Complex Systems **3** (1989) 253–267

Fractal and Recurrent Behavior of Cellular Automata

Karel Culik II
Simant Dube
Department of Computer Science, University of South Carolina, Columbia, SC 29208, USA

Abstract.
In recent years, cellular automata (CA) have been found capable of producing complex behavior. Some examples of cellular automata show remarkably regular behavior on finite configurations. On simple initial configurations, the generated pattern might be fractal or self-similar. In this paper, regular evolution of totalistic linear CA is investigated. In particular, it is shown that additive CA will always produce a highly regular behavior on an arbitrary finite configuration as the initial seed. Totalistic CA with binary function code of the form $0^n1^{2m}0^n$ are also studied. The results are extended to trellis automata.

1. Introduction

Recently, cellular automata (CA) have been intensively studied as models of complex natural systems containing large numbers of simple identical components with local interactions [9,14]. They consist of an array of sites or cells, each with a finite set of possible states. The states of the cells evolve synchronously in discrete time steps according to identical rules. A cellular automaton is given by its local rule (CA rule), which specifies how the state of a cell is determined by the previous states of the cells in a neighborhood around it. The local rule specifies a deterministic global function on the configurations of the system.

An interesting application of CA has been in modeling chaotic behavior, as they provide a large class of examples of apparent chaos. However, there are some other examples of CA which retain a great deal of regularity that become mathematically tractable, and it is possible to analyze their long-term behavior. Based on statistically observed long-term behavior of CA, Wolfram [15] suggested a classification of cellular automata. He observed that cellular automata appear to fall into four classes. For a formal definition of similar classes, see reference [5]. The majority of the examples of CA seems to belong to the third class in which the evolution leads to a chaotic pattern.

They correspond to the physical systems with so called "strange attractors." However, not all the CAs in this class are equally chaotic as already observed in [14]. Interestingly, some of them can generate self-similar figures or fractals when the initial seed is a short finite configuration, and others can produce fractal patterns for arbitrary finite initial configurations. Wilson has studied the generation of fractals by additive CA, that have mod 2 addition (XOR) local rule, when the initial seed is single 1 and has computed the fractal dimensions for several of such cases [10–13]. A detailed study of the evolution patterns of XOR trellis automaton of radius 1 has been made in [2].

We will be interested in analyzing the regular behavior, which is self-similar to some extent, of CA when initial seeds are random finite configurations. We will discuss the simulation results and restrict ourselves to totalistic one-dimensional CA. In section 4, we prove that additive (XOR) CA will always show a highly regular behavior on an arbitrary finite configuration as the initial seed. In section 5 we study other CA with fractal-like behavior, in particular the totalistic CA with arbitrary radius r and binary local function of the form $0^n1^{2m}0^n$, $n \geq 1$, $m \geq \epsilon_r \geq 1$, $2(m+n) = 2r+2$, where ϵ_r is some constant depending on r. Here, we have a proof only for finite seeds of certain special form. However, we conjecture that these CA show regular behavior for all finite configurations as initial seeds. In a similar way, we also study totalistic trellis automata in section 6. Note that every CA or trellis automaton can be simulated by a totalistic one [1].

2. Preliminaries

Formally a cellular automaton is a 3-tuple $A = (S, r, f)$, where S is the finite set of states, r is the *neighborhood radius*, and f is the *local function*. A CA can be viewed as a linear biinfinite array of cells. The neighborhood of the CA is the sequence of relative positions $\{-r, -r+1, \ldots, -1, 0, 1, \ldots, r-1, r\}$. In other words, the neighborhood of a cell consists of the cell itself and r of its neighbors at each side. The local function $f : S^{2r+1} \to S$ is a complete function which computes the next state of a cell from the current states of all cells in its neighborhood.

A *configuration* c is a function $c : Z \to S$, which assigns a state in S to each cell of the CA. The set of all configurations is denoted by S^Z. The local function f is extended to the *global function* $G_f : S^Z \to S^Z$ of the set of configurations into itself. By definition, for $c_1, c_2 \in S^Z$, $G_f(c_1) = c_2$ if and only if

$$c_2(I) = f(c_1(I-r), \ldots, c_1(I), \ldots, c_1(I+r)),$$

for all I in Z.

The function G_f describes the dynamic behavior of the CA: the CA moves from the configuration c at time t to the configuration $G_f(c)$ at time $t+1$. The state of a cell depends only on the states of the cells in its neighborhood. Notice that besides being locally defined, the global function G_f is total and translation-invariant. The sequence $c, G_f(c), G_f^2(c), \ldots$ describes the *evolution* of the CA on initial configuration (or seed) c.

