Phenomenology of Non-local Cellular Automata

Wentian Li

SFI WORKING PAPER: 1991-01-001

SANTA FE INSTITUTE

Phenomenology of Non-local Cellular Automata

Wentian Li
Santa Fe Institute, 1120 Canyon Road, Santa Fe, NM 87501.

January 23, 1991

Abstract. The dynamical systems with non-local connections have potential applications to economic and biological systems. This paper studies the dynamics of non-local cellular automata. In particular, all two-state three-input non-local cellular automata are classified according to the dynamical behaviors starting from random initial configurations and random wirings. The rule space is studied with the mean-field parameterization which provides an improvement over the previous used "λ parameterization". The concept of *robust* universal computation, concerning the ability for a system to do universal computation with random setups of initial condition, is introduced. It is argued that since non-local connections provide a handy way for information transmission, it is much easier for a non-local cellular automaton to be a universal computer than for a local one, though it may not be robust. A particularly interesting "edge of chaos" non-local cellular automaton, the rule 184, is studied in detail. It exhibits irregular fluctuations of the density, large coherent structures, and long transient times.

Contents

1 Introduction **3**
- 1.1 Why study non-local cellular automata? 3
- 1.2 Wiring schemes of non-local connection 4
- 1.3 Topics to be discussed in this paper 6

2 Review of the elementary local cellular automata **8**
- 2.1 Rule tables, mean-field parameters and the λ parameter 11
- 2.2 Classification of the local cellular automata rules 12
- 2.3 Rule space with the mean field parameterization 14

3 Elementary cellular automata with partially-local and fully non-local connections **16**
- 3.1 Classification of the partially-local rules, and their rule space . 17
- 3.2 Classification of the fully non-local rules, and their rule space . 22

4 Bifurcation-like phenomena in multi-parameter dynamical systems **26**
- 4.1 Critical surfaces in higher dimensional space 26
- 4.2 The return map of density in mean field theory: and the determination of the critical surface 29

5 Further comments on the relationship between complex dynamics and the universal computation **34**
- 5.1 Robust universal computation 34
- 5.2 Computation at the level of bits 35

6 Density fluctuation, coherent structures, long transients and the edge of chaos behavior for rule 184 **39**
- 6.1 Irregular fluctuation of the density 43
- 6.2 Long transients before settling down in one of the limiting attractors 46

1. Introduction

1.1 Why study non-local cellular automata?

Is is known that many systems with a large number of interactive components can exhibit emergent properties and dynamical behaviors which are not deductible from its components. There could be a whole spectrum of emergent properties and dynamical behaviors crucially depending on which rules are used. The 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 the 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; see, for example, the molecular interaction in 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 non-local connection at the low-level description of the system.

The dynamical behaviors of systems with local interactions are studied in partial differential equations, coupled ordinary differential equations [49], coupled map lattices [5], and cellular automata [50, 59]. It is well understood now that by moving from one extreme point to another in the dynamical 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 (spatio-temporal intermittency, breaking up of the domain walls, complex glider interactions, etc.), and the global chaotic dynamics (turbulence phase).

For systems with non-local connections, the dynamics does not necessary differ from those of the locally-connected system, especially if the locally-connected system already exhibits a globally chaotic dynamics. Nevertheless, the modes for "edge of chaos" dynamics in the non-local 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 non-local systems. One purpose of this paper is to identify the edge of chaos dynamics in the non-local systems.

The study of the dynamical behaviors in non-locally 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 brains and how information of the stock price is shared by agents who read the same ticker-tape—these fields could be good candidates for applications of the non-local systems studied here.

To start the study of dynamical behaviors in non-local systems, I will concentrate on a particular class of systems: the *non-local 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 value of each component is discrete (and the state value is probably finite); (b) the dynamical rule applied to one component is the same with those applied to all other components; (c) the updating of the state values of every component is synchronized. It is almost the same with the definition of cellular automata except that non-local connection is allowed.

If the requirement (b) is violated, we have, for example, the Kauffman's network [24, 25], or other inhomogeneous random networks (the inhomogeneous cellular automata [51] are the examples with local connections). If the requirement (c) is violated, we have, for example, the Hopfield's network [17], many neural network models [37, 43] (the filter cellular automata [40, 11] are the examples with local connections). The dynamics for these systems should also be interesting, though 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 k represents the number of inputs each component receives, $k = 3$ throughout the paper. This restriction will 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 the globally coupled maps or oscillators, see, for example, Refs.[20, 21, 57, 48, 35].

The only previous study I am aware of on the similar systems was done by Walker and his co-workers [52, 53, 54, 55, 56]. They also consider $k = 3$ case exclusively, but require 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 loosely referred to as "a class of complex binary nets" or "a sparsely connected Boolean nets." To be consistent, I will call them *partially-local cellular automata*, to distinguish them from the *fully non-local cellular automata*. Various types of connection will be further discussed in the next subsection.

1.2 Wiring schemes of non-local connection

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

$$x_i^{t+1} = f(x_{j_1(i)}^t, x_{j_2(i)}^t, x_{j_3(i)}^t). \tag{1.1}$$

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

We identify four different types of wiring schemes, i.e., how $j_1(i)$, $j_2(i)$, and $j_3(i)$ are chosen. For a 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 between 1

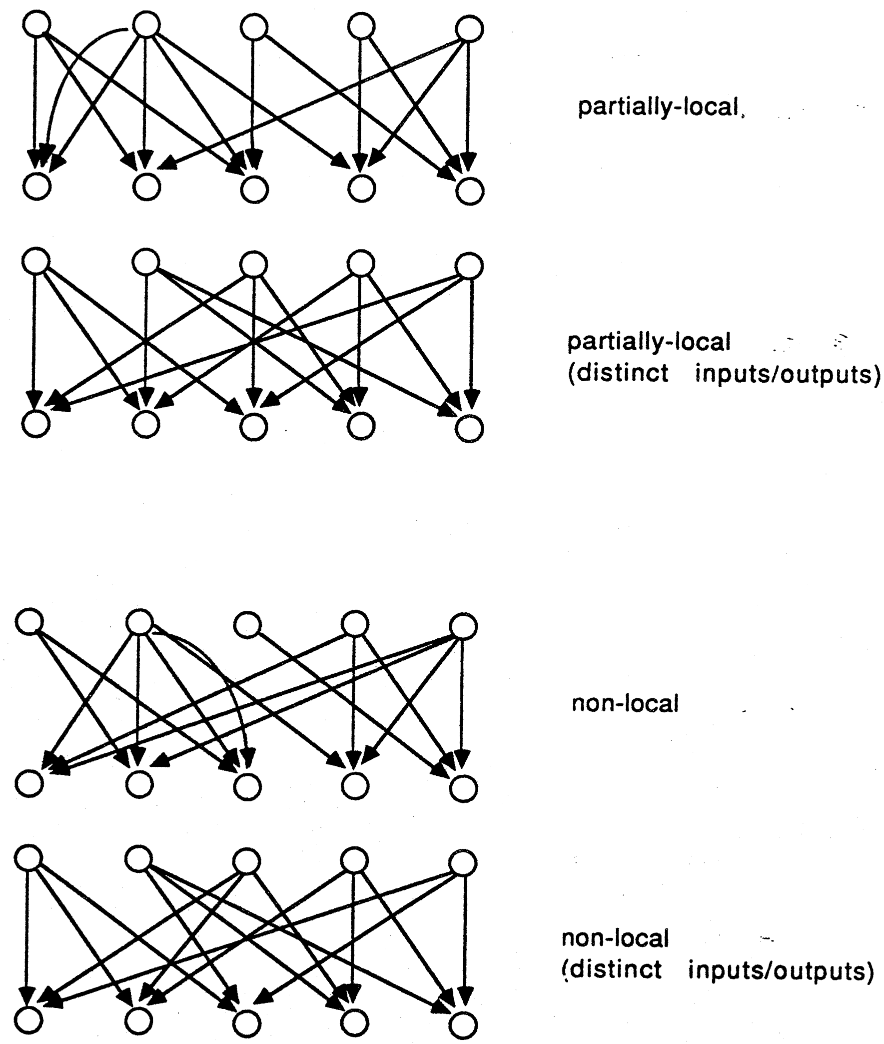


Figure 1: Illustration of the four wring schemes of the non-local connection.
(a) partially-local connection with possibly degenerate inputs;
(b) partially-local connection with distinct inputs;
(c) fully non-local connection with possibly degenerate inputs;
(d) fully non-local connection with distinct inputs.

and N for any i, then assigns the two as the first and the third inputs of site i. The degeneracy of inputs might be possible, whenever two among the three inputs are the same.

(2) Partially-local connection with distinct inputs: $j_2(i) = i$, $j_1(i)$ and $j_3(i)$ are randomly chosen but being checked that none of the two inputs of 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 either two of them are the same, re-generate another index again randomly to replace one of the degenerate index. Repeat the comparison until none of two among them are the same.

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

(4) Fully non-local connection with distinct inputs: all inputs $j_1(i)$, $j_2(i)$, and $j_3(i)$ are randomly chosen, and being checked that none of the two inputs of site i are the same.

It is important to distinguish the partially-local connections from the fully non-local ones, because the former can still share some features with the locally-connected dynamical systems, for example, the single-site "domain wall", whereas the latter will not have anything similar. It is also important to distinguish the case of degenerate inputs from the case of distinct inputs. If, for example, the first input is identical with the second input, the part of the rule 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. The four wiring schemes of non-local connection are illustrated in Fig.1.

Finally, note that there is a difference between *all possible* non-local connections and a *typical* non-local 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 non-local wiring using some random number generators. Similarly, the dynamics of a non-local cellular automaton refers to the dynamics exhibited by that rule with a typical non-local connection.

1.3 Topics to be discussed in this paper

This paper studies the dynamical behavior of the three-input two-state non-local cellular automata, and compares them with those of the corresponding local cellular automata. Special attention will be paid to observations which could lead to new concepts that are absent or less important than the locally-connected dynamical systems. I will discuss three of them: (1) the "cleanness" of the rule space; (2) the robust universal computation; and (3) a new mode of the "edge of chaos" dynamics.

The first topic is about the cellular automata rule space. As studied by Langton, Packard and the author [27, 30, 31, 32], 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 is like the critical point of the phase transition, although it is not a "point" but a high-dimensional surface, and the transition from one dynamical behavior to another can either be sudden (the first order) or continuous (the second order). This study of the structure of the rule space can also be considered as the high-dimensional analogy of the study of bifurcation phenomena, which are done mostly in the lower-dimensional dynamical systems.

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 parameterization [32]. The simplest parameterization, also called the λ-parameterization, is by varying a single parameter which is the fraction of the non-zero entries in the rule table [26], . 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 by increasing the number of states [64]. In the next simplest parameterization, the mean-field parameterization, more parameters are used, which not only counts the fraction of the configurations which map to non-zero values, but also examine what kinds of configurations map to non-zero values. The mean-field parameterization improves the description of the critical surface, but it still fails frequently to make the correct predictions on the behavior of local cellular automata. The study of the non-local cellular automata rule space will show that mean-field parameterization gives a far better prediction of the location of the critical surface. A similar observation that mean-field theory describes much better the non-local systems than the local systems is also made in Ref.[46].

The second topic is on the relationship between a rule's computational ability and its generic dynamical behavior. It is first suggested by Wolfram [60, 62] and further elaborated in [27, 32] that for a local cellular automaton, its ability to do universal computation is related to its exhibiting complex dynamics. One simple argument is that if a cellular automaton can be programmed to do any computation, including the very difficult ones, the computing time will be very long. If the computation is viewed as a dynamical process with the computation being the transient, and the termination of the computation being the end of the transient (or the onset of the limiting dynamics), a universal computer should have an arbitrarily long transient, or the dynamics is "complex."

The above argument is basically correct except one point. A universal computer may carry out computation of a difficult problem by a special arrangement of the initial condition, which on the other hand, will perhaps never be realized if the initial condition is randomly chosen. Actually, it has been checked for a class of cellular automata which are proven to be universal computers that only some of them exhibit moderately complex dynamical behavior if the initial condition is randomly chosen [34]. For a cellular automaton with local connections, the very fact that it can carry out universal computations at least means that it is capable of generating propagating local configurations which carry the information for a computation, which in turn has a strong restriction on the form of the rule. Nevertheless, the connection between the universal computation and the complex dynamics becomes weaker in non-local cellular automata. There are so many non-local cellular automata which are universal computers, but they never exhibit complex dynamics with the randomly chosen initial configurations and wirings. The computation of a difficult program can only be carried out in these non-local cellular automata by specially designing the initial configuration *and* the wiring. Based on this observation, I will suggest a name "robust universal computation" to describe the situation.

The last topic to be discussed in some detail is a new mode of the "edge of chaos" dynamics[1] in non-local cellular automata. The edge of chaos dynamics in locally-connected dynamical systems is studied in several systems, such as the spatio-temporal intermittency in the coupled map lattices, the glider activity in class-4 cellular automata, etc. In these systems, the small but positive Lyapunov exponents, the long-rang correlation, the long transients, and the poor convergence of the statistical quantities are all considered as the hallmarks of the edge of chaos dynamics. In non-local systems, the criteria of the long transients and the poor convergence of the statistical quatities are still applicable, but it is difficult to define the long-range correlation together with any other concepts related to the space. Here a concept of the "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 three-input two-state local cellular automata; Section 3 discusses the corresponding cellular automata with partially local and fully non-local connections, the dynamics are classified and the rule space in mean-field parameterization is shown; Section 4 attempts an understanding of the rule space in mean-field parameterization discussed in the last section; Section 5 discusses the relationship between the universal computation and the complex dynamics, and clarify the concept of robust universal computation; and Section 6 studies the most interesting three-input two-state non-local cellular automaton (rule 184), discusses various features of its dynamics.

2. Review of the elementary local cellular automata

The simplest cellular automata are perhaps those with two inputs and two states. Since there are 2^2 different input configurations, each configuration can map 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., Ref.[9]. Actually, the number of independent two-state two-input rules is only 7. Such seven rules consist

[1]The name "edge of chaos" is first used by Packard [39], other names such as the complex dynamics, critical dynamics, boundary dynamics, in-between dynamics, etc. can also be used.

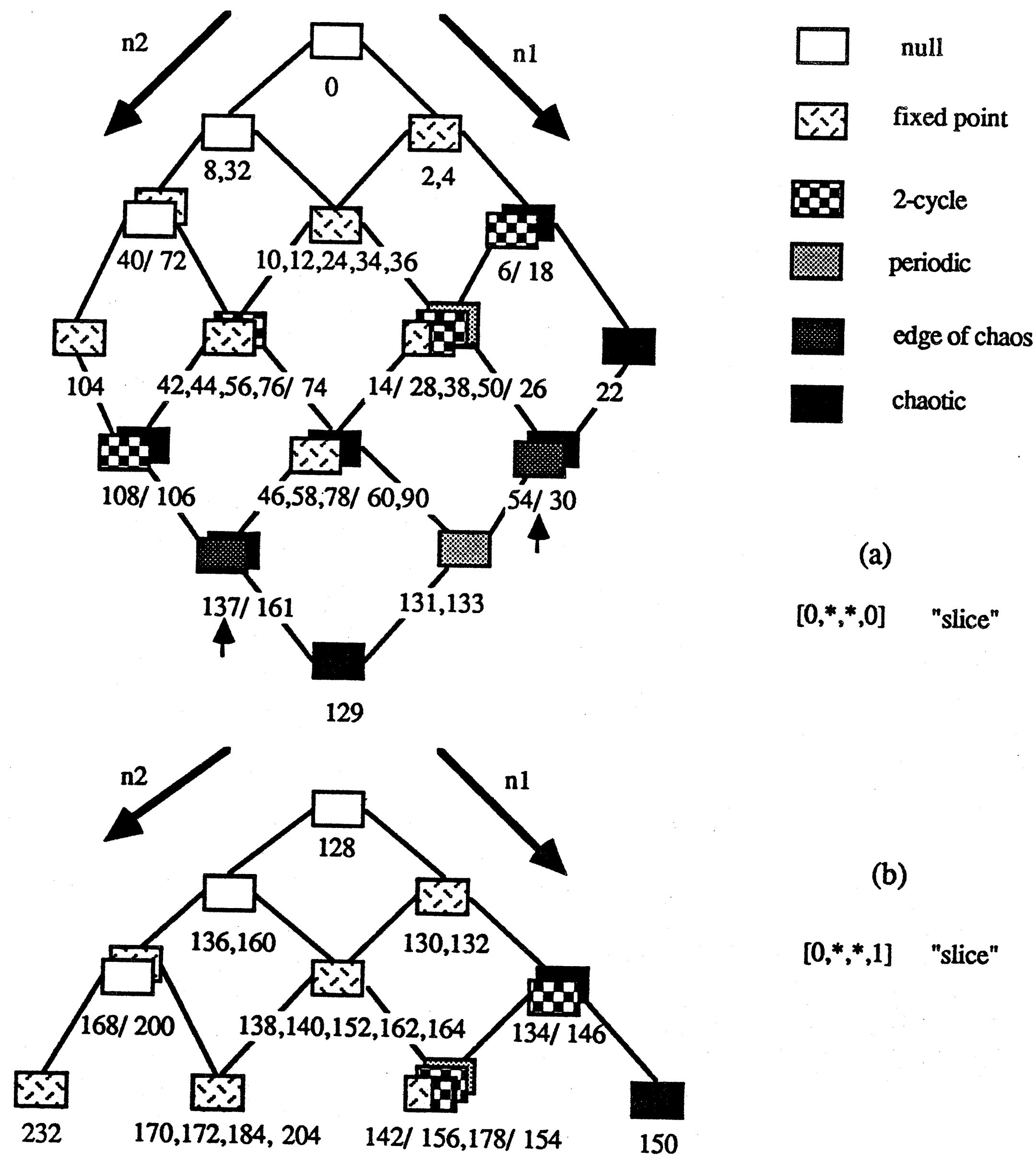


Figure 2: (to be continued)

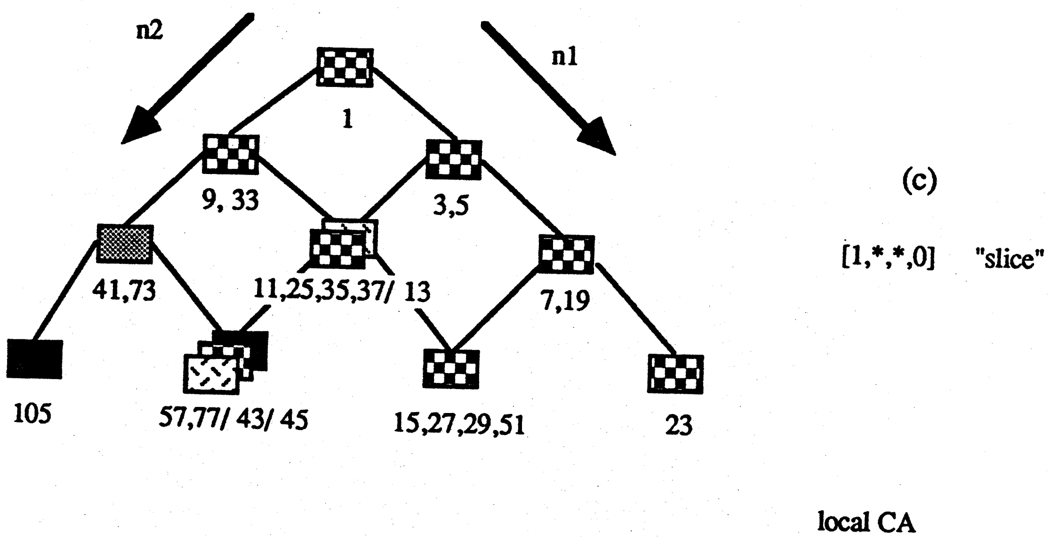


Figure 2: The rule space for local cellular automata with mean-field parameterization. The rule numbers contained in each mean-filed cluster are listed. The second and the third mean-field parameters n_2 and n_3 vary from 0 to 3. The rules with long transients (rules 54 and 137) are marked.
(a) The slice 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 $n_3 = 0$.

a set too small to study the generic structure of the cellular automata rule space.

