# Analysis of 90/150 Two Predecessor Nongroup Cellular Automata*

Sung-Jin Cho[1], Un-Sook Choi[2], Han-Doo Kim[3],
Yoon-Hee Hwang[4], and Jin-Gyoung Kim[5]

[1] Division of Mathematical Sciences, Pukyong National University
Busan 608-737, Korea
sjcho@pknu.ac.kr
[2] Department of Multimedia Engineering, Tongmyong University
Busan 626-847, Korea
choies@tu.ac.kr
[3] School of Computer Aided Science, Institute of Basic Science
Inje University, Gimhae 621-749, Korea
mathkhd@inje.ac.kr
[4] Department of Information Security, Graduate School, Pukyong National University
Busan 608-737, Korea
yhhwang@pknu.ac.kr
[5] Division of Mathematical Sciences, Pukyong National University
Busan 608-737, Korea
kimjg@pknu.ac.kr

**Abstract.** Many researchers have been studied synthesis method of 90/150 group CA. However, there is a lack of researches for synthesis method of 90/150 nongroup CA. In this paper we propose an algorithm for finding 90/150 Two Predecessor Cellular Automata. Using the proposed algorithm we analyze 90/150 two predecessor CA. Especially, we analyze 90/150 TPSACA and TPMACA which are useful to study hashing. Also we analyze two types of 90/150 two predecessor CA. One is two predecessor CA for the minimal polynomial whose type is of the form $xp(x)$ which is useful to study two predecessor CA whose depth is 1. Another is two predecessor CA for the minimal polynomial whose type is of the form $x(x+1)p(x)$ which is useful to study pseudorandom number generation based on 90/150 two predecessor CA, where $p(x)$ is some primitive polynomial.

## 1 Introduction

Cellular Automata(abbreviately, CA) have been introduced by Von Neumann and Ulam as models of self-organizing and self-reproducing behaviors ([1], [2]). A CA is a discrete time dynamical system, which consists of a uniform array of memories called cells. The states of cells in the array are updated according to

* This work was supported by grant No. (R01-2006-000-10260-0) from the Basic Research Program of the Korea Science and Engineering Foundation.

H. Umeo et al. (Eds): ACRI 2008, LNCS 5191, pp. 128–135, 2008.

a rule : the state of a cell at a given time depends only on its own state and the states of its nearby neighbors at the previous step. A CA is a necessity in many application areas such as test pattern generation, pseudorandom number generation, cryptography, error correcting codes and signature analysis([3] $\sim$ [10]). The analysis of the state-transition behavior of group CA was studied by many researchers ([6] $\sim$ [15]). Although the study of nonsingular linear machines has received considerable attention from researchers, the study of the class of machines with singular state-transition matrix has not received due attention. The state-transition matrix of group CA is nonsingular. But the state-transition matrix of nongroup CA is singular. Recently some interesting properties of nongroup CA have been employed in several applications([3], [5], [16] $\sim$ [20]). Especially, in ([3], [16]) they investigated a special class of nongroup CA denoted as D1*CA. Based on this investigation, D1*CA has been proposed as an ideal test machine which can be efficiently embedded in a finite state machine to enhance the testability of the synthesis design. Also in [5] they investigated 90/150 two predecessor CA whose minimal polynomial is of the form $x(x+1)p(x)$, where $p(x)$ is primitive. The use of these CA configurations simplifies the hardware implementations and avoids several precomputations to obtain the matrix associated to a quadratic function. Thus they studied several cases for different CA lengths. But they didn't show that there exists an $n$-cell 90/150 two predecessor CA for each $n \geq 6$. In this paper, using our algorithm for finding 90/150 two predecessor CA, we analyze 90/150 two predecessor CA. Especially, we analyze $n$-cell 90/150 TPSACA (whose minimal polynomial is $x^n$) and $n$-cell TPMACA (where minimal polynomial is $x^{n-1}(x+1)$) which are useful to study hashing [16]. Also we analyze two types of 90/150 two predecessor CA. One is two predecessor CA for the minimal polynomial whose type is of the form $xp(x)$ which is useful to study 90/150 two predecessor CA like D1*CA [3] whose depth is 1. The proposed $n$-cell 90/150 two predecessor CA has a maximum-length cycle whose length is $2^{n-1}-1$ which is larger than that of D1*CA. Another is 90/150 two predecessor CA for the minimal polynomial whose type is of the form $x(x+1)p(x)$ which is useful to study pseudorandom number generation based on 90/150 nongroup CA [5], where $p(x)$ is primitive.