Frequently, a state $\tilde{q}$ with the property $f(\tilde{q}, \tilde{q}, \ldots, \tilde{q}) = \tilde{q}$ is distinguished and called the *quiescent state*. In a CA, there may be more than one state with the above property, but at most one of them is distinguished as the quiescent state.

A configuration c is called to be a *finite configuration* if finitely many cells are in nonquiescent states.

A CA rule is called *totalistic* if the states are integers and the next state of a cell depends only upon the numerical sum of the states of its neighbors. Let k and r be the number of states and the neighborhood radius, respectively, of a totalistic CA. If $k = 2$, the function code of the CA can be then denoted by a binary number

$$b_{2r+1} \ldots b_2 b_1 b_0,$$

for $b_i \in \{0, 1\}, 0 \leq i \leq 2r + 1$, and it means that

$$f(s_{-r}, \ldots, s_0, \ldots, s_r) = b_i, \text{ if } s_{-r} + \ldots + s_0 + \ldots + s_r = i.$$

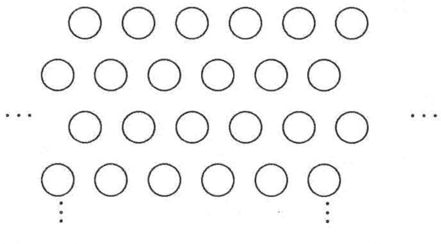
These binary function codes are often converted into decimal numbers for convenience. For example, for $r = 1$, the code 10 denotes the CA in which the next state of a cell is the XOR (exclusive-or) of the values of its neighbors. In this paper, we will focus only on totalistic CA with states $\{0, 1\}$, the state "0" being the quiescent state.

Trellis automata are quite similar to CA and we will informally define them. The time-space evolution of a trellis automaton of some neighborhood radius r is shown in figure 1a. In figure 1b, the first five steps of evolution of the XOR trellis automaton for radius 1 is shown when the initial seed is a single 1. Note that the alternate rows are displaced by half a unit with respect to each other. The initial configuration resides on the topmost row. In any time step, the state of a cell on a row is determined by its $2r$ neighbors on the row above it, with r neighbors on its each side. Since, this is a space-time evolution pattern, only one row is active at a time. Notice that only the case when $r = 1$ is referred to as a trellis in literature [3,4,7] and we have generalized this definition to arbitrary neighborhood radius. Also note that, for $r = 1$, trellis automata are equivalent to one-way CA in computational power [3].

A self-similar figure is one in which a part, after proper rescaling, resembles the whole. In other words, the figure contains the copies of itself. Self-similar figures are often called *fractals*. Mendlebrot [8] has given many examples of such figures. Fractals can be generated using an initiator and a generator. A drawing rule recursively describes how to generate the fractal as the limiting case of the drawing process which is started with the initiator.

3. Classification of CA

Wolfram characterized some qualitative features of cellular automaton evolution [15] and gave empirical evidence for the existence of four basic classes



```
     1
    1 1
   1 0 1
  1 1 1 1
 1 0 0 0 1
1 1 0 0 1 1
     ⋮
```

(a) (b)

Figure 1: (a) Time-space diagram of a trellis automaton; (b) evolution of XOR trellis for $r = 1$.

of behavior in CA (the example codes in the parentheses are out of the 32 possible legal totalistic rules with $k = 2$ and $r = 2$):

1. Evolution leads to a homogenous state (codes 0, 4, 16, 32, 36, 48, 54, 60, and 62).
2. Evolution leads to a set of separated simple stable or periodic structures (codes 8, 24, 40, 56, and 58).
3. Evolution leads to a chaotic pattern (codes 2, 6, 10, 12, 14, 18, 22, 26, 28, 30, 34, 38, 42, 44, 46, and 50).
4. Evolution leads to complex localized structures, sometimes long lived (codes 20 and 52).

The above classification is based on empirical observations of the pattern of configurations generated by evolution, when a CA is started with a random initial configuration. For a more formal definition of similar classification see [5], which agrees with Wolfram's classification in all examples given above.