The next simplest cellular automata are those with two states and three inputs, also called *elementary rules* in Ref.[59]. 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 [52, 59, 31]. The elementary cellular automata with local connections are extensively studied in Refs.[13, 18, 28, 31, 58, 59, 61, 63], among others. This section will review some of the most relevant properties of these rules. For more detail, see the original references.

2.1 Rule tables, mean-field parameters and the 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 $2^3 = 8$. We write

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

or equivalently

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

to specify the rule.

The rule is completely determined by these a_i's, which can either be a *rule table*

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

or a *rule number*, which is the decimal representation of the above binary string [59]:

$$(a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0)_2.$$

For example, rule table $\{0,0,0,0,0,0,0,1\}$ has the rule number 1; and rule table $\{1,0,0,0,0,0,0,1\}$ corresponds to the rule number 129.

Since it is not important of 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 $\overline{1} = 0$. Since the space is not directional, interchanging left and right input 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 meantioned above, one has the 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 the complete information about the rule being represented. Nevertheless, sometimes one might not want to know all the information, and consider two or more rules are "more or less" the same. It is where the mean-field parameters come in [44, 14, 31]. 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 the similar role, because they are related to the input configurations which contain one 1 and two 0's. Similarly, the entries a_3, a_5 and a_6 are all related to the input configurations which contain two 1's and one 0. We define the mean-field parameters

$$n_1 \equiv \text{number of non-zero bits among } a_1,\ a_2 \text{ and } a_4$$

$$n_2 \equiv \text{number of non-zero bits among } a_3,\ a_5 \text{ and } a_6,$$

and for the 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 the right inputs does not change the mean-field parameters.

An even cruder information about a rule table is how many entries in the rule table are non-zero. It is the so-called λ parameter (not normalized). The normalized λ parameter is the fraction of the non-zero entries in a rule table:

$$\lambda \equiv \text{number of non-zero bits among } \{a_i\}\ (i\text{=}0,1,\ldots,7)/8.$$

The information about a rule table provided by the λ parameter seems to be almost minimum, but satisfactorily, 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:

$$\lambda = (n_0 + n_1 + n_2 + n_3)/8$$

2.2 Classification of the local cellular automata rules

The dynamics of a cellular automaton rule typically refers to the 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 should be guaranteed.

One simple classification of all elementary rules is the following, see also the Refs.[60, 31]:

- Null rules: the limiting configuration is all 0's or all 1's.

  0, 8, 32, 40, 128, 136, 160, 168.

- 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 a *), excluding the all 0's or the 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.

- 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 a *). Rules 14 and 142 can also be fixed point rules 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*.

- Periodic rules: the limiting configuration is invariant by applying the updating rule L times, with the cycle length L either independent or weakly dependent on the sequence length (in the latter case, one could introduce a subclass). In particular, rules 131 and 133 exhibit three-cycle dynamics. Rule 73 has regions with chaotic behavior and can be called a locally-chaotic rule [31].

  26, 41, 73, 131 (or 62), 133 (or 94), 154.

- 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 more than linear 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 properties (for example, the transient time).

  54, 137 (or 110).

- Chaotic rules: non-periodic dynamics. They are characterized by the exponential divergence of the cycle length with the system size, and the instability with respect to perturbations.

  18, 22, 30, 45, 60, 90, 105, 106, 129 (or 126), 146, 150, 161 (or 122).

The rule number inside the parenthesis is the representative rule number used in Ref.[63], which is the smallest value among all rule numbers of equivalent rules. The representative rule number used in this paper as well as Ref.[31] is the one with the smallest λ parameter value. If several equivalent rules have the same λ value, then the one with the smallest rule number is picked.

There can be many different classification schemes, depending on, for example, the degree of the coarse graining. The crudest classification scheme might be the one which only distinguishes chaotic and non-chaotic rules.

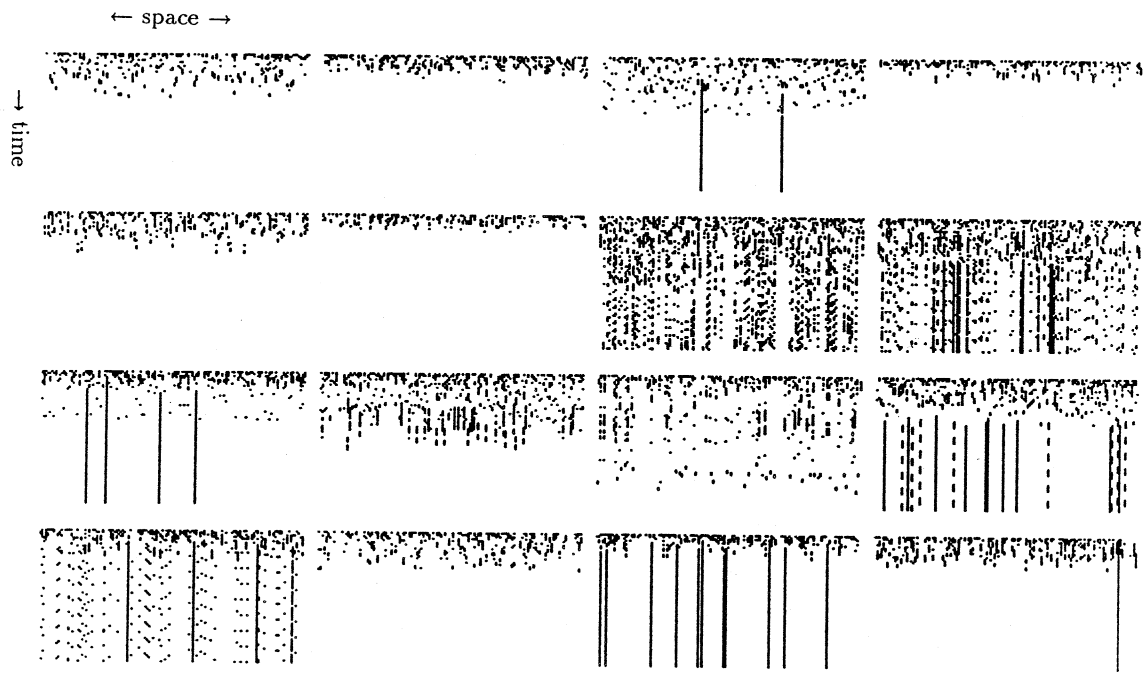


Figure 3: Spatio-temporal patterns for rule 74 of partially-local connections, with possibly the degenerate inputs. Sixteen different initial configurations as well as different random wirings are sampled: some of them lead to all 0 configuration, some lead to the inhomogeneous configurations with a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

Most of the rules are easy to classify whatever the classification scheme is used. For example, a rule is classified as being chaotic by either one of the criteria: large spatio-temporal entropy, positive expansion rate of perturbation, absence of periodic dynamics in the infinite system size limit, etc. 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 parameterization

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 non-negative) 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 original elementary cellular automata rule space has dimension equal to 8, and the distance along each dimensional axis is 1, so it is an 8-dimensional hyper-cube. The location of a rule in the rule space can be considered part of the *genotype* of the rule, and the dynamical behavior exhibited by the rule is the *phenotype* [30, 31]. The interplay between the genotype and the phenotype determines the structure of the rule space, which is studied in Ref.[31].

Here I will review the result about rule space with mean-field parameterization. Although this parameterization is not that perfect for local rules, it gives extremely good picture of the structure of the fully non-local rule space (to be discussed in later sections). Since each mean-field cluster is labeled by four parameters $\{n_0, n_1, n_2, n_3\}$, the space is 4-dimensional, and the distance along n_0 and n_3 is 1, and the distance along n_1 and n_2 is 3.

A 4-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 less number of values for n_0 and n_3. Secondly, 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.[31].

There are four possibilities for the n_0, n_3 values: $n_0 = 0$, $n_3 = 0$; $n_0 = 0$, $n_3 = 1$; $n_0 = 1$, $n_3 = 0$; and $n_0 = 1$, $n_3 = 1$; The last case is equivalent to the first case by interchanging 0 and 1, which left three independent "slices" of the rule space. The first slice $\{n_0, n_1, n_2, n_3\} = \{0, *, *, 0\}$ consists of "nonlinear clusters", the second slice $\{0, *, *, 1\}$ consists of "linear clusters" and the third slice $\{1, *, *, 0\}$ consists of "inversely linear cluster."[31] (* is the "do not care" symbol.) The n_1 and n_2 are increased from 0 to 3 along the two parameter axes. See Fig.2 (a)(b)(c) for the three slices of the rule space.

Usually, each cluster contains more than one rule, and all possible dynamical behaviors exhibited by the rule in that cluster is indicated by different textures of the block. The darker the texture, the more chaotic the dynamics is. 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, 57 (fixed point dynamics), rule 43 (2-cycle dynamics), and rule 45 (chaotic dynamics).

There are several general observations of the rule space:

(1) The slice with nonlinear clusters contains rule 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 the rule space, one observes the familiar bifurcation to chaos series reminiscent of that in the logistic map [36]. Es-

pecially, the three-cycle rules 131, 133 are located inside the chaotic regime, an almost perfect analogy of the 3-cycle window in the logistic map.

(2) The slice with the linear clusters contains mostly fixed point rules, with the exception of the lower-right corner of the space.

(3) The slice with the inversely-linear clusters contains mostly two-cycle rules, with the exception of the lower-left corner of the rule space.

(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 the next section and the section 4.

3. Elementary cellular automata with partially-local and fully non-local connections

It is known quite well how different dynamics are generated in the local elementary cellular automata:

- For null rules, it is usually the case that there is a large percentage of the three-input configurations that lead to the same state value such as 0, and the three-input configuration 000 also leads to 0, so all 0's configuration is the attracting invariant configuration.
- For inhomogeneous fixed point rules, some non-zero state can survive due to the updating rules like 010 $\rightarrow$ 1. And at the mean time, other spatial regions are all 0's.
- For periodic rules, there is an alternating activations 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.
- The rules with complex dynamics is most difficult to understand. But typically, there are spatial regions become either homogeneous or periodic (called *background*), whereas in other regions some moving local configurations are emerged. If the local configuration moves with a constant speed, it is usually called a *glider*; if it moves irregularly (only when the background with which it interacts is periodic), it is called a *defect*, or *domain wall.* (If the distinction between background and the gliders or defects is not that all clear, especially at the early stage of the transient, a name *creature* can also be used [29].) 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 co-exist in an equilibrium state.

- 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 the question of how the non-local connection affect the mechanisms for generating different 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 all 0's configuration is equally strong. Nevertheless, some rules having null dynamics with local connections might have fixed point dynamics with non-local connections. For fixed point rules with a shift, the effect is to increase the cycle length from 1 to a large value depending on the wiring. The case is best illustrated by rule 1, which has updating rule $001 \rightarrow 1$ and all other three-input configurations map to 0. The reason that the cycle length is 1 along certain spatio-temporal direction in the local connection is purely because of the third input being the right input. In a non-local connection, 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 of the components. As for chaotic rules, the effect of introducing non-local connection is not large either. Again, almost all three-input configurations are activated throughout the system. The only class of dynamical behavior that will be changed dramatically is the complex or edge of chaos dynamics, because there are no longer concepts such as the background or the gliders. More will be discussed in section 6.

The characterization of the dynamical behavior of a non-local cellular automaton rule is complicated by the fact that there are so many choices of the random wiring, and it is not guaranteed that all random wirings will give the same dynamics. Although a similar problem exists in local cellular automata since one has to choose many different initial configurations, it deserves more attention for non-local cellular automata because wiring seems to be more important than the initial configuration.

In the following, I will list the numerically observed dynamical behaviors for all elementary cellular automata with partially-local and the fully non-local connections. The definite classification of a rule's dynamics might be impossible as some random wirings give one behavior while other random wirings give another. I will try to make footnotes as much as possible to explain what I have observed.

3.1 Classification of the partially-local rules, and their rule space

The partially-local cellular automata were first studied by Walker and Ashby [52], and some statistical quantities such as the cycle lengths are determined for small system sizes. In this subsection, I will base my classification on the the observation of the spatio-temporal patterns for several randomly chosen initial conditions. Comments are made whenever applicable if the dynamics with the possibly degenerate inputs connections differ from those with the distinct inputs connections. Even when the type of wiring is fixed, different

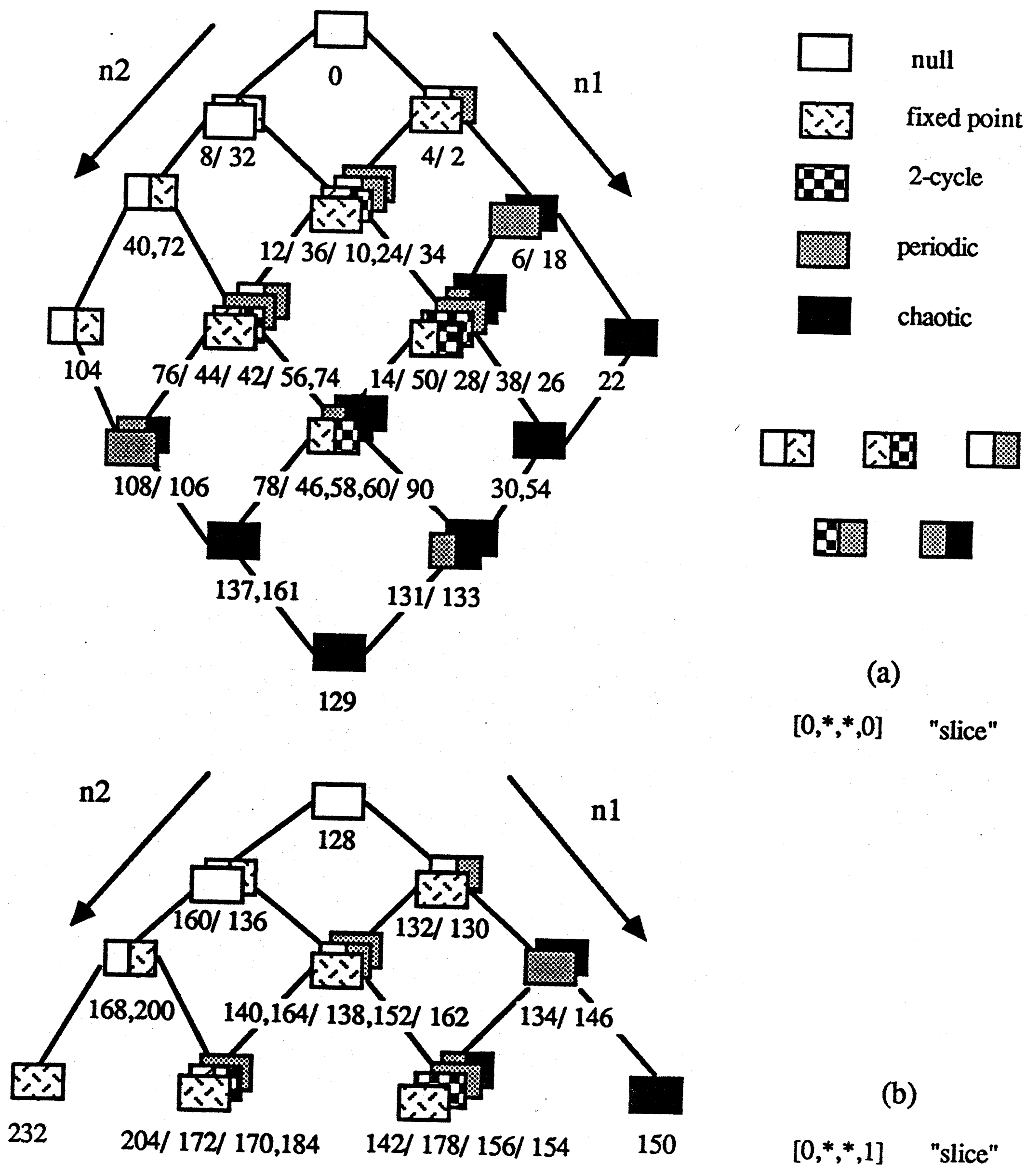


Figure 4: (to be continued)

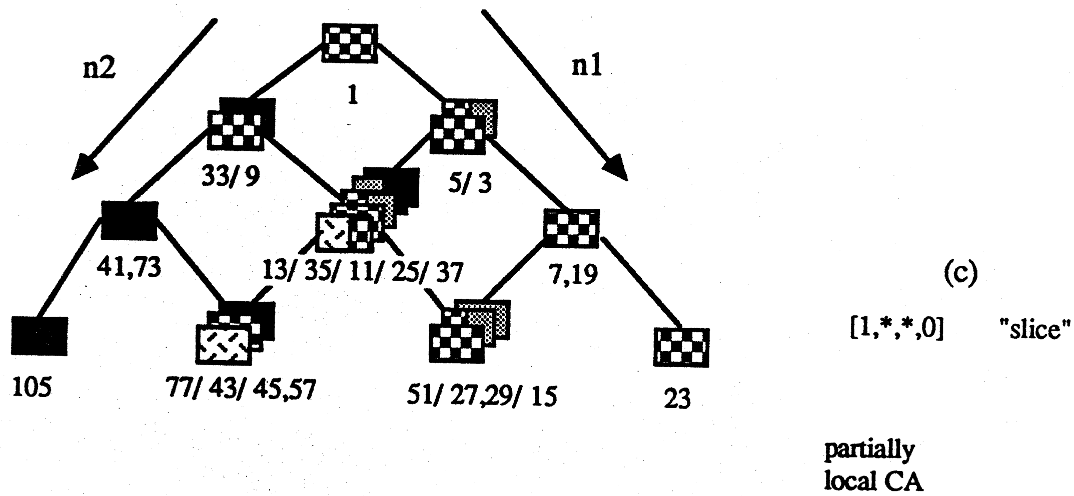


Figure 4: The rule space for partially-local cellular automata with mean-field parameterization, similar to the Fig.2 for the local cellular automata.

samplings of the wiring can lead to different dynamics, as illustrated in Fig.3 for rule 74.

