Computing in Spiral Rule Reaction-Diffusion Hexagonal Cellular Automaton

Andrew Adamatzky[*]
Faculty of Computing, Engineering, and Mathematical Sciences,
University of the West of England,
Bristol BS16 1QY, United Kingdom

Andrew Wuensche[†]
Discrete Dynamics Lab, United Kingdom

A hexagonal ternary-state two-dimensional cellular automaton is designed which imitates an activator-inhibitor reaction-diffusion system, where the activator is self-inhibited in particular concentrations and the inhibitor dissociates in the absence of the activator. The automaton exhibits both stationary and mobile localizations (eaters and gliders), and generators of mobile localizations (glider-guns). A remarkable feature of the automaton is the existence of spiral glider-guns, a discrete analog of a spiral wave that splits into localized wave-fragments (gliders) at some distance from the spiral tip. It is demonstrated that the rich spatio-temporal dynamics of interacting traveling localizations and their generators can be used to implement computation, namely manipulation with signals, binary logical operations, multiple-value operations, and finite-state machines.

1. Introduction

Reaction-diffusion chemical systems are widely known for their ability to perform various types of computation, from image processing and computational geometry to the control of robot navigation and the implementation of logical circuits. In a reaction-diffusion computing medium, data are represented by the spatial configuration of the medium (e.g., local drastic changes of reagent concentrations or excitations), information is transferred by spreading diffusion or excitation waves and patterns, computation is implemented by interactions between spreading patterns, and the results of computations are represented by the final concentration profile or the dynamic structure of excitations. Numerous examples of simulated and chemical laboratory computers can be found in [1].

[*]Electronic mail address: andrew.adamatzky@uwe.ac.uk.
[†]Electronic mail address: andy@ddlab.org.

Computation in a reaction-diffusion medium can be perceived as structureless, or architectureless, because every microdomain of the medium can potentially conduct information in the form of a diffusion front or phase wave front. This absence of compartmentalization in reaction-diffusion computing systems fits extremely well within the paradigm of collision-based computing [2], with its logical computation roots in Conway's Game-of-Life [3], Fredkin–Toffoli's conservative logic [4], and Margolus's physics of computation [5]. In collision-based computing, quanta of information are represented by compact patterns traveling in an "empty" space and performing computation by mutual collisions. The absence or presence, as well as the type, of traveling patterns encode values of logical variables. The trajectories of patterns approaching a collision site represent input variables, and the trajectories of the patterns ejected from a collision, and traveling away from the collision site, represent the results of logical operations, output variables. The compact patterns can be billiard balls in theoretical models, solitons, kinks, or breathers in studies of molecular systems, and cellular automaton (CA) gliders. There is a particular type of reaction-diffusion chemical system, the Belousov–Zhabotinsky reaction in subexcitable mode [6], that supports the existence of localized wave-fragments (somewhat analogous to dissipative solitons [7]) which can play the role of the "billiard-balls" in a collison-based computing system.

Previously we demonstrated that by using localized wave-fragments in experimental and simulated reaction-diffusion systems we could implement functionally complete sets of logical gates and varieties of binary logic circuits [1]. The functionality of these constructions, however, lasts for a markedly brief time because the unstructured reaction-diffusion excitable devices lack stationary localizations (which could be used as memory units) and stationary generators of mobile localizations (which are essential for implementing negation).

In our search for real-life chemical systems exhibiting both mobile and stationary localizations we discovered a CA model [8] of an abstract reaction-diffusion system, which ideally fits the framework of the collision-based computing paradigm and reaction-diffusion computing. The phenomenology of the automaton was discussed in detail in [8], therefore in the present paper we draw together the computational properties of the reaction-diffusion CA.

Why have we chosen cellular automata (CAs) to study computation in reaction-diffusion media? Because CAs can provide just the right fast prototypes of reaction-diffusion models. The examples of "best practice" include models of Belousov–Zhabotinsky reactions and other excitable systems [9, 10], chemical systems exhibiting Turing patterns [11–13], precipitating systems [1], calcium wave dynamics [14], and chemical turbulence [15].

We therefore consider it reasonable to interpret the CA rules we have discovered in terms of reaction-diffusion chemical systems. We envisage that this interpretation will provide the basis for experimental chemical laboratory designs of reaction-diffusion computers, allowing stationary localizations to be used as memory units [2].

Constructing logical gates is a prerequisite for demonstrating the computational universality of a system. However, to build working prototypes we need to have more detailed techniques for manipulating signals, memorizing the intermediary results of a computation, and feeding data into the computing device, to name but a few. This is why we mainly concentrate on these "auxiliary" means of computation here.

The paper is structured as follows. The reaction-diffusion CA and its phenomenology is defined in section 2. We show how to input information into the automaton in section 3. Section 4 deals with the implementation of memory devices. Possible ways of routing signals are presented in section 5. Nontrivial binary operations implemented in collisions between mobile localizations are studied in section 6. And lastly, in section 7 we construct a finite state machine realized by stationary and mobile localizations.

2. The reaction-diffusion cellular automaton

We design a totalistic CA where a cell updates its state depending on just the number of different cell-states in its neighborhood. Consider a ternary state automaton, where every cell takes one of the following cell-states: substrate S, activator A, or inhibitor I. The update rule can be written as:

$$x^{t+1} = f(\sigma_I(x)^t, \sigma_A(x)^t, \sigma_S(x)^t),$$

where $\sigma_p(x)^t$ is the number of cell x's neighbors with cell-state $p \in \{I, A, S\}$ at time step t. As for all classical CA, cell updates are made synchronously across the whole lattice in discrete time steps. Our CA is based on a two-dimensional lattice with hexagonal tiling. The neighborhood size is seven, containing the central cell and its six closest neighbors.

To give a compact representation of the CA rule, we adopt the formalism in [16], and represent the cell-state transition rule as a matrix $\mathbf{M} = (m_{ij})$, where $0 \le i \le j \le 7$, $0 \le i + j \le 7$, and $m_{ij} \in \{I, A, S\}$. The output state of each neighborhood is given by the row-index i (the number of neighbors in cell-state I) and column-index j (the number of neighbors in cell-state A). We do not have to count the number of neighbors in cell-state S because it is given by $7 - (i + j)$. A cell with a neighborhood represented by indices i and j will update to cell-state M_{ij}, which can be read off the matrix. The cell-state transition function can be presented as $x^{t+1} = M_{\sigma_2(x)^t \sigma_1(x)^t}$.

