Title	3次元可逆自己増殖セル・オートマトンについて (計算モデルとアルゴリズム)
Author(s)	Hori, Takahiro; Imai, Katsunobu; Morita, Kenichi
Citation	数理解析研究所講究録 (1999), 1093: 206-211
Issue Date	1999-04
URL	http://hdl.handle.net/2433/62946
Right	
Type	Departmental Bulletin Paper
Textversion	publisher

数理解析研究所講究録
1093 巻 1999 年 206-211

3 次元可逆自己増殖セル・オートマトンについて

堀貴博 †, 今井克暢, 森田憲一 ‡

† NEC アイシーマイコンシステム
thori@nims.nec.co.jp
‡ 広島大学工学部
{imai, morita}@ke.sys.hiroshima-u.ac.jp

1 Introduction

A reversible cellular automaton (RCA) is one of the reversible computing models. Its global function is injective and every configuration has at most one predecessor. Intuitively, it "remembers" the initial configuration and one can reconstruct its initial configuration from a configuration of any time. So reversible property is a strong constraint and one cannot generate nor extinct signals freely. Toffoli showed that there exist computation-universal RCAs[13] and the BBM cellular automaton (BBMCA), was introduced by Margolus[7]. It is computation-universal and it has a direct relation with a physical reversible and conservative computing model (the Billiard Ball Model)[3]. The Billiard Ball Model (BBM) has an important aspect that it is possible to compute any function without dissipation of balls as garbage. Once von Neumann conjectured that computing without erasing information is impossible and erasing one-bit information must dissipate at least $ln2\ kT$ joules of energy. But the BBM is a conservative model and it is possible to construct a computer that can computes with no energy dissipation in principle.

This fact means garbage collection can be performed with reversed sub-computations and such garbage collector can be constructed with reversible manner. This technique is first introduced by Bennett. He showed that his reversible Turing machine is computation-universal and garbage collection can be implemented in reversible manner[1, 2]. But the fact does not mean any computing process can be simulated effectively in reversible[10, 4]. Especially, it is very difficult to place a preferred initial configuration and start computing on reversible computing models. This problem is described as a restriction of generation and extinction of signals in RCAs, and synchronizing signals and distributing specific patterns on RCAs are also quite difficult.

So we proposed a simple self-reproducing RCA based on a shape-encoding mechanism (SR_8)[11]. In SR_8, self-reproduction can be performed without garbage in two-dimensional reversible cellular space.

In this paper, we extend SR_8 into three-dimensional reversible cellular space. Even if its cellular space is reversible, it can self-reproduce various three-dimensional patterns without garbage dissipation. In order to design an RCA we use a framework of partitioned cellular automaton (PCA). In the next section, first we define PCA.

2 Definitions

Partitioned cellular automaton (PCA)[8] is regarded as the subclass of standard cellular automaton. Each cell is partitioned into the equal number of parts to the neighborhood size and the information stored in each part is sent to only one of the neighboring cells (Fig. 1). In PCA, injectivity of global function is equivalent to injectivity of local function, thus a PCA is reversible if its local function is injective. Using PCA, we can construct a reversible CA with ease.

A deterministic two-dimensional partitioned cellular automaton (PCA) P is defined by

$$P = (\mathbf{Z}^2, (C, U, R, D, L), \varphi, (\#, \#, \#, \#, \#))$$

where $\mathbf{Z}$ is the set of all integers, C, U, R, D, L are non-empty finite sets of center, up, right, down, left parts of each cell, $\varphi : C \times D \times L \times U \times R \to C \times U \times R \times D \times L$ is a local function (Fig.2), and $(\#, \#, \#, \#, \#) \in C \times U \times R \times D \times L$ is a quiescent state which satisfies $\varphi(\#, \#, \#, \#, \#) = (\#, \#, \#, \#, \#)$.

