A = 2, B = 301000030200022, C = 3030113, and D = 22310211. See also diagrams D:53 to D:58, D:68, D:69, D:71, and D:72,
- stimulated decay (fission) of quasi-filtrons; diagrams D:67 and D:70, D:71,
- bouncing of filtrons; diagrams D:25, D:28, D:32, D:51,
- emission to a bound state; an example is shown in Fig. 16,
- jumps over a bundle; diagrams D:53 to D:58, and the right hand side of D:72 - a fleeting filtron jumps instantaneously over the bundle of 8 filtrons,
- how a bundle of fleeting filtrons does interact with a quasi-filtron; presented on Plate 8,
- immediate change in speed of a bundle of filtrons with any width; diagram D:72 - on the right hand side after the second collision with a filtron the whole bundle immediately changes its speed,
- annihilation of filtrons; presented in Example 10.

Most of the diagrams use the filter automaton F_3 specified in the Example 3. To illustrate the phenomena of emission to a bound state and annihilation of filtrons we present two additional examples.

Example 9. Consider a binary filter automaton $F_8 = (X, \iota, \mu, \kappa)$ with $\iota(0) = 0$, $\iota(1) = 1$, $\mu^{-1}(0) = \{*000\}$, $\mu^{-1}(1) = \{*01\}$, and program $\kappa = (h_0, h_1, f_1, f_2, f_3, f_4)$, where: $h_0 = 0$, $h_1 = f_1 = N$, $f_2 = f_3 = f_4 = id$. The collisions presented in Fig. 16 result in a breather that looks like an emission to a bound state. Notice the fleeting filtron emitted periodically to the left and 'attracted' again.

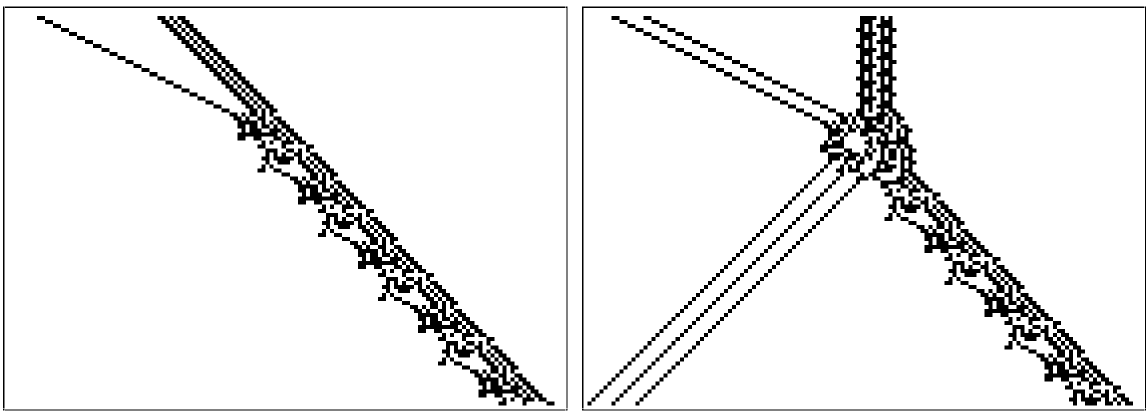

Fig. 16. Breather of the automaton F_8 looks like an emission to a bound state.

There are many filter automata, especially with the extinction conditions defined by various totalistic functions [Albert and Culik II, 1987], [Smith, 1992], which support the annihilating filtrons. Note, that the phenomenon of annihilation was conjectured [Park, *et. al.*, 1986] to be non-existing in PRR filtering.

Example 10. Consider a binary filter automaton $F_9 = (X, \iota, \mu, \kappa)$. $X = \{0, 1\}$, $\iota(0) = 0$, $\iota(1) = 1$, $\mu^{-1}(0) = X^4 \setminus \{1110, 1111\}$, and $\kappa = (h_0, h_1, f_1, f_2, f_3)$, where: $h_0 = 0$, $h_1 = f_1 = N$, $f_2 = f_3 = id$. The diagram below shows the phenomenon of an annihilation of two filtrons; the first is a strange filtron, and the second is the fleeting filtron. The range is $q = 1$.

```
t= 0:   ···111-----··1---···
t= 1:   ····1011·····1---···
t= 2:   ····111-----·1---···
t= 3:   ·····1011····1---···
t= 4:   ·····111-----1---···
t= 5:   ······1011···1---···
t= 6:   ······11100001······
t= 7:   ·······1011·1---····
t= 8:   ·······111001--·····
t= 9:   ········101-········
t=10:   ········11--········
t=11:   ····················
```

8. Final remarks

We have presented here the computational model of filter automata. Their interactions with strings of symbols create amazing and fascinating, rich behaviour with striking resemblance to the physical phenomenon of solitary waves. It makes them an interesting, new tool which should be useful in the domain of looking for physics in computational models.

