Journal of Statistical Physics, Vol. 68, Nos. 5/6, 1992

Phenomenology of Nonlocal Cellular Automata

Wentian Li[1,2]

Received February 5, 1991; final March 2, 1992

Dynamical systems with nonlocal connections have potential applications to economic and biological systems. This paper studies the dynamics of nonlocal cellular automata. In particular, all two-state, three-input nonlocal cellular automata are classified according to the dynamical behavior starting from random initial configurations and random wirings, although it is observed that sometimes a rule can have different dynamical behaviors with different wirings. The nonlocal cellular automata rule space is studied using a mean-field parametrization which is ideal for the situation of random wiring. Nonlocal cellular automata can be considered as computers carrying out computation at the level of each component. Their computational abilities are studied from the point of view of whether they contain many basic logical gates. In particular, I ask the question of whether a three-input cellular automaton rule contains the three fundamental logical gates: two-input rules AND and OR, and one-input rule NOT. A particularly interesting "edge-of-chaos" nonlocal cellular automaton, the rule 184, is studied in detail. It is a system of coupled "selectors" or "multiplexers." It is also part of the Fredkin's gate—a proposed fundamental gate for conservative computations. This rule exhibits irregular fluctuations of density, large coherent structures, and long transient times.

KEY WORDS: Nonlocal cellular automata; automata networks; classification of cellular automata; cellular automata rule space; critical hypersurface; self-organized criticality; mean-field theory; universal computation; "game of life"; Fredkin's gate; coupled selectors or coupled multiplexers; edge-of-chaos dynamics; density fluctuations; long transient behaviors; cooperative dynamics.

1. INTRODUCTION

1.1. Why Study Nonlocal Cellular Automata?

It is known that many systems with a large number of interactive components can exhibit emergent properties—including dynamical

[1] Santa Fe Institute, Santa Fe, New Mexico 87501.
[2] Current address: Box 167, Rockefeller University, 1230 York Avenue, New York, New York 10021.

0022-4715/92/0900-0829$06.50/0

behaviors—that are not deducible from their components. There could be a whole spectrum of emergent properties and dynamical behaviors, crucially depending on which rule is used. Dynamical behavior as a function of the form of the rule is actively studied under the names of bifurcation theory and the structure of the rule space.

There is another factor which determines the dynamical behavior of a system, that is, the way components are coupled to each other. In the modeling of the physical world, there are not many choices for the form of the coupling because the interaction is always local; consider, for example, the molecular interaction in a fluid. Even if some long-range correlations exist, they are caused by the accumulation and propagation of the local interactions, and there is no need for introducing nonlocal connections explicitly at the low-level description of the system.

Dynamical behavior of systems with local interactions is studied in partial differential equations, coupled ordinary differential equations,[60] coupled map lattices,[9] and cellular automata.[58, 61, 71] It is well understood now that by moving from one "polar point" to another in the rule space, a series of dynamical behaviors can be exhibited by the system. A typical such series includes the fixed-point dynamics (laminar phase), periodic dynamics, locally chaotic dynamics, various types of "edge-of-chaos" dynamics (spatiotemporal intermittency, breaking up of the domain walls, complex glider interactions, etc.), and the global chaotic dynamics (turbulent dynamics).

For systems with a nonlocal connection, the dynamics does not necessarily differ from that of the locally-connected system, as will be shown in this paper. This is especially true if the locally-connected system already exhibits a globally chaotic dynamics. Nevertheless, the modes of "edge-of-chaos" dynamics in the nonlocal systems are expected to change drastically from those in local systems, because in the latter case the slow propagation of perturbation in *space* is important in separating them from either the regular or the chaotic systems, whereas the concept of space is completely modified in nonlocal systems. One purpose of this paper is to identify the edge-of-chaos dynamics in nonlocal systems.

The study of dynamical behaviors in nonlocally coupled systems is not purely academic. There could be many possible applications in the real world. For example, since the basic assumption of locality in physics is violated in economics and biology—considering how neurons are connected in the brain and how information concerning the price of a stock is shared by agents who read the same tickertape—these fields could be good candidates for applications of the nonlocal systems studied here.

To start the study of dynamical behaviors in nonlocal systems, I will concentrate on a particular class of systems: the *nonlocal cellular automata.*

This name is used to identify a subset of the *network* or *automata network* by the following three requirements: (a) the space, time, and state values of each component are discrete (and the state value is usually finite); (b) the dynamical rule applied to one component is the same as those applied to all other components; (c) the updating of the state value of every component is synchronized. In brief, the definition of nonlocal cellular automata is almost the same as that of usual cellular automata except that here nonlocal connection is allowed.

If the requirement (b) is violated, we have, as examples of both nonlocal and local connections, the random networks studied by Kauffman[26, 27] and the inhomogeneous cellular automata.[62] If the requirement (c) is violated, we have, again as examples of both nonlocal and local connections, many neural network models[49] and statistical-physics-motivated models.[43, 44] Asynchronous updating seems to be a more realistic description of brain activities. The dynamics for these systems should also be interesting, but will not be covered in this paper.

Another feature of the systems studied in this paper is that they are sparsely connected instead of fully connected. If n represents the number of inputs each component received, n is equal to 3 throughout the paper. This restriction may not make the system a good model for, e.g., the stock market, because the interaction among agents in a stock market is usually global. For some recent studies of globally coupled maps or oscillators, see refs. 22, 23, 69, 59, and 45.

The only previous study of which I am aware on similar systems was done by Walker and Ashby.[63–67] They also consider the $n=3$ case exclusively. Nevertheless, it is required that one of the three inputs is from the component itself. No special name is given by Walker to the system with this particular requirement; sometimes these are called "Ashby nets," other times they are loosely referred to as "a class of complex binary nets" or "sparsely connected Boolean nets." To be consistent, I will call them *partially-local cellular automata*, to distinguish them from the *fully-nonlocal cellular automata*. Various types of connection will be further discussed in the next subsection.

1.2. Wiring Schemes of Nonlocal Connection

Considering a system with N components, each component i has a state value x_i^t at time t ($i=1, 2,..., N$). A three-input ($n=3$) rule $f(\cdot)$ can be specified in the form

$$x_i^{t+1} = f(x_{j1(i)}^t, x_{j2(i)}^t, x_{j3(i)}^t) \qquad (1.1)$$

where $j_1(i)$, $j_2(i)$, and $j_3(i)$ are somehow randomly chosen from among the N components.

I identify four different types of wiring schemes, i,e., how $j_1(i)$, $j_2(i)$, and $j_3(i)$ are chosen (illustrated in Fig. 1). For comparison, the local connection is also listed:

(0) Local connection: $j_1(i) = i - 1$, $j_2(i) = i$, $j_3(i) = i + 1$. The first input is the left neighbor, the second input is the site itself, and the third input is the right neighbor.

(1) Partially-local connection: $j_2(i) = i$, but $j_1(i)$ and $j_3(i)$ are randomly chosen. In practice, one simply generates two indices randomly

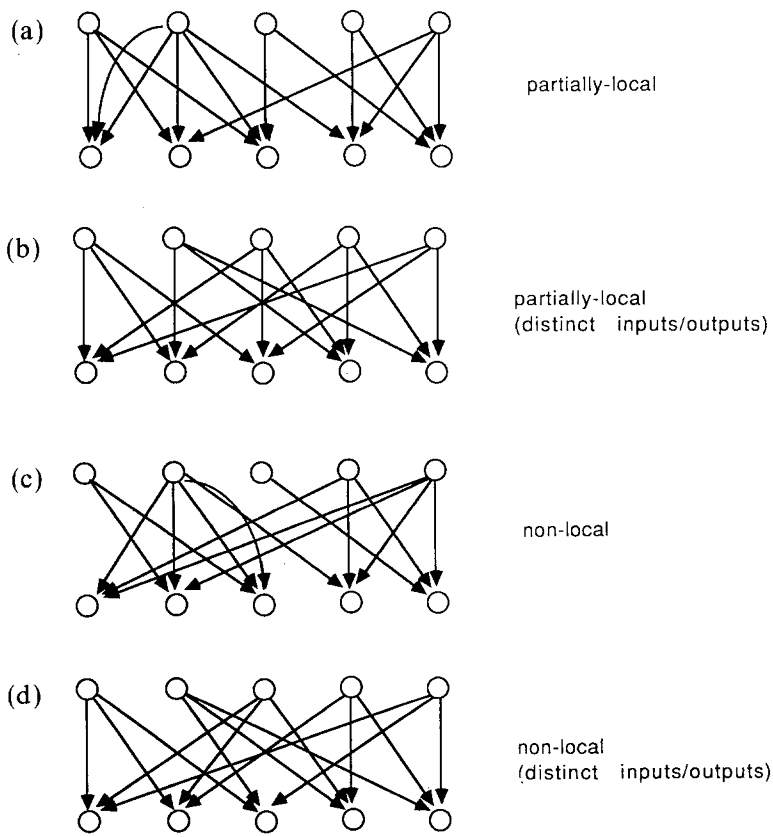


Fig. 1. Illustration of the four possible wiring schemes of nonlocal connection. (a) Partially-local connection with possibly degenerate inputs; (b) distinct-input partially-local connection; (c) fully-nonlocal connection with possibly degenerate inputs; (d) distinct-input fully-nonlocal connection.

between 1 and N for each i, then assigns the two as the first and the third inputs for that size i. Degeneracy of inputs might be possible, whenever two among the three inputs are the same. For example, in Fig. 1a, site $i=1$ obtains two inputs from site $i=2$.

(2) Distinct-input, partially-local connection: $j_2(i)=i$, $j_1(i)$ and $j_3(i)$ are randomly chosen, but it is checked that none of the two inputs among the three inputs for site i are the same. In practice, one generates two indices randomly between 1 and N, then compares the two indices as well as the index i. If two of them are the same, regenerate another index again randomly to replace one of the degenerate indices. Repeat the comparison until none of the two are the same.[3]

(3) Fully nonlocal connection: all inputs $j_1(i)$, $j_2(i)$, and $j_3(i)$ are randomly chosen. Again, degeneracy of inputs might be possible whenever two of the three inputs happen to be the same.

(4) Distinct-input, fully nonlocal connection: all inputs $j_1(i)$, $j_2(i)$, and $j_3(i)$ are randomly chosen, and it is checked that none of the two inputs among the three inputs for site i are the same.

It is important to distinguish the partially-local connection from the fully-nonlocal one, because the former can still share some features with the locally-connected dynamical systems—for example, the single-site "domain-wall"—whereas the latter does not have anything similar.

It is also useful to distinguish the case of a degeneracy-permitted connection from that of a distinct-input connection. If, for example, the first input is identical with the second input, the part of the rule table which specifies the updating of the state value when $x^t_{j_1(i)} \neq x^t_{j_2(i)}$ will never be used. On the other hand, the distinct-input case makes full use of the rule table. Intuitively, the difference between cases (1) and (2), and between (3) and (4), should be diminishingly small when the system size is large. Nevertheless, it has been observed that for some rules, sample statistical quantities (e.g., average cycle length) are always different between the degeneracy-permitted and distinct-inputs connections even in the infinite-size limit.

Finally, note that there is a difference between *all possible* nonlocal connections and a *typical* nonlocal connection. The first case should also include the local connection as a special case, whereas the second case refers to a typical realization of the nonlocal wiring using some random number generators.

[3] It is possible, though, that the last remaining site does not have any choice but to choose the degenerate inputs. In this case, one has to restart the selection process.

1.3. Topics To Be Discussed in This Paper

This paper studies the dynamical behavior of two-state, three-input nonlocal cellular automata, and compares them with those of the corresponding local cellular automata. Special attention will be paid to observations which lead to new concepts that are absent or less important in locally-connected dynamical systems. I will discuss three of them: (1) the "cleanness" of the rule space parameterized by mean-field parameters; (2) computation at the level of each component; and (3) new modes of "edge-of-chaos" dynamics.

The first topic is about how rules with different dynamical behaviors are distributed in the cellular automata rule space. As studied by Langton, Packard, and Li,[31, 32, 35–37] it is now well understood that if all the cellular automata rules are organized in the rule space with some appropriate choice of the distance such as the Hamming distance between two rule tables, the rules with similar dynamical behavior tend to reside in the same region of the rule space.

The boundary between two regions with different dynamical behaviors plays the role of the critical point in the phase transition (though the dimensionality of this boundary is much higher than a typical critical "point" that we are familiar with in statistical physics). In the large-rule-space limit, the presumably discrete jump from one rule to another is almost continuous. In this limit, one can classify the transition from one dynamical behavior to another as being either first order (if the change of dynamics is sudden) or second order (if the change of dynamics is smooth).

This study of the structure of the rule space can also be viewed as a multiparameter, high-dimensional analog of the study of bifurcation phenomena, which is done mostly in single-parameter, lower-dimensional dynamical systems. Notice though that one should distinguish the dimension of the rule space (the number of parameters) with that of the real space (the number of components). Since high-dimensional systems usually need more parameters to specify, the dimensionality of the rule space may also become larger.

It has been observed that the boundary separating periodic and chaotic dynamics in the local cellular automata rule space is highly rugged, and defies an easy parametrization.[37] The simplest parametrization, called the λ-parametrization, is by varying a single parameter which, in the case of a binary state, is the fraction of entries in the rule table that are equal to state 1.[30] The transition from periodic to chaotic dynamics can occur at different points on the λ axis, although it is suggested that the transition might be made sharper in the λ-parametrization by increasing the number of states.[76]

In the next simplest parametrization, the mean-field parametrization, more parameters are used, which not only count the fraction of the configurations which are mapped to state 1, but also examine what type of configurations are mapped to 1. The mean-field parametrization improves the description of the transition surface or critical surface, but it still fails frequently to make the correct predictions on the behavior of local cellular automata.[4] The study of the nonlocal cellular automata rule space will show that mean-field parametrization gives a far better prediction of the location of the transition surface. A similar observation that mean-field theory describes nonlocal systems much better than local systems is also made in ref. 56.

The second topic is about nonlocal cellular automata as computers. There has been much discussion about local cellular automata as computers, such as the two-dimensional, two-state, nine-input rule "game of life."[(2)] This rules as well as many other local cellular automata can carry out any computation because as certain moving local patterns ("gliders") emerge and interact with other local patterns, basic logical functions (AND, OR, NOT) can be simulated. Since any computational task can be coded as a series of basic logical functions, these rules are equivalent to the universal computer. It is known that the number of local cellular automata rules which are equivalent to the universal computer is small. Most rules cannot carry out universal computation.

The situation is quite different for nonlocal cellular automata. Since any two components can be connected directly, there is no need to produce gliders that simulate basic logical functions. In fact, a rule is a universal computer as long as it contains the basic building blocks of all logical functions, for example, (AND, NOT), or (OR, NOT), or (NAND, COPY),[5] or (NOR, COPY). This condition is easily satisfied by many two-state, three-input nonlocal cellular automata. As a result, there are many more universal computers in nonlocal cellular automata than in local ones.

On the other hand, being a universal computer does not mean it is "easy" to carry out a specific computation. We all know how slow and clumsy it is for a Turing machine to add two numbers together. The situation is similar for nonlocal cellular automata. A rule which contains (AND, NOT, OR) will carry out a computational task more easily than a rule containing only (AND, NOT), though both of them are universal computers. Then we can ask which rules contain the largest number of logical functions. This tells us something about the computational ability of the rule.

[4] A discussion of other parametrization schemes for local cellular automata rule space is given in ref. 33.

[5] NAND ≡ NOT(AND), and NOR ≡ NOT(OR).

The last topic to be discussed in some detail is a new mode of the "edge-of-chaos" dynamics[6] in nonlocal cellular automata. The edge-of-chaos dynamics in locally-connected dynamical system is studied in several systems, such as the spatiotemporal intermittency in coupled map lattices, the glider activity in "class-4"[(72)] cellular automata, etc. In these systems, small but positive Lyapunov exponents, long-range correlations, long transients, and poor convergence of the statistical quantities are all considered as the hallmarks of the edge-of-chaos dynamics. In nonlocal systems, the criteria of the long transients and the poor convergence of the statistical quantities are still applicable, but is is difficult to define long-range correlation, as well as any other concepts related to the space. Here the concept of "coherent structure" is used, and some preliminary calculations are carried out to identify the existence of such cooperation among components.

This paper is organized as follows: Section 2 reviews the basic results on two-state, three-input local cellular automata. Section 3 discusses the corresponding cellular automata with partially-local and fully-nonlocal connections; the dynamics is classified, and the rule space with mean-field parametrization is shown. Section 4 attempts an understanding of the rule space in the mean-field parametrization discussed in the previous section. Section 5 discusses the computational abilities for nonlocal cellular automata rules, especially whether they contain the basic logical gates AND, OR, and NOT. Section 6 studies what I consider to be the most interesting two-state, three-input nonlocal cellular automaton, the rule 184, also called "coupled selectors" or "coupled multiplexers," and discusses various aspects of its dynamics.

