Pol. J. Appl. Sci., **2018**, *4*, 91-95

APPROACH FOR DEFINING STOCHASTIC 2D CELLULAR AUTOMATA

Piotr Gnyś

Department of Computer Science
Polish-Japanese Academy of Information Technology, Warsaw, Poland

E-mail: pgnys@pja.edu.pl

Abstract: The aim of this article is to present a way of defining a stochastic cellular automaton that can be considered a generalisation of Conway's *Game of Life*. Due to the non-deterministic nature and the non-binary cell state the proposed approach is able to create many more diverse sets of automatons without losing the ability to define the classical ones.

Key words: stochastic cellular automaton, generalized cellular automaton, stability of cellular automaton, mixed cellular automaton, Game of Life

DOI: 10.34668/PJAS.2018.4.3.02

Cellular automaton: General concepts and terminology

The first concepts that gave the basis for cellular automaton (CA) design were created by Stanisław Ulam in 1962 during his research on crystalline structures [1]. While the model used there met all requirements of cellular automaton, it was John von Neumann that in 1966 proposed a more general way of defining CA [2].

The algebra of CA is given by a triple:

$$A = \{S, t, =\}, \tag{1}$$

where S is the universe containing all possible states of a single cell, t is the transition function and "=" denotes equality relation between two cells. A specific automaton can be described as a pair:

$$C = \{A, M\}, \tag{2}$$

where A is the algebraic system that describes the CA, while M is the cell matrix on which operations are executed. A is usually called a 'rule set' of the automaton, and this term will be used in this article. The symbol An denotes the number of a cycle in which a given state combination will appear and M0 is initial condition of cell matrix.

While a transition function is defined for a single cell, we can use it with a whole cell matrix parameter. In a such case the result of the function will be another cell matrix with all cells transformed according to the transition rules. The result of such a transformation is considered a subsequent cycle so that:

$$t(A_n) = A_{n+1}. \tag{3}$$

Cellular automaton: Conway's *Game of Life*

One of the most famous and well-analysed CA is the Conway's *Game of Life* (CGL) [3]. The CGL is a particular case of the CA which is defined as follows [4]:

1. cells are the elements of the two dimensional grid (d=2);
2. $N(c)$ is set of the neighbour cells for the cell c;
3. the Moore neighbourhood (8-neighbour cells), where neighbours of a cell c are all cells that have a common edge or vertex with c, or alternatively by the von Neumann neighbourhood (4-neighbour cells), where neighbours of a cell c are all cells that have only a common edge with c, can be used;
4. $S = \{0, 1\}$ where the state 0 means that the cell is dead and the state 1 means that the cell is alive;
5. the cell is alive or dead according to the following rules:
   (a) a dead cell with two living neighbours becomes alive;
   (b) a living cell stays alive if it has two or three living neighbours;
   (c) in all other cases the cell is dead;
6. the function f_t determines the state $s(c, t)$ of each cell c in the instant t as follows:

$$s(c, t+1) = f(\{s(c', t) : c' \in N(c)\}) \tag{4}$$

In the CGL one can observe the following features:

1. each cell may live indefinitely, and this produces still life configurations;

2. the life of a cell is determined only by the state of its last neighbourhood;
3. automaton is deterministic;

Proposed generalization

In this section a generalised version of the CGL, called the *Generalised Game of Life* (GGL), is proposed. In particular the following assumptions are made:

- each cell will have a finite and randomly assigned lifespan from the instant of becoming alive; when the lifespan of a cell reaches zero it will be considered dead;
- the life history of each cell is determined by a non-Markovian process and not by the last preceding state only;
- life history of each cell is determined by a non-Markovian process and not by the last preceding state only;
- workings of generalized automaton will be determined by the random transition functions.

