THE SET OF REVERSIBLE 90/150 CELLULAR AUTOMATA IS REGULAR

Palash Sarkar * Rana Barua†

July 31, 1996

Indian Statistical Institute
Technical Report No. ASC/96/11

Keywords: Continuants, Regular Expressions, Finite Automata, Reversible 90/150 Cellular Automata

1 Introduction

In this article we pose and solve two problems regarding a special class of matrices over F_2, the field of two elements. Let, M_b, $b \in \{0,1\}$, be a square matrix over F_2, having the following structure.

$$M_b = \begin{bmatrix} a_1 & 1 & 0 & 0 & 0 & \ldots & 0 & 0 & b \\ 1 & a_2 & 1 & 0 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 1 & a_3 & 1 & 0 & \ldots & 0 & 0 & 0 \\ . & & & & & & & & . \\ . & & & & & & & & . \\ . & & & & & & & & . \\ 0 & 0 & 0 & 0 & 0 & \ldots & 1 & a_{n-1} & 1 \\ b & 0 & 0 & 0 & 0 & \ldots & 0 & 1 & a_n \end{bmatrix} \qquad (I)$$

*Computer Science Unit, Indian Statistical Institute, 203, B.T. Road, Calcutta 700035, INDIA. e-mail: palash@isical.ernet.in

†Stat/Math Unit, Indian Statistical Institute, 203, B.T. Road, Calcutta 700035, INDIA. e-mail: rana@isical.ernet.in

First assume $b = 0$. Then M_0 is uniquely specified by the string $a_1 \ldots a_n$ over the alphabet $\{0, 1\}$ and is said to encode the matrix M_0. Now consider the following problems.

1. Obtain a characterization of the set of strings $a_1 \ldots a_n$ which encode non-singular matrices of the form M_0.

2. Find the number of non-singular matrices M_0 of order n.

We show that the set of strings which encode non-singular matrices is a regular set with a very simple structure. This solves the first problem. Using the "canonical" regular expression for this set, we completely solve the second problem. It turns out that approximately two-thirds of the strings encode non-singular matrices of the form M_0. The novel features of our proof are the use of continuants for tackling the first problem and the use of regular expression for counting. We believe that these ideas can be profitably applied to other similar situations.

For the case $b = 1$, the situation seems more complicated. However, using the results for matrices of the form M_0, we are able to satisfactorily solve the corresponding problems for matrices of the form M_1.

These problems arise very naturally in connection with the study of what are known as 90/150 Cellular Automata (CA). Cellular Automata (CA) are simple discrete dynamical systems, capable of exhibiting quite complex behaviour (see [6,11]). A finite one dimensional CA is an array of cells, where each cell can assume *state* 0 or 1, which are regarded as elements of the $0-1$ field F_2. It is an autonomous machine and evolves deterministically in discrete time steps. This evolution is effected by each cell changing its value according to a local rule. The *local rule* R_i for the ith cell c_i is a three variable boolean function and the state of c_i in the tth time step, denoted by x_i^t, is given by,

$$x_i^t \;=\; R_i(x_{i-1}^{t-1}, x_i^{t-1}, x_{i+1}^{t-1})$$

If the subscript i of x_i^t is taken modulo n, the number of cells of the CA, then the CA is called *periodic boundary* CA. If on the other hand, the array is considered to be placed between two cells having a fixed value zero, the CA is called *null boundary* CA. If all the R_i's are linear functions (SHIFT,EX-OR), then the CA is called *linear* (or additive) and the *global rule* is a linear transformation of the vector space F_2^n into itself that yields the *configuration* at the next time step during the evolution of the CA. For a more elaborate discussion on CA see [6,11].

Linear CA have been proposed as a basic structure in several areas of VLSI design ([5,8,9]). In fact the most useful structure from the VLSI point of view is a 90/150 structure where the local rule R_i is given by,

$$x_i^t \;=\; R_i(x_{i-1}^{t-1}, x_i^{t-1}, x_{i+1}^{t-1}) \;=\; x_{i-1}^{t-1} \;+\; a_i\, x_i^{t-1} \;+\; x_{i+1}^{t-1} \; (1 \leq i \leq n)$$

where $a_i \in \{0,1\}$ and addition is modulo 2 i.e, over F_2. If $a_i = 0$, R_i is called rule 90, else R_i is called rule 150. (See [11] for a nomenclature of local rules). The global rule of such a CA is specified by the matrix (called its *transition matrix*) M_b of (I), where b is 0 or 1 according as there is a null or periodic boundary condition. Once the boundary condition is fixed the string $a_1 \ldots a_n$ over $\{0,1\}$ completely specifies the structure of the CA and hence we shall identify the CA and its transition matrix with the string $a_1 \ldots a_n$. The CA is *reversible* iff its transition matrix is non-singular.

The transition matrix for null boundary 90/150 CA is M_0 (from (I)). The determinant of M_0 can be elegantly expressed in terms of multivariate polynomials called continuants, which were first introduced and studied by Euler ([2]). A continuant in n variables $K_n(x_1,\ldots,x_n)$ is defined by the following recurrence.

