Studies on Finite Linear Cellular Automata

Thesis submitted to the Indian Statistical Institute in partial fulfillment of the requirements for the award of the degree of Doctor of Philosophy

by

Palash Sarkar
Applied Statistics Unit
Indian Statistical Institute
203, B.T. Road,
Calcutta 700035.
INDIA
e-mail: palash@isical.ac.in

under the supervision of

Dr. Rana Barua
Stat-Math Unit
Indian Statistical Institute
203, B.T. Road,
Calcutta 700035.
INDIA
e-mail: rana@isical.ac.in

(Revised Version)

To my father.
He would have been very happy today.

ACKNOWLEDGEMENTS

Writing a thesis is a job that takes several years to complete. In this period I have been enriched in numerous ways by interaction with many people. At this time of acknowledgement, I would like to thank them all.

My supervisor Dr. Rana Barua has been a great source of inspiration. He taught me to look deeply into mathematical problems. His interest and curiosity have in many ways stimulated and shaped my approach to a problem. I feel really fortunate to have been able to work with him.

I was led into cellular automata by joining a project headed by Dr. Pabitra Pal Choudhury. I have shared many hours of discussions on CA with him. I thank him for providing me the opportunity to work on CA. I am grateful to Dr. Bimal Roy for reading all my papers and providing encouragement to carry on research.

My interest in mathematics was groomed at an early stage by Dr. T.K. Mukhopadhyay of Jadabpur University. Later motivation to theoretical computer science was provided by Dr. K. Sikdar, Dr. Rana Barua and Dr. S.M. Srivastava. Today I thank them and also all my other teachers.

In the early stages of my work, I had discussions with Paramartha Dutta who very willingly agreed to be a test board for many of my initial ideas. I thank him for this and also for being a good friend. I must also thank Samik Sengupta for providing me copies of several articles from journals which are not subscribed to by the ISI library. The computational and Latex work was carried out at CSSC and the ASU computer lab. I thank the technical staff of both places for providing and maintaining a healthy atmosphere.

Last, but not the least I must express my thanks to my family. My mother has always been a great source of strength. Sakuntala has been understanding throughout, even when "polynomials" seemed to fill my entire thoughts. Finally, I must thank little Torsha for just being herself.

Contents

Chapter 1

Introduction

Cellular Automata were originally proposed by John von Neumann as formal models of self reproducing organisms. The structure studied was mostly on one and two dimensional infinite grids, though higher dimensions were also considered. Computation universality and other computation theoretic questions were considered important. See Burks [24] for a collection of essays on important problems on cellular automata during this period. Later physicists and biologists began to study cellular automata for the purpose of modelling in their respective domains. In the present era, cellular automata is being studied from many widely different angles, and the relationship of these structures to existing problems are being constantly sought and discovered.

An important boost to the study of cellular automata was provided by Wolfram, in his experimental and theoretical studies conducted in the mid-eighties. It is in this period that finite cellular automata began to be studied seriously. The paper by Martin, Odlyzko and Wolfram in 1986 [115], is the first attempt to study a special class of finite cellular automata called additive (or linear) cellular automata. By using algebraic techniques, they were able to derive a large number of results on the state transition diagram (STD) of this class of cellular automata. In the process it became evident that algebraic techniques were going to play a pivotal role in the further exploration of linear cellular automata.

The next important step in the study of finite linear cellular automata is its application to designing VLSI structures. It has been argued that the regular and homogeneous nature of cellular automata are well suited for VLSI implementation and has spurred a great body of research in suggesting alternative cellular automata based structures for many practical applications. This work has almost exclusively concentrated on linear (or affine) cellular automata, since this kind can be tackled using linear algebra. On the other hand non-linear cellular automata are not amenable to known mathematical techniques and hence little work has been done towards finding VLSI application of this kind of cellular automata. However, the diversity of non-linear cellular automata holds out great promise. The work on application oriented aspects of finite linear cellular automata have led to some interesting results. On the other hand, finite linear cellular automata were also being studied from a more theoretical viewpoint and led to interesting algebraic techniques. The important object from both theoretical and practical viewpoints is the structure of the state transition diagram (STD). While VLSI applications concentrate on designing hybrid cellular automata whose STD have certain desirable properties, theoretical investigations fix a cellular automata and try to analyse the STD completely.

In this thesis, we take a more theoretical approach, though a VLSI implementable cellular automata based private key cryptosystem is proposed in Chapter 7. An important theme of the whole thesis is the study of the reversibility of different kinds of finite linear cellular automata. A major contribution of the thesis is to provide and in some cases develop algebraic foundation for the study of the above class of cellular automata on one or more dimensions.

1.1 Thesis plan

In the rest of the thesis we will abbreviate both cellular automata and cellular automaton by CA. We will consider different varieties of CA, but the exact structure meant will always be clear from the context. The thesis is based on [147, 145, 148, 146]. Results which have not been cited as appearing elsewhere are original to the best of our knowledge. Next we provide a brief summary of the chapters which appear in the thesis.

We begin the thesis with a review of CA research in Chapter 2. There is an excellent survey of CA by A.R. Smith, III [158]. However, it is more than two decades old. Currently, it is perhaps quite impossible to survey the whole of CA research. There is a good survey on computation theoretic aspects of CA [43]. There are also books on CA [64, 104]. In this survey we try to cover the major topics in CA research which are closer to computer science than physics or other applications. However, we would like to point out that any review of CA is bound to be incomplete. We have been motivated in choosing topics based on our knowledge and interest.

In Chapter 3, we provide the necessary preliminary material required in the later chapters. The structure studied in this chapter is uniform one-dimensional CA. The basic concepts of CA are introduced with reference to this structure, and in later chapters we only point out the modifications required for other varieties of CA. A major portion of the chapter concentrates on providing a detailed proof of the derivation of the minimal polynomial for null and periodic boundary one-dimensional CA. The results have already appeared elsewhere - for null boundary CA in [168, 151] and for periodic boundary CA in [168]. We provide a new simple proof of the result for null boundary CA. The motivation for providing a proof of the result for periodic boundary CA is twofold. The first is that it is an important result, and the proof in [168] is sketchy and depend on ideas scattered throughout the paper. The second reason is that the proof is a nice illustration of the kind of argument required in the analysis of linear CA. We also prove some new results on the inverse of the global map of null boundary CA. In analogy with the order of irreducible polynomials, we introduce the notion of exponent of a matrix and prove some interesting results on the exponent of the matrix representing the global map of one-dimensional null boundary CA. These results are used in Chapter 6.

In Chapter 4 we study hybrid CA, which is a CA where each cell has its own local rule. From the VLSI point of view, the hybrid 90/150 CA is the most studied structure. We provide an algebraic setting for the study of this structure in terms of continuant polynomials. This is a crucial connection which we exploit extensively to prove results on the reversibility of this particular variety of CA. In particular we show that the strings which encode reversible 90/150 CA (both null and periodic boundary), form a regular set. We use the regular expression for null boundary CA to count the number of reversible strings (both null and periodic boundary) of a fixed length. These results are then used to prove some negative results on the synthesis problem for this type of CA.

Uniform two-dimensional CA have been studied before by Sutner [161] and Barua and Ramakrishnan [15]. In Chapter 5 we tackle two open problems on two-dimensional CA proposed by Sutner in [161]. One of them is completely solved, while investigation of the other leads to some interesting results on the factorisation of a certain class of trinomials. Factorisation tables of such trinomials are presented in Appendix A. The analysis indicates that the reversibility problem for uniform two-dimensional CA on square grids is intimately related to the roots of a certain sequence of polynomials called π-polynomials, which are binary versions of the Chebyshev polynomials.

In Chapter 6, we tackle multi-dimensional CA. A key contribution is the representation of the linear operator as a sum of Kronecker products. Based upon this representation, a necessary and sufficient condition for reversibility is obtained, again in terms of roots of π-polynomials. Using this condition a lot of interesting results are derived. The characteristic polynomial of the linear operator is obtained in terms of resultant of π-polynomials. We also extend the results to different variations of multidimensional CA.

Lastly in Chapter 7 we provide an application of linear CA to private key cryptosystem. The advantage of using CA as a cryptographic primitive is the ease of implementation in VLSI. We introduce the notion of composite CA and construct suitable composite CA, having desired cycle lengths. A complete characterisation of composite CA is presented in a more formal setup of products of finite autonomous automata. A suitably constructed composite CA serve as the core of the block enciphering and deciphering hardware.

1.2 Prerequisites

It is assumed that the reader is familiar with basic linear algebra and polynomials over finite fields. Some introductory material on Kronecker products and resultants is provided in the Appendix. The reader is referred to [13] for Kronecker products and to [120] for resultants. All necessary material on finite fields is available in [110]. In Chapter 4 we use some basic ideas of finite automata and regular expressions, all of which can be found in [82]. The thesis requires no previous knowledge of CA.

Chapter 2

A Review of Cellular Automata Research

2.1 Classical

2.1.1 Beginnings

Cellular Automata (CA) were originally introduced by von Neumann. The simplest description of a CA is a one-dimensional array (possibly two-way infinite) of cells. Time is discrete and at each time point each cell is in one of a finite set of possible states. The cells change state at each clock tick, and the new state is completely determined by the present state of the cell and its left and right neighbours. The function (called the local rule) which determines this change of state is the same for all cells. The automaton does not have any input and hence is autonomous. The collection of the cell states at any time point is called a configuration or global state of the CA and it describes the stage of evolution of the CA. At time $t = 0$, the CA is in some initial configuration and henceforth proceeds deterministically under the effect of the local rule, which is applied to each cell at each clock tick. The application of the local rule to each cell of the CA results in a transformation from the set of all configurations into itself. This transformation is called the global map or global rule of the CA. This is a very simple description of a CA though it is perhaps the most studied structure. The automaton described by von Neumann is a two-dimensional infinite array of uniform cells, where each cell is connected to its four orthogonal neighbours. This was originally called a cellular space, but the term CA is more popular now. It was introduced by von Neumann [180] as a formal model of self reproducing biological systems. Key ideas of the construction can be traced back even earlier to his talk on modelling of biological systems [178]. The main purpose of von Neumann was to bring the rigour of axiomatic and deductive treatment to the study of "complicated" natural systems. The basic idea of a self reproducing automaton is presented in [178] and is a beautiful adaptation of the idea of constructing a universal Turing Machine (TM). Here we present a brief sketch of the idea.

First let us note that it is not very difficult to imagine the following two kinds of automata. The first kind is an automaton A which when given an instruction I can use it to construct an automaton (or machine) which is encoded by I. In fact I can be considered to be composed of simpler instructions, each of which is used to construct the basic parts along with instructions which specify how to put these basic parts together. The second automaton (say B) is even more simple. It copies an instruction I into the control part of some other automaton. Now consider A and B along with a control automaton C which operates as follows. Given an instruction I, C runs A to create an automaton A_1 corresponding to I and then runs B to copy the instruction I into the control part of A_1. Let D consist of A,B and C. Then clearly D is an automaton which require an instruction I to operate. Let I_D be the instruction which codes D. Let E be an automaton formed from D by copying I_D into the control portion of D. Now it is easy to see that E constructs itself and hence is capable of self reproduction. This simple description ignores the coding and other formal details. These were later formalized by von Neumann himself in [180], where he describes a cellular space where each cell can be in any one of 29 possible states. The structure is capable of non-trivial self reproduction in the sense that it can support a universal computer. The process of self reproduction can be visualized as follows [158]. Initially the machine is placed in an environment where in each direction there is any amount of hardware available (a "hardware soup"). Following the local rules the initial configuration goes through a sequence of steps, whereby it extends an "arm" into the hardware soup and creates a copy of itself and then detaches the newly created machine from itself. The original proof of von Neumann was simplified and reformulated several times [7, 11, 171]. The notion of self-reproduction introduced by von Neumann is asexual, in the sense that the offspring is derived from a single parent. In this form of reproduction the offspring is constructed from a single "genetic" tape which contains an encoding of the machine. Sexual reproduction have also been considered, and [176] contains a description of a machine which constructs an automaton from two "genetic" tapes, where the resulting offspring is not an exact copy of either parent.

It is important to note that a self reproducing machine is to be non-trivial in the sense of being capable of universal computation. Otherwise, a 1-d array with a single quiescent cell and a local rule copying this cell to the left and right neighbours can be considered to be self reproducing. This brings up the question of CA capable of universal computation and universal constructors. If a machine can construct a set of automata then it is called an universal constructor over this set. If this set contains the automaton itself, then it is self reproducing. Before we discuss the question of universal computation, we briefly mention the general problem of pattern replication.

Amoroso and Cooper in an interesting paper [5] have described 1-d and 2-d CA which after finitely many steps reproduces its initial pattern. The rule used is very simple. For 1-d it is the sum of the left neighbour and itself modulo k, where k is the number of states a cell can assume. For 2-d the rule is modified to include the neighbour vertically above the cell. A generalisation to higher dimensions is proved in [132]. Moreover, the pattern "reproduces" in a quiescent environment if k is prime. The CA rule used is linear and is one of the early examples of linear CA.

It is not very difficult to see that a CA is capable of universal computation. The basic idea is that a CA can perform a step by step simulation of a single tape Turing Machine (TM). For convenience assume that the tape of the TM is two way infinite. Each cell of the simulating CA will have two components. The first component stores the tape symbol of the corresponding cell of the TM tape and the second component indicates whether the head is scanning the corresponding cell of the TM. Then from the TM's transition function it is easy to derive the local rule for the CA. The essential idea is the following.

1. If the head is not scanning the cell or its left or right neighbour, the contents of the cell do not change.

2. If the head is scanning the left cell and there is a right move, then in the next step the head scans the present cell. Similarly for the other direction.

