

1 次元 2 近傍可逆的セル・オートマトンについて

山形大学工学部 森田 憲一 (Kenichi Morita)

要旨 可逆的（単射）セル・オートマトンとは、その各々の状態が 1 時刻前の状態を高々 1 つしか持たないような、つまり、一意的に時間を逆戻りできるようなセル・オートマトン（CA）である。著者らはこれまでに、1 次元 3 近傍可逆的 CA が計算万能であること、つまり、任意の Turing 機械を模倣できることを示している。本稿では、1 次元 2 近傍可逆的 CA について考察する。ここでは、1 次元 Partitioned セル・オートマトン（PCA）と呼ぶ 1 次元 CA のあるサブクラスを考え、任意の 3 近傍可逆的 PCA が 2 近傍可逆的 PCA によって模倣できることを示す。これと 3 近傍可逆的 PCA が計算万能であるという結果から、2 近傍可逆的 PCA（および CA）の計算万能性が導かれる。

1. まえがき

可逆的なオートマトンのシステムとは、そのシステムの各時刻における全状態（計算状況）から直前の時刻の全状態が常に唯一に定まる、つまり逆方向の状態遷移が決定的であるようなものをいう。このような性質（論理的な可逆性）は一見非常に特殊な性質に見えるかもしれないが、実際には物理的あるいは化学的な可逆性と密接な関係があり、システムにおける不可避免的なエネルギー消費の問題を理論的に論じる上で重要な意味を持つことが知られている^{(2)・(4)}。また、エネルギーの問題から離れたとしても、可逆性という特別な性質を持つシステムがどのような振る舞いをするか（あるいは計算能力を持つか）という問題自体に興味深い点がある。（可逆的システムの研究の歴史は文献(3)を参照。）

本論文の研究対象は可逆的なセル・オートマトンであるが、その前に、可逆的 Turing 機械について述べておく。可逆的 Turing 機械を最初に定式化したのは、Bennett である⁽¹⁾。Bennett は任意の（一般には非可逆な）Turing 機械が、可逆的

な3テープ多記号Turing機械によって模倣できることを示し、可逆性という制約がTuring機械の計算能力に対する制限にはならないことを明らかにした。ここで注意すべきことは、Turing機械を可逆化するには、各ステップにおいてどの遷移規則が使われたかを特別の「履歴テープ」に逐一記録しておけばよく、ある意味で非常に簡単にできることである。しかしこれだけだと計算終了時に履歴情報が大量のゴミ情報として残されてしまい、問題解決を遅らせたにすぎない（ゴミ情報の単純な消去は非可逆的な過程となる）。Bennettの構成法で重要なのは、可逆性を保ったままでゴミ情報を最終的にきれいに消去できるという点である。また森田ら⁽⁵⁾は、Bennettの方法を改良して、任意のTuring機械から、それを模倣する1テープ2記号の可逆的Turing機械を構成する具体的方法を与えた。

セル・オートマトン(CA)は、同一の有限オートマトンが空間的に一様に配列、接続されたシステムであるが、このようなシステムの可逆性は、大域写像(状態から状態への写像)の単射性として定義される。可逆的CAの研究としては、大域写像の単射性と全射性の関係を研究したものもあるが、計算能力の研究にはToffoli⁽⁷⁾のものがある。Toffoliは k 次元の非可逆なCAが $k+1$ 次元の可逆的CAによって模倣できることを示し、これにより2次元可逆的CAが計算万能であるという結果を得ている。一方、森田ら⁽⁶⁾はこの結果を強め、1次元3近傍可逆的CAが計算万能であるという結果を与えている。後者の結果は、可逆的な1テープTuring機械を1次元3近傍可逆的CAで模倣することによって得られるが、その際、Partitionedセル・オートマトン(PCA)と呼ぶ、CAのあるサブクラスを新しく導入して証明がなされている。PCAは、1つのセルがいくつかの部分に分割されたようなCAで、大域写像の単射性(つまり可逆性)と局所写像の単射性とが等価になるようなシステムである。それゆえ、1テープ可逆的Turing機械を模倣する可逆的PCAが容易に設計できるという利点がある。

