TPSACA를 기반으로 하는 완전 해싱 알고리즘

조성진*, 김한두**, 최언숙*, 허성훈***, 황윤희*

*부경대학교 수리과학부, **인제대학교 컴퓨터응용과학부, ***부경대학교 정보보호학(협)

TPSACA based Perfect Hashing Algorithm

Sung-Jin Cho*, Han-Doo Kim**, Un-Sook Choi*, Seong-Hun Heo***, Yoon-Hee Hwang*

*Division of Mathematical Science, Pukyong National Univ.,

**School of Computer Aided Science, Inje Univ.,

***Interdisciplinary Program of Information Security, Pukyong National Univ.

E-mail : sjcho@pknu.ac.kr

요 약

본 논문에서는 선형 TPMACA와 TPSACA의 특성을 이용하여 만들어진 트리구성알고리즘을 구성하고, TPSACA를 기반으로 하는 완전 해싱 알고리즘을 생성하고자 한다.

키워드 : 셀룰라 오토마타, 트리, 선형 nongroup CA, Attractor, 직전자, TPMACA, TPSACA, 트리구성알고리즘.

I. 서론

셀룰라 오토마타(Cellular Automata, 이하 CA)는 동역학계(dynamical system)를 해석하는 한 방법으로 공간과 시간을 이산적으로 다루고, 이산적인 공간을 셀룰라 공간(cellular space)의 기본 단위인 각 셀(cell)이 취할 수 있는 상태를 유한하게 처리하며, 각 셀들의 상태가 국소적인 상호작용에 의해서 동시에 갱신되는 시스템이다. Group CA의 상태전이 행동의 분석은 그동안 많은 연구가 이루어졌다([1], [2], [3], [4]). Group CA의 전이행렬은 역행렬이 존재하지만 nongroup CA의 전이행렬은 역행렬이 존재하지 않는다.

Group CA에 비하여 nongroup CA에 대한 연구는 그리 활발하지는 못하였으나 최근 해시함수 생성이나 암호, 부울 방정식의 해법, 논리회로 검사 등에 응용이 되면서 관심을 받기 시작하였다([5], [6], [7], [8]).

본 논문에서는 두 개의 직전자와 하나의 끝개를 갖는 CA(Two-predecessor single-attractor CA, 이하, TPSACA)를 이용하여 완전해시함수([9], [10])를 생성하고자 한다. 그리한 CA의 상태전이 그래프에서는 모든 상태들이 0-상태의 끝개에 뿌리를 둔 역 이진트리를 이룬다. 키의 주소를 계산하기 위하여 CA를 시드(seed)로서 키를 적재하고 그것이 끝개에 도달할 때까지 자동으로 실행한다. 각 상태전이는 CA의 자동 모드(mode) 연산에 대한 각 시간 단계에서의 진행을 나타낸다. 해싱에 대해 선택된 CA와 주어진 키 집합에 의존하여 이미 결정된 시간 단계에서 특별한 CA 셀의 내용들은 해시 주소를 구성하기 위하여 연결된다. 2장에서는 CA의 용어, TPMACA([11], [11])와 TPSACA의 정의와 간단한 성질들을 밝히고 3장에서는 TPSACA를 기반으로 하는 완전 해싱 기술을 서술하고 4장에서 결론을 맺는다.

*This work was supported by IITA : 2001-050-2

II. TPSACA와 트리구성알고리즘

1. CA의 용어

• 선형 nongroup CA : Nongroup CA에서 다음 상태를 결정짓는 상태전이 함수가 XOR 논리로만 이루어져 있어서 이 함수를 행렬로 표현할 수 있다. 이러한 CA를 선형 nongroup CA라 한다.

• Attractor : Nongroup CA의 상태전이 그래프에서 순환상태들 중 사이클의 길이가 1인 상태를 말한다.

• 직전자 : 임의의 도달가능한 상태 x에 대하여 x에 대한 이전상태를 말하며 선형 CA에서 전이행렬을 T라 할 때 $Ty = x$ 를 만족하는 상

태 y를 나타낸다.