3. If the head is scanning the cell, then at the next clock tick, the contents of the first component of the cell is updated and the head no more scans the cell.

Note that this step for step simulation of TM by CA destroys the inherent parallelism of CA. There have been attempts to bring out the power of this parallelism [157]. Later work has shown how to simulate TM by reversible CA [55]. Albert and Culik [4] describes a universal CA A_U with 14 states which can simulate step by step any CA, whose initial configuration and local rule is encoded as an initial configuration of A_U. Computation universality of one-way CA and totalistic CA (See subsection 2.1.2) have also been proved [4, 43]. The problem of deciding whether a CA is computation universal based on the local rule is undecidable, since otherwise the problem of deciding whether a Turing machine is universal would be decidable.

An early technical question regarding CA was different kinds of trade offs - between the size of cell (number of possible states) and the size of the neighbourhood and between the size of cell and the speed of computation. The idea of trade off is an immediate consequence of reformulation of von Neumann's original proof of self reproducing machines. The original CA described by von Neumann used 29 states per cell. Codd [35] gave an 8-state machine. Arbib [7] provided a simple description where each cell can execute a short program - and hence the number of states per cell is large. Banks [11] provided a 4-state cell which could be used to build a self-reproducing CA. Each of these constructions is for 2-d infinite CA and uses the so called von Neumann or 5-cell (orthogonal ones and itself) neighbourhood. Generalisation of these trade off ideas to construction and computation universal machines is natural and has been studied in some depth. The simplest known construction universal machine with 4 states per cell and von Neumann neighbourhood is that of Banks [11]. He has also described the simplest known computation universal 2-d CA. (3 states per cell and von Neumann neighbourhood). However, for 9 cell or unit square neighbourhood (also called Moore neighbourhood), 2 states per cell is sufficient and a particular local rule called "Game of Life" (see Section 2.2.2) has been shown to be computation universal [158]. Smith [156] provides a list of neighbourhood size versus state set size trade off results for computation universal 1-d CA capable of self reproduction.

The other kind of trade off results is related to simulation of a CA by another CA which is a basic technique for proving results on CA. Specialization of such results to computation universal CA yields the results just described. It has been observed (but not proved) that the cost of reducing neighbourhood or increasing speed leads to an increase in the size of the state set. For a neighbourhood of M cells and n states per cell the size of the state set increases to about M^n when reduction is to Moore neighbourhood (a generalisation of the 9 neighbourhood for 2-d CA). Reduction of Moore neighbourhood to von Neumann neighbourhood is difficult and increases the state set size from n to n^V, where V is the volume (number of cells) in a d-dimensional sphere of radius $2d^{\frac{3}{2}}$ [156]. For 2-d and 3-d case this cost can be significantly reduced [25, 75]. Simulations can be carried with neighbourhoods smaller than von Neumann. For example, a neighbourhood consisting of the cell itself and a neighbour in each dimension suffices for a step by step simulation of an arbitrary CA. In fact, the cell itself can also be left out [156]. If a strict step by step simulation is not required then the initial encoding may be omitted and the CA can itself perform the initial encoding. The reverse trade off - decreasing the state set size by increasing the neighbourhood is also possible [156].

Given a CA it is possible to design another CA which simulates the given CA k times faster at a cost of increase of state set size, assuming Moore neighbourhood before and after simulation [156]. Both decrease in neighbourhood and speed up can also be achieved at a cost of increase in the state set size. But there seem to be no theoretical results on the limits of trade off possible. For example, assuming finite neighbourhood, what is the maximum speed-up possible at a cost of increase in state set size ? Investigation of this and similar questions can lead to interesting results.

2.1.2 Variants of Cellular Automata

A CA is characterised by four features - the geometry of the underlying medium which contain the cells, the local transition rule, the states of the cell and the neighbourhood of a cell. In the following paragraphs we briefly discuss different types of CA that can arise by varying the four features mentioned above.

Geometry

This can be a d-dimensional (possibly infinite) grid. Usually the term CA is used for such structures. In case of finite grids it is possible to define different boundary conditions. The grid is supposed to have a periodic boundary condition in some dimension if it is considered folded in that dimension. The dimension has a fixed boundary condition if the extreme cells are considered to be adjacent to cells in some prespecified state whose value does not change during the computation. In case this prespecified state is the quiescent state, then the boundary condition is called null boundary condition. For linear CA, the quiescent state is the state zero and in general the quiescent state is a state q such that the local rule maps $(q, \ldots, q)$ to q. Note that it is possible for more than one such state to exist but usually one particular state is defined to be the quiescent state. Among the fixed boundary conditions only the null boundary condition have been studied seriously. However, see [115] for a brief discussion of other possibilities. It is also possible to consider one end to have periodic boundary condition and the other end to have fixed boundary condition [12].

A more abstract way of defining the geometry is through group graphs. The following definition is from [77]. A group graph is a tuple $N = (G, h)$, where G is a group which defines the nodes for the cells and h defines a map from G to G^k by $h(g) = (h_1.g, \ldots, h_k.g)$, where $h_i \in G$ and . is the group operation. The map h provides the neighbourhood for the cells. The concept of group graph is a convenient way to describe "uniform" geometry - a connection pattern which "looks same" at all points. Non-uniform connections have also been studied, though the relation between uniform and non-uniform geometry have not been fully understood (see [94]).

So far we have considered, what is called static CA - the node set and the interconnection pattern do not change with time. It is possible to consider node static CA where the node set does not change with time but the interconnection pattern may change. Such a structure is still considered static and has not received much attention (see [175]). However, dynamic CA - both node set and connections may change - have been studied extensively due to its use in modelling of biological systems.

Neighbourhood

In some cases like group graphs the geometry itself determines the neighbourhood of a cell. However, if we are considering a d-dimensional grid it is possible to define different kinds of neighbourhood. The von Neumann (orthogonal) neighbourhood and the Moore (unit cube) neighbourhood have already been mentioned in connection with the trade off results. It is possible to define input and output neighbourhoods of a cell. A cell takes its input from its input neighbourhood and its state is available to the cells of its output neighbourhood. If the sizes of the input and output neighbourhoods are equal, then the CA is balanced. For balanced but non-uniform neighbourhoods, the connection to uniform neighbourhood has been studied in [94]. A variant of CA where the local rule depends on the sum of the states of the neighbouring cells is called totalistic CA and was introduced by Wolfram. Computation universality of this kind of CA have been proved in [43].

Cell States

A CA where the cells can have different state sets is called a polygeneous CA. Such CA have not received much attention except for the work of Holland [80].

Local Rule

The local rule is usually assumed to be deterministic. This however is not necessary and non-deterministic maps have been studied in connection with language theory [157, 150] and reliable computation [131]. A CA where each cell has its own local rule is called hybrid. Such structures have been studied in connection to VLSI applications [151, 31, 86]. It is possible for a cell to change its local rule at each time step. In the VLSI context, this is called a programmable CA [128] and in theoretical studies on CA the structure has been called a tessellation automata.

Next we discuss three variants of CA which have received more attention.

Tessellation Automata

This is a CA with an input line distributed to all cells. The setup can be visualized as each cell having a finite set of local rules and the input is used to choose the particular local rule to apply. See [195, 197] for a nice discussion on tessellation spaces. An interesting problem which is inherently tessellation automata theoretic is the completeness problem and is related to the "Garden of Eden" problem for CA. The problem is stated as follows. Starting from an initial configuration with only one non quiescent state, is it possible to apply input to drive the automaton to any specified finite configuration ? If the answer is yes for some subclass of automata, then the subclass is called complete. There are only partial answers to this question [196, 116, 117]. Tessellation automata have also been called time varying CA and their formal language theoretic properties have been studied [114].

Iterative Automata

This is a CA where only one particular cell is given an input. Such structures have been considered in connection with language recognition studies [103, 150, 29]. Different trade off results (similar to CA) for this class have been considered in [36]. In [157] it is shown that this class is an inherently slower device than usual CA. Iterative automata languages contain the context free languages [103]. A 1-d iterative automaton requires $O(n^2)$ steps to accept a string of a CFL of length n. The non-deterministic 2-d version of iterative automata can accept in linear time any language accepted in linear time by a non-deterministic multi-head TM with a tape of arbitrary dimension [150]. The paper also contains the result that the non-deterministic d-dimensional iterative spaces can accept in linear time any language accepted in time n^d by a non-deterministic multi head TM but with a 1-d tape. See [90] for additional results.

An interesting application is a linear time multiplier designed by Atrubin [9]. The binary representation of the multiplicands are fed to the first cell (least significant digit) first and the product is output from the first cell (again least significant digit first) with no delay. See [102] for a good exposition of the algorithm. Iterative linear arrays have also been used in VLSI applications [105].

Note that the concepts of tessellation and iterative automata can be generalised to tessellation and iterative graph automata by defining such structures on group graphs [158].

One Way CA

A one way CA allows only one way communication, i.e., in a 1-d array each cell depends only on itself and its left neighbour. One can also consider dependence on the cell and its right neighbour. However, both side dependence is not allowed. This lack of two way flow of information can be considered to be a restriction on the power of the automaton. However, there are results which indicate otherwise. Morita [125] has shown the computation universality of 1-d, one way reversible CA. See Subsection 2.1.5 for formal language properties of this class of CA. A related class of automata motivated by design of systolic systems and algorithms is the class of systolic trellis automata, which have been quite extensively studied by Culik et al [23, 26, 40, 41, 42]. This class is equivalent to bounded space real-time one-way CA. Study of systolic arrays modelled as 1-d, 2-d, one-way CA and iterative arrays have been carried out by Ibarra et al [89, 90]. This work has resulted in the development of many easy-to-implement systolic algorithms . One way CA on Cayley Graphs have also been studied [144].

2.1.3 Biological Connection

CA were originally proposed by von Neumann to provide a formal framework for study of "complicated" natural systems. Later work in this direction used dynamic CA for modelling of biological systems. One of the early attempts was by Lindenmayer [111], who proposed a model of growth for filamentary organisms based on ideas of CA. Later work on such systems were mainly formal language theoretic and a survey appears in [98].

Another interesting biological connection has been studied by Holland [81]. He used CA as a model to study the spontaneous emergence of self-replicating systems. The CA is used as a model of the universe (called the α-universe) where each cell has two parts. The first part stores the state of the cell and the second part indicates the nature of the bond (strong or weak) the cell has with its left or right neighbours. Stochastic operators are used to manipulate the states in accordance with the bonds and in a conservative manner - elements are never created or destroyed, they are only moved about and rearranged by the operators. The operators are themselves encoded by the states of the cells. The crucial parameter studied is the expected time till the emergence of self replicating systems, which is an arrangement of the universe which can replicate itself.

The first attempt at modelling artificial life with CA was von Neumann's self reproducing automata. An implementation of this construction has been done in [137]. For other work on modelling of artificial life see [91, 1]. A great amount of work has been done in using CA for modelling of biological systems. Examples of recent work in this direction can be found in [108, 101].

2.1.4 Fault Tolerant Computing

The idea of fault tolerant computing also originates from von Neumann [179], who showed how to build a reliable Boolean circuit out of unreliable components. For the case of CA, the unreliable components are taken to be the cells. Each cell can misoperate and assume an incorrect state, i.e one not dictated by the local rule. Early work in this area assumed a fault model called k-separated misoperation [131], i.e, there exists a finite set K of Z^d such that given a cell $x \in Z^d$ at most one cell in the set $x + K$ will misoperate (here d is the dimension of the grid). In [131] it is shown how to construct a CA which will correctly simulate an unreliable CA with k separated misoperation, step for step. The basic idea is to encode the initial configuration of the unreliable automaton suitably to form the initial configuration of the simulating automaton. The coding is carefully designed so that each cell in the coded configuration can use a majority voting rule to decide its state. The local rule of the simulating automaton is almost the same as the original one, except that at each step each cell of the simulating automaton corrects any error in its neighbouring cells before applying the local rule. This leads to an increase in the neighbourhood size. It has been shown that under the same fault model, unreliable CA over group graphs can also be simulated in an error free way [76].

Later Gacs [60] has shown how to construct a 1-d CA which can reliably perform arbitrarily large computations, and where each cell can perform an error with a positive probability. The fault model so considered is important from ergodic theory point of view and Gac's result leads to the refutation of the "positive probability conjecture" in statistical physics, which states that any one dimensional infinite particle system with positive transition probabilities is ergodic. For recent work on reliable cellular automata see [61].

2.1.5 Language and Pattern Recognition

A finite CA can be thought of to be a language acceptor by considering the initial configuration to be the input string and acceptance or rejection is determined by a specific cell (say the rightmost) going to an accept or reject state. For a 2-d CA the problem is one of pattern recognition and the accept cell can be the northeast one in a rectangular grid or it could be the easternmost cell in the northernmost row for a general 2-d layout. Certain language classes can be defined by both restricting and enhancing the power of CA. This is done by introducing the following four conditions.

1. One way communication giving rise to *one way CA*.
2. For an input of n symbols, the number of steps of computation required is exactly n. This is called *real time* computation.
3. For an input of n symbols, the number of steps of computation is proportional to n. This is called *linear time* computation.
4. The local rule is *nondeterministic*, giving rise to nondeterministic CA.