本稿では、2近傍のPCA(2PCA)について考察する。ここで考える2PCAは、着目するセルの次の時刻の状態が自身と左隣のセルの状態だけによって決まる、つまり、情報が1方向にしか流れないCAである。ここではまず、任意の3近傍PCA(3PCA)が2PCAによって模倣できること、また特に、任意の可逆的3PCAが可逆的2PCAによって模倣できることを示す。可逆的3PCAの計算万能性は既に得られているので、これにより、可逆的2PCA(および2近傍の可逆的CA)が計算万能であるという結果が導かれる。この模倣法では、模倣される側が静止状態付きのPCAである場合、模倣するPCAも静止状態付きのものとして構成される。

2. 諸定義と準備

決定性1次元セル・オートマトン (CA) とは、

$$A = (Z, Q, N, f, q_0)$$

によって定義されるシステムである。但し、各項は以下の通り。

- (1) Z はすべての整数の集合。
- (2) Q は各セルの内部状態の空でない有限集合。
- (3) $N = (n_1, n_2, \dots, n_k)$ は近傍 (各 n_i は Z の要素)。
- (4) $f : Q^k \rightarrow Q$ は各セルの状態遷移を定める局所写像。
- (5) $q_0 \in Q$ は静止状態 ($f(q_0, q_0, \dots, q_0) = q_0$ を満たす)。

CA の内で特に、 $N = (-1, 0, +1)$ であるような CA を 3 近傍セル・オートマトン (3CA)、 $N = (-1, 0)$ であるような CA を 2 近傍セル・オートマトン (2CA)、あるいは 1 方向セル・オートマトンと呼ぶ。

集合 Q 上の (A の) 状相とは、

$$c : Z \rightarrow Q$$

なる写像 c のことである。 Q 上のすべての状相の集合を $\text{Conf}(Q)$ で表す。つまり、

$$\text{Conf}(Q) = \{c \mid c : Z \rightarrow Q\}$$

である。 f によって定まる次のような写像 $F : \text{Conf}(Q) \rightarrow \text{Conf}(Q)$ を大域写像 (並列写像) と呼ぶ。

$$F(c)(i) = f(c(i+n_1), c(i+n_2), \dots, c(i+n_k))$$

A が 可逆的 であるとは、大域写像 F が単射であることをいう。

決定性1次元3近傍 Partitioned セル・オートマトン (3PCA) ⁽⁶⁾ とは、

$$P = (Z, (L, C, R), g, (l_0, c_0, r_0))$$

によって定義されるシステムである。但し、各項は以下の通り。

- (1) Z はすべての整数の集合。
- (2) L, C, R はそれぞれ、左状態、中央状態、右状態と呼ばれる、状態の空でない有限集合で、 $L \times C \times R$ の元が1個のセルの状態を表す。
- (3) $g : R \times C \times L \rightarrow L \times C \times R$ は各セルの状態遷移を定める局所写像。
- (4) $(l_0, c_0, r_0) \in L \times C \times R$ は静止状態。
($g(r_0, c_0, l_0) = (l_0, c_0, r_0)$ を満たす。)

3PCA は、直観的には図 2. 1 のように、1つのセルが3つの部分に分割されており、各セルの次の時刻の状態が、左隣のセルの右状態、そのセルの中央状

態、右隣のセルの左状態だけに依存して決まるようなCAである。

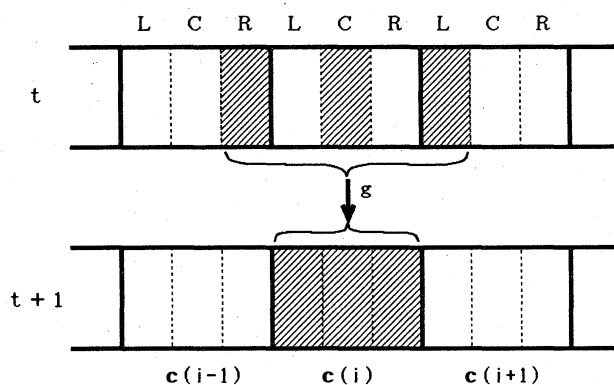


図2. 1 3PCAの状態遷移

集合 $L \times C \times R$ 上の (P) 状相とは、

$$c : Z \rightarrow L \times C \times R$$

なる写像 c のことである。 $L \times C \times R$ 上のすべての状相の集合をCAの場合と同様 $\text{Conf}(L \times C \times R)$ で表す。つまり、

$$\text{Conf}(L \times C \times R) = \{c \mid c : Z \rightarrow L \times C \times R\}$$