A configuration over $C \times U \times R \times D \times L$ is a mapping $c : \mathbf{Z}^2 \to C \times U \times R \times D \times L$. Let $\mathrm{Conf}(C \times U \times R \times D \times L)$ denote the set of all configurations over $C \times U \times R \times D \times L$.

$$\mathrm{Conf}(C \times U \times R \times D \times L) = \{c | c : \mathbf{Z}^2 \to C \times U \times R \times D \times L\}$$

Global function

$$\Phi_A : \mathrm{Conf}(C \times U \times R \times D \times L) \\ \rightarrow \mathrm{Conf}(C \times U \times R \times D \times L)$$

is defined by

$$\begin{aligned}\Phi_A(c)(x) = \varphi(&\mathrm{CENTER}(c(x,y)),\\ &\mathrm{DOWN}(c(x,y+1)),\\ &\mathrm{LEFT}(c(x+1,y)),\\ &\mathrm{UP}(c(x,y-1)),\\ &\mathrm{RIGHT}(c(x-1,y)))\end{aligned}$$

where CENTER (UP, RIGHT, DOWN, LEFT, respectively) is the projection function which picks out the center (up, right, down, left) element of a quintuple in $C \times U \times R \times D \times L$. It has been proved that P is reversible iff φ is one-to-one[8].

P is called a *rotation-symmetric* (or *isotropic*) PCA iff (i) and (ii) hold.

(i) $U = R = D = L$.

(ii) $\forall (c,u,r,d,l),(c',u',r',d',l') \in C \times U^4$: if $g(c,d,l,u,r) = (c',u',r',d',l')$ then $g(c,r,d,l,u) = (c',l',u',r',d')$.

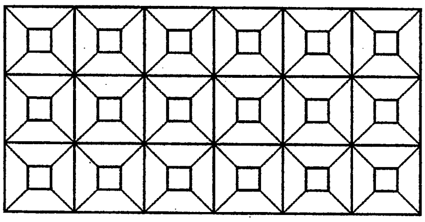

Figure 1: Cellular space of PCA.

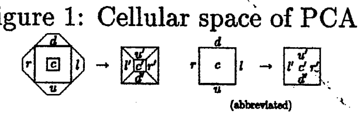

Figure 2: A representation of a rule.

A deterministic three-dimensional partitioned cellular automaton (PCA) P_3 is also defined by

$$P_3 = (\mathbf{Z}^3, (C,U,R,D,L,F,B), \varphi_3, \\ (\#,\#,\#,\#,\#,\#,\#))$$

where $\mathbf{Z}$ is the set of all integers, C, U, R, D, L, F, B are non-empty finite sets of center, up, right, down, left, forward, back parts of each cell, $\varphi_3 : C \times D \times L \times U \times R \times B \times F \rightarrow C \times U \times R \times D \times L \times F \times B$ is a local function (Fig.3), and $(\#,\#,\#,\#,\#,\#,\#) \in C \times U \times R \times D \times L \times F \times B$ is a quiescent state which satisfies $\varphi_3(\#,\#,\#,\#,\#,\#,\#) = (\#,\#,\#,\#,\#,\#,\#)$.

A configuration over $C \times U \times R \times D \times L \times F \times B$ is a mapping $c : \mathbf{Z}^3 \rightarrow C \times U \times R \times D \times L \times F \times B$. Let $\mathrm{Conf}(C \times U \times R \times D \times L \times F \times B)$ denote the set of all cnfigurations over $C \times U \times R \times D \times L \times F \times B$.

$$\mathrm{Conf}(C \times U \times R \times D \times L \times F \times B) = \\ \{c | c : \mathbf{Z}^3 \rightarrow C \times U \times R \times D \times L \times F \times B\}$$

Global function is also defined in the same way as in the two-dimensional case.

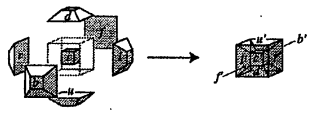

Figure 3: Domain and rage of local function in 3D 7-neighbor PCA.

3 Self-reproduction in a two-dimensional RPCA

3.1 Definition of SR_8

In this section, we construct non-trivial self-reproducing structures can be constructible in a reversible cellular space. The idea of our model is based on Langton's sheathed Loop[6], and to achieve more flexibility we introduced a self-inspection method.

In the cellular space of SR_8[11], encoding the shape of an object into a "gene" represented by a command sequence, copying the gene, and interpreting the gene to create an object, are all performed reversibly. By using these operations, various objects called Worms and Loops can reproduce themselves in a very simple manner.