The symbols O,r,l and N are used as prefixes to the word CA to denote a particular language class. As an example, rOCA denotes the class of languages accepted by real time one way CA. The relationships among CA language classes as well as their relationship to the classical language classes have been extensively studied. See [114] for a good survey of results and techniques used in this area. Here we briefly mention several important results. The first (and easy) result is that the language class CA is equal to DSPACE(n). The class lCA is a subset of OCA [29, 88, 90]. This is obtained by considering the relationship of both OCA and lCA to sweeping automata [29]. It is also known that rOCA is a proper subset of rCA [26, 40], and rCA is equal to lOCA [26]. The PSPACE-complete language QBF (quantified Boolean formulae) belongs to OCA [88] and NSPACE($\sqrt{n}$) and ATIME(n) are subsets of OCA. The class OCA lies between NSPACE($\sqrt{n}$) and CA=DSPACE(n) and proper containment between OCA and CA would separate these two classes, improving Savitch's result. It is also conjectured that lCA is properly contained in OCA, since lCA is a suset of P and OCA contains QBF, any proof that lCA=OCA will imply that P=PSPACE, a rather unlikely result. For the nondeterministic language classes it has been proved in [58] that NOCA=NCA=NSPACE(n), the class of context sensitive languages. Further it is known that rNOCA contains an NP-complete problem [89]. Open problems and examples of languages contained by rOCA, rCA, lCA and OCA can be found in [114].

2.1.6 Invertibility and Garden of Eden

A major focus of research in CA has been related to question of invertibility. A CA rule ρ is called invertible if there exists another rule ρ^{-1}, called the inverse rule, which drives the CA backward, i.e., if application of ρ to a configuration c produces a configuration d, then application of ρ^{-1} to d produces c. A CA is called invertible if its local rule is invertible. Richardson [142] has proved that a CA is invertible iff its global map is injective. The technique does not provide an inverse as topological arguments are used to prove the result. For an automata theoretic approach to the problem see [37]. Amoroso and Patt have proved that there is an effective procedure to determine invertibility of 1-d CA based on the local rule [6]. Kari [95, 97] has shown that for a 2-d CA the question of determining invertibility from the local rule is undecidable. The reduction is from the tiling problem in conjunction with a special version of the tiling problem called the directed tiling problem.

The surjectivity of the global map of a CA have also been studied. A configuration is called a "Garden-of-Eden" configuration if it is not "reachable", i.e it can occur only as initial configuration in any evolution. Existence of such a configuration shows that the global map is not surjective. Myhill and Moore [127] have proved that a global map is surjective iff its restriction to finite configurations is injective. The surjectivity of 1-d CA is decidable [6]. Kari [95, 97] proves that the problem is undecidable for two dimensions by showing that the injectivity problem restricted to finite configurations is undecidable. To tackle finite configurations Kari introduces a special class of tilings having the "finite tiling property". See [95, 97] for details.

Given a 1-d CA, it is possible to construct an invertible 1-d CA which can simulate the original CA [126]. It is even possible to simulate TM by invertible CA [55]. Toffoli [172] has shown how to simulate any k-d CA by an invertible $(k+1)$-d CA. This proves the computation universality of invertible CA for dimensions higher than one and from the result of [126] 1-d invertible CA is also capable of universal computation. However, the question of whether a k-d CA can be simulated by a k-d invertible CA is still open for $k > 1$. The invertibility question is of fundamental importance to physics, as it can be used for modelling microscopically reversible dynamical systems. See [174] for a survey.

For a finite CA, an injective global map has to be bijective. Moreover, if the global map of a finite CA is injective it does not necessarily mean that there is an inverse CA, in the sense that there is an inverse local rule that can be used to force a configuration to retrace the original evolution. So a finite CA is said to be invertible if the global map is a bijection. In this case it is trivial to see that the non-existence of Garden-of-Eden configuration is a necessary and sufficient condition for invertibility of the global map. It is in general difficult to determine invertibility of finite CA. See [77] for a discussion of the dynamics of finite CA. If the global map is a linear transformation, then the problem becomes more manageable. Extensive discussion on properties of linear or additive CA can be found in [115]. In this thesis we will discuss the invertibility of different kinds of linear CA on one or more dimensions.

2.2 CA Games

2.2.1 Firing Squad Problem

This is basically a synchronization problem but can also be thought of as a game. The problem was first proposed by Minsky around 1957 and it first appeared in print in [123]. The following is a simple description of the problem. Consider n soldiers (out of which one is a general) to be standing in a row. The soldiers (including the general) can communicate only with their immediate left and right neighbours. The general gives the command to fire. Ultimately the soldiers and the general are required to fire simultaneously and for the first time. In CA terms the problem is to design a cell and a local rule such that starting from an initial configuration where only one cell is on and the other $n-1$ cells are off, there is an evolution such that all the cells enter a predesignated state all at once and for the first time. Note that the problem can also be considered on an infinite 1-d array, but then the other cells must all be in the quiescent state and remain so throughout. The basic problem is to design a cell which is independent of the number of soldiers and hence will work for an array of arbitrary length. This means that none of the cells can count upto n. In case the general is one of the end cells, it is easy to see that the minimum time required for synchronization is $2n-2$ steps. Waksman [181] provides a solution in $2n-2$ steps. The solution depends heavily on the idea of signals propagating through the array at different speeds. Implementing such signals using CA cells are discussed in [39]. For a solution to the problem where the general can be any cell see [124]. Culik [38] has considered several other variations and have used the results to disprove a conjecture of Ibarra and Jiang that real time one way CA cannot accept certain languages. The problem has also been generalised to higher dimensions [130, 154] and node static and dynamic CA [79, 175]. A generalisation to arbitrary graphs called the Firing Mob Problem have been introduced in [39] where an efficient solution is also provided. The introduction to [39] also contains a brief history of the Firing Squad Problem and the solutions attempted by various researchers. The central result that it is possible to design such a CA is called the firing squad theorem and has been used in language and pattern recognition studies of CA [157, 38]. A related "desynchronization" problem is to design a CA such that all cells are initially in the same state and ultimately only one cell goes to a predesignated state. The problem has been called the "queen bee" problem [158].

2.2.2 Game of Life

This game was originally proposed by Conway and has been made popular through Martin Gardner's column in Scientific America [62, 63]. The original motivation for the problem was to design a simple set of rules to study the macroscopic behaviour of a population. The criterion for choosing the rules were based on the principle that growth or decay of the population should not be easily predictable. After a great deal of experimentation, Conway chose the following set up. The population is represented by a configuration of a 2-d infinite array of cells with Moore (unit square) neighbourhood, where each cell can be in one of the states 1 or 0. The local rule is described by the following rules. The resulting CA is an example of totalistic CA (see Subsection 2.1.2).

1. Survival : If a cell is in state 1 (alive) and has 2 or 3 neighbours in state 1, then the cell survives, i.e remains in state 1.

2. Birth : If a cell is in state 0 and has exactly 3 neighbours in state 1, then in the next time step the cell goes to state 1.

3. Deaths : A cell in state 1 dies (goes to state 0) if it has 0 or 1 neighbour (loneliness). Also it dies if it has 4 or more neighbours (overcrowded).

Each configuration is called a population and the evolution of the population is studied. As with many CA evolutions, the "Game of Life" show fantastic variation in the growth patterns of the initial population. It has been shown that there is a simple initial configuration, that grows without limit. The configuration grows into a "glider gun" and after 40 steps fires the first "glider" and thereafter continues firing gliders after every 30 moves. It has been informally proved that the "Game of Life" is capable of universal computation. For a good account of the game and for some good pictures see [62, 63].

2.2.3 $\sigma(\sigma^+)$-Game

This game was first proposed by Sutner [165] and is based on the battery operated toy MERLIN [136]. It is a two person game and is played on a 2-d finite grid where each node has a bulb which can be either on or off. A move is made by choosing a node and as a result the states of all the bulbs in orthogonal neighbourhood positions toggle. A configuration of the game is a state of the grid where some of the bulbs are on and the others are off. Player A chooses two configurations, the initial and the target configurations. Player B has to make a sequence of moves starting from the initial configuration and reach the target configuration. It is easy to see that choosing a node twice is the same as not choosing it at all. Also the order of the choice of nodes is not important. Thus any winning strategy (solution) for B can be viewed as a set rather than a sequence. This set of nodes can then be thought of as a configuration of the grid (the bulbs in the set are on, the others are off). Suppose the initial configuration is the all 0 configuration and the target configuration is X_t. If Z is a solution to this instance then $\sigma(Z) = X_t$, where σ is the global rule of a finite 2-d

CA whose local rule is the sum (modulo 2) of the four orthogonal neighbours. Again Z is a solution for the pair (X_s, X_t) iff $\sigma(Z) = X_s + X_t$ and hence the number of solutions (if any exists) is 2^k, where k is the corank of the linear map σ. Thus the study of σ-game reduces to the study of linear 2-d CA [15, 166]. The corresponding game where the state of the chosen bulb also changes is called the σ^+-game. Both σ and σ^+-game have been studied on 2-d and multidimensional grid. In fact this thesis will present results on multidimensional CA which have direct relevance to multidimensional $\sigma(\sigma^+)$-game. The game has also been considered over arbitrary graphs [161, 163] but results are more difficult to obtain in this setting.

2.3 Modern Research

2.3.1 Empirical Study

The mid-eighties form an important period in the history of CA and this is largely due to the work carried out by Wolfram. The nature of the questions asked represent a paradigm shift in CA research. Wolfram carried out an extensive phenemological analysis of the growth patterns of CA. The early paper by Wolfram [189], discusses several statistical parameters of the space-time patterns of CA evolution. Later work extended and clarified much of the intuition in several directions. An excellent source of papers on this period of CA research is the book by Wolfram [193]. The major viewpoint was to consider CA as models of complex systems, in the sense that very simple CA rules can give rise to extremely complicated patterns. The mathematical simplicity in CA description is thought to be a significant advantage in using CA for modelling rather than using systems of differential equations. A related phenomenon of CA evolution is self-organization. Starting from random unordered configurations having maximum entropy a CA evolves to states of lesser entropy. This is contrary to the second law of thermodynamics which states that reversible systems evolve to states of maximal entropy. The microscopic irreversibility of CA is the reason behind this selforganizing behaviour. The type of CA extensively studied by Wolfram is the 1-d, 3 neighbourhood, binary CA. A numbering system for the possible local rules of such CA can be found in [193]. Two important rules are 90 and 150. Rule 90 is the sum modulo 2 of the states of the nearest two neighbours. Rule 150 is the sum modulo 2 of the states of the nearest two neighbours and the state of the cell itself. Note that both 90 and 150 are linear rules.

The approach taken in [191] to study the growth patterns of CA was to define several local and global statistical parameters and study their behaviour. Some important local parameters are

1. average density of non-zero sites, which is a "rough" measure of the growth of CA evolution.

2. the average number of triangles or triangle density $T(n)$ of triangles of base length n, in the space time pattern.

3. sequence density $Q_i(n)$, which is the density of sequences of exactly n adjacent sites with the same value i.

Both the triangle and the sequence density follow an exponential rule for evolution from initial disordered state. For example, for large n, $T(n) \sim \lambda^{-n}$ and the parameter λ distinguishes between linear ($\lambda \approx 2$) and non-linear ($\lambda \approx \frac{4}{3}$) rules. Another important feature of the space time evolution from initial disordered state is that triangles of all sizes are obtained and hence the structure is generated on all scales.

For a finite N-cell CA one can consider the finite set of 2^N configurations to be an ensemble where each configuration have equal probability of occurrence. After evolution for a few time steps an equilibrium is achieved where the configurations have different probability according to some distribution function. On taking average over the ensemble, properties of configurations with higher probability dominates. This indicates the self organizing character of CA evolution. Another measure of self organization is entropy. For a finite CA, the entropy is defined as $\sum p_i \log p_i$, where p_i is the probability of configuration i. For irreversible CA, this entropy decreases from an initial maximum (for random initial configuration) to lesser values. A corresponding entropy called "block" or "Renyi" entropy can be defined for infinite 1-d CA and shows a similar phenomenon. For second order (next state depends on present and previous state of neighbours) reversible infinite CA, the entropy almost always increases with time.

Another interesting approach to characterisation of CA evolution comes from formal language theory. It has been shown in [190] that the set of configurations that can appear after t time steps forms a regular language. The size of the minimal DFA after t steps provides an indication of the complexity of the set of configurations after t steps. For many CA rules, the minimal DFA becomes more complicated at each step and do not appear to exhibit any overall structure. Again for some CA rules the infinite limit set of configuration (the set of configurations reachable at arbitrarily large time steps) is also a regular language, but there are others whose regular language complexity grows with time and hence seem to generate non-regular language in the limit. In fact Hurd [87] has provided examples of CA having strictly non-regular, non-context free and non-r.e. limit sets. In [69], a CA is described whose limit set is NP-hard. A modification to this approach associates to each node of the minimal DFA a weight corresponding to the probability P_i that they are visited. One then computes the entropy measure $\sum P_i \log P_i$ and uses it to study the growth pattern of the configurations. For details of this approach see the Appendix (Table 11) of [193].

2.3.2 Classification of CA

A major problem that stemmed from Wolfram's work is one of classifying CA rules according to their behaviour. The initial empirical classification was proposed by Wolfram himself in [191]. His classification is based on entropy measures and identifies the following four classes.

1. Evolution leads to a homogeneous state.

2. Evolution leads to a set of separated simple stable or periodic structures.

3. Evolution leads to a chaotic pattern.

4. Evolution leads to complex localized structures which are sometimes long lived. It is believed that this class is capable of universal computation.

Later work concentrated on formalizing the intuitive classification by Wolfram. Culik and Yu [46] have proposed the following classification. Let ρ be the local rule for a CA. Then,

1. Rule ρ is in class one iff every finite configuration, i.e configurations in which only a finite number of cells are in non-quiescent states, evolves to a stable configuration in finitely many steps.