Here is the exact matrix structure, which corresponds to matrix M_3 in [8]:

$$M = \begin{Bmatrix} S & A & I & A & I & I & I & I \\ S & I & I & A & I & I & I & \\ S & S & I & A & I & I & & \\ S & I & I & A & I & & & \\ S & S & I & A & & & & \\ S & S & I & & & & & \\ S & S & & & & & & \\ S & & & & & & & \end{Bmatrix}.$$

Do these matrix entries correspond to phenomena in reaction-diffusion chemical systems? Indeed they do. Thus, $M_{01} = A$ symbolizes the diffusion of activator A, $M_{11} = I$ represents the suppression of activator A by the inhibitor I, and $M_{z2} = I$ ($z = 0, \ldots, 5$) can be interpreted as self-inhibition of the activator in particular concentrations. $M_{z3} = A$ ($z = 0, \ldots, 4$) means a sustained excitation under particular concentrations of the activator. $M_{z0} = S$ ($z = 1, \ldots, 7$) means that the inhibitor is dissociated in absence of the activator, and that the activator does not diffuse in subthreshold concentrations. And, finally, $M_{zp} = I$, $p \geq 4$ is an upper-threshold self-inhibition.

The cell-state transition rule reflects the nonlinearity of activator-inhibitor interactions for subthreshold concentrations of the activator. Namely, for a small concentration of the inhibitor and for threshold concentrations (values 1 and 3), the activator is suppressed by the inhibitor, while for critical concentrations of the inhibitor (value 2) both inhibitor and activator dissociate producing the substrate, as symbolized in the following set of quasi-chemical reactions:

$$\begin{array}{lrr} A + 6S \to A & A + I \to I & A + 3I \to I \\ & A + 2I \to S & 2A \to I \\ & 3A \to A & \beta A \to I \\ & & I \to S. \end{array}$$

Starting in a random initial configuration the automaton will evolve towards a quasi-stationary configuration, with typically two types of stationary localizations, and a spiral generator of mobile localizations (Figure 1). By analogy with Conway's Game-of-Life we call mobile localizations *gliders*, the generators of mobile localizations *glider-guns*, and stationary localizations (glider) *eaters*. Eaters usually annihilate gliders that collide into their central body, but they can also modify gliders that brush past, interacting with the outer edge. The core of a glider-gun is a discrete analog of a "classical" spiral wave, commonly found in excitable chemical systems like the Belousov–Zhabotinsky reaction (Figure 2). However, at some distance from the spiral wave

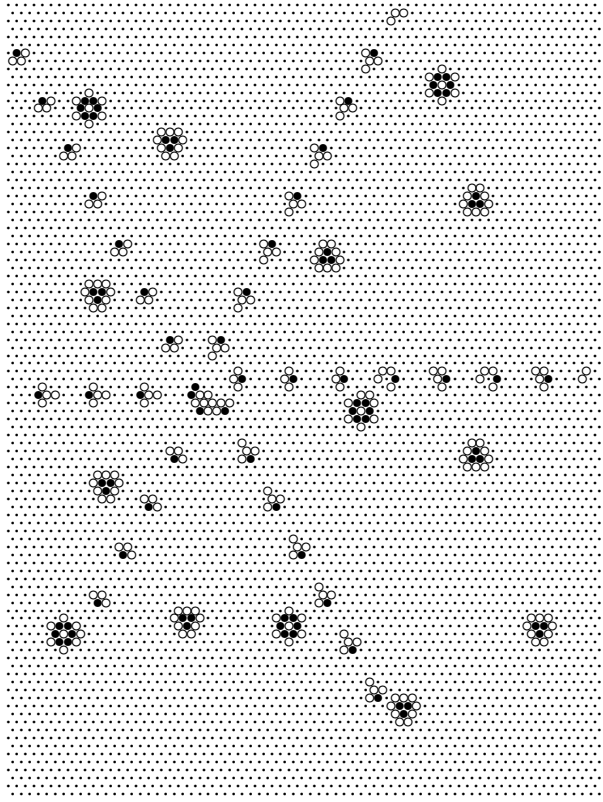


Figure 1. A typical quasi-stable configuration of the CA which started its development in a random initial configuration (with 1/3 probability of each cell-state). Cell-state I (inhibitor) is shown by a black disk, cell-state A (activator) by a circle, and cell-state S (substrate) by a dot. We can see there are two types of stationary localizations (glider eaters) and a spiral glider-gun, which emits six streams of gliders, with a frequency of one glider per six time steps in each glider stream.

tip the wave front becomes unstable and splits into localized wave-fragments. The wave-fragments continue traveling along their originally determined trajectories and keep their same shape and velocity vector unless disturbed by other localizations. So, the wave-fragments behave as in subexcitable Belousov–Zhabotinsky systems [6].

There are five types of basic gliders, those with one (activator) head (Figure 3), which vary by the number of trailing inhibitors. Three types (G_{34}, G_{24}, G_{43}) alternate between two forms. Two types (G_4, G_5) have just one form. The spiral glider-gun in Figures 1 and 2 release G_{34} gliders. An alternative, low frequency, spiral glider-gun [8] (not shown) releases G_4 gliders. These basic gliders, and also a variety of more

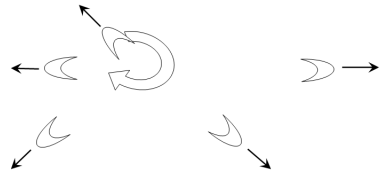


Figure 2. The principle scheme of the glider-gun in "spiral rule" hexagonal CA: the core of the spiral wave rotates clockwise, wave-fragments break off from the tail of the spiral wave and travel in six directions: East, South-East, South-West, West, and North-West; the wave-fragment that will travel North-East has not been generated yet.