To define the properties of an automaton that meet the requirements given above, the following four elements are needed:

$$G \equiv (P_l, P_s, f_a, N), \tag{5}$$

where P_l is the distribution of the probability of lifespan values given to a cell during its spawning, P_s is the distribution of spawning probability in relation to the number of neighbours. By f_a ageing function that reduces a cell's lifespan force is understood, and if the cell life force drops to zero, the cell becomes dead. Distributions P_l,P_s and the function f_a together form the transition function T_a. In addition P_l generates $S_l \equiv \langle L_{min}, L_{max} \rangle$, where L_{min} is the smallest value of the life force that can be given to a new cell, and L_{max} is the largest one. In all the examplespresented here, P_l is adopted as a Gaussian distribution function. Consequently, when describing settings for specific automaton, parameters m_l and σ_l of a Gaussian distribution function to describe initial lifetime distribution is used.

Unlike CGL, a cell can end up in more than two states because of possible distinct values of life force L. This does not concern functions P_l, P_s, f_a as they check only whether the cell is dead or alive, in other words, whether the life force of cell equals zero or not.

Conway's *Game of Life as a particular case of our generalization*

As the automaton defined in this paper in a generalisation of a CGL it should be possible to recreate CGL as special case of GGL. To get an automaton that behaves like Conway Game of Life, one has to assign $P_l = 1$ which leads to the set of states $S = (0, 1)$. As for spawning distribution and ageing function we have to configure them as presented in Table 1. Setting P_l to 1 to 1 reduces possible

Table 1: Configuration of the Conway's *Game of Life*

n	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Ps	0	0	0	1	0	0	0	0	0
fa	1	1	0	0	1	1	1	1	1

states to only two; hence, setting values of P_s and f_a to only 1 or zero changes the automaton into a deterministic one. When governed by those rules our automaton, it will behave exactly like Conway's *Game of Life*.

Stochastic automaton

One of main advantages of GGL over CGL is its ability to create a stochastic automaton within the same rule set. An example of such an automaton is one that exhibits simple chain reaction like behaviour.

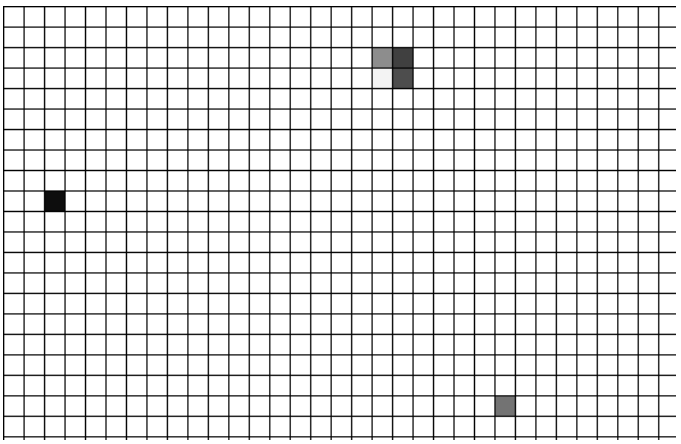

Fig. 1: Automaton before the chain reaction started.

To create such an automaton one has once again to set life energy distribution to constant $P_l = 20$. However, this time spawn the probability can have values other than 0 or 1, see Table 2.

Table 2: Configuration of the wavefront automaton.

n	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Ps	0.0001	0.001	0.01	0.3	0.5	0	0	0	0
fa	1	1	1	1	1	1	1	1	1

It can be concluded from this configuration that not only the resulting automaton is stochastic, but also, unlike Conway's, it cannot have immortal cells (no still life forming) and is not stable in a void state as it is possible for cells to spontaneously rejuvenate even without any neighbours. The resulting automaton will generate a cell matrix

where during each cycle a small number of cells will come to life. Because all cells have a limited lifetime, this process will be countered by the death of cells that were rejuvenated 20 cycles before. Until, by chance, enough cells spawn close to one another, the cell matrix will be filled with living cells scattered sparsely along it. However, when a larger group of cells forms, then it becomes the epicentre of a chain reaction. While cells belonging to the epicentre will die just like all other cells, the structure of epicentre will not disappear, instead it will replicate itself becoming in time an oscillator. Besides the oscillating behaviour, it will also create rings, like structures moving away from it in a wavefront pattern similar to waves on surface of water after a rock has been thrown into it, see Fig. 2. This behaviour may be