2. Rule ρ is in class two iff every finite configuration evolves to a periodic configuration in finite number of steps.

3. Rule ρ is in class three iff it is decidable whether a configuration occurs in the orbit of another.

4. Class four comprises all local rules.

They show that the problems of deciding membership of a rule ρ in classes one and two are Π_1^0-hard. Similarly class three is Σ_1^0-hard.. Sutner [164] has shown that class one and two are Π_2^0-complete and class three is Σ_3^0-complete. The arguments are based on encoding of TM instantaneous descriptions by natural numbers and the simulation of TM by CA. It is important to note that the above classification considers only finite configurations. Infinite configurations in general cannot be finitely described and hence cannot be tackled by conventional computability theory. A classification of periodic boundary condition CA (whose configurations can be thought of as spatially periodic configurations of an infinite CA) have also been proposed [165]. Using a non-standard simulation of a TM by a CA, it is shown that the problem of deciding membership in the hierarchy is undecidable.

In a recent study Braga et al [22], have provided a classification of CA based on their pattern growth. The pattern growth properties are show to be dependent on the truth table of the local rule of the corresponding CA. This provides an algorithm for classification of CA rules and hence defines an effective hierarchy of CA rules, which is in sharp contrast to the undecidability results discussed above. The essential technique is the fact that certain shift like dynamics in the evolution can be discovered by looking at the truth table of the local rule. Then a proper grouping of rules exhibiting similar dynamics yields a classification which is close to that of Wolfram's.

Other attempts at classification have been reported. Gutowitz [72] provides a hierarchical classification of CA based on action of CA on n-step Markov measures. There is also an algorithm which efficiently constructs all rules in a given class at a given level of the hierarchy. Gilman [65] has introduced a dynamical system based classification of CA. See [43] for additional details on various classification schemes.

A preliminary study of 2-d CA [133] shows that it is possible to classify 2-d CA along the same lines as 1-d CA. This suggests that the global behaviour of 2-d CA is similar to

1-d CA. However, 1-d and 2-d CA show marked difference with respect to other properties. Golze [67] has shown that for 1-d CA every recursive configuration (a configuration each of whose cell values can be effectively calculated) has a recursive predecessor but in the 2-d case even a finite configuration may fail to have a recursive predecessor. Again invertibility of 1-d CA is decidable while it fails to be so for 2-d (and higher dimensional) CA.

2.3.3 Limit Sets and Fractal Properties

One of the important directions of CA research in the modern era is the study of the limit sets of CA space time patterns. Early work in this area was done by Willson [184, 185] and the topic received an impetus from Wolfram [189, 191]. However, the notion of a limiting set of configuration obtained by evolving a CA was introduced by Podkolzin [140]. Later we mention some of the work done in this area. The space time pattern that is observed during simulation shows several kinds of interesting characteristics (see appendix of [193]). One of the important features is a sort of scale invariance and self similarity on different scales. This immediately suggests the idea of computing the fractal dimension of such patterns. Wolfram's empirical investigation [189] outlines two natural ways to do this. In the first approach a parameter $T(n)$ is defined which measures the density of triangles of base length n. A geometrical construction shows that for rule 90, $T(n) \sim n^{-1.59}$ and for rule 150, $T(n) \sim n^{-1.69}$. The invariants 1.59 and 1.69 then gives the limiting fractional dimension of the patterns. In the second approach, the space time configurations are scaled to fit the same perimeter and one considers the set of all limit points. This gives rise to a fractal dimension which is a "geometric" dimension and is also called the Kolmogoroff dimension. Willson [187] investigates theoretically why the two approaches to computing dimension should coincide and provides examples where the Kolmogoroff dimension differs from the more usual Hausdorff-Besicovitch dimension.

Theoretical study of the limit sets of CA evolution via geometric invariants have been performed by Willson [186]. The basic object of study is the sequence

$$\omega, F\omega, F^2\omega, \ldots, F^p\omega, \ldots,$$

where ω is a configuration of an n-dimensional CA and F is the global rule of some CA. If we fix a state q, then we can think of the set of cells (in space time configuration) having value q as a set of points where each point is given by an $(n+1)$-dimensional vector. Let X_p be the above set corresponding to $F^p\omega$. Consider the set X_p/p where the vectors of X_p/p are obtained by dividing each vector of X_p by p. This scaling ensures that the space time configurations fit the same perimeter at each time step. Let $Lim(\omega, q)$ be the set of points in the limit $p \to \infty$. This limit is taken as an approximation of X_p and properties of the limit indicate the nature of growth pattern of the space time configuration. For example if $Lim(\omega, 1)$ is a tetrahedron then one would expect the configurations to grow into a tetrahedral form. When the CA rule is linear (mod 2) it has been shown that the limit set is a compact subspace of Euclidean space and can have fractional Hausdorff dimension. For linear CA this provides a formal proof of Wolfram's basic intuition. Space time patterns of arbitrary linear CA have also been studied [170]. The corresponding limit sets are generally fractals. The self similar structure is characterised by a transition matrix, whose maximum eigen value determines its Hausdorff dimension.

Limit sets have also been studied from a different direction using formal language theoretic methods [87, 45]. In this approach the set of configurations rather than the space time patterns is considered. For a d-dimensional infinite CA having S as the set of cell states, the set of configurations is S^{Z^d}. When S is endowed with the discrete topology then S^{Z^d} with the product topology is compact by Tychonoff's theorem and the global map G of the CA is a continuous function. Letting $S^{Z^d} = \Omega_0$ and $\Omega_i = G(\Omega_{i-1})$ for $i \geq 1$, each Ω_i is a compact subspace of S^{Z^d} and $\Omega = \bigcap_{i \geq 0} \Omega_i$ is the limit set for the CA. This Ω is the object of study. It has been shown in [45] that for $d \geq 2$, it is undecidable whether Ω contains a finite configuration. Using the notion of limit set of a CA it is possible to define a limit language as follows. Consider a 1-d CA. Then every configuration is a biinfinite word over S. For a configuration c, define

$$L[c] = \{w \in S^* : w \text{ is a finite subword of } c\}$$

and let $L[C] = \bigcup_{c \in C} L[c]$ for a set of configurations C. Then $L[\Omega]$ is the limit language. The membership problem for such limit language is undecidable [45]. For a survey of result regarding this limit language see [43]. Given a CA the complement of the limit language is r.e. [43]. Also for any language whose complement is r.e., one can construct a CA whose limit language yields the chosen language after intersection with a regular language and a ϵ-limited homomorphism. This can be used to show that there exists a CA whose limit language is not r.e. Similar properties have been obtained for Π, the closure of the points periodic under the global CA map. See [43, 44] for details.

One can define a State Transition Diagram (STD) for an infinte CA by considering an infinite directed graph whose vertices are the configurations of the CA and the edges represent one step evolution of the CA. This has been done by Podkolzin [140], where it has been shown that the STD either has a single connected component or has uncountably many connected components. If a CA has only one single connected component it is called nilpotent. It has been proved in [140] that for two or more dimensions the problem of CA nilpotency is undecidable. For one dimension the same result has been proved by Kari [96]. Podkolzin [140] has also shown that for any CA either the limit set is a singleton and the CA is nilpotent or the limit set contains an infinite number of elements. See [43] for further discussion on limit sets.

Another interesting approach to the study of dynamical properties of CA is to consider the CA as a computational device acting on bi-infinte strings on one hand and as a continuous function on a compact metric space on the other. This gives rise to considerations of symbolic dynamics on bi-infinite strings [17]. If S is the state set for a cell of a 1-d CA and Z is the set of integers then S^Z is the set of all configurations of the CA. It should be noted that if G is a global CA map then it is a shift invariant continuous map from S^Z to S^Z. The converse that any shift invariant continuous map from S^Z to S^Z arises as a CA map has been proved by Hedlund [78]. A topologically closed subset of S^Z is called a subshift if it is invariant under the shift map. A subshift is said to be of finite type if any bi-infinite word in it does not contain any block from an excluded finite set. A sofic system is the image of a shift invariant continuous map acting on a subshift of finite type. It has been shown that each sofic system is a $\omega\omega$-regular set and for each $i \geq 0$, $G^i(S^Z)$ is an $\omega\omega$-regular set [47], where G is the global map of a CA. See [47, 43] for a more detailed discussion.

2.3.4 Computational Complexity

Early indication of the study of computational complexity of CA is the study of the minimum number of steps required to perform certain computation. Serious attempts at studying complexity theoretic questions regarding CA is a later development. Wolfram [190] shows how to construct a graph to represent configurations reachable after one time step of a 1-d CA. All possible infinite paths through the graph represent all possible configurations. The notion can be generalised to finite number of time steps and also to limit sets. The graph can be regarded to be the state transition graph of a finite automaton which may be non-deterministic. The equivalent minimum state DFA can be constructed and the number of states in such a DFA provides a measure of the complexity of the corresponding configuration set. For some interesting properties of this measure see [193]. Another way of looking at this problem is to view a CA configuration as a bi-infinite word. Then the set of configurations reachable in one time step is a sofic system and from the results of [47], it is an $\omega\omega$-regular set. A consequence of the above result is that the predecessor existence problem (PEP - given configuration X : does there exist a configuration Y, such that Y evolves to X in one time step ?) for 1-d CA is decidable. Note that all configurations must be finite, since infinite configurations cannot be tackled by ordinary computability theory.

This lead to a more formal study of the computational complexity of CA. In particular, it was an important question to find NP-complete problems for CA. First such results appear in [69] where a CA is constructed for which the following problems are NP-complete.

- determining if a given subconfiguration s can be generated after $|s|$ time steps.
- determining if a given subconfiguration s will recur after $|s|$ time steps.
- determining if a given temporal sequence (values of a particular cell taken over time) of states s can be generated in $|s|$ time steps.

The particular CA described is quite complicated since an arbitrary structure of the 3-SAT problem has to be encoded in the essentially local communication mechanism of a CA. For an infinite CA, certain problems [162] such as configuration reachability (CREP - source configuration X; target configuration Y; is Y reachable from X ?), PEP, are undecidable. Undecidability of CREP is easy to see since a CA can simulate a TM and configurations of the CA encode instantaneous descriptions of TM. Hence the halting problem for TM can be translated to CREP by asking whether a halting configuration is reachable from the initial configuration. In fact CREP is Σ_1^0-complete for infinite CA of any dimension. However, for

PEP there is a marked difference for the 1-d and higher dimensional CA. From Wolfram's characterisation of 1-d CA using regular grammars [190] it follows that PEP is decidable. On the other hand, Yaku [194] has shown that for 2-d CA restricted to finite configurations, PEP is equivalent to the problem of whether a TM halts on the empty tape and hence is Σ_1^0-complete.

Similar results for finite CA have been studied in [169]. For 1-d CA, PEP is NLOG-complete and is NP-complete for all dimensions higher than one. In [169], examples of local rules are constructed such that CREP is PSPACE-complete/NP-complete for 1-d CA. For 1-d CA, if one restricts attention to polynomially bounded version of CREP (i.e, the number of steps is less than or equal to some polynomial in the number of cells), it is possible to construct a local rule such that CREP is P-complete (w.r.t. log space reductions). For 2-d CA, example of a rule ρ is provided such that CREP is NP-complete. A classification of CA rules similar to that of Culik and Yu (for infinite CA) is connected to several deep problems in complexity theory.

Durand [56, 57] provides complexity results for CA with a different flavour. The injectivity problem for 2-d CA restricted to finite configurations and von Neumann neighbourhoods is co-NP complete [56]. This result is about arbitrary CA and is different from the above results where examples of CA are provided for which a problem is complete for some complexity class. Hence this kind of result may be called uniform complexity results. Durand also proves [57] that the reversibility problem for 2-d CA restricted to certain types of finite configurations is complete for the class RNP introduced by Levin in [109].

2.3.5 Linear CA and VLSI application

For finite CA, the dynamical properties are completely captured by the State Transition Diagram (STD), which is a directed graph whose nodes are configurations of the CA and there is a edge from node i to node j iff configuration i leads to configuration j in one time step. The notion of STD have also been defined for infinite CA (see Subsection 2.3.3). Since a finite CA is an autonomous deterministic machine, it is easy to see that the STD will consist of components with each component having an unique cycle and trees of height ≥ 0 rooted on the cycle vertices. The cycles capture the steady state behaviour of the system and are sometimes called attractors, while a branch in a tree captures the initial transient behaviour. One can ask several important questions regarding the dynamical parameters of the system; the number of cycles, length of the cycles, height of the trees, branching degree of each node, etcetera. For an arbitrary CA such questions are very difficult to answer. For CA with periodic boundary condition some results for reversibility and maximal cycle length is presented in [77]. See [107] for recursive formulae describing STD of finite CA. However, complete characterisation is not known and generalisation to higher dimensions is difficult. For linear CA, much more information can be obtained using algebraic methods. The STD in this case shows more uniform behaviour [115]; the trees rooted on any cycle vertex is isomorphic to the tree rooted on the null configuration, the indegrees of all the nodes are equal and is equal to the dimension of the kernel of the linear map, etcetera.