The FA models are simple and elegant. They allow easy control of their parameters. They seem to be the right choice for sites in cellular spaces to represent many aspects of physical reality. Note, that identifying a subclass of finite state machines proper to play this role is by no means a trivial task [Requardt, 1995].

There are many indications that investigating discrete solitons based on filter automata has the potential to dramatically increase our understanding of the behaviour of some complex physical phenomena. According to the present model we are able to give some answer to the question: why do two solitons jump, one over the other? It seems that the answer is because the underlying field is excited for a while after the passage of the first object. In terms of filter automata model we say that a trace was left in the form of a certain segment of activated automata. The consequence is that both objects can 'communicate'.

Here we mention a few non-rigorous comments about possible applications of filter automata drawn from the literature and current research.

Quantum discrete models. It has been shown that solitons observed experimentally in optical fibres have quantum-mechanical properties [Drummond, *et. al.*, 1993]. There is a hope that discrete models will be more appropriate to describe this phenomena. At least, they should enrich the approach represented by solutions to the NLS equations.

Space as a transducer. Some data series cannot be simulated by cellular automata, so serial transducers are necessary to do it, as was concluded in many papers. Filter automata represent another proposition. They are treated as OLIA models with cells similar to Mealy type automata. Such FA nodes, when activated, are able to create very particular, propagating segments. These segments have the potential to transmit signals without a lapse of time. On the contrary, cellular automata with Moore type cells are non-propagating in this sense.

Solitons and elementary nuclear particles. The importance of particle substrings in arranging computations within complex systems, which spread over space, is well recognised [Garzon, 1995], [Requardt, 1995], [Zurek, 1990]. But so far only 1-D particle substrings (here the filtrons) have been studied. The current results show that it should be worthwhile look for 2-D and 3-D filtrons and investigate their properties. One may expect that these new 3-D discrete solitons will reveal significant similarities with the behaviour of elementary nuclear particles.

Optical fibre processors. It is highly probable that solitons will be used in future optical fibre processors. The computations will be performed by interacting streams of solitons during their transmission through the optical fibre. Note that technologically it is already possible to shape the transmitting properties of optical fibres, e.g. by means of modification of their geometry, application of mechanical tensions and applying chemical doping [Majewski, 1993], [Taylor, 1992].

Easy and efficient simulations. There are gigantic problems with simulation of solitons by means of nonlinear PDEs; namely how to interpret physical reality in equation parameters, and how to perform efficiently the numerical simulations. Quite recently (see e.g. "Soliton surprise" - October 1993 SIAM News) Mac Hyman from The Center for Nonlinear Studies at Los Alamos NL wrote: "some of the computer budgets went through the roof, but it was worth it". He discovered compactons and peakons. The discrete models, like filter automata, seem to offer new possibilities here. They make it possible to follow the behaviour of and experiment with hundreds of filtrons per second using only PCs. There is a rather small hope of improving the numerical approximations and calculations of nonlinear PDEs.

Discrete wave equations. Some authors [Feddersen, 1991] are trying to use discrete approximations of NLS equations, but still fundamental problems with interpretation and efficiency remain.

Clearly defined components. Many researchers come to the conclusion that discrete sequential models not only are of sufficient computational power to perform computations in extended complex systems [Lindgren and Nordal, 1990], [Zurek, 1990], [Martin, 1992], [Smith, 1992], [Garzon, 1995], but also are useful in classifying, extracting and handling crucial behavioural components of such events.

Predicting new soliton phenomena. Many surprising properties of solitons have been discovered recently in physics, like special collisions, bouncing, annihilations, breaking-up etc. Using filter automata we can generate, classify and experiment efficiently with single filtrons as well as with streams of filtrons. As we have demonstrated, the emerging filtron interaction picture is remarkably rich; it covers all reported phenomena, and we believe that it allows us to predict many more which are not yet experimentally discovered.

Complex solitons. In the relevant literature the study is restricted almost entirely to simple solitons or those that resemble singular pulses. The reason is that large solitons with complicated and periodically changing shapes (which are said to be of higher orders) are extremely difficult in analytical treatment. FA models help to reveal clearly the orbital history of very complex solitons.

