Asynchronous 1D Cellular Automata and the Effects of Fluctuation and Randomness

— Extended version —

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

Abstract

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 target of the present paper is to show the following two effects and several other phenomena caused by existence or nonexistence of noise in the computation order in one-dimensional asynchronous cellular automata (1D-ACA) experimentally. One major effect is that certain properties of 2-neighbor 1D-ACA are fully expressed in their patterns if certain level of noise exists, though they are only partially expressed if no noise 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 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. Another observed phenomenon is that randomized 1D-ACA generates patterns that are similar to those generated by coupled map lattices (CMLs). This phenomenon suggests that the chaos built into CMLs works similarly to random numbers in 1D-ACA.

1. Introduction

Artificial life [Lan 89, 91, 93] is a research domain, in which biological-life-like emergent behavior of complex systems in real world is studied. Fluctuation or randomness, or noise, is unavoidable and sometimes plays positive roles in real world. Living objects, as autopoietic systems [Mat 79], continuously receive environmental noise. Thus, I believe, the main target of artificial life is to reveal complex system behavior under environmental noise. However, cellular automata, which are often used as models of artificial life, are usually experienced under deterministic and noise-less environment. Although cellular automata with local or internal randomness, i.e., with probabilistic state transitions, have been studied [Vic 89, Kau 84], effects of non-local or environmental noise have not been deeply studied. Ingerson and Bavel [Ing 84] compares patterns of synchronous automata and those of randomized asynchronous automata. However, their observation and discussion is limited, and they do not distinguish the effects of randomness from those of asynchronism.

Prigogine [Pri 77] and Haken [Hak 78] pointed out that both determinacy and nondeterminacy are necessary for generating behavior of complex continuous physical systems, such as Bénard convection or Belusov-Zhabotinsky reaction system. I intend to show that this statement is also true in artificial life or complex discrete computing systems. Emergent computation without randomness in the computation process, such as that in synchronous cellular automata, sometimes causes phantoms, which are phenomena that are so fragile that they almost never occur in real world where noise or randomness exists. Such a case may easily occur in a continuous dynamical system, as illustrated in **Figure 1**. If the potential of a system takes the minimal value, the system is stable. If the potential takes a value between the minimal and maximal value, the state of the system changes. However, if the potential takes the maximal value and no noise exists or the temperature is zero, the system continues to stay in the unstable state. This does not happen in real world. Emergent computation without randomness may also fail to show their important features, which almost always occur in real world, in their behavior.

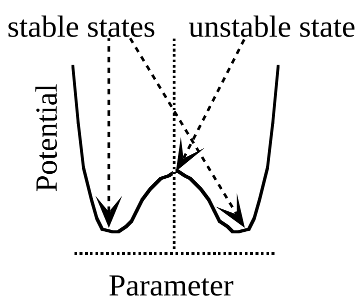


Figure 1. Bifurcation and unstable equilibrium state

The type of CA, which is most widely used, is synchronous deterministic CA (e.g., [Wol 84]) as mentioned above. In this type of automata, randomness exists only in the initial state. However, there are two types of CA, in which randomness is introduced to their computation processes. The first type is CA with probabilistic (or randomized) state transitions (e.g., [Vic 89]). In this type of automata, the major source of randomness except the initial state lies *in* each cell. Thus, they can be called CA with local (or internal) noise. This type of automata has also been widely studied, because they are useful for simulating natural phenomena.[1] The second type is random asynchronous CA [Ing 84, Lub 87, Hof 87]. In this type of automata, the major source of randomness lies in the order of state transitions of the cells, and thus, the noise is non-local or environmental to each cell.[2] Ingerson and Buvel [Ing 84] compared patterns of synchronous automata and those of two types of random asynchronous automata, and found several interesting features of the latters. However, their observation and discussion is limited, and they do not distinguish the effects of randomness from those of asynchronism. Effects of non-local or environmental noise have not been deeply studied.

… [Ber 94], [Lum 94] …

The objective of the present paper is to show examples that support the above statements. A definition of one-dimensional asynchronous cellular automata (1D-ACA) is given in Section 2. Example patterns generated by 1D-ACA, in which the order of computation is randomized, and those generated by deterministic ones are compared in Section 3. The look-up table of 1D-ACA is interpreted and their properties, which are fully expressed only when stronger noise exists, is argued in Section 4. Very noise-sensitive (thus, phantom) patterns generated by 1D-ACA are discussed in Section 5. Conclusions are given in Section 6.

2. Definition of 1D Asynchronous Cellular Automata

Wolfram [Wol 84] analyzes a type of one-dimensional cellular automata. Time is discrete in this model. The state of each cell is determined by the states of itself and the two neighbor cells in the previous time step. Wolfram's model is synchronous, as same as most other models. The states of all the cells are changed at the same time. However, an asynchronous model, which is called 1D-ACA, is used in the present paper. Many variations of asynchronous cellular automata can be devised, but a basically sequential model, which has similarity to the asynchronous Hopfield neural networks, is used in the present paper because of simplicity.

1D-ACA is defined as follows. The state of i-th cell at time t is binary and expressed as $s(i; t)$ (i.e., $k = 2$ and $s(i; t) \in \{0, 1\}$). The initial states of cells, $s(i; 0)$, are given, and the state of a cell is computed from the previous states of itself and two neighbor cells (i.e., $r = 1$) using the following rule when $t > 0$ (where function f is mentioned later).

[1] However, this type of CA is probably not very useful for engineering or designing purpose, because this type of randomness does not reveal the microscopic structures of systems, which are relevant to design. Understanding is not enough for making.

[2] This type of CA is suitable for engineering or designing purpose. They are more suitable for large-scale parallel computers than the first type, which requires global synchronization.

$$s(i; t) = f(s(i-1; t-1), s(i; t-1), s(i+1; t-1)) \quad \text{when } i = i_C(t), \text{ and}$$
$$s(i; t) = s(i; t-1) \quad \text{when } i \neq i_C(t),$$

where $s(-1; t) = s(N-1; t)$ and $s(N; t) = s(0; t)$
(periodic boundary condition holds.)
($i = 0, 1, \ldots, N-1$, and $t = 1, 2, \ldots$).

State transitions are sequential, i.e., for each time step t, there is only one value of i (= $i_C(t)$) where the value of $s(i; t)$ is updated. The order of computation, i.e., sequence $i_C(1)$, $i_C(2)$, … is defined by one of the following three methods.

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

(2) *Fixed random order*: The first N elements of the sequence is uniform random numbers, and these values are repeated in the sequence. Thus, the sequence is periodic. This order is used only in Section 5.

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

Interlaced orders have the maximum possible parallelism when $C = \lfloor (N-1)/2 \rfloor$ or $\lceil (N+1)/2 \rceil$. An example of interlaced orders is shown in **Figure 2**. Cells 0, 3 and 6 can be computed in parallel (i.e., the parallel computation makes no difference in the results), and cells 1, 4 and 7 can be computed in parallel in the next step, and so on.

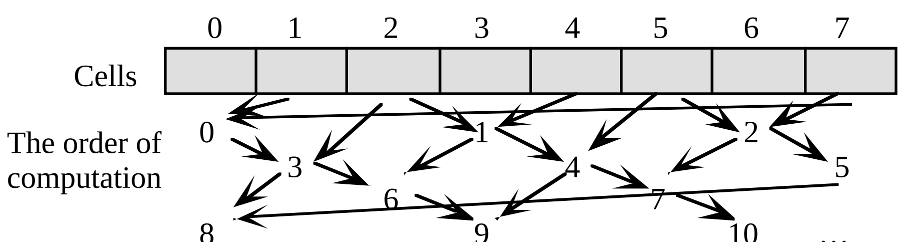


Figure 2. An example of interlaced orders (N = 8, C = 3)

Function f is defined using eight parameters or look-up table elements, 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, or the patterns generated by the automata. An automaton is identified using binary number $f_7\, f_6\, f_5\, f_4\, f_3\, f_2\, f_1\, f_0$ [Wol 84]. For example, the identifier is #3 (in decimal) if the table elements are 1, 1, 0, 0, 0, 0, 0, 0.

3. Patterns Generated by the Automata

Examples of spatiotemporal patterns generated by 1D-ACA are shown in the present section. They are classified into mostly noise-insensitive patterns, fluctuated patterns, merging and/or splitting patterns, and chaotic or partially chaotic patterns. Many new observation results, which Ingerson and Bavel [Ing 84] do not mention, are included.

3.1 Mostly noise-insensitive patterns

Mostly noise-insensitive patterns generated by 1D-ACA are shown and explained here. Patterns generated by automata #32, #160 and #100 are shown for example in **Figure 3** to **Figure 5**. Black means 1, and white means 0 in these figures. The number of cells is 152. The cells are arrayed horizontally. Time goes in downward direction. Figure 5 and the following figures show range $0 \le t < 152 \cdot 152 / 2$. The order of computation is random, or interlaced with $C = 75$. These conditions are the same for following examples too. Both patterns using random orders and those using an interlaced order are shown for each automaton. The results can be summarized as follows.