For a wealth of results on the STD of 1-d periodic boundary CA see [115]. Some additional results can be found in [71]. For 2-d CA, Kawahara et al [99], investigates when the configuration reachable in one time step from the all ones configuration lies on a cycle. The dimension of the kernel of 2-d linear CA have been studied by several authors [15, 161, 166, 168] and is related to the σ-game mentioned before. For multidimensional CA, it is difficult to obtain a characterisation of the dimension of the kernel but a characterisation of reversibility will be presented in this thesis. An important problem in the algebraic analysis of linear CA is the representation of the linear global map. Martin et al in [115] use dipolynomials to represent the configuration of a periodic boundary CA. The next configuration is obtained by multiplying the present configuration with a fixed polynomial (which represents the local rule) modulo $X^N - 1$, for an N cell CA. The algebra of dipolynomials is then used in the algebraic analysis of the map. In fact dipolynomials are not necessary and polynomials can be used as has been shown in [14]. The extension of this method to multidimensional CA is possible but requires working with multivariate dipolynomials which is difficult (see [115] for details of this approach). However, the technique of dipolynomials cannot be directly used for null boundary condition. Modification of this approach where a truncation operator is applied at each stage have been reported in [141]. Another way to use dipolynomials (or polynomials) to handle null boundary condition arises from a nice technique introduced by Martin et al [115], whereby an N-cell null boundary 1-d CA can be embedded in a $(2N+2)$-cell periodic boundary 1-d CA. Kawahara et al [99], have extended this approach to study 2-d null boundary CA. However, the polynomial method fails for hybrid CA. A different approach to the problem and one that is extensively used in VLSI applications, is to represent the global rule of a CA by a matrix. For an uniform periodic boundary 1-d CA, the matrix is circulant and for nearest neighbourhood null boundary 1-d CA, the matrix is tridiagonal. The characteristic and minimal polynomial for this matrix encodes all information about the STD of the CA. For details of this approach see [168, 15]. A generalisation to multidimensional CA results in the linear operator being represented by a sum of Kronecker products of certain special matrices. We will see more of this approach in this thesis. Another approach to the study of multidimensional linear CA can be found in [106], where each cell state is considered to be a vector. All the above discussion is for CA on grids. However, linear CA on arbitrary graphs have been studied by Sutner [163, 160]. In [163], it is shown that the all-ones configuration is not a Garden-of-Eden for a linear binary CA on any finite graph. For a CA on a finite undirected graph with addition carried out in some finite abelian monoid, the predecessor existence problem is studied in [160]. It is shown that the problem is polynomial time solvable if the underlying monoid is a group and is NP-complete for an arbitrary monoid. Further, a linear time algorithm is presented to decide reversibility over a special class of graphs.

For infinite linear CA, there is a quadratic time algorithm to determine reversibility and surjectivity of the global map [167]. The algorithm is based on the representation of a configuration of a linear CA by a finite graph (a De Bruijn graph) as used by Wolfram in [190]. Earlier linear CA with the state space taken as Z_m for some positive integer m were studied by Ito, et al [92] and criteria for surjectivity and injectivity of the global transition function are presented. A more abstract treatment of linear CA, where the cell space is an Abelian group and the state space is a finite commutative ring can be found in [8]. An interesting decomposition of a CA with state space Z_m, into a set of CA with state space power of a prime which divides m, is also presented in [8]. In yet another approach to study of linear CA, the generating function for the temporal sequence of a cell is studied and is shown to be an algebraic series [112]. Additional results on linear CA can be found in the work of Jen [93].

One important area of application for finite CA is in VLSI design [151, 141, 84, 128, 54]. See [30] for details of applications of additive cellular automata to VLSI. The local communication structure of CA and the homogeneous nature of each cell are provided as strong arguments in favour of using CA for VLSI. In its use as a VLSI structure it is often offered as a replacement for the Linear Feedback Shift Register (LFSR). Perhaps the most successful area of VLSI application for CA is generation of pseudo random sequence [85, 52, 83] and their use in built-in self-test (BIST) [21, 86, 83, 54]. The successive configurations of a CA are taken as a random sequence. The possibility of random number generation by CA was first explored by Wolfram [192] who also proposed its use in cryptography [188]. However, such CA sequences are not secure as has been shown in [119]. Other areas of VLSI where CA has been used are error correcting codes [31, 32], private key cryptosystem [128], design of associative memory [32], aliasing [151], testing of FSM [28], architectures for exponentiation and inversion over finite fields [134, 16], etcetera. A VLSI architecture for CA machine based on linear CA have also been proposed [100].

In the VLSI context, the 1-d binary CA is most common though use of 2-d structure have been reported [54]. Since non-linear CA cannot be analyzed satisfactorily, these are not used in applications. Most applications are based on CA where the global map is a linear or affine map. Another important feature of CA used in VLSI applications is the null boundary condition, since periodic boundary condition require "long distance" communication between the end cells. Also the CA structure is usually a hybrid one, where each cell has its own local rule. For theoretical questions regarding hybrid 1-d CA see [12, 129, 152, 148].

2.4 Related Work

CA have been studied from several different angles other than the ones mentioned here. These approaches are important but have not been included here mainly because they are either new or have an extensive literature which require a separate survey. A (perhaps incomplete) list of these topics would include modelling in Physics [48, 49, 193], Asynchronous CA [139], Cellular Neural Networks [34, 33], CA machine [173], Quantum CA [143, 182], relation to polyomino tilings [2], ergodicity of CA [149, 60], application to cryptography [188, 51, 119, 50, 70, 128, 19] and the interesting work done at Santa Fe Institute on evolving a CA with genetic algorithms [122, 121, 53]. See [177, 3, 158, 43, 113, 104] for additional surveys/books and [189, 193] for additional bibliographies.

Acknowledgement : The survey has been greatly improved by several suggestions made by a referee, including pointers to references and detailed comments.

Chapter 3

Uniform one-dimensional CA

3.1 Introduction

In this chapter we will present basic results on uniform 1-d linear cellular automata (CA) and also present some new results on the inverse and exponent of the corresponding linear operator. We introduce the simplest variety of CA, and state its properties. In later Chapters we will consider more complicated types of CA.

3.2 Preliminaries

By a 1-d CA we will mean a finite 1-d array of cells where each cell can be in state 0 or 1. The array can be circular giving rise to what is called *periodic boundary condition* or it can be placed between two cells in the fixed state 0 leading to the *null boundary condition.* The local rule is the same for all cells and hence the CA is called *uniform CA*. Note that this is the usual definition of CA and we call it uniform CA only to distinguish from a more general class of CA considered in Chapter 4, where each cell can have its own local rule. The next state of any cell is determined by the *local rule* and depends on the previous states of the cell itself and its left and right neighbours. Thus there are a total of $2^{2^3} = 256$ possible local rules. A numbering system for local rules is presented in [189]. The idea is to consider any three variable Boolean function to be encoded by an eight bit string where each bit specifies the function value for a certain combination of the inputs. The decimal value of the eight bit string then gives the rule number. For example, rule 90 is encoded by 01011010 and is given by the following truth table. Here x_i^t is the state of the i^{th} cell in the t^{th} time instant.

$x_{i-1}^t x_i^t x_{i+1}^t$	111	110	101	100	011	010	001	000
x_i^{t+1}	0	1	0	1	1	0	1	0

Note that rule 90 is nothing but the sum modulo 2 (EXOR) of the states of the left and right neighbours. Similarly rule 150 is encoded by 10010110 and is the sum modulo 2 of the states of the cell itself and its left and right neighbours.

A *configuration of a CA* is an assignment of states 0 or 1 to each cell. At any point in time a CA is in some configuration and evolves in one time step by the application of the local rule to each of the cells. Thus at time $t = 0$, the CA is in some *initial configuration* and evolves deterministically in discrete time steps to successively new configurations. For an n-cell CA the local rule determines a *global map* which is a function from the set of all n-bit vectors into itself. If the local rule involves only the EXOR operation then the corresponding global map is a linear transformation from F_2^n into itself (here F_2 is the field of cardinality 2). In this situation *the local rule and also the CA are called linear.* There are eight possible linear local rules including rules 90 and 150.

The global dynamics of a CA is captured by a directed graph (V, A) called the *State Transition Diagram* (STD), where V is the set of all possible configurations of the CA and an arc exists from configuration i to configuration j iff in one step configuration i evolves to configuration j. It is easy to see that the STD for a CA consists of components where each component has a cycle with trees of height greater than or equal to 0 rooted on each cycle vertex [115]. In case the CA is linear, algebraic techniques can be used to study the nature of the STD [115]. If the global map is a bijection, then the CA is *reversible or invertible,* and the STD consists only of cycles with no tree configuration. In this thesis we will study the reversibility of linear CA in great detail.

The *σ-automata* are a class of binary CA (state values 0 or 1) on a graph where the local rule for any cell is the sum modulo 2 of the states of all its neighbours. If the graph is a *path* P_n, the σ-automaton is same as an n-cell null boundary CA with rule 90 for each cell. If the graph is a *cycle* C_n, then we have an n-cell CA with periodic boundary condition. A *σ^+-automaton* is similar to a σ-automaton with the only difference being that each cell is also considered to be one of its neighbours. On paths and cycles it is the same as rule 150 CA with null and periodic boundary condition respectively. Hence throughout this thesis we will use the terms σ and σ^+-automata to denote the corresponding CA.

To study the behaviour of a linear CA, one has to choose a representation for the global linear transformation. In [115], dipolynomials are used to represent both the configuration of a periodic boundary CA and the transition function. In fact, dipolynomials are not strictly necessary and the same results can be obtained using polynomials [14]. With dipolynomial or polynomial algebra, it becomes difficult to tackle null boundary CA. However attempts have been made to do so [141, 99]. Another approach is to use matrix algebraic tools, where a configuration is an element of F_2^n, and the global map is represented by a matrix with respect to the standard basis. Our approach to the analysis of linear CA will be using matrix algebra.

We will denote the field of two elements by $GF(2)$ (or F_2) and by $GF(2^l)$ we will denote the extension field of dimension l over $GF(2)$. The set $V_l = \{(i_1, \ldots, i_l) : i_j \in GF(2), 1 \leq j \leq l\}$ with the usual $+$ operator is a vector space of dimension l over $GF(2)$. Under suitably defined multiplication, V_l is isomorphic to $GF(2^l)$. Hence we will drop the distinction between the two and use the notation $GF(2^l)$ throughout. The exact meaning will be clear from the context. Throughout the thesis the base field is $GF(2)$ and we will denote the identity matrix of order n by I_n. By $\phi(n)$ we will denote the Euler totient whose value is the number of positive integers less than n and coprime to n and $sord_n(2)$, for odd n is the least positive integer j, such that $2^j \equiv \pm 1 \bmod n$.

DEFINITION 3.2.1.

1. *An S-matrix of order l, S_l, is a square tridiagonal matrix of order l, defined as,*

$$\begin{aligned}[s_{ij}] &= 1 \quad \textit{if } |i-j| = 1\\ &= 0 \quad \textit{elsewhere}\end{aligned}$$

2. *A C-matrix is a square matrix of order l, denoted by C_l, and is defined as,*

$$\begin{aligned}[c_{ij}] &= 1 \quad \textit{if } |i-j| = 1\\ &= 1 \quad \textit{if } (i = 1\, \textit{and}\, j = l) \textit{ or } (i = l\, \textit{and}\, j = 1)\\ &= 0 \quad \textit{elsewhere}\end{aligned}$$

Thus the forms of S-matrix and C-matrix are,

$$S_l = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0\\ 1 & 0 & 1 & \dots & 0 & 0\\ . & . & . & \dots & . & 0\\ . & . & . & \dots & . & 1\\ 0 & . & . & \dots & 1 & 0 \end{bmatrix}, \quad C_l = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 1\\ 1 & 0 & 1 & \dots & 0 & 0\\ . & . & . & \dots & . & 0\\ . & . & . & \dots & . & 1\\ 1 & . & . & \dots & 1 & 0 \end{bmatrix}$$

It is easy to see that a C-matrix is circulant. The C-matrix operator corresponds to the global rule for an uniform one dimensional periodic boundary condition CA with rule 90. Basic properties of this transformation have been studied in [115] using the algebra of dipolynomials. The S-matrix on the other hand, corresponds to an uniform null boundary condition CA with rule 90. Null boundary CA is of special importance in VLSI applications, since it maintains local connection.

DEFINITION 3.2.2. *(cf. [168]) The π-polynomials are a sequence of polynomials over $GF(2)$ defined as,*

$$\begin{aligned}\pi_0(x) &= 0\\ \pi_1(x) &= 1\\ \pi_i(x) &= x\pi_{i-1}(x) + \pi_{i-2}(x) \quad \textit{for } i \geq 2\end{aligned}$$

This definition of π-polynomials was introduced in [168], and in [159] they are called binary Chebyshev polynomials. Similar polynomials were studied in [15]. Alternatively $\pi_n(x)$ can also be written as (see [168])

$$\pi_n(x) \;=\; \sum_i \binom{n+i}{2i+1} x^i \bmod 2$$

The π-polynomials have several interesting properties. These have been systematically studied in [168]. Here we list several of these properties, which are relevant to our work. For proofs and more elaborate discussion, the reader is referred to [168].

Lemma 3.2.1. *(cf. [168, 15])*

1. $\pi_{p+q} \;=\; \pi_{q+1}\pi_p \;+\; \pi_q\pi_{p-1}$

2. $m \,|\, n \iff \pi_m \,|\, \pi_n$

3. $\gcd(\pi_n, \pi_m) \;=\; \pi_{\gcd(n,m)}$

4. $\pi_{2^k n} \;=\; x^{2^k-1}\pi_n^{2^k}$

5. $$\begin{aligned} \pi_{2n+1} &= \pi_{n+1}^2 + \pi_n^2 \\ &= (\pi_{n+1} + \pi_n)^2 \end{aligned}$$

Definition 3.2.3. *(cf. [168]) Let τ be an irreducible polynomial. Then the depth of τ is defined as $dp(\tau) = min\{n > 0\colon \tau \text{ divides } \pi_n\}$.*

Let,

$$\rho_n \;=\; \prod_{dp(\tau)=n} \tau^{2^{\bullet}}$$

Then ρ_n is called the critical term of π_n.