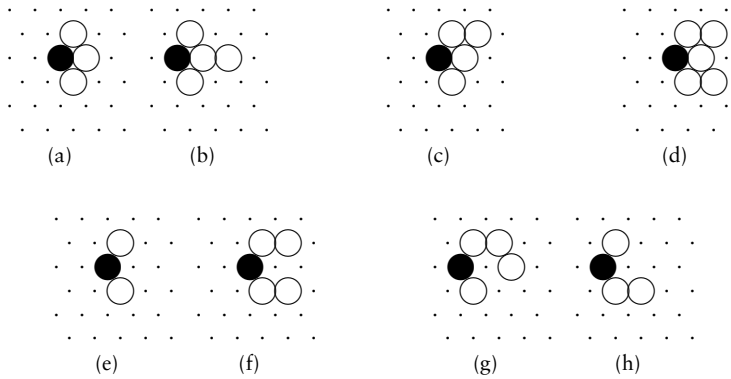


Figure 3. The basic gliders come in five types, shown here traveling West, in the direction of their activator head (cell-state A), with a tail of trailing inhibitors made up of several cell-states I. The glider designator G_{ab} refers to the number of trailing inhibitors: (a) and (b) two forms of glider G_{34}, (c) glider G_4, (d) glider G_5, (e) and (f) two forms of glider G_{24}, (g) and (h) two forms of glider G_{43}.

complicated gliders including mobile glider-guns, are also generated by many other interactions.

The existence of stationary localizations, or eaters, (Figure 4) is yet one more important feature of the CA. Eater E_3 (Figure 4(a)) consists of three activator-states surrounded by nine inhibitor-states. Eater E_6 (Figure 4(b)) has a core of one inhibitor-state surrounded by six activator-states, which in turn are encircled by six (in its minimal symmetric form) inhibitor-states.

We can speculate that our CA is analogous to a combination of two types of chemical systems in one physical space: excitable systems where classical spiral waves are formed, and subexcitable systems where no spiral waves are formed, but only traveling localized

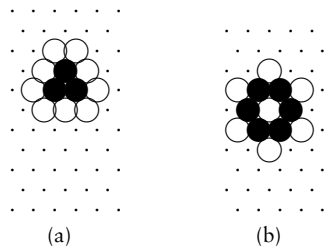


Figure 4. Stationary localizations (eaters): (a) eater E_3, (b) eater E_6.

wave-fragments (assuming space is uniform and homogeneous). Such "hybrid-functionality" systems were never observed experimentally, however there is evidence of complete spiral breakup and a subsequent transition to spatio-temporal chaotic states, for example, reported in [17]. Also, in a modified Barkley model of an excitable reaction-diffusion system, a break-up of a spiral wave far away from the rotating tip was reported in [18]. However, this was achieved in somewhat "artificial" conditions, in which the ratio of time-scales, of the local dynamics of the activator and inhibitor variables, were dynamically changing, increasing during simulation.

3. Input interface

How can we input information into the hexagonal CA computing device? One sensible way to input a quantum of information might be to activate (or inhibit) just one site of the lattice, however such an action can lead to the generation of several gliders (Figure 5) [19], and thus potentially "pollute" the computational space.

What if we try to "stimulate" localizations E_3 and E_6, so they can play the role of stationary sensors, or elements of an input interface?

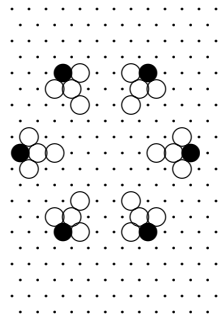


Figure 5. Activation of one site of the lattice leads to formation of six G_{34} gliders.

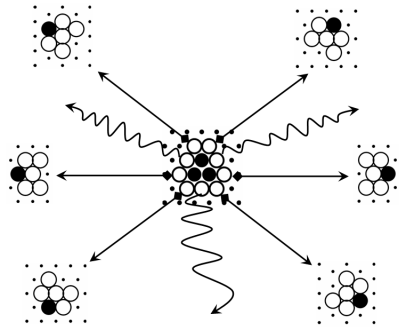


Figure 6. Outcomes of the activation of the inhibitor-sites of the stationary localization (eater) E_3. When an inhibitor-site (marked by the rhomboid end of an arrow) is switched externally to the activator-state, E_3 is transformed into a G_5 glider. The activation of sites marked by the zig-zag arrows leads to the annihilation of E_3.

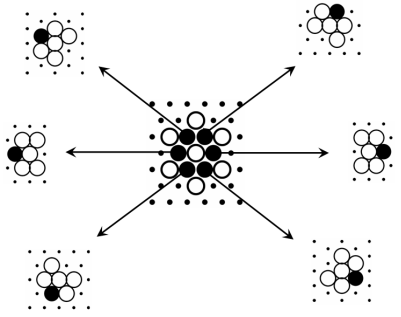


Figure 7. Outcomes of external switching of an activator-site in the stationary localization (eater) E_6 to a resting- or inhibitor-state. In both cases E_6 is transformed into a G_5 glider.

Let us take the eater E_3 and stimulate—switch to the activator-state—one of its inhibitor-sites (Figure 6). The activation of six of the nine inhibitor-sites leads to the transformation of E_3 into a G_5 glider traveling in one of six directions, as shown by the straight arrows in Figure 6. The activation of the other inhibitor-sites (zig-zag arrows in Figure 6) cause the annihilation of E_3.

The inhibition of any of three activator-sites in E_3 will destroy it. E_3 is also stable to the external switching of an inhibitor-state to a resting substrate-state: an inhibitor-site is restored in one time step. However, when one of the activator-sites in E_3 is switched to a substrate-state, the localization E_3 is destroyed.

Eater E_6 is more sensitive to external inputs than E_3. Thus when we switch one of the activator-sites in E_6 to either a substrate- or inhibitor-state, E_6 is transformed into a G_5 glider (Figure 7).

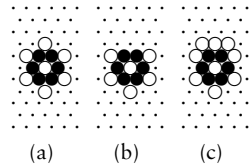


Figure 8. Switching the northern inhibitor-site of eater E_6 to the substrate-state leads to the formation of two more inhibitor-sites in the "inhibitor-shell" of E_6. (a) E_6 in its "normal" form, (b) the northern inhibitor-site is forced to a substrate-state, (c) the configuration of inhibitor-sites is updated.