When the initial configurations are restricted to random finite configurations (it does not make sense to investigate fractal behavior of CA on infinite configurations), the above classification allows further refinement. It is well known that some CA in Class 3 show more regular behavior, and some may even generate fractal-like patterns. We will investigate this kind of regular behavior of Class-3 CA and give mathematical justification for some cases. Based on empirical evidence, we propose that when restricted to finite configurations, Class 3 could be subdivided in the following three types:

Subclass 1. Evolution leads to a highly regular and recurrent (almost fractal) pattern (codes 12, 30, 42). On small initial configurations, the pattern is pure fractal with computable fractal dimension.

Subclass 2. For random initial configurations, evolution leads to a chaotic pattern, except on short initial configurations for which the pattern is fractal or almost fractal (codes 2, 6, 10, 14, 34, 38).

Subclass 3. Evolution leads to a chaotic pattern (codes 18, 22, 26, 28, 44, 46, 50).

By a pure fractal, we mean that the evolution can be characterized by a recursive formula, which essentially will correspond to Mendlebrot's drawing rule. Pieces of the evolution pattern, when appropriately magnified, are same as the whole pattern. One can compute the fractal dimension of the pattern from the recursive formula. Note that because of the presence of Subclass 2, the CA form a rather continuous spectrum when classified on the basis of their capability to generate fractals.

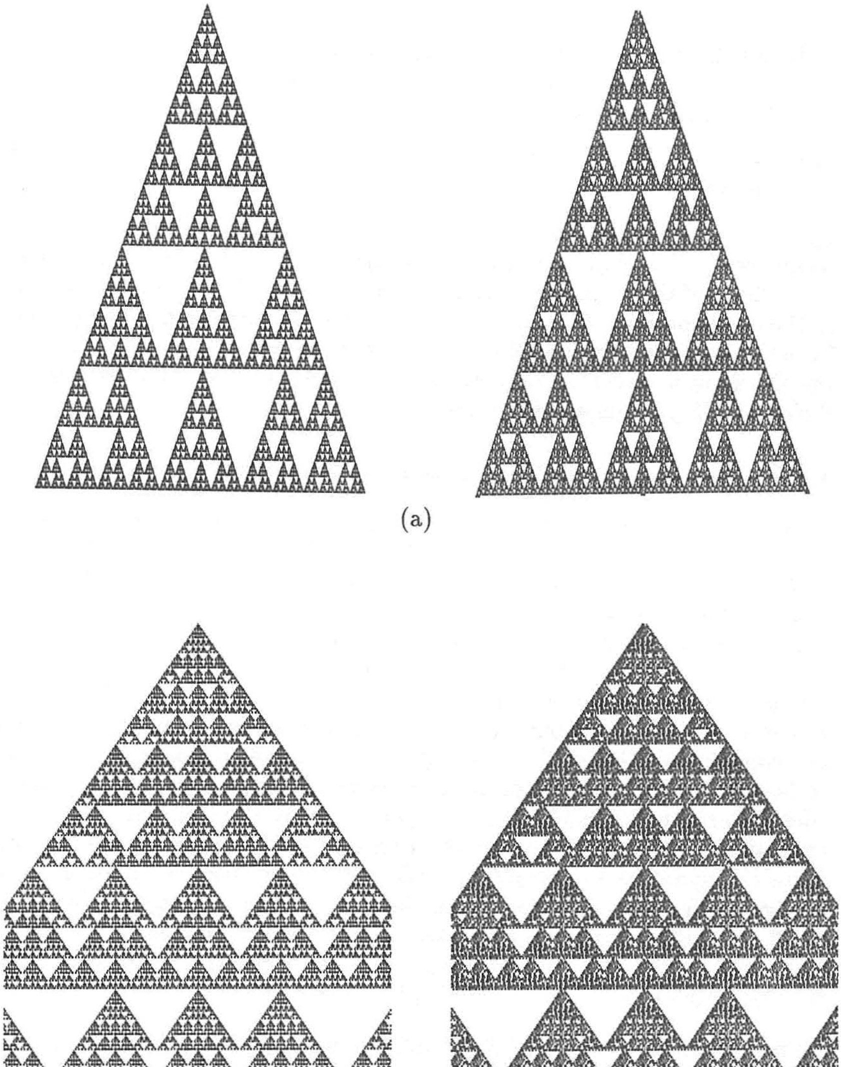
4. Fractal generation by CA with additive rules