2. REVIEW OF ELEMENTARY LOCAL CELLULAR AUTOMATA

The simplest cellular automata (excluding the trivial zero-input and one-input cases) are those with two inputs and two states. Since there are 2^2 different input configurations, each configuration can be mapped to either 0 or 1, the total number of possible rules is $2^{2^2} = 16$. The names of these 16 rules are listed in, e.g., refs. 13 and 68. Actually, the number of independent two-state, two-input rules is only seven. Out of these seven rules, one is actually a zero-input rule and two are one-input rules. So the number of independent "true" two-input rules is four. Out of these four rules, one is chaotic ("exclusive or," or XOR) and three are nonchaotic.

[6] The name "edge-of-chaos" was first used by Packard;[(48)] other names, such as the complex dynamics, critical dynamics, boundary dynamics, in-between dynamics, etc., can also be used.

Such a rule space is too small to study the generic structure of rule space in general.

The next simplest cellular automata are those with two states and three inputs, also called *elementary rules* in ref. 71. I will use the same name in this paper. The number of all possible elementary rules is $2^{2^3} = 256$, and the number of independent rules is actually 88.[63, 71, 36] There are 14 rules that are actually zero-, one-, or two-input rules,[7] so the number of independent "true" three-input rules is 74. The elementary cellular automata with local connections are extensively studied in refs. 17, 20, 33, 36, 70, 71, 73, and 75, among others. This section will review some of the most relevant properties of these rules. For more detail, see the original publications.

2.1. Rule Tables, Mean-Field Parameters, and λ Parameter

As shown by Eq. (1.1), a cellular automaton rule is specified when the value of x_i^{t+1} is given for all possible input configurations. With two states and three inputs, the number of the input configurations is $2^3 = 8$. I write

$$\begin{aligned} &a_0 = f(0, 0, 0), \quad a_1 = f(0, 0, 1), \quad a_2 = f(0, 1, 0), \quad a_3 = f(0, 1, 1) \\ &a_4 = f(1, 0, 0), \quad a_5 = f(1, 0, 1), \quad a_6 = f(1, 1, 0), \quad a_7 = f(1, 1, 1) \end{aligned} \tag{2.1}$$

or equivalently

$$\begin{aligned} &000 \to a_0, \quad 001 \to a_1, \quad 010 \to a_2, \quad 011 \to a_3 \\ &100 \to a_4, \quad 101 \to a_5, \quad 110 \to a_6, \quad 111 \to a_7 \end{aligned} \tag{2.2}$$

to specify the rule.

The rule is completely determined by these a_is, which can be represented either by a *rule table*

$$RT \equiv \{a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0\}$$

or by a *rule number*, which is the decimal representation of the above binary string[71]:

$$RN \equiv \sum_{i=0}^{7} a_i 2^i$$

For example, the rule number for tule table $\{0, 0, 0, 0, 0, 0, 0, 1\}$ is 1; and that for rule table $\{1, 0, 0, 0, 0, 0, 0, 1\}$ is 129.

[7] The rule numbers (to be explained in this section) for these rules are: (1) zero-input: 0; (2) one-input: 15, 51, 170, 204; and (3) two-input: 3, 5, 10, 12, 34, 60, 90, 136, 160.

Since it is not important which state is labeled as 0 and which as 1, interchanging 0 and 1 leads to an equivalent rule:

$$\{\overline{a_0}, \overline{a_1}, \overline{a_2}, \overline{a_3}, \overline{a_4}, \overline{a_5}, \overline{a_6}, \overline{a_7}\}$$

where $\overline{0} = 1$ and $\overline{1} = 0$. Also, since the space is not directional, interchanging left and right inputs also leads to an equivalent rule:

$$\{a_7, a_3, a_5, a_1, a_6, a_2, a_4, a_0\}$$

Finally, applying both interchanging operations mentioned above, one has the third equivalent rule:

$$\{\overline{a_0}, \overline{a_4}, \overline{a_2}, \overline{a_6}, \overline{a_1}, \overline{a_5}, \overline{a_3}, \overline{a_7}\}$$

The number of independent rules, 88, is derived by checking all possible equivalence relations among the rules.

A rule table gives complete information about the rule being represented. Nevertheless, sometimes one might not want to know all the information, and consider several rules to be "more or less" the same. This is where the mean-field parameters come in.[53, 18, 36] In the case of elementary cellular automata, the entries a_1, a_2, and a_4 in the rule table may be considered to play a similar role, because their input configurations all contain one 1 and two 0's. Similarly, the entries a_3, a_5, and a_6 are all related to the input configurations containing two 1's and one 0. The mean-field parameters are defined by

$n_1 \equiv$ number of bits among a_1, a_2, and a_4 that are equal to 1

$n_2 \equiv$ number of bits among a_3, a_5, and a_6 that are equal to 1

and for a similar reason,

$$n_0 \equiv a_0$$

$$n_3 \equiv a_7$$

Now instead of a rule table, we have a mean-field cluster $\{n_0, n_1, n_2, n_3\}$ which contains several rule tables that are "similar" to each other.

It can be easily checked that relabeling 0 and 1 transforms the mean-field cluster $\{n_0, n_1, n_2, n_3\}$ to $\{1-n_3,\ 3-n_2,\ 3-n_1,\ 1-n_0\}$, while the interchange of left and right inputs does not change the mean-field parameters.

An even cruder piece of information about a rule table is the number of entries in the rule table that are equal to 1 (for binary states case). It is the so-called λ parameter:[30]

$\lambda \equiv$ number of bits among $\{a_i\}$ $(i = 0, 1, ..., 7)$ that are equal to 1

One can also define the normalized λ parameter as the fraction of these entries (I will use the same symbol for both unnormalized and normalized λ parameters. The context should make it clear which one is used.) The information about a rule table provided by the λ parameter seems to be almost minimum. But satisfyingly, λ is a reasonably good indicator of how regular or how chaotic the dynamics is. The relationship between the λ parameter and the mean-field parameters is

$$\lambda = n_0 + n_1 + n_2 + n_3$$

2.2. Classification of the Local Cellular Automata Rules

The dynamics of a cellular automaton rule typically refers to dynamical behavior exhibited by the rule when starting from a random initial configuration. If one starts from a special initial configuration, for example, $x_i = 0$ for all i's, only the function $f(0, 0, 0)$, or the entry a_0 in the rule table, is used. Consequently, the dynamics as started from this special initial configuration will not characterize the generic behavior of the rule. In other words, the "democracy" of all entries in the rule table must be guaranteed.

One simple classification of all elementary rules is the following (see also refs. 72 and 36):

1. Null rules: the limiting configuration is all 0's or all 1's.
   0, 8, 32, 40, 128, 136, 160, 168.
2. Fixed-point rules: the limiting configuration is invariant by applying the updating rule (with possibly a spatial shift; if this is the case, the rule is marked by an asterisk), excluding all-0's or all-1's configurations.
   2*, 4, 10*, 12, 13, 24*, 34*, 36, 42*, 44, 46*, 56*, 57*, 58*, 72, 76, 77, 78, 104, 130*, 132, 138*, 140, 152*, 162*, 164, 170*, 172, 184*, 200, 204, 232.
3. Two-cycle rules: the limiting configuration is invariant by applying the updating rule twice (with possibly a spatial shift; if this is the case, the rule is marked by an asterisk). The dynamics for rules 14 and 142 can also be fixed point with a shift for some initial conditions.
   1, 3*, 5, 6*, 7*, 9*, 11*, 14*, 15*, 19, 23, 25, 27*, 28, 29, 33, 35*, 37, 38*, 43*, 50, 51, 74*, 108, 134*, 142*, 156, 178*.
4. Periodic rules: the limiting configuration is invariant by applying the updating rule L times, with the cycle length L either independ-

ent or weakly dependent on the sequence length (in the latter case, one could introduce a subclass). In particular, rules 131 and 133 typically exhibit local three-cycle dynamics, and their global cycle length can either be 3 or 6. Rule 73 has regions with chaotic behavior and can be called a locally-chaotic rule.[36]
26, 41, 73, 131 (or 62), 133 (or 94), 154.

5. Edge-of-chaos rules ("complex rules" or "boundary rules"): although their limiting dynamics may be periodic, the transient times reaching the limiting configuration can be extremely long, and they typically increase at least linearly with the system size. One hallmark of this class of rules is its marginal stability (or instability) with respect to perturbations, and another is its poor convergence of any statistical property such as the transient time.
54, 137 (or 110).

6. Chaotic rules: nonperiodic dynamics. They are characterized by the exponential divergence of the cycle length with the system size, and the instability with respect to perturbations. The transient time can either be long or short.
18, 22, 30, 45, 60, 90, 105, 106, 129 (or 126), 146, 150, 161 (or 122).

The rule number inside parentheses is the representative rule number used in ref. 75, which is the smallest value among all rule numbers of equivalent rules. The representative rule number used in this paper as well as ref. 36 is the one with the smallest λ parameter value. If several equivalent rules have the same λ value, then pick the one with the smallest rule number.

There can be many different classification schemes, depending on the degree of coarse graining. The crudest classification scheme might be the one which only distinguishes chaotic and nonchaotic rules. Most of the rules are easy to classify whatever the classification scheme used. For example, a rule is classified as being chaotic by any of the following criteria: large spatiotemporal entropy, positive expansion rate of perturbation, absence of periodic dynamics in the infinite-system-size limit. On the other hand, the classification of the edge-of-chaos rules is destined to be difficult. Take rule 137 (or 110), for example; its cycle length for the limiting dynamics with a finite system size is much shorter than a typical chaotic rule, so in some sense it belongs to the class of periodic rules. Nevertheless, a perturbation in rule 137 usually spreads in space (though slowly), so it is similar to a chaotic rule. The notable examples of the "hard-to-classify" elementary rules are rules 73, 54, and 137 (or 110).

2.3. Rule Space with the Mean-Field Parametrization

When all cellular automata rules are organized in a single space with an appropriate choice of the distance between two rules, the space is called a *rule space*. The standard measure of the distance between two integer sequences of the same length is the Hamming distance, which is the sum of the differences (always nonnegative) between two values at the two corresponding sites of the two sequences. For example, the Hamming distance between $\{0, 0, 0, 1, 1, 0, 0, 0\}$ and $\{1, 0, 0, 1, 1, 0, 1, 0\}$ is two.

The dimension of the elementary cellular automata rule space is equal to eight, and the distance along each dimensional axis is one, so it is an eight-dimensional hypercube. The location of a rule in the rule space can be considered part of the *genotype* of the rule, and dynamical behavior exhibited by the rule is the *phenotype*.[35, 36] How genotype determines phenotype describes the structure of the rule space, which is studied in ref. 36.

Here I will review the result about rule space with mean-field parametrization. Although this parametrization is not that perfect for local rules, it gives an extremely good picture of the structure of the fully-nonlocal rule space (to be discussed in later sections). The dimension of the space is four, since each mean-field cluster is labeled by four parameters $\{n_0, n_1, n_2, n_3\}$. The distance along n_0 and n_3 is 1, and the distance along n_1 and n_2 is 3.

The picture of a four-dimensional space is still difficult to draw, and I will fix two parameters (n_0 and n_3), and vary the other two (n_1 and n_2). There are good reasons to fix n_0, n_3 instead of n_1, n_2. First of all, there are only two possible values for n_0 and n_3. Second, these parameter values are more important in determine the dynamical behavior of the rule. Actually, the entries a_0 $(=n_0)$ and a_7 $(=n_3)$ of the rule table are called "hot bits" in ref. 36.

There are four possibilities for the n_0, n_3 values: (1) $n_0 = n_3 = 0$; (2) $n_0 = 0$, $n_3 = 1$; (3) $n_0 = 1$, $n_3 = 0$; and (4) $n_0 = n_3 = 1$. The last case is equivalent to the first case by interchanging 0 and 1, which leaves three independent "slices" of the rule space. The rule clusters in the first slice $\{n_0, n_1, n_2, n_3\} = \{0, *, *, 0\}$ are called "nonlinear clusters" in ref. 36 because the x_i^{t+1} versus $\sum_{k=1}^{3} x_{jk(i)}^t$ plot looks like a nonlinear logistic map (that is, both low-density and high-density configurations lead to state 0). Those in the second slice $\{0, *, *, 1\}$ are called "linear clusters," and those in the third slice $\{1, *, *, 0\}$ are called "inversely linear clusters."[36] (The wild card symbol $*$ can be either 0 or 1.) The n_1 and n_2 are increased from 0 to 3 along the two parameter axes. See Figs. 2a–2c for the three slices of the rule space.

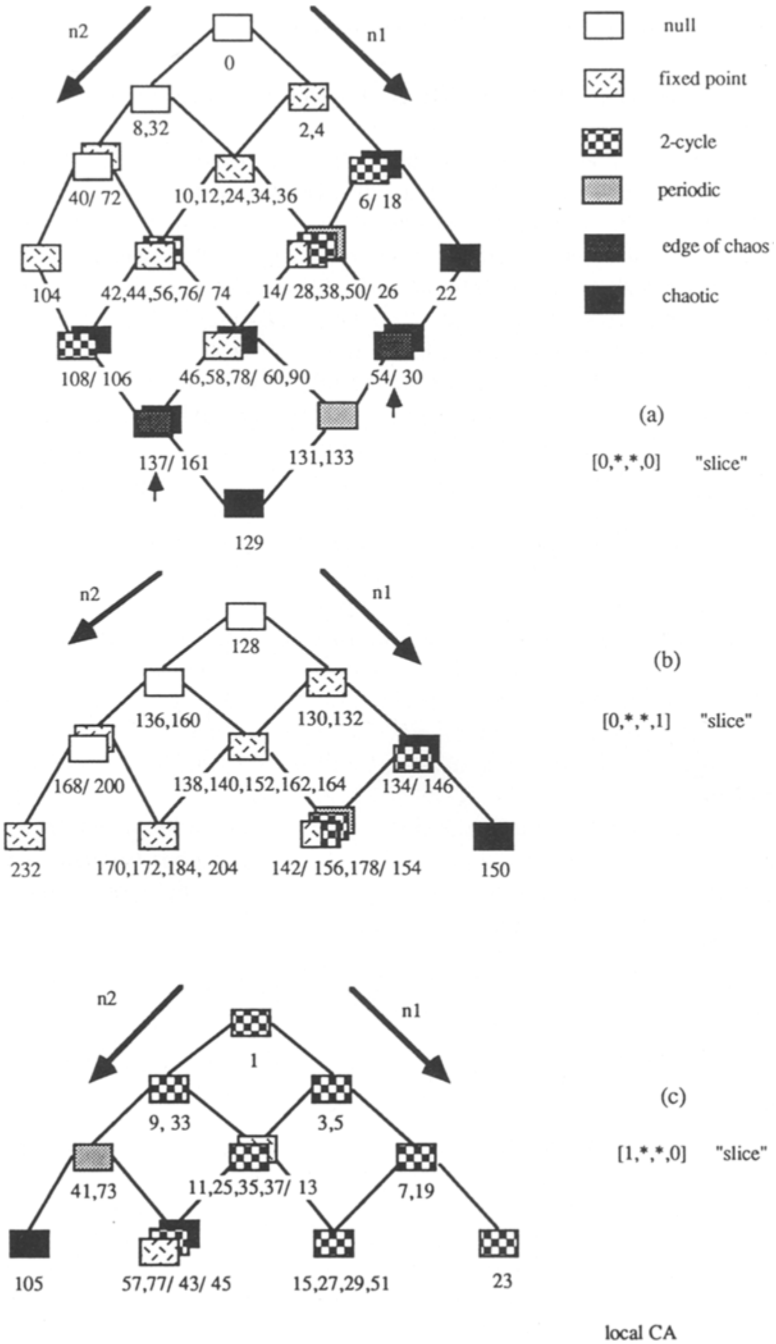


Fig. 2. The rule space for local cellular automata with mean-field parametrization. The rule numbers contained in each mean-field cluster are listed. The second and the third mean-field parameters n_2 and n_3 vary from 0 to 3. The rules with long transients (rules 54 and 137) are marked. (a) The slice of the rule space containing "nonlinear clusters" with $n_0 = 0$ and $n_3 = 0$. (b) The slice of the rule space containing "linear clusters" with $n_0 = 0$ and $n_3 = 1$. (c) The slice of the rule space containing "inversely linear clusters" with $n_0 = 1$ and $n_3 = 0$.

Usually, each cluster contains more than one rule, and each dynamical behavior exhibited by a rule in that cluster is indicated by a different texture of the block. The darker the texture, the more chaotic the dynamics. Sometimes, rules in a mean-field cluster all have the similar dynamical behavior; for example, the cluster $\{0, 1, 1, 1\}$, which contains fixed-point rules 138, 140, 152, and 162. Other times, the mean-field description is not that good; see, for example, the cluster $\{1, 1, 2, 0\}$, which contains rules 57, 77 (fixed-point dynamics), rule 43 (two-cycle dynamics), and rule 45 (chaotic dynamics).