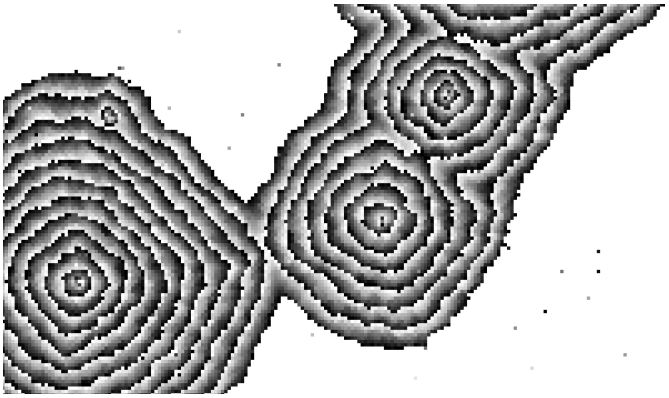

Fig. 2: Automaton with multiple epicentres.

considered similar to Conway's glider guns as after covering whole matrix with cells that were rejuvenated as result of epicentre oscillations, the automaton begins to behave in a deterministic way.

Stability of stochastic and semi-deterministic automaton: Stochastic automaton

One of most interesting properties of CGL is the ability to create stable cell formations. They can be divided into the following categories:

- still life – patterns that remain unchanged;
- oscillators – patterns that oscillate between two or more configurations;

It should be noted that other patterns like spaceships (patterns that move through cell array) or guns (oscillators that generate spaceships) can be stable depending on border conditions. However, in this experiment we will only focus on those two that are not influenced by border conditions as we want to limit the number of parameters to a minimum. The blinker will be analysed in more detail now.

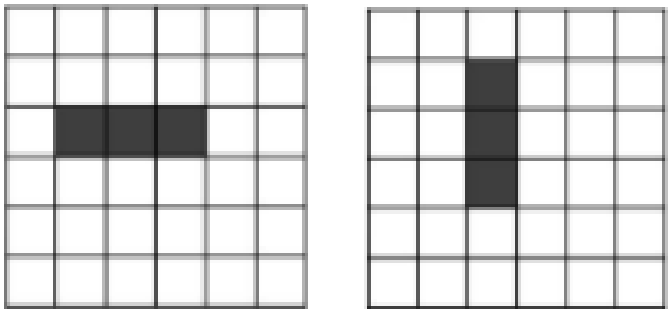

Fig. 3: Blinker pattern as an example of an oscillator.

When referring to cells by numbers value 0 indicates top left cells and values will increment towards the right (see Table 3).

Table 3: Cell array.

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19
20	21	22	23	24

If automaton start with the state $S_0 = [7, 12, 17]$, then in case of the deterministic CGL its state diagram will look as follows, see Fig. 4.

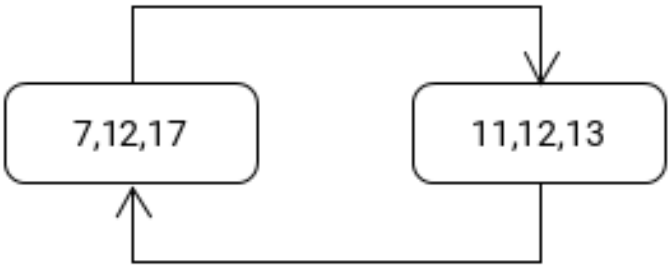

Fig. 4: States of the deterministic blinker automaton

However, after changing the spawning probability in such a way that instead of deterministic values it will take the ones shown in Table 4, then as a result, the number of possible states rises. With six cells that can end up

Table 4: Probabilities of the cell rejuvenation for *Conway's Game of Life* with scale 0.41.