$K_0() = 1$, $K_1(x_1) = x_1$

$$K_n(x_1,\ldots,x_n) \;=\; x_1\, K_{n-1}(x_2,\ldots,x_n) \;+\; K_{n-2}(x_3,\ldots,x_n) \qquad (II)$$

In fact the continuants satisfy a more general recurrence ([2,pp 289])

$$\begin{aligned} &K_{m+n}(x_1,\ldots,x_m,x_{m+1},\ldots,x_{m+n}) \\ &\quad= K_m(x_1,\ldots,x_m)K_n(x_{m+1},\ldots,x_{m+n}) \\ &\qquad + K_{m-1}(x_1,\ldots,x_{m-1})K_{n-1}(x_{m+2},\ldots,x_{m+n}) \qquad (III) \end{aligned}$$

and using the realtion in [2,pp 304], we have,

$$K_n(a_1,\ldots,a_n) \;=\; \det M_0$$

Also the characteristic polynomial of M_0 is $K_n(x+a_1,\ldots,x+a_n)$ (note that over F_2, $-1 = +1$). Hence M_0 is non-singular iff $K_n(a_1,\ldots,a_n) = 1$. Expanding M_0 by the first and the last row it is easy to see that

$$K_n(a_1,\ldots,a_n) \;=\; K_n(a_n,\ldots,a_1) \qquad (IV)$$

Thus it is most natural to consider continuants in the analysis of 90/150 null boundary CA and we know of no other place where this has been done.

In fact the problems that we have posed can be framed entirely in terms of continuants, and hence our work can also be considered to be a contribution to the study of continuants.

Finally we point out the implications of our results to the theory of linear finite state machines. The counting results show that certain kinds of linear machines cannot be synthesised using 90/150 CA.

In what follows all arithmetic is over F_2 and ϵ will denote the empty string. Also $|x|$ denotes the length of a string x, and the cardinality of a set S is denoted by $|S|$.

2 Null Boundary CA

As stated in the introduction, the characteristic polynomial of the transition matrix of a null boundary 90/150 CA is a continuant $K_n(x+a_1, \ldots, x+a_n)$. The CA is reversible iff the constant term of $K_n(x+a_1, \ldots, x+a_n)$ is 1. The constant term is obtained by putting $x = 0$ and is equal to $K_n(a_1, \ldots, a_n)$. Since the CA is uniquely identified by the string $a_1 \ldots a_n$ over {0,1}, we will write that the string $a_1 \ldots a_n$ is reversible to mean that the corresponding CA is reversible. First note that the empty string ϵ is reversible. Next we have the following

LEMMA 2.1. *Let $y \in \{0,1\}^*$ and $i \in \{0,1\}$. Then*

a) 0iy is reversible iff y is reversible.

b) 10y is reversible iff 1y is reversible.

c) 11y is reversible iff 0y is reversible.

Proof : a) Using (III), we can write,
$K_n(0, i, a_3, a_4, \ldots, a_n) = K_2(0, i)K_{n-2}(a_3, a_4, \ldots, a_n) + K_1(0)K_{n-3}(a_4, a_5, \ldots, a_n)$
Now, $K_2(0, i) = 0.i + 1 = 1$
Therefore, $K_n(0, i, a_3, a_4, \ldots, a_n) = K_{n-2}(a_3, a_4, \ldots, a_n)$. This proves (a).
(b) $K_n(1, 0, a_3, \ldots, a_n)$

$$\begin{aligned} &= K_2(1,0)K_{n-2}(a_3, \ldots, a_n) + K_1(1)K_{n-3}(a_4, \ldots, a_n), \text{ by } (III) \\ &= K_{n-2}(a_3, \ldots, a_n) + K_{n-3}(a_4, \ldots, a_n) \\ &= K_{n-1}(1, a_3, \ldots, a_n) \text{ by } (II) \end{aligned}$$

This proves (b).
(c) is similar to (b). □

Given a string y we can repeatedly "reduce" it from the left to obtain shorter strings which are reversible iff the original string is reversible. To formalise this, for any two strings u, v we write $u \rightarrow v$ and say u reduces to v if

1. $u = 0iv, \quad i \in \{0, 1\}$ or

2. $u = 10x$ and $v = 1x$ or

3. $u = 11x$ and $v = 0x$

Note that if $u \rightarrow v$ then $|v| < |u|$. By abuse of notation we will write $u \rightarrow v$ (and also say u reduces to v) if there exists strings $u_0, \ldots, u_n$ such that $u = u_0 \rightarrow u_1 \rightarrow \ldots \rightarrow u_n = v$.

REMARK 2.1. *Similar reduction from the right is also possible.*

The irreducible strings are simple to characterize.

PROPOSITION 2.1. *Let $y \in \{0, 1\}^*$. Then y reduces in zero or more steps to exactly one of the strings in $\{\epsilon, 0, 1\}$. Furthurmore, y is reversible iff y can be reduced to either ϵ or 1.*

Proof : By the reduction rules, any string of length ≥ 2 can be reduced. Hence the only irreducible strings are $\{\epsilon, 0, 1\}$. That the reduction is unique follows from the fact that at any stage at most one of the rules apply. The last statement holds since by lemma 2.1, any reduction preserves reversibility. □

From this we get the following linear time algorithm for determining reversible null boundary 90/150 CA. (See [10] for algorithms to determine reversibility of other kinds of CA).

Algorithm $\mathcal{A}$
input : A string $x = a_1 \ldots a_n$ over $\{0, 1\}$
output : "yes" if x is reversible, else "no"
while (x not in $\{\epsilon, 0, 1\}$) do
    if (($x = 00y$)or($x = 01y$) then $x = y$
    else if ($x = 10y$) then $x = 1y$

else if ($x = 11y$) then $x = 0y$
od
if ($x = \epsilon$ or $x = 1$) then output "yes" else output "no"

Using the idea of reversibility preserving reduction, one can obtain a Deterministic Finite Automata (DFA) to recognize all reversible strings. Since any initial prefix of the string can be reduced, all that the DFA has to do is to remember the effective (from the point of reversibility) amount of input seen so far. More formally let $M = (\{0,1\}, Q, s_\epsilon, \delta, F)$ be a DFA where,

1. $Q = \{s_\epsilon, s_0, s_1\}$ is the set of states.
2. The transition function δ is defined as follows. Let $i \in \{0,1\}$. Then,
   (a) $\delta(s_\epsilon, i) = s_i$
   (b) $\delta(s_0, i) = s_\epsilon$
   (c) $\delta(s_1, 0) = s_1$
   (d) $\delta(s_1, 1) = s_0$
3. $F = \{s_\epsilon, s_1\}$ is the set of final states.

