

CLT를 활용한 비선형 CA의 분석

권민정¹ · 조성진^{1*} · 김한두² · 최연숙³ · 이규진¹ · 공길탁¹

Analysis of Nonlinear CA Using CLT

Min-jeong Kwon¹ · Sung-jin Cho^{1*} · Han-doo Kim² · Un-sook Choi³ · Kue-jin Lee¹ · Gil-tak Kong¹

¹Department of Applied Mathematics, Pukyong National University, Busan 48513, Korea

²Institute of Basic Sciences and Department of Applied Mathematics, Inje University, Gimhae 50834, Korea

³Department of Information and Communications Engineering, Tongmyong University, Busan 48520, Korea

요 약

선형/가산 CA에서 끌개는 패턴인식, 패턴 분류 또는 연관기억장치 디자인, 질문처리 등과 같은 분야에서 활용되고 있기 때문에 이를 어떻게 찾을 수 있는지는 주요한 연구 대상이 되어 왔다. 그러나 복잡한 현실을 모델링하고 그 해결방안을 모색하기 위해서는 비선형 CA의 도입이 불가피하게 되었다. 본 논문에서는 RMT를 개선한 CLT를 도입하여 비선형 CA의 끌개와 도달불가능 상태를 효율적으로 찾는 방법을 제시한다.

ABSTRACT

Method for finding the attractors is the important object to investigate in the linear/additive CA because it is a primary interest in applications like pattern recognition, pattern classification, design of associative memory and query processing etc. But the research has been so far mostly concentrated around linear/additive CA and it is not enough to modelize the complex real life problem. So nonlinear CA is demanded to devise effective models of the problem and solutions around CA model. In this paper we introduce CLT as an upgraded version of RMT and provide the process for finding the attractors and nonreachable states effectively through the CLT.

키워드 : CA, 비선형 CA, NBCA, CLT, 끌개, 도달불가능 상태

Key word : CA, nonlinear CA, NBCA, CLT, attractor, nonreachable state

Received 27 May 2015, Revised 22 June 2015, Accepted 06 July 2015

* Corresponding Author Sung-Jin Cho(E-mail:sjcho@pknu.ac.kr, Tel:+82-51-629-5527)

Department of Applied Mathematics, Pukyong National University, Busan 48513, Korea

Open Access <http://dx.doi.org/10.6109/jkiice.2015.19.12.2968>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서론

Cellular Automata(CA)는 Von Neumann[1]에 의해 처음으로 연구되기 시작하여 그 구조에 대한 연구[2-4]가 활발하게 이루어졌다. Wolfram[5-7]은 복잡한 현실을 모델링하기 위해 1차원 CA에 대해 자세히 연구함으로써 3개의 이웃을 갖는 이산적인 개체로서의 CA를 수학적으로 구조화하였다. 이와 같은 대수적인 구조에 의해 수학적으로 CA를 다루는 것이 용이해졌고, 더욱 체계적으로 그 성질을 파악할 수 있게 되었다. CA의 상태 연구를 통해 상태전이과정을 거쳐 끝개로 접근하는 CA가 존재함을 확인하였다[8]. 이러한 성질을 갖는 CA는 패턴 인식, 패턴 분류 또는 연관기억장치 디자인, 질문처리기와 같은 분야에서 활용되고 있다[9-11]. Chaudhuri[10], Ganguly[11], Cho 등[12]은 비그룹 CA의 끝개를 어떻게 구성하고 찾을 수 있는지에 대해 연구하였다. 그러나 다양한 분야의 복잡한 현실을 모델링하고 그 해결방안을 모색하기 위해서는 선형/가산 CA에 제한된 연구에서 벗어나 비선형 CA를 도입하는 것이 불가피하게 되었다[13-15].