n	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Ps	0	0	0.015	0.97	0.015	0	0	0	0

in one of two stateswe have $2^6 = 64$ possible outcomes for just the first transition. It means that probability to

create a state that corresponds to the blinker structure is $P_B = 0.97 * (1 - 0.015) * (1 - 0.015) = 0,941$. As the second state of blinker has exactly the same relative cell distribution as the first one, its probabilities are exactly the same. Due to that one can describe the probability of blinker remaining stable for n cycles as $P_{b(n)} = 0.941^n$. Because this function is exponential in the cycle number, the probability of sustained stability drops quickly, see Fig. 5. Due to this

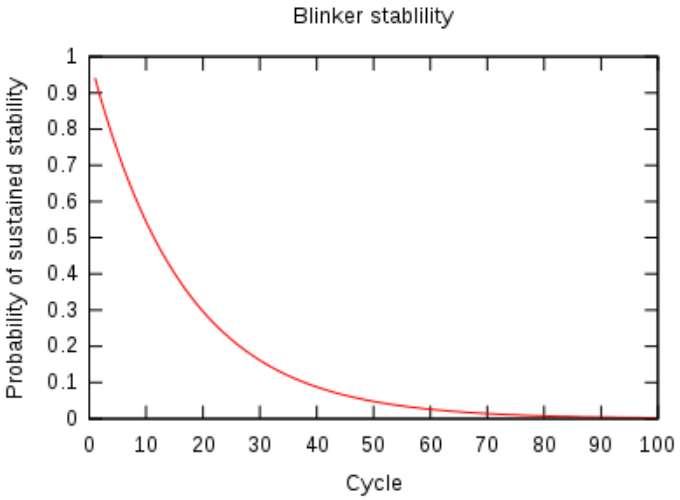

Fig. 5: Stability of the stochastic blinker.

fact, it is impossible to create long lasting stable patterns with stochastic cellular automaton. One of solutions to this problem is provided by mixed automaton that have a group of states that behave in a stochastic manner and other that behave in a deterministic one.

Stability of stochastic and semi-deterministic automaton: Mixed automaton

The automaton from Table 4 is also of the mixed type although it does not display any interesting behaviour. This happens because its deterministic states are either stable void or are states that lead directly to it. The definition of another automaton CA is proposed; this time its deterministic states as well as ageing function are inspired by CGL. However, unlike in CGL we will set all remaining spawning probabilities to non-zero values, and L will be set to 5 instead of 1, see Table 5. With those settings we get

Table 5: Configuration of the mixed automaton.

n	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Ps	0.05	0.1	0.2	1.0	0.2	0.1	0.05	0.025	0.0125
fa	1	1	1	1	1	1	1	1	1

an automaton that starting from void state will stabilize around activity level 0.6, see Fig. 6. Visual analysis of this automaton shows behaviour patterns similar to white noise, see Fig. 7.

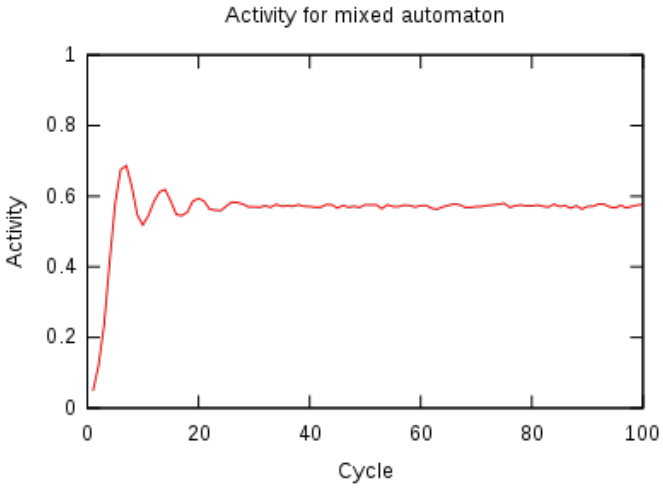

Fig. 6: Activity of the mixed automaton.

If we feed the cell array generated by the previous example into fully deterministic automaton with Conway like settings (except for maximum life energy which remains at the level $L = 5$), then the activity level of automaton will not change drastically, see Fig. 8. However, upon visual inspection we can clearly see that cell array display more regular maze like pattern. By looking at the given examples and at

Fig. 7: Visualization of the mixed automaton.

wavefront automaton, two types of mixed automaton can be identified.