• Depth : Nongroup CA의 상태전이 그래프에서 임의의 도달불가능한 상태에서 가장 가까운 순환상태로 가는데 걸리는 최소의 단계 수를 말한다.

• Level : 어떤 상태 x가 α-트리의 level l ($l \leq$ depth)에 있다는 것은 상태 x가 정확히 l 단계 후 상태 α가 되는 위치에 있다는 것이다. 즉, $T^p x = \alpha$가 되는 p값 중 최소값이 l이다.

• r-직전자 : 임의의 도달가능한 상태 x에 대하여 $T^r y = x$을 만족하는 상태 y를 상태 x의 r-직전자라 한다. ($1 \leq r \leq 2^n - 1$)

2. TPMACA와 TPSACA

선형 TPMACA는 선형 nongroup CA 중 모든 순환상태들이 attractor 이고 임의의 도달 가능한 상태에 대하여 적전자의 수가 2개인 CA를 말한다. 선형 TPMACA의 특별한 경우인 TPSACA는 임의의 하나의 도달 가능한 상태에 대하여 서로 다른 두 개의 직전자를 가지며 상태 0만을 순환상태인 선형 CA를 말한다. n-셀 선형 TPSACA의 트리의 깊이는 n이고 최소다항식은 x^n이다.

다음은 선형 TPMACA 의 중요한 특성들을 살펴본다.

① 전이행렬의 영공간의 차원 : 선형 TPMACA에서 α를 임의의 도달 가능한 상태라고 하면 $|\{y \mid Ty=\alpha\}| = 2$이다. 또한 $|\{y \mid Ty=\alpha\}| = |\{x \mid Tx=0\}| = 2$이 고 $Tx=0$인 x의 수가 2^1이며 자유변수의 개수는 1이다. 그러므로 T 의 영공간의 차원은 1 이다.

② Attractor의 수 : 어떤 상태 x가 attractor 이면 이 순환 상태 x의 사이클의 길이가 1이므로 $Tx = x$이다. 이는 $(T \oplus I)x = 0$이므로 attractor는 $(T \oplus I)x = 0$ 을 만족하는 해 x이다. 그러므로 attractor의 수는 $(T \oplus I)$의 계수가 k라면 영공간의 차원은 $n-k$이고 가능한 해의 수는 2^{n-k}이다. 선형 TPMACA의 attractor수와 같다.

③ 최소다항식 : Nongroup CA는 최소다항식을 통해 트리의 깊이와 주기를 알 수 있다. Nongrouop CA의 최소다항식은 $x^d\Phi(x)$ 로 표현된다. 여기서 d는 트리의 깊이를 나타내고 $\Phi(x)$를 통해 이 CA의 주기를 알 수 있다. $\Phi(x)$가 x^c+1를 나누는 c중 최소의 c가 주기가 된다. 그런데 선형 TPMACA는 모든 순환상태의 사이클의 길이가 1이므로 주기는 1이다. 그러므로 트리의 깊이가 d인 선형 TPMACA의 최소다항식은 $x^d(x+1)$ 이다.

④ 서로 다른 두 직전자의 합 : 상태 w와 u를 임의의 한 도달 가능한 상태의 서로 다른 직전자라 하면 $Tw=Tu$이고 $T(w \oplus u)=0$ 이다. 그런데 가정에서 w와 u가 서로 다른 상태이므로 $w \oplus u$는 상태 0의 0이 아닌 직전자이다. 그러므로 선형 TPMACA에서 임의의 도달 가능한 상태에 대한 서로 다른 두 직전자의 합은 상태 0의 0이 아닌 직전자와 같다. 여기서 상태의 합은 각 상태를 이진법의 수로 표현하였을 때 각 비트들간의 bitwise 합이다.