본 논문에서는 Das[13]가 제시한 Rule Min Term(RMT)를 개선한 Class Location Term(CLT)을 도입하고 CLT 표를 이용함으로써 비선형 CA의 끝개와 도달 불가능 상태를 효율적으로 찾는 방법을 제시한다.

II. 배경지식

CA는 이진 n 차원 벡터 공간이고 이산시간에 따라 상태가 바뀐다. 하나의 CA는 n 개의 셀로 이루어져 있으며 각 셀의 다음 상태는 식 (1)과 같이 그 셀의 현재 상태와 그 셀의 이웃 상태의 영향을 받아 결정된다. 0 또는 1의 상태를 갖는 3-이웃 1 차원 CA에서 셀의 다음 상태는 다음과 같이 결정된다.

$$s_i^{t+1} = R_i(s_{i-1}^t, s_i^t, s_{i+1}^t) \quad (1)$$

여기서 R_i 는 i 번째 셀의 상태전이 함수이고, s_i^t 은 시간 t 에서 i 번째 셀의 상태, s_{i-1}^t 은 i 번째 셀의 왼쪽 이웃 상태, s_{i+1}^t 은 i 번째 셀의 오른쪽 이웃 상태를 나타낸다. 즉 시간 t 에서 n 셀 CA의 전체 상태 S^t 는 시간 t 에서 각 상

태 $s_i^t (i=1, 2, \dots, n)$ 의 집합이며 $S^t = (s_1^t, s_2^t, \dots, s_n^t)$ 로 나타낸다. 이 때, 가장 왼쪽에 있는 셀의 왼쪽 이웃과 가장 오른쪽에 있는 셀의 오른쪽 이웃을 모두 0이라 둔다. 즉 $s_0^t = 0, s_{n+1}^t = 0$ 이다. 이런 CA를 null boundary CA(NBCA)라 하고 본 논문에서는 NBCA를 다룬다.

정의 2.1. n 셀 CA의 각 셀에 적용되는 함수 R_i 를 규칙이라 하고 R_i 들을 각 셀에 적용되는 순서에 따라 벡터 $R = \langle R_1, R_2, \dots, R_n \rangle$ 으로 표현한다. 이때 각 셀에 적용된 R_i 가 모두 같으면 uniform CA, 그렇지 않으면 hybrid CA라 한다. 또한 R 의 모든 $R_i (i=1, 2, \dots, n)$ 가 XOR 연산만을 사용하면 선형 CA라 하고 그렇지 않으면 비선형 CA라 한다.

Table. 1 90, 150 rule

state	111	110	101	100	011	010	001	000	rule
s_i^{t+1}	1	0	0	1	0	1	1	0	150
s_i^{t+1}	0	1	1	0	1	0	0	1	105

예제 2.2. $R = \langle 105, 177, 171, 75 \rangle$ 에 대하여 상태 $S^t = (0111)$ 의 다음 상태는 $S^{t+1} = (0011)$ 이다.

정의 2.3. n 셀 CA의 시간 t 에서의 상태 S^t 는 R 에 의해 다음 상태 S^{t+1} 로 바뀌고 식 (2)와 같이 나타낸다.

$$R(S^t) = S^{t+1} \quad (2)$$

이때 주어진 S^{t+1} 에 대하여 S^t 가 존재하는 경우 S^{t+1} 은 도달가능상태(reachable state)라 하고 S^t 가 존재하지 않으면 도달불가능 상태(nonreachable state)라 한다. 특별히 $S^{t+1} = S^t$ 을 만족하는 S^t 를 끝개(attractor)라 한다.

