数理解析研究所講究録 1244 巻 2002 年 69-74

Discrete Dynamics on Cellular Automata

永井喜則(Yoshinori NAGAI), and Ted MADDESS*

国士舘大学 情報科学センター及政経学部(Center for Information Science, and Faculty of Political Science & Economy, Kokushikan University) and *Visual Sciences, Research School of Biological Sciences, Australian National University.

Cellular automata (CA) have many applications and give a convenient scheme of discretized modeling of a system [1]. There are several type of rule construction methods such as filtered CA [1,2] or Wolfram type CA [1,3]. The difference these rule constructions seems to be the same when we see a general class of CA, namely, n neighbors and L states (or L levels). However, filtered CA has a different feature from CA with other rule construction methods, because the filtered CA uses semi-infinite neighbors usually.

Here we develop Wolfram type CA to the CA of n neighbors and L levels which denoted by CA^{L}_{n}. We show a method how organize rules for n neighbor and L level CA (CA^{L}_{n}). We recognized what kind of nature gives fractal feature of CA, that is how comes self-similarity out on CA rule structures. We also know that CA properly has the rule-dynamical property. Note that rule-dynamics was proposed by Aizawa [4]. In this note, we briefly sketch these subjects and give a scope to see a discrete world of cellular automata.

From Two-Level CA to Multi-Level CA

Here we explain wolfram type CA and show a general scheme of CA. a simple Wolfram type CA is a one-dimensional CA is consisting of N cells each of which has two states and three neighbor relation to determine the state of a cell at next time step. An example is shown as follow:

○●○○○●●●○○●○○○○○○ (cell states at time t),
●○○●●○●○○●○○●●●●● (cell states at time t+1)

The temporal development of each cell state is governed by following equation,

$$S_i(t+1)=F(S_{i-1}(t), S_i(t), S_{i+1}(t)),$$

where $S_i(t)$ denotes i-th cell state, takes one of two values such as {0,1}, at time t, and F(...) is a mapping function called "rule" depicted as below.

```
i-1  i  i+1
 o   o   o        at time t
     ↓ F
     o            at time t+1
     I
```

As well known [3], the number of rules (mappings) for CA^2_3 (two levels and three neighbors) is 2^8 =256.

The general class of CA can obtain simply by extending neighbors and levels. The extended scheme is explained as follows:

```
      Si(t)
       i
...o o o o o... n neighbors
      ↓ F
      io
     Si(t+1)
```

where Si(t) takes one of L levels, namely, $S_i(t) \in \{s_1, s_2, s_3, \ldots, s_L\}$ and the temporal development of Si(t) is given by the recurrence equation, i.e.,

$$S_i(t+1)=F(S_{i-n/2}(t), \ldots, S_{i-1}(t), S_i(t), S_{i+1}(t), \ldots, S_{i+n/2}(t)).$$

Notice that F(...) means rules (i.e., mappings), and n is assumed to be an odd integer. We can know any case of temporal patterns of 1-D cell array if we get the manner to construct entire rules for CA^L_n.

To apply for the cellular automata modeling of a given system, an important

matter is what kind of spatial relation between cells must be introduced. Level and neighbor are also required to make an appropriate model on CA. Usually, there are several rules that give the same temporal development of cell states which is called pattern dynamics. A class of rules on L-level and n-neighbor CA exists to realize the same pattern dynamics.

Rule Description of CA^L_n

Now we consider a method to construct the rules for general type CA. Let a CA system has L-levels and be denoted by $\{s_1, s_2, s_3,\ldots,s_L\}$. Here we show a construction method to use a recurrence relation. We start from one-variable functions and organize a recurrence relation by use of point functions shown below. There are L^L numbers of one-variable functions $\{ g_1(X), g_2(X), g_3(X), \ldots, g_L{}^L(X)\}$. We tabulate the Boolean like relation of levels (or states) for these one-variable functions in Table I.

Table I One-variable functions for CA^L_n

X	g(X)				
	g_1	g_2	g_3		$g_L{}^L$
S_1	S_1	S_1	S_1	.....	S_L
S_2	S_1	S_1	S_1	.....	S_L
•	•	•	•	.....	•
•	•	•	•	.....	•
•	•	•	•	.....	•
S_{L-1}	S_1	S_1	S_2	.....	S_L
S_L	S_1	S_2	S_1	.....	S_L

Whenever we introduce a function for every level s_j defined by

$$Is_j(X) = \begin{cases} 1 & \text{for } X=s_j \\ 0 & \text{Otherwise,} \end{cases}$$

we can construct any rule for any neighbors. We call the above function "point function". There exist L point functions for L-level CA, namely, $\{Is_1, Is_2, Is_3, \ldots, Is_L\}$. We can concrete the form of point functions by a typical description for level s_j as below.

$$Is_j(X)=X/s_j \cdot (1-X/s_1)/(1-sj/s_1)\ldots(1-X/s_{j-1})/(1-sj/s_{j-1}) \cdot (1-X/s_{j+1})/(1-sj/s_{j+1})\ldots(1-X/s_L)/(1-s_j/s_L)$$

At the usage of above form, the definition for singular cases X/0=1 and (1-X/0)/(1-s/0)=1 should be followed.