There are several general observations of Fig. 2:

1. The slice with nonlinear clusters contains rules of all types of dynamical behavior. Roughly speaking, by moving from the upper left part (small n_1) to the lower right part (large n_1) of Fig. 2a, one observes the familiar bifurcation to chaos series reminiscent of that in the logistic map.[46] Especially, rules 131, 133 with local cycle length equal to 3 are located inside the chaotic regime. Whether this is an accidental event or not, the similarity with the three-cycle window in the logistic map is striking.
2. The slice with the linear clusters contains mostly fixed-point rules, with the exception of the lower right corner of Fig. 2b.
3. The slice with the inversely-linear clusters contains mostly two-cycle rules, with the exception of the lower left corner of Fig. 2c.
4. The λ parameter is increased by moving from top to bottom. It is clear that although it points to the correct direction from regular dynamics to chaotic dynamics, it is not exactly perpendicular to the bifurcation plane.

These observations will be further discussed in Section 4.

3. ELEMENTARY CELLULAR AUTOMATA WITH PARTIALLY-LOCAL AND FULLY-NONLOCAL CONNECTIONS

Before going into the discussion of nonlocal cellular automata, let me summarize what we know about local cellular automata:

1. For null rules, it is usually the case that a large percentage of the three-input configurations are mapped to the same state value such as 0, and the three-input configuration 000 also leads to 0, so the all-0's configuration is the attracting invariant configuration.
2. For inhomogeneous fixed-point rules, some nonzero state can survive due to updating rules like $010 \rightarrow 1$. At the mean time, most

other three-input configurations are mapped to 0, so the dynamics is static but the spatial configuration is not all 0's.

3. For two-cycle rules, there is typically an alternating activation of the three-input configurations. For example, if $000 \rightarrow 1$ and $111 \rightarrow 0$, the 000 configuration activates the 111 configuration, and vice versa; then the chance for a two-cycle dynamics is large.
4. The rules with complex dynamics are most difficult to understand. But typically, there are spatial regions that become either homogeneous or periodic (called *background*), whereas in other regions some moving local configurations emerge. If the local configuration moves with a constant speed, it is called a *glider*; if it moves irregularly (only when the background with which it interacts has structure in itself, such as periodic), it is called a *defect*, or *domain wall*. (If the distinction between background and the gliders or defects is not all that clear, especially at the early stage of the transient, the name *creature* can also be used.[34]) The gliders or defects interact with each other when they collide, generate other gliders or defects, or are simply annihilated, until all of them disappear or coexist in an equilibrium state.
5. For chaotic rules, almost all three-input configurations are activated everywhere in the space. There are no localization effects: any change in one region of the space will propagate to other regions of the space.

Now we ask how the introduction of nonlocal connection affects the mechanisms behind the various dynamics as discussed above. For null rules, the effect is very small. Whether the three inputs are taken from the neighbor or from three unrelated sites, the convergence to the all-0's configuration is equally strong. Nevertheless, examples exist where some rules having null dynamics with local connections might have fixed-point dynamics with nonlocal connections.

For fixed point with a shift rules, the effect is to increase the cycle length from 1 to a large value depending on the wiring. This case is best illustrated by rule 1, which has updating rule $001 \rightarrow 1$ and all other three-input configurations are mapped to 0. The reason that the cycle length is 1 along certain spatiotemporal direction in the case of a local connection is purely because of the third input being the right input. In a nonlocal connection, however, any state value 1 will jump from a third input of a site to the site itself, and to another site which takes the site as the third input. Such jumping is determined completely by the wiring diagram.

As for chaotic rules, the effect of introducing nonlocal connection is not very large. Almost all three-input configurations are activated throughout the system whether or not the connection is local. The only class of dynamical behavior that will be changed dramatically is the complex or edge-of-chaos dynamics, because concepts such as background and gliders no longer exist. This will be discussed in Section 6.

The characterization of the dynamical behavior of a nonlocal cellular automaton rule is complicated by the fact that there are so many choices of random wiring, and there is no guarantee that all random wirings will give the same dynamics. Although a similar problem exists in local cellular automata with respect to different initial conditions, it deserves more attention for nonlocal cellular automata because wiring seems to have a stronger control over the dynamics than the initial configuration.

In the following, I will list the numerically observed dynamical behaviors for all elementary cellular automata with partially-local and fully-nonlocal connections. The definite classification of the dynamics of some rules might be impossible, as some random wirings lead to one behavior while other random wirings lead to another, and the classification used here for some rules should not be considered as conclusive.

3.1. Classification of Partially-Local Rules, and Their Rule Space

The partially-local cellular automata were first studied by Walker and Ashby[63] and some statistical quantities such as the cycle lengths are determined for small system sizes. In this subsection, I will base my classification of rules on the observation of spatiotemporal patterns starting from randomly chosen initial conditions. Comments are made whenever applicable if the dynamics from the degeneracy-permitted connections (Fig. 1a) differ from those with the distinct-inputs connections (Fig. 1b). Even if we have chosen the wiring type, different samplings of the wiring as well as initial conditions may still lead to different dynamics, as illustrated in Fig. 3 for rule 74. More careful studies are needed to determine the probability of having one dynamics versus another dynamics (for example, by determining the distribution of cycle lengths). These studies will be included in a future publication.

The classification is the following:

1. Null dynamics for almost all wirings (N):
   0, 32, 128, 160.
2. These rules can have either null dynamics or fixed-point dynamics, although some rules (e.g., rule 8) tend to have more null dynamics, whereas others (e.g., rule 200) tend to have more

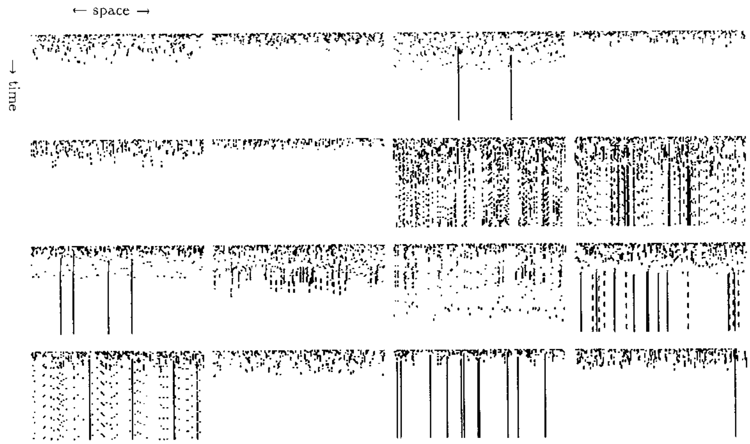


Fig. 3. Spatiotemporal patterns for rule 74 of degeneracy-permitted partially-local connections. Sixteen different initial configurations as well as different random wirings are sampled: some of them lead to all-0 configuration, some lead to the inhomogeneous configurations with a few walls, and some of them exhibit periodic dynamics. The system size is 134, the number of time steps is 68.

fixed-point dynamics. Rules 72 and 168 behave more like null rules in the distinct-input connections (N-F):
8, 40, 72, 104, 136, 168, 200.

3. Fixed-point dynamics (F):
   4, 12, 76, 77, 132, 140, 142, 164, 204, 232.

4. These rules have seemingly fixed-point dynamics (many domain walls), but actually sometimes have local two-cycle or local periodic modes. Rules 36 and 44 almost always have local periodic modes. Except for these two rules, others are more like fixed-point rules for the distinct-input connections (F-P):
   13, 14, 36, 44, 78, 172.

5. These rules can have either null, fixed-point, or periodic dynamics. For the distinct-input case, both rules no longer exhibit the null dynamics, and rule 138 no longer exhibits the fixed-point dynamics (N-F-P, or simply N-P):
   74, 138.

6. Two-cycle dynamics (2):
   1, 5, 7, 19, 23, 33, 35, 43, 50, 51, 178.

7. These rules can have either two-cycle dynamics or longer-cycle

periodic dynamics. The distinct-input connection makes them behave more like periodic rules (2-P):
3, 11, 27, 29.

8. Periodic rules. The distinct-input connection makes the cycle lengths much longer (P):
6, 15, 28, 34, 42, 108, 134, 156, 162, 170, 184.
9. These rules are basically periodic rules, but can also have null dynamics. They are most likely to correspond to the fixed-point dynamics with a shift when the connection is back to local. The distinct-input connection usually changes them to purely periodic rules (N-P):
2, 10, 24, 56, 130, 152.
10. These rules can exhibit either periodic dynamics or chaotic dynamics (at least with very long cycle lengths). The distinct-input connection turns them into chaotic (or periodic with extremely long cycle lengths) dynamics (P-C):
25, 38, 46, 58, 60, 106, 131, 154.
11. Chaotic rules (C):
9, 18, 22, 26, 30, 37, 41, 45, 54, 57, 73, 90, 105, 129, 133, 137, 146, 150, 161.

The rules space as "decorated" by the above classification is shown in Fig. 4. One can see that is is not unusual that one mean-field cluster contains rules with several different dynamical behaviors: an indication that mean-field theory fails badly. One obvious explanation is that for partially-local connections, the central input and the two other inputs are not equal, and play quite different roles. In order to take this into account, consider the following refined mean-field parameters:

$$n_1(0) \equiv \text{number of bits among } a_1 \text{ and } a_4 \text{ that are equal to } 1$$

$$n_1(1) \equiv a_2$$

The reason for this is that in $001 \to a_1$, $100 \to a_4$, and $010 \to a_2$, though all three input configurations contain one 1 and two 0's, the 1 in configuration 010 is located in the central position, whereas in 001 and 100 it is not. These two parameters simply split the original n_1 parameter: $n_1 = n_1(0) + n_1(1)$. Similarly, define

$$n_2(1) \equiv \text{number of bits among } a_3 \text{ and } a_6 \text{ that are equal to } 1$$

$$n_2(0) \equiv a_5$$

to split the original n_2 parameter.

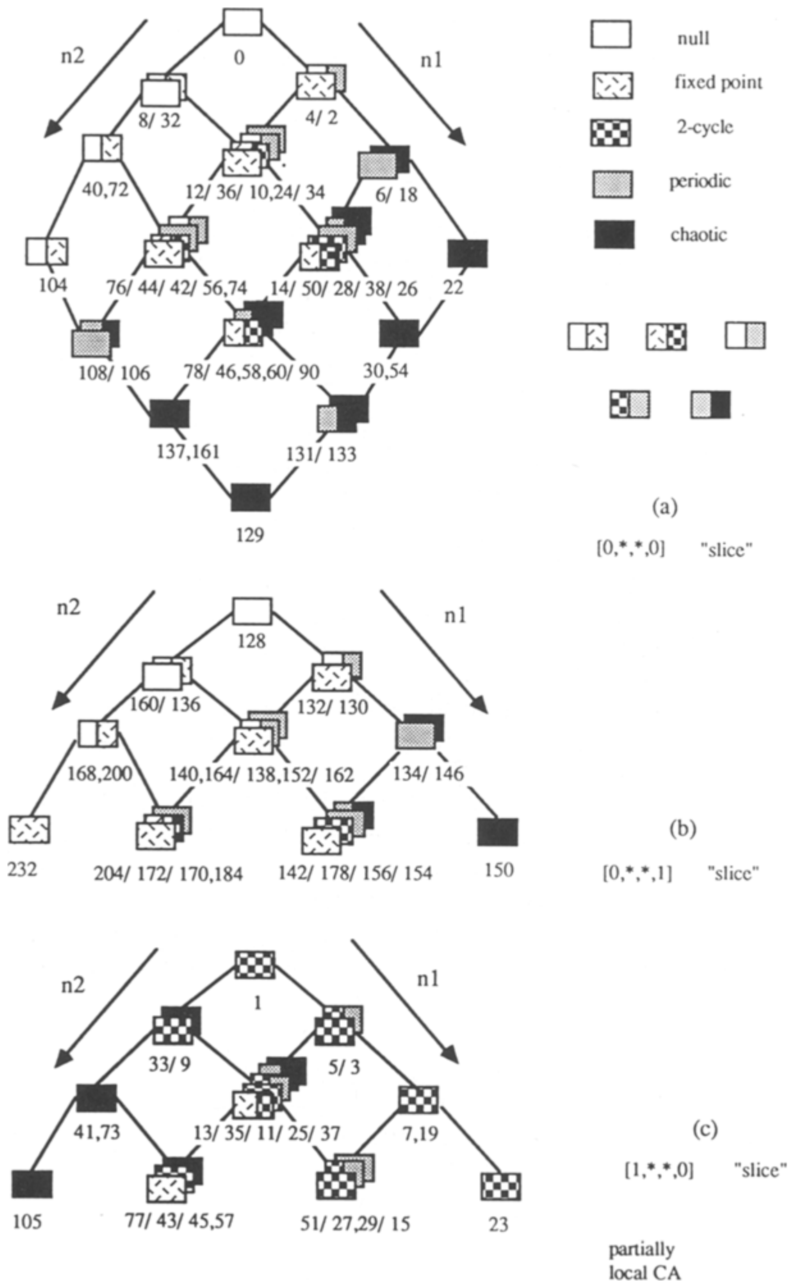


Fig. 4. The rule space for partially-local cellular automata with mean-field parametrization, similar to Fig. 2 for the local cellular automata.

Table I

n_0, $(n_1(0), n_1(1))$, $(n_2(0), n_2(1))$, n_3	Rules	Behavior[a]
0, (1, 0), (0, 1), 0	10, 24	(N-P)
0, (1, 0), (1, 1), 0	42, 56	P, (N-P)
0, (1, 1), (0, 1), 0	14, 28	(F-P), P
0, (1, 1), (1, 1), 0	46, 60	(P-C)
0, (1, 0), (0, 1), 1	138, 152	(N-P)
0, (1, 0), (1, 1), 1	170, 184	P
0, (1, 1), (0, 1), 1	142, 156	F, P
1, (1, 0), (0, 1), 0	11, 25	(2-P), (P-C)
1, (1, 0), (1, 1), 0	43, 57	2, C
1, (1, 1), (0, 1), 0	15, 29	P, (2-P)

[a] N, null; F, fixed point; 2, two cycle; P, periodic; C, chaotic. If more than one symbol is linked with a hyphen, the dynamics can be either one of them starting from different wirings and different initial conditions.

With the refined mean-field parameters, each cluster is labeled by six parameter values: $\{n_0,\ (n_1(0),\ n_1(1)),\ (n_2(0), n_2(1)),\ n_3\}$. All except ten clusters contain only one rule. These ten excepted clusters each contains two rules. The clusters and dynamics of these rules are given in Table I. One can see that even with the refined mean-field parameters, the prediction is not as good as expected.

3.2. Classification of Fully Nonlocal Rules, and Their Rule Space

The number of independent rules in fully nonlocal connection is smaller than that in local and partially-local connections (which is 88), because one can switch and relabel not only the first and the third inputs, but also the first and the second inputs, or the second and the third inputs. When the first and the second inputs are interchanged, we have an equivalent rule:

$$\{a_7, a_6, a_3, a_2, a_5, a_4, a_1, a_0\}$$

and when the second and the third inputs are interchanged, we have another equivalent rule:

$$\{a_7, a_5, a_6, a_4, a_3, a_1, a_2, a_0\}$$

From these equivalent rules, we can get more equivalent rules by interchanging 0 and 1. By considering all possible equivalent relations, there are only 46 independent rules left.

The classification of these 46 rules is listed below. In order to make a comparison with the partially-local and the local cellular automata easier, the remaining 42 ($=88-46$) rules are also listed in parentheses.

1. Null rules:
   0, 8 (32), 40 (72), 104, 128, 136 (160), 168 (200), 232.
2. Two-cycle rules:
   1, 3 (5), 7 (19), 23.
3. Periodic rules: Rule 27 tends to have two-cycle dynamics if the degenerate inputs are allowed, periodic and longer transients if inputs are distinct. Rule 172 can have null, fixed-point, and periodic dynamics if the degenerate inputs are allowed, but periodic after an extremely long transient if the inputs are distinct. Rules 2, 10, 24, 44, 130, 138, and 152 can also have null dynamics if the degenerate inputs are allowed. Rule 170 has much longer cycle lengths, even comparable with some rules listed as chaotic rules, but its transient time is very short.
   2 (4), 10 (12, 34), 15 (51), 24 (36), 27 (29), 42 (76), 44 (56, 74), 130 (132), 138 (140, 162), 152 (164), 170 (204), 172 (184).
4. Chaotic rules: Rules 11, 14, 43, and 142 have much shorter cycle lengths than other rules listed here.
   6 (18), 9 (33), 11 (13, 35), 14 (50), 22, 25 (37), 26 (28, 38), 30 (54), 41 (73), 43 (77), 45 (57), 46 (58, 78), 60 (90), 105, 106 (108), 129, 131 (133), 134 (146), 137 (161), 142 (178), 150, 154 (156).