III. 비선형 CA 규칙의 표현

정의 3.1. n 셀로 이루어진 CA의 상태 S^t 를 가장 왼쪽부터 시작하여 한 비트씩 오른쪽으로 옮겨가며 2개 비트씩 읽어서 0부터 3까지의 십진수 $n-1$ 개로 표현하

고, $n-1$ 번째 숫자를 한 번 더 사용하여 n 개의 숫자로 나타낸 것을 CLT(Class Location Term) $C^t = (c_1^t c_2^t \cdots c_{n-1}^t c_n^t)$ 라 한다. S^t 로부터 얻은 C^t 를 S^{t+1} 로 대응시키는 전이규칙을 R_C 라 하고, S^t 를 C^t 로 대응시키는 함수 T 를 식 (3)과 같이 정의한다.

$$T : S^t \rightarrow C^t$$

$$(s_1^t s_2^t \cdots s_n^t) \mapsto (c_1^t c_2^t \cdots c_n^t), \quad (3)$$

단, $\begin{cases} c_i^t = s_i^t \cdot 2 + s_{i+1}^t \cdot 1, & i = 1, 2, \dots, n-1 \\ c_n^t = c_{n-1}^t \end{cases}$

그림 1은 함수 T 를 이용하여 4셀 CA의 상태 S^t 를 C^t 로 표현하는 과정을 나타낸 것이다. 예를 들어 $S^t = (1101)$ 에 대하여 $C^t = (3211)$ 이다.

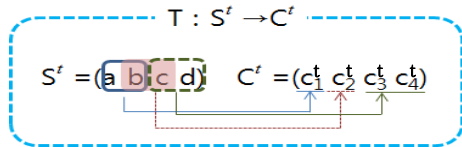


Fig. 1 Relation between 4-cell CA state S^t and CLT C^t

$C^t = (c_1^t c_2^t \cdots c_{n-1}^t c_n^t)$ 에서 $i(i=2, \dots, n-1)$ 번째 셀의 클래스 번호는 c_{i-1}^t , 위치 번호는 c_i^t 이다. 첫 번째 셀과 n 번째 셀의 c_1^t, c_n^t 를 클래스 번호라 두며, 위치 번호는 고려하지 않는다. 표 2는 CLT에 대한 클래스 번호, 위치 번호의 위치를 나타낸다.

Table. 2 The places of class and location number

3	2	1	0	Class number
3	2	1	0	Location number
⋮	⋮	⋮	⋮	⋮
3	2	1	0	Class number

NBCA에서는 가장 왼쪽 이웃을 0으로 두기 때문에 규칙 $R = \langle R_1, R_2, \dots, R_n \rangle$ 에 대하여 R_1 에서는 실제로 000, 001, 010, 011 만 사용되므로 첫 번째 비트를 제외한 나머지 두개의 비트만 사용하여 다음 상태를 결정

할 수 있다. 따라서 R_C 의 첫 번째 행에서 c_1^t 의 값은 0, 1, 2, 3 중 하나가 된다. 마찬가지로 NBCA에서는 가장 오른쪽 이웃을 0으로 두기 때문에 R_n 에서는 실제로 000, 010, 100, 110에 적용되는 규칙만 사용되므로 마지막 비트를 제외한 두개의 비트만 사용하여 다음 상태를 결정할 수 있다. 따라서 이 값은 $n-1$ 번째의 CLT와 같고 R_C 의 마지막 행에서 c_n^t 의 값도 0, 1, 2, 3 중 하나이다.

RMT[13]에서는 i 번째 비트의 RMT k_i 로부터 k_{i+1} 을 구하기 위해 mod8 처리과정을 거쳐야 하지만 CLT를 이용하여 다음 상태를 구하는 과정에서는 두 비트만 읽기 때문에 mod4 처리과정을 거친다. 따라서 CLT를 이용하면 끝개와 도달불가능 상태를 효과적으로 찾을 수 있다. 표 3은 CA 규칙 R 과 CLT에 적용할 수 있도록 만든 R_C 를 나타낸 것이다.