#32: Patterns A1 and A2, shown in Figure 3, are generated by automata #32. Pattern A1 is generated using a random order, and pattern A2 is generated using an interlaced order. These automata generate patterns that die out almost immediately These patterns are quite similar to those generated by synchronous automata classified to Class I (homogeneous) by Wolfram [Wol 84]. However, the randomized automaton generates longer patterns.

A1. Random order A2. Interlaced order

Figure 3. Patterns generated by automata #32 (= 00100000_2)

#160: [...]

Random order Interlaced order

Figure 4. Patterns generated by automata #160 (= 10100000_2)

#100: There are no significant differences between both patterns, B1 and B2 (shown in Figure 5), generated by automata #100. However, the randomized automaton generates longer transient patterns. These patterns are similar to those generated by synchronous automata of Class II (periodic) [Wol 84].

B1. Random order B2. Interlaced order

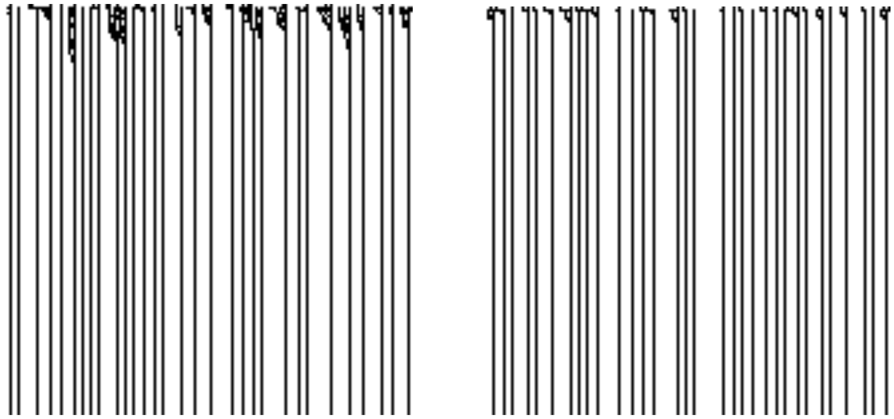


Figure 5. Patterns generated by automata #100 (= 01100100_2)

3.2 Fluctuated patterns

Patterns, which are fluctuated by randomization but whose major characteristics are preserved, are shown and their properties, such as their life-time, are argued here. Patterns generated by automata #226, #146 and #22 (See [Ing 84]) are shown for example in **Figure 6** to **Figure 11**.

#152: The properties of the both patterns shown in Figure 6 are similar, though the shape is quite different. Each black domain (domain of ones) grows first and then shrinks and dies. The final state is uniformly white (zero).

Random order Interlaced order

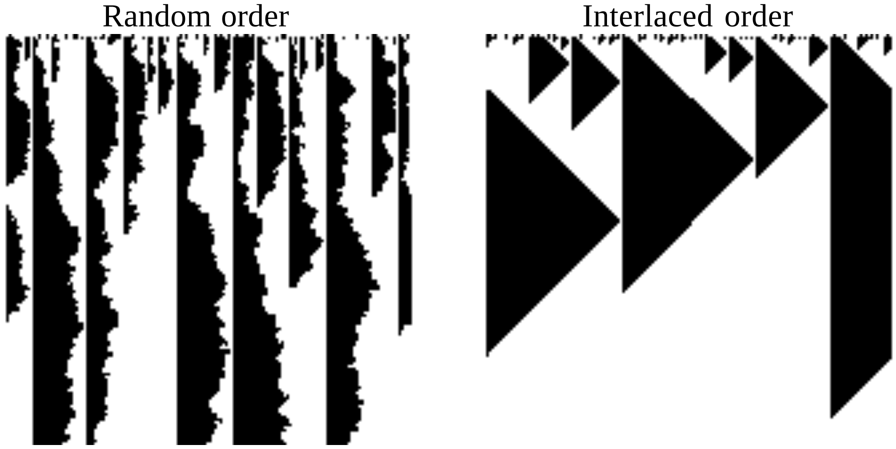


Figure 6. Patterns generated by automata #152 (= 10010110_2)

#198: [**…**] The difference between the patterns generated by automata #198 and #100, which are shown in Figure 7, is that white-to-black boundaries, the boundaries that their left side is white and their right side is black, can move both left and right in #198, but they cannot in #100. The difference between #198 and #150 is that black-to-white boundaries cannot move in #198, but they can move both left and right in #150.

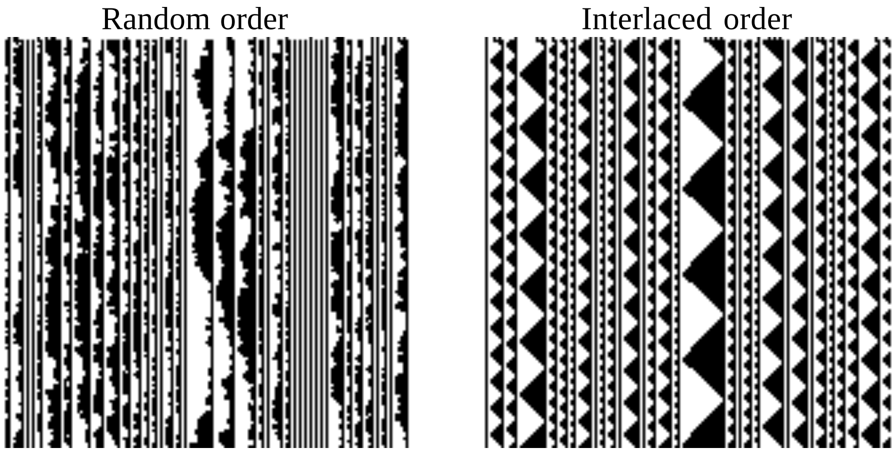


Figure 7. Patterns generated by automata #198 (= 11000110_2)

#150: The number of black domains (and thus, the number of white domains) are invariant in time in the patterns shown in Figure 8. This is the same as in #100. However, the boundaries go left and right in #150, but it does not move in #100. The major difference between the random and deterministic cases are that waves go left and those go right can be seen in latter, but they cannot be seen in former.

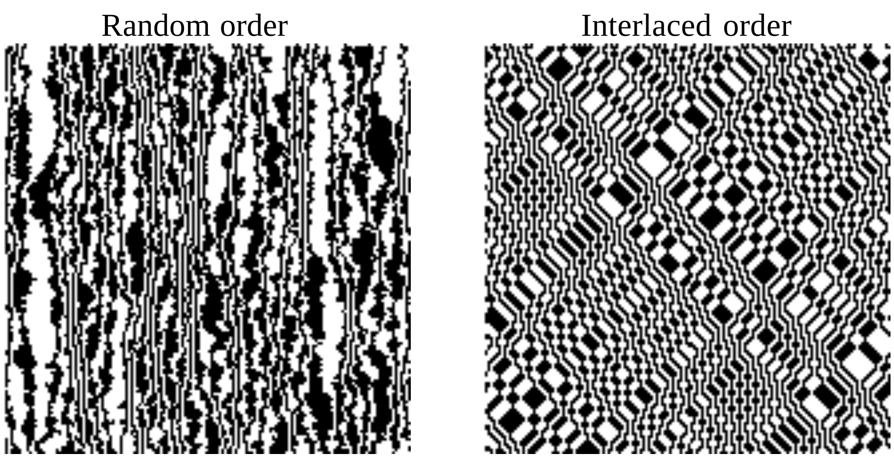


Figure 8. Patterns generated by automata #150 (= 10010110_2)

#226: Patterns C1 and C2, which are shown in Figure 9, are generated by automata #226. The shapes of black or white domains in pattern C1 and those in pattern C2 are quite different. However, these patterns have the same characteristic. Many black domains (domain of 1's) grow first, then shrink and die in both patterns. However, the white-to-black borders, i.e., the borders whose left side is 0 and right side is 1, move like Brownian particles in the random case. The randomized automaton generates longer transient patterns. The final state of the interlaced case is determined solely by the initial state (by fate), but that of the random case is determined by both the initial state and the random numbers.

C1. Random order C2. Interlaced order

Figure 9. Patterns generated by automata #226 (= 11100010_2)

#146: Both patterns, D1 and D2 shown in Figure 10, are different in several points. First, black domains are mortal in the random case, i.e., the final state is always uniformly white (0's). However, black domains or the slanting lines exist forever in the interlaced case, unless those move right and those move left exactly *cross out* each other. Second, a border between a black domain and a white domain goes left or goes right at a constant speed in the interlaced case, but one can move left or right freely, depending on the order of computation, in the random case.