The classification is the following:

- Null dynamics for almost all wirings (N):

  0, 32, 128, 160.

- 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 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.

- Fixed point dynamics (F):

  4, 12, 76, 77, 132, 140, 142, 164, 204, 232.

- These rules have seemingly fixed point dynamics (many domain walls), but actually have local 2-cycle or local periodic modes sometimes. Rules 36 and 44 almost always have local periodic modes. Except these two rules, others are more like fixed point rules for the distinct-input connections (F-P):

  13, 14, 36, 44, 78, 172.

- These rules can have either null, fixed point, or periodic dynamics. For 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.

- Two-cycle dynamics (2):

  1, 5, 7, 19, 23, 33, 35, 43, 50, 51, 178.

- 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.

- Periodic rules. The distinct-input connection makes the cycle lengths much longer (P):

  6, 15, 28, 34, 42, 108, 134, 156, 162, 170, 184.

- 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.

- 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.
- Chaotic rules (C):
  9, 18, 22, 26, 30, 37, 41, 45, 54, 57, 73, 90, 105, 129, 133, 137, 146, 150, 161.

The rule space as "decorated" by the above classification is shown in Fig.4. One can see that it 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 three inputs are not equal. In order to take this into account, considering the following refined mean-field parameters:

$$n_1(0) \equiv \text{number of non-zero bits among } a_1 \text{ and } a_4$$

which is the number of non-zero entries in the rule table which correspond to the three-input configuration which has one 1 among the first and the third inputs; and

$$n_1(1) \equiv a_2.$$

which is the number of non-zero entries in the rule table which correspond to the three-input configuration (actually only one such configuration) which has no 1's among the first and the third inputs. These two parameters simply split the original n_1 parameter: $n_1 = n_1(0) + n_1(1)$. Similarly, define

$$n_2(1) \equiv \text{number of non-zero bits among } a_3 \text{ and } a_6$$

and

$$n_2(0) \equiv a_5$$

to split the original n_2 parameter.

With the refined mean-field parameters, each cluster is labeled by six labels: $\{n_0, n_1(0), n_1(1), n_2(0), n_2(1), n_3\}$. All except ten clusters contain only one rule. These ten exception clusters each contains two rules. These clusters and the dynamics of their rules are:

$\{n_0, n_1, n_2, n_3\}$	rules	behavior
0110	10, 24	(N-P)
0120	42, 56	P, (N-P)
0210	14, 28	(F-P), P
0220	46, 60	(P-C)
0111	138, 152	(N-P)
0121	170, 184	P
0211	142, 156	F, P
1110	11, 25	(2-P),(P-C)
1120	43, 57	2,C
1210	15, 29	P,(2-P)

(3.1)

where N for null, F for fixed point, 2 for 2-cycle, P for periodic, and C for chaotic. If more than one symbol is linked with hyphen, the dynamics can be either with different initial conditions. One can see that even with the refined mean-field parameters, the prediction is not as good as expected.

3.2 Classification of the fully non-local rules, and their rule space

The number of independent rules in fully non-local connection is smaller than that for the local and partially-local connections (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. None of the three inputs should be distinguished as the special one. By considering all possible equivalent relations, there are finally 46 independent rules. 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 the parenthesis.

- Null rules:
  0, 8(32), 40(72), 104, 128, 136(160), 168(200), 232.
- Two-cycle rules:
  1, 3(5), 7(19), 23.
- 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 15 can look like a chaotic rule because the its cycle length can be very long.
  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).
- Chaotic rules:
  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.

Similar to what have been drawn for the local and the partially-local cellular automata, the rule space rules with mean-field parameterization for fully non-local cellular automata is shown in Fig.6. 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

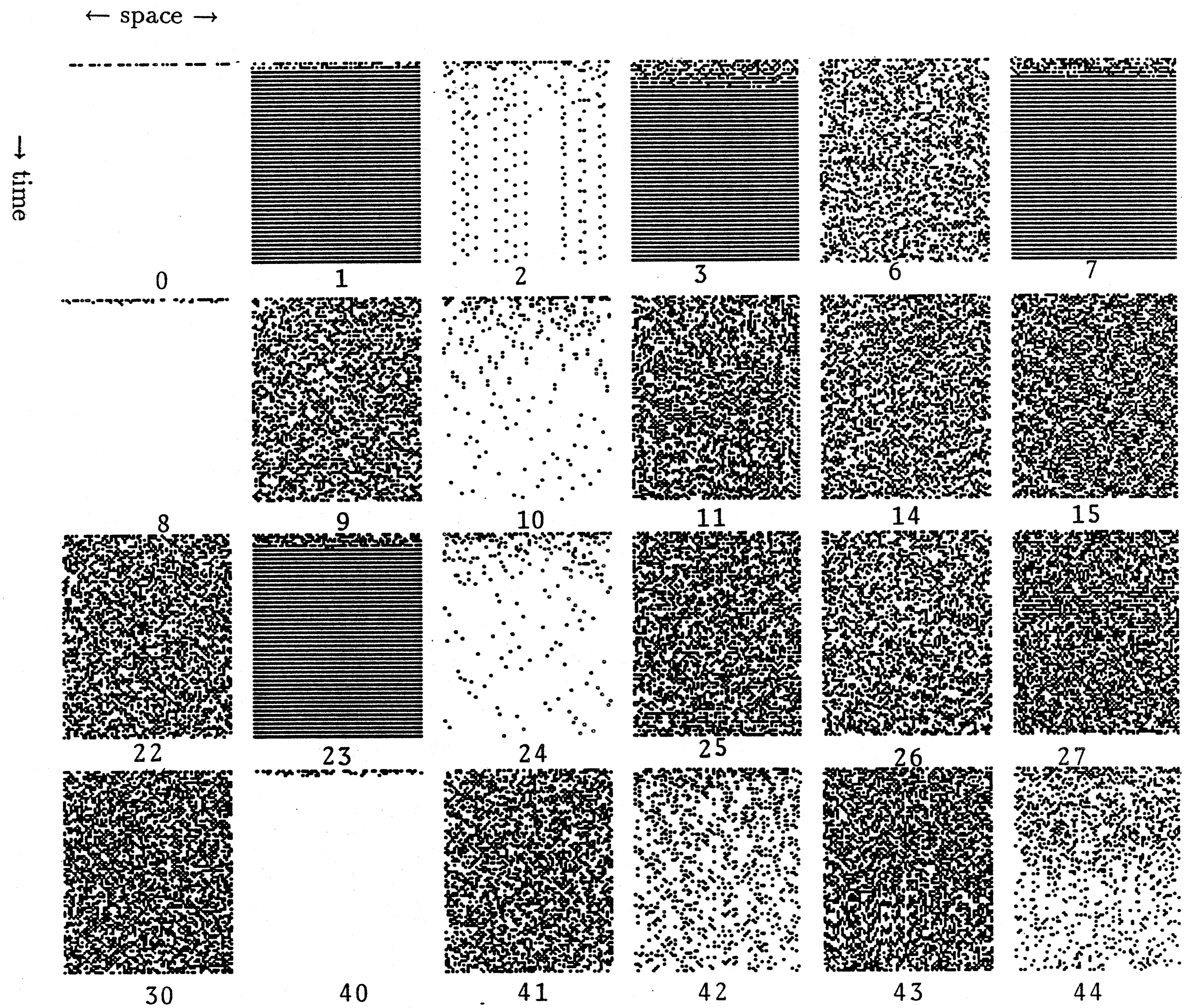


Figure 5: (to be continued)

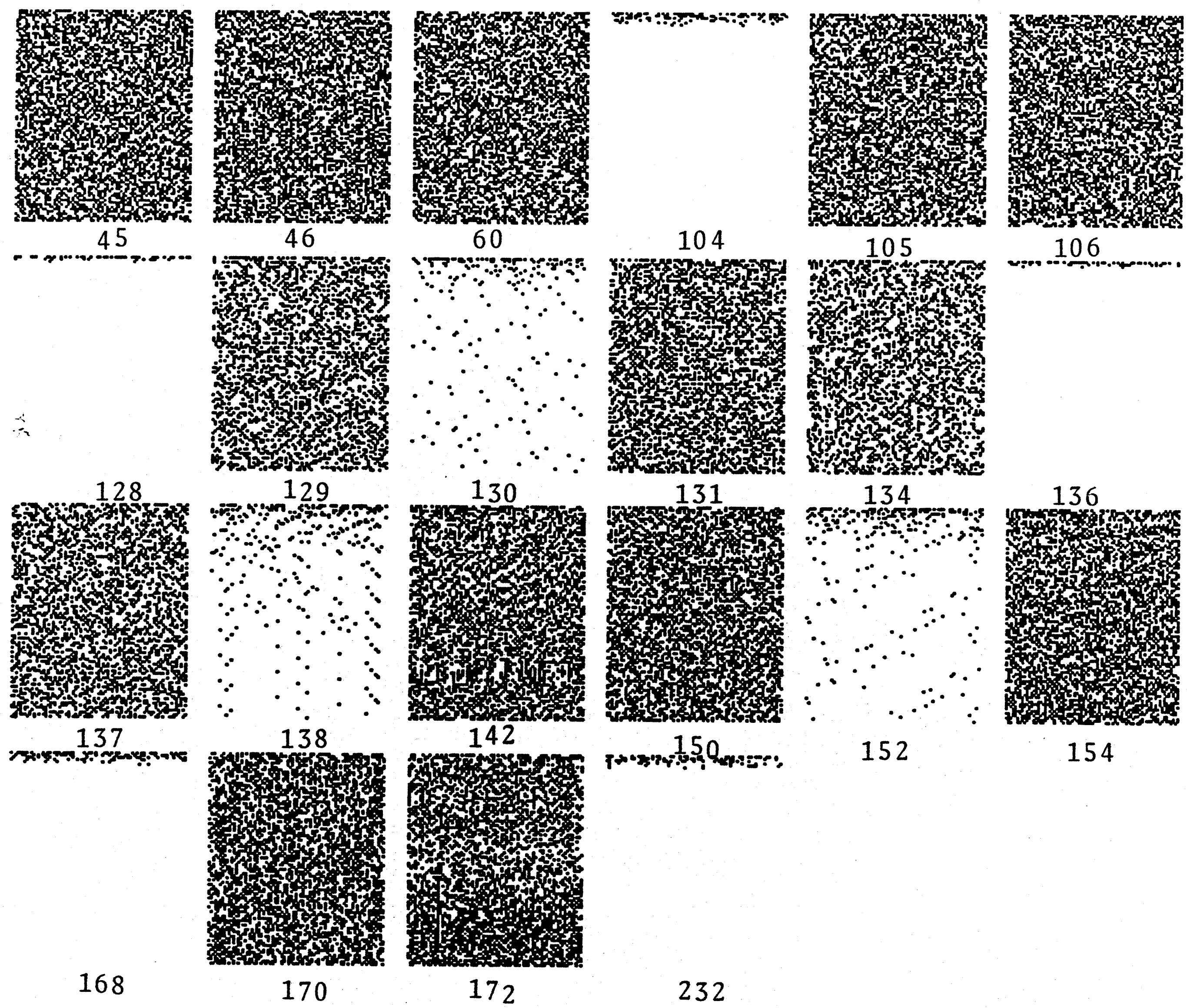


Figure 5: The spatio-temporal patterns of all 46 independent elementary cellular automata with distinct inputs fully non-local connection. The system size is 77 and the number of time steps is 94. It is typically easy to tell which rule is 2-cycle periodic, which is chaotic, etc, from these patterns except periodic rules with very long transients (e.g., rule 27 and rule 172) or very long cycle lengths (e.g., rule 15).

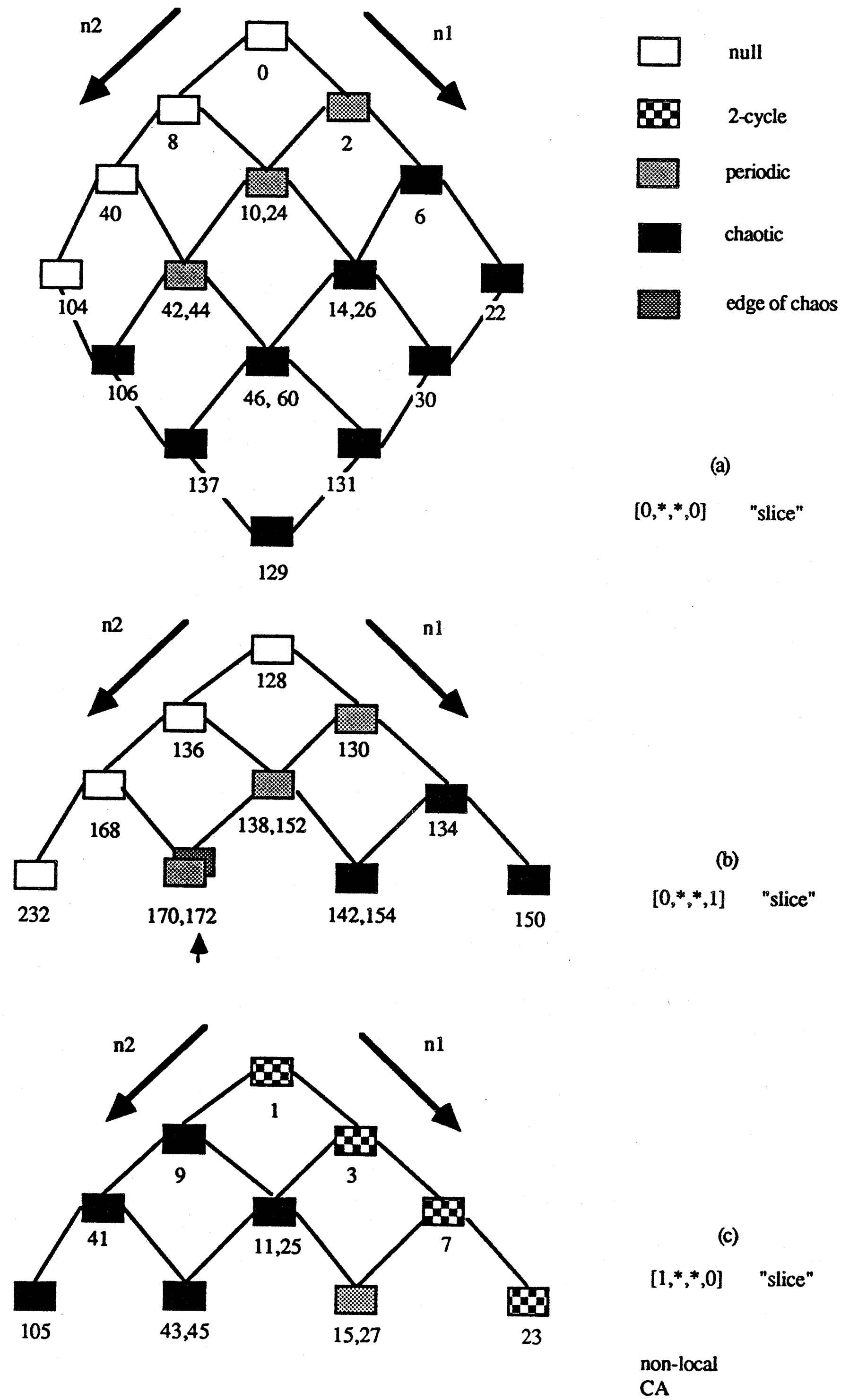


Figure 6: The rule space for fully non-local cellular automata with mean-field parameterization. The rule with long transient (rule 172) is marked.

that cluster, as illustrated by the 10 clusters which contain two rules:

$\{n_0, n_1, n_2, n_3\}$	rules	behavior
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.

(3.2)

Note that although rule 170 and rule 172 are both classified as periodic, the rule 172 has much more interesting transient behaviors.

The "cleanness" of the non-local cellular automata rule space makes it possible for explaining the observed results by mean-field theory. In particular, we want to predict at what mean-field parameter values the transition to chaotic dynamics occurs. The next section will discuss this problem in detail.

4. Bifurcation-like phenomena in multi-parameter 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 [36]. This sudden change of the dynamical behavior with the smooth change of the parameter is studied in the bifurcation theory [12].

Higher dimensional dynamical systems typically need a larger number of parameters to describe the details of the interaction among components. Take elementary cellular automata for example, a rule is specified by eight bits, $a_0, a_1, \ldots, a_7$, and each of them can be considered as a parameter varying from 0 to 1. An incomplete description requires fewer numbers of parameter, for example, the mean-field parameterization has four parameters n_0, n_1, n_2, n_3 varying either from 0 to 1, or from 0 to 3.

The bifurcations or the transition phenomena in multi-parameter 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 n, there is a critical surface (or hyper-surface to emphasize the higher dimensionality) with at most a dimension $n-1$, which separates rules of regular and chaotic dynamics. Moving through the critical hyper-surface can experience a sudden change of the dynamical behavior with no other "in-between" dynamics, a case called the first order

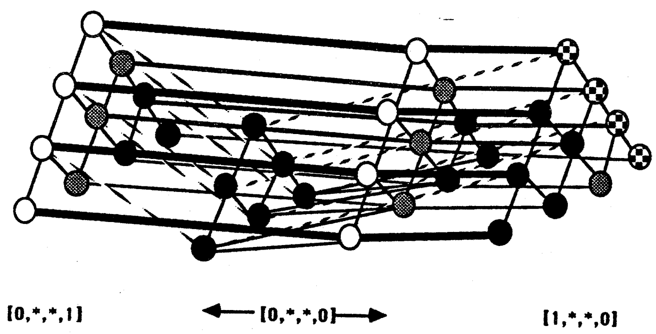


Figure 7: The complete "folded" rule space for fully non-local cellular automata. The λ parameter is increased from top to bottom. Each connection between two clusters is a change of 1 in one of the mean-field parameters, which either increases or decreases the λ value by 1/8.