In this paper, we are mainly interested in CA that exhibit regular (fractal) behavior. In this section, we will study the CA with additive (XOR) rules that are are most tractable to theoretical analysis. For $r = 2$, such rule is code 42, but we will analyze the XOR rules for arbitrary r. We will prove that for any arbitrary r, the CA with XOR rule always generates a regular pattern, namely when started on a string w whose length n is a power of 2 (any arbitrary initial finite string can be made to have a length which is a power of 2 by appending appropriate number of 0s), after n time steps the configuration is w^{2r+1}. The XOR code for radius r is $2(4^{r+1} - 1)/3$. In figure 2a, two example evolution patterns for $r = 1$, code 10 are shown. The pattern on the left-hand side is generated by the simplest initial seed consisting of single 1, and the pattern on the right-hand side has a random finite configuration as the initial seed. Notice the regular appearance of triangular clearings in the patterns. Similarly, in figure 2b the evolution of $r = 2$, code 42 is shown, the left pattern generated by the seed with single 1. In the right-hand side pattern, which is generated on a random seed, only the first level of the recurrence is clearly visible.

Lemma 1. *Let f be the XOR CA rule for radius r. Then, when started on the configuration consisting of a single 1, for any n where n is a power of 2, we obtain after n time steps the configuration which is $2r+1$ repetitions of the string formed by a 1 followed by $n-1$ 0s, i.e., $G_f^n(1) = (10^{n-1})^{2r+1}$.*

The above lemma, which can be proved by induction, explains why XOR CAs generate perfect fractal patterns on an initial seed containing a single 1. The following lemma is the well-known speed-up theorem for linear CA stated in our terminology.

Lemma 2. *Let f be a CA rule for radius r. Then there exists another CA rule g for radius rn, $n \geq 1$, such that n time steps of G_f are simulated in one time step of G_g i.e. $G_f^n(\alpha) = G_g(\alpha)$ where α is an arbitrary configuration.*



(a)

(b)

Figure 2: Evolution of (a) code 10, $r = 1$, and (b) code 42, $r = 2$.

We will denote g by $f^{(n)}$, if it simulates f for n steps. An interesting question that can be asked is how easy is it to compute $f^{(n)}$ for an arbitrary CA rule f and an arbitrary n. In general, it may require exponential effort in n, but in the case of XOR CA rules, due to the properties of the XOR function, one can come up with a shorthand formula. The following lemma tells us how, for the case of XOR CA rules, this speed-up rule can be computed if we know the fractal pattern generated by the CA on an initial seed containing a single 1 at cell 0. More precisely, it states that if we know the configuration in the pattern after nth time step, then we can write down $f^{(n)}$. We will denote the state of cell i after n time steps by $(G_f^n(1))_i$, where it is assumed that cell 0 contains 1 and all other cells contain 0 at the beginning.

Lemma 3. *Let f be the XOR CA rule for radius r. Then, for all $n \geq 1$, the speed-up CA rule $f^{(n)}$ is given by*

$$f^{(n)}(s_{-k}, \ldots, s_0, \ldots, s_k) = s_{j_1} \oplus s_{j_2} \oplus \ldots \oplus s_{j_m}$$

where $k = rn$ and j_ts, $1 \leq t \leq m$, are exactly those values of i for which $(G_f^n(1))_i = 1$.

Proof. The proof follows by induction on n. For the basis ($n = 1$), we have $(G_f^1(1))_i = 1$ for $-r \leq i \leq r$. Hence,

$$\begin{aligned} f^{(1)}(s_{-r}, \ldots, s_0, \ldots, s_r) &= s_{-r} \oplus \ldots \oplus s_0 \oplus \ldots \oplus s_r \\ &= f(s_{-r}, \ldots, s_0, \ldots, s_r) \end{aligned}$$

Clearly, f is the speed-up of itself for one time step. Assume that the hypothesis holds for some $n \geq 1$. Consider the speed-up of f for $n+1$ time steps. Let $k = r(n+1)$. Now,

$$f^{(n+1)}(s_{-k}, \ldots, s_k)$$

$$= f^{(n)}(f(s_{-k}, \ldots, s_{-k+2r}), f(s_{-k+1}, \ldots, s_{-k+1+2r}), \ldots, f(s_{k-2r}, \ldots, s_k))$$

$$= f^{(n)}(s_{-k} \oplus \ldots \oplus s_{-k+2r}, s_{-k+1} \oplus \ldots \oplus s_{-k+1+2r}, \ldots, s_{k-2r} \oplus \ldots \oplus s_k)$$

$$= \bigoplus_{(G_f^n(1))_j = 0} (s_{j-r} \oplus \ldots \oplus s_{j+r}) \text{ (by Induction Hypothesis)} \quad (1)$$