Sutner [168] proves that for any irreducible polynomial τ, dp(τ) exists. In other words, any irreducible polynomial will divide a non-trivial π polynomial.

Theorem 3.2.1. *(cf. [168]) For all positive $n = 2^k p$, where p is odd*

$$\pi_n(x) \;=\; x^{2^k-1}\prod_{d|p}\rho_d^{2^k}(x) \;=\; x^{2^k-1}\prod_{d|p}\rho_d(x^{2^k})$$

Furthermore, deg $\rho_d = \phi(d)$ unless $d = 1$.

This factorisation of $\pi_n(x)$ in terms of irreducible factors is a crucial result for the analysis of multidimensional CA.

LEMMA 3.2.2. *(cf. [168])*

a¿ All irreducible factors of the critical term ρ_n must have the same degree.

b¿ The number of irreducible factors in ρ_n is $\frac{\phi(n)}{2\, sord_n(2)}$.

From this we note that if n is a prime such that $\phi(n) = 2\, sord_n(2)$, then $\pi_n = \tau^2$ with τ irreducible.

LEMMA 3.2.3. *(cf. [168]) For any factor τ of π_{n+1}, $cork\, \tau(S_n) = deg\, \tau$, where $cork(S)$ denotes the dimension of the kernel of the linear operator S.*

The connection between S_l, C_l and the π-polynomials is given by the following.

PROPOSITION 3.2.1. *(cf. [15]) The characteristic polynomial for S_l is $\pi_{l+1}(x)$ and for C_l it is $x\pi_l(x)$.*

LEMMA 3.2.4. *(cf. [15]) S_n is invertible iff n is even and S_n^+ is invertible iff $n \not\equiv 2 \bmod 3$. Consequently $x \mid \pi_n(x)$ iff n is odd.*

3.3 Minimal Polynomial

The minimal polynomials of S_l and C_l can be obtained in terms of the π-polynomials as was shown in [168] by Sutner. The minimal polynomial for S_l was also obtained in [151] in the context of hybrid 90/150 CA. Here we provide a new simple proof for deriving the minimal polynomial for S_l. We also provide a detailed proof for deriving the minimal polynomial for C_l, since we feel that the proof is more involved than what has been presented in [168]. Needless to say, the ideas in the proof can be found at various places in [168].

THEOREM 3.3.1. *(cf. [168])*

1. *The minimal polynomial for S_l is $\pi_{l+1}(x)$.*
2. *The minimal polynomial for C_l is $x\pi_{\frac{l}{2}}(x)$ for even l and is $x\sqrt{\pi_l(x)}$ for odd l.*

Proof : 1. Since S_l is a tridiagonal matrix with sub and super diagonal entries all ones, it is easy to prove by induction that the $(i+1)^{th}$ column of the first row of S_l^i $(i \leq l-1)$ is one and the j^th columns for $j > i+1$ are all zero for all $1 \leq i < l$. Suppose $p(x)$ is the minimal polynomial for S_l and degree of $p(x)$ is $r < l$. But then it is easy to see that in $p(S_l)$ the $(r+1)^{th}$ column of the first row is one and hence $p(S_l)$ cannot be zero. Thus $p(x)$ cannot be the minimal polynomial. Therefore the degree of the minimal polynomial must be l and since $\pi_{l+1}(x)$ is of degree l and annihilates S_l, it must be the minimal polynomial for S_l.

Before we prove 2 we will require the following two Lemmata.

Lemma 3.3.1. *Let x be a configuration of σ-automata on P_l such that the first cell is in state 0 and the last cell is in state 1 and the total weight of the configuration is even. Then $S_l x$ has odd weight, i.e, the configuration $y = S_l x$ has odd number of cells in state 1.*

Proof : The proof is by induction on l. For the base case we may take $l = 3$, (since $l = 1, 2$ are trivial) and then it is easy to verify the result for $l = 3$. So suppose the result holds for all $k < l$. Assume that x has at least four cells in state 1. The case where exactly two cells of x are in state 1 is easy to settle. We write $x = y'1z$ where z has exactly two ones. Put $y = y'1$, and then length of y is less than x and hence by the induction hypothesis the image of y under rule σ has odd weight. Now several cases can arise.

1. $y = w01$ and
   (a) $z = 11$
   (b) $z = 011$
   (c) $z = 0^i u,\ i \geq 2$
   (d) $z = 0101$
   (e) $z = 010^i 1,\ i \geq 2$
2. $y = w11$ and z as above.

In each of the above cases it is easy to verify that $S_l x$ has odd weight. We verify just case 1(a). The image of y under rule σ, is of the form $v0$, and the image of $x = y11 = w0111$ under rule σ is of the form $v101$. Hence the parity of the weight of x is the same as the parity of the weight of v, which by the induction hypothesis is odd. □

One can also prove a similar result for the case where the first cell is in state 1 and the last cell is in state 0.

Lemma 3.3.2. *If x is a vector of even weight and y is a vector of odd weight then $x + y$ is a vector of odd weight.*

Proof : Let weights of x and y be p and q respectively and suppose r many 1's are cancelled in $x + y$. Then weight of $x + y$ is $(p - r) + (q - r)$ which is odd since p is even and q is odd. □

Now we return to the proof of Theorem 3.3.1.(2).

<u>Case l odd</u> : Let τ be the minimal polynomial of C_l. Then $deg\,\tau \geq \lceil \frac{l}{2} \rceil$ is clear by an argument similar to the one provided for the minimal polynomial for S_l. We will show that $x\sqrt{\pi_l}$ is an annihilating polynomial for C_l, and hence is the minimal polynomial for C_l, since degree of $x\sqrt{\pi_l}$ is $\lceil \frac{l}{2} \rceil$. This is achieved by showing that $x\sqrt{\pi_l}$ is an annihilating polynomial for all basis vectors of the standard basis. Since C_l is circulant it is sufficient to consider any particular basis vector. We choose the vector having the one in the last position and denote it by e_l (we will denote the basis vector with the 1 in the ith position by e_i). Now

$C_l e_l = e_1 + e_{l-1}$ and then the result is achieved by showing that $\sqrt{\pi_l}$ is an annihilating polynomial for $e_1 + e_{l-1}$ under rule σ on path P_{l-1}. This is so, since in any further evolution the l^{th} cell is always in state 0. In fact, we prove a much stronger result that $\sqrt{\pi_l}$ is the minimal polynomial for the subspace of symmetric patterns of σ-automaton on P_{l-1}, whose global map is given by S_{l-1}.

Let $\tau = \sqrt{\pi_l}$, then $\tau^2 = \pi_l$ and since l is odd, $x \not| \pi_l$. Also from Lemma 3.2.3.,

$$cork(\tau(S_{l-1})) = deg(\tau) = \frac{l-1}{2}.$$

Putting $F = \tau(S_{l-1})$ we get,

$$F^2 = \tau^2(S_{l-1}) = \pi_l(S_{l-1}) = 0$$

since π_l is the minimal polynomial for S_{l-1}.

Let $y \in rg(F)$, the range of F. Then there is an x such that $Fx = y$ and hence $Fy = F^2x = 0$. Therefore, y is in $Ker\, F$, and hence $rg(F) \subseteq Ker\, F$. Moreover, since $dim(rg(F)) + dim(Ker\, F) = l - 1$ and $dim(Ker\, F) = cork\, F = \frac{l-1}{2}$, it follows that $rg(F) = Ker\, F$.

Since F is symmetric, it is self adjoint and hence for any $y = Fx$ in the range of F and z in $Ker\, F$, we have

$$< y, z > = < Fx, z > = < x, Fz > = 0$$

where $<,>$ denotes the inner product. Hence $rg(F) \subseteq (Ker\, F)^{\perp}$. Putting all this together it follows,

$$Ker\, F = rg(F) \subseteq (Ker\, F)^{\perp}.$$

So, any kernel vector of F has even weight, since it is orthogonal to itself. Now the subspace $Ker\, F$ is invariant under S_{l-1}, since S_{l-1} commutes with itself and so any y in the orbit of x under S_{l-1} must have even weight.

Claim 1 : *$\tau(x)$ is the minimal polynomial for $Ker\, F = Ker\, \tau(S_{l-1})$ under rule σ on P_{l-1}.*

Proof : Let $\delta(x)$ be the minimal polynomial for $Ker\, F$. Then $\delta \,|\, \tau$ since τ is an annihilating polynomial for $Ker\, F$. Also for any $x \in Ker\, F$, $\delta(S_{l-1})x = 0$ which implies $x \in Ker\, \delta(S_{l-1})$. Therefore $Ker\, \tau(S_{l-1}) \subseteq Ker\, \delta(S_{l-1})$ which implies $cork(\tau(S_{l-1})) \leq cork(\delta(S_{l-1}))$. Since both τ and δ divide π_l, it follows from Lemma 3.2.3. that $cork\, \tau = deg\, \tau$ and $cork\, \delta = deg\, \delta$ which implies $deg\, \tau \leq deg\, \delta$. This combined with $\delta \,|\, \tau$ gives the claim. □

Claim 2 : *$Ker\, F$ is the subspace of all symmetric patterns of rule σ on P_{l-1}.*

Proof : Let $x \in Ker\, F$. Then x has even weight and any y in the orbit of x is also in $Ker\, F$ and hence also has even weight. This we claim forces x to be symmetric. Suppose not and assume that the two extreme ones of x are not equidistant from the two ends. Hence after a finite number of steps x will evolve to a configuration y where one extreme end is in state 0 and the other is in state 1 and the weight of y is even (if the weight is odd we already have a contradiction). By Lemma 3.3.1. it follows that the configuration obtained in the next step will have odd weight, which is a contradiction since all patterns in the orbit of x must have even weight. So the two extreme ones in x must be equidistant from the two ends. Let x_1 be the configuration obtained from x by removing the two extreme ones. Now if the orbit of x_1 contains an odd weight pattern then so must the orbit of x (this follows from linearity and Lemma 3.3.2.). Again repeating the argument for x_1 and by induction it follows that x must be symmetric.

This completes the proof for the case l odd.

<u>Case l even</u> : In this case, as above we show that $x\pi_{\frac{l}{2}}$ is the minimal polynomial for e_l. Again as above $C_l\, e_l = e_1 + e_{l-1} = z$ (say) and in this case we can write z as $z = y0y^r0$ where y is a configuration of length $\frac{l-2}{2}$ with the first cell 1 and all others 0 and y^r denotes the reverse of y. Note that the l^{th} and $\frac{l}{2}^{th}$ positions of z is 0 and by the form of z it is easy to see that any configuration in the orbit of z will also have these positions as 0. Thus the minimal polynomial for z is the minimal polynomial for y under rule σ with null boundary condition. Since y is of length $\frac{l-2}{2}$, $\pi_{\frac{l}{2}}$ is an annihilating polynomial for y. Hence $\pi_{\frac{l}{2}}$ is an annihilating polynomial for z and $x\,\pi_{\frac{l}{2}}$ is an annihilating polynomial for e_l and so also for C_l. But again by an argument similar to the one given for null boundary condition, no polynomial of lesser degree can annihilate C_l. □

3.4 Exponent

We introduce the notion of exponent of a matrix. This is analogous to the idea of order of an irreducible polynomial.

DEFINITION 3.4.1. *The exponent of an invertible $n \times n$ matrix A is defined to be the least positive integer ϵ such that,*

$$A^{\epsilon} = I_n .$$

Since we are considering matrices over finite fields, the existence of such an ϵ is guaranteed. The exponent is the *lcm* of the cycle lengths in the STD of the corresponding CA. Thus the cycle lengths are going to occur as divisors of the exponent. This underlines the importance of the exponent. Here we study the exponent of S_l for even l. To do so we need the following result from [115], where it is proved using the algebra of dipolynomials. Here we provide a proof using matrix algebra, which essentially mirrors the proof in [115].

LEMMA 3.4.1. *(cf. [115]) For odd l, there exists an integer $p > 0$, such that for any $x \in GF(2^l)$,*

$$C_l^{p+1}\, x = C_l\, x$$

and the least such integer p divides $2^{sord_l(2)} - 1$. Consequently, $C_l^{p+1} = C_l$.

Proof : Let $C_l = R_l + L_l$, where,

$$R_l = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ . & . & . & \dots & . & . \\ 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix} \quad \text{and} \quad L_l = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 \\ . & . & . & \dots & . & . \\ 0 & 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & 0 & \dots & 0 & 0 \end{bmatrix}$$

Now R_l is the right circular shift operator and L_l is the left circular operator. So the following immediately follows.

1. $R_l L_l = L_l R_l = I_l$.

2. $R_l^{il} = L_l^{jl} = I_l$ for all $i, j \geq 0$.

Then,

$$C_l^{2^{sord_l(2)}} = (R_l + L_l)^{2^{sord_l(2)}} = R_l^{2^{sord_l(2)}} + L_l^{2^{sord_l(2)}}.$$

The last equality follows by 1 and the fact that all operations are over $GF(2)$. Therefore,

$$C_l^{2^{sord_l(2)}} = R_l + L_l$$

since if $2^{sord_l(2)} \equiv 1 \bmod l$, then we use 2, else if $2^{sord_l(2)} \equiv -1 \bmod l$, then we use 2 alongwith the fact that $R_l^{l-1} = L_l$ and $L_l^{l-1} = R_l$. □

The least p such that the above Lemma holds is the length of the longest cycle of C_l and by the above Lemma it must divide $2^{sord_l(2)}$.