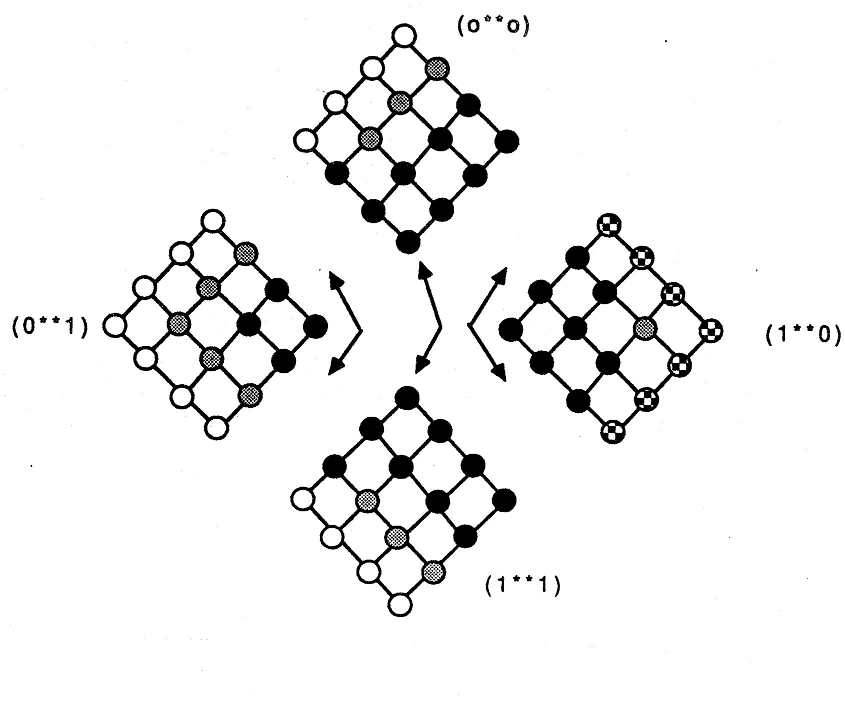


Figure 8: The unfolded rule space for fully non-local cellular automata. The folding, or the equivalent relation among clusters, is indicated by the two-way arrow. The connection between different slices of the rule space is not drawn, but one can refer to Fig.7.

phase transition using an analogy of the statistical physics; or it can experience a gradual change of the dynamical behavior, with some rules with "in-between" dynamics actually sitting on the critical hyper-surface, a case called the second order phase transition.

Rules with "in-between" dynamics (or complex, critical dynamics, etc.) do not cover the critical hyper-surface completely, leaving other regions of the critical hyper-surface as "holes". If one hits the hole while passes through the critical hyper-surface, no rules with complex dynamics will be encountered. Considering this fact, as well as that the dimension of the critical hyper-surface 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 critical hyper-surface, but also the regions of the critical hyper-surface with the critical rules.

As a footnote, I want to relate the general picture presented above of the critical hyper-surface in multi-parameter higher-dimensional dynamical systems with the discussions of the "self-organized criticality"[1]. 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 rule is sitting at a particular point on the critical hyper-surface 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 critical hyper-surface. It might be the case that some parameters are *irrelevant* parameters which only move the rule along the critical hyper-surface. It is thus important to identify the parameter which is *relevant* for making the transition happen.

In the previous studies of the cellular automata rule space, it is the λ parameter which is used as the "relevant" parameter [27, 30, 32]. One can consider the λ parameter as a measure of the activation of the entries in the rule table [26]. 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 the λ is close to 0.5, all three-input configurations are activated, and the dynamics is likely to be chaotic. By tuning λ from 0 to 0.5, it is expected that one will pass through the critical hyper-surface.

As a crude estimation, the critical hyper-surface has the λ value around $\lambda_c \approx 1/k$, where k is the number of inputs (e.g., $k = 3$ for elementary rules). This λ_c is the estimation of the onset of non-zero entropy values [64]. Another estimation of λ_c, which is the onset of non-zero expansion rate of the perturbation, gives $\lambda_c \approx 1/2 - 1/2\sqrt{1 - 4/(k+1)} \approx 1/(k+1)$ [32]. These estimations should become more accurate as the number of inputs goes to infinity. Numerically, the maximum spatial mutual information is used to locate the critical region using the fact that the correlation length become longer near the critical hyper-surface [32]. This estimation of λ_c is always larger than the two estimations mentioned above.

One conclusion from the numerical study of the critical hyper-surface in local cellular automata rule space is that the λ value at which a transition occurs from periodic to chaotic dynamics is not unique. It means that the critical hyper-surface is not a hyper-plane which the λ is perpendicular to. To characterize how the critical hyper-surface bends, one has to identify different regions of the critical surface, and in order to do so, introduce more parameters.

For elementary cellular automata with fully non-local connections, mean field parameters can characterize the critical hyper-surface almost completely. From Fig.6, one can see that if $n_0 = n_3 = 0$, $1 < (n_1)_c < 2$; if $n_0 = 0$ and $n_3 = 1$, $2 < (n_1)_c < 3$; and if $n_0 = 1$ and $n_3 = 0$, $0 < (n_2)_c < 1$. This characterization is better than the λ parameter. For example, in the case of $n_0 = n_3 = 0$, if one chooses $3 < \lambda_c < 4$, four chaotic clusters (which contain rule 6,14,26, and 22) will be counted as being below the critical line (remember that λ is pointing down), whereas $1 < (n_1)_c < 2$ will miss only one chaotic cluster (which contain rule 106).

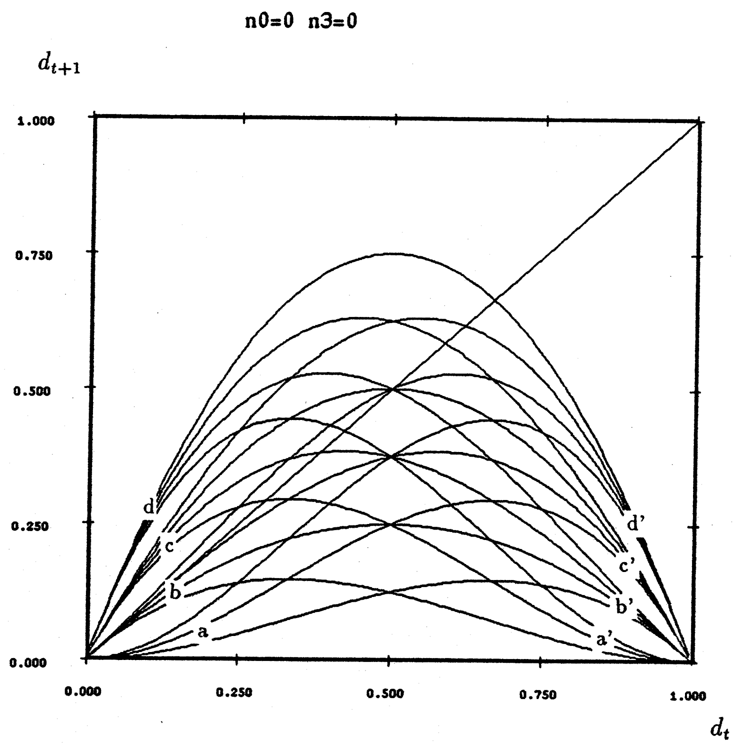
As a final note, the above discussion about the critical surface applies to the "folded" rule space, which contains only the independent rules [31]. The folded rule space with the mean-field parameterization for the fully non-local elementary cellular automata is shown in Fig.7. Part of slice with the nonlinear clusters $\{0, *, *, 0\}$ is raised to the top so that the λ is always increased from top to bottom. Also shown is the connection between nonlinear clusters with the linear and inversely-linear cluster. Fig.8 shows the original rule space with the possibly degenerate clusters, and the equivalence relations (or "foldings") are indicated by the two-way arrows. In the next section, we will try to determine the critical hyper-surface by the mean field theory.

4.2 The return map of density in mean field theory: and the determination of the critical surface

The approach adopted in this subsection is to neglect the details in the dynamics and only examine how macroscopic quantities such as the density change with time in the mean-field theory (for a review, see Ref.[14], note that the term "first-order Markov approximation" is the same with mean-field theory. Other relevant studies are included in Refs.[44, 59, 13, 64]). The mapping for the density:

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

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 non-zero fixed-point solutions. These assumptions can not be always true, for example, some periodic rules oscillate only the local densities instead of the global density, and some fixed-point rules can have large values of the density.



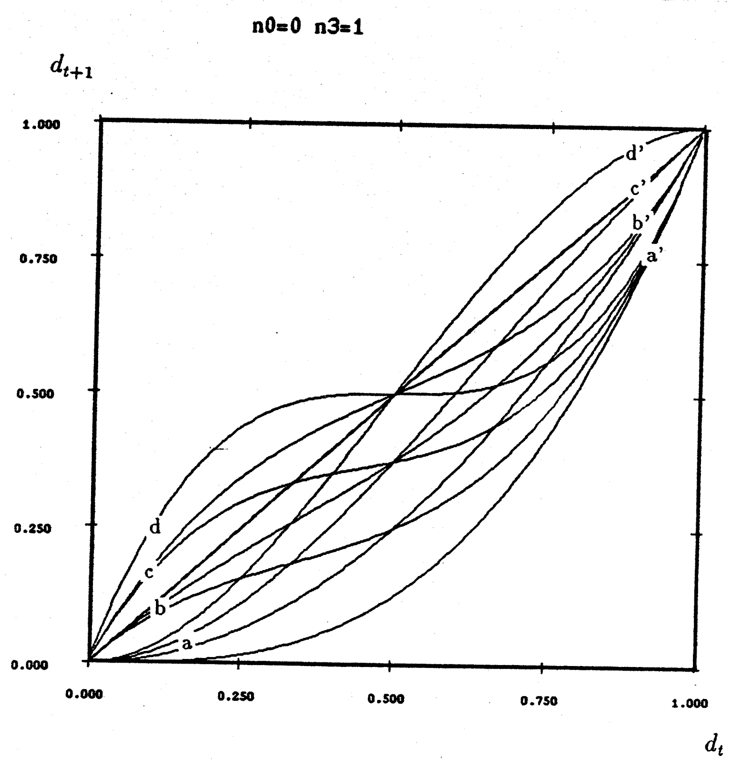


Figure 9: (to be continued)

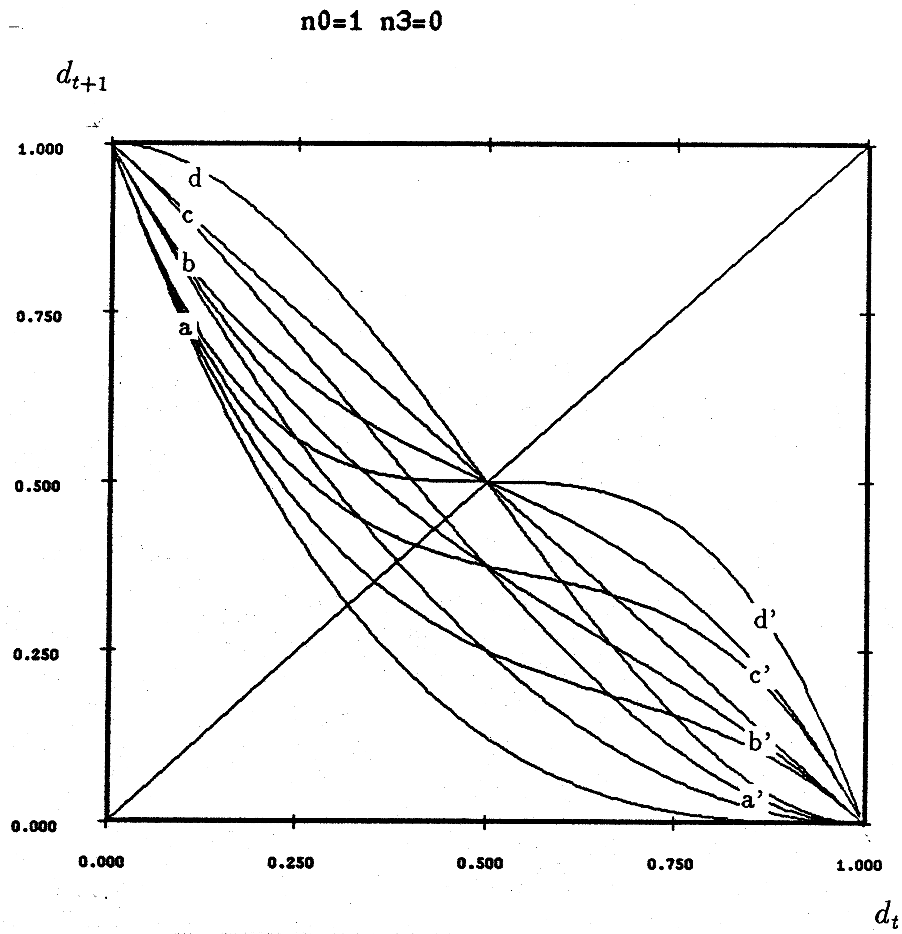


Figure 9: The return maps (the density at time $t+1$ versus the density at time t) by mean field theory. Those with the same n_1 values are labeled by letters a ($n_1 = 0$), b ($n_1 = 1$), c ($n_1 = 2$) and d ($n_1 = 3$); and those with the same n_2 values are labeled by letters a' ($n_2 = 0$), b'($n_2 = 1$), 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 ($n_0 = 1$, $n_1 = 0$).

In mean field theory as applied to the 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 the return map changes with the mean-field parameters, we can hopefully understand why and when the transition occurs. As before, we 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.9(a) 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 $(d_t, d_{t+1}) = (1,0)$ 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 intersection 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 non-zero values with 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 [36] if $n_1 = n_2$. Especially, the return map at $n_1 = n_2 = 3$ gives the logistic map at the first period-doubling bifurcation point. In 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 the current 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 Eq.(4.3) to become chaotic. There are similar discussions on the difficulties for having macroscopic chaotic fluctuations in Ref.[3].

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

When $n_1 = 0$, no return maps intersect with the diagonal line at non-zero values, and null rules are expected. When $n_1 = 1$ and $n_2 \neq 2$, there is no non-zero 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. Only when $n_1 > 1$, there are non-zero 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.9(c) 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 non-zero value, so no rule is expected to exhibit fixed-point 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, especially 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 increase the n_2 parameter.

In conclusion of this subsection, 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 non-local cellular automata, when the statistical averaging in the mean field theory is closer to reality. If the same mean field theory is applied to the local cellular automata, there will be many exceptions (compare Fig.2 with Fig.6). The topic of how to improve the mean-field theory so that one can predict correctly the dynamical behavior of local cellular automata is currently being discussed [15]. The observation that either n_1 or n_2 can be the relevant parameter which is perpendicular to the critical surface is equivalent to say that the critical hyper-surface is not a hyper-plane. Generally speaking, a critical dynamics results from a balance among many competing factors in the multi-parameter dynamical system. If we move from one part of the rule space to another, the "environment" in which the previously relevant parameter should vary in order to achieve a balance is different. And that relevant parameter may no longer perpendicular to the critical surface in the new "environment". It will be intersting to determine the critical hyper-surface for larger cellular automata rule space as well as the rule space for other multi-parameter, many degrees of freedom dynamical systems.

5. Further comments on the relationship between complex dynamics and the universal computation

5.1 Robust universal computation

The early observation that the game of Life, a two-dimensional, two-state nine-input local cellular automaton rule, has both the ability of doing universal computation and the complex dynamics [2] inspires others to speculate that local cellular automata rules with complicated glider activities (i.e., rules with complex dynamics) are capable of universal computation [60, 62].

This conjectured connection can be understood as follows: because systems capable of doing universal computation should be able to carry out computations of arbitrary difficulties, the computing time should be arbitrarily long. One can consider the transient time of a dynamical system as also being the computing time, whereas the onset of the limiting attractor is the end of the computation. The result of the computation should be extractable from the limiting attractor. For cellular automata rules whose transient times are short and independent of the system size, they cannot carry out universal computation because they cannot do difficulty computations. This fact naturally leads to complex rules, whose transient times are almost always very long, as the best candidate for universal computers.

One question remains is what about the cellular automata rules whose transients are typically short *but* can be arbitrarily long with specially designed initial configurations? Are they also universal computers even though they are not classified as complex rules? It seems that the answer is yes, due to the recent numerical observation that a class of cellular automata known to be capable of universal computation can nevertheless be periodic rules or chaotic rules [34]. This observation only illustrates the fact that being universal computer is a property related to the dynamics starting from all possible initial configurations, whereas being a complex rule is a feature related to the dynamics starting from random initial configurations.

Similar to the concepts of the attractor and the basin of attractor, we feel the need to introduce the name *robust universal computation*. Roughly speaking, if a cellular automaton can carry out universal computation with a set of initial configurations whose percentage in all possible initial configurations is not zero, then that cellular automaton is said to be able to do robust universal computation. The distinction between the robust universal computer and non-robust ones should be able to separate universal computers with periodic and chaotic dynamics from those with complex dynamics.

The game of Life should be one example of the robust universal computer. The computation in the game of Life is operated on the level of gliders and the interaction between gliders can simulate the basic logical gates such as AND, OR and NOT [2]. On the other hand, these gliders can easily emerge from a random initial configuration. Although one still has to design the initial configuration in such a way that two gliders aiming at each other with a certain angle, it seems that many configurations (analogous to the basin of the attractor) can lead to the same required glider collision.

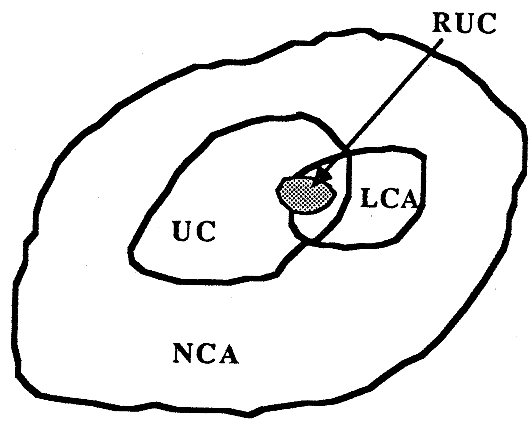


Figure 10: Schematic illustration for the relation among local cellular automata, non-local cellular automata, universal computers, and the robust universal computers.

In the next section, I will discuss the concept of robust universal computation in the context of non-local cellular automata.

5.2 Computation at the level of bits

There are two aspects of non-local cellular automata which are different from the local ones: (1) Because the concept of locality loses much of the sense, there will be no local configurations such as the gliders. The absence of the gliders, however, does not make it impossible for non-local cellular automata to carry out universal computation, because the computation in non-local cellular automata is operated on the level of bits instead of gliders.[2] (2) The wiring among components is chosen at time 0. In other words, the wiring is part of the initial condition instead of part of the dynamical rule.