The state s_ϵ correspond to the empty string, and any state $q \in Q$ remembers the effective amount of input seen so far. The transition function δ specifies the reduction rules. So we get the following

THEOREM 2.1. *Let $\mathcal{L}(M)$ be the language accepted by the DFA M. Then $y \in \mathcal{L}(M)$ iff y correspond to a reversible null boundary CA.*

Next we obtain the corresponding regular expression. Let $\boldsymbol{R}, \boldsymbol{R_0}, \boldsymbol{R_1}$ respectively correspond to the regular expressions for s_ϵ, s_0, s_1. Then we get,

$$\begin{aligned} \boldsymbol{R} &= \boldsymbol{R_0(1+0)} + \boldsymbol{\epsilon} \\ \boldsymbol{R_0} &= \boldsymbol{R\,0} + \boldsymbol{R_1\,1} \\ \boldsymbol{R_1} &= \boldsymbol{R_1\,0} + \boldsymbol{R\,1} \end{aligned}$$

We can solve this set of equations using Arden's lemma ([4,pp 54]), which states that for regular expressions $\boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{R}$ if $\boldsymbol{R} = \boldsymbol{P} + \boldsymbol{RQ}$, then $\boldsymbol{R} = \boldsymbol{PQ}^*$. So by a sequence of simple manipulations, we get,
$\boldsymbol{R} = \boldsymbol{((0+10^*1)(1+0))^*}$, $\boldsymbol{R_1} = \boldsymbol{R10^*}$, $\boldsymbol{R_0} = \boldsymbol{R(0+10^*1)}$
and the regular expression for $\mathcal{L}(M)$ is $\boldsymbol{R} + \boldsymbol{R_1}$. This leads us to the following

THEOREM 2.2. *The regular expression for the set of all reversible strings which correspond to null boundary CA is given by* $\boldsymbol{\alpha} + \boldsymbol{\alpha}\mathbf{10^*}$, *where* $\boldsymbol{\alpha} = \mathbf{((0+10^*1)(1+0))^*}$.

Given this regular expression, it is possible to enumerate the number of reversible strings of length n. Let S denote the set of reversible strings. Then, $S = L_\epsilon \cup L_1$, where L_ϵ (resp. L_1) is the set of all strings which reduces to ϵ (resp. 1). From proposition 2.1, $L_\epsilon \cap L_1 = \phi$. Let $S^{(n)}, L_\epsilon^{(n)}, L_1^{(n)}$ denote the subset of all strings of length n belonging to S, L_ϵ, L_1 respectively. Then, $|S^{(n)}| = |L_\epsilon^{(n)}| + |L_1^{(n)}|$. Next we prove,

PROPOSITION 2.2. *For* $n \geq 0$, $|S^{(n)}| = \sum_{i=0}^{n} |L_\epsilon^{(i)}|$

Proof : The regular expression for L_1 is $\boldsymbol{\alpha}10^*$ where $\boldsymbol{\alpha}$ is the regular expression for L_ϵ. Let $x \in L_1^{(n)}$ be such that the last zero is chosen $i \geq 0$ times. Then $x = y10^i$ where $y \in L_\epsilon^{(n-1-i)}$. Conversly for any $y \in L_\epsilon^{(n-1-i)}$ we get an unique $x \in L_1^{(n)}$. Therefore,

$$|L_1^{(n)}| = |L_\epsilon^{(n-1)}| + |L_\epsilon^{(n-2)}| + \ldots + |L_\epsilon^{(0)}|$$

Hence,

$$|S^{(n)}| = |L_\epsilon^{(n)}| + |L_1^{(n)}| = \sum_{i=0}^{n} |L_\epsilon^{(i)}| \qquad \square$$

So the problem reduces to computing $|L_\epsilon^{(n)}|$. It turns out that $|L_\epsilon^{(n)}|$ satisfies a nice recurrence relation.

LEMMA 2.2. $|L_\epsilon^{(0)}| = 1$, $|L_\epsilon^{(1)}| = 0$

$$|L_\epsilon^{(n)}| = |L_\epsilon^{(n-1)}| + 2\,|L_\epsilon^{(n-2)}|, \quad \textit{for } n \geq 2$$

Proof : Let $x \in L_\epsilon^{(n)}$. If $|x| < 2$, then it is easy to see that $|L_\epsilon^{(0)}| = 1$ and $|L_\epsilon^{(1)}| = 0$. So for $|x| \geq 2$, x can be written as $x = aby$ where $|y| = n-2$.
<u>Case 1</u> : $ab = 00$ or $ab = 01$. Then we have $x \to y$. So x reduces to ϵ iff y reduces to ϵ. Hence for each reversible string $y \in L_\epsilon^{(n-2)}$, we get two strings in $L_\epsilon^{(n)}$.
<u>Case 2</u> : $ab = 10$ or $ab = 11$.

If $ab = 10$, then x reduces to ϵ iff $1y$ reduces to ϵ.