## 2 CA Preliminaries

A CA consists of a number of interconnected cells arranged spatially in a regular manner [2], where the state-transitions of each cell depends on the states of its neighbors. If the next-state function of a cell is expressed in the form of a truth table, then the decimal equivalent of the output is conventionally called the rule number for the cell [2].

| Neighborhood state : | 111 | 110 | 101 | 100 | 011 | 010 | 001 | 000 | |
|---|---|---|---|---|---|---|---|---|---|
| Next state: | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | (rule 90) |
| Next state: | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | (rule 150) |

**Definition 2.1.** ([16], [18] $\sim$ [20])

i) *Group CA*: A CA is called a *group CA* if all the states in its state-transition diagram lie on cycles, otherwise it is referred to as a *non-group CA*.

ii) *Attractor*: A state having a self-loop is referred to as an *attractor*. An attractor can be viewed as a cyclic state with unit cycle length.

iii) *Depth*: The maximum number of state transitions required to reach the nearest cyclic state from any non-reachable state in the CA state-transition diagram is defined as the *depth* of the non-group CA.

iv) *Multiple-attractor CA(MACA)*: The non-group CA for which the state-transition diagram consists of a set of disjoint components forming (inverted) tree-like structures rooted at attractors are referred to as *multiple-attractor CA*. Single attractor CA(SACA) is a MACA whose the number of attractors is just one.

v) *TPMACA*: *TPMACA* is a MACA such that every reachable state in the state-transition diagram has only two predecessors. *TPSACA* is a SACA such that every reachable state in the state-transition diagram has only two predecessors. The minimal polynomial of an $n$-cell TPSACA is $x^n$.

## 3 An Algorithm for Finding 90/150 Two Predecessor CA

In this section we introduce an algorithm for finding 90/150 two predecessor CA.

Let $U$ be the following upper triangular matrix.

$$U = \begin{pmatrix} 1 & a_1 & * & \cdots & * & * & * \\ 0 & 1 & a_2 & \cdots & * & * & * \\ 0 & 0 & 1 & \cdots & * & * & * \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & a_{n-2} & * \\ 0 & 0 & 0 & \cdots & 0 & 1 & a_{n-1} \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix}$$

And let $T$ be the following 90/150 tridiagonal matrix.

$$T = \begin{pmatrix} d_1 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & d_2 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & d_3 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & d_{n-1} & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & d_n \end{pmatrix}$$

(Hereafter we write $T$ by $T =< d_1, d_2, \cdots, d_n >$, where $d_i \in \{0, 1\}$.)

Moreover, let $f(x) = x^n + c_{n-1}x^{n-1} + c_{n-2}x^{n-2} + \cdots + c_1x + c_0$, where $c_i \in GF(2)$. Then the following $n \times n$ matrix $C$ is said to be the *companion matrix* of $f(x)$.

$$C = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & c_0 \\ 1 & 0 & 0 & \cdots & 0 & c_1 \\ 0 & 1 & 0 & \cdots & 0 & c_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & c_{n-1} \end{pmatrix}$$

**Definition 3.1.** ([21]) For a given $n$-vector $x$ and $n \times n$ matrix $M$, let

$$K(M, x) = (x; Mx; M^2x; \cdots; M^{n-1}x)$$

We call $K(M, x)$ the Krylov matrix and $x$ is called a *seed vector*.

**Theorem 3.2.** ([15]) Let $T =< d_1, d_2, \cdots, d_n >$ and $C$ be the companion matrix of the characteristic polynomial of $T$. Let $U$ be the upper triangular matrix as the above form satisfying $TU = UC$. Then we obtain the following equation:

$$\begin{cases} d_1 = a_1 \\ d_2 = a_1 \oplus a_2 \\ d_3 = a_2 \oplus a_3 \\ \quad \vdots \\ d_{n-1} = a_{n-2} \oplus a_{n-1} \\ d_n = a_{n-1} \oplus c_{n-1} \end{cases} \tag{3.1}$$

Let $f(x)$ be a polynomial corresponding to a 90/150 two predecessor CA, then $f(x)$ is called a 90/150 two predecessor CA polynomial.