The RPCA "SR_8" is defined by

$$SR_8 = (\mathbf{Z}, (C,U,R,D,L), g, (\#,\#,\#,\#,\#)),$$
$$C = U = R = D = L = \{\#, *, +, -, \mathrm{A,B,C,D}\}.$$

Hence, each of five parts of a cell has 8 states. The states A, B, C and D mainly act as signals that are used to compose "commands". The states $*, +$, and $-$ are used to control these signals. The local function g contains 765 rules. It is a one-to-one mapping and rotation-symmetric.

3.2 Signal transmission on a Wire

A *wire* is a configuration to transmit signals A, B, and C. Fig. 4 shows an example of a part of a simple (i.e., non-branching) wire.

A *command* is a signal sequence composed of two signals. There are six commands consisting of signals A, B and C as shown in Table 1. These commands are used for extending or branching a wire.

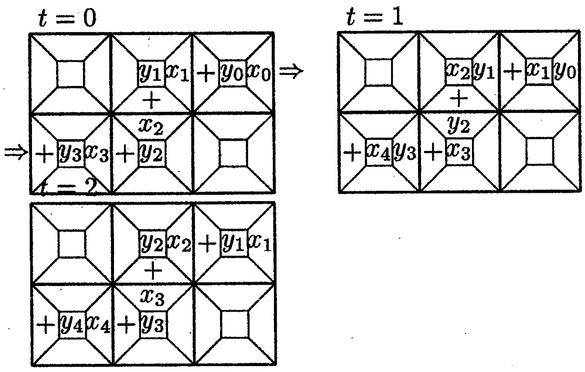

Figure 4: Signal transmission on a part of a simple wire ($x_i, y_i \in$ {A,B,C}).

Command		Operation
First signal	Second signal	
A	A	Advance the head forward
A	B	Advance the head leftward
A	C	Advance the head rightward
B	A	Branch the wire in three ways
B	B	Branch the wire in two ways (making leftward branch)
B	C	Branch the wire in two ways (making rightward branch)

Table 1: Six commands composed of A, B, and C.

3.3 A Worm

A *Worm* is a simple wire with open ends that are called a *head* and a *tail.* It crawls in the reversible cellular space as shown in Fig. 5.

Commands in Table 1 are decoded and executed at the head of a Worm. That is, the command AA extends the head straight, while the command AB (or AC, respectively) extends it leftward (rightward). On the other hand, at the tail cell, the shape of the Worm is "encoded" into an advance command. That is, if the tail of the Worm is straight (or left-turning, right-turning, respectively) in its form, the command AA (AB, AC) is generated. The tail then retracts by one cell.

3.4 Self-reproduction of a Worm

By giving a branch command, *any* Worm can self-reproduce indefinitely provided that it neither cycles nor touches itself in the branching process.

3.5 Self-reproduction of a Loop

A *Loop* is a simple closed wire, thus has neither a head nor a tail as shown in Fig. 6.

If a Loop contains only advance or branch commands, they simply rotate in the Loop and self-

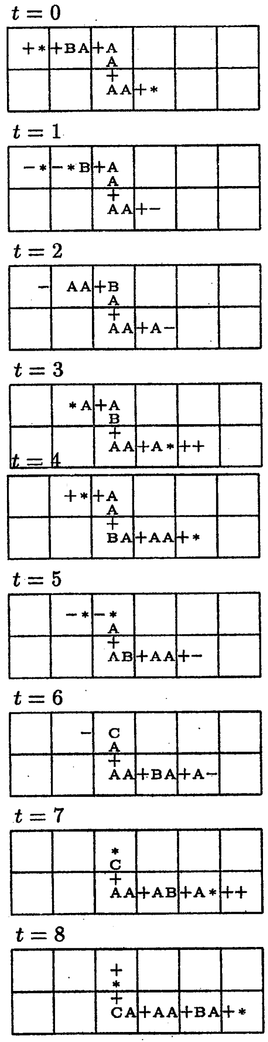

Figure 5: Behavior of a Worm.

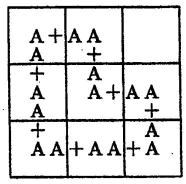

Figure 6: An example of a Loop.

Command		Operation
First signal	Second signal	
D	B	Create an arm
D	C	Encode the shape of a Loop

Table 2: Commands DB and DC.

reproduction does not occur. In order to make a Loop self-reproduce, commands in Table 2 are used.