These two aspects of the non-local cellular automata have important implications on the computational ability of the non-local rules. The first makes it easier for non-local cellular automata to be able to carry out universal computations. The second makes it more difficult for them to do robust universal computations. Let me be more specific in the following.

The difficulty for a local cellular automaton to be computationally universal is because the rule should be able to generate enough numbers of gliders whose interaction can simulate the basic logical gates. This requirement imposes a strong restriction on the rule table. Because these rules can maintain the local configurations and at the same time propagate them, they are able to store as well as to transmit the information, a desirable feature for the computation [27]. The possibility for having non-local connections relaxes much of the restrictions imposed on the rule table. The propagation of the

[2]I would like to thank C. Langton for emphasizing this point.

information is taken care of by the direct non-local connection between components, and the carrier of the information is simply the state value on each site, with no requirement for the rule to create any local configurations.

Since the ability for non-local cellular automata to do universal computation is largely due to the new freedom for designing any desirable connections, random choice of wirings easily destroys this freedom. If we revise our definition of the robust universal computers as those which require a non-zero measure of the initial configurations *and* the initial wirings to do the universal computation, most of the non-local cellular automata which are universal computers should not be robust universal computers. Fig.10 shows schematically the relationship among local cellular automata (LCA), non-local cellular automata (NCA), universal computers (UC) and robust universal computers (RUC). In particular, the picture is drawn in such a way that more non-local cellular automata are universal computers, but more local cellular automata are robust universal computers. It is hoped that a precise definition of the robust universal computation will be given in the future and one can actually determine the percentage of the universal computers which are robust.

In order to have a feeling of how easy for a non-local cellular automaton to do universal computation, and how difficult to do robust universal computation, I will carry out a simple exercise to determine all the elementary rules which have logical gates AND and NOT for some of their inputs (then these rules are close to being universal computers), and all elementary rules which have logical gates AND, NOT, and OR for some of their inputs (since they do not have to construct OR by the combination of AND and NOT, the computation requires less sophistication in the wiring, and in some sense, robust).

(1) AND and NOT:

Without losing generality, suppose the logical 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 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

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

with the wild card symbol * representing the "do not care" symbol.

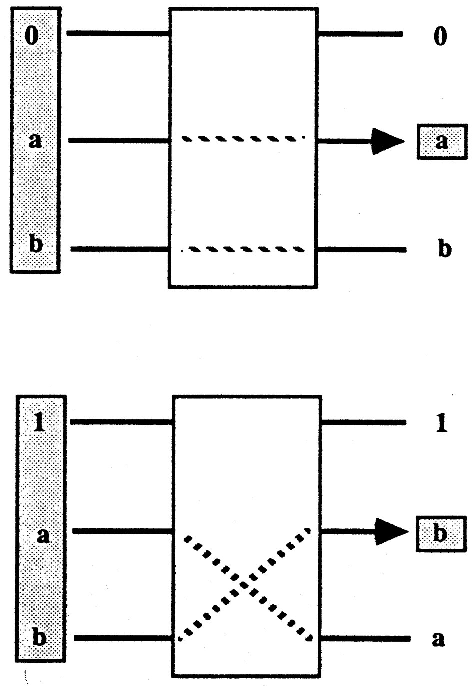


Figure 11: Fredkin's gate: the basic logical gate for conservative computation.

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, operated on the third input (while $x_{j_1}=x_{j_2}=0$, or $x_{j_1}=0$, $x_{j_2}=1$):

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

All the rules in Eq.(5.1) and Eq.(5.2) have AND and NOT in their rule table.

(2) AND, NOT and OR:

We first list all the elementary rules with AND and OR. Suppose the AND operates on the second and the third inputs while the first input is fixed at $x_{j_1} = 0$, and the OR operates on the first and the second inputs (while $x_{j_3} = 1$); or, the OR operates on the first and the third inputs (while $x_{j_2} = 1$); or, operates on the second and the third inputs (while $x_{j_1} = 1$).

The rules satisfying all these conditions include:

$$\begin{array}{lcccc} 000 & \rightarrow & 0 & 0 & 0 \\ 001 & \rightarrow & 0 & 0 & 0 \\ 010 & \rightarrow & 0 & 0 & 0 \\ 011 & \rightarrow & 1 & 1 & 1 \\ 100 & \rightarrow & * & * & 0 \\ 101 & \rightarrow & 1 & * & 1 \\ 110 & \rightarrow & * & 1 & 1 \\ 111 & \rightarrow & 1 & 1 & 1. \end{array} \tag{5.3}$$

Similarly, suppose the AND operates on the second and the third inputs while the first input is fixed at $x_{j_1} = 1$, then OR can only operate on the second and the third inputs without conflicting with the AND operation. This case includes only one rule:

$$\begin{array}{lcc} 000 & \rightarrow & 0 \\ 001 & \rightarrow & 1 \\ 010 & \rightarrow & 1 \\ 011 & \rightarrow & 1 \\ 100 & \rightarrow & 0 \\ 101 & \rightarrow & 0 \\ 110 & \rightarrow & 0 \\ 111 & \rightarrow & 1. \end{array} \tag{5.4}$$

The overlap between Eq.(5.1) and Eq.(5.3), or between Eq.(5.2) and Eq.(5.4) should give the rules with all three logical gates: AND, NOT, and OR (there is no overlap between Eq.(5.1) and Eq.(5.4), or between Eq.(5.3) and Eq.(5.2).) There are three rules resulted from the overlap: rule 184, rule 216 and rule 142. Rule 216 is equivalent to rule 172 by switching 0 and 1 and switching the first and the third inputs. Rule 172 in turn is equivalent to rule 184 by switching the first and the second inputs. Since rule 142 is already listed above, I will list the rule 184 below:

$$\begin{array}{lcc} 000 & \rightarrow & 0 \\ 001 & \rightarrow & 0 \\ 010 & \rightarrow & 0 \\ 011 & \rightarrow & 1 \\ 100 & \rightarrow & 1 \\ 101 & \rightarrow & 1 \\ 110 & \rightarrow & 0 \\ 111 & \rightarrow & 1 \end{array} \tag{5.5}$$

Actually, rule 184 is closely related to the basic logical gate, called *Fredkin's gate*, for conservative universal computers [10].[3] Fredkin's gate has three inputs and three outputs. Among the three inputs, one is called the

[3] I would like to thank S. Lloyd for pointing out this connection.

control input: whenever the control input takes the value 0, the other two input remain unchanged as the outputs; whenever the control input takes the value 1, the other two inputs switch (see Fig.11 for an illustration). Rule 184 can be represented in a similar form:

$$x_i^{t+1} = \begin{cases} x_{j_1}^t & \text{if } x_{j_2}^t{=}0 \\ x_{j_3}^t & \text{if } x_{j_2}^t{=}1, \end{cases} \tag{5.6}$$

that is, whenever the second input takes the value 0, the first input is chosen as the output; whenever the second input takes the value 1, the third input is chosen. In some sense, rule 184 is a *selector*.

6. Density fluctuation, coherent structures, long transients and the edge of chaos behavior for rule 184

The discussions in the previous sections on the structure of the rule space, the return map in mean-field theory, and the computational ability, all provide some information one way or another for locating rules with interesting behaviors. By searching the critical surface separating the periodic and the chaotic rules in the rule space, one might find rules with critical behaviors; by identifying rules whose return maps in the mean-field theory are marginally unstable, the corresponding rules might also exhibit "in-between" dynamics; and finally, rules contain a large number of logical gates in their rule table are more likely to carry out robust universal computation, as well as exhibit complex dynamics.

It is not clear that complex dynamics, or edge of chaos dynamics should exist at all in the non-local cellular automata. It is because all the modes of edge of chaos dynamics we know of in the local cellular automata will not survive the scramble of the wiring. The purpose of this section is to point out that complex dynamics does exist in non-local cellular automata. Instead of the case in local cellular automata whose edge of chaos dynamics is to create local configurations at small length scales and interactions among these local configurations at longer length scales, the edge of chaos dynamics in the non-local cellular automaton to be discussed generates cooperations among a cluster of components, which will be called the *coherent cluster*, or coherent structure. The counterpart of a large coherent cluster discussed above in local systems would be the existence of long range correlations.

This only edge of chaos elementary non-local cellular automaton is rule 184 (or rule 172), which has the diagonal line as its return map in mean-field theory (so does rule 170, but rule 170 does not exhibit interesting dynamics), and has AND, NOT, and OR logical gates in its rule table (so does rule 142, which is nevertheless not interesting). The rule 184 exhibits interesting dynamics only when all the inputs are distinct. If degenerate inputs are allowed, the dynamics is either periodic (for partially-local connections, see Fig.12(a)), or null/periodic (for fully non-local connections, see Fig.12(b)).

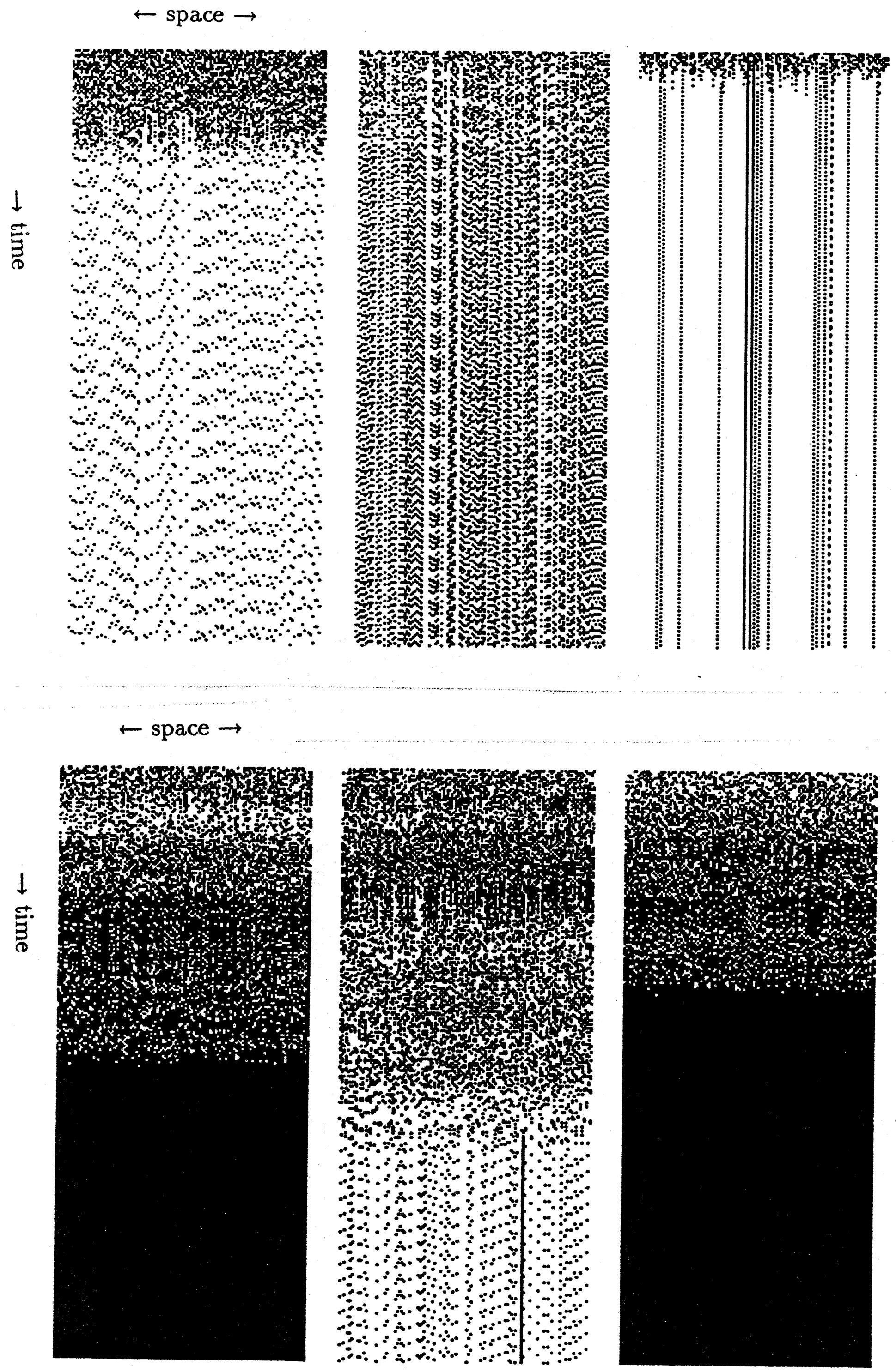
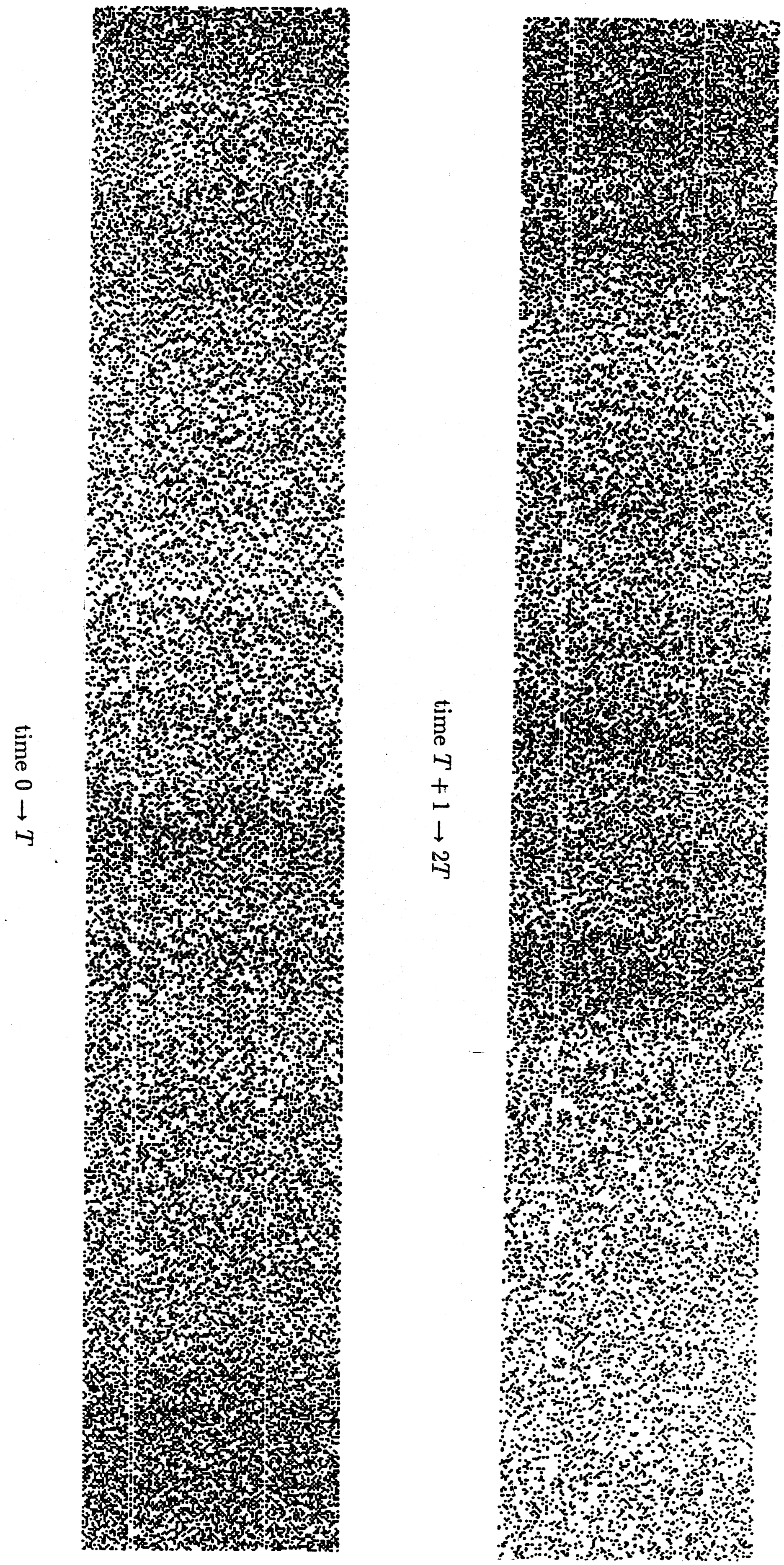


Figure 12: Illustration that the rule 184 exhibits periodic dynamics in (a) partially-local connection (with possibly the degenerate inputs); and null or periodic dynamics in (b) fully non-local connection (with possibly the degenerate inputs). The system size is 124, and the number of time steps is 294.



time 0 → T
time T + 1 → 2T

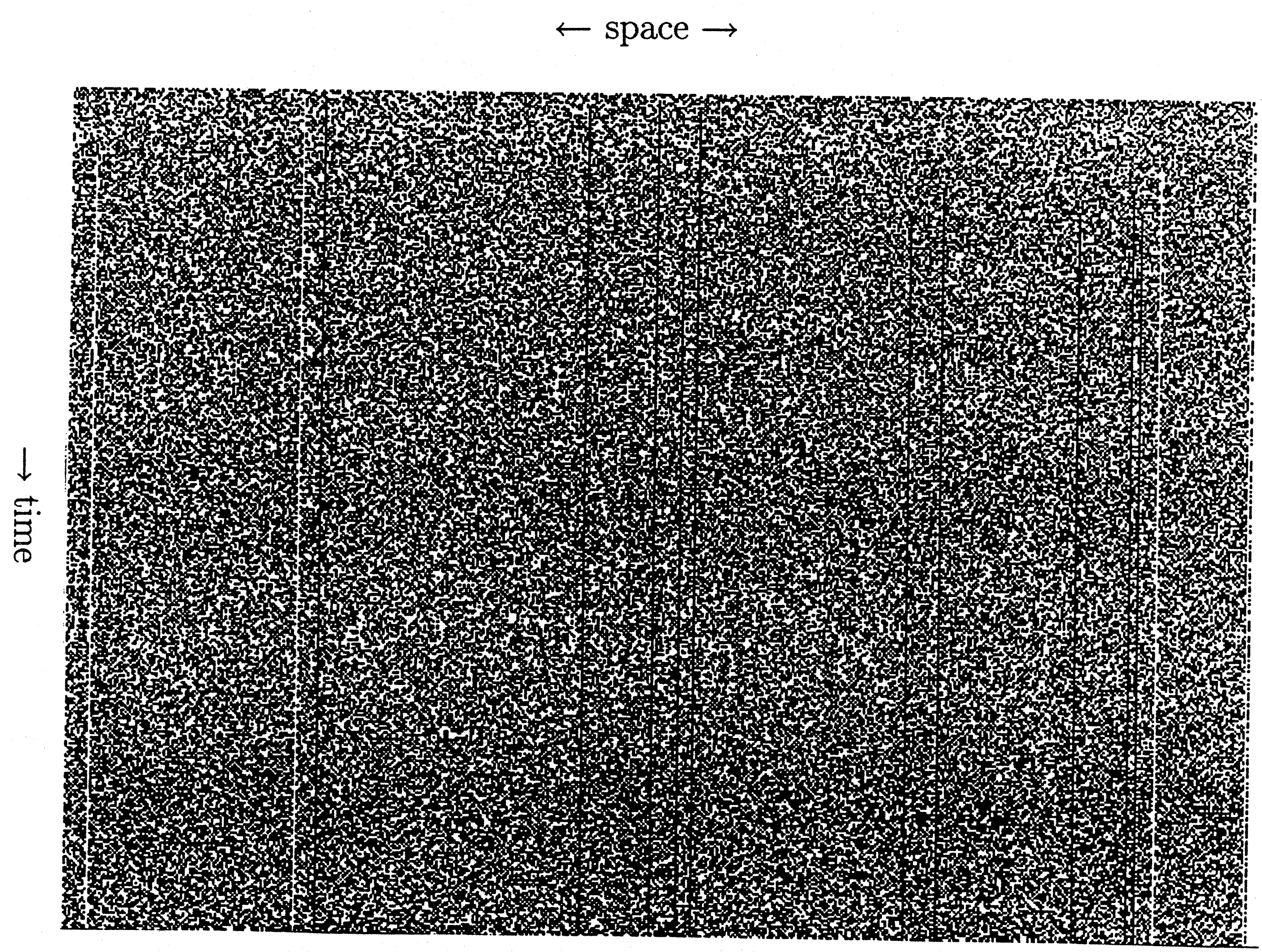


Figure 13: Spatial-temporal patterns for rule 184 with partially-local, distinct inputs connections:
(a) The system size is 129, and the number of time steps is 1572;
(b) The system size is 549, and the number of time steps is 393.