$$= s_{j_1} \oplus s_{j_2} \oplus \ldots \oplus s_{j_m}$$

where, s_{j_t}, $1 \leq t \leq m$, is a term in the expression

iff s_{j_t} appears odd number of times in (1)

iff there are odd number of js in (1) such that $(G_f^n(1))_j = 1$ and $j_t - r \leq j \leq j_t + r$

iff $(G_f^n(1))_{j_t - r} \oplus \ldots \oplus (G_f^n(1))_{j_t + r} = 1$

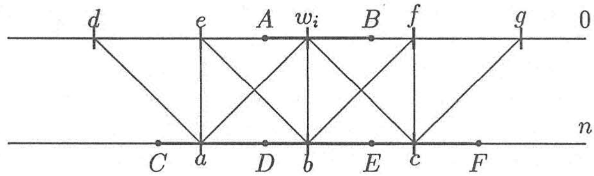


Figure 3: Proof of theorem 1 for $r = 1$.

iff $(G_f^{n+1}(1))_{j_t} = 1$.

As an example of the above lemma, for $r = 2, n = 4$, and therefore $k = 8$, we have

$$f^{(4)}(s_{-8}, \ldots, s_8) = s_{-8} \oplus s_{-4} \oplus s_0 \oplus s_4 \oplus s_8.$$

Theorem 1. *Let f be the XOR CA rule for radius r. Then, for all n, where n is a power of 2,*

$$G_f^n(w) = w^{2r+1}$$

for every finite configuration w of length n (note that w may start or end with zeros).

Proof. The best way to present the proof is by illustrating it with a diagram. Consider the case when $r = 1$. Let w be the initial finite configuration of length n, where n is some power of 2, shown as AB in figure 3. In the figure, $|AB| = |CD| = |DE| = |EF| = n$. After n time steps, the configuration is CF. Consider three cells a, b, and c, at a distance i not more than n, to the right of C, D, and E respectively. By lemma 1 and lemma 3,

$$a = d \oplus e \oplus w_i = 0 \oplus 0 \oplus w_i = w_i,$$

$$b = e \oplus w_i \oplus f = 0 \oplus w_i \oplus 0 = w_i,$$

$$c = w_i \oplus f \oplus g = w_i \oplus 0 \oplus 0 = w_i.$$

i.e., each of the three cells is in state w_i. Therefore, $CD = DE = EF = w$, i.e., w repeats itself three times after n time steps.

For the general r, by lemma 1 and lemma 3

$$f^{(n)}(s_{-k}, \ldots, s_k) = \bigoplus_{i=0}^{2r} s_{-k+in}$$

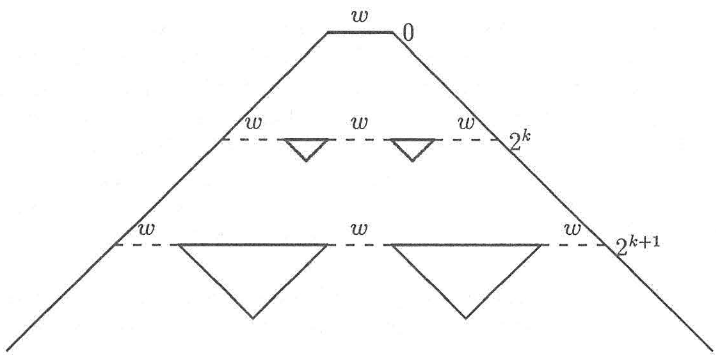


Figure 4: Illustration of corollary 1.

where $k = rn$. Let w be the initial finite configuration of length n, where n is a power of 2, residing in cells 0 to $n-1$. Then, after n time steps, $(G_f^n(w))_{-k+jn+i}$, $0 \leq i \leq n-1$, is the exclusive-or of $2r+1$ cell values, exactly one of which is w_i and the rest are 0 and is therefore equal to w_i for all values of j such that $0 \leq j \leq 2r$. The reader can easily verify this and see why it implies that $G_f^n(w)$ will be w^{2r+1}. ∎

An immediate corollary of the above theorem:

Corollary 1. *Let w be a seed that starts and ends with 1, and let $2^{k-1} < |w| \leq 2^k$ for some $k > 0$. Then, if f is the XOR CA rule for radius r,*

$$G_f^{2^n}(w) = (w0^{2^n-|w|})^{2r}w$$

for all $n \geq k$.

As an illustration, see figure 4 for $r = 1$ and for a random initial seed w. Corollary 1 explains the appearance of triangular clearings of 0s in the evolution of the CA and the fractal nature of the generated pattern (see figure 2a,b for $r = 1$ and $r = 2$, respectively).