Using the point functions, we can construct the rules for any neighbors CA of L levels. The n-neighbor rules are identical to n-variable discrete functions. Two variable discrete functions can be obtain from one-variable discrete functions through point finctions by the following recurrence equation:

$$F_{j1j2\ldots jL}(X, Y) = Is_1(X)g_{j1}(Y)+Is_2(X)g_{j2}(Y)+\ldots+Is_L(X)g_{jL}(Y),$$
$$(j_1, j_2, j_3, \ldots, j_L = 1, 2, 3, \ldots, L^L).$$

As known from above expression, there are L^W $(W=L^2)$ two-variable discrete functions. We can establish the same type recurrence relation between n and n+1 variable discrete functions. The result is

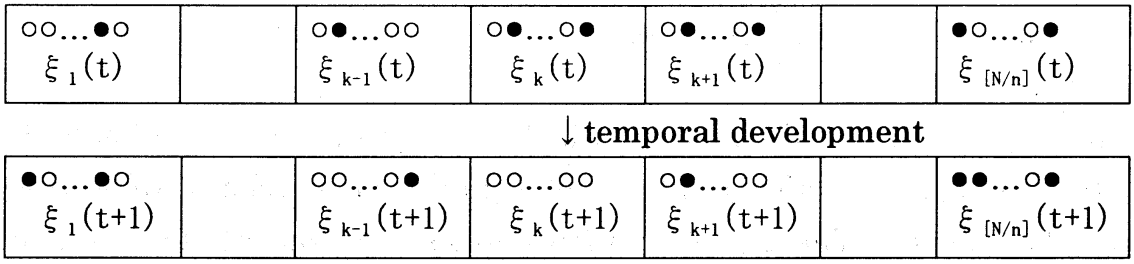
$$F_{j1j2\ldots jL}(X_{n+1}, X_n, \ldots, X_1)=Is_1(X_{n+1})f_{j1}(X_n,\ldots,X_1)+\ldots+Is_L(X_{n+1})f_{jL}(X_n,\ldots,X_1),$$
$$(j_1, j_2, j_3, \ldots, j_L = 1, 2, 3, \ldots, L^{W(W=L^n)}),$$

where $f_{j1}(X_n,\ldots X_1)$, $f_{j2}(X_n,\ldots,X_1)$, ..., $f_{jL}(X_n,\ldots,X_1)$ are n variable discrete functions generated by n time iterations started from two and one variable relation.

Self-Similarity and Rule Dynamical Property of CA

Here, we consider self-similarity of CA which leads us to a recognition that CA has the rule-dynamical property (concise explanation is seen in [4]). A higher level CA is produced by dividing cell array to neighbor size blocks. In this dividing, each cell block can be described by using L^n levels. Every cell block is affected by both neighbored blocks and itself in its temporal state changes. A half of cells in a neighbored block contribute to another neighbored

blocks to each other. The situation is illustrated below.



We note that N is the system size (i.e., number of cells in the system), as already stated above. On these blocking, we conclude that three-neighbor CA gives a general picture on Wolfram type CA. We can describe above CA of block cells by the same type difference equation of temporal development of every block cells by using block level variable ξ, i.e.,

$$\xi_k(t+1) = F(\xi_{k-1}(t), \xi_k(t), \xi_{k+1}(t)),$$

where $\xi_i(t)$ denotes i-th block clustered together with neighbor size at time t. We can completely recognize states for each block cells with preparing L^n levels for a block. Then the original CA reaches to the CA of level L^n and neighborhood three.

It is possible such blocking for the cell block automata. We therefore know that neighborhood three may be fundamental in CA, and that cellular automata have the self-similarity feature by above blocking. If we introduce a coarse graining of block state by use of original number of levels, and if the rule is invariant for this coarse graining, the fractal property of spatio-temporal pattern of cell states in CA is realized.

Now we discuss rule-dynamical property of CA. As seen from above block making, it is an interesting aspect that CA is consisting of blocks each of which develops its states temporally by its own manner. The temporal development of a block cells can be described by a mapping ψ from the block level ξ(t) at time t to the block level ξ(t+1) at time t+1. This mapping actually depends on neighbored blocks. They yield boundary conditions for this mapping because cells in a block near the boundary require extra-cells to determine the state at next time step. These extra cells are supplied from two neighbored blocks. Taking these points into account, we can describe the temporal development of k-th block state (or

level) by following equation:

$$\xi_k(t+1) = \Sigma_{(a,b)}\, I_{(a,b)}\,(\xi_{k-1}(t)=a,\ \xi_{k+1}(t)=b)\ \psi_{a,b}(\xi_{k-1}(t),\ \xi_k(t),\ \xi_{k+1}(t)),$$

where $I_{(a,b)}(...)$ signifies the function which takes the value 1 if and only if $\xi_{k-1}(t)$ and $\xi_{k+1}(t)$ take the specified cell levels of boundary effect cells, or takes the value 0 for other cases . The above equation implies that the rule to determine the level at next time step in k-th block is temporally changed by the both side of neighbored blocks. In other words, every block undergoes rule-dynamics in the sense of temporal changing of rule (or mapping). Each block has $L^{2[n/2]}$ possible rules. Notice that [x] denotes the integer part of x.

References

[1] A. Ilachinski, Cellular Automata. World Scientific, Singapore 2001.

[2] A. Ankiewicz, Y. Nagai, "Solitons in multi-level cellular automata". to be appeared in Chaos Solitons & Fractals.

[3] S. Wolfram, Theory and Application of Cellular Automata. World Scientific, Singapore 1986.

[4] Y. Nagai, Y. Aizawa, "Rule-dynamical generalization of McCulloch-Pitts neuron networks". BioSystems: 58 (2000) 177-185.