By putting a command DB at an appropriate position, *every* Loop having only AA commands in all the other cells can self-reproduce in this way. When DB reaches the bottom right corner, it starts making an "arm" and this corner become a transmitter of commands. First, all AA commands in the mother Loop are transmitted through the arm and generated DC commands encode whole shape of the mother Loop into command sequences simultaneously and these commands are transmitted after all static AA commands are transmitted. Finally DC commands reaches the bottom right corner and the arm is split from the mother Loop.

3.6 Controlling the position of daughter Loops in SR_8

One of our main motivations is to place preferred initial patterns to a reversible cellular space. As mentioned above, a closed Loop has only AA commands. If AB or AC commands are placed in the Loop, generated position of the daughter Loop can be changed.

But DB (create an arm command) advances the bottom side of a loop and the length of the Loop does not equal to the running length of the whole commands. Thus the embedded position of turning commands in the daughter Loop differ from the mother Loop.

Although such a shifting of reading-frames of its command sequence is interesting phenomenon, it is difficult to control. So we modify SR_8 for solving this timing problem.

Fig.7 is the process of modified version of SR_8. DB signal is not advance bottom side and the reproducing process starts from the bottom right corner as soon as the Worm reaches at this position. So created daughter Loop is rotated in 90 degrees. Because of this rotation, Loops make collision after 4 generations. But this collision can be avoidable by inserting direction commands into the mother Loop (Fig.8) and this modification acts important roll in extending SR_8 to three-dimensional one in the next section.

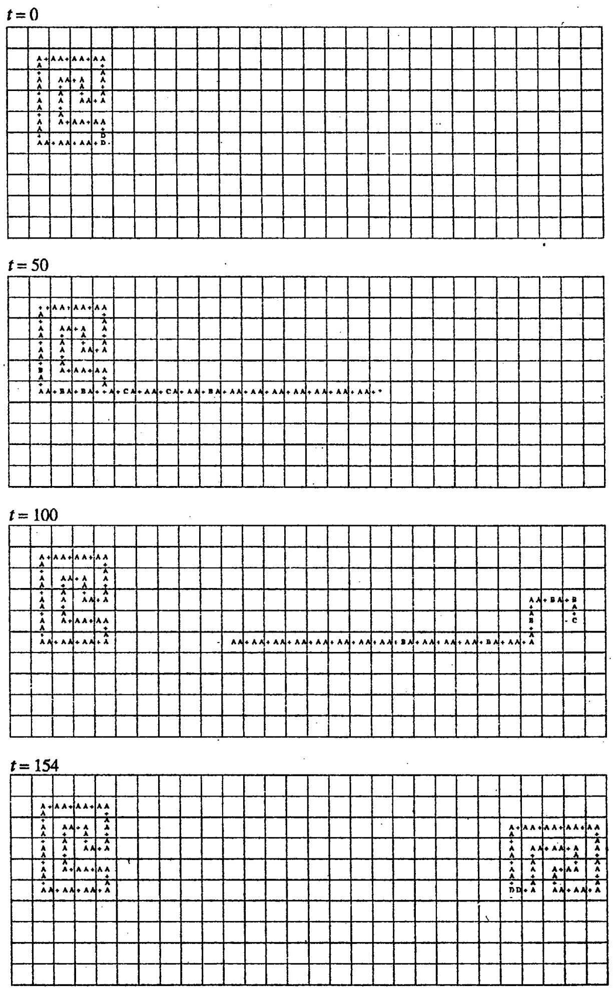

Figure 7: Self-reproducing process of a Loop of modified SR_8.

4 Self-reproduction in a three-dimensional RPCA

4.1 Three-dimensional self-reproducing RPCA (SR_9)

In this section, we extend SR_8 into a three-dimensional RPCA.

A two-dimensional 5-neighbor PCA can be embedded directly into a three-dimensional 7-neighbor PCA. But due to the rotation-symmetric condition of SR_8, the Worm cannot know the directions of right, left, up and down. In three-dimensional rotation-symmetric CA, up to 24 rotated rules are regarded as the same rule. So we introduce another glue state '=' for SR_8 and combine two Worms whose command sequences are complementaly placed as presented in table 3.