5. Fractal generation by other CA

Another CA which generates regular behavior is the one with code 12. The reason why this CA shows recurrent behavior on seeds of the form $(00+11)^*$ is that it simulates XOR trellis automaton for radius 1, under the mapping $00 \to 0, 11 \to 1$. However, even when the CA is started on an arbitrary configuration, it seems that it always settles down to a configuration of the form $(00+11)^*$ and then generates a regular pattern as the XOR rules for trellises always produce regular evolution (see theorem 2 in section 6). In terms of physics, an arbitrary configuration can be seen as consisting of nice subwords or "phases" of the form $(00+11)^*$ separated by "defects" consisting of 01 and 10. It can be seen empirically that these two defects do not grow and mutually annihilate each other. Ultimately, the system settles down to a single phase without any defects. In figure 5a, the evolution pattern for code 12 on a random initial finite configuration is shown. The defects in the evolution, shown in figure 5b, exhibit Class-4 behavior. Another example is given in figure 5c whose defects, shown in figure 5d, disappear sooner. A similar observation for physical systems was made by Grassberger et al. [6]. The XOR trellis rule is also simulated by code 6 for radius 1, in a similar way. Code 30 for $r = 2$, in turn, simulates code 6 when restricted to configurations of the same form; code 38 for $r = 2$ simulates XOR CA rule for $r = 1$ (code 10).

Note that code 12 has the binary form 001100 and for code 30 it is 011110. Based on empirical observations, we propose the following conjecture.

Conjecture. Every totalistic CA with radius r and binary function code of the form $0^n1^{2m}0^n, n \geq 1, m \geq \epsilon_r \geq 1, 2(m+n) = 2r+2$, where ϵ_r is a constant depending on r, produces regular behavior for every finite initial configuration. This behavior shows some initial "defects" which eventually disappear. Empirical evidence suggests that $\epsilon_1 = \epsilon_2 = 1$ and $\epsilon_3 = 2$.

An interesting question is whether it is possible to exploit the regular evolution of Subclass-1 CA to compute the state of an arbitrary cell at an arbitrary time step in a way simpler than to run the CA. The answer to the question is positive. First of all, note that the evolution of a XOR CA rule on a configuration with a single 1 is fractal, and therefore it can be specified by a recursive rule. Hence, it is possible to compute the value of $(G^n_f(1))_i$ in lemma 3 in $\log n$ applications of the recursive rule, for any i. It follows then from the lemma, that we can compute $f^{(n)}$, and hence with an extra work linear in the length of initial configuration, the state of any cell can be computed at nth time step. Note that in some cases it is possible to have a closed form formula to compute $(G^n_f(1))_i$. For example, XOR trellis automaton for radius 1 generates on seed "one" the well-known Pascal's triangle of binomial coefficients in which an odd coefficient is replaced by 1 and an even one by 0.

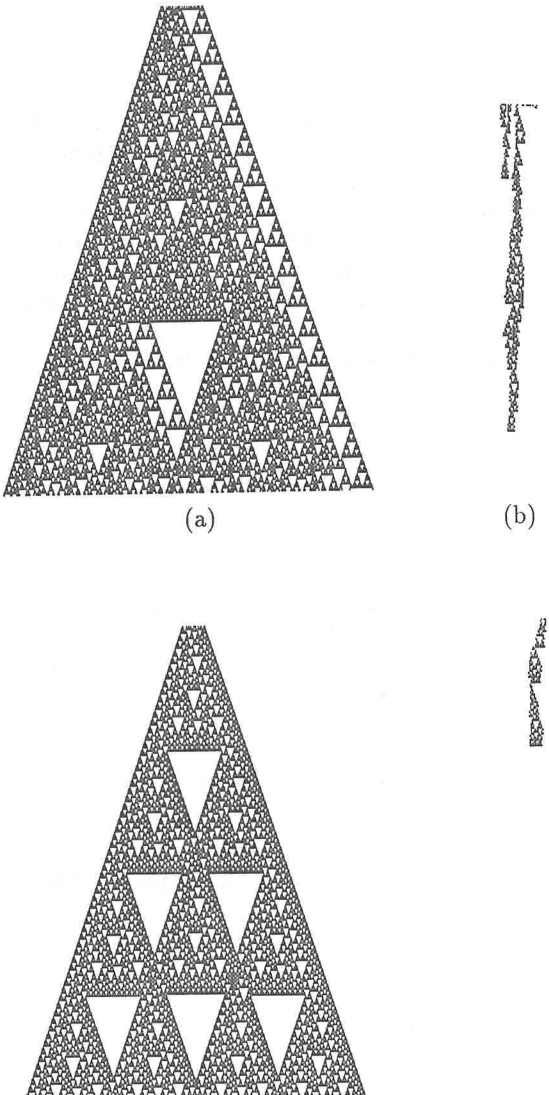
In the case of code 12, the situation is different. Here, one has to carry out the initial irregular part of the evolution in which the CA settles down to a configuration of the form $(00+11)^*$. Only after this settling down it is possible to exploit lemma 3.