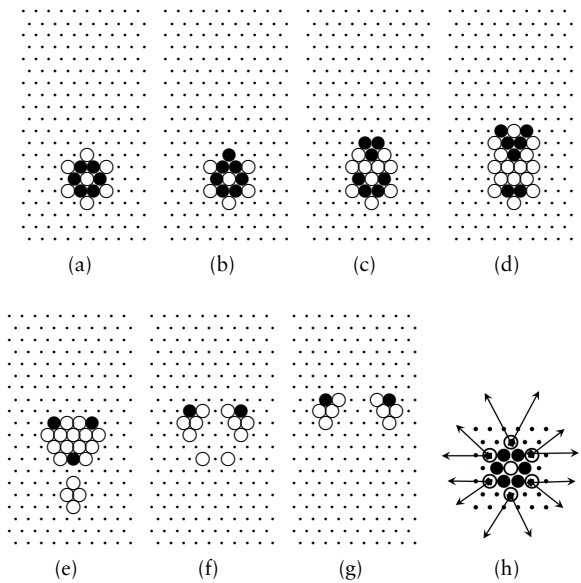


Figure 9. The external switching of one distal inhibitor-site of E_6 to the activator-state transforms E_6 into two G_4 gliders (a) the "normal" form E_6, (b) the northern inhibitor-site is switched to the activator-state, (c) through (f) two G_4 gliders are formed, and (g) travel outward. The velocity vectors of the gliders formed by activating the inhibitor-sites are shown in (h).

The external switching of one of the distal inhibitor-sites in E_6, the northern inhibitor-site in Figure 8(a), to the substrate-state (Figure 8(b)), leads to the recovery of the site and the switching of two neighboring sites to the inhibitor-state (Figure 8(c)). The updated configuration (Figure 8(c)) can be detected by gliders, as shown in later sections.

The activation of one of the outer inhibitor-sites of E_6 transforms the localization into two G_4 gliders, as shown in Figure 9.

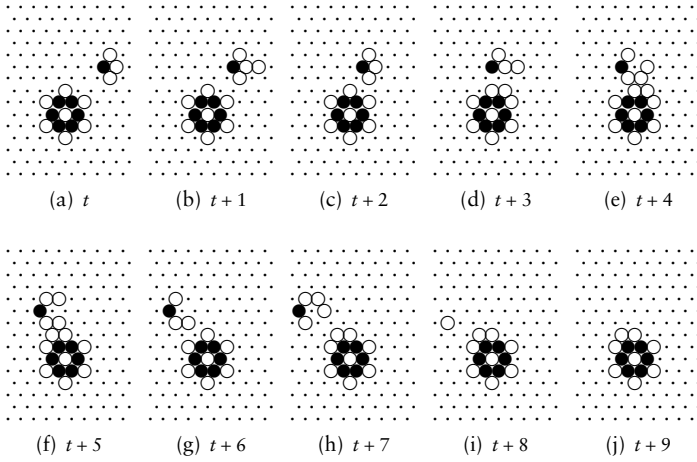


Figure 10. Write bit.

4. Memory device

The eater E_6 can play the role of a 6-bit flip-flop memory device. The substrate-sites (bit-down) between inhibitor-sites (Figure 4) can be switched to an inhibitor-state (bit-up) by a colliding glider.

An example of writing one bit of information to E_6 is shown in Figure 10. Initially E_6 stores no information. We aim to write one bit in the substrate-site between the northern and north-western inhibitor-sites (Figure 10(a)). We generate a glider G_{34} (Figures 10(b) and (c)) traveling West. G_{34} collides with (or brushes past) the North edge of E_6 resulting in G_{34} being transformed into a different type of glider, G_4 (Figures 10(g) and (h)). There is now a record of the collision, evidence that writing was successful. The structure of E_6 now has one site (between the northern and north-western inhibitor-sites) changed to an inhibitor-state (Figure 10(j)), meaning a bit was saved.

To read a bit from the E_6 memory device with one bit-up (Figure 11(a)), we collide (or brush past) with glider G_{34} (Figure 11(b)). Following the collision, the glider G_{34} is transformed into a different type of basic glider, G_4 (Figure 11(g)), and the bit is erased (Figure 11(j)).

5. Routing and tuning signals

To route signals we can potentially employ either stationary localizations (to act as reflectors) or use other gliders to act as mobile reflectors. In practice, we were unable to find a stationary (eater) reflector of gliders; in all cases studied, gliders were either transformed into a different type, or were annihilated, but never changed their trajectory when colliding with an eater. However, mobile reflectors do exist.

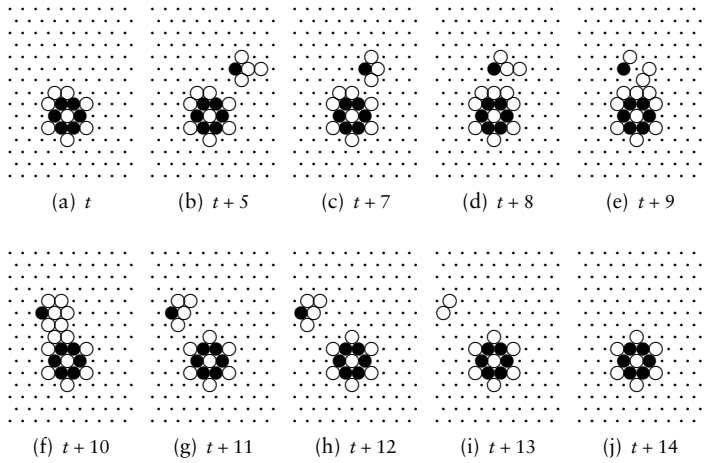


Figure 11. Read and erase bit.

Figure 12 shows how a glider traveling North-West collides with a glider traveling West, and is reflected South-West as a result of the collision. However both gliders are transformed into different types of gliders. This is acceptable on condition that both types of glider represent the same signal, or signal modality.

There are two more gates which though not essential in demonstrating computational universality, are nevertheless useful in designing practical collision-based computational schemes. They are the FANOUT and ERASE gates.