The three-dimensional self-reproducing RPCA

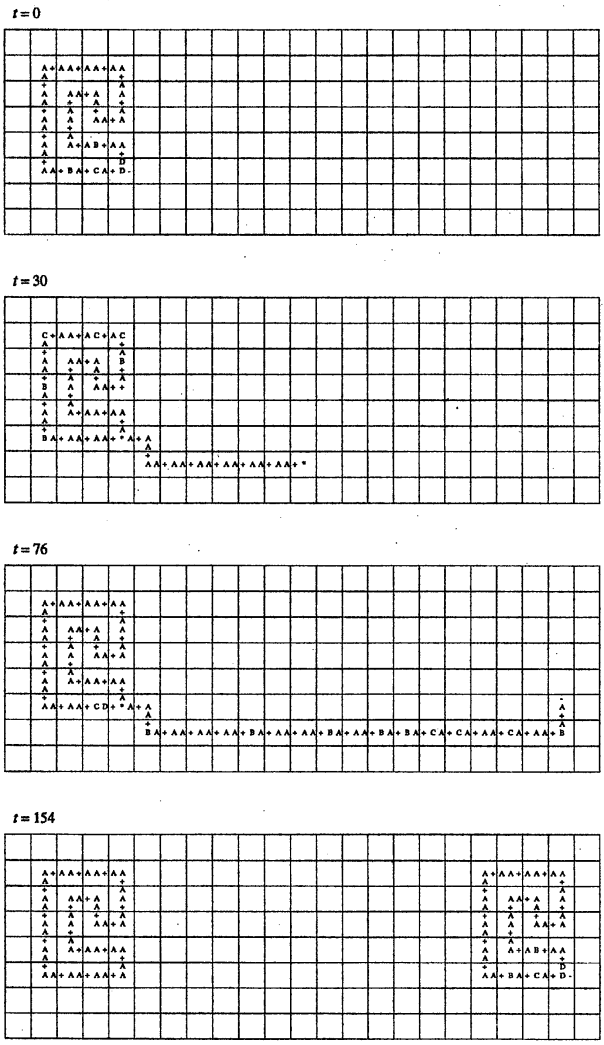

Figure 8: Self-reproducing process of a Loop of modified SR_8.

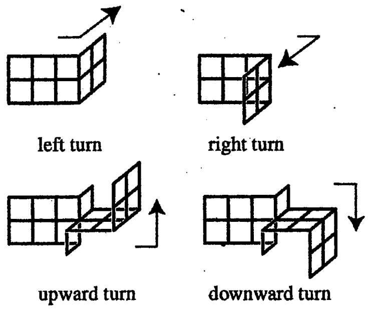

Figure 9: Four kind of turns in SR_9.

width 2 ladder approach in the previous section is impossible. So we add a center wire and Fig.10 is a simple Worm in SR_9.

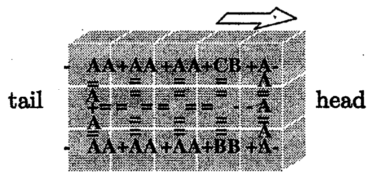

Figure 10: A simple Worm in SR_9.

Table 3: 'A'-Commands for width 3 shaped worm.

Command				Operations
wire 1		wire 3		
First Signal	Second Signal	First Signal	Second Signal	
A	A	A	A	Advance the head forward
A	B	A	C	Advance the head leftward
A	C	A	B	Advance the head rightward
A	B	A	B	Start rotating (leftward)
A	C	A	C	Start rotating (rightward)

"SR_9" is defined by

$$SR_9 = (\mathbf{Z^3}, (C, U, R, D, L, F, B), g, \\ (\#, \#, \#, \#, \#, \#, \#)), \\ C = U = R = D = L = F = B = \\ \{\#, *, +, -, =, A, B, C, D\}.$$

Local rules are available via WWW.
`http://kelp.ke.sys.hiroshima-u.ac.jp/projects/rca/sr3d/`

Although SR_9 has 6886 rules, if rotated rules are regarded as equivalent, it become only 338 rules[5].