Table. 3 Table of R and R_C

rule	R								R_C							
	111	110	101	100	011	010	001	000	3	2	1	0				
R_1	×	×	×	×	D	C	B	A	D	C	B	A				
R_2	d_3	d_2	c_1	c_0	b_3	b_2	a_1	a_0	d_3	d_2	c_1	c_0	b_3	b_2	a_1	a_0
R_3	d_3'	d_2'	c_1'	c_0'	b_3'	b_2'	a_1'	a_0'	d_3'	d_2'	c_1'	c_0'	b_3'	b_2'	a_1'	a_0'
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
R_n	D'	×	C'	×	B'	×	A'	×	D'	C'	B'	A'				

상태 $S^t = (1101)$ 에 대하여 $C^t = T(S^t) = (3211)$ 이고 표 3의 R_C 로부터 다음 상태 S^{t+1} 을 찾는 과정은 다음과 같다.

먼저 $c_1^t = 3$ 으로부터 $s_1^{t+1} = D$ 이다. $c_1^t = 3$ 과 $c_2^t = 2$ 로부터 클래스 번호 3과 위치 번호 2를 얻고 그 위치에서 $s_2^{t+1} = d_2$ 이다. $c_2^t = 2$ 와 $c_3^t = 1$ 로부터 클래스 번호 2에서 위치 번호 1을 얻고 $s_3^{t+1} = c_1'$ 이다. $c_3^t = 1$ 이고 이 값은 c_4^t 와 같으므로 $s_4^{t+1} = B'$ 이다. 따라서 CA의 상태 $S^t = (1101)$ 의 CLT는 $C^t = (3211)$ 이고 R_C 에 의해 $S^{t+1} = (Dd_2c_1'B')$ 가 된다.

그림 2는 R_C 에서 다음 상태를 얻는 과정을 나타낸다. 이 과정을 통해 얻은 다음 상태는 RMT (3652)에 R 을 적용한 결과와 같다.

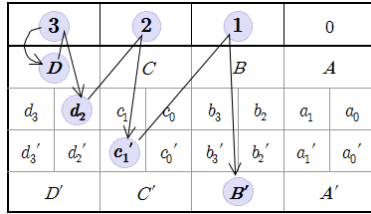


Fig. 2 The process of obtaining the next state from R_C

정리 3.2. S^t 의 다음 상태 S^{t+1} 는 함수 T 와 R_C 에 대하여 다음을 만족한다.

$$S^{t+1} = (R_C \circ T)(S^t) \quad (4)$$

증명. $S^t = (s_1^t s_2^t \dots s_n^t)$ 에 대해 S^{t+1} 의 RMT는 $(2s_1^t + s_2^t, 4s_1^t + 2s_2^t + s_3^t, \dots, 4s_{k-1}^t + 2s_k^t + s_{k+1}^t, \dots, 4s_{n-2}^t + 2s_{n-1}^t + s_n^t, 4s_{n-1}^t + 2s_n^t)$ 이고 CLT는 $(2s_1^t + s_2^t, 2s_2^t + s_3^t, \dots, 2s_k^t + s_{k+1}^t, \dots, 2s_{n-1}^t + s_n^t, 2s_{n-1}^t + s_n^t)$ 이다.

(a) 첫 번째 비트의 경우, S^{t+1} 의 RMT와 CLT는 모두 $2s_1^t + s_2^t$ 로 같기 때문에 R 표와 R_C 에서 같은 위치에 대응된다. 따라서 함숫값이 같다.

(b) $k (k = 2, 3, \dots, n-1)$ 번째 비트의 경우, $s_{k-1}^t = 0$ 이면 $s_k^{t+1} = c_k^{t+1}$ 이므로 RMT와 CLT의 값은 같다. 즉 R_C 는 R 표와 같은 위치의 값에 대응되기 때문에 s_k^{t+1} 의 값은 같다. $s_{k-1}^t = 1$ 이면 s_k^{t+1} 의 값은 4, 5, 6, 7 중 하나를 갖고 c_k^{t+1} 의 값은 0, 1, 2, 3 중 하나이다. 그러나 $s_{k-1}^t = 1$ 인 경우 c_k^{t+1} 의 값은 2 또는 3이고 이로부터 c_k^{t+1} 의 클래스 번호가 2 또는 3임을 알 수 있다. 클래스 번호 2 또는 3인 경우 위치 번호 0, 1, 2, 3의 위치는 RMT 4, 5, 6, 7의 위치와 같다. 즉, R_C 는 R 표와 같은 위치의 값에 대응되기 때문에 s_k^{t+1} 의 값은 같다.

