The Effects of Randomness in Asynchronous 1D Cellular Automata

Yasusi Kanada
Tsukuba Research Center, Real World Computing Partnership
Takezono 1-6-1, Tsukuba, Ibaraki 305, Japan, *kanada@trc.rwcp.or.jp*

Cellular automata are used as models of emergent computation and artificial life. They are usually simulated under synchronous and deterministic conditions. Thus, they are evolved without existence of noise, i.e., fluctuation or randomness. However, noise is unavoidable in real world. The objective of the present paper is to show the following two effects and several other effects caused by existence or nonexistence of randomness in the computation order in one-dimensional asynchronous cellular automata (1D-ACA) experimentally. One major effect is that certain properties of two-neighbor 1D-ACA are fully expressed in their patterns if certain level of randomness exists, though they are only partially expressed if no randomness exists. The patterns generated by 1D-ACA may have characteristics, such as mortality of domains of 1's or splitting domains of 0's into two. These characteristics, which are coded in the "chromosome" of the automata, i.e., the look-up table, are fully expressed only when the computation order is random. The other major effect, which is ommitted from this summary, is that phantom phenomena, which almost never occurs in real world, sometimes occur when there is no noise. The characteristics of patterns generated by several 1D-ACA are drastically changed from uniform patterns to patterns with multiple or chaotic phases when only low level of noise is added.

1. Definition of 1D-ACA

The state of i-th cell at time t is binary and expressed as $s(i; t)$. The initial states of cells, $s(i; 0)$, are given, and the state of a cell is computed from its previous state and two neighbor cells (i.e., $r = 1$) using the following rule.

$$s(i;t) = f(s(i-1;t-1),\ s(i;t-1),\ s(i+1;t-1)) \quad \text{when } i = i_C(t), \text{ and}$$
$$s(i;t) = s(i;t-1) \quad \text{when } i \neq i_C(t),$$
where $s(-1; t) = s(N-1; t)$ and $s(N; t) = s(0; t)$ (periodic boundary condition holds. $i \geq 0$, $t \geq 1$).

State transitions are sequential, opposed to [Wol 84], but similar to [Ing 84]. The order of computation, i.e., sequence $i_C(1)$, $i_C(2)$, … is defined by one of the following methods.

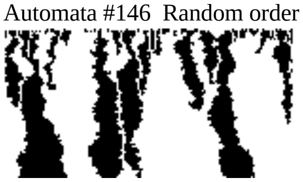
(1) *Random order*: The elements of the sequence is uniform random numbers between 0 and $N-1$.

(2) *Interlaced order*: $i_C(t) = Ct \bmod N$, where C is an integral parameter and prime to N ($gcd(C, N) = 1$).

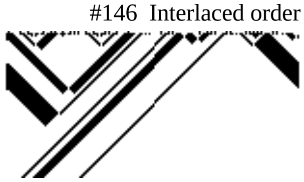
Function f is defined using eight parameters or look-up table elements, i.e., the values of $f_0 = f(0, 0, 0)$, $f_1 = f(0, 0, 1)$, $f_2 = f(0, 1, 0)$, …, $f_7 = f(1, 1, 1)$. This table can be regarded as a set of genes, or a *chromosome*. The genes determine the behavior of the automaton. An automaton is identified using binary number $f_7 f_6 f_5 f_4 f_3 f_2 f_1 f_0$.

2. Examples of Generated Patterns

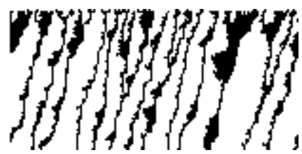
Automata #146 Random order



#146 Interlaced order

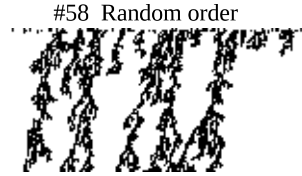


#166 Random order

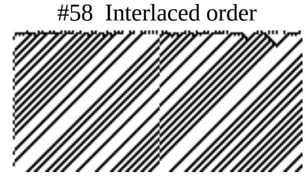


#166 Interlaced order

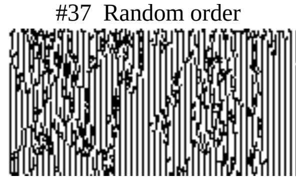
#58 Random order



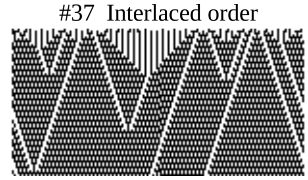
#58 Interlaced order



#37 Random order



#37 Interlaced order



3. Interpretation of the Chromosome and Patterns

Some characteristics of the patterns shown in the previous section can be explained by the chromosomes of the automata. The chromosome, or the look-up table, contains eight genes, $f_0, f_1, \ldots, f_7$, each of which is one-bit length. These genes can be interpreted as shown in the figure below.

[Ing 84] Ingerson, T. E., and Buvel, R. L.: Structure in Asynchronous Cellular Automata, *Physica D*, 10, 59–68, 1984.

[Wol 84] Wolfram, S.: Universality and Complexity in Cellular Automata, *Physica D*, 10, 1–35, 1984.

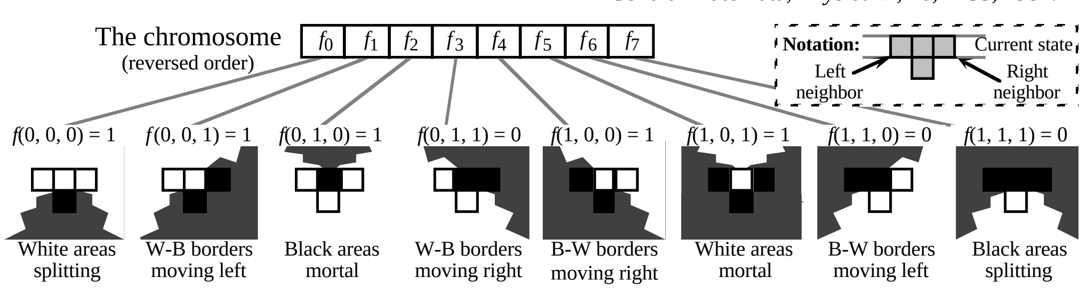


Figure: Interpretation of the chromosome