D1. Random order D2. Interlaced order

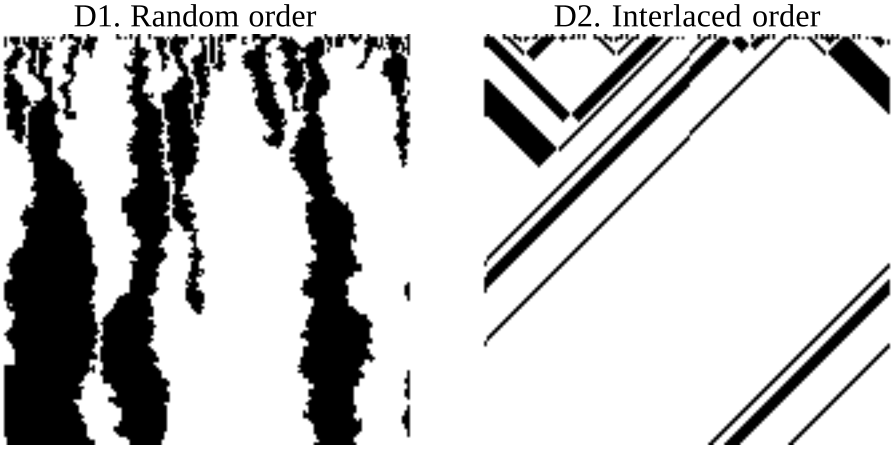


Figure 10. Patterns generated by automata #146 (= 10010010_2)

#22: Both patterns, E1 and E2 shown in Figure 11, generated by automata #22 have stripes as their background. Particles or lattice defects moving left or right can be seen in both cases. Particles can move in both directions, such as Brownian particles, in the random case. This pattern is very similar to a pattern generated by a (deterministic) CML (coupled map lattice) in "diffusion of defect" phase [Kan 89]. Particles can move either left or right in the interlaced case. If two particles crash, they always seem to disappear in the random case, but they often cross or reflect each other in the interlaced case. Particles can also crash and disappear in the latter case, but this type of reaction can occur only in the early stage.

E1. Random order E2. Interlaced order

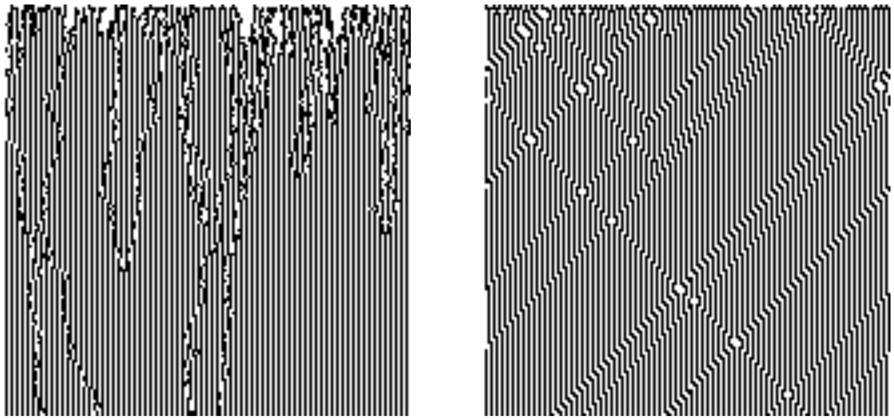


Figure 11. Patterns generated by automata #22 (= 00010110_2)

Mask pattern E-1’. E-1 masked

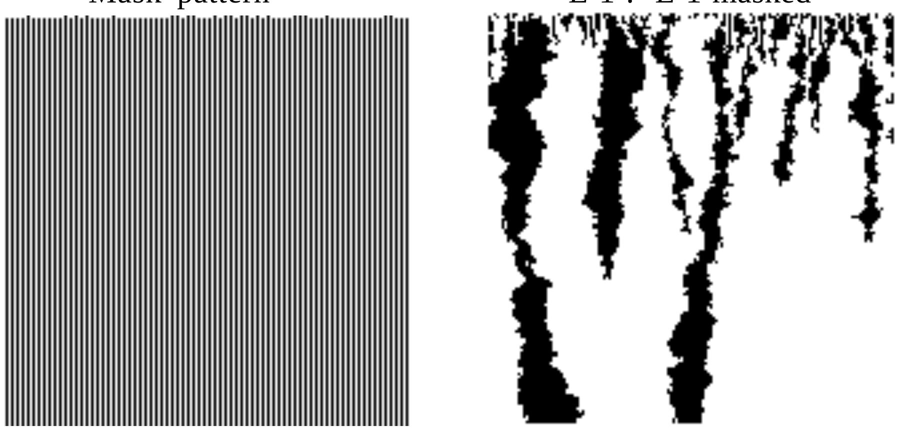


Figure 12. A mask and a masked pattern by #22

[**Lifetime analysis of #226 and #146, Figures 7A & 7B**]

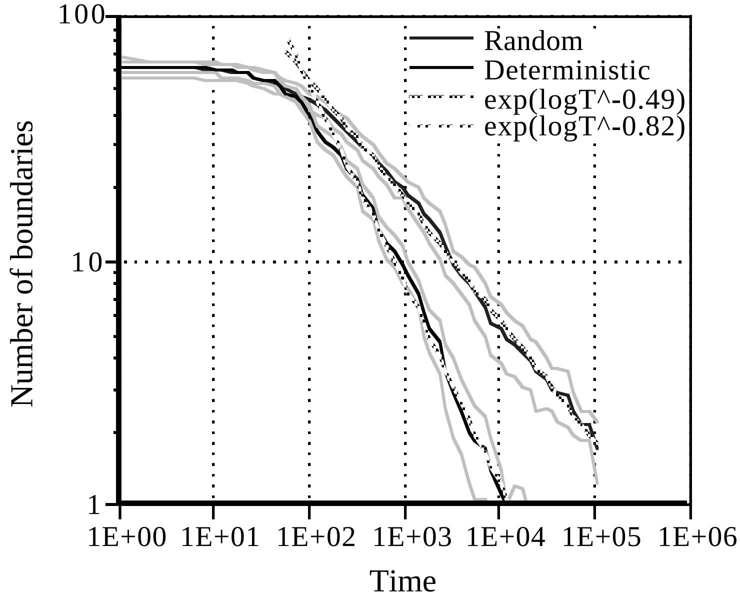


Figure 13. Lifetime of domains in patterns generated by automata #226 (N = 256) * 1.2
Average of 10 times

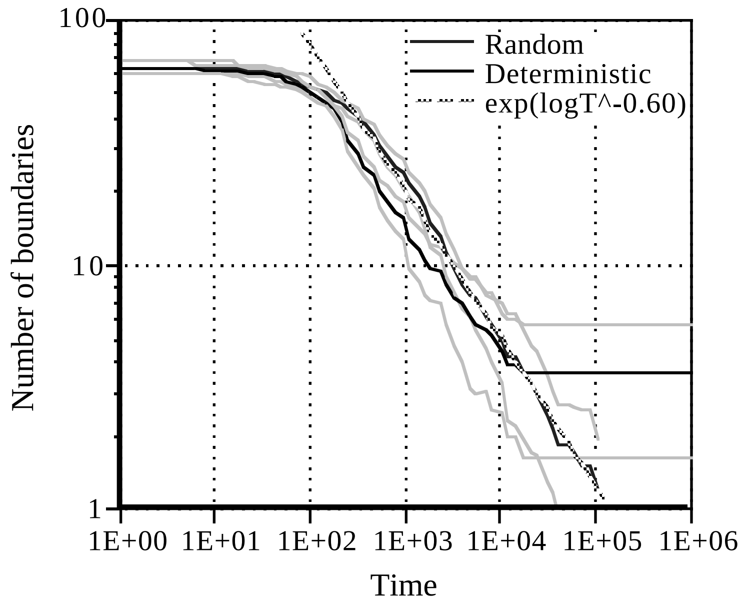


Figure 14. Lifetime of domains in patterns generated by automata #146 (N = 256) * 1.2
Average of 10 times

Relation to fractal dimensions (?)

3.3 Merging and/or splitting patterns

Patterns, in which domains are merging and/or splitting, generated by asynchronous automata are shown here. Patterns generated by automata #166, #58 and #38 are shown for example in **Figure 15** to **Figure 18**.

#166: The differences between the random and interlaced cases, F1 and F2 shown in Figure 15, are as follows. First, two black domains sometimes merge into one in the former, but they do not in the latter. The final state is uniformly black (not white!) in the former. Second, black domains move left in both cases, but the speed of this motion is much slower in the former. However, they never move right.