である。 $L \times C \times R$ に属する3項組から、左、中央、右、の要素を取り出す射影関数をそれぞれ、“LEFT”, “CENTER”, “RIGHT” とする。このとき、Pによって定まる次のような写像 $G : \text{Conf}(L \times C \times R) \rightarrow \text{Conf}(L \times C \times R)$ を大域写像と呼ぶ。

$$G(c)(i) = g(\text{RIGHT}(c(i-1)), \text{CENTER}(c(i)), \text{LEFT}(c(i+1)))$$

CAの場合とは異なり、PCAに対しては大域可逆性と局所可逆性の2種類の可逆性が定義できる。つまり、Pが大域可逆的であるとは大域写像Gが単射であることをいい、Pが局所可逆的であるとは局所写像gが単射であることをいう。CAの局所写像は $f : Q^k \rightarrow Q$ であるため、つまらない場合を除いて単射にはなり得ないが、3PCAの局所写像は $g : R \times C \times L \rightarrow L \times C \times R$ であるので単射になり得る。

3PCAの中で、集合Lの要素数が1個であるようなものを特に、決定性1次元2近傍(あるいは1方向) Partitioned セル・オートマトン(2PCA)と呼ぶが、上の3PCAの定式化中のLを省いて次のようなシステムとして定義することができる。

$$P = (Z, (C, R), g, (c_0, r_0))$$

但し、局所写像gは

$$g : R \times C \rightarrow C \times R$$

であり、大域写像 G は g から次のように定められる。

$$G(c)(i) = g(\text{RIGHT}(c(i-1)), \text{CENTER}(c(i)))$$

直観的には 2PCA は、図 2. 2 のように状態遷移をするシステムである。

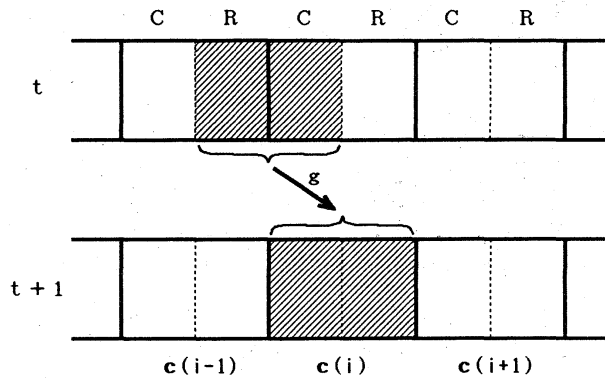


図 2. 2 2PCA の状態遷移

PCA や CA に関してはこれまでに以下のような結果が得られている。

【命題 2. 1】⁽⁶⁾ $P = (Z, (L, C, R), g, (l_0, c_0, r_0))$ を任意の 3PCA とする。 P は、大域可逆的であるとき、かつそのときに限り、局所可逆的である。

この命題は、2PCA に対しても成り立つ（2PCA は L の要素数が 1 であるような特別な 3PCA であるので）。この命題により、以下では PCA が大域可逆的あるいは局所可逆的であることを単に可逆的と呼ぶことにする。

【命題 2. 2】⁽⁶⁾ 任意の 3PCA, $P = (Z, (L, C, R), g, (l_0, c_0, r_0))$ に対し、大域写像がそれと一致するような（つまり、 $F = G$ となるような）3CA, $A = (Z, Q, (-1, 0, +1), f, q_0)$ が存在する。

これと同様の命題が 2PCA と 2CA の間にも成り立つ。証明も同様であるので結果だけを次に示す。

【命題 2. 3】 任意の 2PCA, $P = (Z, (C, R), g, (c_0, r_0))$ に対し、大域写像がそれと一致するような（つまり、 $F = G$ となるような）2CA, $A = (Z, Q, (-1, 0), f, q_0)$ が存在する。

一方、可逆的Turing機械に関しては以下のように、Bennettの結果およびそれを改良した結果が導かれている。

【命題 2. 4】⁽¹⁾ 任意の（一般に非可逆な）Turing機械 T に対し、それを模倣するような 3 テープ多記号可逆的Turing機械 R が存在する。

【命題 2. 5】⁽⁶⁾ 任意の（一般に非可逆な）Turing機械 T に対し、それを模倣するような 1 テープ 2 記号可逆的Turing機械 R が存在する。