**Theorem 3.3.** ([15]) Let $B$ be the $n \times n$ matrix obtained by reducing the $n$ polynomials

$$x^{i-1} + x^{2i-1} + x^{2i} \ \ (mod\ f(x))\ (i = 1, 2, \cdots, n) \tag{3.2}$$

where $f(x)$ is a reducible polynomial. And let the set $\{v|Bv = (0, \cdots, 0, 1)^t\}$ be nonempty set, then the elements of $\{v|Bv = (0, \cdots, 0, 1)^t\}$ become seed vectors for the Krylov matrix, where $A^t$ is the transpose of $A$.

**Theorem 3.4.** Let the Krylov matrix in Theorem 3.3 have an $LU$ factorization. Then $f(x)$ in Theorem 3.3 is a 90/150 two predecessor CA polynomial.

The following algorithm is an algorithm for finding a 90/150 two predecessor CA for the given reducible polynomial.

**Algorithm.** SynthesisOf90/150TPNCA
Input : Polynomial $f(x)$
Output : 90/150 two predecessor CA
Step 1 : Make the matrix $B$ from (3.2).
Step 2 : Solve the equation $Bv = (0, \cdots, 0, 1)^t$. If there doesn't exist a solution, then STOP.
Step 3 : Construct a Krylov matrix $H = K(C^t, v)$ by the seed vector $v$ which is a solution of the equation in Step 2.

Step 4 : If $H$ doesn't have an $LU$ factorization, then STOP.
Step 5 : Compute the LU factorization $H = LU$.
Step 6 : Compute 90/50 two predecessor CA for $f(x)$ by the matrix $U$ using (3.1).

## 4 Analysis of 90/150 Two Predecessor CA

In this section we analyze 90/150 two predecessor CA.

**Theorem 4.1.** Let $\Delta_{2m}$ be the characteristic polynomial of

$$< d_1, d_2, \cdots, d_m, d_m, \cdots, d_2, d_1 >$$

Then the following equation hold.

$$\Delta_{i+1}\Delta_{2m-i-1} + \Delta_i\Delta_{2m-i-2} = \Delta_{i+2}\Delta_{2m-i-2} + \Delta_{i+1}\Delta_{2m-i-3},$$

where $i = 1, \cdots, 2m-2$, $\Delta_{-1} = 0$ and $\Delta_0 = 1$.

**Theorem 4.2.** Let $\Delta_{2m}$ be the characteristic polynomial of

$$< d_1, d_2, \cdots, d_m, d_m, \cdots, d_2, d_1 >$$

and let $f(x)$ be the characteristic polynomial of $< d_1, d_2, \cdots, d_m \oplus 1 >$. Then the following holds:

$$\Delta_{2m} = \{f(x)\}^2$$

**Theorem 4.3.** Let ${\mathbf{C_S}}^k =< d_1, \cdots, d_k >$ be a $k$-cell 90/150 TPSACA. Then the following hold:
(i) ${\mathbf{C_S}}^{2k} =< d_1, \cdots, d_k \oplus 1, d_k \oplus 1, \cdots, d_2, d_1 >$ is a $2k$-cell TPSACA with the minimal polynomial $x^{2k}$.
(ii) ${\mathbf{C_S}}^{2k+1} =< d_1, \cdots, d_k, 0, d_k, \cdots, d_1 >$ is a $(2k+1)$-cell TPSACA with the minimal polynomial $x^{2k+1}$.