The FANOUT gate is based on glider multiplication. There are a few scenarios where one glider can be multiplied by another glider (for details see the original beehive rule [19], though this does not feature a spiral glider-gun). In Figure 13 we see how a glider moving East collides with another moving West (Figures 13(a) and (b)), four new gliders are formed as a result of the collision (Figure 13(g)), traveling East, West, North-East, and South-West. This is an example of a one-to-three FANOUT gate.

We can make a FANOUT gate by colliding glider G_{34} with glider G_{24}, as shown in Figure 14. Glider G_{34} traveling North-West collides with glider G_{24} traveling West (Figures 14(a) and (b)). The gliders are almost annihilated as a result of the collision—just a tiny fragment, two sites made up of one activator- and one inhibitor-state remain (Figure 14(e)). The activator-inhibitor pair grows into a more complicated pattern (Figures 14(f) and (g)), which finally splits into three G_5 gliders. One glider continues traveling West along the original trajectory of glider G_{24}. Ignoring the fact that the glider types change in the collision, we can assume that both G_{24} and G_5 gliders represent a "control" signal traveling

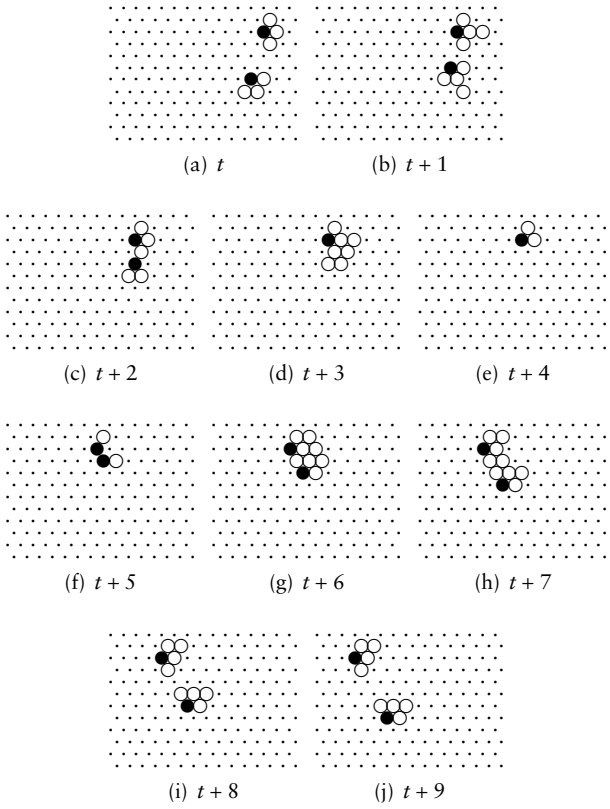


Figure 12. Glider reflection.

West. Two other gliders, the result of multiplication, travel South-West and South-East (Figures 14(h) through (j)), while gliders initially involved in the collision continue along their original trajectories.

To annihilate a glider we can collide it with the central body of an eater, as demonstrated in Figure 15, or with another glider (head-on collisions usually lead to annihilation).

6. Binary operations

The boolean logical universality of the spiral rule CA can be proved using the collision-based computing paradigm [2], where a glider represents the value TRUE, and the absence of a glider represents the value FALSE. When two gliders collide their trajectories may change or new gliders may be generated. A glider emerging on a new trajectory stands for conjunction, the gate AND. So boolean variables can be represented by colliding gliders.

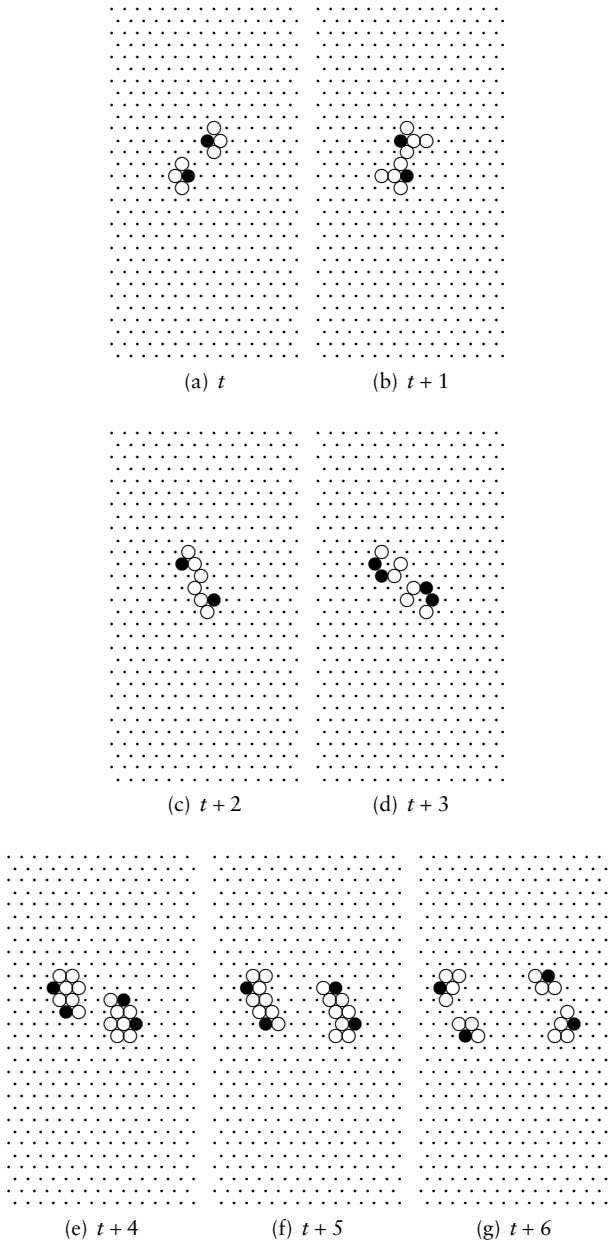


Figure 13. Signal 1-to-3 multiplication, FANOUT gate.