(c) n 번째 비트의 경우, RMT는 $4s_{n-1}^t + 2s_n^t$ 이므로 0, 2, 4, 6 중 하나의 값이 되고 CLT는 $2s_{n-1}^t + s_n^t$ 이므로 0, 1, 2, 3 중 하나의 값이 된다. 이 경우 그에 대응되는 위치는 각각 다르지만 R_C 의 값과 R 표의 값은 같기 때문에 s_k^{t+1} 의 값은 같다.

(a), (b), (c)에 의하여 $S^{t+1} = (R_C \circ T)(S^t)$ 이다. □

Table. 4 $R_C = \langle 105, 177, 171, 75 \rangle$

		3	2	1	0	
3	1	0	0	0	0	1
2	1	0	1	1	0	0
1	1	0	1	0	1	0
0	1	0	0	0	1	1

예제 3.3. 4셀 CA의 규칙 $\langle 105, 177, 171, 75 \rangle$ 를 R_C 로 표현하면 표4와 같다.

CA 상태 $S^t = (0001)$ 의 CLT는 $C^t = (0011)$ 이고 이것은 $c_1^t = 0$ 이므로 클래스 번호 0의 값 1, $c_2^t = 0$, $c_3^t = 0$ 이므로 클래스 번호 0에서 위치 번호 0의 값 1, $c_4^t = 1$ 이므로 클래스 번호 0에서 위치 번호 1의 값 1, $c_5^t = 1$ 이므로 $c_4 = 1$ 이고 클래스 번호 1의 값 0에 대응되므로 $S^{t+1} = (1110)$ 이다.

IV. 비선형 CA의 끝개 찾는 방법

S^t 에 대한 다음 상태 S^{t+1} 이 현재 상태 S^t 와 같게 되는 규칙은 $R = \langle 204, 204, \dots, 204 \rangle$ 이고 R 에 대한 R_C 를 I_C 라 하면 I_C 는 표 5와 같다.

Table. 5 $I_C = \langle 204, 204, \dots, 204 \rangle$

		3	2	1	0	
3	1	1	0	0	0	0
2	1	1	0	0	1	1
1	1	1	0	0	1	1
0	1	1	0	0	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	1	0	0	1	1

정리 4.1. 전이규칙이 R_C 인 CA의 끝개는 식 (5)를 만족하는 C^t 에 대하여 $T^{-1}(C^t)$ 이다.

$$(R_C + I_C)(C^t) = 0 \quad (5)$$

증명. S^t 를 끝개라 하면 정리 3.2에 의하여

$$(R_C \circ T)(S^t) = S^t \quad (6)$$

$$S^t = (I_C \circ T)(S^t) \quad (7)$$

이다. 식 (6), (7)에 의하여 $R_C(T(S^t)) = I_C(T(S^t))$ 이고 $R_C(C^t) = I_C(C^t)$ 이므로

$$(R_C + I_C)(C^t) = O$$

이다. 따라서 S^t 는 $(R_C + I_C)(C^t) = O$ 를 만족한다. □

예제 4.2. $R_C = \langle 8, 112, 44, 68 \rangle$ 와 이에 대한 $R_C + I_C$ 는 표 6과 같다. 정리 4.1의 식 (5)에 의해 표 6(b)에서 $O = (0000)$ 이 만들어지는 CLT를 찾기 위해 첫 번째 행을 살펴보면 0의 값을 가지고 있는 c_1^t 는 0,1,3이다.