F1. Random order F2. Interlaced order

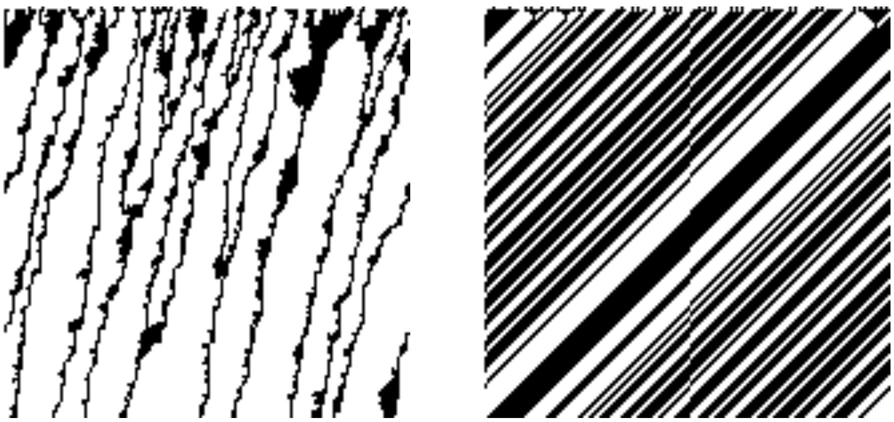


Figure 15. Patterns generated by automata #166 (= 10100110_2)

#58: The characteristics of both patterns, G1 and G2 shown in Figure 16, are completely different in this case. The differences are as follows. The two points described for #146 are the same for #58. Third, a black domain sometimes splits into two, and two black domains sometimes merge into one in the random case. Patterns generated by a random automaton is similar to some of those generated by synchronous automata of Class IV (complex) with more neighbors (a larger value of *r*) [Wol 84]. However, similar patterns are never generated in the interlaced case.

G1. Random order G2. Interlaced order

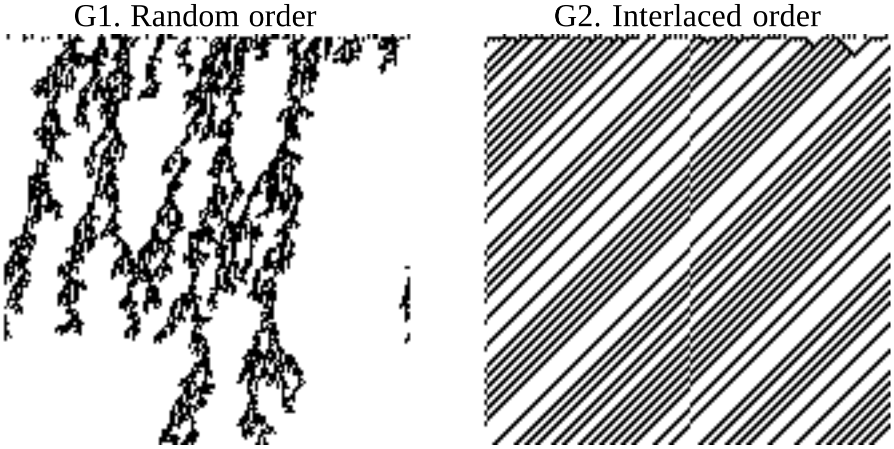


Figure 16. Patterns generated by automata #58 (= 00111010_2)

#74: The both patterns shown in Figure 17 are similar to those generated by #58 automata. An important diffrence between these cases are that the borders between black and white domains can go left or right in #58, but they cannot go right in #74. This condition holds both for the random and deterministic cases.

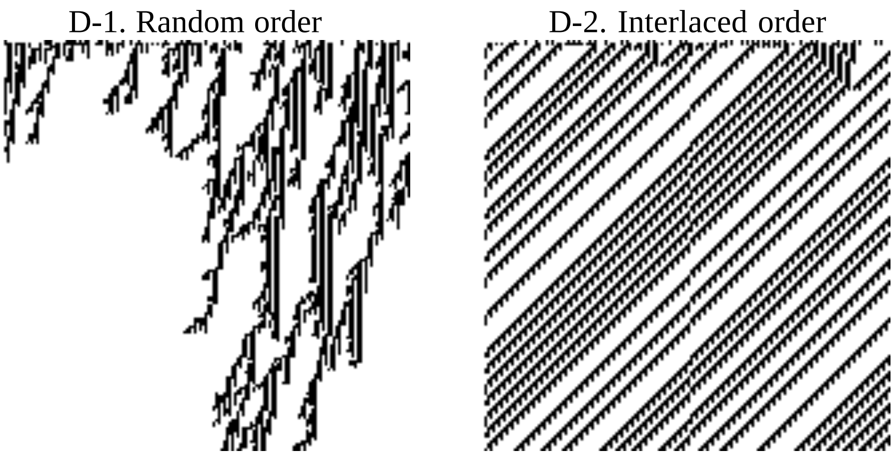


Figure 17. Patterns generated by automata #74 (= 01001010_2)

#38: Patterns H1 and H2, shown in Figure 18, have similarity to those shown in Figure 15. However, pattern L1 is more complicated than pattern F1 because the black domains not only merge but also sometimes split into two domains.

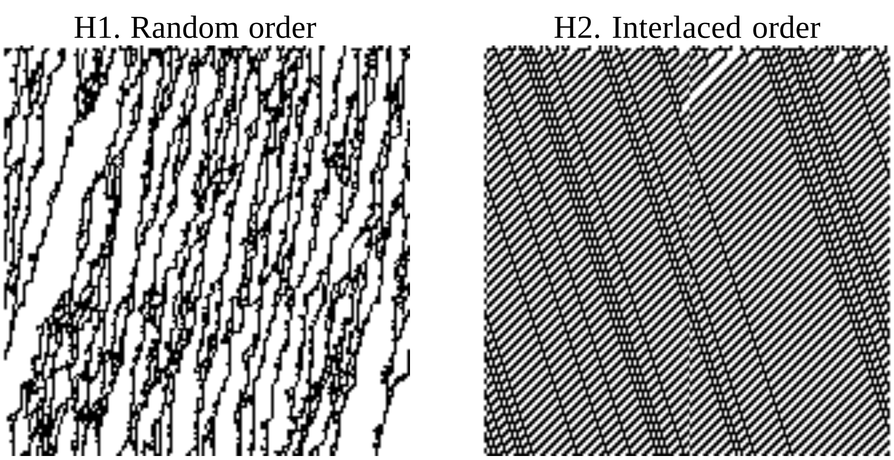


Figure 18. Patterns generated by automata #38 (= 00100110_2)

[**Lifetime analysis of #166, #58 and #74**]