If $ab = 11$, then x reduces to ϵ iff $0y$ reduces to ϵ.

So for each string in $L_\epsilon^{(n-1)}$ there exists exactly one string in $L_\epsilon^{(n)}$ and all strings in $L_\epsilon^{(n)}$ arise as Case 1 or Case 2. Hence,

$$|L_\epsilon^{(n)}| \;=\; |L_\epsilon^{(n-1)}| \,+\, 2\,|L_\epsilon^{(n-2)}| \qquad \Box$$

Corollary 2.1. *For $n \geq 2$,*

1. $|L_\epsilon^{(n)}| \;=\; 2\sum_{i=0}^{n-2} |L_\epsilon^{(i)}|$.
2. $|S^{(n)}| \;=\; \frac{3}{2}\,|L_\epsilon^{(n)}| \,+\, |L_\epsilon^{(n-1)}|$.

Proof :
1. Follows from the above lemma by induction.
2. Follows from 1 and Proposition 2.2 $\Box$

The next step is to obtain an expression for $|L_\epsilon^{(n)}|$ via its generating function.

Lemma 2.3. *For $n \geq 0$, $|L_\epsilon^{(n)}|$ is the coefficient of x^n in*

$$G(x) \;=\; \frac{1-x}{1-x-2x^2}$$

and hence is given by,

$$|L_\epsilon^{(n)}| \;=\; \frac{2}{3}\,[2^{n-1} \,+\, (-1)^n]$$

Proof : The generating function is obtained by standard manipulations and hence we shall omit it. To see the second statement, note

$$G(x) \;=\; \frac{1-x}{1-x-2x^2} \;=\; \frac{1}{3}[\frac{2}{1+x} \,+\, \frac{1}{1-2x}]$$

Hence the coefficient of x^n in $G(x)$ is

$$\frac{1}{3}[2.(-1)^n \,+\, 2^n] \;=\; \frac{2}{3}[2^{n-1} \,+\, (-1)^n]$$

$\Box$

We finally obtain,

THEOREM 2.3. *For $n \geq 0$,*

$$|S^{(n)}| = \frac{1}{3}[2^{n+1} + (-1)^n]$$

Consequently $|S^{(n)}|$ satisfies the following recurrence.

$$|S^{(0)}| = 1 \text{ and } |S^{(n)}| = 2\,|S^{(n-1)}| + (-1)^n \text{ for } n \geq 1$$

Proof : For $n = 0, 1$ it is easy to check that $|S^{(n)}| = 1$. From Corollary 2.1, for $n \geq 2$,

$$|S^{(n)}| = \frac{3}{2}|L_\epsilon^{(n)}| + |L_\epsilon^{(n-1)}|$$

By the above lemma, $|L_\epsilon^{(n)}| = \frac{2}{3}[2^{n-1} + (-1)^n]$. Hence,
$|S^{(n)}| = \frac{3}{2}[\frac{2}{3}(2^{n-1} + (-1)^n)] + \frac{2}{3}[2^{n-2} + (-1)^{n-1}]$
$= \frac{1}{3}[2^{n+1} + (-1)^n]$ $\Box$

REMARK 2.2. *There are an exponential number of reversible strings. In fact approximately two thirds of all strings of length n are reversible.*

3 Periodic Boundary CA

Next we turn to the characterization of periodic boundary CA. The transition matrix for such a CA is of the following form.

$$P = \begin{bmatrix} a_1 & 1 & 0 & 0 & \dots & 0 & 1 \\ 1 & a_2 & 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & a_3 & 1 & \dots & 0 & 0 \\ . & . & . & . & \dots & . & . \\ . & . & . & . & \dots & & 1 \\ 1 & 0 & . & . & \dots & 1 & a_n \end{bmatrix}$$

In analogy with continuants, let us denote the determinant of P by $P_n(a_1, \dots, a_n)$. We will only consider $n \geq 2$. Then we have the following

PROPOSITION 3.1. $P_n(a_1, \dots, a_n) = K_n(a_1, \dots, a_n) + K_{n-2}(a_2, \dots, a_{n-1})$. *Consequently $a_1 \dots a_n$ is reversible under periodic boundary condition iff exactly one of $a_1 \dots a_n$ and $a_2 \dots a_{n-1}$ is reversible under null boundary condition.*

Proof : Expanding the determinant by the first row, we get,

$$
\begin{aligned}
&P_n(a_1,\ldots,a_n)\\
&= a_1\,K_{n-1}(a_2,\ldots,a_n)\\
&\quad + \begin{vmatrix} 1 & 1 & \ldots & 0 & 0\\ 0 & a_3 & \ldots & 0 & 0\\ 0 & 1 & \ldots & 0 & 0\\ 0 & 0 & \ldots & 0 & 0\\ . & . & \ldots & . & .\\ . & . & \ldots & . & .\\ 0 & 0 & \ldots & . & 1\\ 1 & 0 & \ldots & 1 & a_n \end{vmatrix} + \begin{vmatrix} 1 & a_2 & 1 & 0 & \ldots & 0\\ 0 & 1 & a_3 & 1 & \ldots & 0\\ . & . & . & . & \ldots & .\\ . & . & . & . & \ldots & .\\ . & . & . & . & \ldots & .\\ . & . & . & . & \ldots & .\\ . & . & . & . & \ldots & a_{n-1}\\ 1 & 0 & . & . & \ldots & 1 \end{vmatrix}\\
&= a_1\,K_{n-1}(a_2,\ldots,a_n) + K_{n-2}(a_3,\ldots,a_n) + 1\\
&\quad + K_{n-2}(a_2,\ldots,a_{n-1}) + 1\\
&\quad \text{(by expanding each of the two determinants by the first column)}\\
&= K_n(a_1,\ldots,a_n) + K_{n-2}(a_2,\ldots,a_{n-1}), \text{ by } (II)
\end{aligned}
$$