Hidden parameters. There is a possibility of explaining the phenomenon of instantaneous jumping of one soliton over another (even a very long one). The Mealy type sites are crucial here to support translational shift of interacting filtrons. In such a case there are no "hidden parameters", nevertheless "jumps" in the cellular space are possible, because of activated segments of cells.

There arise also the question of where filter automata and their filtrons could be of interest. Here is an opinion. Filter automata constitute an elegant theoretical model of some complex systems and offer efficient analytical methods for investigating their dynamics. It seems that the cognitive value of this approach could be verified in the following disciplines.

- **Theoretical physics**. The pipeline of filter automata may represent a transmitting medium whereas the travelling strings of symbols are related to signals. In FA the mutual interaction of these two elements can be easily and numerically efficiently analysed.
- **Signal processing**. There are a number of methods of signal processing; pipeline, cellular, systolic, iterative, data flow driven, wavefront, etc. The models of FA integrate many of them into a uniform, parallel, wavefront-like processing.
- **Parallel computations**. The theory and the practice of parallel computations may take advantage from FA: by developing the ideas of solitonic computations shown in [Steiglitz, *et. al.*, 1988] and by performing such computations in some specialised linear solitonic processors, e.g. in properly shaped optical fibres.
- **Dynamics of nonlinear systems**. The space-time diagram represents a special case of a phase space. This means that iterative computations in FA can be analysed as phase portraits. On space-time diagrams one may observe chaos, strange attractors and all the other phenomena of nonlinear dynamical systems. Filter automata easily allow one to tune their parameters so it should be possible to check many ideas from nonlinear dynamical systems, e.g. the edge of chaos conditions.
- **Software engineering**. FA make use of notions similar to the iterating loop, concatenation and conditional branch. This may suggest replacing some programs by an equivalent sequence of appropriate (hardware) filter automata (segments of a specialised optical fibre).
- **Theoretical genetics**. Simple FA can be determined by four fundamental functions. Their operation over four element sequences suggest possible relations with the processing of ACTG sequences.

9. Acknowledgements

After a presentation of some ideas associated with filter automata at the Workshop on New Connections between Mathematics and Computer Science at the University of Cambridge Isaac Newton Institute (UK), held 20-24 November, 1995, I had a number of interesting questions, especially from Dr. Keith Bowden from the University of East London. I would like to express my thanks for stimulating discussion and for his encouragement to prepare this paper.

References

Ablowitz, M.J. [1990], “Nonlinear evolution equations, inverse scattering and cellular automata.” In: *Solitons in Physics, Mathematics and Nonlinear Optics*, edited by P.J. Olver, D.H. Sattinger, Springer-Verlag, New York, pp. 1-26.

Ablowitz, M.J. J.M. Keiser and L.A. Takhtajan [1991], “Class of stable multistate time-reversible cellular automata with rich particle content.” *Physical Review A*, **44** (10), pp. 6909-6912.

Albert, J. and K. Culik II [1987], “A simple universal cellular automaton and its one-way and totalistic version.” *Complex Systems*, **1**, pp. 1-16.

Buning, H.K. and L. Priese [1980], “Universal asychronous iterative arrays of Mealy automata.” *Acta Informatica*, **13**, pp. 269-285.

Drazin, P.G. and R.S. Johnson [1989], *Solitons: an introduction*. Cambridge University Press, Cambridge.

Drummond, P.D., R.M. Shelby, S.R. Friberg, and Y. Yamamoto [1993], "Quantum solitons in optical fibers." *Nature*, **365** (6444), pp. 307-313.

Feddersen, H. [1991], "Solitary wave solutions to the discrete nonlinear Schrodinger equation." In: *Nonlinear Coherent Structures in Physics and Biology*, edited by M. Remoissenet, M. Peyrard, Lecture Notes in Physics, v. 393, Springer-Verlag, pp. 159-167.

Fitch, J.P., E.J. Coyle and N.C. Gallagher [1984], "Median filtering by threshold decomposition." *IEEE Transactions on Acoustic, Speech, and Signal Processing*, **32**(6), pp. 1183-1188.

Fokas, A.S., E.P. Papadopoulou and Y.G. Saridakis [1989a], "Particles in soliton cellular automata." *Complex Systems*, **3**, pp. 615-633.

Fokas, A.S., E.P. Papadopoulu, Y.G. Saridakis and M.J. Ablowitz, [1989b], "Interaction of simple particles in soliton cellular automata." *Studies in Applied Mathematics*, **81**, pp. 153-180.