The spatio-temporal patterns for all these 46 rules with the distinct-inputs connection are shown in Fig. 5.

The rules space with mean-field parametrization for fully nonlocal cellular automata is shown in Fig. 6 similar to Fig. 2 and 4. Besides the fact that the 36 mean-field clusters contain only 46 independent rules instead of 88, each cluster excellently characterizes the dynamical behavior of rules contained in that cluster, as illustrated by the ten clusters which contain two rules (see Table II). Note that although rule 170 and rule 172 are both classified as periodic, rule 170 has much longer cycle lengths and rule 172 has much longer transient times. Also, rule 15 has much longer cycle lengths than rule 27.

The "cleanness" of the nonlocal cellular automata rule space as illustrated in Fig. 6 implies that the basic features of the dynamics can be accounted for by simple mean-field theory. In the next section, I will present some crude estimation of mean-field parameter values at which the transition from nonchaotic to chaotic dynamics occurs. I will also discuss in general terms the transition phenomena in cellular automata rule space.

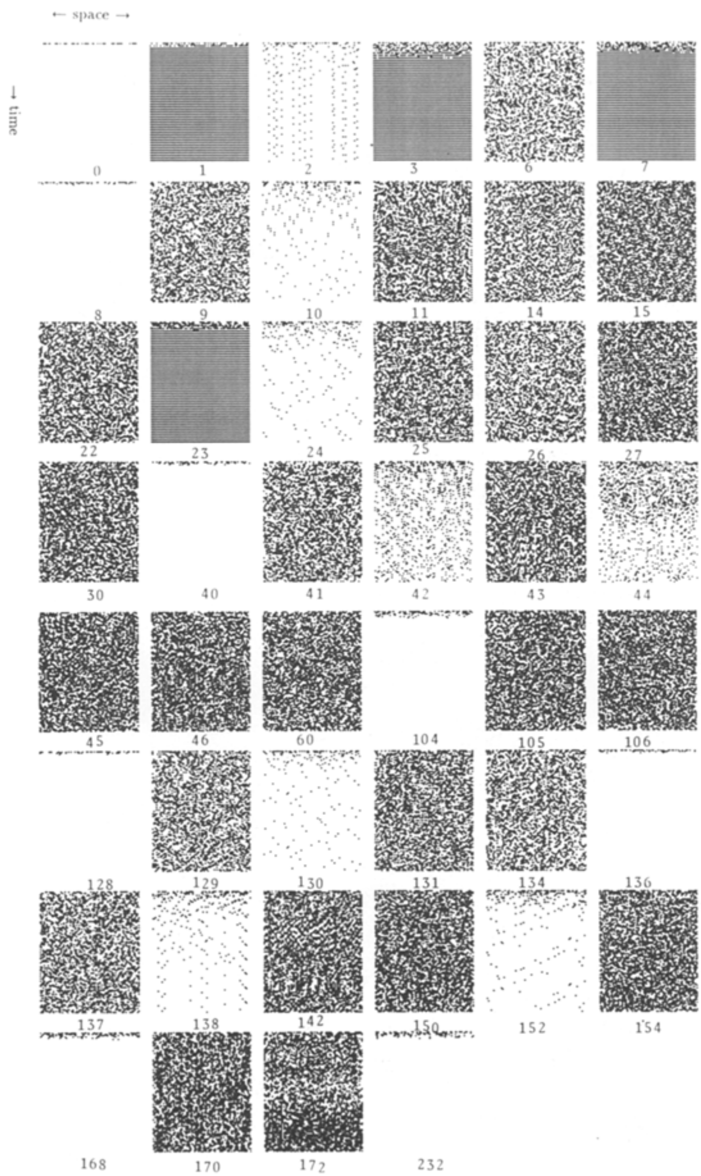


Fig. 5. The spatiotemporal patterns of all 46 independent elementary cellular automata with distinct-input, fully-nonlocal connection. The system size is 77 and the number of time steps is 94. It is typically easy to tell which rule is two-cycle periodic, which is chaotic, etc., from these patterns, except for periodic rules with very long transients (e.g., rule 27 and rule 172) or very long cycle lengths (e.g., rule 15).

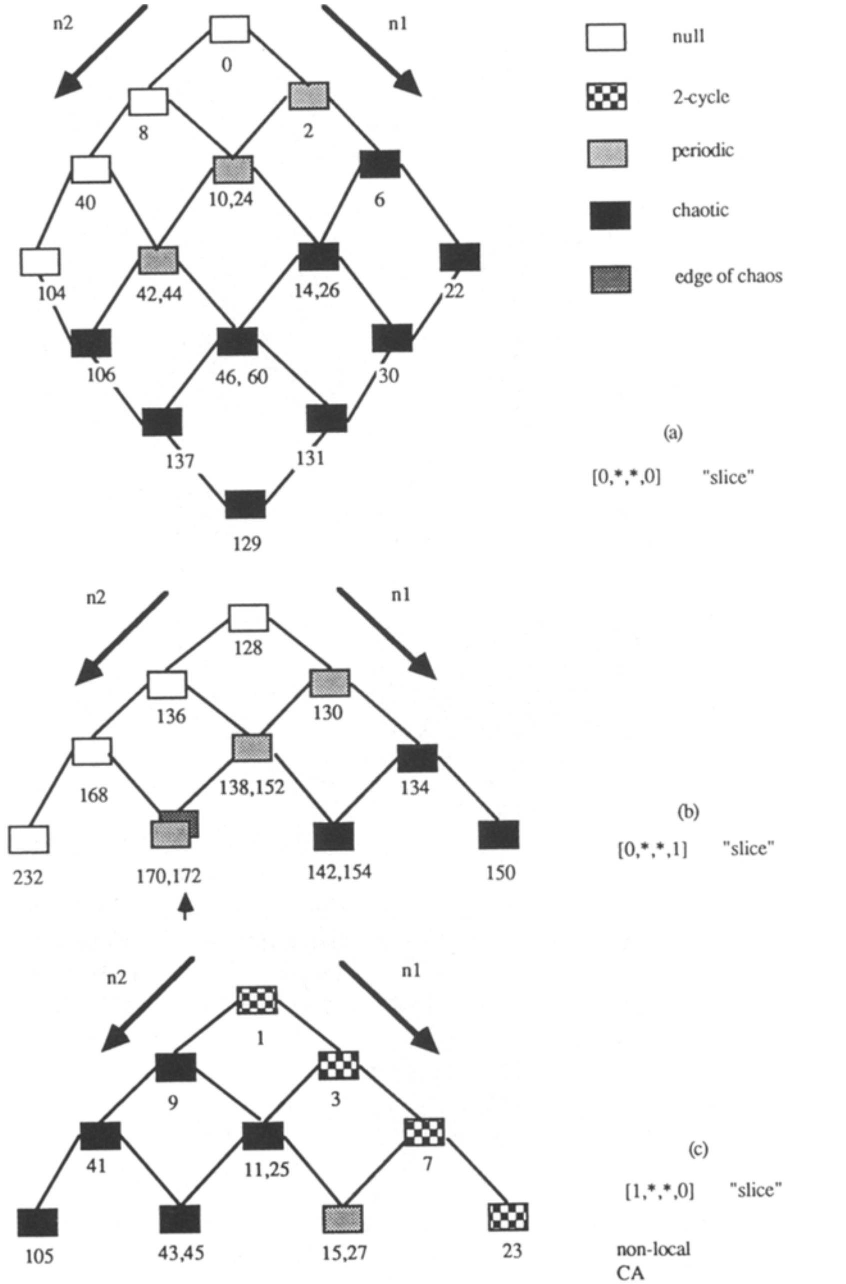


Fig. 6. The rule space for fully-nonlocal cellular automata with mean-field parametrization. The rule with long transient (rule 172) is marked.

Table II

$\{n_0, n_1, n_2, n_3\}$	Rules	Behavior[a]
0110	10, 24	P
0120	42, 44	P
0210	14, 26	C
0220	46, 60	C
0111	138, 152	P
0121	170, 172	P
0210	142, 154	C
1110	11, 25	C
1120	43, 45	C
1210	15, 27	P

[a] P, periodic; C, chaotic.

4. BIFURCATION-LIKE PHENOMENA IN MULTIPARAMETER DYNAMICAL SYSTEMS

4.1. Critical Surfaces in Higher-Dimensional Space

It is well understood that in lower-dimensional dynamical systems with one parameter, there is a transition from regular dynamics to chaotic dynamics by tuning that parameter.[46] This sudden change of dynamical behavior with the smooth change of the parameter is studied in bifurcation theory.[16]

High-dimensional dynamical systems typically need more parameters to describe the details of the interaction among components (though one can increase the degrees of freedom to infinity while keeping the number of parameters finite). Take elementary cellular automata, for example; a rule is specified by eight bits (a_0, a_1,..., a_7), and each of them can be considered as a parameter jumping between 0 and 1. An incomplete description requires fewer parameters, for example, the mean-field parametrization has four parameters n_0, n_1, n_2, n_3 varying either from 0 to 1, or from 0 to 3; or the λ parameter choosing an integer value between 0 and 8.

The bifurcations or the transition phenomena in multiparameter, higher-dimensional dynamical systems are more complicated than those in single-parameter (perhaps lower-dimensional) dynamical systems. Generally speaking, if the dimension of the parameter space is m, there is a transition surface (I will use the term *hypersurface* to represent surfaces in high-dimensional spaces which can partition that space) with at most a dimension $m-1$, which separates rules of regular and chaotic dynamics (though it is also possible to have nonchaotic "bubbles" inside the chaotic regime). Passing through the transition hypersurface, one can experience a sudden change of dynamical behavior with no other "in-between" dynamics, a case called a first-order phase transition, using an analogy with statistical physics; or one can experience a gradual change of dynamical behavior, with some rules having "in-between" dynamics actually sitting on the transition hypersurface, a case called a second-order phase transition.

Rules with "in-between" dynamics (or complex dynamics, critical dynamics, etc.) do not cover the transition hypersurface completely, leaving other regions of the transition hypersurface as "holes." If one hits a hole while passing through the transition hypersurface, no rules with complex dynamics will be encountered. Considering this fact, as well as the fact that the dimension of the transition hypersurface is strictly smaller than that of the whole parameter space, the chance for observing a complex dynamics is *extremely* small. One has to tune some structural parameters to reach not only the transition hypersurface, but also the regions of the transition hypersurface with the critical rules.

As a footnote, I want to relate the general picture presented here on transition hypersurface in multiparameter, higher-dimensional dynamical systems with the discussions of "self-organized criticality."[1] (For readers who are not familiar with this model, please skip the next three paragraphs.) From our global point of view of rule space, it is easy to recognize that one claim about the self-organized criticality is misleading or perhaps incorrect, that the criticality in many-degree-of-freedom dynamical systems is "fundamentally different from the critical point at phase transitions in equilibrium statistical mechanics which can be reached only by tuning of a parameter."[1]

This claim overlooks the basic fact that defining a rule is equivalent to setting the parameters. Any particular critical or complex rule is sitting at a particular point on the transition hypersurface which is difficult to reach in a random setting of parameters. Also, the claim that the critical dynamics is "insensitive to the parameters of the model"[1] should not be correct if the parameter axis is perpendicular to the transition hypersurface. It might be the case that some parameters are *irrelevant* parameters which only move the rule along the transition hypersurface. It is thus important to identify the parameter which is *relevant* for making the transition happen.

In fact, the criticality in the "self-organized criticality" models can be destroyed not only by changing the structural parameters (e.g., introducing inertia terms[50, 23]), but also by changing some seeming trivial conditions: (1) Change the boundary condition. If the boundary condition is periodic instead of being close on one side and open on the other side, the dynamics of the avalanche process can either be subcritical—if the initial average slope is low, or supercritical (chaotic)—if the initial average slope is very high.[5,7] (2) Remove the adding sand process. Besides the avalanche process—which follows a cellular automaton rule, there is also an adding sand process in the model: one grain of sand is added only if a global examination of every site ensures that no slope at some site is larger than the threshold value. This globally monitoring process violates the basic principle of locality in cellular automata, and together with the close/open boundary condition, it tunes the parameter (i.e., average slope) to its critical value. The tuning parameter effect will be completely absent if no sand is allowed to come in to, as well as go out from, the system.

In the previous studies of cellular automata rule spaces, it is the λ parameter which is used as the "relevant" structural parameter.[31,32,35,37] One can consider the λ parameter as a measure of the activation of entries in the rule table.[30] When λ is close to zero, most three-input configurations collapse to the 000 configuration, and it is more likely that the dynamics is a fixed point. On the other hand, if λ is close to 0.5, all three-input configurations are activated, and dynamics is likely to be chaotic. By tuning λ from 0 to 0.5, it is expected that one will pass through the transition hypersurface.[8]

As a crude estimation, the transition hypersurface has a λ value around[76]

$$\lambda_c \approx 1/n$$

where n is the number of inputs (e.g., $n = 3$ for elementary rules). This λ_c is the estimation of the onset of nonzero entropy values. Another estimation of λ_c, which is the onset of the nonzero expansion rate of the perturbation, gives[37]

$$\lambda_c \approx \frac{1}{2} - \frac{1}{2}\left(1 - \frac{4}{n+1}\right)^{1/2} \approx \frac{1}{n+1}$$

These estimations become more accurate as the number of inputs goes to infinity (while the number of states is kept at 2). Numerically, the maximum spatial mutual information is used to locate the transition region using the fact that the correlation length become longer near the transition hypersurface.[37] This numerically determined λ_c is observed to be larger than the two estimations mentioned above.[37]

One conclusion from the numerical study of the transition hypersur-

[8] Rules with λ parameter equal to 0.5 do not have to be chaotic rules. Actually some nonchaotic rules always have λ parameter equal to 0.5, such as the *identity rule*, *Gacs–Kurdyumov–Levin rule*,[15] etc. There is a whole class of "unbiased" rules with $\lambda = 0.5$ that are able to lead the system to consensus and nonchaotic dynamics.[40]

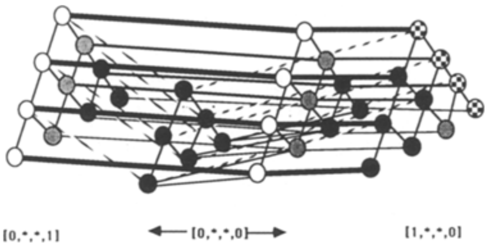


Fig. 7. The complete "folded" rule space for fully-nonlocal cellular automata. The λ parameter is increased from top to bottom. Each connection between two clusters is a change of 1 in one of the mean-field parameters, which either increases or decreases the λ value by 1.

face in local cellular automata rule space is that the λ value at which a transition occurs from periodic to chaotic dynamics is not unique. This means that the transition hypersurface is not a hyperplane (a *hyperplane* is a hypersurface on which some linear function of the coordinates is a constant) to which λ is perpendicular. To characterize how the transition hypersurface "bends," one has to identify different regions of the transition surface, and in order to do so, introduce more parameters.

For elementary cellular automata with fully nonlocal connections, mean-field parameters can characterize the transition hypersurface almost

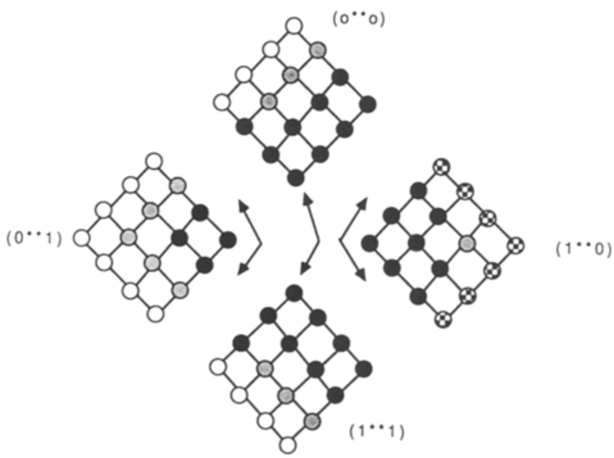


Fig. 8. The unfolded rule space for fully nonlocal cellular automata. The folding, or the equivalent relation among clusters, is indicated by the two-way arrows. The connection between different slices of the rule space is not drawn, but one can refer to Fig. 7.

exactly. From Fig. 6, one can see that if $n_0 = n_3 = 0$, then $1 < (n_1)_c < 2$; if $n_0 = 0$ and $n_3 = 1$, then $1 < (n_1)_c < 2$; and if $n_0 = 1$ and $n_3 = 0$, then $0 < (n_2)_c < 1$. This characterization of the transition hypersurface is better than the one by the λ parameter. For example, in the case of $n_0 = n_3 = 0$, if one chooses $3 < \lambda_c < 4$, three chaotic clusters (which contain chaotic rules 6, 14, 26, and 22) will be counted as being below the transition line (remember that λ is pointing down in Fig. 6), whereas using $1 < (n_1)_c < 2$ will miss only one chaotic cluster (which contains one chaotic rule 106).