The sensitivity of the dynamics with respect to the type of the wiring is a further evidence that the rule is on the edge of chaos!

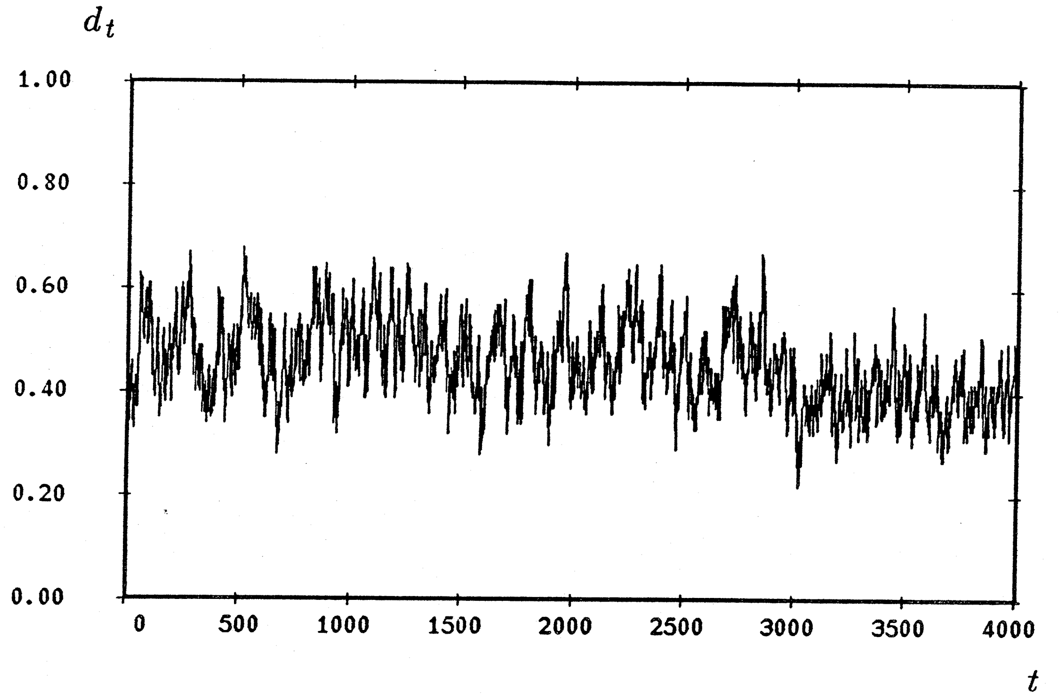
In the following two subsections, I will discuss the dynamical behavior of rule 184 for partially-local connection with distinct inputs and for fully non-local connection with distinct inputs, respectively.

6.1 Irregular fluctuation of the density

Fig.13(a) shows the spatio-temporal pattern for rule 184 of partially-local connection with distinct inputs. One feature of the pattern is that there exist horizontal stripes with either lighter or darker textures, indicating that the density fluctuates. The magnitude of the density fluctuation becomes smaller as the system size is increased, as shown by another spatio-temporal pattern for a larger system size (Fig.13(b)). Fig.14 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 further evidence that the density fluctuation does indeed become smaller in 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.15. Just like what is predicted by the mean-field theory, the return map is almost the diagonal line. If the mean-field theory is exactly correct, there should be no fluctuation, 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 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 the non-linear map 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 [45]. As discussed in the subsection 4.2, even when the return map for the density in mean-field theory can be of a non-linear form, it does not have chaotic attractors. Then, if the data is taken from simulations or experiments, there is only a dot on the plot corresponding to the fixed point solution.

Except for smaller system sizes (e.g., $N = 70$), when the dynamics can be locked into periodic oscillations, the random fluctuation as illustrated in Fig.14 seems to go on forever. In other words, it is not a transient phenomenon. Although it is relatively easier for open systems to have fluctuation in macroscopic quantities, due to the fluctuation of the environment, it is nevertheless rare for a closed deterministic system to do so. In order for some macroscopic quantities to fluctuate, 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, only when other components, including those which 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).



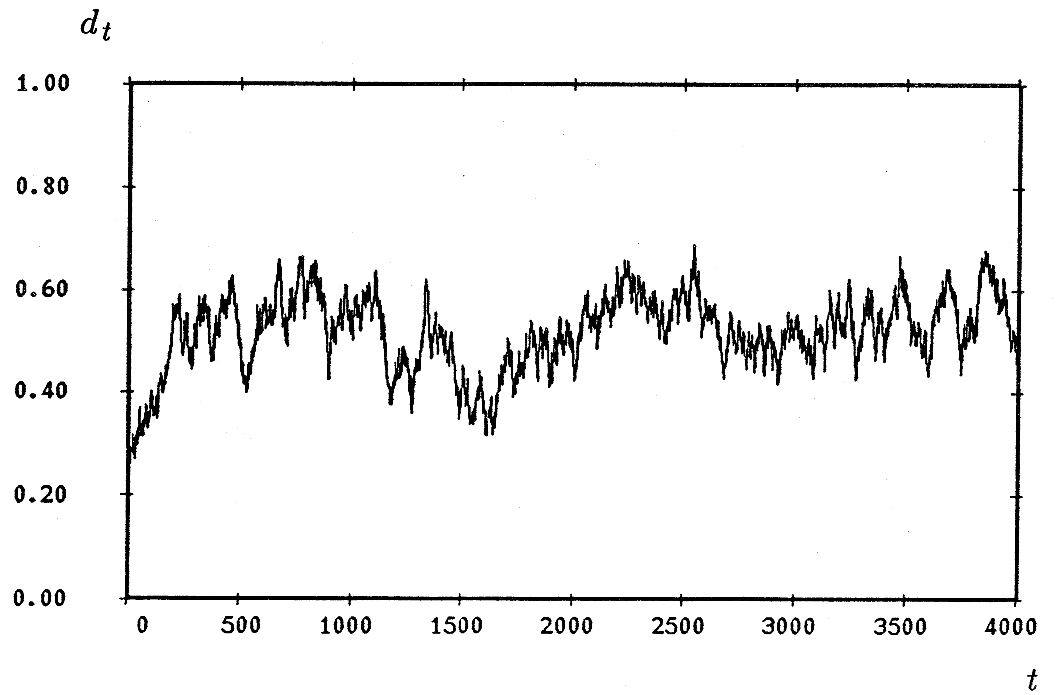
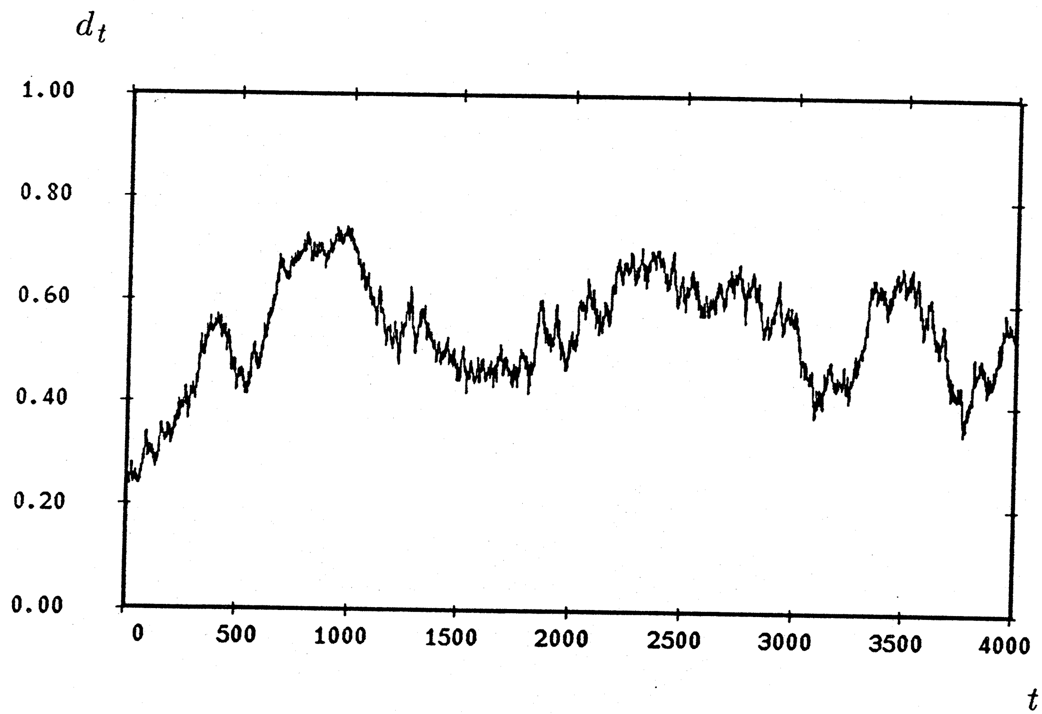


Figure 14: (to be continued)



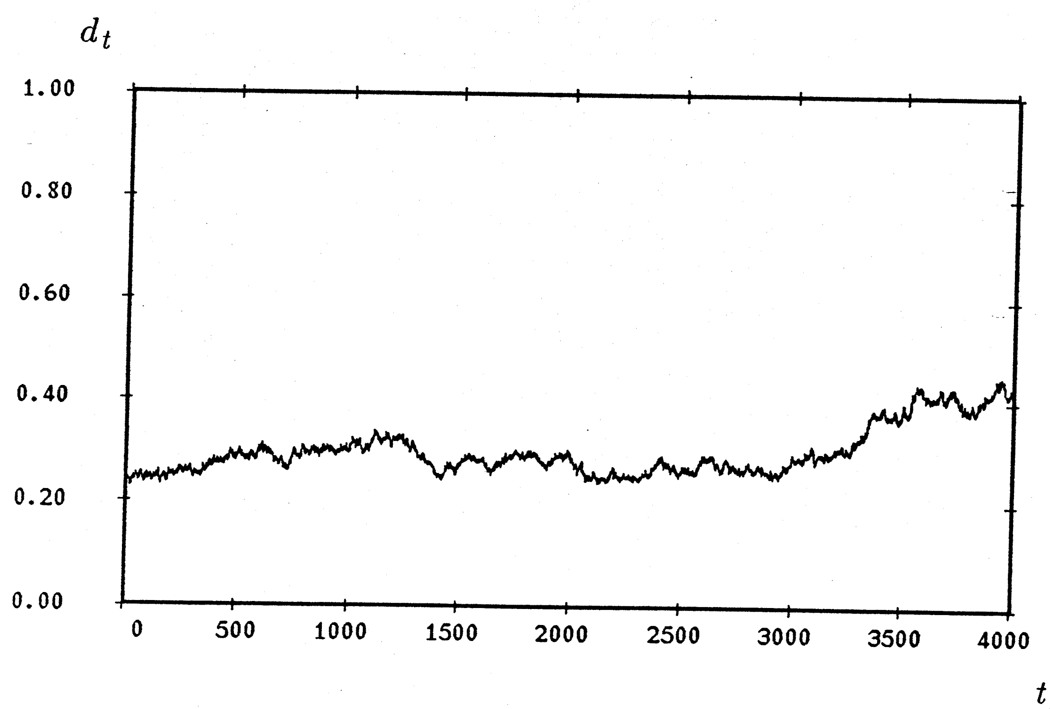


Figure 14: The density fluctuation of rule 184 with partially-local, distinct inputs connections, for system size equal to
(a) 100; (b) 501; (c) 1021; and (d) 5022.
It is clear that the magnitude of the fluctuation becomes smaller for larger systems.

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 than linearly with the system size. As a result, the number of coherent clusters is smaller when the system becomes larger, and the averaging over more numbers 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, such as the coupled map lattices [3, 16]. It is argued in these references 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 Lyapunov exponent [23, 4]. Other studies show that the correlation length is inversely proportional to the Lyapunov exponent [3, 16, 42], 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.[4]

No matter which result is correct, they all agree that the correlation length can be infinite or comparable with the system size only when the Lyapunov exponent is zero or small, i.e., the system is on the edge of chaos. The existence of the large coherent clusters in the rule 184 then provides another indication that the rule is a complex, edge of chaos rule.

6.2 Long transients before settling down in one of the limiting attractors

Fig.16(a) shows the spatio-temporal pattern for the rule 184 of fully non-local connections with distinct inputs. Again, one can notice the density fluctuation, which is even more prominent than that in partially-local, distinct-input connections. Similar to the partially-local distinct-input case, the magnitude of the fluctuation becomes smaller for larger system sizes, as evidenced by the spatio-temporal pattern in Fig.16(b) as well as the density as the function of time plots for two different systems sizes (Fig.17). There is, however, a major difference between the two: the density fluctuation with the fully non-local distinct-input connection is a transient behavior, wheras that with the partially-local distinct-input connection is not.

After the transient dies out, the system settles in either one of the two types of the attractors. Both types of the attractors are periodic, but one has density very close to 0 and another very close to 1. Sometimes, after waiting for a long period of time, one can be sure which attractor the system will settle into, because the trend towards one of the attractor is obvious. On other occasions, the picture is not that clear. The density can fluctuate

[4]In cellular automata, since the Lyapunov exponent cannot be defined with two discrete state values, it is the expansion rate of perturbation that is used to measure the degree of chaos [38, 32, 63]. For another discussion on the expansion rate of perturbation in the context of coupled map lattices, see Ref.[19].

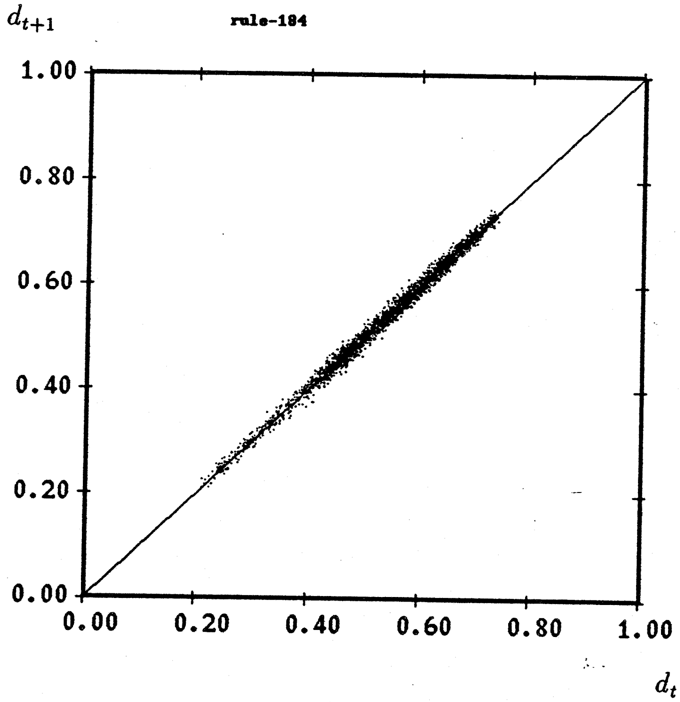


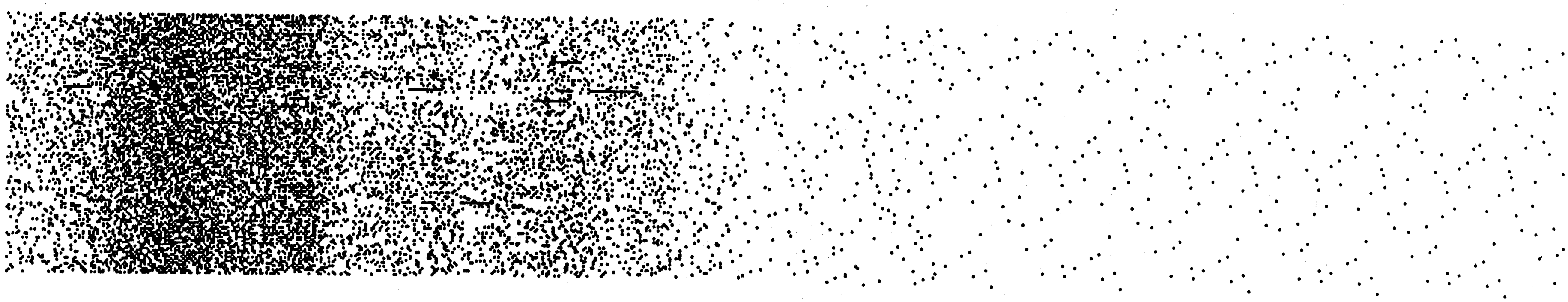
Figure 15: Numerically determined return map (d_{t+1} versus d_t) for rule 184 with partially-local distinct inputs connection, at system size 1021.

towards one of the attractor at the beginning, but for some reason, it can swing towards another attractor. Right now, it is poorly understood what has happened.