⑤ 서로 다른 트리에서 같은 위치에 놓여있는 상태들의 합 : 선형 TPMACA의 상태전이 그래프에서 α_{ij}를 α-트리의 level i의 j 번째 상태라 하고 β_{ij}를 β-트리의 level i의 j 번째 상태라 하자. 또한 P_{ij}를 0-트리의 level i의 j 번째 상태라 하면 $\alpha_{ij} = P_{ij} \oplus \alpha$ 이고, $\beta_{ij} = P_{ij} \oplus \beta$ 이다. 그리므로 $\alpha_{ij} \oplus \beta_{ij} = (P_{ij} \oplus \alpha) \oplus (P_{ij} \oplus \beta) = \alpha \oplus \beta$이 므로 서로 다른 트리에서 같은 위치에 놓여있는 상태들의 합은 각 트리의 attractor의 합과 같다.

3 트리구성 알고리즘

Step 1. 주어진 전이행렬이 T일 때 $(T \oplus I)x = 0$ 을 만족하는 attractor x 를 찾는다.

Step 2. T의 최소다항식 $m(x)$를 나누는 x^k 중 최대정수 k를 찾아 CA의 depth d를 구한다.

Step 3. $T^d y = 0$이고 $T^{d-1} y \neq 0$인 0-트리의 도달불가능상태 y하나를 찾는다.

Step 4. y를 시작으로 하는 0-트리의 0-기본경로 $y \rightarrow Ty \rightarrow \cdots \rightarrow 0$를 찾는다.

Step 5. $S_{l,k} = S_{l,0} + \sum_{i=1}^{l-1} b_i S_{i,0}$에 의하여 0-트리를 구성.

Step 6. $S^{\alpha}_{l,0} = S_{l,0} \oplus \alpha$에 의하여 α-트리의 α-기본경로를 찾는다.

Step 7. $S^{\alpha}_{l,k} = S^{\alpha}_{l,0} + \sum_{i=1}^{l-1} b_i S_{i,0}$에 의하여 나머지 α-트리를 구성.

/* 여원을 갖는 CA C′ 트리의 구성*/

Step 8. 여원벡터 F가 0이 아닌 attractor이면 C′의 0-기본경로를

$$\overline{S_{l,0}}=\begin{cases} S_{l,0} & l:\text{짝수} \\ S_{l,0}\oplus\beta & l:\text{홀수} \end{cases}$$

에 의하여 구하고 Go To Step 10.

Step 9. 여원벡터 F가 선형 TPMACA의 비순환 상태이고 도달 불가능한 상태이면 $0\rightarrow\overline{T}0(=F)\rightarrow\cdots\rightarrow\overline{T^{d}}0(=\overline{T}^{l-1}F)$를 $\overline{T}^{l-1}F$ - 트리의 기본경로로 하고, 도달 가능한 상태이면 선형 TPMACA의 0-트리의 도달 불가능한 상태 y에 대하여 $y\rightarrow\overline{T}y\rightarrow\cdots\rightarrow\overline{T^{d}}y(=\overline{T}^{l-1}F)$를 $\overline{T}^{l-1}F$-트리의 기본경로로 한다.

Step 10. $\overline{S_{l,k}}=\overline{S_{l,0}}+\sum_{i=1}^{l-1}b_iS_{i,0}$에 의하여 $\overline{T}^{l-1}F$-트리를 구성한다.

Step 11. $\overline{S^{\alpha}_{l,0}}=\overline{S_{l,0}}\oplus\alpha$에 의하여 또 다른 트리의 기본경로를 구성.

Step 12. $\overline{S^{\alpha}_{l,k}}=\overline{S^{\alpha}_{l,0}}+\sum_{i=1}^{l-1}b_iS_{i,0}$에 의하여 트리의 나머지 부분을 구성.