**Theorem 4.4.** Let $N(T_m) = \{(a_1, a_2, \cdots, a_m)^t | a_1, a_2, \cdots, a_m \in \{0,1\}\}(:= [(a_1, \cdots, a_m)^t])$ be the null space of the state-transition matrix $T_n$ of an $n$-cell 90/150 TPSACA. Then the following hold:
(i) If $n = 2m(m \in \mathbf{N})$ and $N(T_m) = \{(a_1, a_2, \cdots, a_m)^t | a_1, a_2, \cdots, a_m \in \{0,1\}\}(:= [(a_1, a_2, \cdots, a_m)^t])$, then $N(T_n) = [(a_1, a_2, \cdots, a_m, a_m, \cdots, a_2, a_1)^t]$.
(ii) If $n = 2m+1(m \in \mathbf{N})$ and $N(T_m) = [(a_1, a_2, \cdots, a_m)^t]$, then $N(T_n) = [(a_1, a_2, \cdots, a_m, 0, a_m, \cdots, a_2, a_1)^t]$.

**Example 4.5.** Since $< 0, 0, 0 >$ is a 3-cell 90/150 TPSACA, $< 0, 0, 1, 1, 0, 0 >$ is a 6-cell 90/150 TPSACA and $< 0, 0, 0, 0, 0, 0, 0 >$ is a 7-cell 90/150 TPSACA.

**Theorem 4.6.** Let $\mathbf{C}_S^n =< d_1, \cdots, d_n >$ be an $n$-cell 90/150 TPSACA. Then $\mathbf{C}_M^{2n+1} =< d_1, \cdots, d_n, 1, d_n, \cdots, d_1 >$ is a $(2n+1)$-cell 90/150 TPMACA with the minimal polynomial $x^{2n}(x+1)$.

**Table 1.** TPSACA and TPMACA

| n | TPSACA | $N(T_S)$ | TPMACA | $N(T_M)$ | $N(T_M \oplus I)$ |
|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 0 | 1 |
| 2 | 11 | 11 | | | |
| 3 | 000 | 101 | 010 | 101 | 111 |
| 4 | 1001 | 1111 | | | |
| 5 | 11011 | 11011 | 11111 | 11011 | 10101 |
| 6 | 001100 | 101101 | | | |
| 7 | 0000000 | 1010101 | 0001000 | 1010101 | 1101011 |
| 8 | 10000001 | 11111111 | | | |
| 9 | 100101001 | 111101111 | 100111001 | 111101111 | 101111101 |
| 10 | 1101001011 | 1101111011 | | | |
| 11 | 11011011011 | 11011011011 | 11011111011 | 11011011011 | 10111111101 |
| 12 | 001101101100 | 101101101101 | | | |
| 13 | 0011000001100 | 1011010101101 | 0011001001100 | 1011010101101 | 1101011101011 |

**Remark.** For the case $n$ is even, there does not exist $n$-cell 90/150 TPMACA whose minimal polynomial is $f(x) = x^n + x^{n-1}$.

**Theorem 4.7.** Let $N(T_m) = [(a_1, \cdots, a_m)^t]$ be the null space of the state-transition matrix $T_m$ of an $m$-cell 90/150 TPSACA $\mathbf{C_S^m}$. Then the null space of the $(2m+1)$-cell 90/150 TPMACA $\mathbf{C_M^{2m+1}}$ derived from $\mathbf{C_S^m}$ is

$$N(T_{2m+1}) = [(a_1, \cdots, a_{m-1}, a_m, 0, a_m, a_{m-1}, \cdots, a_1)^t])$$

In Table 1, $N(T_S)$ means the null space of $n$-cell 90/150 TPSACA and $N(T_M)$ means the null space of $n$-cell 90/150 TPMACA. Also $N(T_M \oplus I)$ means the set of all attractors for each $n$-cell 90/150 TPMACA. 101 means $[(1, 0, 1)^t]$.

Chattopadhyay[22] presented an algorithm for finding MACA with all linear rules $(60, 90, 102, 150, 170, 204, 240)$, but in this paper we present a method which synthesize TPMACA using rule 90 and rule 150.

**Theorem 4.8.** Let $f(x) = xp(x)$, where $p(x)$ is a polynomial of degree $n-1$. Then there exists a primitive polynomial $p(x)$ such that $f(x)$ is the minimal polynomial corresponding to the $n$-cell 90/150 two predecessor CA.