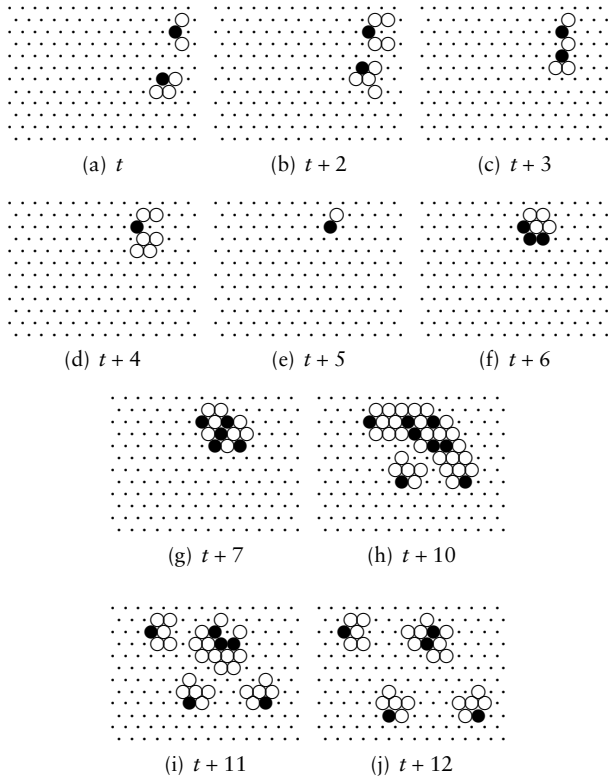


Figure 14. Signal 1-to-2 multiplication, FANOUT gate.

The details of basic logical gates implemented in glider collisions were fully demonstrated in our previous paper [16], so we do not provide any examples here. We should just mention that in contrast to the hexagonal reaction-diffusion CAs studied in [16], the spiral rule CA exhibits stationary glider-guns, or generators of mobile localizations, which are essential in implementing negation. The computing medium represented by the spiral rule CA is fully programmable because, as demonstrated in [8], not only can we generate stationary localizations (eaters) in collisions between gliders, but we can also transform stationary localizations to make generators of mobile localizations (glider-guns), and destroy glider-guns when required.

Conjunction and negation are sufficient to demonstrate the logical functional completeness of the CA. In this section we go a bit further and discuss the implementation of an asynchronous XOR gate and a five-valued binary operation.

The asynchronous XOR gate can be constructed from the memory device in Figures 10 and 11, employing the eater E_6 and the glider G_{34}.

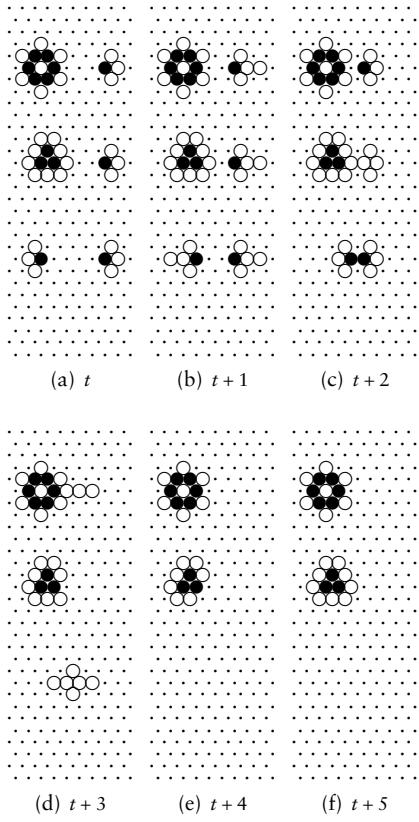
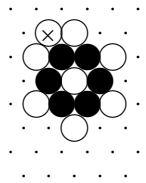


Figure 15. Glider annihilation, ERASE gate, with eaters (top and middle collisions in each subfigure), and another glider (bottom collision).

The incoming trajectory of gliders is an input $x = \langle x, y \rangle$ of the gate, and the state of the cell which is switched to the inhibitor-state by gliders, is an output z of the gate (this cell is shown by $\otimes$ in Figure 16(a)). As seen in Figure 10, when glider G_{34} brushes by the eater E_6 it "adds" one inhibitor-state to the eater configuration (Figure 10, $t + 7$), and transforms itself into glider G_{43}. If glider G_{34} brushes by E_6 with an additional inhibitor-state (Figure 11, t) it "removes" this additional state and transforms itself into glider G_4 (Figure 11, $t + 11$).

Assume that the presence of glider G_{34} symbolizes input logical TRUE and its absence, input FALSE; inhibitor-state I in cell $\otimes$, output TRUE; and substrate-state S, output FALSE. The result of this logical operation can be read directly from the configuration of E_6 or by sending a control glider to brush by E_6 to detect how the glider is transformed (see details of glider transformations in section 7). Then the structure implements exclusive disjunction (Figure 16(b)). The gate constructed is asyn-



(a)

x_1	x_2	y
0	0	S
G_{34}	0	I
0	G_{34}	I
G_{34}	G_{34}	S

(b)

Figure 16. Asynchronous XOR gate. (a) position of output cell is shown by $\otimes$. (b) operation implemented by the gate, input state G_{34} is logical TRUE, output state S is FALSE, and output state I is TRUE.

chronous because the output of the operation does not depend on the time interval between the signals but only on the value of signals: when an inhibitor-state is added or removed from E_6 the configuration of E_6 remains stable and does not change until another glider collides into it.

Interpreting different gliders as states of a multiple-valued logic variable could bring a new dimension to the study of collision-based computing. Multiple-valued gates will be invaluable in designing CA representations of fuzzy reasoning, emotions, and consciousness.

Let us look at just one example of the interpretation, and consider pair-wise collisions involving any two out of four types of glider: G_{34}, G_4, G_5, and G_{24}. One of the pair moves West, the other North-West, positioned before the collision as follows. In this particular example of binary collisions we assume the activator-head (state A) of the glider traveling West is positioned at cell (i, j). Then $(0, 0)$ is a northwest-most corner of the lattice and the activator-head of the glider traveling North-West occupies the cell with coordinates $(i - 1, j + 2)$ (see the example initial position in Figure 12(a)). We assume that the glider traveling West represents the value of variable x, and the glider traveling North-West represents the value of variable y. Following the collision, one new glider continues traveling West (let it represent the value of variable z_1, the result of operation $\odot_1$), another is "reflected" South-West (let it represent the value of variable z_2, the result of operation $\odot_2$). We encode glider G_{34} by the symbol a, G_4 by b, G_5 by c, and G_{24} by d, and the absence of a glider by 0.