Ⅲ. TPSACA를 기반으로 하는 완전 해싱 기술

1. TPSACA를 기반으로 하는 완전 해싱이론 및 MRT 계산

k개의 키를 갖는 n-비트 키 집합을 k'비트 주소($\log_2 k\le 2^{k'}<n$)로 완전히 해시하는 기초 기술은 다음과 같다:

1. 선형 TPSACA와 여원 TPSACA의 기본경로를 이용하여 해시하기 위한 키 값 x의 위치를 찾는다.

2. 키 값 x에 대하여 완전 해시된 주소인 k' 비트들이 그 CA에 의해서 생성된 연속적인 상태들의 특정한 비트 위치로부터 얻어진다. 임의의 키들의 쌍에 대한 해시된 주소가 적어도 한 비트 위치에서 달라지므로 생성된 주소의 유일성을 보장한다.

위에서 언급한 기술을 상세하게 기술하면 다음과 같다. $X=\{x_1,x_2,\cdots,x_k\}$를 주어진 k개의 키를 원소로 갖는 집합이라 하고 W를 완전해시 주소를 생성하기 위하여 사용된 특성행렬 T를 갖는 TPSACA라 하자. 집합 X의 각 원소가 W의 상태전이 그래프의 제일 위쪽 level에 속한다고 하자. x_i와 x_j를 X에 있는 두 개의 키 값이라 하자. W의 상태전이 그래프에서 x_i와 x_j가 l단계 후 처음으로 상태 y로서 같아지는 위치에 있다고 하자. 즉, $l=\min\{k\mid T^kx_i=T^kx_j=y\}$라 하자. 이때 l을 MRT (minimal running time)이라 한다. 그러면 $T^l(x_i)=T^l(x_j)=y$가 성립한다. $T^{l-1}(x_i)(:=p_1)$과 $T^{l-1}(x_j)(:=p_2)$는 y의 직전자 이다. 따라서 $p_1\oplus p_2(:=z)$는 상태 0의 0이 아닌 직전자이다. 그러므로 만일 z의 r번째 비트($r=0,1,2,\cdots,(n-1)$)가 1이면 p_1과 p_2중 한 상태만 r번째 비트가 1이 되어야 하며, z의 r번째 비트가 0이면 p_1과 p_2의 r번째 비트가 서로 같아야 한다. 이러한 이유로 우리는 pivot 비트 위치의 개념이 필요하다.

<정의 1> Pivot Bit (PB) 위치 : 주어진 TPSACA의 상태전이 그래프에서 상태 0의 0이 아닌 직전자의 이진표현에서 가장 오른쪽에 있는 1의 비트 위치를 말한다. □

그러므로 p_1과 p_2는 PB 위치에서 서로 구별될 수 있다. x_i의 해시된 주소는 p_1로부터 얻어진 PB 위치의 비트 값을 가지며 x_j의 해시된 주소는 p_2로부터 얻어진 PB 위치의 비트 값을 가지게 된다. 결국 x_i와 x_j는 서로 다른 해시된 주소로 전달된다.

<예 1> 4-셀 선형 CA의 rule이 <150, 150, 102, 150>인 CA를 W라 할때 W의 의 전이행렬 T는 다음과 같다.

$$T=\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

□

그러므로 T의 계수는 3이고 $(T\oplus I)$의 계수는 4이다. 또한 W의 특성다항식과 최소다항식은 모두 x^4이다. 그러므로 W는 TPMACA의 특별한 분류인 TPSACA이다. 2장에서 제안한 트리구성 알고리즘에 따라 0-트리의 기본경로 $1\rightarrow2\rightarrow5\rightarrow8\rightarrow0$를 구하고 (Step 1~Step 4), Step 5에 의하여 0-트리의 나머지 부분을 구성한다. 그림 1은 W의 구조, 그림 2는 상태전이그래프이다.

