ALGORITHM FOR FINDING 90/150 TRIDIAGONAL MATRICES

Sung Jin CHO [1], Un Sook CHOI [2], Han Doo KIM [3], Yoon Hee HWANG [4] and Jin Gyoung KIM [1]

1) *Division of Mathematical Sciences, Pukyong National University, Busan 608-737, KOREA*
2) *Department of Multimedia Engineering,Tongmyong University, Busan 626-847, KOREA*
3) *Institute of Mathematical Sciences and School of Computer Aided Science, Inje University, Gimhae 621-749, KOREA*
4) *Department of Information Security, Graduate School, Pukyong National University, Busan 608-737, KOREA*

Corresponding Author: Han Doo KIM, mathkhd@inje.ac.kr

ABSTRACT

In this paper, we propose a new method for finding 90/150 Linear Hybrid Group Cellular Automata(LHGCA) for CA-polynomials.

INTRODUCTION

In this paper, we propose a new method to find CA for a given CA-polynomial by using the concept of similarity transformation and Lanczos tridiagonalization algorithm in $GF(2)$. By this method we propose a new efficient algorithm and show that for a given irreducible polynomial there exist two CA. Our method is different from the method of Cattell et al. We can obtain very large cell CA. We obtain the 1600 cell CA by about 44 CPU seconds. The program is written in C language and run a PC(Microsoft Windows XP Professional, Pentium(R) 4 CPU, 2.80 GHZ, 504 MB RAM).

THE REVISED LANCZOS TRIDIAGONALIZATION METHOD

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 $f(x)$.

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

Let

$$U=\begin{pmatrix} 1 & a_1 & d_{11} & \cdots & d_{1(n-4)} & d_{1(n-3)} & d_{1(n-2)} \\ 0 & 1 & a_2 & \cdots & d_{2(n-4)} & d_{2(n-3)} & d_{2(n-2)} \\ 0 & 0 & 1 & \cdots & d_{3(n-4)} & d_{3(n-3)} & d_{3(n-2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & a_{n-2} & d_{(n-2)(n-2)} \\ 0 & 0 & 0 & \cdots & 0 & 1 & a_{n-1} \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix}, \quad T=\begin{pmatrix} d_1 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & d_2 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & d_3 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & d_{n-1} & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & d_n \end{pmatrix}$$

(Hereafter we write T by $T=<d_1,d_2,\cdots,d_n>$.) T is called a 90/150 tridiagonal matrix.

For a given $(2n-1)$-tuple $(h_1,h_2,\cdots,h_{2n-2},h_{2n-1})$, a Hankel matrix H has the form

$$H=\begin{pmatrix} h_1 & h_2 & h_3 & \cdots & h_n \\ h_2 & h_3 & h_4 & \cdots & h_{n+1} \\ h_3 & h_4 & h_5 & \cdots & h_{n+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_n & h_{n+1} & h_{n+2} & \cdots & h_{2n-1} \end{pmatrix}$$

For a given n-vector x and $n\times n$ matrix M, let

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

$K(M,x)$ is a Krylov matrix.

THEOREM 1 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:

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

THEOREM 2 Let B be the $n\times n$ matrix obtained by reducing the n polynomials

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

where $f(x)$ is a polynomial. Then the elements of the set $\{v \mid Bv=(0,0,\cdots,0,1)^T\}$ satisfy the equation in Theorem 1, where $v=(h_1,h_2,\cdots,h_n)^T$.

THE PROPOSED ALGORITHM

ALGORITHM SynthesisOfLHGCA

Input: CA-polynomial $f(x)$

Output: 90/150 CA

Step1: Make the matrix B from (1).

Step2: Solve the equation $Bv = (0,0,\cdots,1)^T$.

Step3: Construct a Krylov matrix $H = K(C^T, v)$ by the seed vector v which is a solution of the equation in Step2.

Step4: Compute the LU factorization $H = LU$.

Step5: Compute CA for $f(x)$ by the matrix U using the equation in Theorem 1.

4 CONCLUSION

In this paper, we proposed a new method for the synthesis of one-dimensional 90/150 LHGCA for CA-polynomials. By a different method from Cattell et al.'s method we showed that for a given irreducible polynomial there exist just two CA. We obtained large cell CA very rapidly using our algorithm which does not use some Maple program. And we gave the table which contains very large cell CA.

Table 1 Comparison between Running Time[2] and Our's Running Time of LHGCA Synthesis Program

(In this table, only the exponents of the terms with nonzero coefficients are represented, so that (20, 3, 0) stands for the polynomial $x^{20} + x^3 + 1$.)

Polynomial	Time	Ours	LHGCA
(20,3,0)	0.2	0	01101010000111010110
(60,1,0)	0.6	0.2	11100111101001011101000010110011111010000 10111010010111100111
(100,37,0)	1.2	0.2	11111111100100000001111001011100111000001… 01100011111111010011110000000100111111111
(294,61,0)	15.56	0.2	00110001110010011100100001110010101110011… 10011101010011100001001110010011100011100
(295,48,0)	2.20	0.3	01100100011111100100111010011011110111001… 00111011110110010111001001111110001001110
(297,5,0)	2.23	0.3	01110010110110001101111001101101110111001… 00111011101101100111101100011011010011110
(300,7,0)	18.1	0.3	00001001011101111101010010000000001100001… 00001100000000010010101111101110100100000
(1600,1345,725,347,0)	600	44	01101011011110110110100000110110010011111… 11110010011011000001011011011111011010110

REFERENCES

1. M. Serra and T. Slater, “A Lanczos Algorithm in a Finite Field and its Application,” *J. Combinatorial Math. and Combinatorial Computing*, Vol. 7, 1990, pp. 11-32.
2. K.M. Cattell and Jon C. Muzio, “Synthesis of One-Dimensional Linear Hybrid Cellular Automata,” *IEEE Trans. on Computer Aided Design of Circuits and Systems*, Vol. 15-3, 1996, pp. 325-335.
3. S.J. Cho, U.S. Choi, Y.H. Hwang and H.D. Kim, “Analysis of Complemented CA Derived from Linear Hybrid Group CA,” *Computers Math. Applic., To appear.*