We now present some new results on the inverse and exponent of the S-matrix operator. The next result gives an $O(n^2)$ algorithm for finding the inverse of S_l and also an $O(n^2)$ algorithm for finding the predecessor of a given configuration. The naive method of obtaining the predecessor is to solve a system of n linear equations in n variables and requires $O(n^3)$ steps.

THEOREM 3.4.1. *For even n, the inverse of S_n satisfies the recurrence*

$$S_n^{-1} = \left[\begin{array}{cc|cccccccc} 0 & 1 & 0 & 1 & 0 & 1 & \dots & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & & & & & & & \\ 1 & 0 & & & & & & & \\ . & . & & & & & & & \\ . & . & & & & S_{n-2}^{-1} & & & \\ . & . & & & & & & & \\ 0 & 0 & & & & & & & \\ 1 & 0 & & & & & & & \end{array} \right]$$

$$S_2^{-1} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Proof : By induction one can show that,

$$\begin{aligned} S_n^{-1}\, S_n &= \left[\begin{array}{cc|c} 0 & 1 & 0\ 1\ \ldots\ 0\ 1 \\ 1 & 0 & 0\ 0\ \ldots\ 0\ 0 \\ 0 & 0 & \\ 1 & 0 & \\ \cdot & \cdot & \\ \cdot & \cdot & S_{n-2}^{-1} \\ \cdot & \cdot & \\ 0 & 0 & \\ 1 & 0 & \end{array}\right] \left[\begin{array}{cc|c} 0 & 1 & 0\ 0\ \ldots\ 0\ 0 \\ 1 & 0 & 1\ 0\ \ldots\ 0\ 0 \\ 0 & 1 & \\ 0 & 0 & \\ \cdot & \cdot & \\ \cdot & \cdot & S_{n-2} \\ \cdot & \cdot & \\ 0 & 0 & \\ 0 & 0 & \end{array}\right] \\ &= \begin{bmatrix} I_2 & O \\ O & I_{n-2} \end{bmatrix} \\ &= I_n \end{aligned}$$

□

Next we obtain a similar result for generalised inverse of S-matrix when n is odd.

THEOREM 3.4.2. *For odd n, the matrices obtained by the following recurrence are generalised inverses of the corresponding S-matrices.*

$$S_n^{-} = \left[\begin{array}{cc|c} 0 & 1 & 0\ 1\ 0\ 1\ \ldots\ 1\ 0 \\ 1 & 0 & 0\ 0\ 0\ 0\ \ldots\ 0\ 0 \\ 0 & 0 & \\ 1 & 0 & \\ \cdot & \cdot & \\ \cdot & \cdot & S_{n-2}^{-} \\ \cdot & \cdot & \\ 0 & 0 & \\ 1 & 0 & \\ 1 & 0 & \end{array}\right]$$

$$S_3^{-} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Proof : By induction one can verify that,

$$S_n\, S_n^{-}\, S_n = S_n$$

□

THEOREM 3.4.3. *For even n, S_n satisfies,*

$$S_n^{2^{1+sord_{n+1}(2)}-2} = I_n.$$

Thus, the exponent of S_n divides $2^{1+sord_{n+1}(2)} - 2$

Proof : The minimal polynomial for S_n is,

$$\begin{aligned}\pi_{n+1} &= \prod_{d|n} \rho_d^2 \\ &= \rho^2, \text{ say}\end{aligned}$$

The minimal polynomial for C_{n+1} is $m(x) = x\,\rho$, by Theorem 3.3.1.. Also we know from Lemma 3.4.1.,

$$C_{n+1}^{2^{sord_{n+1}(2)}} = C_{n+1} \qquad (*).$$

Let $e = 2^{sord_{n+1}(2)}$. Then $(*)$ yields,

$$\begin{aligned} & m(x) \mid x^e - x \\ \Rightarrow\ & x\,\rho \mid x(x^{e-1} - 1) \\ \Rightarrow\ & \rho \mid (x^{e-1} - 1) \\ \Rightarrow\ & \rho^2 \mid (x^{e-1} - 1)^2 = x^{2e-2} - 1 \\ \Rightarrow\ & \pi_{n+1}(x) \mid (x^{2e-2} - 1) \\ \Rightarrow\ & S_n^{2e-2} = I_n \\ \Rightarrow\ & S_n^{2^{1+sord_{n+1}(2)}-2} = I_n \ \Box \end{aligned}$$

COROLLARY 3.4.1. *For even n, S_n^+ satisfies*

$$(S_n^+)^{2^{1+sord_{n+1}(2)}} = (S_n^+)^2$$

If also $n \not\equiv 2 \bmod 3$, then,

$$(S_n^+)^{2^{1+sord_{n+1}(2)}-2} = I_n$$

In fact, we can prove a stronger result.

THEOREM 3.4.4. *For even n, the exponent ϵ of S_n equals $2e - 2$, where e is the smallest integer such that $C_{n+1}^e = C_{n+1}$. Consequently, the lcm of the cycle lengths of S_n is twice the length of the longest cycle in C_{n+1}.*

To prove the theorem we require the following lemma, which can be easily proved by induction.

LEMMA 3.4.2. *Let $\underset{\sim}{r}^{i}_{j}$, $j = 1, \ldots, n$ be the row vectors for S_n^i, where S_n^i is the i^{th} power of S_n. Then if i is odd we have,*

$$\begin{aligned}
\underset{\sim}{r}^{i}_{j} &= c_{j1}\underset{\sim}{r}^{1}_{1} + c_{j3}\underset{\sim}{r}^{1}_{3} + \ldots + c_{j,n-1}\underset{\sim}{r}^{1}_{n-1} && \textit{for odd } j \\
&= c_{j2}\underset{\sim}{r}^{1}_{2} + c_{j4}\underset{\sim}{r}^{1}_{4} + \ldots + c_{jn}\underset{\sim}{r}^{1}_{n} && \textit{for even } j
\end{aligned}$$

and if i is even,

$$\begin{aligned}
\underset{\sim}{r}^{i}_{j} &= c_{j2}\underset{\sim}{r}^{1}_{2} + c_{j4}\underset{\sim}{r}^{1}_{4} + \ldots + c_{jn}\underset{\sim}{r}^{1}_{n} && \textit{for odd } j \\
&= c_{j1}\underset{\sim}{r}^{1}_{1} + c_{j3}\underset{\sim}{r}^{1}_{3} + \ldots + c_{j,n-1}\underset{\sim}{r}^{1}_{n-1} && \textit{for even } j
\end{aligned}$$

where $c_{jk} \in \{0, 1\}$, $1 \le j \le n$ and $1 \le k \le n$

Proof : (of Theorem 3.4.4.) From the proof of the above theorem it is clear that $\epsilon \,|2e - 2$. This implies $\frac{\epsilon}{2} + 1 \le e$.

Using the above lemma, we can say that for even i, $S_n^i \neq S_n$. This is so since for even i, the first row is a linear combination of $\underset{\sim}{r}^{1}_{k}$ for even k. But for all even k, the second entry of $\underset{\sim}{r}^{1}_{k}$ is 0 and hence the second entry in the first row of S_n^i cannot be 1.

Thus it follows that the exponent ϵ of S_n must be even. For if ϵ is odd, then $S_n^\epsilon = I_n$ implies $S_n^{\epsilon+1} = S_n$ and $\epsilon + 1$ is even which is a contradiction to the above.

Now we can complete the proof.

$$\begin{aligned}
& S_n^\epsilon = I_n \\
\Rightarrow\ & \pi_{n+1}(x) \,|\, x^\epsilon - 1 \\
\Rightarrow\ & \rho^2 \,|\, x^\epsilon - 1 = (x^{\frac{\epsilon}{2}} - 1)^2, \text{ where } \rho \text{ is as in the proof of Theorem 3.4.3.} \\
\Rightarrow\ & \rho \,|\, x^{\frac{\epsilon}{2}} - 1 \\
\Rightarrow\ & x\rho \,|\, x^{\frac{\epsilon}{2}+1} - x \\
\Rightarrow\ & C_{n+1}^{\frac{\epsilon}{2}+1} = C_{n+1}, \text{ since } x\rho \text{ is the minimal polynomial of } C_{n+1} \text{ by Theorem 3.3.1.} \\
\Rightarrow\ & e \le \tfrac{\epsilon}{2} + 1
\end{aligned}$$

Then it follows that $e = \frac{\epsilon}{2} + 1$ and so $\epsilon = 2e - 2$. □

A similar result can be proved for odd n, using the fact that $C_{n+1}^{\delta+\pi} = C_{n+1}^{\delta}$, where δ is the height of a tree and π is the length of the longest cycle in the STD. However there are no simple description of δ and π for C_{n+1} with n odd. In fact these parameters can be described in terms of the orders of the irreducible factors of the minimal polynomial for C_{n+1}, and their multiplicities (see Elspas [59]).

The fact that ϵ is even can also be proved using a nice trick introduced in [115]. Let $(a_1, \ldots, a_N)$ be any configuration of an N cell null boundary CA A_1. Then the evolution from this configuration will be equivalent to the evolution from a $2N + 2$ cell periodic boundary CA A_2, which starts from the initial configuration $(0, a_1, \ldots, a_N, 0, a_N, a_{N-1}, \ldots, a_1)$ (we are assuming σ-automaton evolution which is rule 90). Let L'_N be the length of the largest cycle in an N cell null boundary CA and let L_N be the length of the largest cycle in a N cell periodic boundary CA. Then by the above embedding we have $L'_N = L_{2N+2}$. Again from [115] we have $L_{2N+2} = 2L_{N+1}$ and this implies that L'_N and hence ϵ must be even. However, the lemma that we have used is interesting in its own right.

Let $K_{n+1} = 2^{sord_{n+1}(2)} - 1$. In [115] it is noted that for almost all even n ($n+1$ is odd), $e = K_{n+1} + 1$. By the above theorem, the exponent of S_n is $2^{1+sord_{n+1}(2)} - 2 = 2K_{n+1}$, exceptions occurring exactly at values for which exceptions occur for K_{n+1}. The first exception occurs for $n+1 = 37$, where, $e = 1 + \frac{K_{n+1}}{3}$. A list of subsequent exceptions are

$n+1$	95	101	141	197	199	203
e	$1+\frac{K_{95}}{3}$	$1+\frac{K_{101}}{3}$	$1+\frac{K_{141}}{3}$	$1+\frac{K_{197}}{3}$	$1+\frac{K_{199}}{7}$	$1+\frac{K_{203}}{105}$

The technique of embedding a null boundary CA in a periodic boundary CA have further applications. It can be used to get an efficient algorithm for computing the configuration to which a null boundary CA will evolve after a finite number of time steps. In [27], such an algorithm is provided and though the description of the algorithm is different, the essential idea is similar to the above embedding. A second application of this technique is to use polynomials to study null boundary CA. For any configuration of an n-cell null boundary CA one can get a $(2n+1)$-degree polynomial which encodes the configuration. The transition function is the one chosen for periodic boundary CA. This approach has been used in [99] where it has also been extended to two dimensional CA.

Chapter 4

Hybrid One-Dimensional 90/150 CA

4.1 Introduction

A *hybrid CA* is one where each cell has its own local rule which may be different from the local rule of any other cell. The CA is linear if all the local rules are linear. Note that this notion of CA is different from the more standard notion of CA considered in Chapter 3, where all cells have the same local rule. In fact linear hybrid CA have been proposed as a basic structure in several areas of VLSI design [85, 129, 151]. The most useful structure from the VLSI point of view is a 90/150 structure where the local rule R_i for the i^{th} cell is given by,

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

where $a_i \in \{0, 1\}$ and addition is modulo 2 i.e, over F_2. If $a_i = 0$, R_i is rule 90, else R_i is rule 150. Henceforth in this chapter by CA we will mean hybrid 90/150 CA, which is a linear hybrid CA since both rules 90 and 150 are linear rules. In Chapter 7 we propose a private key cryptosystem based on hybrid 90/150 CA.

As in the case of uniform CA the one-dimensional array may be circular, giving rise to the periodic boundary condition, or it may be placed within two cells which are always in state zero, giving rise to the null boundary condition. A configuration is considered to be a vector over $GF(2)$, and the global rule of an n-cell CA is represented by the following matrix

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

where a_i is 0 or 1 according as the i^{th} cell has local rule 90 or 150 and the b in M_b is 0 or 1 according as the boundary condition is null or periodic. The next configuration y is obtained as $y = M_b x$, where x is the present configuration. The CA is *reversible* iff the corresponding matrix is non-singular. Several properties of 90/150 CA have been studied [129, 52, 155, 152, 141, 12]. Here we study the reversibility problem for 90/150 CA with both null and periodic boundary condition. Let us first state the problem for null boundary condition, since this is the easier of the two cases.

For null boundary condition the global rule is given by M_0. The matrix M_0 is uniquely specified by the string $a_1 \dots a_n$ over the alphabet $\{0, 1\}$. Then the characterisation of the reversibility of null boundary 90/150 CA reduces to the following two problems.

1. Obtain a characterisation of the set of strings $a_1 \dots a_n$ 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. For periodic boundary condition we have $b = 1$, and the situation is more complicated. However, using the results for null boundary CA, we are able to satisfactorily solve the corresponding problems for periodic boundary CA. The novel features of our proof are the use of continuants for tackling the first problem and the use of regular expression for counting. The determinant of M_0 can be elegantly expressed in terms of multivariate polynomials called continuants, which were first introduced and studied by Euler [68]. A continuant in n variables $K_n(x_1, \dots, x_n)$ is defined by the following recurrence.

$$K_0() = 1, \; K_1(x_1) = x_1$$
$$K_n(x_1, \dots, x_n) = x_1 K_{n-1}(x_2, \dots, x_n) + K_{n-2}(x_3, \dots, x_n) \qquad (II)$$

In fact the continuants satisfy a more general recurrence [68, pp 289])

$$K_{m+n}(x_1, \dots, x_m, x_{m+1}, \dots, x_{m+n})$$
$$= K_m(x_1, \dots, x_m) K_n(x_{m+1}, \dots, x_{m+n}) + K_{m-1}(x_1, \dots, x_{m-1}) K_{n-1}(x_{m+2}, \dots, x_{m+n}) (III)$$

and using the relation in [68, pp 304], we have,

$$K_n(a_1, \dots, a_n) = \det M_0$$

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

$$K_n(a_1, \dots, a_n) = K_n(a_n, \dots, a_1) \qquad (IV)$$

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.