可逆的Turing機械と可逆的PCAの関係については次の結果が得られている。

【命題 2. 6】⁽⁶⁾ 任意の 1 テープ 2 記号可逆的Turing機械 R に対し、それを模倣するような可逆的 3 PCA, P が存在する。

命題 2. 2, 2. 5, 2. 6 より、次の結果が得られる。

【命題 2. 7】⁽⁶⁾ 可逆的 3 PCA および、可逆的 3 CA は計算万能である。

3. 2 PCA による 3 PCA の模倣と、可逆的 2 PCA の計算万能性

【定理 3. 1】 任意の 3 PCA, P_3 に対し、その動作を模倣する 2 PCA, P_2 が存在する。

（証明） 与えられた P_3 を

$$P_3 = (Z, (L, C, R), g, (l_0, c_0, r_0))$$

とすると、それを模倣する 2 PCA, P_2 は次のようにして構成できる。

$$P_2 = (Z, (X, Y), g', (x_0, y_0))$$

$$X = (L \times C - \{(l_0, c_0)\}) \cup (L - \{l_0\}) \cup \{x_0\}$$

$$Y = (R - \{r_0\}) \cup (R \times C - \{(r_0, c_0)\}) \cup \{y_0\}$$

局所写像 g' は次のように定める。

(1) 任意の

$$r \in (R - \{r_0\})$$

$$l \in (L - \{l_0\})$$

$$c \in (C - \{c_0\})$$

に対して、 g' を次のように定める。

$$(1a) \quad g'(r, (l, c)) = (l, (r, c))$$

$$(1b) \quad g'(r, (l, c_0)) = (l, (r, c_0))$$

$$(1c) \quad g'(r, (l_0, c)) = (x_0, (r, c))$$

$$(1d) \quad g'(r, x_0) = (x_0, (r, c_0))$$

$$(1e) \quad g'(y_0, (l, c)) = (l, (r_0, c))$$

$$(1f) \quad g'(y_0, (l, c_0)) = (l, y_0)$$

$$(1g) \quad g'(y_0, (l_0, c)) = (x_0, (r_0, c))$$

$$(1h) \quad g'(y_0, x_0) = (x_0, y_0)$$

(2) 任意の

$$(r_1, c_1) \in (R \times C - \{(r_0, c_0)\})$$

$$l_1 \in (L - \{l_0\})$$

$$(l_2, c_2) \in (L \times C - \{(l_0, c_0)\})$$

$$r_2 \in (R - \{r_0\})$$

に対して、 g' を次のように定める。

$$(2a) \quad g(r_1, c_1, l_1) = (l_2, c_2, r_2) \text{ ならば}$$

$$g'((r_1, c_1), l_1) = ((l_2, c_2), r_2)$$

$$(2b) \quad g(r_1, c_1, l_1) = (l_2, c_2, r_0) \text{ ならば}$$

$$g'((r_1, c_1), l_1) = ((l_2, c_2), y_0)$$

$$(2c) \quad g(r_1, c_1, l_1) = (l_0, c_0, r_2) \text{ ならば}$$

$$g'((r_1, c_1), l_1) = (x_0, r_2)$$

$$(2d) \quad g(r_1, c_1, l_1) = (l_0, c_0, r_0) \text{ ならば}$$

$$g'((r_1, c_1), l_1) = (x_0, y_0)$$

$$(2e) \quad g(r_1, c_1, l_0) = (l_2, c_2, r_2) \text{ ならば}$$

