The Thirteenth International Symposium on Artificial Life and Robotics 2008(AROB 13th '08),
B-Con Plaza, Beppu, Oita, Japan, January 31-February 2, 2008

Reverse Engineering of Spatial Patterns in Cellular Automata

Yuuichi Ichise[1] and Yoshiteru Ishida[1,2]

[1]*Department of Knowledge-based Information Engineering*
Toyohashi University of Technology
Toyohashi, Aichi, 441-8580 Japan.
[2]*Intelligent Sensing System Research Center*
Toyohashi University of Technology,
Tempaku, Toyohashi, 441-8580 Japan

***Abstract*:** Cellular automata produce spatial patterns when specific rules for time development are given. This paper deals with an inverse problem of identifying the rules for spatial patterns given. Although only rules of one-dimensional elementary cellular automata and one-dimensional probabilistic cellular automata have shown here, the system can deal with two-dimensional one. When the rule identification has not been fully successful due to the lack of information in the spatial pattern, the system is able to give an identifiable part of the rules with a format of Wolfram's rule number.

***Keywords*:** complex systems, reverse engineering, cellular automata, state transition rules, rule number

I. INTRODUCTION

A cellular automaton (CA) consists of cells arranged in a *k*-dimensional lattice where *k* is a natural number. Each cell is an automaton which has a certain number of states; whose inputs are the state of neighbor cells; and the output is the state of the cell itself. In this paper, we restrict ourselves to the case of binary state: 0 and 1 and one-dimensional lattice where each cell has two neighbor cells: right and left. Each cell changes its state at the next time step based on the transition rules (which are identical for all the cells) and the current states of the neighbor cells. When the transition rule and initial configuration are specified, CA can produce a spatio-temporal pattern (Fig. 1).

In spite of simplicity of CA (regular structure of the lattice and rules are identical for every cell), CA is able to produce complex spatio-temporal patterns ranging from biological phenomena such as cancer growth and shell pigmentation to physical phenomena such as crystal growth, turbulent flow and soliton. It is well known that von Neuman used CA in his seminal work of self-reproducing automata [1]. In recent work, Wolfram proposed a qualitative classification into four classes [2,3]. Langton used CA to study artificial life that has a simple module capable of reproducing itself [4]. Not only deterministic cellular automaton (DCA) with deterministic rules, but also probabilistic cellular automaton (PCA) has been studied extensively. Domany and Kinzel proposed a simple PCA which can be regarded as Ising model [5].

When the rule and the initial configuration are specified, it is rather straightforward to obtain the spatio-temporal pattern of CA, however, the inverse problem of identifying the rule from a given spatio-temporal pattern is difficult and even impossible when enough information is not available in a given spatio-temporal pattern or the correspondence between rules and patterns is not one to one (Fig. 1). Spatio-temporal pattern becomes higher dimension when the dimension of the lattice for the CA becomes higher (Fig. 2).

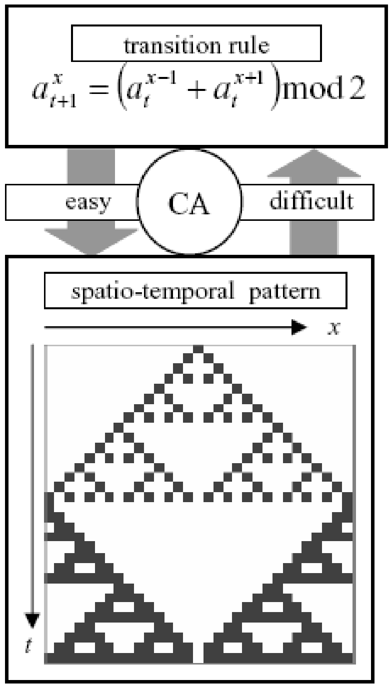

Fig. 1. From rules (above) to patterns (below) is simple, but from patterns to rules is difficult.

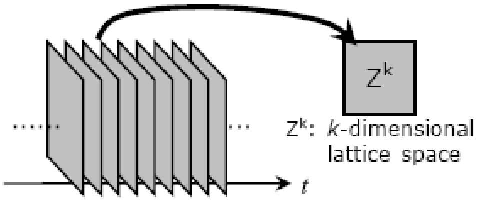

Fig. 2. Spatio-temporal pattern with *k*-dimensional lattice.

This paper deals with the inverse problem. We have constructed a system that will identify the rule for a given pattern by logically identifying the Boolean function.