Table. 6 $R_C = \langle 8, 112, 44, 68 \rangle$ and $R_C + I_C$

3	2	1	0		3	2	1	0
1	0	0	0	→	0	1	0	0
0	1	1	0		0	0	0	0
0	0	1	0		1	1	0	0
1	0	1	0		1	1	0	0
					0	0	0	0

(a) $R_C = \langle 8, 112, 44, 68 \rangle$ (b) $R_C + I_C$

$c_1^t = 3$ 인 경우를 살펴보면,

두 번째 행 : 클래스 번호 3에서 위치 번호 2,

세 번째 행 : 클래스 번호 2에서 위치 번호 0,

네 번째 행 : 클래스 번호 0일 때, $C^t = (3200)$ 이므로 $S^t = (1100)$ 이다. 나머지 c_1^t 에 대해서도 같은 방법으로 클래스 번호와 해당 클래스에 대응되는 위치 번호를 이용하여 0이 되는 부분을 찾으면 된다. 그림 3은 나머지 끝개를 찾는 과정을 나타낸다.

모든 C^t 에 R_C 를 적용하여 결정되는 다음 상태에 대해서는 항상 직전자가 존재한다. 즉 모든 C^t 에 R_C 를 적용한 결과에서 나타나지 않는 상태는 도달불가능 상태이다. 따라서 모든 C^t 를 R_C 에 의해 상태전이시켰을 때 나타나는 S^{t+1} 을 전체 상태집합에서 제외하면 도달불가능 상태를 찾을 수 있다. 도달불가능 상태는 다음 과정을 통해 효율적으로 찾을 수 있다.

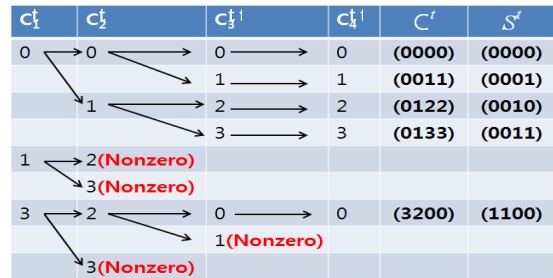


Fig. 3 The process of finding attractors from $R_C + I_C$

Algorithm. Finding Nonreachable States

- Step 1. R_C 표에서 $S^{t+1} = (00*...*)$, $S^{t+1} = (01*...*)$, $S^{t+1} = (10*...*)$, $S^{t+1} = (11*...*)$ 가 나타나도록 하는 C^t 를 찾아서 각 경우에 대해 c_1^t 을 결정한다.
- Step 2. 각 c_1^t 에 의하여 선택될 수 있는 C^t 를 결정한다.
- Step 3. $C^t = (c_1^t c_2^t *...*)$ 에 의하여 생성되는 S^{t+1} 을 나열한다.
- Step 4. 각 S^{t+1} 에서 나타나지 않는 상태를 찾는다.

예제 4.3. 예제 3.3의 4셀 CA의 도달불가능 상태를 R_C 를 이용하여 찾아보자.

i) $S^{t+1} = (00**)$: $C^t = (12**)$ 또는 $C^t = (13**)$ 일 때의 결과이므로 두 경우에 대하여 만들어질 수 다음 상태를 점검한다. $C^t = (12**)$ 에서 마지막 2 비트가 될 수 있는 것을 모두 찾으면, 클래스 번호 2에서 위치 번호 0, 클래스 번호 0이므로 $S^{t+1} = (**01)$, 클래스 번호 2에서 위치 번호 1, 클래스 번호 1이므로 $S^{t+1} = (**10)$ 이다. $C^t = (13**)$ 에서 마지막 2 비트를 찾으면, 클래스 번호 3에서 위치 번호 2, 클래스 번호 2이므로 $S^{t+1} = (**00)$, 클래스 번호 3에서 위치 번호 3, 클래스 번호 3이므로 $S^{t+1} = (**11)$ 이다. 따라서 첫 번째, 두 번째 비트의 (00**)에 대해 3번째, 4번째 비트에서는 (00), (01), (10), (11)이 모두 나타나므로 (00**)인 도달불가능 상태는 존재하지 않는다.