Consequently, under periodic boundary condition, $a_1\ldots,a_n$ is reversible iff $P_n(a_1,\ldots,a_n) = 1$, i.e, iff exactly one of $K_n(a_1,\ldots,a_n)$ and $K_{n-2}(a_1,\ldots,a_{n-1})$ is 1, i.e, iff exactly one of $a_1\ldots a_n$ and $a_2\ldots a_{n-1}$ is reversible under null boundary condition. □

REMARK 3.1. 1. *The continuant $K_n(a_1,\ldots,a_n)$ can be obtained by the following simple rule ([2]). Start with the term $a_1a_2\ldots a_n$ and then cancel out pairs a_ka_{k+1} in all possible ways. From the above proposition a similar rule holds for $P_n(a_1,\ldots,a_n)$ with the following modification. When considering pairs a_ka_{k+1}, consider a_na_1 to be one such pair, i.e, consider the terms $a_1,\ldots,a_n$ to be arranged in a circle.*

2. *The expression $P_n(a_1,\ldots,a_n)$ is invariant under a circular shift of its arguments.*

Based on the above proposition we can construct a DFA G to recognize all possible strings which correspond to reversible periodic boundary CA. The idea is to run two DFAs M_1 and M_2 in parallel, where M_1 and M_2 are copies of the DFA for recognizing reversible null boundary CA. The DFA M_1 will run on the entire string $a_1\ldots a_n$ while DFA M_2 will effectively run only on $a_2\ldots a_{n-1}$. Then we accept iff exactly one of M_1 and M_2 accepts. It is easy to design G such that M_2 skips the first symbol, i.e, a_1. When G reads $a_i,\ \ i > 1$, it makes a transition from q_1 to q_2 in M_1 and from p_1 to p_2 in

M_2, following the rules of lemma 2.1. Skipping the last symbol is a bit more tricky since G cannot know that a_n is the last symbol until it has read it. To tackle this we allow the control of G to have one more bit of memory (say b), which is used in the following way. When G makes a transition from p_1 to p_2 in M_2, it puts a value of 1 in b if p_1 was an accepting state for M_2 else it puts a value of 0 in b. So at the end of the input b indicates whether $a_2 \ldots a_{n-1}$ was an accepting string for M_2. Then G accepts iff either b is 1 and M_1 is in a rejecting state or b is 0 and M_1 is in an accepting state.

So from this description we get

THEOREM 3.1. *The set of all strings which correspond to reversible periodic boundary CA, form a regular set.*

Consequently there exists a linear time algorithm to determine reversibility of periodic boundary 90/150 CA.

Proof : We provide a formalization of the above description.

Let $M = (\{s_\epsilon, s_0, s_1\}, s_\epsilon, \delta, \{s_\epsilon, s_1\})$ be the DFA for the null boundary CA. Let,

$$r : \{\epsilon, 0, 1\} \to \{0, 1\}, \text{ where } r(\epsilon) = r(1) = 1, \text{ and } r(0) = 0$$

Define $G = (Q_p, s, \delta_p, F_p)$, to be a DFA, where

1. $Q_p = \{s, s_0, s_1\} \cup \{s_\epsilon, s_0, s_1\} \times \{s_\epsilon, s_0, s_1\} \times \{0, 1\}$
2. Let $i, j \in \{0, 1\}$ and $x, y \in \{0, 1, \epsilon\}$.
   (a) $\delta_p(s, i) = s_i$
   (b) $\delta_p(s_0, i) = (s_\epsilon, s_i, 1)$
   (c) $\delta_p(s_1, 0) = (s_1, s_0, 1)$
   (d) $\delta_p(s_1, 1) = (s_0, s_1, 1)$
   (e) $\delta_p((s_x, s_y, i), j) = (s_{\delta(x,j)}, s_{\delta(y,j)}, r(y))$
3. $F = \{(s_x, s_y, i) : r(x) \neq i\}$

It is easy to see that G formalizes the DFA described above and G accepts a string x iff x correspond to a reversible periodic boundary 90/150 CA. $\Box$

We now enumerate the number of strings which correspond to reversible periodic boundary 90/150 CA. In this case the regular expression is more complicated, so we use the results for null boundary CA.

Let $T^{(n)}$ be the set of all strings of length n which correspond to reversible periodic boundary CA. From Proposition 3.1, $T^{(n)}$ can be written as,

$$T^{(n)} \;=\; A^{(n)} \cup B^{(n)} \cup C^{(n)} \cup D^{(n)}$$

where,

$$\begin{aligned}
A^{(n)} &= \{x \in T^{(n)} : x = azb,\ a, b \in \{0,1\} \text{ and } x \to \epsilon,\ z \to 0\} \\
B^{(n)} &= \{x \in T^{(n)} : x = azb,\ a, b \in \{0,1\} \text{ and } x \to 1,\ z \to 0\} \\
C^{(n)} &= \{x \in T^{(n)} : x = azb,\ a, b \in \{0,1\} \text{ and } x \to 0,\ z \to \epsilon\} \\
D^{(n)} &= \{x \in T^{(n)} : x = azb,\ a, b \in \{0,1\} \text{ and } x \to 0,\ z \to 1\}
\end{aligned}$$

and $A^{(n)}, B^{(n)}, C^{(n)}, D^{(n)}$ are pairwise disjoint. Hence

$$|T^{(n)}| \;=\; |A^{(n)}| + |B^{(n)}| + |C^{(n)}| + |D^{(n)}| \qquad (V)$$

Next we prove two results which are crucial for enumerating $|T^{(n)}|$.