II. TRANSITION RULES OF CA

Transition rules of one dimensional DCA with binary states: 0 and 1, are expressed in a form (1). In the form (1), the upper part of the cross-bar shows the pattern *k* in the neighborhood when the time is *t*, while the lower part indicates the state in the next time.

$a_t^x (\in S)$ denotes the state of the cell in a coordinate (*t*, *x*). $\sigma_k^r (\in S)$ denotes the next state when the neighborhood configuration *k* with the radius *r* of the neighborhood. *K* (=2^n) is the number of σ_k^r with the size of neighbor cells *n* (=2*r*+1).)

$$\frac{a_t^{x+r} \cdots a_t^x \cdots a_t^{x-r}}{a_{t+1}^x} : \frac{1\cdots1}{\sigma_{K-1}^r}, \cdots, \frac{0\cdots0}{\sigma_0^r} \quad (1)$$

For example, the rule can be specified by (2), (3) and (4), when *r* = 0, *r* = 1, and *r* = 2, respectively.

$$\frac{a_t^x}{a_{t+1}^x} : \frac{1}{\sigma_1^0}, \frac{0}{\sigma_0^0} \quad (2)$$

$$\frac{a_t^{x-1} a_t^x a_t^{x+1}}{a_{t+1}^x} : \frac{111}{\sigma_7^1}, \frac{110}{\sigma_6^1}, \frac{101}{\sigma_5^1}, \frac{100}{\sigma_4^1}, \frac{011}{\sigma_3^1}, \frac{010}{\sigma_2^1}, \frac{001}{\sigma_1^1}, \frac{000}{\sigma_0^1} \quad (3)$$

$$\frac{a_t^{x+2} a_t^{x+1} a_t^x a_t^{x-1} a_t^{x-2}}{a_{t+1}^x} : \frac{11111}{\sigma_{31}^2}, \frac{11110}{\sigma_{30}^2}, \cdots, \frac{00001}{\sigma_1^2}, \frac{00000}{\sigma_0^2} \quad (4)$$

The rules can be expressed by a number called the rule number in the form (5) when the radius of the neighborhood is *r*. Again, for example, the rule numbers corresponding to the cases (2), (3) and (4) can be expressed as (6), (7) and (8) , respectively.

$$r = r : \sigma_{K-1}^r \sigma_{K-2}^r \cdots \sigma_1^r \sigma_0^r \quad (5)$$

$$r = 0 : \sigma_1^0 \sigma_0^0 \quad (6)$$

$$r = 1 : \sigma_7^1 \sigma_6^1 \sigma_5^1 \sigma_4^1 \sigma_3^1 \sigma_2^1 \sigma_1^1 \sigma_0^1 \quad (7)$$

$$r = 2 : \sigma_{31}^2 \sigma_{30}^2 \sigma_{29}^2 \cdots \sigma_2^2 \sigma_1^2 \sigma_0^2 \quad (8)$$

For example, when *r* = 1 the rule number of the rule ($\sigma_7^1 = \sigma_5^1 = \sigma_2^1 = \sigma_0^1 = 0$, $\sigma_6^1 = \sigma_4^1 = \sigma_3^1 = \sigma_1^1 = 1$) can be expressed as "01011010" in the form of (7), which is 90 in the decimal numeration; hence called rule-90.

To specify the rule of PCA, p_k^r is used instead of σ_k^r. p_k^r is a probability of being "1" when the neighborhood pattern *k*. Thus, the rule of PCA with the neighborhood radius *r* can be specified as (9).

$$r = r : \left[p_{K-1}^r, p_{K-2}^r, \cdots, p_1^r, p_0^r\right] \quad (9)$$

III. IDENTIFYING RULES OF CA

1. An algorithm for identifying the rule of DCA

When a subset of spatio-temporal patterns was give, the following calculation is carried out at each coordinate (*t*, *x*) of the subset.

Calculation at a coordinate (*t*, *x*)

[1] The radius *r* is fixed to be 0 through the calculation.

[2] All σ_k^r are set to be the symbol *u* (which stands for unknown).

[3] Set the neighborhood pattern *k* to a_t^{x+r}, … , a_t^x, … , a_t^{x-r} by the given spatio-temporal pattern.

[4] When σ_k^r is *u*, σ_k^r is reset to be a_{t+1}^x by the given spatio-temporal pattern.

[5] σ_k^r is set to be the symbol *d* (which stands for discrepancy) when σ_k^r is neither *u* nor a_{t+1}^x.