As a final note, the above discussion about the transition surface applies to the "folded" rule space, which contains only the independent rules.[36] The folded rule space with the mean-field parametrization for the fully-nonlocal elementary cellular automata is shown in Fig. 7. Part of the slice with the nonlinear clusters $\{0, *, *, 0\}$ bends up to the top, so that the λ is always increased from top to bottom (from 0 to 0.5). Also shown is the connection between nonlinear clusters with the linear and inversely-linear cluster. Figure 8 shows the original "unfolded" rule space containing degenerate clusters. The equivalence relations (or "foldings") are indicated by the two-way arrows. In the next section, I will try to determine the transition hypersurface by mean-field theory.

4.2. The Return Map of Density in Mean-Field Theory and the Determination of the Transition Surface

The approach adopted in this subsection is to ignore the details in dynamics and only examine macroscopic quantities such as the density of 1's in the mean-field theory. For a review of this approximation scheme, see ref. 18. Note that the term "first-order Markov approximation" used in ref. 18 is the same with mean-field theory. Other relevant studies are included in refs. 17, 53, 71, and 76. The mapping for the density of 1's

$$d_{t+1} = f(d_t) \tag{4.1}$$

is also called the *return map.* As a crude approximation, fixed-point cellular automata rules have zero fixed-point solutions of the return map; periodic rules have periodic solutions; and chaotic rules have nonzero fixed-point solutions. These assumptions are not always true; for example, for some periodic rules only the local densities oscillate instead of the global density, and some fixed-point rules can have large values of density. The logic behind this approximation is that rules with fixed-point dynamics usually also have a low density of 1's, whereas rules with chaotic dynamics usually have a high density of 1's (that is, close to 0.5).

In the mean-field theory as applied to elementary rules, the return map is

$$d_{t+1} = n_0(1 - d_t)^3 + n_1 d_t(1 - d_t)^2 + n_2 d_t^2(1 - d_t) + n_3 d_t^3 \tag{4.2}$$

By studying how the solution of Eq. (4.2) changes with the mean-field parameters, we hope to understand why and when the transition occurs. As before, I discuss the three slices of the rule space separately.

1. The nonlinear clusters ($n_0 = n_3 = 0$): The return map is

$$d_{t+1} = n_1 d_t (1 - d_t)^2 + n_2 d_t^2 (1 - d_t) \tag{4.3}$$

which is shown in Fig. 9a for all possible n_1 and n_2 values.

The return maps are grouped into four bundles near the origin. The lowest one corresponds to $n_1 = 0$, the next one to $n_1 = 1$, and so on. Similarly, the return maps near the point $(d_t, d_{t+1}) = (1, 0)$ are also grouped into four bundles, with the lowest one corresponding to $n_2 = 0$, the next one to $n_2 = 1$, etc. The intersection of the return map with the diagonal line is the fixed-point solution of Eq. (4.3). The slope at the inter-

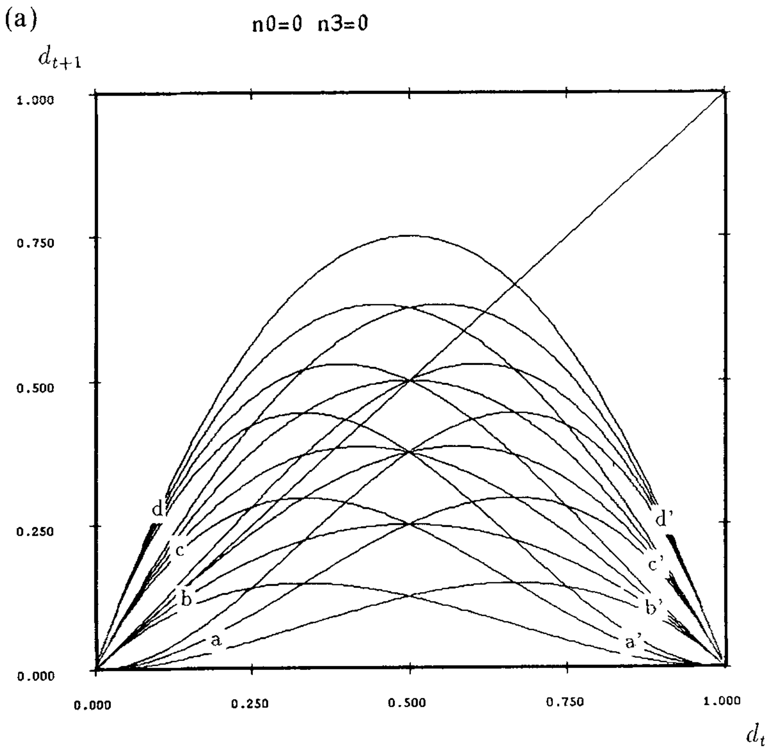
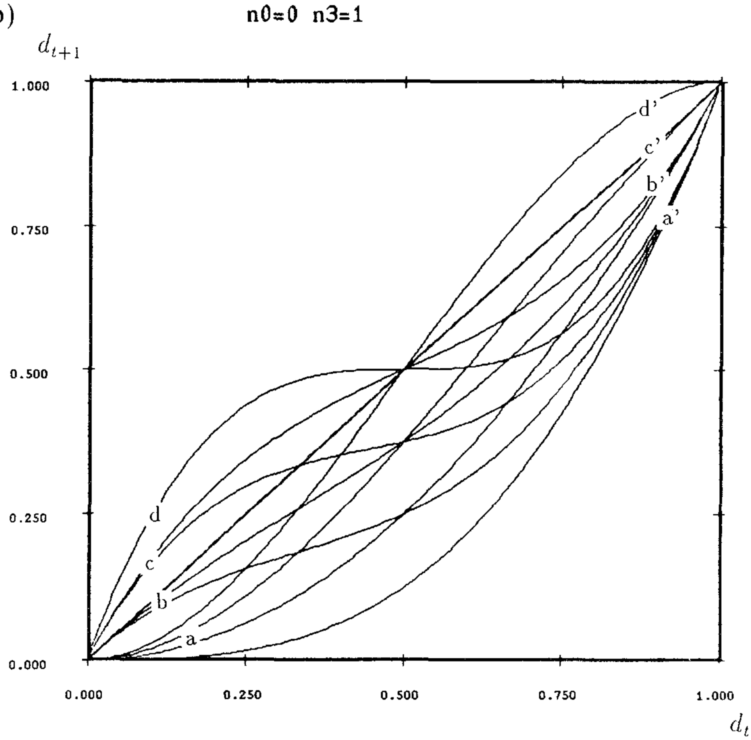


Fig. 9. The return maps (the density at time $t+1$ versus the density at time t) by mean-field theory. Those with the same n_1 values are labeled by letters a ($n_1 = 0$), b ($n_1 = 1$), c ($n_1 = 2$), and d ($n_1 = 3$); and those with the same n_2 values are labeled by letters a′ ($n_2 = 0$), b′ ($n_2 = 1$), c′ ($n_2 = 2$), and d′ ($n_2 = 3$). (a) nonlinear clusters ($n_0 = n_3 = 0$); (b) linear clusters ($n_0 = 0$, $n_3 = 1$); (c) inversely-linear cluster ($n_0 = 1$, $n_3 = 0$).

(b)



(c)

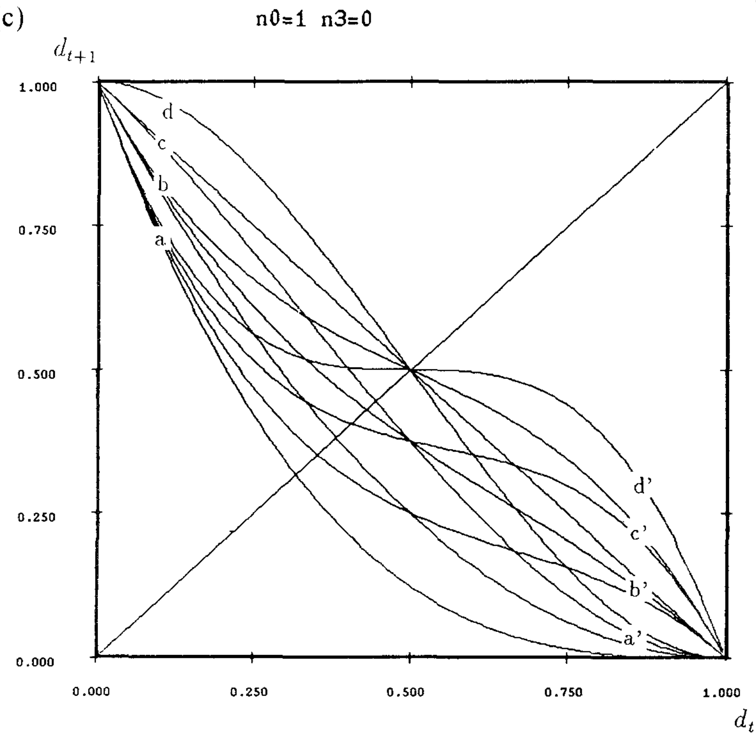


Fig. 9. (*Continued*)

section measures the stability of that solution, which becomes unstable when the absolute value of the slope is larger than 1.

When $n_1 = 0$, none of the return maps intersect with the diagonal line rather than the origin, so the null dynamics is expected. When $n_1 = 1$, only one return map intersects with the diagonal line (that of $n_2 = 3$), and indeed the corresponding cluster contains chaotic rules. When $n_1 > 1$, all the return maps intersect with the diagonal line at nonzero values, which leads to stable fixed-point solutions (except the $n_1 = n_2 = 3$ cluster, whose fixed point is marginal unstable).

Note that Eq. (4.3) becomes the logistic map[46] when $n_1 = n_2$. In particular, the return map at $n_1 = n_2 = 3$ gives the logistic map at the first period-doubling bifurcation point. In the logistic map, the transition to chaos is by consecutive period-doubling bifurcations when the fixed-point or the periodic solutions of the map lose stability. For our case, because macroscopic quantities such as the density generally *do not* have chaotic fluctuation even if the microscopic variables behave chaotically, we do not expect the return map (4.3) to become chaotic. There are similar discussions in ref. 3 on the difficulties for having macroscopic chaotic fluctuations.

2. The linear clusters ($n_0 = 0$, $n_3 = 1$): The return map is

$$d_{t+1} = n_1 d_t (1 - d_t)^2 + n_2 d_t^2 (1 - d_t) + d_t^3 \tag{4.4}$$

which is shown in Fig. 9b for all possible n_1 and n_2 values. Again, one can recognize the bundles near the origin (organized by n_1 value) and near the point $(d_t, d_{t+1}) = (1, 1)$ (organized according to n_2 values).

When $n_1 = 0$, no return maps intersect with the diagonal line at nonzero values, and null rules are expected (except for $n_2 = 3$, but then the nonzero fixed-point solution is unstable). When $n_1 = 1$ and $n_2 \neq 2$, there is no nonzero fixed-point solution either. If $n_1 = 1$ and $n_2 = 2$, the return map *is the diagonal line itself*, and all possible density values are fixed-point solutions (to be discussed in the last section). Only when $n_1 > 1$ are there nonzero stable fixed-point solutions. So, similar to the nonlinear clusters, the transition is induced by increase the n_1 parameter.

3. The inversely-linear clusters ($n_0 = 1$, $n_3 = 0$): The return map is

$$d_{t+1} = (1 - d_t)^3 + n_1 d_t (1 - d_t)^2 + n_2 d_t^2 (1 - d_t) \tag{4.5}$$

which is shown in Fig. 9c for all possible n_1 and n_2 values. The bundles near the point $(d_t, d_{t+1}) = (0, 1)$ are organized by the n_1 parameter, whereas those near the point $(d_t, d_{t+1}) = (1, 0)$ are organized by the n_2 parameter.

All the return maps intersect with the diagonal line at some nonzero value, so no rule is expected to exhibit null dynamics. The difference among these return maps is that some of them have stable fixed-point solutions (then the dynamics is probably chaotic), while others have unstable fixed-point solutions (then the dynamics is periodic; in particular, if the two-cycle is the stable solution, the dynamics is also two-cycle). When $n_2 = 0$, no fixed-point solution is stable, so the dynamics is periodic (two-cycle). When $n_2 = 1$, except for one case where the return map is the off-diagonal line itself ($d_{t+1} = 1 - d_t$) with $n_1 = 2$, all the fixed-point solutions are stable. When $n_2 > 1$, all the fixed-point solutions are stable. As a result, the transition from periodic to chaotic dynamics can be accomplished by increasing the n_2 parameter.

In conclusion of this section, by examining the solution of the return map by mean-field theory, one can almost completely determine the transition point at different parts of the rule space. Not surprisingly, the method is successful only for fully-nonlocal cellular automata, when the statistical averaging in mean-field theory is close to reality. If the same mean-field theory is applied to local cellular automata, there will be too many exceptions for the theory to be called successful (compare Fig. 2 with Fig. 6). On some discussion of how to improve the existing parametrization schemes so that one can predict more correctly the dynamical behavior of local cellular automata, see ref. 39.

The observation that either n_1 or n_2 can be the relevant parameter which is perpendicular to the transition surface is equivalent to saying that the transition hypersurface is not a single hyperplane. Generally speaking, a critical dynamics results from a balance among many competing factors in the multiparameter dynamical system. If one moves from one part of the rule space to another, the "environment" in which the parameters previously achieved a balance is now changed. And that previously relevant parameter may no longer be perpendicular to the transition surface in the new "environment." It will be interesting to determine the transition hypersurface for a larger cellular automata rule space and rule spaces of other multiparameter, many-degrees-of-freedom dynamical systems.

5. NONLOCAL CELLULAR AUTOMATA AS COMPUTERS

5.1. Computation at the Level of Each Component

One of the original motivations to study local cellular automata is to construct parallel computers which can carry out arbitrary computations (universal computers). One way to show that a cellular automaton is capable of doing universal computation is to establish the equivalence relation between this cellular automaton and the Turing machine.[8, 42] Another way to show this is to establish that the cellular automaton can perform three fundamental logical operations: AND, OR, NOT. In fact, the famous "game of life" is shown to be able to carry out these three logical operations by the collision of gliders.[2]

From the proof for the game of life being a universal computer, one can see why it is so difficult for a local cellular automaton to be one. First of all, the cellular automaton has to produce gliders—moving local configurations on either the blank or periodic background. This condition imposes a very strong restriction on the rule table. Second, there should be different types of gliders and various collision events, so that these collisions can simulate the logical operations.

With this particular realization of a universal computer, the dynamics of the "game of life" starting from a random initial condition is quite complex. Different gliders emerge, interact with each other, new gliders are created from the collision, and so on, until the total number of gliders decreases to a minimum. The transient time is long since gliders do not all disappear together after collisions. The system is sensitive to perturbations in spatial configuration, since flipping a site value can destroy a glider, and consequently change the forthcoming glider activities.

These observations led several authors to speculate that local cellular automata which are capable of doing universal computation also show complex dynamical behaviors starting from random initial conditions.[72, 74, 31, 32] Admittedly, this speculation is correct in many cases. Programming a universal computer to solve a very hard problem consumes a long computing time. This long computing time can be easily interpreted as a long transient if the computer is viewed as a dynamical system, and a long transient is an important trait of complex dynamics.

There is one point that is overlooked by the above speculation, though: complex dynamics starting from a specially designed (i.e., programmed) initial condition do not imply complex dynamics starting from any random initial condition. It has actually been shown that some local cellular automata which are capable of doing universal computation typically exhibit periodic or chaotic dynamics starting from random initial conditions[42] (but there are also others that indeed exhibit complex dynamics).

For nonlocal cellular automata, the traditional concept of space is destroyed and there will be no gliders. If they are viewed as computers, these are "old fashioned" computers with true logical gates—the same on each site—that carry out computation on the state value. So, a computation at the level of a glider in local cellular automata and that at the level of a site in nonlocal ones are quite different.

Since the rule table of local cellular automata takes much of the responsibility for constructing gliders, in order to have these gliders, the entries in the rule table have to be much restricted. The situation is completely different for nonlocal cellular automata. As long as the rule table contains fundamental logical gates, it will be a universal computer. This lesser degree of restriction on the rule table makes the connection between universal computation and complex dynamics much weaker.

In the next subsection, I will examine more specifically what logical functions a three-input rule contains. By doing so, we will understand better the computational abilities for a nonlocal cellular automaton.