$$g'((r_1, c_1), x_0) = ((l_2, c_2), r_2)$$

$$(2f) \quad g(r_1, c_1, l_0) = (l_2, c_2, r_0) \text{ ならば}$$

$$g'((r_1, c_1), x_0) = ((l_2, c_2), y_0)$$

$$(2g) \quad g(r_1, c_1, l_0) = (l_0, c_0, r_2) \text{ ならば}$$

$$g'((r_1, c_1), x_0) = (x_0, r_2)$$

(2h) $g(r_1, c_1, l_0) = (l_0, c_0, r_0)$ ならば

$$g'((r_1, c_1), x_0) = (x_0, y_0)$$

(2i) $g(r_0, c_0, l_1) = (l_2, c_2, r_2)$ ならば

$$g'(y_0, l_1) = ((l_2, c_2), r_2)$$

(2j) $g(r_0, c_0, l_1) = (l_2, c_2, r_0)$ ならば

$$g'(y_0, l_1) = ((l_2, c_2), y_0)$$

(2k) $g(r_0, c_0, l_1) = (l_0, c_0, r_2)$ ならば

$$g'(y_0, l_1) = (x_0, r_2)$$

(2l) $g(r_0, c_0, l_1) = (l_0, c_0, r_0)$ ならば

$$g'(y_0, l_1) = (x_0, y_0)$$

(3) 任意の

$$r \in (R - \{r_0\})$$

$$l \in (L - \{l_0\})$$

$$(r_1, c_1) \in (R \times C - \{(r_0, c_0)\})$$

$$(l_1, c_2) \in (L \times C - \{(l_0, c_0)\})$$

に対して、 g' を次のように定める。

$$(3a) \quad g'(r, l) = (l, r)$$

$$(3b) \quad g'((r_1, c_1), (l_1, c_2)) = ((l_1, c_2), (r_1, c_1))$$

P_2 は図3.1のように、 P_3 の1ステップを2ステップかかって模倣する。 P_2 は第1のステップで P_3 の各セルが状態遷移をするのに必要な情報を適当な場所に集め、第2のステップで実際の状態遷移を模倣する。 P_2 の各セルは、第1のステップの初め(図3.1の時刻 t)では、 X の部分によって P_3 の L と C の部分の状態を、 Y の部分によって R の部分の状態を保持する。第2のステップの初め(図3.1の時刻 $t+1$)では、 X の部分によって L の部分の状態を、 Y の部分によって R と C の部分の状態を保持する。また、 P_2 の静止状態は (x_0, y_0) であるが、 x_0 は (l_0, c_0) と l_0 に、 y_0 は r_0 と (r_0, c_0) に(多義的に)対応しており、各時点でそのいずれを表しているかは、その時点でどちらのステップを実行する段階にあるかで決まる。

上述の規則(1a)-(1h)は第1のステップ、(2a)-(2l)は第2のステップを実行するためのものである。それらの内で、(1b)-(1h), (2b)-(2l)は、静止状態を適切に扱うためのものである。従って、静止状態を持たないようなシステムではこれらの規則は不要である。また、規則(3a), (3b)は模倣には不必要なものなので右辺の値はどのように定めてもよいが、このようにしておくで後述の定理3.2で可逆性

の問題を扱うときに都合がよい。

l_0, c_0, r_0 に関係しないセルの状態遷移を P_2 が正しく行っていることは規則 (1a) と (2a) から容易にわかるが、そうでないセルに対しては場合分けをして個別に確かめる必要がある。場合の数は多いが、それぞれの確認は容易であるのでここでは省略する。 ■