PROPOSITION 3.2. *Let $v \in \{0, 1, \epsilon\}$. Then there does not exist strings y ($|y| \geq 2$) such that $y = ax$, $a \in \{0,1\}$ and both $y \to v$ and $x \to v$.*

Proof : We will only prove the result for $v = 0$. The other two cases are similar. We prove by induction (on the length of strings) that there does not exist strings z such that, $z \to 0$ and $az \to 0$.
Base Step : For $|z| = 0$, $z = \epsilon$ and the result is easy.
Inductive Step : Suppose that the result holds for all strings of length less than n.

Let $|z| = n$. Suppose if possible $z \to 0$ and $az \to 0$. Recall that the regular expression for the strings reducing to 0 is $\boldsymbol{\alpha(0 + 10^*1)}$, where $\boldsymbol{\alpha}$ is the regular expression for strings reducing to ϵ. Since $z \to 0$, either

1. $z = y0$ and $az = ay0$ or,
2. $z = y10^i1$ and $az = ay10^i1$

with $|y| < |z|$.

Now in both cases $y \rightarrow \epsilon$ and $ay \rightarrow \epsilon$. For Case 1 this is clear. In Case 2, if $ay \rightarrow 0$, then $az \rightarrow \epsilon$ or $az \rightarrow 1$ according as i is odd or i is even. If on the other hand $ay \rightarrow 1$, then $az \rightarrow \epsilon$ or $az \rightarrow 1$ according as i is even or odd. Since $az \rightarrow 0$ it follows that ay must reduce to ϵ.

If $|y| = 0$ then we immediately have a contradiction. So suppose $|y| > 0$. Now $y \rightarrow \epsilon$ implies $y = wc$ ($c \in \{0,1\}$) such that $w \rightarrow 0$ and $ay \rightarrow \epsilon$ implies $aw \rightarrow 0$, where $0 \leq |w| < |y| < |z|$.

By the induction hypothesis such w does not exist. Hence the proof. □

Proposition 3.3. *For $n \geq 2$, let*

$$X_0^{(n)} = \{y \in L_\epsilon^{(n)} : y = ax \textit{ and } x \rightarrow 0\}$$

$$X_1^{(n)} = \{y \in L_\epsilon^{(n)} : y = ax \textit{ and } x \rightarrow 1\}$$

Then, $|X_0^{(n)}| = |X_1^{(n)}| = \frac{1}{2}|L_\epsilon^{(n)}|$

Proof : Let $y \in L_\epsilon^{(n)}$, with $y = ax$.

Let $x = zb$ so that $y = azb \rightarrow \epsilon$. Now $az0 \rightarrow \epsilon$ iff $az1 \rightarrow \epsilon$. So the strings in $L_\epsilon^{(n)}$ can be paired as $az0$ and $az1$. Then exactly one of the strings $z0$ and $z1$ reduces to 1 and the other reduces to 0. (By Proposition 3.2 none can reduce to ϵ).

Hence, $|X_0^{(n)}| = |X_1^{(n)}| = \frac{1}{2}|L_\epsilon^{(n)}|$ □

Now we can find the cardinalities of $A^{(n)}, B^{(n)}, C^{(n)}, D^{(n)}$. Following [2], we will let $[\phi]$ denote the value of a boolean predicate ϕ.

Lemma 3.1. *For all $n \geq 2$,*

1. $|A^{(n)}| = 0$
2. $|B^{(n)}| = [2 \nmid n] + \frac{1}{2}\sum_{i=1}^{n-1} |L_\epsilon^{(i)}|$
3. $|C^{(n)}| = [2|n] + \frac{1}{2}\sum_{i=1}^{n-2} |L_\epsilon^{(i)}|$
4. $|D^{(n)}| = \frac{1}{2}|L_\epsilon^{(n-1)}|$

Proof : 1. This is proved by proving that $A^{(n)} = \phi$. To see this first note that $x \in A^{(n)}$ iff $x = azb \to \epsilon$ and $z \to 0$. But $azb \to \epsilon$ implies $az \to 0$, hence $x \in A^{(n)}$ iff there exists string z such that $az \to 0$ and $z \to 0$. But by Proposition 3.2 such strings do not exist.

2. In this case $x \in B^{(n)}$ iff $x = azb \to 1$ and $z \to 0$.

If $b = 1$, $az \to \epsilon$ and $z \to 0$. There are $\frac{1}{2}|L_\epsilon^{(n-1)}|$ such strings (by Proposition 3.3).

If $b = 0$, then two cases arise

a) $z = 0^{n-2}$, $a = 1$ where $10^{n-1} \to 1$ and $0^{n-2} \to 0$. But then $n-2$ and hence n must be odd. This contributes the term $[2 \not| n]$ to $|B^{(n)}|$.

b) $z = y10^i$ where $0 \le i \le n-3$ and both $ay10^i \to 1$ and $y10^i \to 0$. Therfore $ay \to \epsilon$ and $y \to c$ for some $c \in \{0,1\}$. By Proposition 3.3 there are $\frac{1}{2}|L_\epsilon^{(n-2-i)}|$ such strings.

So, $|B^{(n)}| = [2 \not| n] + \frac{1}{2}\sum_{i=1}^{n-1} |L_\epsilon^{(i)}|$.

3 and **4** are similar to above. □.