5.2. How Many Fundamental Logical Gates Does a Rule Contain?

In the traditional framework of Boolean algebra, there are three fundamental operations: AND, OR, NOT. These operations are not independent of each other: NOT can be represented by some combination of AND and OR; or, given NOT, AND and OR can represent each other. In practice, any operation pairs listed below can be used to construct a universal computer: (AND, NOT), (OR, NOT), (NAND, COPY), (NOR, COPY). Note that AND, OR are two-input rules, and NOT, COPY are one-input rules.

For a three-input rule, we can check whether two inputs carry out an AND (or OR) operation when the remaining third input is fixed. Similarly, we can check whether one input carries out NOT (or COPY) when the remaining two inputs are fixed. If a three-input rule contains, say, both AND and NOT, one can construct a universal computer from it if arbitrary wiring is allowed. Let me do this for the (AND, NOT) pair in the following. Other fundamental operation pairs can be checked similarly.

Suppose the AND gate operates on the second and the third inputs, while the first input is fixed at $x_{j_1}=0$. By doing so, four bits in the rule table are fixed. Then suppose the NOT gate operates on the first input (while $x_{j_2}=x_{j_3}=1$); or, operates on the second input (while $x_{j_1}=1$, $x_{j_3}=0$, or $x_{j_1}=x_{j_3}=1$); or, operates on the third input (while $x_{j_1}=1$, $x_{j_2}=0$, or $x_{j_1}=x_{j_2}=1$). The rules satisfying these conditions are

$$
\begin{array}{lllll}
000 \to 0 & 0 & 0 & 0 & 0 \\
001 \to 0 & 0 & 0 & 0 & 0 \\
010 \to 0 & 0 & 0 & 0 & 0 \\
011 \to 1 & 1 & 1 & 1 & 1 \\
100 \to * & 1 & * & 1 & * \\
101 \to * & * & 1 & 0 & * \\
110 \to * & 0 & * & * & 1 \\
111 \to 0 & * & 0 & * & 0
\end{array}
\qquad (5.1)
$$

with the wild card symbol $*$ can be either 0 or 1. Each column on the right-hand side represents one set of allowed rules.

Similarly, if the logical gate AND operates on the second and the third inputs while the first input is fixed at $x_{j_1}=1$, and the logical gate NOT operates on the first input (while $x_{j_2}=x_{j_3}=0$, or $x_{j_2}=0$, $x_{j_3}=1$, or $x_{j_2}=1$, $x_{j_3}=0$); or, operates on the second input (while $x_{j_1}=x_{j_3}=0$, or $x_{j_1}=0$, $x_{j_3}=1$); or, operates on the third input (while $x_{j_1}=x_{j_2}=0$, or $x_{j_1}=0$, $x_{j_2}=1$), we have the following rules:

$$\begin{array}{lcccccc}
000 \to 1 & * & * & 1 & * & 1 & * \\
001 \to * & 1 & * & * & 1 & 0 & * \\
010 \to * & * & 1 & 0 & * & * & 1 \\
011 \to * & * & * & * & 0 & * & 0 \\
100 \to 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
101 \to 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
110 \to 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
111 \to 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{array} \tag{5.2}$$

One can also require the AND gate to operate on the first and the second inputs, or the first and the third inputs. But since for nonlocal cellular automata, exchanging any two inputs leads to other equivalent rules, we will not get any more independent ones. So, Eqs. (5.1) and (5.2) include all independent elementary cellular automata rules that have AND and NOT in their rule table.

Because (AND, NOT) is not the only fundamental operation pair, the rules listed in Eqs. (5.1) and (5.2) do not exhaust all three-input, nonlocal cellular automata which are universal computers. We can see that although the dynamical behaviors for almost all elementary nonlocal cellular automata are not complex, many of them are nevertheless universal computers.

Now we ask the question of which rule tables contain AND, NOT, and OR. Since OR can be constructed from AND and NOT, the absence of OR will not change the fact of whether that rule is a universal computer or not. Nevertheless, with all three gates, the "programming task" can be much simplified.

Checking whether some two inputs operate as OR for all rules listed in Eqs. (5.1) and (5.2), we have the following rules:

$$\begin{array}{llll} 000 \to 0 & 0 & 0 \\ 001 \to 0 & 0 & 1 \\ 010 \to 0 & 0 & 1 \\ 011 \to 1 & 1 & 1 \\ 100 \to 1 & 1 & 0 \\ 101 \to 1 & 0 & 0 \\ 110 \to 0 & 1 & 0 \\ 111 \to 1 & 1 & 1 \end{array} \tag{5.3}$$

These are rules: 184 (AND on the second and third inputs, NOT on the second input, OR on the first and second inputs), 216 (AND on the second and third inputs, NOT on the third input, OR on the first and third inputs), and 142 (AND, OR on the second and third inputs, NOT on the first input). Rule 216 is equivalent to rule 184 by interchanging the second and the third inputs. But these two are not equivalent to rule 142.

The operation represented by rule 184 has an extremely simple interpretation. One can consider the second input as being the control input. When the control input is 0, transmit the first input; when the control input is 1, transmit the third input:

$$x_i^{t+1} = \begin{cases} x_{j_1}^t & \text{if} \quad x_{j_2}^t = 0 \\ x_{j_3}^t & \text{if} \quad x_{j_2}^t = 1 \end{cases} \tag{5.4}$$

This operation is actually a useful information processing device called a *selector*, or *multiplexer*.[55] More complicated selectors or multiplexers can also be defined, for example, those having two control inputs and four transmission inputs.

Rule 184 is also closely related to another basic logical gate, called *Fredkin's gate*, proposed as the fundamental gate for conservative universal computers[14] (see Fig. 10). Fredkin's gate has three inputs and three outputs. Among the three inputs, one is the control input: whenever the control input takes the value 0, the other two inputs remain unchanged as the outputs; whenever the control input takes the value 1, the other two inputs switch.

That rule 184 is the only one of the two independent nonlocal cellular automata to contain three fundamental gates AND, NOT, OR, and that it is part of the fundamental gate for conservative computation, are not the only interesting observations about this rule. In the next section, I will show that rule 184 has very interesting dynamical behaviors, too.

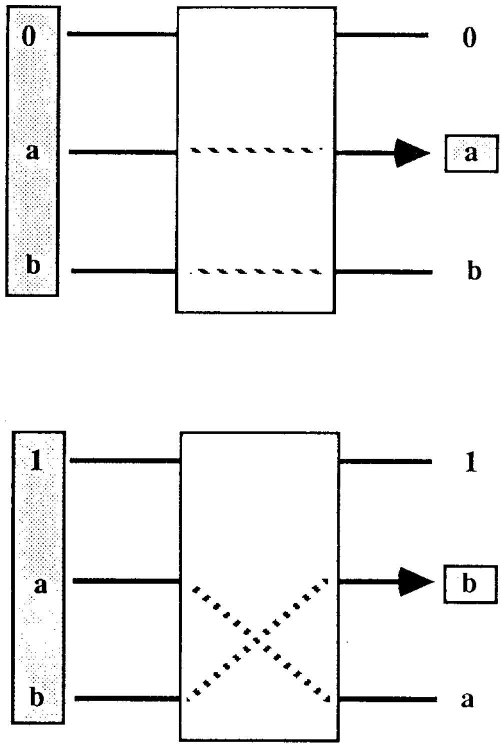


Fig. 10. Fredkin's gate: the proposed fundamental gate for conservative computation.

6. EDGE-OF-CHAOS DYNAMICS IN NONLOCAL CELLULAR AUTOMATA

In local cellular automata, whenever the name "edge-of-chaos" dynamics is mentioned, it almost always refers to rules like the "game of life." For these rules, structures at the level much higher than the individual site have emerged; activities are ongoing but slow; the limiting dynamics may be simple (e.g., periodic), but the transient times are extremely long; perturbation to the system sometime propagates, sometimes does not; and so on. In other words, they are neither chaotic rules nor typical periodic rules.

The picture depicted above is the mode of edge-of-chaos dynamics in locally-coupled dynamical systems. It cannot be applied to nonlocally-coupled dynamical systems. It is not clear what the typical modes of edge-of-chaos dynamics are for nonlocal systems. In this section, I will discuss one type of dynamical behavior which I consider to be on the edge of chaos. This example is the rule 184.

Actually, many of the discussions in the previous sections point to the rule 184 in one way or another: (1) In the rule space parametrized by mean-field parameters, rule 184 is located between null rules and chaotic rules; (2) the return map of rule 184 in mean-field theory is the diagonal line (only one other rule has the same property, rule 170), that is, the density is at a marginally stable position; and (3) rule 184 is the only one of the two rules that contains AND, NOT, OR gates, and with them, it is easier to construct any logical functions.

I will discuss two interesting aspects of the dynamical behavior of rule

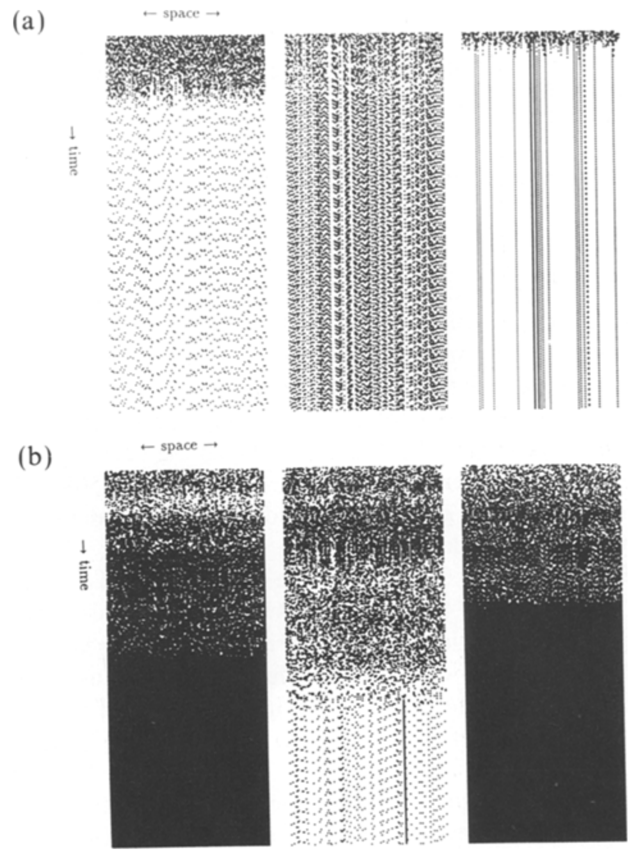


Fig. 11. Illustration that rule 184 exhibits (a) periodic dynamics in degeneracy-permitted partially-local connection; and (b) null or periodic dynamics in degeneracy-permitted fully-nonlocal connection. The system size is 124, and the number of time steps is 294.

184. The first is the fluctuation of density (of 1's). This fluctuation indicates that there is cooperation among components. The larger the magnitude of the fluctuation, the bigger the size of the coherent cluster. The second is the transient time as a function of the system size. I will show that the transient time increases more or less linearly with the system size, but there is an indication that the increase is slightly faster than linear.

The first topic will be discussed with data derived from the case of the distinct-input, partially-local connection (see Fig. 1b). The second topic will be discussed for the distinct-input fully-nonlocal connection (see Fig. 1d). The spatial-temporal patterns for the degeneracy-permitted, partially-local connection (see Fig. 1a) are shown in Fig. 11a. And those for the degeneracy-permitted, fully-nonlocal connection are shown in Fig. 11b. Some statistical results for rule 184, such as the transient times, can be quite different between the distinct-input and degeneracy-permitted connections. Sensitivity to small details of the wiring is another indication for edge-of-chaos dynamics!

6.1. Irregular Fluctuation of Density

Figure 12a shows the spatiotemporal pattern for rule 184 with a distinct-input partially-local connection. One feature of the pattern is that there exist horizontal stripes with either lighter or darker textures, indicating that the density fluctuates (the vertical lines are not printer errors, but due to the partially-local nature of the wiring).

The magnitude of the density fluctuation becomes smaller as the system size is increased, as suggested by another spatiotemporal pattern for a larger system size (Fig. 12b). Figure 13 shows the density as the function of time for different system sizes: (a) $N=100$; (b) $N=501$; (c) $N=1021$; and (d) $N=5022$, which gives better evidence that the density fluctuation does become smaller for larger systems.

To characterize the fluctuation, the return map (i.e., the density at time $t+1$ versus that at time t) is plotted in Fig. 14, exactly as what is predicted by the mean-field theory (see Fig. 9b). If the mean-field theory is completely correct, there should be no fluctuation, because $d_{t+1}=d_t$, and the density at time t is the same as the initial density. This should be the case in the infinite-system-size limit. The fluctuation we have observed in rule 184 is then a finite-size effect.

Note that macroscopic quantities such as the density typically do not exhibit chaotic fluctuations, and the return map for density is unlikely to be of the nonlinear form with a "hump," as observed for the return map for the time interval between two water drops (which can be considered as a microscopic quantity) in the dripping faucet experiment.[54] As discussed in

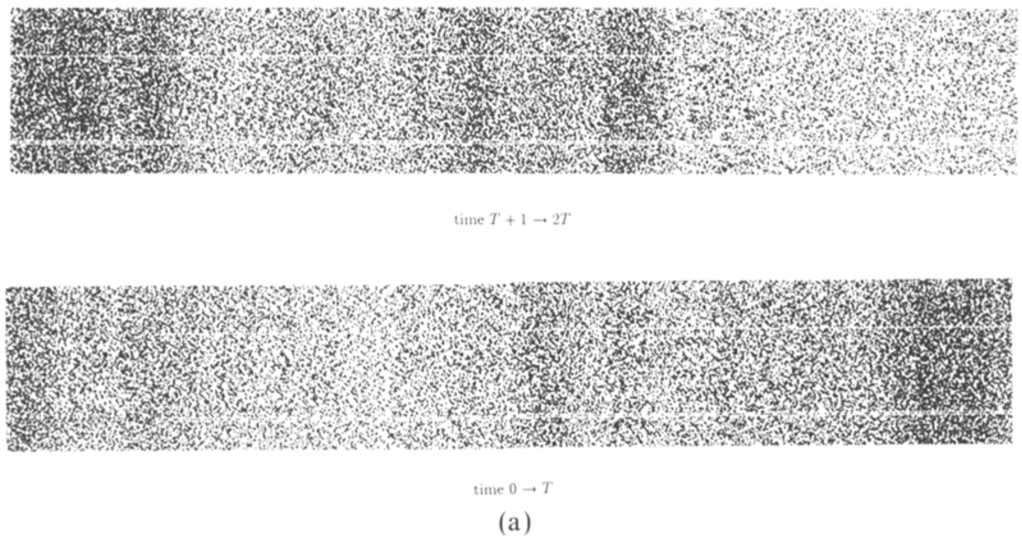
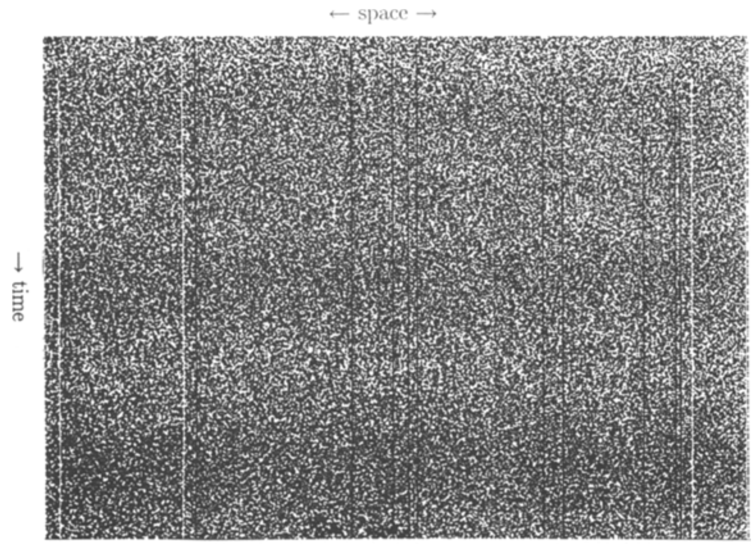


Fig. 12. Spatial-temporal patterns for rule 184 with distinct-input partially-local connections: (a) The system size is 129, and the number of time steps is 1572; (b) the system size is 549, and the number of time steps is 393.