6. Trellis automata

The results and observations of the previous two sections can be extended to trellis automata. Like in the case of CA, it is possible to classify trellis automata into four classes on the basis of their long-term evolution on random initial configurations (the example codes are for $r = 2$):

Class 1. codes 0, 4, 8, 16, 24, 28, 30.

Class 2. code 26.



(a) (b)

(c) (d)

Figure 5: Example evolutions of code 12, $r = 2$ and corresponding defects (see section 5).

Class 3. codes 6, 10, 12, 14, 18, 22.

Class 4. code 20.

It is interesting to note that code 20, which is in Class 4 for CA, $r = 2$, shows Class-4 behavior in the case of trellis automata too. See figure 6a and figure 6b for the evolution of code 26 and code 20 on random initial configurations. Note that figure 6a suggests that the evolution of this trellis leads to a set of separated simple stable states or periodic structures, which is Wolfram's characterization of Class 2. However, for a trellis they might appear as slanted stripes in the time-space diagram (unlike the vertical ones for CA).

When the initial configurations are restricted to random finite configurations, Class 3 can be subdivided in the following three types:

Subclass 1. codes 10, 14.

Subclass 2. codes 6, 12, 18.

Subclass 3. code 22.

The presence of code 10 in Subclass 1 is expected as it is the XOR trellis automaton rule (see figure 6c). An example evolution of code 14 is shown in figure 6d. Note that binary form of code 14 is 01110, which resembles those of code 12 and code 30 of CA classification. It is possible to generalize the conjecture proposed in section 5 to trellis rules with binary function codes of the form $0^n 1^{2m+1} 0^n$.

Moreover, we can prove the following theorem whose proof is along the same lines as that of theorem 1.

Theorem 2. *Let f be the XOR trellis automaton rule for radius r. Then, for all n, where n is a power of 2,*

$$G_f^n(w) = w^{2r}$$

where w is any finite configuration of length n.

The above theorem, which explains the regular pattern in figure 6c, leads immediately to the following corollary, which is analogous to corollary 1.

Corollary 2. *Let w be a seed that starts and ends with 1, and let $2^{k-1} < |w| \leq 2^k$ for some $k > 0$. Then, if f is the XOR trellis automaton rule for radius r,*

$$G_f^{2^n}(w) = (w0^{2^n-|w|})^{2r-1}w$$

for all $n \geq k$.