To construct 'true' three-dimensional structures, a Worm in SR_9 can twist its head in ±90 degrees (Fig.9). This can be possible by employing ribbon of width 3 shaped Worms. As far as using SR_8 command sequences in rotation-symmetric spaces, the length of the path should be kept equal and the

When both wires of a three-dimensional Worm have the same sequence 'AB AA AC' (or 'AC AA AB'), its head is twisted leftward (rightward). Using twisting commands, complex three-dimensional Worms and Loops such as Fig.11 are constructible. Although the existence of twisting commands in SR_9, its self-reproducing mechanism is completely the same as that of SR_8.

4.2 Controlling the position of daughter Loops in SR_9

When extending SR_8 to SR_9, we use the modified version of SR_8 discussed in section 3.6. So Loop positioning commands can also be inserted freely in SR_9. And this modification has an important meaning in the three-dimensional case because it makes possible to generate different topological shapes. Fig.12 is a chain formed from a single Loop.

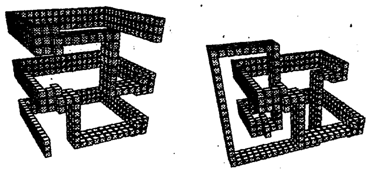

Figure 11: Complex Worm and Loop in SR_9.

This shape-construction technique can be possible by the positioning a daughter Loop with a specific command sequences in the mother Loop.

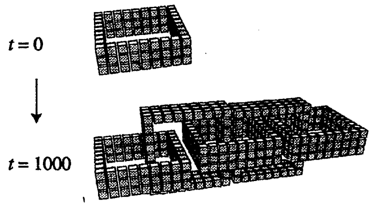

Figure 12: A chain formed from a single Loop in SR_9.

5 Conclusion

In this paper, we extend our two-dimensional self-reproducing reversible PCA SR_8 into a three-dimensional reversible PCA and show its various behaviors. The features of this three-dimensional reversible "turtle graphics" are derived from its self-inspective mechanism. The data as shapes and programs as command sequences are represented in the same manner.

The self-reproducing processes are hard to describe on a paper. They can be seen as QuickTime Movies at the following addresses via WWW.
SR_9: `http://kelp.ke.sys.hiroshima-u.ac.jp/projects/rca/sr3d/`

Ackowledgements: The authors thank Yukio Matsuda of Hiroshma University and Hiroyuki Aga of SONY Co,Ltd. for their support and comments.

References

[1] Bennett, C.H.: Logical reversibility of computation, *IBM J. Res. Dev.*, **17**, 6, (1973), 525–532.

[2] Bennett, C.H.: The thermodynamics of computation, *Int. J. Theoretical Physics*, **21**, 12, (1982), 905–940.

[3] Fredkin, E., Toffoli, T.: Conservative logic, Int. J. Theoretical Physics, **21**, 3/4, (1982), 219–253.

[4] Hertling, P., Embedding cellular automata into reversible ones, *Unconventional Models of Computation*, (eds., C.S.Claude, J. Casti, and M.J. Dinneen), Springer-Verlag, (1998), 243–256.

[5] Hori, T., Expanding the self-reproduction model SR_8 in a 2-dimensional reversible PCA into 3-dimensional one, Master thesis, Hiroshima University, (1998).

[6] Langton, C, G.: Self-Reproduction in Cellular Automata, Physica, **10D**, 1/2, (1984), 135–144.

[7] Margolus, N.: Physics-like models of computation, *Physica*, **10D**, 1/2, (1984), 81–95.

[8] Morita, K., and Harao, M.: Computation universality of one-dimensional reversible (injective) cellular automata, *Trans. IEICE Japan*, **E72**, 6, (1989) 758–762.

[9] Morita, K., Ueno, S: Computation-Universal Models of Two-Dimensional 16-state Reversible Cellular Automata, IEICE Trans. Inf. & Syst., **E75-D**, 1, (1992), 141–147.

[10] Morita, K.: Reversible Simulation of One-dimensional Irreversible Cellular Automata, *Theoretical Computer Science*, **148**, (1995), 157–163.

[11] Morita, K., and Imai, K., Self-reproduction in a reversible cellular space, *Theoret.Comput.Sci.*, **168**, 337–366 (1996).

[12] von Neumann, J.: *Theory of Self-reproducing Automata* (ed. A.W.Burks), The University of Illinois Press, Urbana, (1966).

[13] Toffoli, T.: Computation and Construction Universality of Reversible Cellular Automata, *Journal of Computer and System Sciences*, **15**, (1977), 213–231.