**Theorem 4.9.** Let $f(x) = x(x+1)p(x)$, where $p(x)$ is a polynomial of degree $n - 2(n \geq 6)$. Then there exists a primitive polynomial $p(x)$ such that $f(x)$ is the minimal polynomial corresponding to the $n$-cell 90/150 two predecessor CA.

Table 2 shows that there exists an $n$-cell 90/150 two predecessor CA for the 90/150 two predecessor CA polynomial of the form $xp(x)$ ($p(x)$ is some primitive polynomial) for each $n \geq 4$. Also Table 3 shows that there exists an $n$-cell 90/150 TPMACA for the 90/150 TPMACA polynomial of the form $x(x+1)p(x)$ ($p(x)$ is some primitive polynomial) for each $n \geq 6$.

**Table 2.** 90/150 CA for $xp(x)$
(In this table, 320 stands for the polynomial $x^3 + x^2 + 1$.)

| $n$ | $p(x)$ | CA Configuration | $n$ | $p(x)$ | CA Configuration |
|---|---|---|---|---|---|
| 4 | 320 | 0111 | 13 | 12,10,9,8,6,2,0 | 1011101001000 |
| 5 | 430 | 00010 | 14 | 13,8,5,3,0 | 01100110101000 |
| 6 | 520 | 001001 | 15 | 14,11,9,7,0 | 100010001010000 |
| 7 | 65320 | 0011111 | 16 | 15,12,4,3,0 | 1000010010101010 |
| 8 | 740 | 00000011 | 17 | 16,15,12,10,0 | 11011110100010001 |
| 9 | 86520 | 000010001 | 18 | 17,3,0 | 100011101011110001 |
| 10 | 95320 | 0000100100 | 19 | 18,7,0 | 0001110111000101000 |
| 11 | 10,3,0 | 01011111110 | 20 | 19,10,9,3,0 | 01010100110000000010 |
| 12 | 11,2,0 | 011101000110 | 21 | 20,3,0 | 001001010110100100100 |

**Table 3.** 90/150 CA for $x(x + 1)p(x)$
(In this table, 210 stands for the polynomial $x^2 + x + 1$.)

| $n$ | $p(x)$ | CA Configuration | $n$ | $p(x)$ | CA Configuration |
|---|---|---|---|---|---|
| 4 | 210 | 1100 | 13 | 11,9,7,5,2,1,0 | 1111101110111 |
| 6 | 410 | 100110 | 14 | 12,10,2,1,0 | 01000110010010 |
| 7 | 53210 | 0100101 | 15 | 13,12,10,5,2,1,0 | 000101010001101 |
| 8 | 610 | 00001110 | 16 | 14,12,10,1,0 | 1100110011010011 |
| 9 | 73210 | 010000000 | 17 | 15,12,9,1,0 | 00000111110100111 |
| 10 | 85310 | 0001001001 | 18 | 16,14,12,1,0 | 101100100110001101 |
| 11 | 95410 | 10000110011 | 19 | 17,13,12,1,0 | 0100101010011011100 |
| 12 | 10,7,6,5,2,1,0 | 001111010101 | 20 | 18,17,12,10,9,1,0 | 00111100100000111000 |

## 5 Conclusion

In this paper we proposed an algorithm for finding 90/150 two predecessor CA. Using the proposed algorithm we analyzed 90/150 two predecessor CA. Especially, we analyzed 90/150 TPSACA and 90/150 TPMACA which are useful to study hashing. Also we analyzed two types of 90/150 two predecessor CA. One is two predecessor CA for the minimal polynomial whose type is of the form $xp(x)$. Another is two predecessor CA for the minimal polynomial whose type is of the form $x(x + 1)p(x)$.