ii) $S^{t+1} = (01**)$ 의 경우, $C^t = (20**)$ 또는 $C^t = (21**)$ 일 때의 결과이므로 두 경우에 대하여 만들어질 수 있는 다음 상태를 점검한다. $C^t = (20**)$ 에서 마지막 2 비트를 찾으면, 클래스 번호 0에서 위치 번호 0, 클래스 번호 0이므로 $S^{t+1} = (**11)$, 클래스 번호 0에

서 위치 번호 1, 클래스 번호 1이므로 $S^{t+1} = (**10)$ 이고, $C^t = (21**)$ 에서 마지막 2 비트를 찾으면, 클래스 번호 1에서 위치 번호 2, 클래스 번호 2이므로 $S^{t+1} = (**00)$, 클래스 번호 1에서 위치 번호 3, 클래스 번호 3이므로 $S^{t+1} = (**11)$ 이다. 따라서 첫 번째, 두 번째 비트의 (01**)에 대해 3번째, 4번째 비트에서는 (00), (10), (11)만 나타나므로 (0101)은 도달불가능 상태이다.

iii) (10**), (11**)에 대해서도 같은 방법을 적용하면 (0101), (1101)이 도달불가능 상태임을 알 수 있다.

그림 4는 주어진 알고리즘에 의해 도달불가능 상태를 찾는 과정을 나타낸다.

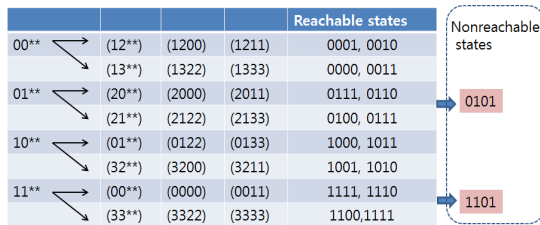


Fig. 4 The process of finding nonreachable states

V. 결론

본 논문에서는 비선형 CA의 규칙을 표현하는 새로운 방법인 CLT를 제안하였고 R_C 와 I_C 를 이용하여 비선형 CA의 끝개를 찾는 방법, R_C 로부터 도달불가능 상태를 찾는 체계적인 방법을 제안하였다. 이로부터 비선형 CA의 상태변화 그래프를 도달가능 트리를 그리지 않고 효과적으로 작성할 수 있으며 비선형 CA의 상태전이 행동분석에 도움이 될 것으로 사료된다.

REFERENCES

[1] John von Neumann, Theory of Self-Reproducing Automata, Arthur W. Burks ed. Univ. of Illinois Press, Urbana and London, 1966.
 [2] M. Arbib, "Simple Self-reproducing universal automata," *Information and Control*, vol. 9, pp. 177-289, 1966.
 [3] E.R. Banks, "Information Processing and Transmission in Cellular Automata," Ph.D. thesis, MIT, 1971.