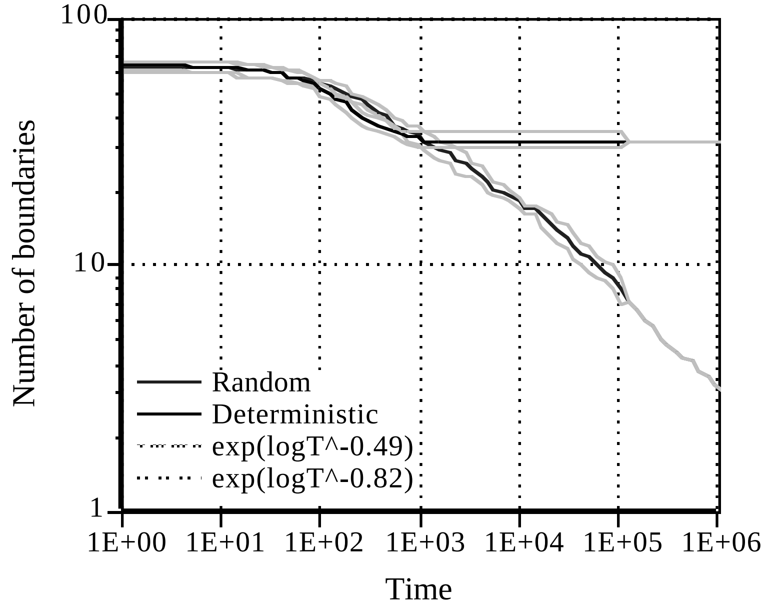


Figure 19. Lifetime of domains in patterns generated by automata #166

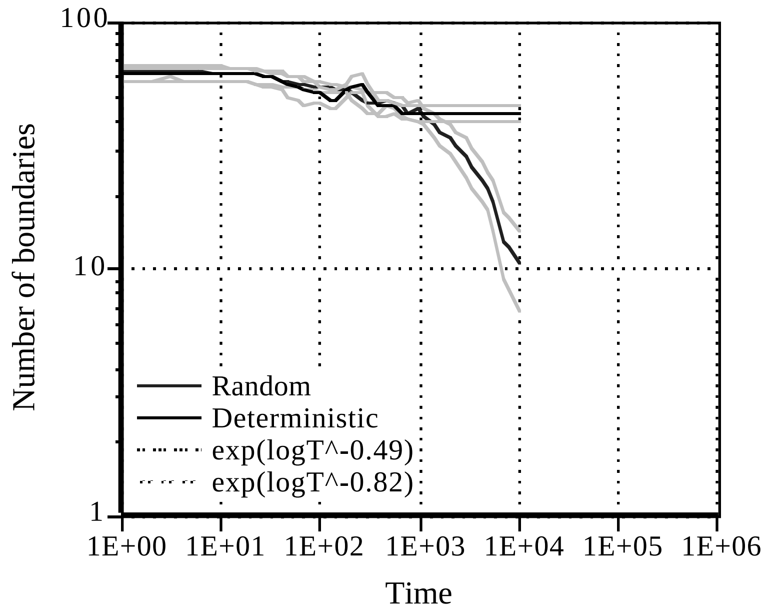


Figure 20. Lifetime of domains in patterns generated by automata #58

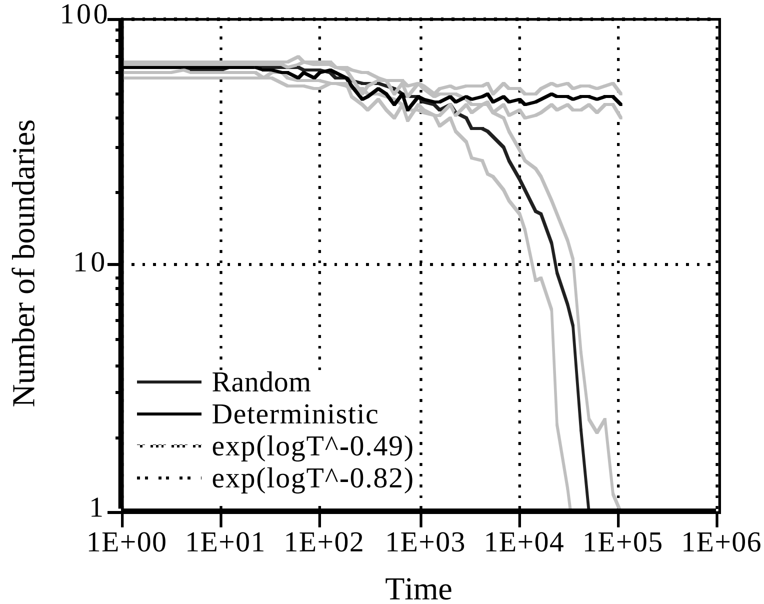


Figure 21. Lifetime of domains in patterns generated by automata #74

3.4 Chaotic and partially chaotic patterns

Chaotic and partially chaotic patterns generated by 1D-ACA are shown here. Some 1D-ACA generate chaotic patterns, some of which is very similar to patterns generated by synchronous automata, and others are quite different from them. Patterns generated by automata #105 and #57 are shown for example in **Figure 22** to **Figure 24**.

#1: No structure can be seen in pattern A-1 shown in Figure 22, which is generated by the randomized automata. Thus, this is classified into chaotic patterns. However, in pattern A-2 shown in Figure 22, which is generated by the deterministic automata, white particles can be seen in dotted background. Thus, this is orderly.

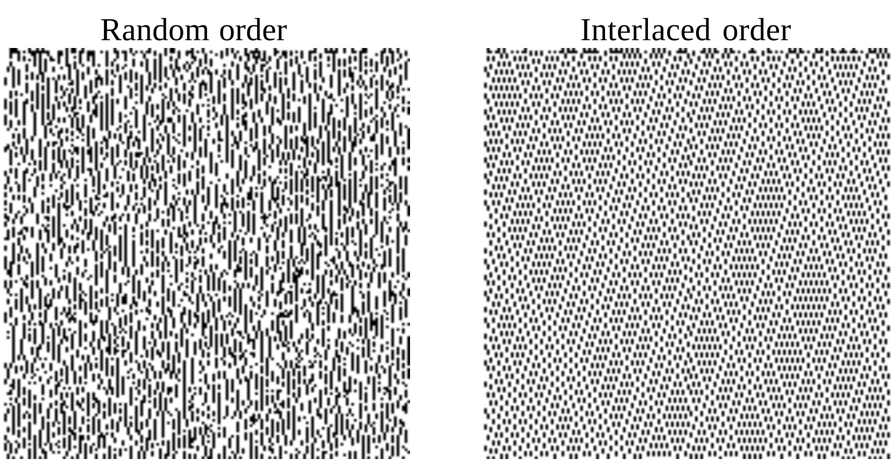


Figure 22. Patterns generated by automata #1 (= 00000001_2)

#105: No significant structure can be seen in pattern I1 shown in Figure 23, which is generated by the randomized automaton. Thus, this is classified into chaotic patterns. However, in pattern I2 shown in Figure 23, which is generated by the interlaced automaton, waves moving to the left or right can be seen. Thus, this is orderly.

I1. Random order　　I2. Interlaced order

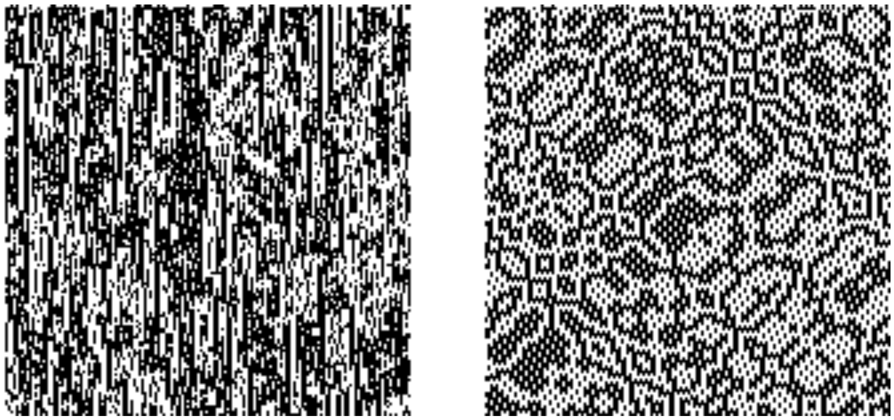


Figure 23. Patterns generated by automata #105 (= 01101001_2)

#57: The patterns generated by automata #57, which are shown in Figure 24, are more complex, but they are similar to those of #1. The pattern is chaotic in the random case, J1. The pattern seems to be more orderly but still chaotic in the interlaced case, J2. The complex structure seen in the interlaced case and its noise-sensitivity is analyzed in Section 5.

J1. Random order　　J2. Interlaced order

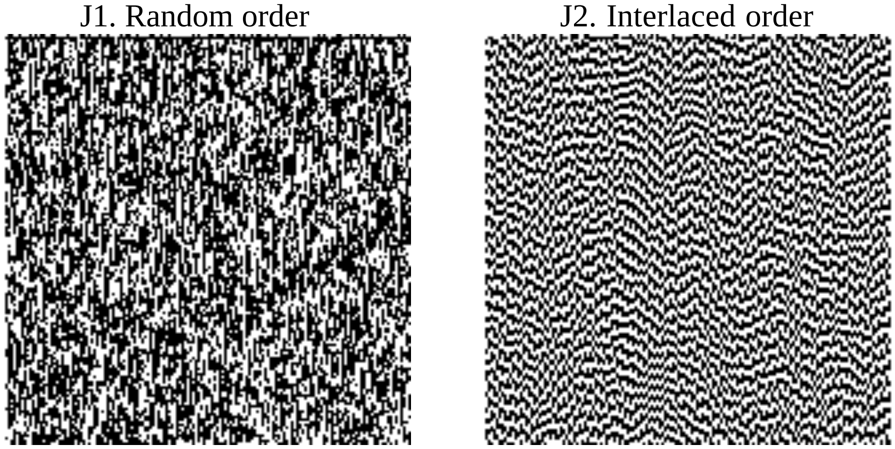


Figure 24. Patterns generated by automata #57 (= 00111001_2)

#54: Pattern C-1 shown in Figure 25, which is generated by randomized automata #54, is less chaotic than those generated by #1 and #43. White holes can be observed in C-1. Black particles can be seen in pattern C-2 shown in Figure 25, which is generated by deterministic automata. This structure has similarity to that in A-2, though their back-grounds are different.