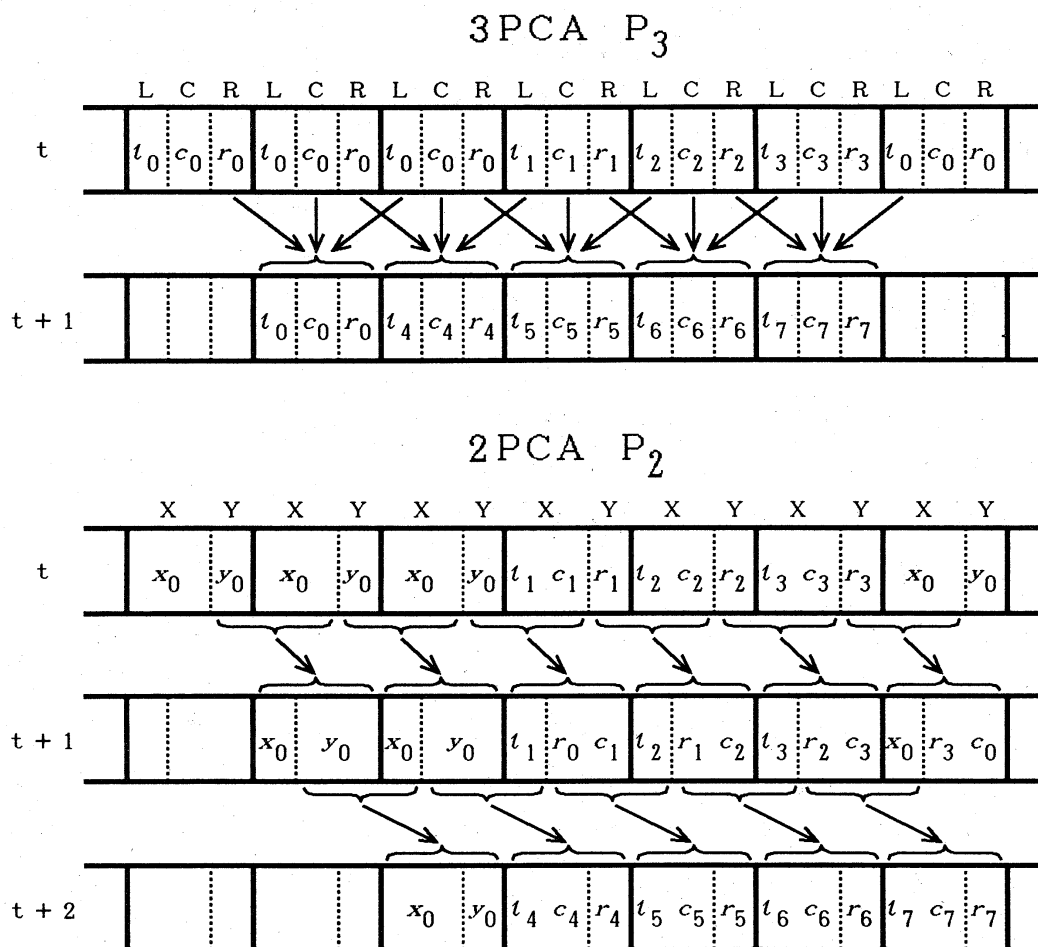


図 3. 1 2PCA, P_2 による 3PCA, P_3 の模倣

【定理 3. 2】 任意の可逆的 3PCA, P_3 に対し、その動作を模倣する可逆的 2PCA, P_2 が存在する。

(証明) 定理 3. 1 の証明における 2PCA の構成法は可逆性を保存する、つまり、 P_3 の局所写像 g が単射なら P_2 の局所写像 g' も単射になることが言える。これは構成法を丹念に調べることによって確かめられるが、その概略は次の通りである。まず規則 (1a) ~ (1h) は、基本的には、状態 l, c, r を保持する位置を P_2 の

セルの中で入れ換える操作をしているだけなので、これら規則の範囲内では g' の単射性が保証される。実際、(1a)~(1h)の中には右辺が一致するような規則は存在しない。規則(2a)~(2l)においても、 g が単射なら g' の単射性は崩れないことがわかる。例えば、(1h), (2d), (2h), (2l)の g' の右辺はすべて (x_0, y_0) で一致しているが、 g が単射なのでそれらの内の高々1つ（実際には(1h)の場合だけ）しか生じない。 g' の右辺が見かけ上一致する他の規則についても同様である。また、規則(3a), (3b)についてもそれらの右辺が一致する規則は他にないことが確かめられるので、これを加えても g' の単射性は崩れない。 ■

定理3.2と命題2.6より系3.3が、系3.3と命題2.3より系3.4が、それぞれ導かれる。

【系3.3】 可逆的2PCAは計算万能である。

【系3.4】 可逆的2CAは計算万能である。

参考文献

- (1) C.H.Bennett: "Logical reversibility of computation", IBM J. Res. & Dev., 6, 11, pp.525-532 (1973).
- (2) C.F.Bennett: "The thermodynamics of computation", Int.J.of Theoretical Physics, 21, 12, pp.905-940 (1982).
- (3) C.H.Bennett: "Notes on the history of reversible computation", IBM J. Res.& Dev., 32, 1, pp.16-23 (1988).
- (4) E.Fredkin and T.Toffoli: "Conservative logic", Int.J.of Theoretical Physics, 21, 3/4, pp.219-253 (1982).
- (5) K.Morita, A.Shirasaki, and Y.Gono: "A 1-tape 2-symbol reversible Turing machine", Trans. IEICE Japan, E72, No.3, pp.223-228 (1989).
- (6) K.Morita, and M.Harao: "Computation universality of one-dimensional reversible (injective) cellular automata", Trans. IEICE Japan, E72, No.6, pp.758-762 (1989).
- (7) T.Toffoli: "Computation and construction universality of reversible cellular automata", J.Comput. & Syst.Sci., 15, pp.213-231 (1977).