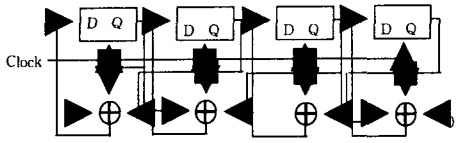

그림 1 : 4-셀 TPSACA의 구조

그림 1과 그림 2에서 0의 0이 아닌 직전자는 12(1100)이므로 PB 위치는 두 번째 비트이다. 1과 5가 키로 주어진다면 두 키의 MRT는 2이다. 4장 3절에 제안된 트리구성 알고리즘을 이용하여 0-트리의 기본경로로부터 키 1과 5의 위치가 level 4의 0번째와 2번째이므로 이 두 상태의 다음상태는 상태 1은 level 3에서 0번째 상태가 되고, 상태 5는 level 3의 1번째 상태가 된다. 0-트리의 기본경로를 이용하여 해당 상태를 구하면 level 3의 0번째 상태는 3(0011)이고 첫 번째 상태는 15(1111)이므로 PB위치의 값을 기준으로 하여 1과 5의 해시된 주소를 0과 1로 한다.

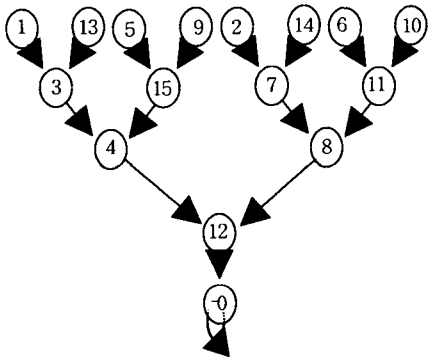

그림 2 : 4-셀 TPSACA의 상태전이 그래프

<MRT의 계산>

집합 X에 있는 주어진 원소 x에 대한 완전 해시된 주소는 주어진 TPSACA에 x를 로드하여 기본경로를 이용하여 원소 x의 위치를 정확히 알아내고 또한 이로써 다음상태의 위치를 알아내어 그 결과의 상태들로부터 PB번째 비트를 추출함으로서 생성될 수 있다. 결과는 $(n-1)$-비트 주소를 생성한다. 그러나 만일 집합 $Z=\{z=x_1 \oplus x_2 | x_1, x_2 \in X,\ x_1 \neq x_2\}$가 상태전이 그래프의 레벨 l의 어떤 상태도 갖지 않는다면, 키 집합 X의 가능한 모든 키의 순서쌍 중 MRT가 l 인 순서쌍은 없다. $(n-l+1)$번째 레벨에 대응하는 주소 비트를 얻을 필요가 없다. 이렇게 함으로써 주소비트가 $(n-2)$비트로 2비트가 줄어든다. 그러므로 주어진 키 집합에 대하여 임의의 키 순서쌍의 MRT를 구하는 것은 해시되는 주소의 비트수를 결정하는데 중요한 요인이 된다.

TPSACA의 같은 트리에서 동일한 레벨에 놓여있는 두 상태의 키 순서쌍 (x_i, x_j)의 MRT는 기본경로에 의하여 구해진 두 상태의 위치를 XOR하여 얻을 수 있다. x_i와 x_j가 최상위 레벨 d에 놓여있는 두 상태라 하자. 그리고 $b_i(i=1, \cdots, d-1)$를 x_i상태의 레벨 d에서 위치를 이진표현으로 나타내었을 때의 각각의 위치 비트라 하고, $b'_i(i=1, \cdots, d-1)$를 x_j상태의 레벨 d에서 위치를 이진표현으로 나타내었을 때의 각각의 위치 비트라 하면 c_i를 다음과 같이 구한다.

$$c_i = b_i \oplus b'_i (i=1, \cdots, d-1) \qquad (1)$$

식(1)에서 구한 c_i값 중 1인 최소 i를 찾는다. 이때 두 순서쌍의 MRT는 $(d-i)$이다.