## References

1. Von Neumann, J.: The theory of self-reproducing automata. In: Burks, A.W. (ed.). University of Illinois Press, Urban (1966)
2. Wolfram, S.: Statistical mechanics of cellular automata. Rev. Mod. Phys. 55, 601–644 (1983)
3. Chakraborty, S., Chowdhury, D.R., Chaudhuri, P.P.: Theory and application of nongroup cellular automata for synthesis of easily testable finite state machines. IEEE Trans. Computers 45(7), 769–781 (1996)
4. Chattopadhyay, S., Chaudhuri, P.P.: Theory and application of nongroup cellular automata in pattern classification, IEEE Trans. Computers, communicated

5. De la Guia Martinez, D., Peinado Dominguez, A.: Pseudorandom number generation based on nongroup cellular automata. In: Security Technology, 1999, Proceedings, IEEE 33rd Annual 1999 International Carnahan Conference, vol. 45, pp. 370–376 (1999)
6. Das, A.K., Chaudhuri, P.P.: Vector space theoretic analysis of additive cellular automata and its application for pseudo-exhaustive test pattern generation. IEEE Trans. Comput. 42, 340–352 (1993)
7. Serra, M., Slater, T., Muzio, J.C., Miller, D.M.: The analysis of one dimensional linear cellular automata and their aliasing properties. IEEE Trans Computer-Aided Design 9, 767–778 (1990)
8. Hortensius, P.D., McLeod, R.D., Card, H.C.: Parallel random number generation for VLSI systems using cellular automata. IEEE Trans. Computers 38, 1466–1473 (1989)
9. Hortensius, P.D., McLeod, R.D., Miller, D.M., Card, H.C.: Cellular automata based pseudorandom number generations for built-in self test. IEEE Trans. on CAD of Integrated Circuits ans Systems 8, 842–859 (1989)
10. Tsalides, P., York, T.A., Thanailakis, A.: Pseudorandom number generators for systems based on linear cellular automata. IEE Proc(Part E) Computers Digital Techniques 138, 241–249 (1991)
11. Pries, W., Thanailakis, A., Card, H.C.: Group properties of cellular automata and VLSI applications. IEEE Trans. Computers 35, 1013–1024 (1986)
12. Nandi, S., Kar, B.K., Chaudhuri, P.P.: Theory and application of cellular automata in cryptography. IEEE Trans. Computers 43, 1346–1357 (1994)
13. Cattell, K., Muzio, J.C.: Synthesis of one-dimensional linear hybrid cellular automata. IEEE Trans. on Computer Aided Design of Circuits and Systems 15-3, 325–335 (1996)
14. Cho, S.J., Choi, U.S., Hwang, Y.H., Pyo, Y.S., Kim, H.D., Kim, K.S., Heo, S.H.: Computing phase shifts of maximum-length 90/150 cellular automata sequences. In: Sloot, P.M.A., Chopard, B., Hoekstra, A.G. (eds.) ACRI 2004. LNCS, vol. 3305, pp. 31–39. Springer, Heidelberg (2004)
15. Cho, S.J., Choi, U.S., Kim, H.D., Hwang, Y.H., Kim, J.G., Heo, S.H.: New synthesis of one-dimensional 90/150 linear hybrid group cellular automata. IEEE Trans. Comput-Aided Des. Integr. Circuits Syst. 26(9), 1720–1724 (2007)
16. Chaudhuri, P.P., Chowdhury, D.R., Nandy, S., Chattopadhyay, C.: Additive cellular automata theory and applications, vol. 1. IEEE Computer Society Press, California (1997)
17. Cho, S.J., Choi, U.S., Hwang, Y.H., Kim, H.D.: Analysis of hybrid group cellular automata. In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 222–231. Springer, Heidelberg (2006)
18. Cho, S.J., Choi, U.S., Kim, H.D.: Analysis of complemented CA derived from a linear TPMACA. Computers Math. Applic. 45, 689–698 (2003)
19. Cho, S.J., Choi, U.S., Kim, H.D.: Behavior of complemented CA whose complement vector is acyclic in a linear TPMACA. Math. Comput. Modelling 36, 979–986 (2002)
20. Cho, S.J., Choi, U.S., Hwang, Y.H., Kim, H.D., Choi, H.H.: Behaviors of single attractor cellular automata over Galois Field $GF(2^p)$. In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 232–237. Springer, Heidelberg (2006)
21. Horn, R.A., Johnson, C.R.: Matrix Analysis. Cambridge University Press, Cambridge (1985)
22. Chattopadhyay, S.: Some studies on theory and application of additive cellular automata, PhD thesis, I.I.T., Kharagpur, India (1995)