So finally we get the following

THEOREM 3.2. *For all $n \ge 2$,*

$$|T^{(n)}| = |S^{(n-1)}| = \frac{1}{3}[2^n + (-1)^{n-1}]$$

Proof : Using the above lemma and (V),

$$\begin{aligned} |T^{(n)}| &= |A^{(n)}| + |B^{(n)}| + |C^{(n)}| + |D^{(n)}| \\ &= \textstyle\sum_{i=0}^{n-1} |L_\epsilon^{(i)}| = |S^{(n-1)}| = \frac{1}{3}[2^n + (-1)^{n-1}] \; \Box \end{aligned}$$

REMARK 3.2. *$|T^{(n)}|$ is approximately half of $|S^{(n)}|$ and one third of the total number of binary strings of length n.*

4 Linear Finite State Machines

In this section we point out the consequences of our results to the synthesis problem for CA. Cellular Automata belong to the class of Linear Finite State Machines (LFSM). The most popular examples of LFSMs are the Linear Finite State Registers (LFSR), which have been quite extensively studied ([3]). A LFSM is completely characterized by its characteristic polynomial, which defines the linear recurrence satisfied by the output bits of the machine.

A CA being an autonomous machine, there is no concept of output. However the successive states of any particular cell (usually one of the end cells) can be considered to be its output. Next we point out the relationship between the characteristic polynomial of the transition matrix of a CA and the linear recurrence satisfied by the output bits of any particular cell. To do that we need the following ([9]).

LEMMA 4.1. *Let M be the transition matrix for a 90/150 null boundary CA. Then M is nonderogatory, i.e, the minimal polynomial for M is the same as the characteristic polynomial for M.*

Now we prove the following

PROPOSITION 4.1. *Let M be the transition matrix of a 90/150 null boundary CA and let*

$$p(x) \; = \; x^n \, + \, c_{n-1}x^{n-1} \, + \ldots + \, c_0$$

be its characteristic polynomial. Then there exists a vector $\boldsymbol{x}$, such that the temporal sequence of any cell of the corresponding CA loaded with initial configuration $\boldsymbol{x}$, satisfies the linear recurrence defined by $p(x)$, i.e,

$$a_i^t \; = \; c_{n-1}a_i^{t-1} + \ldots + c_0 a_i^{t-n} \quad \textit{for } t \geq n$$

Proof : Let $\boldsymbol{x}$ be any non null vector and $q(x)$ be its minimal polynomial, i.e, the polynomial of the least degree such that $q(M)\boldsymbol{x} \, = \, \mathbf{0}$.

Then $q(x) \,|\, p(x)$ and the output of any cell of the CA loaded with initial configuration $\boldsymbol{x}$ will satisfy the linear recurrence defined by $q(x)$.

By the above lemma, $p(x)$ is the minimal polynomial for M and hence there exists a vector $\boldsymbol{x}$, whose minimal polynomial is $p(x)$. Therefore the result follows. $\square$

Given this result it is easy to see that any two CA having the same characteristic polynomial will essentially generate the same bit sequence (modulo a shift).

Given any bit sequence it is possible to synthesise a minimum length LFSR whose output is the given sequence. This is done by the famous Berlekamp-Massey Shift Register Synthesis Algorithm ([1,7]). The algorithm essentially finds the least degree polynomial which defines a linear recurrence satisfied by the given bit sequence. Designing a LFSR from this polynomial is trivial. So the natural question to ask in the context of CA is the following.

Given any bit sequence can we design a 90/150 CA whose output is the given bit sequence and the number of cells in the CA is equal to the number of cells in the minimum length LFSR which generates the same bit sequence ?

Unfortunately, the answer to this question is no and it follows from the fact that the answer to the following related question is also no.

Given any polynomial $p(x)$ of degree n, can we get an n-cell 90/150 CA whose transition matrix has characteristic polynomial $p(x)$?

For the following let us decide to call a polynomial (and the corresponding LFSM) reversible iff its constant term is 1. So there are exactly 2^{n-1} reversible polynomials of degree n. A CA will be said to realize an LFSM characterized by a polynomial $p(x)$ iff the characteristic polynomial of its transition matrix is $p(x)$. Then we get the following

PROPOSITION 4.2. *Using 90/150 null boundary CA, it is not possible to realize all irreversible LFSMs.*

Proof : The number of reversible strings of length n is $|S^{(n)}|$and hence the number of irreversible strings is $2^n - |S^{(n)}| = \frac{1}{3}(2^n + (-1)^{n+1}) = |S^{(n-1)}|$. The total number of irreversible machines is 2^{n-1} and the result follows from the fact that for $n \geq 2$, $|S^{(n-1)}| < 2^{n-1}$ □

Using a similar argument it is possible to prove,

PROPOSITION 4.3. *Using 90/150 periodic boundary CA, it is not possible to realize all reversible LFSMs.*

Since approximately two-thirds of all strings of length n correspond to reversible null boundary 90/150 CA and there are only 2^{n-1} reversible polynomials, one might expect that using null boundary 90/150 CA it is possible to realize all reversible LFSMs. However this is not true and to prove it requires a more delicate argument. First note that it is possible for two CAs to have the same characteristic polynomial. If $a_1 \dots a_n$ encodes a CA, then $K_n(x+a_1, \dots, x+a_n)$ is its characteristic polynomial and since $K_n(x+a_1, \dots, x+a_n) = K_n(x+a_n, \dots, x+a_1)$ (from (IV)), the CA encoded by $a_n \dots a_1$ also has the same characteristic polynomial. Of course if $a_1 \dots a_n$ is a palindrome, i.e, $a_i = a_{n-i}$ for all i, then this is trivially true. Otherwise we have two distinct CAs with the same characteristic polynomial. Let $D^{(n)}$ be the set of all reversible palindromes of length n. Define,

$$
\begin{aligned}
A_n &= 2^{n-1} - |D^{(n)}| \\
B_n &= \tfrac{1}{2}(|S^{(n)}| - |D^{(n)}|)
\end{aligned}
$$

Then there are at least A_n reversible polynomials which are not realised by reversible palindromic strings and there are at most B_n reversible polynomials which are realised by reversible non palindromic strings. So if we can prove that $B_n < A_n$, then we are done. We proceed by first finding $|D^{(n)}|$.