Operations realized by this gate are shown in Figure 17. Let us briefly discuss the algebraic systems $A_1 = \langle \odot_1, \{0, a, b, c, d\} \rangle$ and $A_2 = \langle \odot_2, \{0, a, b, c, d\} \rangle$ implemented in the glider collision. Both systems have neither identities nor zeros. The element 0 is the only idempotent ($0 \odot_1 0 = 0$ and $0 \odot_2 0 = 0$). However 0 is right zero in A_1 and A_2 (for any $x \in \{0, a, b, c, d\}$ we have $x \odot_1 0 = 0$ and $x \odot_2 0 = 0$), and left identity in A_1 (for any $x \in \{0, a, b, c, d\}$ we have $0 \odot_1 x = x$). Operations $\odot_1$ and $\odot_2$ are not associative and not commutative. Singleton $\{d\}$ is the

z_1	0	a	b	c	d
0	0	0	0	0	0
a	$\underline{a}$	b	b	a	c
b	$\underline{b}$	a	a	a	b
c	c	b	a	a	c
d	d	b	a	a	$\underline{b}$

(a)

z_2	0	a	b	c	d
0	0	0	0	0	0
a	$\underline{0}$	b	b	a	c
b	$\underline{0}$	a	a	a	b
c	c	b	a	a	c
d	d	b	a	a	$\underline{a}$

(b)

Figure 17. Binary operations realized in a collision between a glider traveling West and a glider traveling North-West. Two different operations are represented by West (a), and South-West (b), output trajectories. Glider G_{34} is represented by a, G_4 by b, G_5 by c, G_{24} by d, and the absence of a glider is 0.

only minimal generator in A_1 and A_2: $d \odot_1 d = b$, $d \odot_1 b = a$, $d \odot_1 a = c$; and $d \odot_2 d = a$, $a \odot_2 a = b$, $a \odot_2 d = c$.

7. Implementation of the finite state machine

While developing the exact construction of a memory device, described in section 4, we discovered that eater E_6 can take four different configurations resulting from the interactions of gliders brushing past, and there are seven types of gliders produced in collisions with the eater (including some basic types flipped). We therefore envisaged that a finite state machine can be implemented in the eater-glider system. The internal state of such a machine is represented by the configuration of the eater, the type of incoming glider symbolizes the input symbol of the machine, and the type of outgoing glider represents the output state of the machine.

To construct the full state transition table of the eater-glider machine we collided seven types of gliders into four configurations of the eater and recorded the results of the collisions. For the sake of compact representation we encoded the configurations of the eater as shown in Figure 18. We denote the gliders as follows: G_{34} as a, G_{43} as b, G_5 as c, G_4 as d, G_{24} as e, G^4 (glider G_4 flipped horizontally) is f, and G^{43} (glider G_{43} flipped horizontally) is g. The state transition table is shown in Figure 19.

Consider the internal states of the eater-glider machine as unary operators on the set $\{a, b, c, d, e, f, g\}$, that is, the machine's state is reset to its initial state after the collision with the glider. For example, the unary operator α implements the following transformation: $a \to b$, $b \to c$, $c \to a$, $d \to a$, $e \to d$, $f \to e$, $g \to e$. The operators have the following limit sets: operator α has the limit set $\{a, b, c\}$, β has set $\{c\}$, χ has two limit sets $\{a, d\}$ and $\{b, c\}$, and operator δ has two limit sets $\{a, b, c, d\}$ and $\{e, f\}$. Considering unary operators $a, \ldots, g$ operating on the set $\{\alpha, \beta, \chi, \delta\}$ we obtain the limit sets shown in Figure 20. Many of the

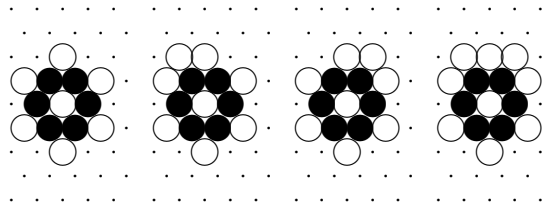


Figure 18. Encoding the internal states of the eater-glider machine in the configuration of eater E_6.

	a	b	c	d	e	f	g
α	βb	δc	αb	αe	δd	αe	δc
β	αd	δe	βc	βc	χg	αa	χe
χ	χd	βe	δf	χa	βb	χa	βe
δ	δb	βc	χg	χe	αf	δe	αa

Figure 19. The state transition table of the eater-glider machine. Tuple xy, a pair made up of an eater state x and glider state y, at the intersection of row i and column j, signifies that being in state i while receiving input j the machine takes state x and generates output y.

Operator	Limit Set
a	$\{\alpha,\beta\},\{\delta\}$
b	$\{\beta,\delta\}$
c	$\{\alpha\},\{\beta\},\{\delta,\chi\}$
d	$\{\alpha\},\{\beta\},\{\chi\}$
e	$\{\alpha,\delta\},\{\beta,\chi\}$
f	$\{\alpha\},\{\chi\},\{\delta\}$
g	$\{\alpha,\delta\},\{\beta,\chi\}$

Figure 20. Limit sets of unary operators $a,\ldots,g$.

operators have more then two limit sets, which may indicate significant computational potential of the eater-glider machine.

To characterize the eater-glider machine in more detail we studied what output strings were generated when the machine received the uniform infinite string s^*, $s \in \{a,\ldots,g\}$ on its input. These input string to output string transformations are shown in Figure 21.