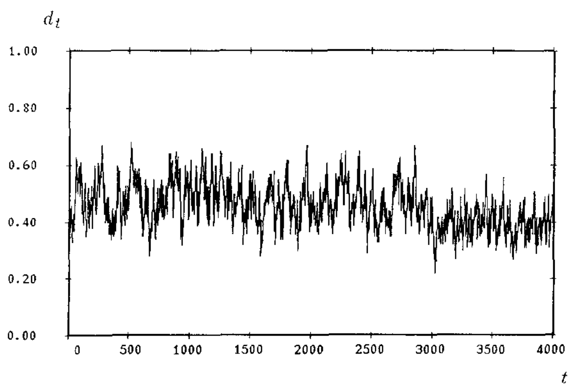


(b)

Fig. 12. (*Continued*)

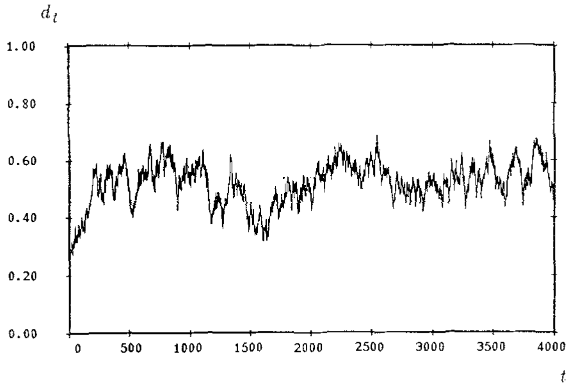
Section 4.2, even when the return map for the density in mean-field theory can be of a nonlinear form, it does not necessarily have chaotic attractors. Then, if the data are taken from simulations or experiments, there is only a dot on the plot corresponding to a fixed-point solution.

Except for smaller system sizes (e.g., $N=70$), when the dynamics is more easily locked in periodic oscillations, the random fluctuations as

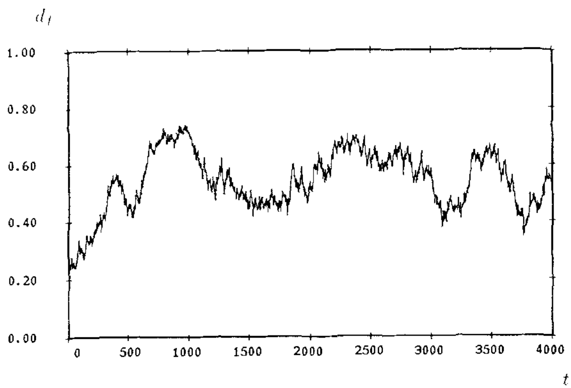


(a)

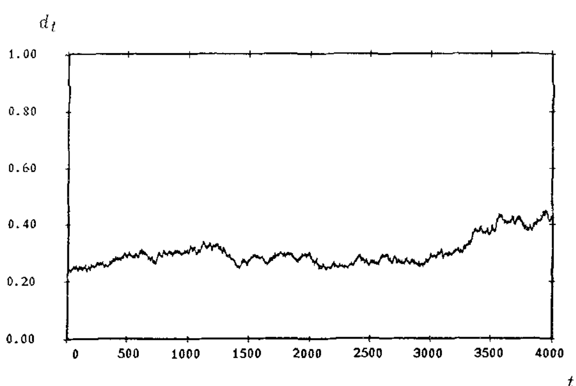
Fig. 13. The density fluctuation of rule 184 with distinct-input partially-local connections, for system size equal to (a) 100; (b) 501; (c) 1021; and (d) 5022. It is clear that the magnitude of the fluctuation becomes smaller for larger systems.



(b)



(c)



(d)

Fig. 13. (*Continued*)

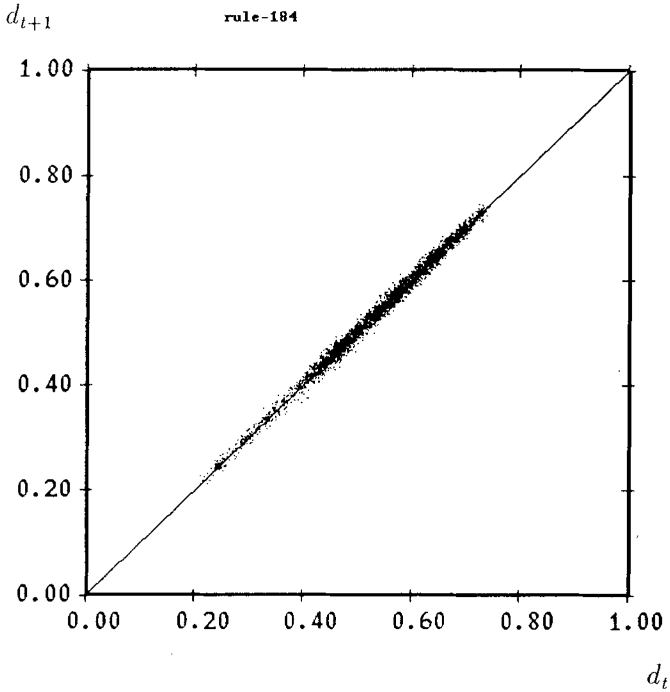


Fig. 14. Numerically determined return map (d_{t+1} versus d_t) for rule 184 with distinct-input partially-local connection, at system size 1021.

illustrated in Fig. 13 seem to go on forever, i.e., these are not transient phenomena. Although it is common for open systems to have fluctuations in macroscopic quantities, due to the fluctuation of the environment, it is nevertheless rare for a closed deterministic system to have them.

In order for some macroscopic quantities to fluctuate in isolated systems, there should be a certain cooperative interaction among the components. Take the density of 1's, for example: if some components switch their state values from 0 to 1, then only when other components, including those that are far away, also have the tendency to switch their state values from 0 to 1 will the density of 1's increase. The fluctuation of density in rule 184 indicates that a large number of components in the system participate in a cooperative dynamics (they comprise a *coherent cluster*).[9]

We can understand now why the magnitude of the density fluctuation becomes smaller as the system becomes larger. It is because the size of the coherent cluster does not increase with the system size, or increases less

[9] For a relevant discussion of the cooperative dynamics in high space dimension (dimension $\geqslant 5$) cellular automata and coupled map lattice, see ref. 6.

than linearly with the system size (though I do not have a quantitative measure of the size of the cluster). As a result, the number of independent coherent clusters is larger when the system becomes larger, and averaging over more of the coherent clusters reduces the magnitude of the fluctuation.

There is a similar discussion of the fluctuation of the macroscopic quantities in locally-connected many-degrees-of-freedom dynamical systems with continuous state variables, such as the coupled map lattices.[3, 19] It is argued in these papers that in locally-coupled maps, the correlation length (corresponding to the coherent cluster size in our case) cannot be infinity in the chaotic regime. More detailed study shows that the correlation length is inversely proportional to the square root of the (maximum) Lyapunov exponent.[25, 4] Other studies show that the correlation length is inversely proportional to the Lyapunov exponent,[3, 19, 52] though there seems to be an error in the argument which assumes that the spatial expansion rate of perturbation is independent of the Lyapunov exponent.

No matter which result is correct, they all agree that the correlation length can be infinite or comparable with the system size only when some measure of the "chaosness" such as the Lyapunov exponent is zero or small, i.e., these systems with continuous state variables are at the edge of chaos.

For systems with discrete state variables such as cellular automata, the Lyapunov exponent cannot be defined. It is the expansion rate of perturbation that is used to measure the chaosness.[47, 37, 75], [10] Analogous to the coupled-map lattice system, where the correlation length diverges when the Lyapunov exponent approaches zero, here the existence of large coherent clusters should also correspond to the almost zero expansion rate of perturbations, or consistent with the fact that the system is at the edge of chaos.

6.2. Long Transients Reaching a Consensus State

Figure 15a shows a spatiotemporal pattern for rule 184 with a distinct-input fully-nonlocal connection. Again, one can notice the fluctuation of density, which is even more prominent than that in Fig. 12a. Similar to the distinct-input partially-local case, the magnitude of the fluctuation becomes smaller for larger system sizes, as evidenced by the spatiotemporal pattern in Fig. 15b as well as the plots of the density as a function of time for two different system sizes (Fig. 16). There is, however, a major difference between the two: the density fluctuation here is a transient behavior; whereas in the distinct-input partially-local connection cases the fluctuation does not seem to be a transient.

[10] For another discussion on the expansion rate of perturbation in the context of coupled map lattices, see ref. 21.

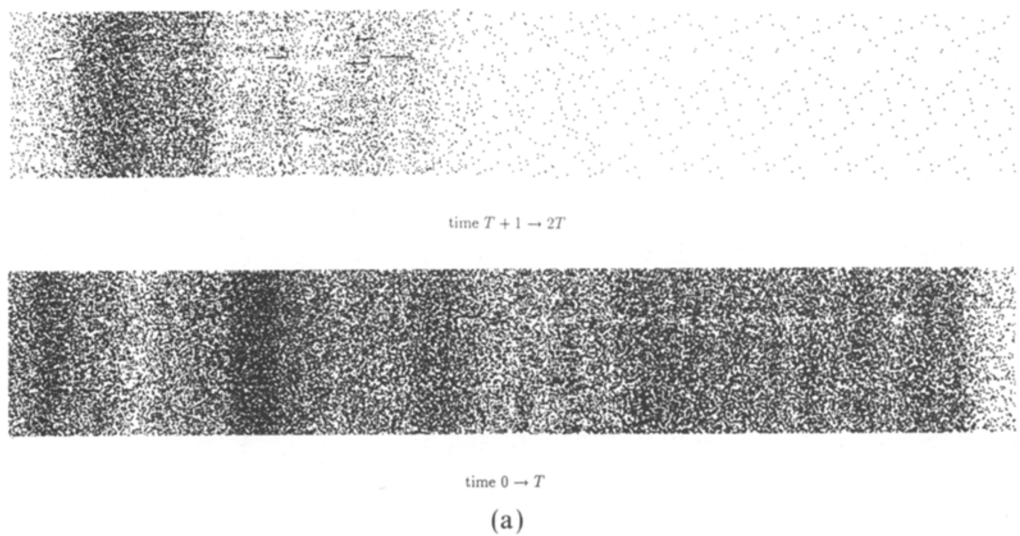
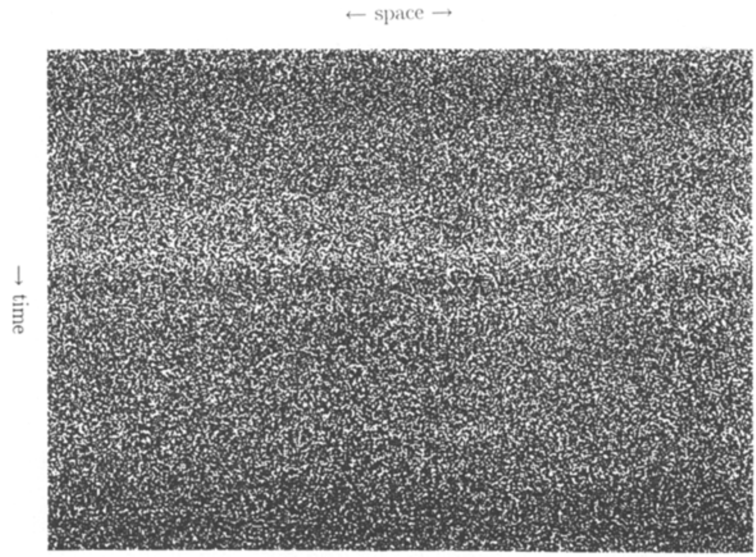


Fig. 15. Spatial-temporal patterns for rule 184 with distinct-input fully-nonlocal connections: (a) The system size is 129, and the number of time steps is 1572; (b) the system size is 549, and the number of time steps is 393.



(b)

Fig. 15. (*Continued*)

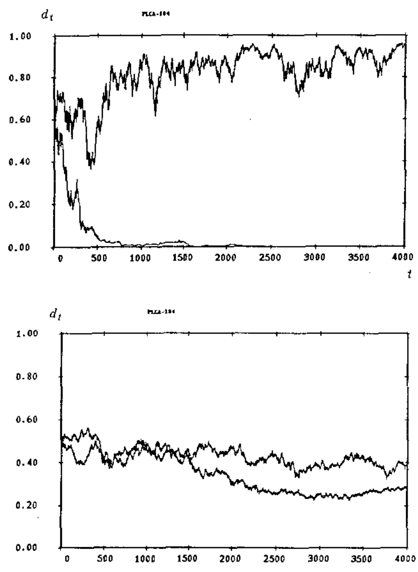


Fig. 16. The density fluctuation of rule 184 with distinct-input fully-nonlocal connections, for system size equal to (two samples for each case) (a) 500 and (b) 5000.

After the transient dies out, the system settles in either one of the two types of attractors. Both types of attractors are periodic (or static), but one has density very close to 0 and another very close to 1. These low- or high-density configurations can be called the *consensus states.* Sometimes, after waiting for a long time, one can be sure which attractor the system will settle into, because the trend is obvious. However, on other occasions, the picture is not that clear. The density can fluctuate toward one of the attractors at the beginning, then for some reason swing back toward another attractor. Right now, it is poorly understood what has happened.

Crutchfield and Kaneko[10] distinguish two classes of the transient behaviors in many-degrees-of-freedom dynamical systems: the first one is monotonic and there is a convergence toward the limiting attractor; the second one is quasistationary, which maintains an attractor-like dynamics for a very long time before failing to the real attractors.[60]

The transient behavior for rule 184 discussed here seems to belong to neither of them. The more appropriate picture here is that the system can look uncertain about which attractor it will fall into eventually, and it performs a random walk before finally "finding out" one of the two attractors. More studies on the mechanism for this transient are needed, especially the phase space structures.[12]

There is another way to classify different classes of the transient behavior: to see how the transient time increases with system size. Although it is shown that coupled map lattices with monotonic transient have power law or exponential divergence of the transient time with the system size, whereas those with quasistationary transient have super-exponential divergence,[10, 24] it is not clear whether the same conclusion holds for other systems.

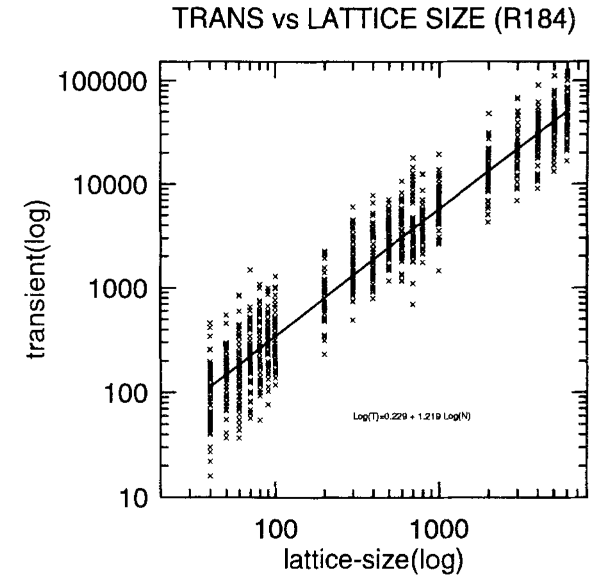
I calculate the transient time T as the function of the system size for rule 184 with distinct-input fully-nonlocal connections. A simple algorithm for determining both cycle length and transient is used, in which the configuration at time t is compared with that at time $2t$ to see whether they match (for details, see pp. 7 and 517 of ref. 28, and ref. 11).

The result is shown in Fig. 17a. For a given system size, 50 different initial configurations and the initial wirings are sampled. The maximum system size in this simulation is 6000. The best-fit straight line for all 1000 data points (following the *fit.c* program in ref. 51),

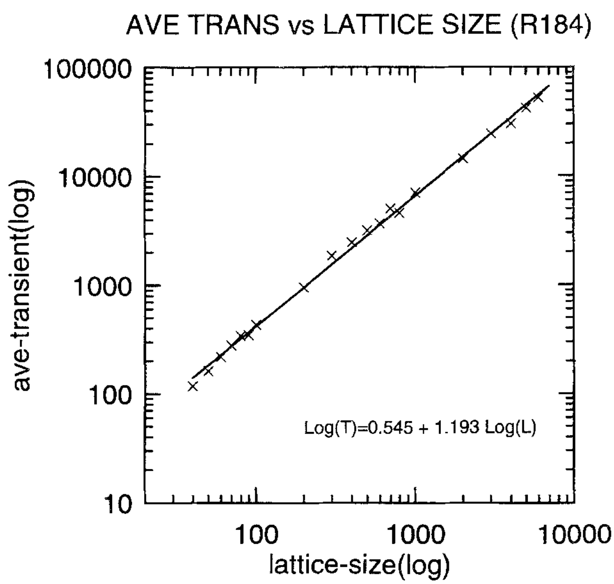
$$\log(T) = b + \alpha \log(N) \tag{6.1}$$

gives $b = 0.229 \pm 0.071$ and $\alpha = 1.219 \pm 0.011$. Figure 17b shows the fitting for the average transient times as the function of the system size, which gives $b = 0.545 \pm 0.089$ and $\alpha = 1.193 \pm 0.014$, or

$$T_{\text{av}} \approx 1.7N^{1.2} \tag{6.2}$$



(a)



(b)

Fig. 17. Transient time as a function of the system size (on log–log scale) for rule 184 in distinct-input fully-nonlocal connections. (a) All data points (1000 points). The fitting line gives $b = 0.229 \pm 0.071$ and $\alpha = 1.219 \pm 0.011$; (b) the average transient times (20 points); the fitting line gives $b = 0.545 \pm 0.089$ and $\alpha = 1.193 \pm 0.014$.