LEMMA 4.2. *For $n \geq 2$, $|D^{(n)}| = 2^{\lfloor \frac{n}{2} \rfloor - 1} + |L_0^{(\lfloor \frac{n}{2} \rfloor - 1)}|$, where $L_0^{(n)}$ is the set of all strings of length n which reduce to* 0.

Proof : We will prove the result for odd n. The result for even n is similar.

Let $n = 2k + 1$. Since n is odd any palindromic string x will have the following form,

$$x = a_1 \ldots a_k a_{k+1} a_k \ldots a_1$$

Now let us find the conditions under which x is reversible. We use (III) to get

$$
\begin{aligned}
&K_n(a_1, \ldots, a_k, a_{k+1}, a_k, \ldots, a_1) \\
&\quad = K_k(a_1, \ldots, a_k)\, K_{k+1}(a_{k+1}, a_k, \ldots, a_1) \\
&\qquad + K_{k-1}(a_1, \ldots, a_{k-1})\, K_k(a_k, \ldots, a_1) \\
&\quad = K_k(a_1, \ldots, a_k)[K_{k+1}(a_{k+1}, a_k, \ldots, a_1) + K_{k-1}(a_1, \ldots, a_{k-1})]
\end{aligned}
$$

So the condition for reversibility of x is the following

$a_1 \ldots a_k$ is reversible and exactly one of $a_1 \ldots a_{k+1}$ and $a_1 \ldots a_{k-1}$ is reversible.

Three cases are to be considered.

a) $a_1 \ldots a_{k-1} \to \epsilon$. Then $a_k = 1$ and $a_{k+1} = 1$. There are $|L_\epsilon^{(k-1)}|$ reversible palindromes of this type.

b) $a_1 \ldots a_{k-1} \to 0$. Then $a_{k+1} = 1$ and a_k can be either 0 or 1. So there are $2\,|L_0^{(k-1)}|$ reversible palindromes of this type.

c) $a_1 \ldots a_{k-1} \to 1$. Then $a_k = 0$ and $a_{k+1} = 1$. In this case we get $|L_1^{(k-1)}|$ reversible palindromes.

So the total number of reversible palindromes of length n is

$|L_\epsilon^{(k-1)}| + 2\,|L_0^{(k-1)}| + |L_1^{(k-1)}|$

$= 2^{k-1} + |L_0^{(k-1)}|$. $\square$

Now we can prove

THEOREM 4.1. *For $n \geq 3$, using n-cell null boundary 90/150 CA it is not possible to realize all reversible LFSMs.*

Proof : This is proved by showing that for $n \geq 3$, $A_n > B_n$. The above lemma gives the expression for $|D^{(n)}|$ in terms of $|L_0^{(n)}|$. Now, $|L_0^{(n)}| = 2^n - |S^{(n)}|$ and the value for $|S^{(n)}|$is already known from Theorem 2.3. Since A_n and B_n is expressed in terms of $|D^{(n)}|$, it is easy to find the expressions for A_n and B_n and check that indeed $A_n > B_n$. $\square$

5 References

[**1**]Berlekamp,E.R., Algebraic Coding Theory, New York: McGraw Hill, 1968.

[**2**]Graham,R.L., Knuth,D.E. and Patashnik,O., Concrete Mathematics, A Foundation for Computer Science, Addison Wesley, 1988.

[**3**]Golomb,S.W., Shift Register Sequences, Aegean Park Press, Laguna Hills, CA, 1982.

[**4**]Hopcroft,J.E. and Ullman,J.D., Introduction to Automata Theory, Languages and Computation, Addison Wesley, 1979.

[**5**]Hortensius,P.D., et al, Parallel pseudo random number generation for VLSI systems using cellular automata, IEEE Trans. on Comp., vol. 38, no. 10, pp 1466-1473, Oct. 1989.

[**6**]Martin O., Odlyzko A.M. & Wolfram S., Algebraic properties of cellular automata, Comm. Math. Phy, 93(1984), 219-258.

[**7**]Massey,J.L., Shift Register Synthesis and BCH decoding, IEEE Trans. on Inf. Th., vol. IT-15, pp. 122-127, Jan. 1969.

[**8**]Nandi,S. and Chaudhuri,P.Pal, Analysis of periodic and intermediate boundary 90/150 cellular automata, IEEE Trans. on Comp., vol. 45, no. 1, pp 1-11, Jan 1996.

[**9**]Serra M., et al, The analysis of one dimensional cellular automata and their aliasing properties, IEEE Trans. Comp. Aided Design of Circuits and Systems, vol. 9, no. 7, pp 767-778, July 1990.

[**10**]Sutner,K., Additive Automata on Graphs, Complex Systems 2(1988), pp 649-661.

[**11**]Wolfram S., Statistical Mechanics of Cellular Automata, Rev. Mod. Phy., 55(1983), 601-644.