Input string *abcdefg* evokes the following output strings when fed into the machine. The machine starting in state α generates string *begabac*, in state β string *dcgabac*, in state χ string *deccgae*, and in state δ string *bcccgae*.

	a^*	b^*	c^*	d^*	e^*	f^*	g^*
α	$(bd)^*$	$c(ce)^*$	b^*	e^*	$(de)^*$	e^*	$(ca)^*$
β	$(db)^*$	$(ec)^*$	c^*	c^*	$(gb)^*$	ae^*	e^*
χ	d^*	$e(ec)^*$	$(fg)^*$	a^*	$b(gb)^*$	a^*	e^*
δ	b^*	$(ce)^*$	$(gf)^*$	ea^*	$(ed)^*$	e^*	$(ac)^*$

Figure 21. Input string to output string transformations implemented by the eater-glider machine. String s, at the intersection of row i and column j, tells us that being initially in state i and receiving a uniform string j, the machine generates string s.

8. Discussions

We have designed a hexagonal cellular automaton (CA) imitating an abstract spatially-extended three-species chemical system with nontrivial interactions between activator, inhibitor, and substrate. The model we have constructed exhibits significant interactions: a range of traveling and standing quasi-particles, or wave-fragments or gliders, and generators of the traveling patterns.

We proved that all the basic components necessary for constructing a general purpose computing device are implemented in the spatio-temporal dynamics of the automaton. They include signal reflectors, multiplicators, erasers, memory devices, binary and multiple-valued gates, and a finite state machine.

Amongst the problems that remain to be unravelled here are a few of the most important.

- To build a configuration of reusable sensors, that is, restorable eaters. Currently, to input a piece of information to the computing medium, we switch the state of one site of a stationary localization, thus transforming the localization into a mobile localization. The sensor-localization is destroyed as a result of "sensing," which may be inconvenient for certain applications.
- To find ways of using eaters to change the trajectories of signal-gliders. So far, when a glider collides (or brushes past) an eater, the glider either changes its type or is annihilated, but the glider never starts moving along a new trajectory.
- To invent techniques for reading a bit from a memory device (see section 4) without destroying the bit, at present, reading is associated with erasing.

On the experimental front, we are eager to see real-life chemical systems which exhibit behavior similar to that discovered in the present paper, particularly concerning localization dynamics. Some promising results have been obtained already by Vanag and Espstein [20], who demonstrated experimentally the existence of spiral waves emitting localized wave-fragments. However, this was not done in a "conven-

tional" setup of a liquid-phase chemical system, but in a Belousov–Zhabotinsky reaction dispersed in water nanodroplets of a water-in-oil microemulsion. We are not aware of any standing localizations existing in the same physical domain of the medium with travelling localizations. We hope to clarify these matters in future experiments.

References

[1] Adamatzky, A., De Lacy Costello, B., and Asai, T., *Reaction-Diffusion Computers* (Elsevier, 2005).

[2] Adamatzky, A., *Collision Based Computing* (Springer, London, 2002).

[3] Berlekamp, E., Conway, J., and Guy, R., *Winning Ways*, Volume 2 (Academic Press, 1982).

[4] Fredkin, E. and Toffoli, T., "Conservative Logic," *International Journal of Theoretical Physics*, **21** (1982) 219–253.

[5] Margolus, N., "Physics-like Models of Computation," *Physica D*, **10** (1984) 81–95.

[6] I. Sediña-Nadal, E. Mihaliuk, J. Wang, W. Pérez-Muñuzuri, and K. Showalter, "Wave Propagation in Subexcitable Media with Periodically Modulated Excitability," *Physical Review Letters*, **86** (2001) 1646.

[7] Liehr, A. W., Bode, M., and Purwins, H.-G., "The Generation of Dissipative Quasi-Particles near Turing's Bifurcation in Three-Dimensional Reaction-Diffusion-Systems," in *High Performance Computing in Science and Engineering 2000*, Transactions of the High Performance Computing Center, Stuttgart (HLRS) 2000, edited by E. Krause and W. Jager (Hrsg.) (Springer, 2001).

[8] Wuensche, A. and Adamatzky, A., "On Spiral Glider-guns in Hexagonal Cellular Automata: Activator-Inhibitor Paradigm," *International Journal of Modern Physics C*, **17** (2006) in press.

[9] M. Gerhardt, H. Schuster, and J. J. Tyson, "A Cellular Excitable Media," *Physica D*, **46** (1990) 392–415.

[10] Markus, M. and Hess, B., "Isotropic Cellular Automata for Modelling Excitable Media," *Nature*, **347** (1990) 56–58.

[11] D. Young, "A Local Activator-Inhibitor Model of Vertebrate Skin Patterns," *Mathematical Bioscience*, **72** (1984) 51.

[12] S. Yaguma, K. Odagiri, and K. Takatsuka, "Coupled-cellular-automata Study on Stochastic and Pattern-formation Dynamics under Spatiotemporal Fluctuation of Temperature," *Physica D*, **197** (2004) 34–62.

[13] X. Yang, "Pattern Formation in Enzyme Inhibition and Cooperativity with Parallel Cellular Automata," *Parallel Computing*, **30** (2004) 741–751.

[14] X. Yang, "Computational Modelling of Nonlinear Calcium Waves," *Applications of Mathematical Modelling*, **30** (2006) 200–208.

[15] H. Hartman and P. Tamayo, "Reversible Cellular Automata and Chemical Turbulence," *Physica D*, **45** (1990) 293–306.

[16] Adamatzky, A., Wuensche, A., and De Lacy Costello, B., "Glider-based Computation in Reaction-Diffusion Hexagonal Cellular Automata," *Chaos, Solitons, and Fractals*, **27** (2006) 287–295.

[17] Q. Ouyang, H. L. Swinney, and G. Li, "Transition from Spirals to Defect-mediated Turbulence Driven by a Doppler Instability," *Physical Review Letters*, **84** (2000) 1047–1050.

[18] M. Bär and L. Brusch, "Breakup of Spiral Waves Caused by Radial Dynamics: Eckhaus and Finite Wavenumber Instabilities," *New Journal of Physics*, **6** (2004) 5.

[19] Wuensche, A., "Glider Dynamics in 3-value Hexagonal Cellular Automata: The Beehive Rule," *International Journal of Unconventional Computing*, **1** (2005) 375–398.

[20] Vanag, V. K. and Epstein, I. R., "Segmented Spiral Waves in a Reaction-Diffusion System," *Proceedings of the National Academy of Sciences, USA*, **100** (2003) 14635–14638.