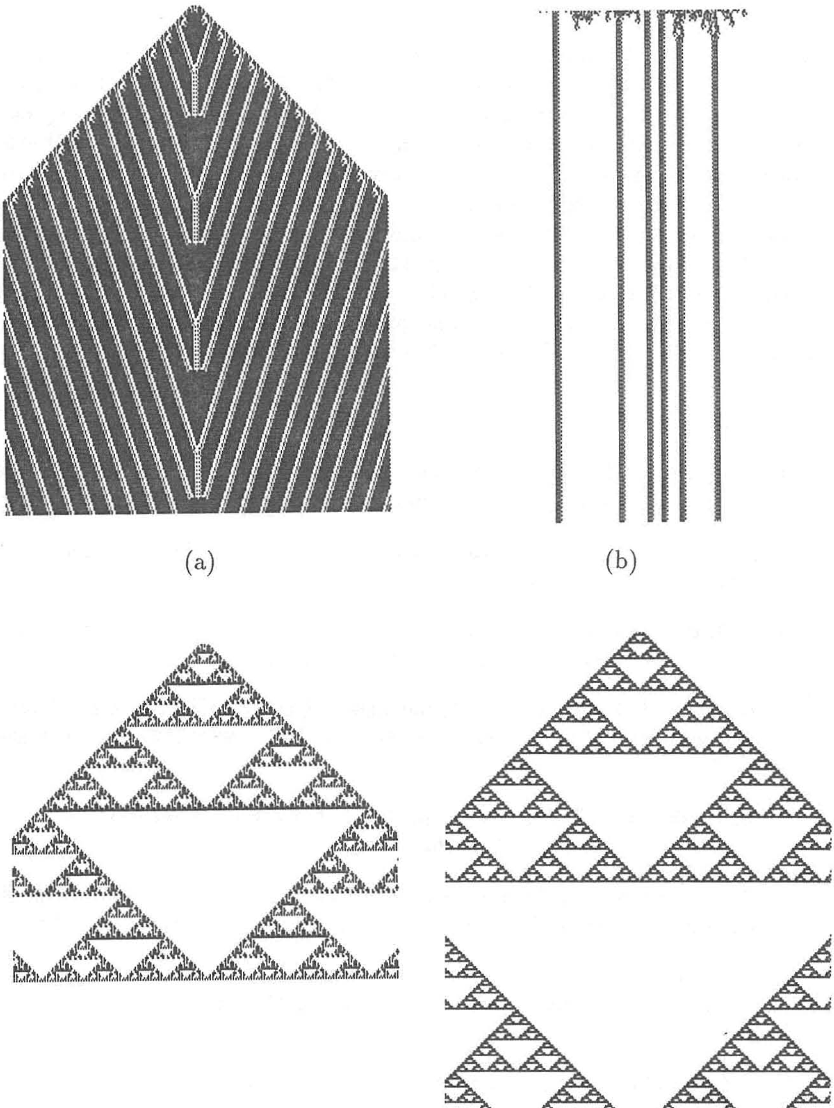


Figure 6: Examples of trellis rules for $r = 2$ (a) code 26, (b) code 20, (c) code 10, (d) code 14.

7. Conclusions

We investigated the fractal and recurrent behavior of linear totalistic CA and of trellis automata. We have proved that XOR (additive mod 2) rules will always generate a highly regular evolution pattern for every size of neighborhood. This regular evolution allows one to compute the value of a cell after arbitrary number of time steps in a more efficient way than to actually run the CA. We have convincing empirical evidence and some mathematical explanation but no proof that the CA and trellis rules with the binary function code of the form $0^n1^{2m}0^n$, e.g., rule 12 for CA with $r = 2$, give always an evolution that, after possible initial "defects," stabilizes to a highly regular pattern. As the defects in the initial part of the evolution show the Class-4 behavior, it might be difficult to prove that they always disappear.

References

[1] J. Albert and K. Culik II, "Simple universal cellular automaton and its one-way and totalistic version," *Complex Systems*, **1** (1987) 1–16.

[2] A. Barbe, "Periodic patterns in the binary difference field," *Complex Systems*, **2** (1988) 209–233.

[3] C. Choffrut and K. Culik II, "On real-time cellular automata and trellis automata," *Acta Informatica*, **21** (1984) 393–407.

[4] K. Culik II, J. Gruska, and A. Salomaa, "Systolic trellis automata," *International Journal of Computer Mathematics*, **15** (1984) 195–212; **16** (1984) 3–22.

[5] K. Culik II and S. Yu, "Undecidability of CA classification schemes," *Complex Systems*, **2** (1988) 177–190.

[6] P. Grassberger, F. Krause, and T. von der Twer, "A new type of kinetic critical phenomenon," *J. Physics A: Math. Gen.*, **17** (1984) L105–L109.

[7] O.H. Ibarra and S.M. Kim, "Characterizations and computational complexity of systolic trellis automata," *Theoretical Computer Science*, **29** (1984) 123–154.

[8] B.B. Mendelbrot, *The Fractal Geometry of Nature* (Freeman, San Francisco, 1982).

[9] T. Toffoli and N. Margolus, *Cellular Automata Machines* (MIT Press, Cambridge, 1987).

[10] S.J. Wilson, "Cellular automata can generate fractals," *Discrete Applied Mathematics*, **8** (1984) 91–99.

[11] S.J. Wilson, "Computing fractal dimensions for additive cellular automata," *Physica*, **24D** (1987) 190–206.

[12] S.J. Wilson, "The equality of fractal dimensions for certain cellular automata," *Physica*, **24D** (1987) 179–189.

[13] S.J. Wilson, "Growth rates and fractal dimensions in cellular automata," *Physica*, **10D** (1984) 69–74.

[14] S. Wolfram, *Theory and Applications of Cellular Automata* (World Scientific, Singapore, 1986).

[15] S. Wolfram, "Universality and complexity in cellular automata," *Physica*, **10D** (1984) 1–35.