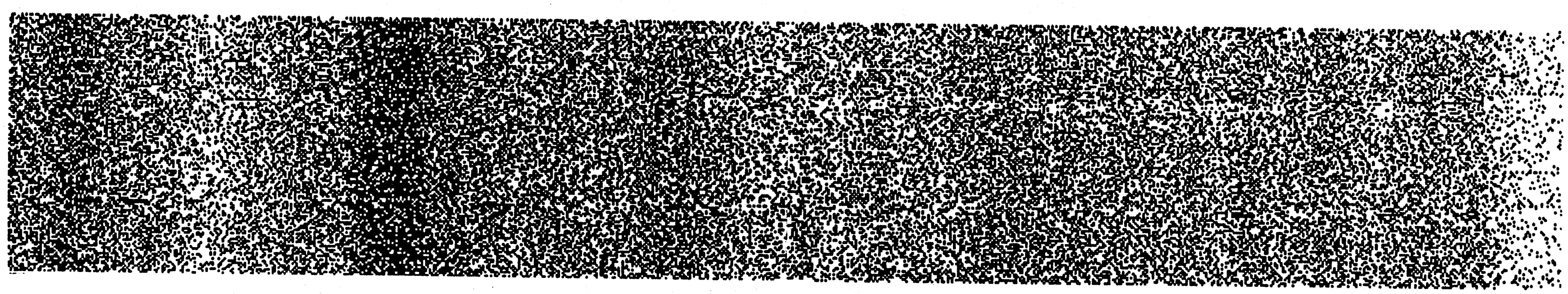
Crutchfield and Kaneko distinguish two classes of the transient behaviors in many degrees of freedom dynamical systems: the first one is monotonic and there is a convergence towards the limiting attractor; the second one is quasi-stationary which maintains an attractor-like dynamics for a very long time before falling to the real attractors [6]. The transient behavior for rule 184 in fully non-local distinct inputs connections 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 takes a long transient for the system to finally "find out". More studies on the mechanism for this transient are needed, especially the phase space structures [8].

There is another way to classify different classes of the transient behavior: to see how the transient time increases with the system size. Although it is shown that some systems with monotonic transient have power law or exponential divergence of the transient time with the system size, whereas those with quasi-stationary transient have super-exponential divergence [6, 22], it is not clear whether the same conclusion holds for other systems.

I calculate the transient time (T) as the function for the system size for the rule 184 with fully non-local distinct inputs connections. A simple



time $T+1 \rightarrow 2T$



time $0 \rightarrow T$

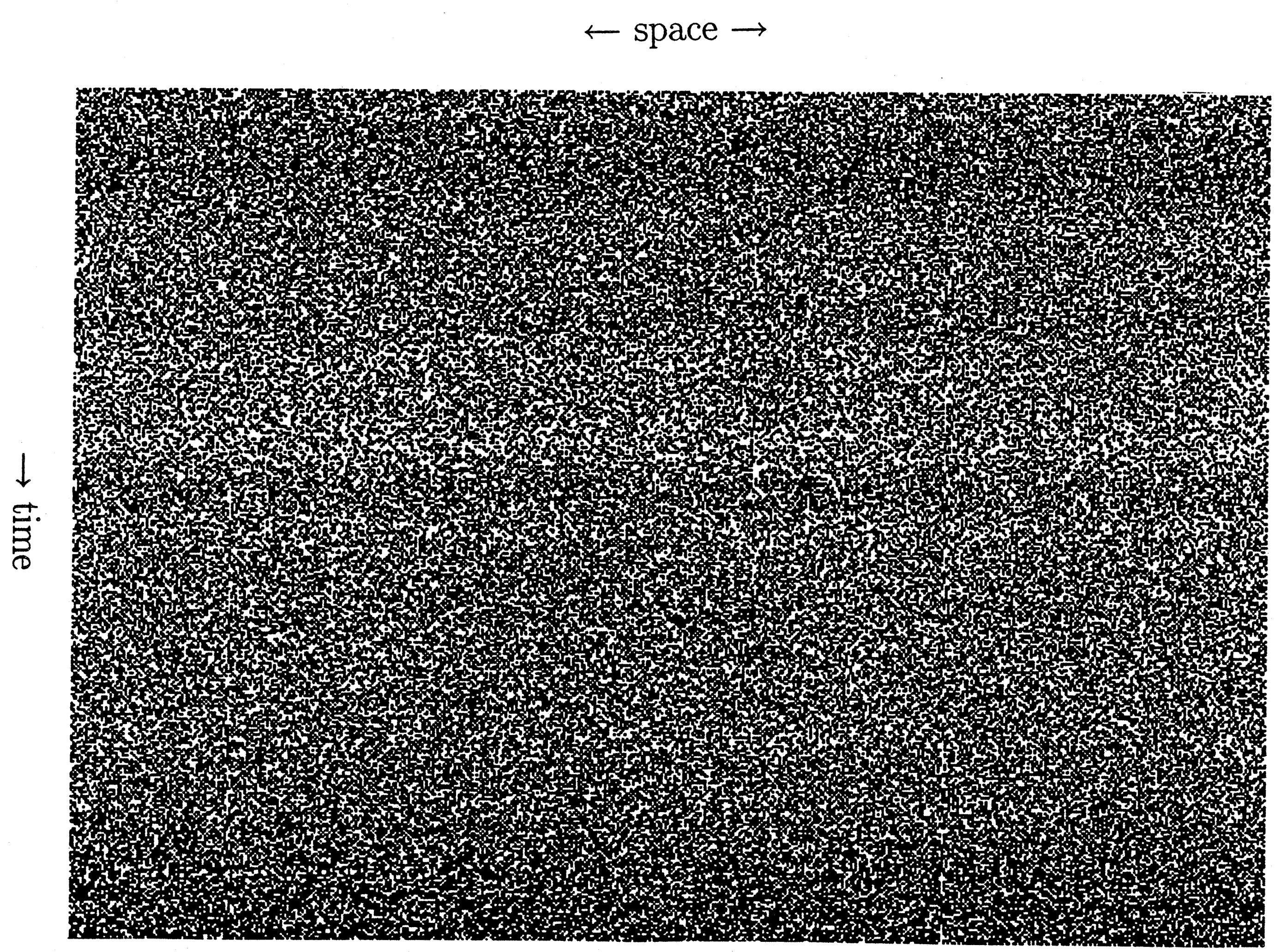


Figure 16: Spatial-temporal patterns for rule 184 with fully non-local, distinct inputs connections:
(a) The system size is 129, and the number of time steps is 1572;
(b) The system size is 549, and the number of time steps is 393.

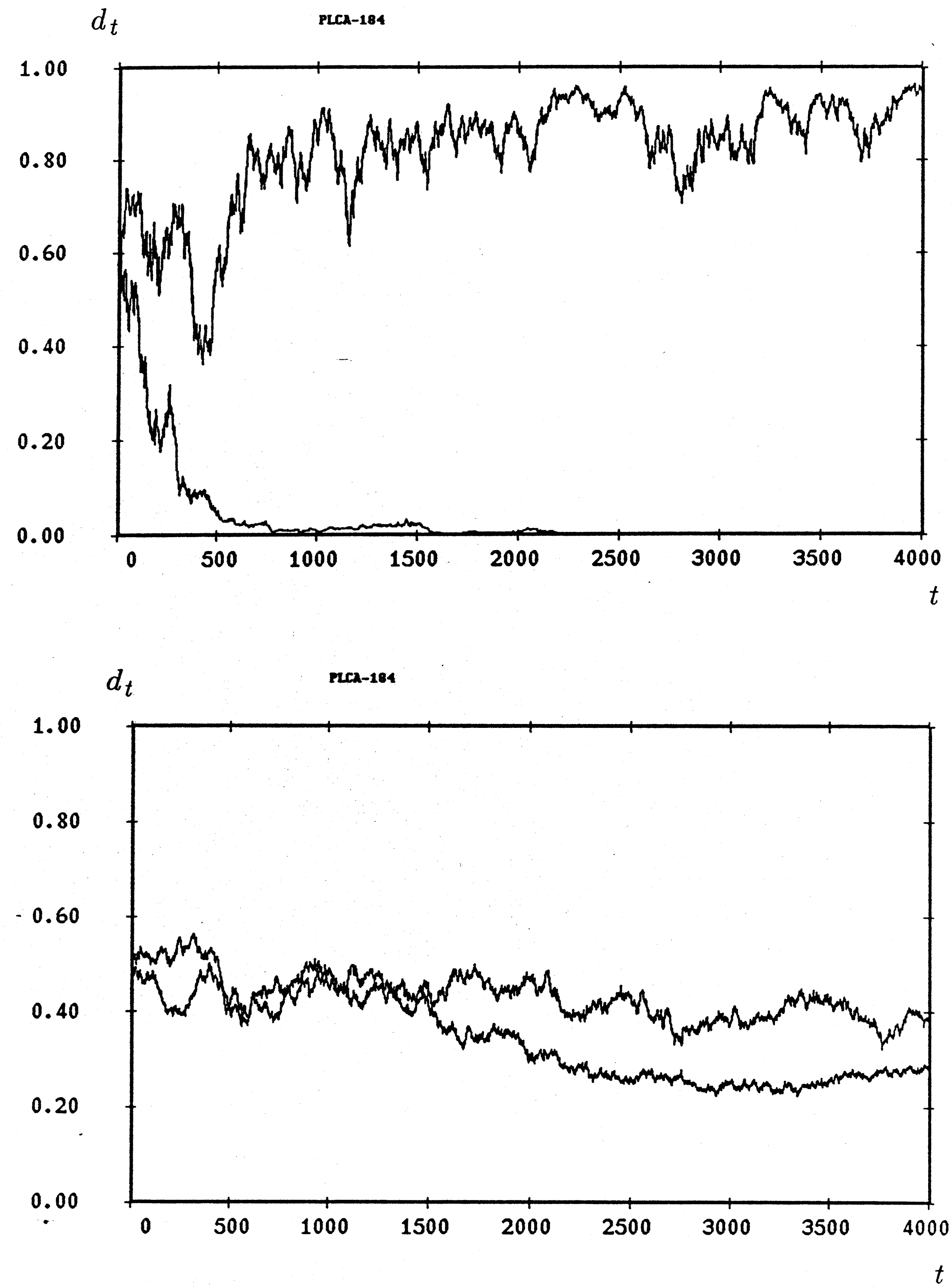


Figure 17: The density fluctuation of rule 184 with fully non-local, distinct inputs connections, for system size equal to (two samples for each case)
(a) 500; and (b) 5000.

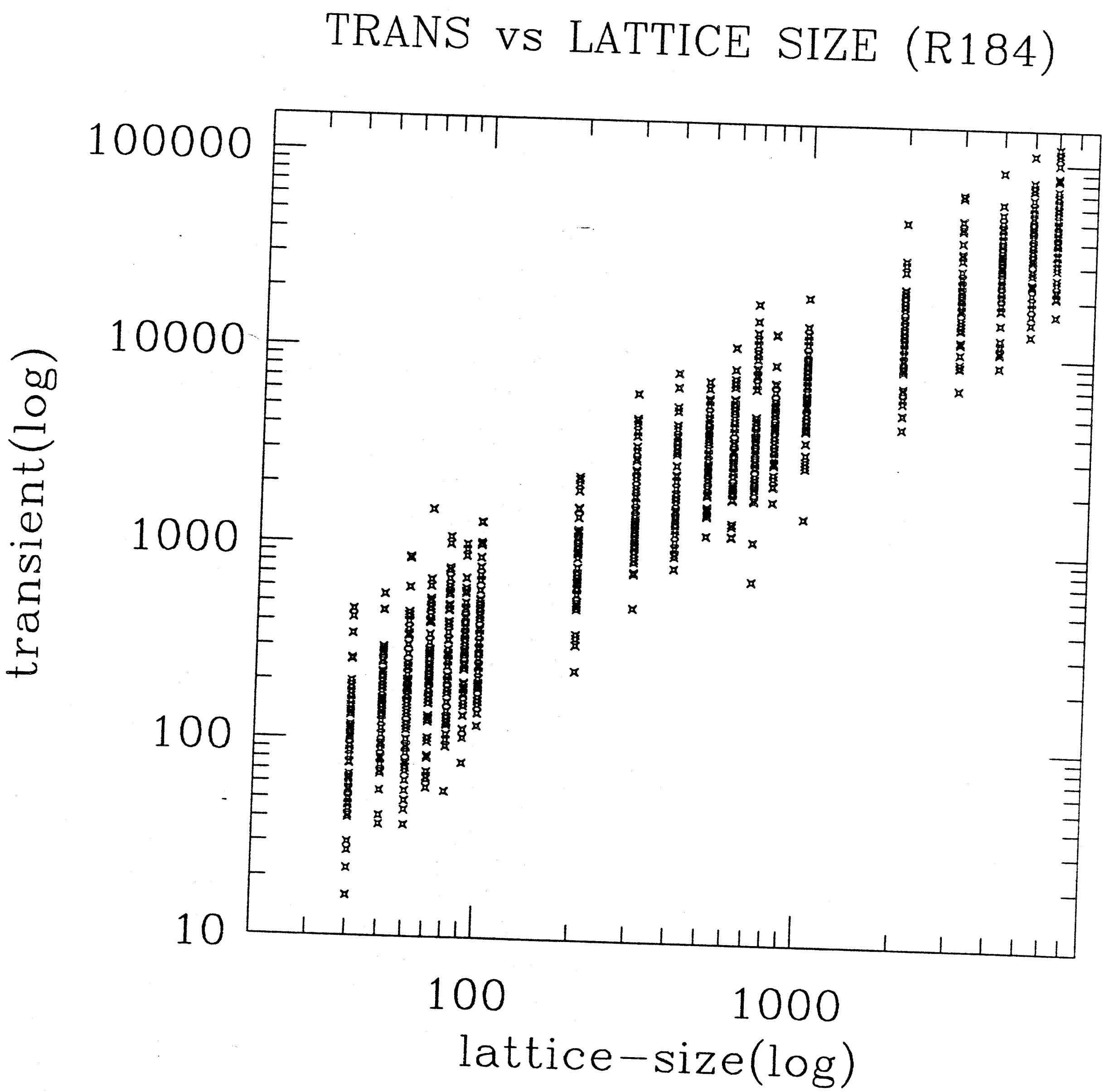


Figure 18: (to be continued)

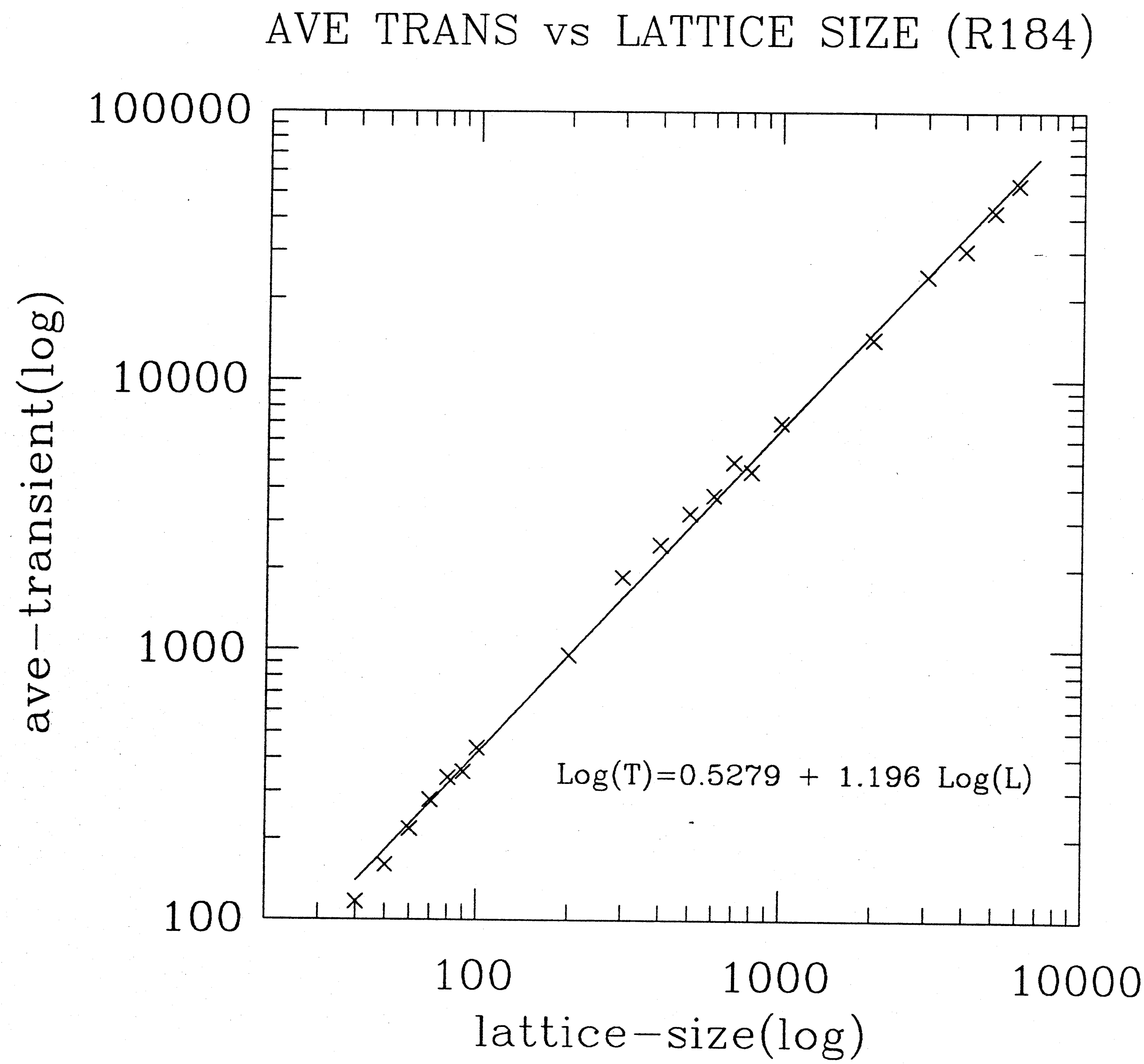


Figure 18: Transient time as the function of the system size (in log-log scale) for rule 184 in fully non-local, distinct inputs connections.
(a) All data points (1000 points). The fitting line gives $b = 0.229 \pm 0.071$, and $\alpha = 1.219 \pm 0.011$;
(b) The average transient times (20 points). The fitting line gives $b = 0.528 \pm 0.087$, and $\alpha = 1.196 \pm 0.014$.