<예 2> 그림 1과 그림 2의 TPSACA W에 대하여 주어진 키 집합이 $X=\{1, 13, 2, 14\}$이라 하면 모든 키들은 최상위 레벨(도달 불가능한)의 상태이다. 상태 0의 0이 아닌 직전자가 12(1100)이므로 PB의 위치는 두 번째 비트이다. 0-트리의 기본경로를 이용하여 키 집합의 각 원소들의 최상위 레벨에서 위치가 각각 0, 1, 4, 5번째 상태이다. 이 상태들이 1단계 상태변화 후 level 3에서 위치는 각각 0, 0, 2, 2이다. 이는 키 순서쌍 (1, 13)과 (2, 14)의 MRT가 1임을 나타내므로 각 상태의 PB위치의 비트에서 구별이 된다.

□

다시 위의 과정을 반복하여 각 키가 2단계 상태변화 후 즉 level 2에서 위치가 0, 0, 1, 1이고 3단계 상태변화 후 모든 상태들은 level 1에서 모두 0 번째 상태인 상태 0의 0이 아닌 직전자 12가 된다. 그러므로 나머지 가능한 키의 순서쌍은 (1, 2), (1, 14), (13, 2), (13, 14)이고 이 순서쌍의 MRT는 모두 3이다. 그러므로 2단계 상태 변화 후 각 상태의 PB 위치에서 비트 값을 선택한다. 각 순서쌍은 MRT가 2인 경우가 없으므로 해시되는 주소의 비트수를 하나 줄일 수 있다. 그러므로 X에 속한 모든 키를 2 비트로 해시된 주소를 생성할 수 있다. 다음 표 1은 주어진 키를 해시하는 과정이다. 한 상태의 현재 레벨의 위치에서 다음 상태의 한 단계 아래의 레벨의 위치는 현재 위치를 2로 나눈 몫과 같다.

표 1 : 주어진 키의 해시 과정

Level 4		Level 3		Level 2		Level 1		해시된 주소
상태	위치	상태	위치	상태	위치	상태	위치	
1 (0001)	0	3 (0011)	0	4 (0100)	0	12	0	01
13 (1101)	1	3 (0011)	0	4 (0100)	0	12	0	11
2 (0010)	4	7 (0111)	2	8 (1000)	1	12	0	00
14 (1110)	5	7 (0111)	2	8 (1000)	1	12	0	10

지금까지는 집합 X의 각 원소들이 주어진 TPSACA의 상태전이 그래프에서 가장 위쪽 레벨에 있는 경우만을 언급하였다. 이것은 분명히 일반적이지 못하다. 따라서 이제부터는 완전히 일반적인 경우에 대하여 언급하고자 한다.

주어진 집합 X가 주어진 부분집합 X_1과 X_2로 이루어졌다고 하자. 여기서 X_1의 원소들은 주어진 TPSACA의 상태전이 그래프에서 최상위 레벨에 위치하고 X_2의 원소들은 나머지 레벨에 위치한다. 여기서 W의 도달 불가능한 상태중 하나 F를 여원벡터로 취함으로써 얻어진 여원 CA(:= V)의 상태전이 그래프에서 상태 0이 도달 불가능한 상태가 된다. 또한 여원 CA V에서 W의 도달 불가능한 상태는 도달 가능한 상태가 되고 도달 가능한 상태는 도달 불가능한 상태가 된다. 2장 3절의 트리구성 알고리즘으로부터 여원 TPSACA의 기본경로와 선형 TPSACA의 기본경로를 이용하여 선형 TPSACA로 해시할 수 없었던 X_2의 키들을 해시한다. 이때 해시된 주소는 주어진 키가 선형 TPSACA로 해시 된 것인지, 아니면 여원 TPSACA로 해시 된 것인지를 구별하는데 한 비트가 소요된다.

<예 3> 그림 2에 대하여 주어진 키 집합이 $X = \{1, 13, 2, 14, 3, 15, 12, 0\}$ 이라 하면 키 집합 X를 키가 선형 TPSACA의 상태 전이 그래프에서 놓여있는 위치에 따라 최상위 레벨인 경우 $X_1 = \{1, 13, 2, 14\}$과 그렇지 않은 경우 $X_2 = \{3, 15, 12, 0\}$로 나눌 수 있다. 집합 X_2의 키를 해시하기 위해 그림 2에서 여원벡터가 2인 여원 TPSACA를 이용한다.