[6] The process from [2] to [5] is repeated until all the coordinates in the given spatio-temporal patterns are scanned. After that if there is still *d* symbol, *r* → *r*+1 and go back to [2], if else (no *d* symbol) the calculation is terminated.

The identification success rate (ISR) is defined to be:

$$\mathrm{ISR} = 1 - \frac{\text{(The number of } u \text{ remained)}}{\text{(The total number of the neighborhood)}} . \quad (10)$$

2. An algorithm for identifying the rule of PCA

In identifying the rules of PCA, we assume the neighborhood radius r is given to reduce the calculation. Except this, the algorithm is similar to that for DCA. The frequency of the state "1" is counted for the neighborhood pattern k, and p_k^r is estimated by the normalized frequency of the state "1".

IV. EXAMPLE OF RESULTS

1. Rule identification with sufficient patterns

Fig. 3 shows a spatio-temporal diagram similar to the in Fig. 1. A white cell is in the state 0 and a black cell is in the state 1. The time increases as it goes downwards. To generate the spatio-temporal pattern (Fig. 3), the rule-96 (in hexadecimal numeral) with r=2 is used. The patterns close to the initial part (indicated as white horizontal zone in Fig. 3) are used to test the rule identification system. Table 1 shows the test results.

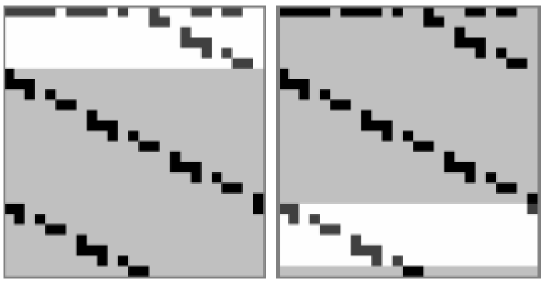

Fig. 3. Given spatio-temporal pattern of rule-96. White horizontal zone close to the initial one (left; referred as L3) and that after 20 steps (right; referred as R3) are given to the system.

Table 1. Test result for the rule-96 (r = 2) using the spatio-temporal pattern of L3 (Fig. 3: left) and R3 (Fig. 3: right).

item	Rule ID in the form of (8)
The rule for the pattern	00000000000000000000000010010110 (r = 2)
The rule estimated from L3	00000u0000u000000000000010010110 (r = 2)
ISR	0.937500
The rule estimated from R3	uuu0uuu0uuu0uuu0u0u0u0u010010110 (r = 2)
ISR	0.500000

Table 1 compares the test results for the same rule (the rule-96) but different zone in spatio-temporal pattern. Although the test using the spatio-temporal patterns close to the initial one is successful: ISR=0.9375, the test when the initial part is not used almost fails: ISR=0.5. This comparison suggests that the initial part of the spaio-temporal patterns (hence transient pattern) might contain more information useful for identifying the rule, since the state is in transient.

2. Rule identification with insufficient patterns

The lack of information in the pattern affects not only the rule identification, but also that of the neighborhood radius. Here we compare the cases of the transient part including initial one (indicated as a white horizontal zone in Fig. 4 left) and of the stationary one after time steps (indicated as a white horizontal zone in Fig. 4 right) for the same rule-232 (in decimal numeral) with r=1.

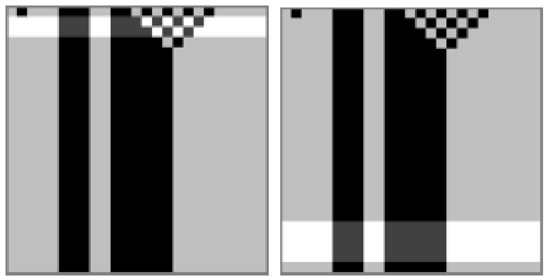

Fig. 4. Given spatio-temporal pattern of rule-232. White horizontal zone close to the initial one (left; referred as L4) and the stationary one (right; referred as R4) are given to the system.

Table 2. Test result for the rule-232 (r = 1) using the spatio-temporal pattern of L4 (Fig. 4: left) and R4 (Fig. 4: right).

item	rule ID in the form of (7)
The rule for the pattern	11101000 (r = 1)
The rule estimated from L4	11101000 (r = 1)
ISR	1.000000
The rule estimated from R4	10 (r = 0)
ISR	1.000000

Due to the lack of information in the stationary pattern (Fig. 4: left), the rule identification system fails to identify even the radius as shown in Table 2, while the test using the transient pattern (Fig. 4: right) is perfect (ISR=1 as in Table 2).