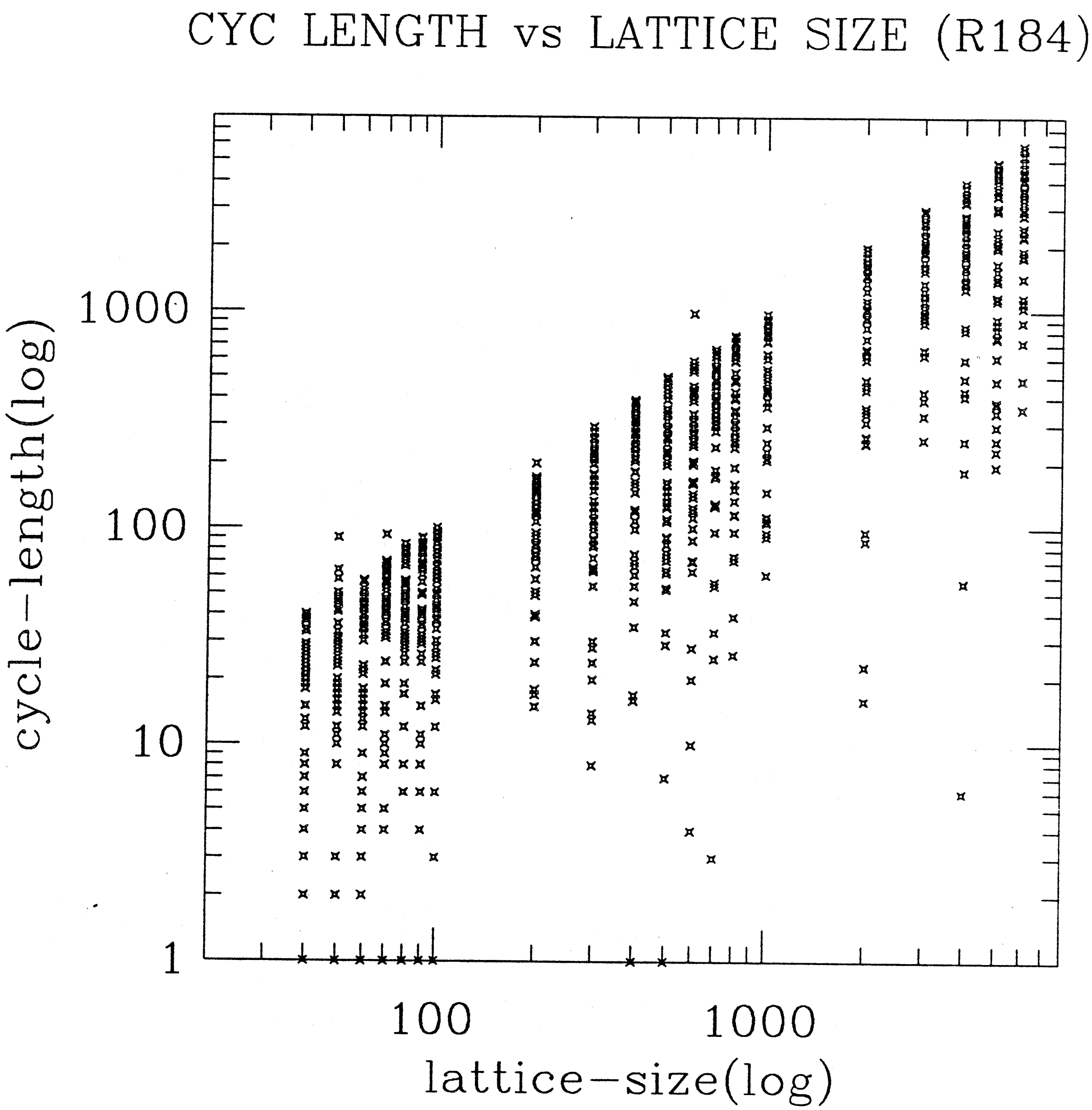


Figure 19: (to be continued)

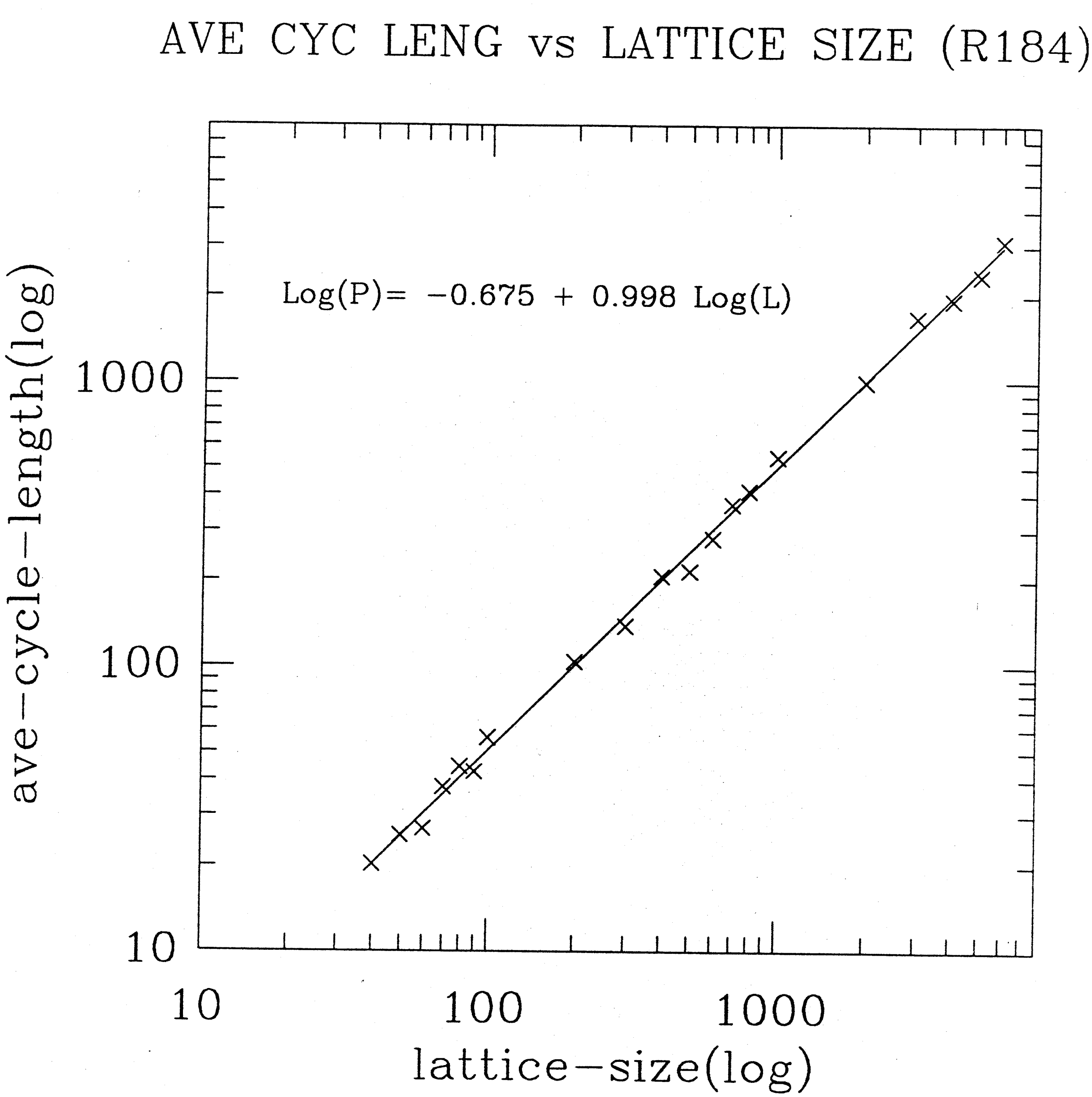


Figure 19: Cycle length as the function of the system size (in log-log scale) for rule 184 in fully non-local, distinct inputs connections.
(a) All data points (total 1000 points). The fitting line gives $b = -1.026 \pm 0.117$ and $\alpha = 1.009 \pm 0.019$;
(b) The average cycle lengths (20 points). The fitting line gives $b = -0.675 \pm 0.068$ and $\alpha = 0.998 \pm 0.011$.

algorithm is used [7], and the result is shown in Fig.18(a). For each 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.[41]):

$$\log_{10}(T) = b + \alpha \log_{10}(N) \tag{6.1}$$

gives $b = 0.229 \pm 0.071$ and $\alpha = 1.219 \pm 0.011$. Fig.18(b) shows the fitting for the average transient times as the function of the system size, which gives $b = 0.528 \pm 0.087$ and $\alpha = 1.196 \pm 0.014$, or,

$$T_{ave} \approx 3.4 N^{1.2}. \tag{6.2}$$

The power law divergence of the transient time with the system size is quite close to that in elementary local cellular automaton rule 110, which has $T \sim N^{\alpha}$ with $\alpha \sim 1.1 - 1.2$ [33].

Fig.19(a) shows the cycle length as a function of the system size. The fitting line for all 1000 data points gives $\alpha = 1.009 \pm 0.019$. Fig.19(b) shows the average cycle lengths as the 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 cycle length scaling, the transient time scaling seem to be non-trivial. It can be argued that for a class of marginal stable systems with the density fluctuating purely by finite size effect, the transient time is proportional to the system size [47]. The fact that the numerically determined exponent is strict larger than 1 indicates that there are more subtle cooperations among components in the system than a simple statistical model can account. More discussions on rule 184 will appear in the forthcoming publications.

Acknowledgements

I would like to thank Jim Crutchfield, Kuni Kaneko, Stuart Kauffman, Chris Langton, Seth Lloyd, Crayton Walker for discussions, valuable suggestions and recommending references, and James Theiler for a critical reading of the draft. 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).

References

[1] P. Bak, C. Tang and K. Wiesenfeld, "Self-organized criticality: an explanation of 1/f noise," *Physical Review Letters*, **59**, 381–384 (1987).

[2] E. Berlekamp, J. Conway, and R. Guy, *Winning Ways for Your Mathematical Plays*, Vol. 2 (Academic, 1984).

[3] T. Bohr, G. Grinstein, Y. He and C. Jayaprakash, "Coherence, chaos, and broken symmetry in classical, many-body dynamical systems," *Physical Review Letters*, **58**(21), 2155–2158 (1987).

[4] T. Bohr and O.B. Christensen, "Size dependence, coherence, and scaling in turbulent coupled-map lattices," *Physical Review Letters*, **63**(20), 2161–2164 (1989).

[5] J.P. Crutchfield and K. Kaneko, "Phenomenology of spatio-temporal chaos," in *Directions in Chaos*, ed. Bailin Hao (World Scientific, 1987).

[6] J.P. Crutchfield and K. Kaneko, "Are attractors relevant to turbulence?" *Physical Review Letters*, **60**(26), 2715–2718 (1988).

[7] J.P. Crutchfield, "Hunting for transients and cycles," unpublished notes (March 1988).

[8] J.P. Crutchfield, "Subbasins, portals, and mazes: transients in high dimensions," *Nuclear Physics B* (Proc. Suppl.), **5A**, 287–292 (1988).

[9] F. Fogelman-Soulié, "Parallel and sequential computation on Boolean networks," *Theoretical Computer Science*, **40**, 275–300 (1985).

[10] E. Fredkin and T. Toffoli, "Conservative logic," *International Journal of Theoretical Physics*, **21**(3/4), 219–253 (1982).

[11] C.H. Goldberg, "Parity filter automata," *Complex Systems*, **2**(1), 143–149 (1988).

[12] e.g., J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields* (Springer-Verlag, 1983).

[13] H. Gutowitz, J. Victor and B. Knight, "Local structure theory for cellular automata," *Physica D*, **28**, 18–48 (1987).

[14] H. Gutowitz, "Hierarchical classification of cellular automata," *Physica D*, **45**(1-3), 136–156 (1990).

[15] H. Gutowitz, C. Langton, private communications.

[16] G. Grinstein, "Stability of nonstationary states of classical, many-body dynamical systems," *Journal of Statistical Physics*, **5**(5/6), 803–815(1988).

[17] J.J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences*, **79**, 2554–2558 (1982).

[18] E. Jen, "Global properties of cellular automata," *Journal of Statistical Physics*, **43**(1/2), 219–242 (1986);

E. Jen, "Invariant strings and pattern-recognizing properties of one-dimensional cellular automata," *Journal of Statistical Physics*, **43**(1/2), 243–265 (1986).

[19] K. Kaneko, "Lyapunov analysis and information flow in coupled map lattices," *Physica D*, **23**, 436–447 (1986).

[20] K. Kaneko, "Chaotic but regular posi-nega switch among coded attractors by cluster-size variation," *Physical Review Letters*, **63**(3), 219–223 (1989).

[21] K. Kaneko, "Clustering, coding, switching, hierarchical ordering, and control in a network of chaotic elements," *Physica D*, **41**, 137–172 (1990).

[22] K. Kaneko, "Super-transients, spatio-temporal intermittency and stability of fully developed spatio-temporal chaos," *Physics Letters A*, **149**(2,3), 105–112 (1990).

[23] F. Kaspar and H.G. Schuster, "Scaling at the onset of spatial disorder in coupled piecewise linear map," *Physics Letters A*, **113**, 451–453 (1986).

[24] S.A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *Journal of Theoretical Biology*, **22**, 437–467 (1969).

[25] S.A. Kauffman, "Emergent properties in randomly complex automata," *Physica D*, **10**, 145–156 (1984).

[26] C. Langton, "Studying artificial life with cellular automata," *Physica D*, **22**(1-3), 120–149 (1986).

[27] C. Langton, "Computation at the edge of chaos: phase transitions and emergent computation," *Physica D*, **42**, 12–37 (1990).

C. Langton, *Computation at the Edge of Chaos*, Ph.D Thesis, University of Michigan (1990).

[28] W. Li, "Power spectra of regular languages and cellular automata," *Complex Systems*, **1**(1), 107–130 (1987).

[29] W. Li, "Complex patterns generated by next nearest neighbors cellular automata," *Computer & Graphics*, **13**(4) 531–537 (1989).

[30] W. Li, *Problems in Complex Systems*, Ph.D Thesis, Columbia University (1989). (Available from the University Microfilm International, Ann Arbor, MI.)

[31] W. Li and N. Packard, "Structure of the elementary cellular automata rule space," *Complex Systems*, **4**(3), 281–297 (1990).

[32] W. Li, N. Packard and C. Langton, "Transition phenomena in cellular automata rule space," *Physica D*, **45**(1-3), 77–94 (1990).

[33] W. Li, "Transient times of cellular automaton rule 110," Santa Fe Institute preprint (1991).

[34] K. Lindgren and M. Nordahl, "Universal computation in simple one-dimensional cellular automata," *Complex Systems*, 4(3), 299–318 (1990).

[35] P.C. Matthews and S.H. Strogatz, "Phase diagram for the collective behavior of limiting-cycle oscillators," *Physical Review Letters*, **65**, 1701– (1990).

[36] R. May, "Simple mathematical models with very complicated dynamics," *Nature*, **261**, 459–467 (1976).

[37] W.S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity, " *Bulletin of Mathematical Biophysics*, **5**, 115–133 (1943).

[38] N. Packard, "Complexity of growing patterns in cellular automata," in *Dynamical systems and cellular automata*, eds. J. Demongeot, E. Goles, and M. Techuente (Academic Press, 1985).

[39] N. Packard, "Adaptation toward the edge of chaos," in *Complexity in Biological Modeling*, eds. S. Kelso and M. Shlesinger (World Scientific, 1988).

[40] J.K. Park, K. Steriglitz, and W.P. Thurston, "Soliton-like behavior in automata," *Physica D*, **19**, 423–432 (1986).

[41] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes in C* (Cambridge University Press, 1988).

[42] D.R. Rasmussen and T. Bohr, "Temporal chaos and spatial disorder," *Physics Letters A*, **125**(2,3), 107–110 (1987).

[43] D. Rummelhart and J. McClelland, *Parallel Distributed Processing* (MIT Press, 1986).

[44] L.S. Schulman and P.E. Seiden, "Statistical mechanics of a dynamical system based on Conway's Game of Life," *Journal of Statistical Physics*, **19**, 293–314 (1978).

[45] R. Shaw, *The Dripping Faucet as a Model Chaotic System* (Aerial Press, 1984).

[46] J. Theiler, "Mean field analysis of systems that exhibit self-organized criticality," Center for Nonlinear Studies preprint, Los Alamos National Lab (1991).

[47] J. Theiler, private communications.

[48] K. Tsang, R. Mirollo, S. Strogatz, and K. Wiesenfeld, "Dynamics of a globally coupled oscillator array," *Physica D*, to appear (1991).

[49] e.g., D.K. Umberger, C. Grebogi, E. Ott, and B. Afeyan, "Spatiotemporal dynamics in a dispersively coupled chain of nonlinear oscillators," *Physical Review A*, **39**, 4835– (1989).

[50] J. von Neumann, *Theory of Self-Reproducing Automata*, ed. A.W. Burks (University of Illinois Press, 1966).

[51] G.Y. Vichniac, P. Tamayo, and H. Hartman, "Annealed and quenched inhomogeneous cellular automata (ICA)," *Journal of Statistical Physics*, **45**(5/6), 875–883 (1986).

[52] C.C. Walker and W.R. Ashby, "On temporal characteristics of behavior in a class of complex systems," *Kybernetik*, **3**, 100–108 (1966).

[53] C.C. Walker, "Behavior of a class of complex systems: the effect of system size on properties of terminal cycles," *Journal of Cybernetics*, **1**(4), 57–67 (1971).

[54] C.C. Walker, "Predictability of transient and steady-state behavior in a class of complex binary nets," *IEEE Transactions on Systems, Man, and Cybernetics*, **3**(4), 433–436 (1973).

[55] C.C. Walker, "Stability of equilibrial states and limit cycles in sparsely connected, structurally complex Boolean nets," *Complex Systems*, **1**(6), 1063–1086 (1987).

[56] C.C. Walker, "Attractor dominance patterns in sparsely connected Boolean nets," *Physica D*, **45**(1-3), 441–451 (1990).

[57] K. Wiesenfeld and P. Hadley, "Attractor crowding in oscillator arrays," *Physical Review Letters*, **62**, 1335– (1989).

[58] W. Wilbur, D. Lipman and S. Shamma, "On the prediction of local patterns in cellular automata," *Physica D*, **19**, 397–410 (1986).

[59] S. Wolfram, "Statistical mechanics of cellular automata," *Review of Modern Physics*, **55**, 601–644 (1983).

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

[61] S. Wolfram, "Computation theory of cellular automata," *Communications in Mathematical Physics*, **96**, 15–57 (1984).

[62] S. Wolfram, "Twenty problems in the theory of cellular automata," *Physica Scripta*, **T9**, 170–183 (1985).

[63] "Appendix: Properties of the k=2 r=1 cellular automata," in *Theory and Applications of Cellular Automata*, ed. S. Wolfram (World Scientific, 1986).

[64] W. Wootters and C.G. Langton, "Is there a sharp phase transition for deterministic cellular automata?" *Physica D*, **45**(1-3) 95–104 (1990).