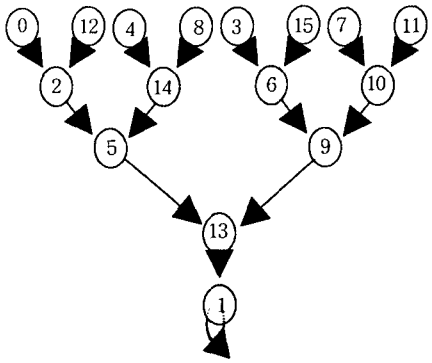

그림 3 : 여원 벡터가 2인 4-셀 여원 TPSACA의 상태전이 그래프

그림 3은 <예 1>의 선형 TPSACA로부터 유도된 여원 TPSACA의 상태전이 그래프이다. 키 3, 15, 12, 0을 해시 하는 방법은 선형 TPSACA를 이용하여 해시하는 방법과 동일하다. 이때 사용되는 여원 TPSACA의 기본경로는 $0 \rightarrow 2 \rightarrow 5 \rightarrow 13 \rightarrow 1$ 이다. 표 2는 주어진 키 집합 X_2를 해시하는 과정이다.

그러므로 주어진 키 집합의 해시된 주소는 다음과 같다. 여기에서 해시된 주소의 맨 마지막 비트의 값이 0인 경우는 키가 선형 TPSACA를 이용하여 해시 된 것임을, 비트 값이 1인 경우는 여원 TPSACA를 이용하여 해시 된 경우임을 나타낸다.

1 : 010　　13 : 110　　2 : 000　　14 : 100
3 : 001　　15 : 101　　12 : 111　　0 : 011

표 2 : 주어진 키 집합 X_2를 해시 과정

Level 4		Level 3		Level 2		Level 1		해시된 주소
상태	위치	상태	위치	상태	위치	상태	위치	
3 (0001)	4	6 (0110)	2	9 (1001)	1	13	0	00
15 (1111)	5	6 (0110)	2	9 (1001)	1	13	0	10
12 (1100)	1	2 (0010)	0	5 (0101)	0	13	0	11
0 (0000)	0	2 (0010)	0	5 (0101)	0	13	0	01

□

2. TPSACA를 기반으로 하는 완전 해싱 알고리즘

본 절에서는 2장 3절의 트리구성 알고리즘과 3장 1절에서 언급한 TPSACA를 기반으로 하는 완전 해싱이론과 MRT 계산법을 이용하여 TPSA CA를 기반으로 하는 완전 해싱 알고리즘을 서술한다.

Step 1. 선형 TPSACA의 기본경로로부터(트리구성 알고리즘: Step 1 ~ Step 4) 최상위 레벨의 상태를 구하고(트리구성 알고리즘: Step 5) 주어진 키 집합 (X)에서 선형 TPSACA의 최상위 레벨에 해당하는 키의 부분집합 (X_1)을 찾고 각 키의 위치를 찾는다.

Step 2. 선형 TPSACA에서 상태 0의 0이 아닌 직전자로부터 PB위치를 찾는다. $X = X_1$이면 Go To Step 4.

Step 3. $X_2 = X - X_1$라하고 선형 TPSACA의 임의의 도달 불가능한 상태가 여원 벡터 F가 되는 선형 TPSACA로부터 유도된 여원 TPSACA의 기본경로 $0 \rightarrow \overline{T}0 (= F) \rightarrow \cdots \rightarrow \overline{T^d}0 (= \overline{T}^n F)$를 구하고(트리구성 알고리즘: Step 9), 여원 TPSACA의 최상위 레벨의 상태를 구한다(트리구성 알고리즘: Step 10). 키 부분집합 X_2의 각 원소에 대하여 여원 TPSACA의 최상위 레벨에서의 위치를 찾는다.