3. Rule identification depends on spatio-temporal complexity

Although stationary patterns are used the identification success rate can differ when the spatio-temporal complexity in the pattern differs. Left and right figures in Fig. 5 show a white horizontal zone when the rule-120 (r=1) is used but the initial configurations are distinct.

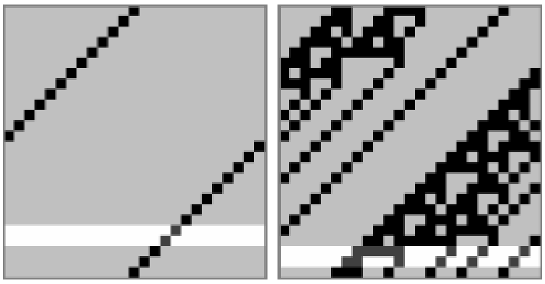

Fig. 5. Given spatio-temporal pattern of the rule-120. White horizontal zone with low complexity (left; referred as L5) and that with high complexity (right; referred as R5) are given to the system.

Table 3. Test result for the rule-120 (r = 1) using the spatio-temporal pattern of L5 (Fig. 5: left) and R5 (Fig. 5: right).

item	rule ID in the form of (7)
The rule for the pattern	01111000 (r = 1)
The rule estimated from L5	uuu1u000 (r = 1)
ISR	0.500000
The rule estimated from R5	01u11000 (r = 1)
ISR	0.875000

As compared in Table 3, the identification success rate using the complex pattern (Fig. 5: right) is much higher (ISR=0.875) than that (ISR=0.5) using monotonous pattern (Fig. 5: left).

4. Rule identification of PCA

In the identification of the rule of PCA, the neighborhood radius is fixed. Fig. 6 shows a spatio-temporal pattern of the Domany-Kinzel model [5] (r=1 in this model). The rule ID of this PCA can be written as: r=1: $[q,p,q,p,p,0,p,0]$ with the probability p=0.8 for (0*1)/1 and q=0.2 for (1*1)/1; from t=0 to 100 steps. As shown in Table 4 the rule estimation using more volumes of time steps (10000 steps) outperforms the estimation using only 100 time steps.

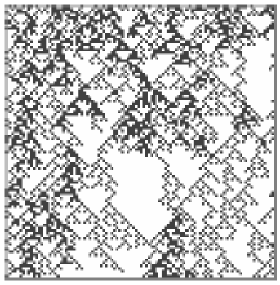

Fig. 6. Given spatio-temporal pattern of PCA; the first 100 steps.

Table 4. Test result for the PCA; with the first 100 steps (Fig. 6) and with the first 10000 steps

item	rule ID in the form of (9)
The rule for the pattern	[0.2, 0.8, 0.2, 0.8, 0.8, 0.0, 0.8, 0.0]
The rule estimated using 100 steps	[0.177, 0.777, 0.209, 0.782, 0.806, 0.000, 0.809, 0.000]
The rule estimated using 10000 steps	[0.203, 0.796, 0.201, 0.800, 0.801, 0.000, 0.801, 0.000]

V. CONCLUSION

We proposed an algorithm for identifying the rule for a given spatio-temporal pattern by cellular automata. As shown by examples, the rule identification will be successful as the complexity in the given spaio-temporal pattern is sufficiently rich. For identifying the rule of probabilistic cellular automata, much volume of spatio-temporal patterns with high complexity is necessary.

Acknowledgements

This work was supported by The Global COE Program "Frontiers of Intelligent Sensing", from the ministry of Education, Culture, Sports, Science and Technology. This work was also supported in part by Grants-in-Aid from Toyohashi University of Technology.

REFERENCES

[1] von Neumann JJ (1966), *Theory of Self-Reproducing Automata*. edited and completed by A. Burks, University of Illinois Press, Champaign, IL

[2] Wolfram S (1984), Universality and Complexity in Cellular Automata. *Physica D* 10:1-35

[3] Packard N and Wolfram S (1985), Two-dimensional cellular automata. *Journal of Statistical Physics* 38:901-946

[4] Langton CG (1992), Life at the Edge of Chaos" *Artificial Life II, Santa Fe Institute Studies in the Sciences of Complexity*, 10:41-91

[5] Domany E and Kinzel W (1984), Equivalence of cellular automata to Ising models and directed percolation. *Phys. Rev. Lett.* 53:311