[4] S.J. Cho, U.S. Choi, H.D. Kim, Y.H. Hwang, J.G. Kim, S.H. Heo, "New Synthesis of One-Dimensional 90/150 Linear Hybrid Group Cellular Automata," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Trans.* vol. 26, no. 9, pp. 1720-1724, 2007.
 [5] O. Martin, A. M. Odlyzko, and Stephan Wolfram, "Algebraic Properties of Cellular Automata," *Comm. Math. Phys.* vol. 93, pp. 219-258, 1984.
 [6] S. Wolfram, "Statistical mechanics of cellular automata," *Rev. Mod. Phys.* 55(3), 601-644, 1983.
 [7] S. Wolfram, "Random sequence generation by cellular automata," *Advances in Applied Mathematics*, vol. 7, no. 2, pp. 123-169, 1986.
 [8] A. Wuensche and M. Lesser, The Global Dynamics of Cellular Automata; An Atlas of Basin of Attraction Fields of One-Dimensional Cellular Automata. Santa Fe Inst. Studies in the Sci. of Complexity, Addison Wesley, 1992.
 [9] N. Ganguly, A. Das, P. Maji, B. K. Sikdar, and P. P. Chaudhuri, Evolution of cellular automata based associative memory for pattern recognition, *High Performance Computing, Hyderabad, India*, 2001.
 [10] P. Pal Chaudhuri, D. Roy Choudhury, S. Nandi, and S. Chattopadhyay, Additive Cellular Automata Theory and Applications, vol. 1. *IEEE Computer Society Press, USA*, 1997.
 [11] N. Ganguly, Cellular Automata Evolution : Theory and Applications in Pattern Recognition and Classification. Ph.D. thesis, Bengal Engineering College (a Deemed University), India, 2004.
 [12] S.J. Cho, H.D. Kim, S.T. Kim, U.S. Choi, K.J. Lee, G.T. Kong, and J.G. Kim, "Finding Nonreachable State of a Linear TPMACA," submitted.
 [13] Sukanta Das, Biplab K. Sikdar, and P. Pal Chaudhuri, "Characterization of Reachable/Nonreachable Cellular Automata States," *ACRI 2004, LNCS 3305*, pp. 813-822, 2004.
 [14] Sukanta Das, Anirban Kundu, Biplab K. Sikdar, and P. Pal Chaudhuri, "Design of Nonlinear CA Based TPG Without Prohibited Pattern Set In Linear Time," *Journal of Electronic testing: Theory and Applications*, vol. 21, no. 1, pp. 95-107, 2005.
 [15] Sukanta Das, Sukanya Mukherjee, and Nazma Naskar, and Biplab K. Sikdar, "Characterization of Single Cycle CA and its Application in Pattern Classification," *Electronic Notes in Theoretical Computer Science*, vol. 252, pp. 181-203, 2009.



권민정(Min-Jeong Kwon)

2002년 부산대학교 교육대학원 수학과 졸업(교육학석사)
2013년 부경대학교 대학원 응용수학과 졸업(이학박사)
2014~현재 한국과학영재학교 교사
※ 관심분야: 셀룰라 오토마타론, 정보보호



조성진(Sung-Jin Cho)

1981년 고려대학교 대학원 수학과 졸업(이학석사)
1988년 고려대학교 대학원 수학과 졸업(이학박사)
1988년 ~ 현재 부경대학교 응용수학과 교수
※ 관심분야: 셀룰라 오토마타론, 정보보호



김한두(Han-Doo Kim)

1984년 고려대학교 대학원 수학과 졸업(이학석사)
1988년 고려대학교 대학원 수학과 졸업(이학박사)
1989년 ~ 현재 인제대학교 응용수학과 교수
※ 관심분야: 전산수학, 셀룰라 오토마타론



최언숙(Un-Sook Choi)

2000년 부경대학교 대학원 응용수학과 졸업(이학석사)
2004년 부경대학교 대학원 응용수학과 졸업(이학박사)
2009년 부경대학교 대학원 정보보호학과 졸업(공학박사)
현재 동명대학교 자율전공학부 교수
※ 관심분야: 셀룰라 오토마타론, 정보보호, 암호이론



이규진(Kue-Jin Lee)

2015. 2 부경대학교 응용수학과 이학석사
※ 관심분야: 셀룰라 오토마타론, 정보보호



공길탁(Gil-Tak Kong)

2014년 부경대학교 응용수학과 졸업(이학사)
2014. 3~현재 부경대학교 응용수학과 석사과정 중
※ 관심분야: 셀룰라 오토마타론, 정보보호