Random order Interlaced order

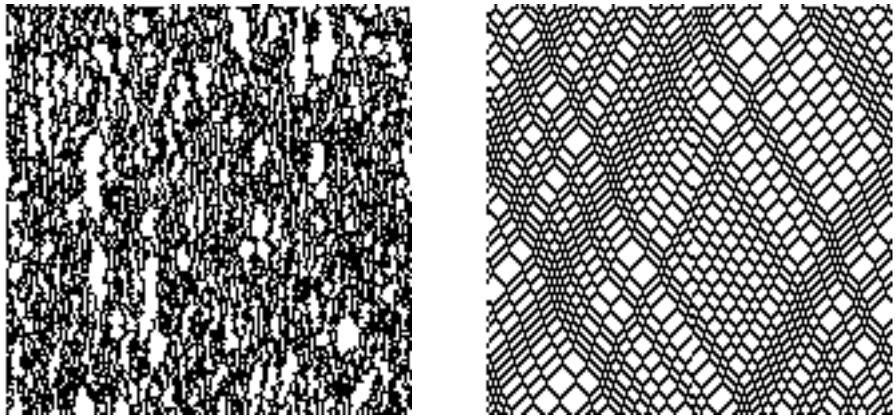


Figure 25. Patterns generated by automata #54 (= 00110110_2)

#60: Pattern D-1 in Figure 26, which is generated by randomized automata #60, seems to have similarity to pattern C-1. However, pattern D-2 in Figure 26, which is generated by deterministic automata #60, is quite different from pattern C-2. Pattern D-2 is similar to that generated by synchronous automata classified to Class III (chaotic automata) by Wolfram.

Random order Interlaced order

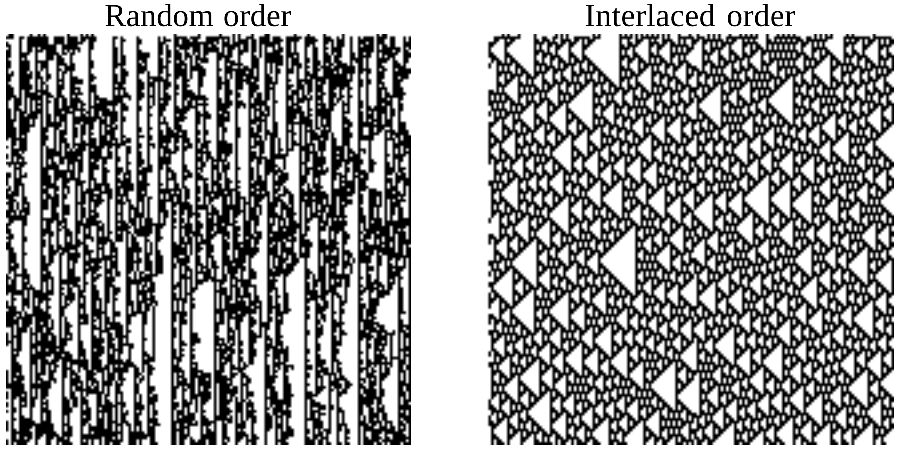


Figure 26. Patterns generated by automata #60 (= 00111100_2)

Some automata generate partially chaotic patterns, such as those shown in **Figure 27** to **Figure 28**.

#73: The thick black stripes in patterns K1 and K2 shown in Figure 27 do not change nor move. The patterns between these stripes are chaotic in the randomized case, and white particles moving left or right can be seen in the interlaced case.

K1. Random order K2. Interlaced order

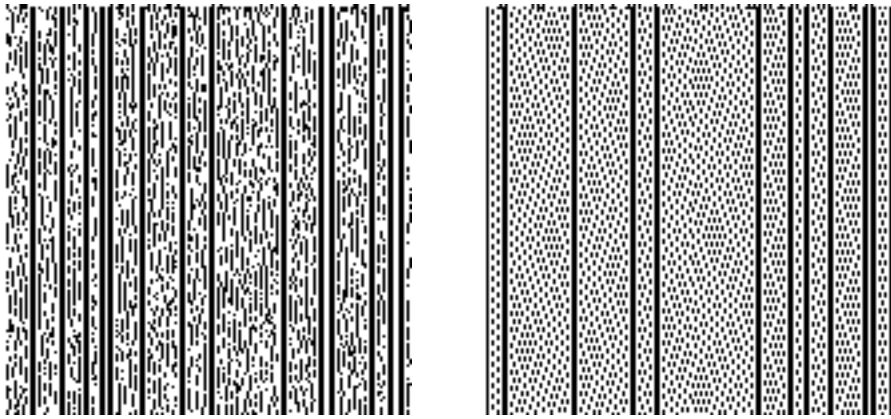


Figure 27. Patterns generated by automata #73 (= 01001010_2)

#37: Patterns L1 and L2, shown in Figure 28, are quite different each other. In the random case, fluctuated particles split and merge, and striped domains are stable. However, in the interlaced case, particles move straight and extinct in pair, and striped domains are unstable. The stability of dark domains in L2 seems to be a phantom. Pattern L1 is also similar to patterns generated by CMLs in “diffusion of defect in chaotic media” phase [Kan 89].

L1. Random order L2. Interlaced order

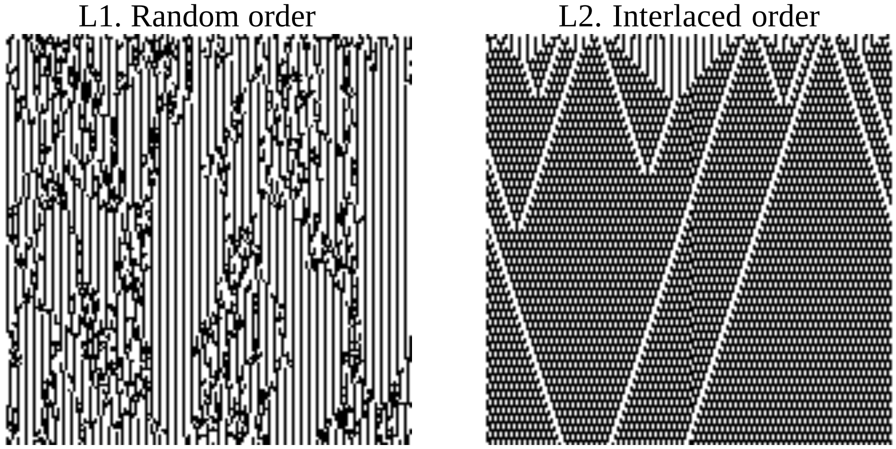


Figure 28. Patterns generated by automata #37 (= 00100101_2)

#90: Patterns L1 and L2, shown in Figure 14A, are quite chaotic, and L2 is similar to patterns generated by synchronous automata #90 of Class III (chaotic ones) [Wol 84]. However, because orderly stripes exist, they are classified to *partially* chaotic patterns here. Many characteristics of L2 are preserved in L1, and, thus, this automata is less noise-sensitive.

[**Simul-transition in interlaced, but not in random**]

L1. Random order L2. Interlaced order

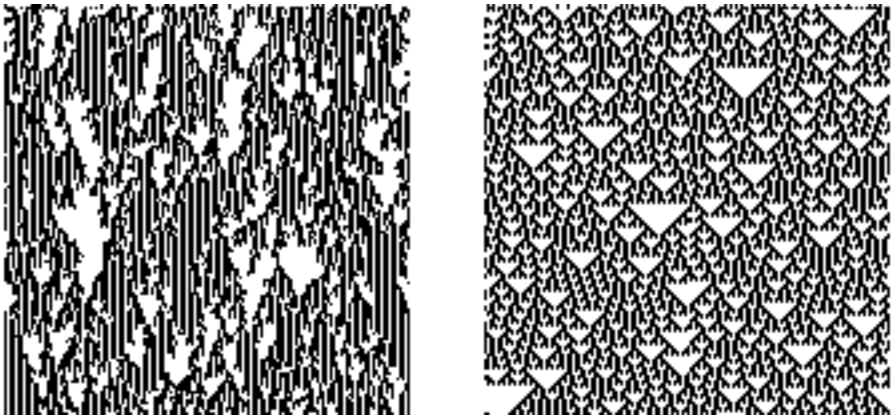


Figure 29. Patterns generated by automata #90 (= 01011010_2)

4. 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 follows.

f_0: If its value is 1, white domains may split into two. Otherwise, they do not split.

f_7: If its value is 0, black domains may split into two. Otherwise, they do not split.

f_2: If its value is 0, black domains are mortal (may die). Otherwise, they are immortal.

f_5: If its value is 1, white domains are mortal. Otherwise, they are immortal.

f_1: If its value is 1, WB borders (white-to-black borders) may move left. Otherwise, they do not move left.

f_6: If its value is 0, BW borders (black-to-white borders) may move right. Otherwise, they do not move right.

f_3: If its value is 0, WB borders may move right. Otherwise, they do not move right.

f_4: If its value is 1, BW borders may move left. Otherwise, they do not move left.

Detailed explanations on the gene functions are omitted because of page limitations. However, the functions can be understood intuitively by **Figure 30**. This figure shows the current state of a cell to be updated, those of the neighbor cells, and the updated state of the cell. For example, the leftmost part of the figure shows the case that all three cells are white. The next state is specified by gene f_0. If the updated state is black as shown, this is the beginning of a black domain, and the white domain splits into two. Other parts of the figure can be interpreted in the same way.