Fokas, A.S. [1990], "New manifestations of Solitons." In: *Solitons in Physics, Mathematics and Nonlinear Optics*, edited by P.J. Olver and D.H. Sattinger, Springer-Verlag, New York, pp. 65-77.

Fokas, A.S., E.P. Papadopoulou and Y.G. Saridakis [1990a], "Soliton cellular automata." *Physica D* **41**, pp. 297-321.

Fokas, A.S., E.P. Papadopoulou and Y.G. Saridakis [1990b], "Coherent structures in cellular automata." *Physics Letters A*, **147**(7), pp. 369-379.

Garzon, M. [1995], *Models of massive parallelism. Analysis of cellular automata and neural networks*. Springer-Verlag, Berlin.

Ibarra, O.H., T. Jiang and H. Wang [1992], "String editing on a one-way linear array of FSMs." *IEEE Transactions on Computers*, **41**(1), pp. 112-118.

Lindgren, K. and M.G. Nordal [1990], "Universal computation in simple one-dimensional cellular automata." *Complex Systems*, **4**, pp. 299-318.

Majewski, A. [1993], *Nonlinear fiber optics*. (In Polish). Publication of Technical University of Warsaw, Poland.

Martin, B. [1992], "Efficient unidimensional universal cellular automaton." In: *Mathematical Foundations of Computer Science Proceedings*, Lecture Notes in Computer Science, **629**, Springer Verlag, pp. 374-382.

Mikolajczak, B. [1991] (editor), *Algebraic and structural automata theory*. Annals of Discrete Mathematics, **44**, Elsevier, Amsterdam.

Newell, A.C. [1985], *Solitons in Mathematics and Physics*. University of Arizona, Society for Industrial and Applied Mathematics, Philadelphia.

Olver, P.J. and D.H. Sattinger [1990], (editors), *Solitons in Physics, Mathematics, and Nonlinear Optics*. Springer-Verlag, New York.

Papatheodorou, T.S., M.J. Ablowitz and Y.G. Saridakis [1988], "A rule for fast computation and analysis of soliton automata." *Studies in Applied Mathematics*, **79**, pp. 173-184.

Papatheodorou, T.S. and A.S. Fokas [1989], "Evolution theory, periodic particles and solitons in cellular automata." *Studies in Applied Mathematics*, **80**, pp. 165-182.

Papatheodorou, T.S. and N.B. Tsantanis [1989], "Fast soliton automata." In: *Optimal algorithms*, edited by H. Djidjev, Lecture Notes in Computer Science, **401**, Springer-Verlag, Berlin, pp. 41-47.

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

Puchalka, T. and P. Siwak [1995], "Automata and languages." In: *Proceedings of the Second International Symposium on Methods and Models in Automation and Robotics*, edited by S. Banka, S. Domek, Z. Emirsajlow, 30 August - 2 September, Międzyzdroje, Publication of Technical University of Szczecin, pp. 467-480.

Requardt, M. [1995], "Discrete mathematics and physics on the Plank-scale." Preprint 9504118 hep-th lanl, Institut fur Theoretische Physic, Universitat Gotingen, Germany.

Siwak, P. [1992], "Particles of parity-rule recursive filtering constructed by means of particle edge automata." Technical Report 1, Departement of Automation, Robotics and Computer Science, Poznan University of Technology, pp. 1-30.

Siwak, P. [1993], "Introduction to filter automata theory." *Studia z Automatyki*, **18**, pp. 87-110.

Siwak, P. [1994], "Models of filter automata." (In Polish). *Studia z Informatyki i Automatyki*. **19**, pp. 77-97.

Siwak, P. [1995], "Filter automata and PRR filtering." *Journal of Technical Physics*. **36**(2), pp. 153-163.

Smith, A.R. [1992], "Simple nontrivial self-reproducing machines." In: *Artificial Life*, edited by C.G. Langton, C. Taylor, J.D. Farmer and S. Rasmussen, Addison-Wesley Publishing Company, Reading, Massachusetts, pp. 709-725.

Steiglitz, K., I. Kamal and A. Watson [1988], "Embedding computation in one-dimensional automata by phase coding solitons." *IEEE Transactions on Computers*, **37**(2), pp. 138-145.

Taylor, J.R. [1992] (editor), *Optical Solitons - Theory and Experiment*. Cambridge University Press, Cambridge.

Zurek, W. [1990] (editor), *Complexity, Entropy, and the Physics of Information*. Addison-Wesley Publishing Company, Reading, Massachusetts.