- an automaton with high spawning probabilities for stochastic states. Because of that they maintain sto-

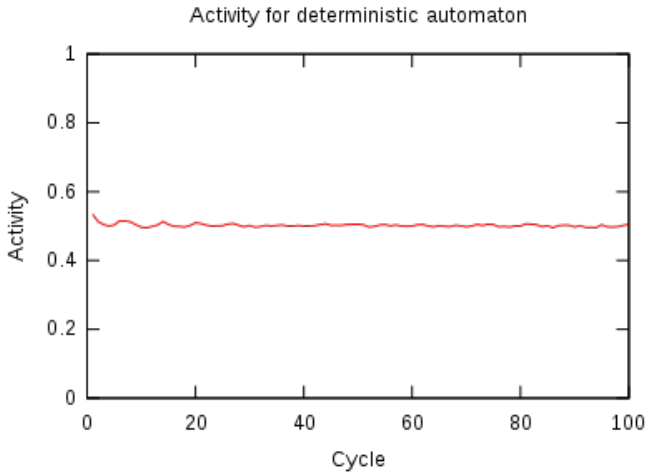

Fig. 8: Activity of deterministic automaton feed with array generated by stochastic one.

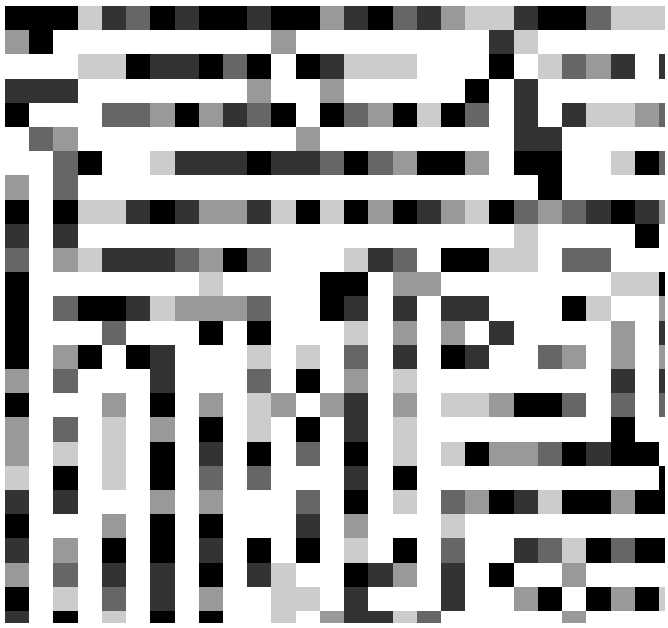

Fig. 9: Mixed automaton visualization.

chastic nature even while deterministic states appear in the cell array.

- an automaton with low spawning probabilities for stochastic states. In that case once the deterministic states start appearing frequently enough the automaton will lose its stochastic nature as spawning in non-deterministic states appearing too infrequently to influence the automaton as a whole.

This nature of mixed automata means that it is possible to generate different results while starting from void state. This is useful for example in procedural level generation [5] for computer games where designers want to hardcode as little initial information and retrieve a large variety of results.

Literature

[1] Ulam S. On some mathematical problems connected with patterns of growth of figures. In *The Proceedings of Symposia in Applied Mathematics*, pages 215–224, 1962.

[2] Neumann J., Burks A. *Theory of self-reproducing automata.* Urbana, University of Illinois Press, 1966.

[3] Wolfram S. *A new kind of science.* 2002.

[4] Gardner M. Mathematical games – the fantastic combinations of John Conway's new solitaire game "life". *Scientific American*, 1970.

[5] Johnson J., Yannakakis G.N., Togelius J. Cellular automata for real-time generation of infinite cave levels. In *Proceedings of the 2010 Workshop on Procedural Content Generation in Games – PCGames '10*, 2010.

Received: 2018
Accepted: 2018