Several examples shown in the previous section are analyzed using the interpretation above.

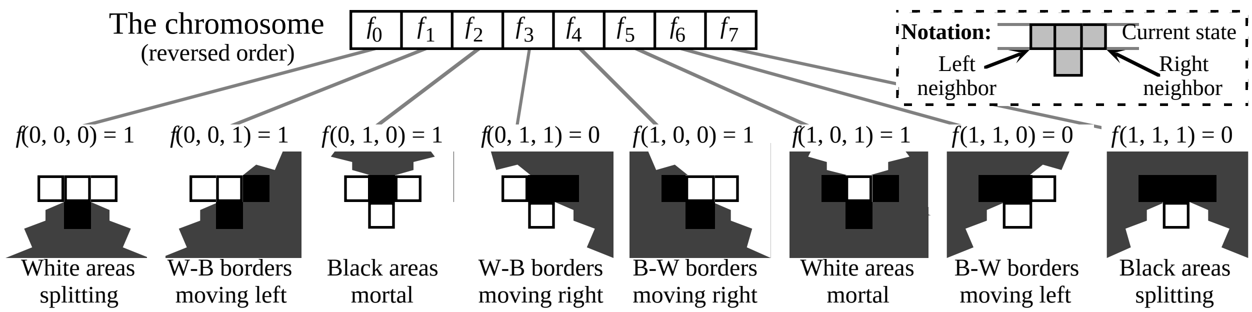


Figure 30. Interpretation of the chromesome

Several examples shown in the previous section are analyzed using the interpretation above.

#226 (= 11100010_2): First, gene f_0 is 0 and gene f_7 is 1. Thus, both white and black domains in patterns C1 and C2 do not split. Second, f_2 is 0 and f_5 is 1. Thus, both white and black domains are mortal. Actually, black domains die in C1 and C2. All the black domains die if the random order is used. However, some black domains continue to exist if the interlaced order is used, because gene f_7 is not used for the state transitions of such domains. Thus, the properties of the automaton are only partially expressed when no noise exists. Third, f_1 is 1 and f_3 is 0. Thus, WB borders can move in both directions. The WB borders actually move in both directions in C1. However, they move in single direction in a period in C2. This is another example of partial expression of genes under noise-free situations. Fourth, f_6 is 1 and f_4 is 0. Thus, BW borders do not move. This property is expressed both in C1 and C2.

#166 (= 10100110_2): f_0 is 0 and f_7 is 1. Thus, both white and black domains in patterns F1 and F2 do not split. Both f_2 and f_5 are 1. Thus, black domains are immortal and white domains are mortal. White domains die in C1, but they continue to exist after the automaton comes into a limit cycle in C2. This is another example of partial expression of genes. **Expression and suppression of other genes can easily be observed**.

#58 (= 00111010_2): Black domains can split because f_0 is 0, but white domains do not split because f_7 is 0. Expression of gene f_7 can easily be seen in G1, but this property is also suppressed in G2. **Expression and suppression of other genes can easily be observed**.

Other patterns, such as those shown in Sections 3.1 to 3.3, can be understood in the same way.

The above simpler relation between the look-up table values and the property of patterns exist only if the computation is sequential. Synchronous or partially synchronous automata cannot be analyzed using the above interpretation. It is much more difficult to analyze these automata.

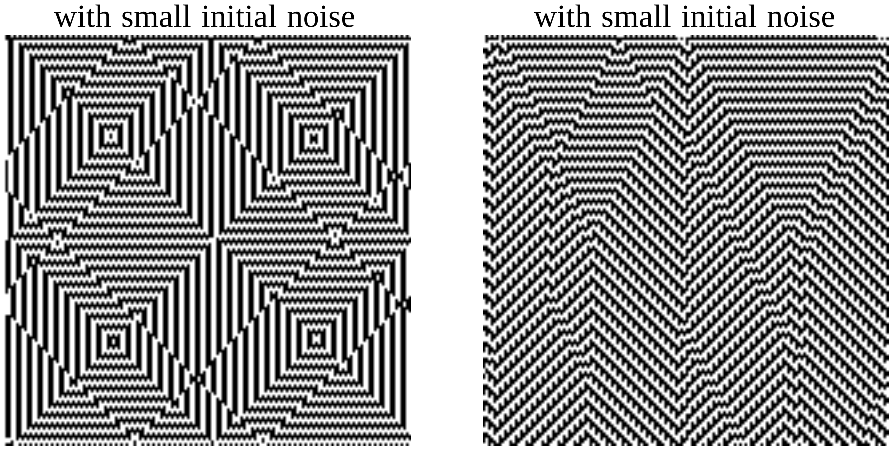
5. Very Noise-sensitive Patterns

Automata #105 and #57, whose example patterns have been shown in Figure 23 and Figure 24, are very sensitive to noise, though not all chaotic automata generate very noise-sensitive patterns. There are two "fluctuations" in the interlaced order with $C = \lfloor (N-1)/2 \rfloor$ or $\lceil (N+1)/2 \rceil$. If $N = 8$ and $C = 3$ (see Figure 2), cells 1 and 5 are "fluctuated." In the first scan of cells, the computations on cells 0, 3 and 6 refer to the initial values of the neighbor cells, those of cells 7, 2 and 5 refer to the new values of the neighbors, and those of cells 1 and 4 refer to the initial value of the right neighbor and the new value of the left neighbor. These differences in the order of computation cause no significant differences in most of 1D-ACA. However, there are significant differences in automata #105, #57 and several others.

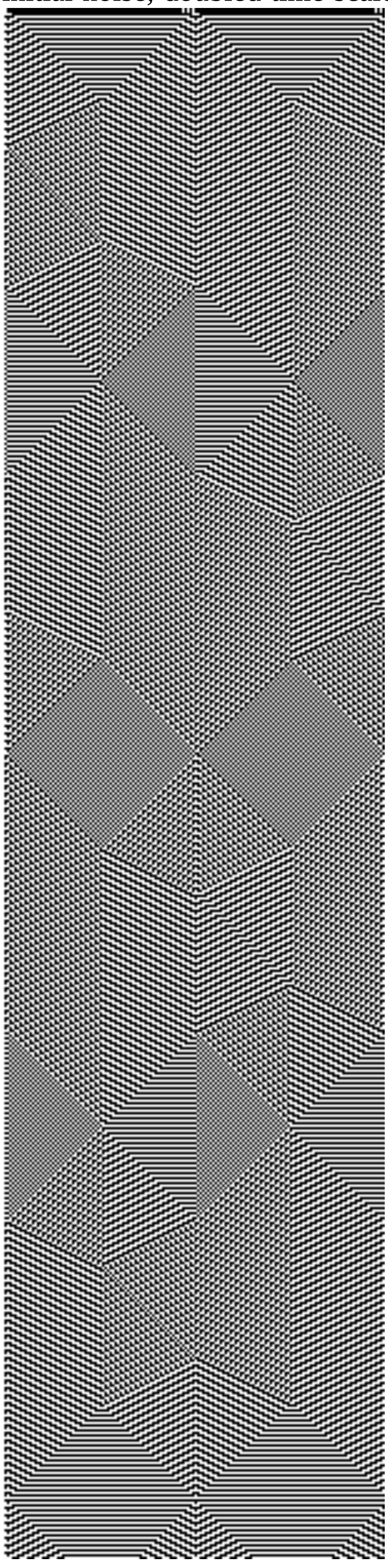
These differences are shown in **Figure 31**. Figure 31 a and b show patterns generated by automata #105 and #37 (See also Figure 23 and Figure 24). The initial state is almost black, but there are two white points. It is easy to see the two different effects caused both by the fluctuation on the order of computation and by the noise in the initial state. The former noise works near the left edge of the patterns and near the center (at cells 1 and 76). It is not easy to see the propagation of the latter noise in this figure. To clarify the non-local structure, the time-scale is reduced by half in Figure 31 c and d. Figure 31 c shows the pattern generated when there is no initial noise. Many domains with different textures, or locally repetitive structures, can be seen in this figure. No such phenomena occur, if the order of computation is alternate, i.e., even cells are computed first and then odd cells are computed. Although the long-scale pattern is periodical, the cycle is so long that it is not possible to show a whole cycle in the figure. Figure 31 d shows a pattern with a small initial noise. The propagation of the noise can be tracked in this figure. They refract when they go into a different domain, and sometimes cause new waves. Detailed analysis of these phenomena is a future work.