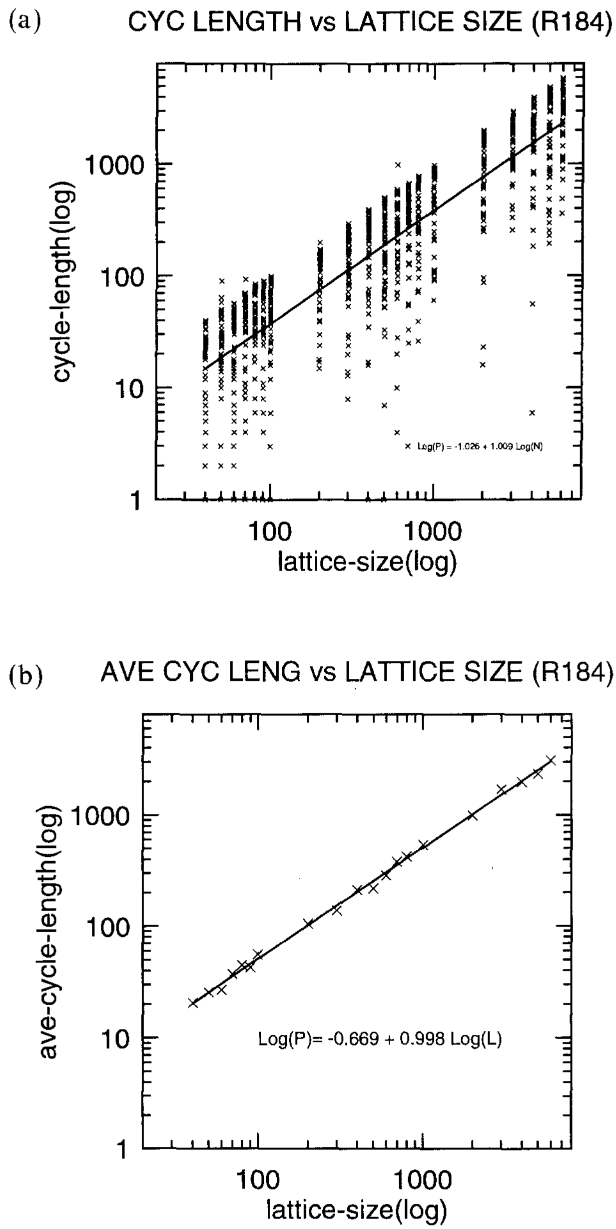


Fig. 18. Cycle length as a function of the system size (on log–log scale) for rule 184 in distinct-input fully-nonlocal connections. (a) All data points (total 1000 points). The fitting line gives $\alpha = 1.009 \pm 0.019$; (b) the average cycle lengths (20 points); the fitting line gives $\alpha = 0.998 \pm 0.011$.

This power law divergence of the transient time with the system size is close to that in elementary local cellular automaton rule 110, which has $T \sim N^{\alpha}$ with $\alpha \approx 1.08$.[38]

There are several practical issues concerning the accurate determination of the exponent α. This paper will not settle these issues, which will be left for future publications. First, some preliminary results show that the transient time for the degeneracy-permitted case is much shorter, and the fitting for those data gives an exponent indistinguishable from 1. It is not clear whether α should be strictly larger than 1. Second, it is not clear what the typical distribution of the transient time is. If the distribution has some unusual long tails, a single data point at the tail can change the average value completely. In that case, averaging over logarithmic transients should be better.

Figure 18a shows the cycle length P as a function of the system size. The fitting line for all 1000 data points gives $\alpha = 1.009 \pm 0.019$. Figure 18b shows the average cycle lengths as a function of the system size. The fitting line gives $\alpha = 0.998 \pm 0.011$. The linear increase of the cycle length with the system size is because the only few surviving 1's (if most of the site values are 0's) is selected or passed on to another site through the wiring; and the random wiring makes it possible that this site value 1 will eventually return to the original site, with the path length being proportional to the system size. Unlike the transient time scaling, the cycle length scaling seems to be trivial.

In summary, the density of 1's in rule 184 with the distinct-input fully-nonlocal wiring fluctuates like a random walk. The magnitude of the fluctuation is decreased when the system size is increased. Since zero and full density regions are traps to the random walker, the fluctuation stops after a trap is reached. From a simplified "mean-field" picture, it can be estimated that $\alpha = 1$ exactly.[57] Although it is not conclusive, if it is established that α is strictly larger than 1, we will know that mean-field theory fails to capture the subtle structures in the underlying wiring diagram. More numerical results will be included in a future publication[41] and implications for the so-called "group meeting problem" (on whether a group meeting can reach a consensus, and how long it takes) will be given elsewhere.[40]

ACKNOWLEDGMENTS

I would like to thank Jim Crutchfield, Kuni Kaneko, Stuart Kauffman, Chris Langton, Seth Lloyd, and Crayton Walker for discussions, valuable suggestions, and recommending references, and James Theiler for a critical reading of the draft. Melanie Mitchell and Steward Wilson made me aware of the term "multiplexer", and I thank them for the pointers. I also thank Joachim Krug and Mitchell Feigenbaum for giving me the preprints before publication. The work at Santa Fe Institute was funded by the MacArthur Foundation, the National Science Foundation (NSF grant PHY-87-14918), and the Department of Energy (DOE grant DE-FG05-88ER25054). The revision of the draft was carried out at Rockefeller University, where I was supported by the DOE grant DE-FG02-88-ER13847.

REFERENCES

1. P. Bak, C. Tang, and K. Wiesenfeld, Self-organized criticality: An explanation of $1/f$ noise, *Phys. Rev. Lett.* **59**:381–384 (1987).
2. E. Berlekamp, J. Conway, and R. Guy, *Winning Ways for Your Mathematical Plays*, Vol. 2 (Academic Press, 1984).
3. T. Bohr, G. Grinstein, Y. He, and C. Jayaprakash, Coherence, chaos, and broken symmetry in classical, many-body dynamical systems, *Phys. Rev. Lett.* **58**(21):2155–2158 (1987).
4. T. Bohr and O. B. Christensen, Size dependence, coherence, and scaling in turbulent coupled-map lattices, *Phys. Rev. Lett.* **63**(20):2161–2164 (1989).
5. J. Carlson, J. Chayes, E. Grannan, and G. Swindle, Self-organized criticality in sandpiles: Nature of the critical phenomenon, *Phys. Rev. A* **42**(4):2467–2470 (1990).
6. H. Chaté and P. Manneville, Collective behaviors in spatially extended systems with local interactions and synchromous updating, *Prog. in Theor. Phys.* **87**:1–60 (1992).
7. A. Chhabra, M. Feigenbaum, L. Kadanoff, A. Kolan, and I. Procaccia, Sandpiles, avalanches, and the statistical mechanics of non-equilibrium stationary states, preprint (1992).
8. E. F. Codd, *Cellular Automata* (Academic Press, New York, 1968).
9. J. P. Crutchfield and K. Kaneko, Phenomenology of spatio-temporal chaos, in *Directions in Chaos*, Bailin Hao, ed. (World Scientific, Singapore, 1987).
10. J. P. Crutchfield and K. Kaneko, Are attractors relevant to turbulence? *Phys. Rev. Lett.* **60**(26):2715–2718 (1988).
11. J. P. Crutchfield, Hunting for transients and cycles, unpublished notes (March 1988).
12. J. P. Crutchfield, Subbasins, portals, and mazes: Transients in high dimensions, *Nucl. Phys. B* (*Proc. Suppl.*) **5A**:287–292 (1988).
13. F. Fogelman-Soulié, Parallel and sequential computation on Boolean networks, *Theor. Computer Sci.* **40**:275–300 (1985).
14. E. Fredkin and T. Toffoli, Conservative logic, *Int. J. Theor. Phys.* **21**(3/4):219–253 (1982).
15. P. Gacs, G. L. Kurdyumov, and L. A. Levin, One-dimensional uniform arrays that wash out finite islands, *Prob. Peredachi. Inf.* **14**:92–98 (1978).
16. J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields* (Springer-Verlag, 1983).
17. H. Gutowitz, J. Victor, and B. Knight, Local structure theory for cellular automata, *Physica D* **28**:18–48 (1987).
18. H. Gutowitz, Hierarchical classification of cellular automata, *Physica D* **45**(1–3):136–156 (1990).
19. G. Grinstein, Stability of nonstationary states of classical, many-body dynamical systems, *J. Stat. Phys.* **5**(5/6):803–815 (1988).
20. E. Jen, Global properties of cellular automata, *J. Stat. Phys.* **43**(1/2):219–242 (1986); Invariant strings and pattern-recognizing properties of one-dimensional cellular automata, *J. Stat. Phys.* **43**(1/2):243–265 (1986).
21. K. Kaneko, Lyapunov analysis and information flow in coupled map lattices, *Physica D* **23**:436–447 (1986).
22. K. Kaneko, Chaotic but regular posi-nega switch among coded attractors by cluster-size variation, *Phys. Rev. Lett.* **63**(3):219–223 (1989).

23. K. Kaneko, Clustering, coding, switching, hierarchical ordering, and control in a network of chaotic elements, *Physica D* **41**:137–172 (1990).
24. K. Kaneko, Super-transients, spatio-temporal intermittency and stability of fully developed spatio-temporal chaos, *Phys. Lett. A* **149**(2, 3):105–112 (1990).
25. F. Kaspar and H. G. Schuster, Scaling at the onset of spatial disorder in coupled piecewise linear map, *Phys. Lett. A* **113**:451–453 (1986).
26. S. A. Kauffman, Metabolic stability and epigenesis in randomly constructed genetic nets, *J. Theor. Biol.* **22**:437–467 (1969).
27. S. A. Kauffman, Emergent properties in randomly complex automata, *Physica D* **10**:145–156 (1984).
28. D. E. Knuth, *The Art of Computer Programming*, Vol. 2: *Seminumerical Algorithms* (Addison-Wesley, 1981).
29. J. Krug, J. Socolar, and G. Grinstein, Surface fluctuations and criticality in a class of 1d sandpile models, preprint (1992).
30. C. Langton, Studying artificial life with cellular automata, *Physica D* **22**(1–3):120–149 (1986).
31. C. Langton, Computation at the edge of chaos: Phase transitions and emergent computation, *Physica D* **42**:12–37 (1990).
32. C. Langton, Computation at the Edge of Chaos, Ph.D. Thesis, University of Michigan (1990).
33. W. Li, Power spectra of regular languages and cellular automata, *Complex Systems* **1**(1):107–130 (1987).
34. W. Li, Complex patterns generated by next nearest neighbors cellular automata, *Computer Graphics* **13**(4):531–537 (1989).
35. W. Li, Problems in Complex Systems, Ph.D. Thesis, Columbia University, New York (1989) (Available from University Microfilm International, Ann Arbor, Michigan).
36. W. Li and N. Packard, Structure of the elementary cellular automata rule space, *Complex Systems* **4**(3):281–297 (1990).
37. W. Li, N. Packard, and C. Langton, Transition phenomena in cellular automata rule space, *Physica D* **45**(1–3):77–94 (1990).
38. W. Li and M. Nordahl, Transient behavior of cellular automaton rule 110, *Phys. Lett. A*, to appear (1992).
39. W. Li, Parametrizations of cellular automata rule space, in preparation.
40. W. Li, Group meeting problems, in preparation.
41. W. Li, Dynamical behavior of coupled selectors, in preparation.
42. K. Lindgren and M. Nordahl, Universal computation in simple one-dimensional cellular automata, *Complex Systems* **4**(3):299–318 (1990).
43. B. D. Lubachesky, Efficient parallel simulations of asynchronous cellular arrays, *Complex Systems* **1**(6):1099–1123 (1987).
44. J. L. Marroquín and A. Ramírez, Stochastic cellular automata with Gibbsian invariant measures, *IEEE Trans. Information Theory* **37**(3):541–551 (1991).
45. P. C. Matthews and S. H. Strogatz, Phase diagram for the collective behavior of limiting-cycle oscillators, *Phys. Rev. Lett.* **65**:1701–1704 (1990).
46. R. May, Simple mathematical models with very complicated dynamics, *Nature* **261**:459–467 (1976).
47. N. Packard, Complexity of growing patterns in cellular automata, in *Dynamical Systems and Cellular Automata*, J. Demongeot, E. Goles, and M. Techuente, eds. (Academic Press, 1985).
48. N. Packard, Adaptation toward the edge of chaos, in *Complexity in Biological Modeling*, S. Kelso and M. Shlesinger, eds. (World Scientific, Singapore, 1988).
49. P. Peretto and J.-J. Niez, Stochastic dynamics of neural networks, *IEEE Trans. Systems, Man, Cybernet.* **16**(1):73–83 (1986).

50. C. Prado and Z. Olam, Inertia and break of self-organized criticality in sandpile cellular-automata model, *Phys. Rev. A* **45**(2):665–669 (1992).
51. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C* (Cambridge University Press, Cambridge, 1988).
52. D. R. Rasmussen and T. Bohr, Temporal chaos and spatial disorder, *Phys. Lett. A* **125**(2, 3):107–110 (1987).
53. L. S. Schulman and P. E. Seiden, Statistical mechanics of a dynamical system based on Conway's Game of Life, *J. Stat. Phys.* **19**:293–314 (1978).
54. R. Shaw, *The Dripping Faucet as a Model Chaotic System* (Aerial Press, 1984).
55. R. J. Smith, *Circuits, Devices, and Systems—A First Course in Electrical Engineering* (Wiley, 1966, 1984).
56. J. Theiler, Mean field analysis of systems that exhibit self-organized criticality, Center for Nonlinear Studies preprint, Los Alamos National Lab (1991).
57. J. Theiler, private communication.
58. T. Toffoli and N. Margolus, *Cellular Automata Machine—A New Environment for Modeling* (MIT Press, 1987).
59. K. Tsang, R. Mirollo, S. Strogatz, and K. Wiesenfeld, Dynamics of a globally coupled oscillator array, *Physica D* **48**:102–112 (1991).
60. D. K. Umberger, C. Grebogi, E. Ott, and B. Afeyan, Spatiotemporal dynamics in a dispersively coupled chain of nonlinear oscillators, *Phys. Rev. A* **39**:4835–4842 (1989).
61. J. von Neumann, *Theory of Self-Reproducing Automata*, A. W. Burks, ed. (University of Illinois Press, 1966).
62. G. Y. Vichniac, P. Tamayo, and H. Hartman, Annealed and quenched inhomogeneous cellular automata (ICA), *J. Stat. Phys.* **45**(5/6):875–883 (1986).
63. C. C. Walker and W. R. Ashby, On temporal characteristics of behavior in a class of complex systems, *Kybernetik* **3**:100–108 (1966).
64. C. C. Walker, Behavior of a class of complex systems: The effect of system size on properties of terminal cycles, *J. Cybernet.* **1**(4):57–67 (1971).
65. C. C. Walker, Predictability of transient and steady-state behavior in a class of complex binary sets, *IEEE Trans. Systems, Man, Cybernet.* **3**(4):433–436 (1973).
66. C. C. Walker, Stability of equilibrial states and limit cycles in sparsely connected, structurally complex Boolean nets, *Complex Systems* **1**(6):1063–1086 (1987).
67. C. C. Walker, Attractor dominance patterns is sparsely connected Boolean nets, *Physica D* **45**(1–3):441–451 (1990).
68. G. Weisbuch, *Complex Systems Dynamics* (Addison-Wesley, 1991).
69. K. Wiesenfeld and P. Hadley, Attractor crowding in oscillator arrays, *Phys. Rev. Lett.* **62**:1335–1338 (1989).
70. W. Wilbur, D. Lipman, and S. Shamma, On the prediction of local patterns in cellular automata, *Physica D* **19**:397–410 (1986).
71. S. Wolfram, Statistical mechanics of cellular automata, *Rev. Mod. Phys.* **55**:601–644 (1983).
72. S. Wolfram, Universality and complexity in cellular automata, *Physica D* **10**:1–35 (1984).
73. S. Wolfram, Computation theory of cellular automata, *Commun. Math. Phys.* **96**:15–57 (1984).
74. S. Wolfram, Twenty problems in the theory of cellular automata, *Physica Scripta* **T9**:170–183 (1985).
75. Appendix: Properties of the $k=2$ $r=1$ cellular automata, in *Theory and Applications of Cellular Automata*, S. Wolfram, ed. (World Scientific, Singapore, 1986).
76. W. Wootters and C. G. Langton, Is there a sharp phase transition for deterministic cellular automata? *Physica D* **45**(1–3):95–104 (1990).