Step 4. X_1과 X_2의 각각의 모든 키 순서쌍 (x_i, x_j)의 MRT를 계산하여 MRT집합 M을 구한다.

Step 5. X_1과 X_2의 각 원소의 위치를 나타내는 $b_i (i = 1, \cdots, d-1)$에 대하여 오른쪽에서부터 한 비트씩 줄여가면서 해당 하위 레벨에서의 상태들을 구하고 M의 원소 $m_i (1 \leq i \leq d-1)$에 대하여 $(d+1-m_i)$ 레벨에서 각 상태의 PB 위치의 비트를 추출한다.

Step 6. $X = X_1$ 또는 $X = X_2$이면 Stop.

Step 7. X_1에 속한 키의 해시된 주소의 맨 오른쪽에 0를, X_2에 속한 키의 해시된 주소의 맨 오른쪽에 1을 부여한다.

Ⅳ. 결론 및 향후 연구 방향

본 논문에서는 TPMACA와 TPMACA의 특수한 경우인 TPSACA의 특성을 이용하여 트리구성 알고리즘을 구성하였고, 트리구성 알고리즘과 TPSACA를 이용하여 완전해시함수를 생성하는 알고리즘을 제시하였다. 향후 연구 방향은 보다 계층적인 면에서의 연구가 필요하다고 사료된다. 예를 들면 기존의 연구는 GF(2) 상에서의 TPMACA 및 TPSACA에 관한 연구를 하였는데 이러한 연구를 $GF(2^p)$ (p:소수) 상에서의 연구로 확장할 필요가 있다.

참고문헌

[1] P.H. Bardell, "Analysis of cellular automata used as pseudorandom pattern generators", Proc. IEEE int. Test. Conf., pp. 762~767, 1990.

[2] A.K. Das and P.P. Chaudhuri, "Efficient characterization of cellular automata", Proc. IEE(Part E), Vol. 137, No. 1, pp. 81~87, 1990.

[3] S. Nandi and P.P. Chaudhuri, "Analysis of Periodic and Intermediate Boundary 90/150 Cellular automata", IEEE Trans. Computers, Vol. 45, No 1, pp. 1~12, 1996.

[4] M. Serra, T. Slater, J.C. Muzio and D.M. Miller, "The analysis of one dimensional linear cellular automata and their aliasing properties", IEEE Trans Computer-Aided Design, Vol. 9, pp. 767~778, 1990.

[5] S.J. Cho, U.S. Choi and H.D. Kim, "Linear nongroup one-dimensional cellular automata characterization on GF(2)", J. Korea Multimedia Soc., Vol. 4, No. 1, pp. 91~95, 2001.

[6] P.P. Chaudhuri, D.R. Chowdhury, S. Nandy and Chattopadhyay, Additive Cellular Automata Theory and Application, 1, IEEE Computer Society Press, California, 1997.

[7] A.K. Das and P.P. Chaudhuri, "Vector space theoretic analysis of additive cellular automata and its application for pseudo-exhaustive test pattern generation", IEEE Trans. Comput., Vol. 42, pp. 340~352, 1993.

[8] S. Nandi, B.K. Kar and P.P. Chaudhuri, "Theory and Application of Cellular Automata in Cryptography", IEEE Trans. Computers, Vol. 43, pp. 1346~1357, 1994.

[9] R. Cichelli, "Minimal Perfect Hash Functions made simple", Comm. ACM, Vol. 23, pp. 17-19, Jan. 1980.

[10] C. Cook and R. Oldehoeft, "More on Minimal Perfect Hash Table", Tech. Report TR-82, 1982.

[11] S.J. Cho, H.D. Kim and U.S. Choi, "Analysis of complemented CA derived from a linear TPMACA" (To appear in Compu- ters & Mathematics with Applications).

[12] S.J. Cho, H.D. Kim and U.S. Choi, "Behavior of Complemented cellular automata derived from a linear cellular automata" (To appear in Mathematical and Computer Modelling).