(a) #105, interlaced order, with small initial noise

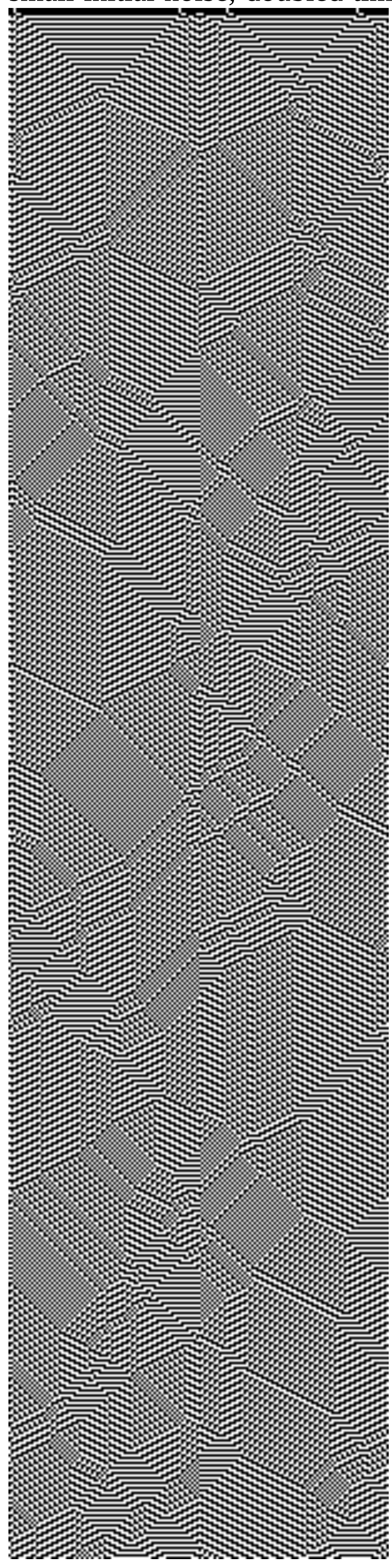
(b) #57, interlaced order, with small initial noise



(c) #57, interlaced order, without initial noise, doubled time scale



(d) #57, interlaced order, with small initial noise, doubled time scale



(e) #105, fixed random order, with initial noise

(f) #146, fixed random order, with initial noise

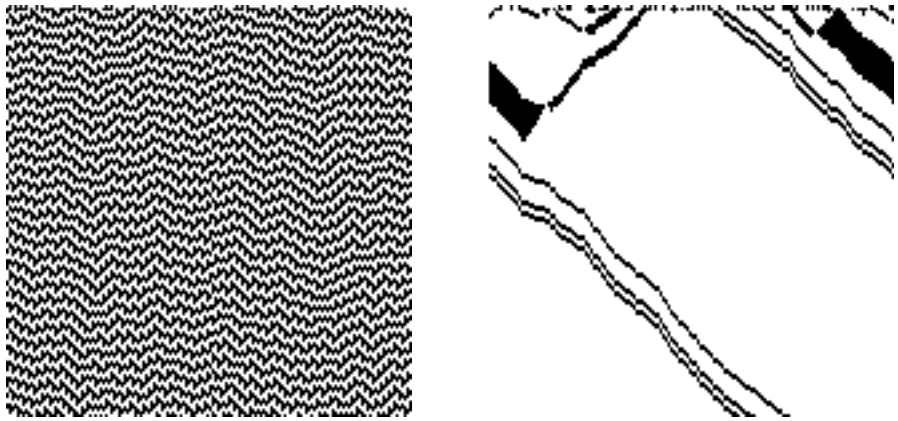


Figure 31. Patterns generated by very noise-sensitive automata #105 and #57 (*N* = 152)

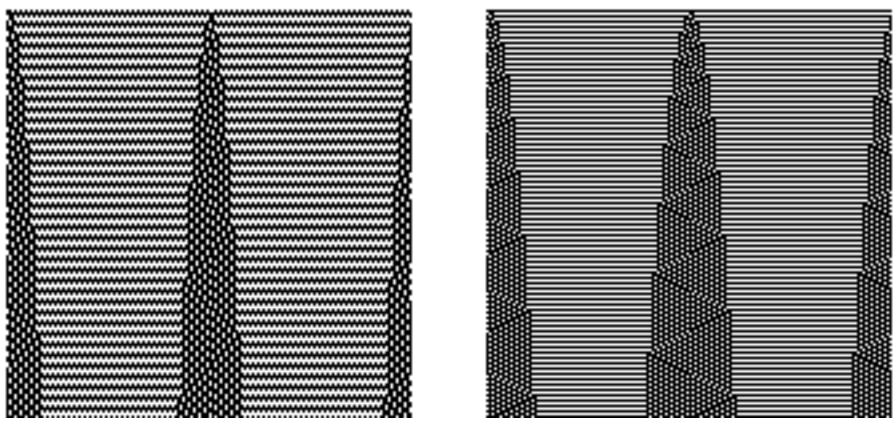


Figure 32. Patterns generated by very noise-sensitive automata #123 (= 10011001_2)

Figure 31 e shows a pattern generated by automata #57 using a fixed random order. The initial state is uniformly white. This pattern is chaotic and completely different from that shown in Figure 31 b. However, the characteristics of most other patterns are not destroyed by this computation order. An example, a pattern generated by automata #146, is shown in Figure 31 f. There are no significant differences between this and pattern D2.

[**Figure 32**]

6. Conclusions

Two major effects caused by adding noise, i.e., randomness or fluctuation, to the order of computation in 1D-ACA are shown in the present paper. One major effect is that properties of 1D-ACA embedded in their "chromosomes" are fully expressed in their patterns when stronger noise exists, i.e., when the order of computation is random. However, the properties are only partially expressed when no noise or weaker noise exists. The other major effect is that very fragile particles and domains, which may be regarded as phantom phenomena because they are almost never seen if noise exists, are sometimes observed in noise-less environments (in Sections 3.4 and 4). The characteristics of patterns generated by several 1D-ACA are drastically changed from uniform patterns to patterns with multiple or chaotic domains even if low level of noise is added. Several other effects, such as delay of pattern motion under existence of noise, are observed. Although 1D-ACA are simpler systems, I believe the results of this research contribute to research of emergent computation, such as CCM (chemical casting model) [Kan 94], and artificial life.

One important direction of future work is to analyze the statistics, such as entropies, of patterns generated by 1D-ACA to support the hypotheses quantitatively. Another direction is to analyze or develop mechanisms of controlling partial expression and suppression of the properties embedded in the chromosomes, because partial expression is the usual case in biological life.

7. References

[Ber 94] Bersini, H., and Detours, V.: Asynchrony Induces Stability in Cellular Automata Based Models, *Artificial Life IV*, 382–387, MIT Press, 1994.

[Hak 78] Haken, H.: *Synergetics — An Introduction*, Springer-Verlag GmbH & Co. KG, 1978.

[Hof 87] Hofmann, M. I.: A Cellular Automaton Model Based on Cortical Physiology, *Complex Systems*, 1, 187–202, 1987.

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

[Kan 89] Kaneko, K.: Pattern Dynamics in Spatiotemporal Chaos, Physica D, Vol. 34, pp. 1–41, 1989.

[Kan 94] Kanada, Y., and Hirokawa, M.: Stochastic Problem Solving by Local Computation based on Self-organization Paradigm, *27th Hawaii International Conference on System Sciences*, 1994.

[Kan 94b] Kanada, Y.: The Effects of Randomness in Asynchronous 1D Cellular Automata, *Artificial Life IV Poster/Demo Session*, 1994.

[Kau 84] Kauffman, S. A.: Emergent Properties in Random Complex Automata, Physica D, Vol. 10, pp. 145–156, 1984.

[Lan 90] Langton, C. G. et al. eds.: *Artificial Life*, Addison Wesley, 1989.

[Lan 92] Langton, C. G. et al. eds.: *Artificial Life II*, Addison Wesley, 1991.

[Lan 93] Langton, C. G. et al. eds.: *Artificial Life III*, Addison Wesley, 1993.

[Lum 94] Lumer, E. D., and Nicolis, G.: Synchronous versus Asynchronous Dynamics in Spatially Distributed Systems, *Physica D*, 71, 440–452, 1994.

[Mat 79] Maturana, H. R., and Varela, F. J.: *Autopoiesis and Cognition: The Realization of the Living*, Reidel, 1979.

[Par 86] Park, J. K., Steiglitz, K., and Thurston, W.: Soliton-Like Behavior in Automata, *Physica D*, 19, 423–432, 1986.

[Pri 77] Nicolis, G., and Prigogine, I.: *Self-organization in Nonequilibrium Systems — From Dissipative Structures to Order through Fluctuations*, John Wiley & Sons, Inc., 1977.

[Vic 89] Vicsek, T.: *Fractal Growth Phenomena*, World Scientific, 1989.

[Wol 83] Wolfram, S.: Statistical Mechanics of Cellular Automata, *Reviews of Modern Physics*, 55, 601–, 1983.

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