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 (or $GF(2)$) 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 $|S|$.

4.2 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 say 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 4.2.1. *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). □

The above lemma shows that a string y can be repeatedly "reduced" from the left to obtain shorter strings which are reversible iff the original string is reversible. This suggests a linear time algorithm to test reversibility of a given 90/150 CA. To formalise this, for any two strings u, v we write $u \hookrightarrow v$ and say u reduces to v if one of the following conditions hold.

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

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

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

In particular, $0i \hookrightarrow \epsilon$, $10 \hookrightarrow 1$ and $11 \hookrightarrow 0$

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

REMARK 4.2.1. *Similar reduction from the right is also possible.*

PROPOSITION 4.2.1. *Let $y \in \{0,1\}^*$. Then y reduces in zero or more steps to exactly one of the strings in $\{\epsilon, 0, 1\}$. Moreover, 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 4.2.1., any reduction preserves reversibility. □

From this we get the following linear time algorithm for determining reversible null boundary 90/150 CA. (See [160] 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.

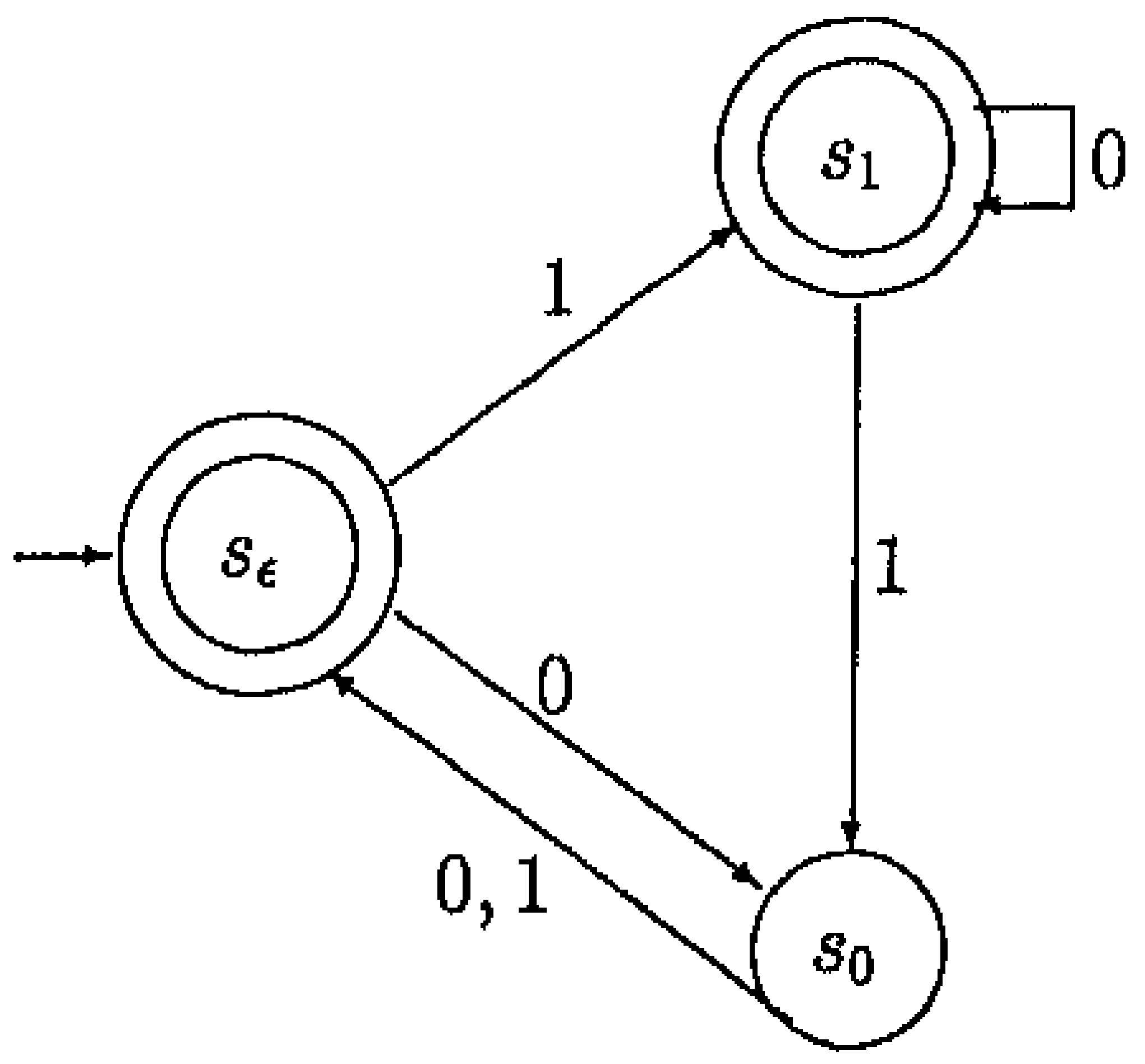

Figure 4.1: DFA to recognise reversible null boundary 90/150 CA.

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 (see Figure 4.1). So we get the following

THEOREM 4.2.1. *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 90/150 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,

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

We can solve this set of equations using Arden's Lemma [82, pp 54], which states that for regular expressions $\boldsymbol{P, Q, R}$ 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 4.2.2. *The regular expression for the set of all reversible strings which correspond to null boundary CA is given by $\boldsymbol{\alpha} + \boldsymbol{\alpha 10^*}$, where $\boldsymbol{\alpha} = \boldsymbol{((0+10^*1)(1+0))^*}$.*

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 4.2.1. , $L_\epsilon \cap L_1 = \phi$. Let $S^{(n)}, L_\epsilon^{(n)}, L_1^{(n)}$ denote the sets of 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 4.2.2. *For* $n \geq 0$, $|S^{(n)}| = \sum_{i=0}^{n} |L_\epsilon^{(i)}|$

Proof : The regular expression for L_1 is $\alpha 10^*$ where α is the regular expression for L_ϵ. Let $x \in L_1^{(n)}$ be such that $x = y10^i$ where $0 \leq i \leq n-1$ times. Then $y \in L_\epsilon^{(n-1-i)}$. Conversely for any $y \in L_\epsilon^{(n-1-i)}$ we get an unique $x \in L_1^{(n)}$. This establishes a 1-1 correspondence between the set of all such x's and $L_\epsilon^{(n-1-i)}$. 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 of computing $|S^{(n)}|$ reduces to computing $|L_\epsilon^{(i)}|$, for each i. It turns out that $|L_\epsilon^{(n)}|$ satisfies a nice recurrence relation.

LEMMA 4.2.2. $|L_\epsilon^{(0)}| = 1$, $|L_\epsilon^{(1)}| = 0$

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

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 $|y| = n-2$.
Case 1 : $ab = 00$ or $ab = 01$. Then we have $x \hookrightarrow y$. 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 $L_\epsilon^{(n)}$.
Case 2 : $ab = 10$ 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,

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

COROLLARY 4.2.1. *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 4.2.2.. □

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

LEMMA 4.2.3. *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

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

□

We finally obtain,

THEOREM 4.2.3. *For $n \geq 0$,*

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

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

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

Proof : For $n = 0, 1$ it is easy to check that $|S^{(n)}| = 1$. From Corollary 4.2.1., 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,
$|S^{(n)}| = \frac{3}{2}[\frac{2}{3}(2^{n-1} + (-1)^n)] + \frac{2}{3}[2^{n-2} + (-1)^{n-1}]$
$= \frac{1}{3}[2^{n+1} + (-1)^n]$ □

REMARK 4.2.2. *Approximately two thirds of all strings of length n are reversible.*

4.3 Periodic Boundary CA

We now characterise reversibility of 90/150 CA with periodic boundary condition. The transition matrix for such a CA is of the following form.

$$M_1 = \begin{bmatrix} a_1 & 1 & 0 & 0 & \ldots & 0 & 1 \\ 1 & a_2 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & a_3 & 1 & \ldots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \ldots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \ldots & & 1 \\ 1 & 0 & \cdot & \cdot & \ldots & 1 & a_n \end{bmatrix}$$

In analogy with continuants, let us denote the determinant of M_1 by $P_n(a_1, \ldots, a_n)$. We will only consider $n \geq 2$. Then we have the following

PROPOSITION 4.3.1. $P_n(a_1, \ldots, a_n) = K_n(a_1, \ldots, a_n) + K_{n-2}(a_2, \ldots, a_{n-1})$. *Consequently, $a_1 \ldots a_n$ is reversible under periodic boundary condition iff exactly one of $a_1 \ldots a_n$ and $a_2 \ldots a_{n-1}$ is reversible under null boundary condition.*

Proof : Expanding the determinant by the first row, we get (note that all operations are over $GF(2)$),

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

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 4.3.1. *1. The continuant $K_n(a_1, \ldots, a_n)$ can be obtained by the following simple rule [68]. Start with the term $a_1 a_2 \ldots a_n$ and then cancel out pairs $a_k a_{k+1}$ 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 $a_k a_{k+1}$, consider $a_n a_1$ 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 4.2.1.. 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 4.3.1. *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,

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

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. □

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 4.3.1., $T^{(n)}$ can be written as,

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

where,

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

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 4.3.2. *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 $y \hookrightarrow v$ and $x \hookrightarrow v$.*

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, $z \hookrightarrow 0$ and $az \hookrightarrow 0$.

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 $z \hookrightarrow 0$ and $az \hookrightarrow 0$. 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 $z \hookrightarrow 0$, either

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

with $|y| < |z|$.

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

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

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

PROPOSITION 4.3.3. *For $n \geq 2$, let*

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

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

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 $y = azb \hookrightarrow \epsilon$. Now $az0 \hookrightarrow \epsilon$ iff $az1 \hookrightarrow \epsilon$. So the strings in $L_\epsilon^{(n)}$ can be paired as $az0$ and $az1$. Then it is easy to check that exactly one of the strings $z0$ and $z1$ reduces to 1 and the other reduces to 0. (By Proposition 4.3.2. 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 [68], we will let $[\phi]$ denote the value of a Boolean predicate ϕ.

LEMMA 4.3.1. *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 $x = azb \hookrightarrow \epsilon$ and $z \hookrightarrow 0$. But $azb \hookrightarrow \epsilon$ implies $az \hookrightarrow 0$, hence $x \in A^{(n)}$ iff there exists string z such that $az \hookrightarrow 0$ and $z \hookrightarrow 0$. But by Proposition 4.3.2., such strings do not exist.
2. In this case $x \in B^{(n)}$ iff $x = azb \hookrightarrow 1$ and $z \hookrightarrow 0$.

If $b = 1$, $az \hookrightarrow \epsilon$ and $z \hookrightarrow 0$. There are $\frac{1}{2}|L_\epsilon^{(n-1)}|$ such strings (by Proposition 4.3.3.).

If $b = 0$, then two cases arise

a) $z = 0^{n-2}$, $a = 1$ where $10^{n-1} \hookrightarrow 1$ and $0^{n-2} \hookrightarrow 0$. But then $n-2$ and hence n must be odd. This contributes the term $[2 \nmid n]$ to $|B^{(n)}|$.

b) $z = y10^i$ where $0 \leq i \leq n-3$ and both $ay10^i \hookrightarrow 1$ and $y10^i \hookrightarrow 0$. Therefore $ay \hookrightarrow \epsilon$ and $y \hookrightarrow c$ for some $c \in \{0,1\}$. By Proposition 4.3.3. there are $\frac{1}{2}|L_\epsilon^{(n-2-i)}|$ such strings.

So, $|B^{(n)}| = [2 \nmid n] + \frac{1}{2}\sum_{i=1}^{n-1} |L_\epsilon^{(i)}|$.

3 and **4** are similar to above. □.

So finally we get the following

THEOREM 4.3.2. *For all $n \geq 2$,*

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

Proof : Using the above lemma and (V),

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

REMARK 4.3.2. *$|T^{(n)}|$ is approximately half of $|S^{(n)}|$ and one third of the total number of binary strings of length n.*

4.4 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 Feedback Shift Registers (LFSR), which have been quite extensively studied [66]. 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 from [151].

LEMMA 4.4.1. *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.*

However the proof given in [151] is involved. In fact it is easy to see that the proof of Theorem 3.3.1.(1) applies in this case also.

An n-cell CA is of *maximal length* if the characteristic polynomial of its transition matrix is primitive, which is true iff in the STD for the CA there are two cycles, with the null configuration on a cycle of length one and all other configurations on a cycle of length $2^n - 1$. The following result about maximal length CA is presented in [151], where it is proved using matrix algebra. Here we present a short combinatorial proof of the same.

LEMMA 4.4.2. *If a linear null boundary n-cell nearest neighbourhood CA is of maximal length, then it must use only rules 90 and 150 as the local rules.*

Proof : Suppose not and let r be the number of the first cell from the left end where the local rule R_r is neither 90 nor 150. If R_r does not depend on any of its neighbours then we can divide the CA into three parts of $r-1$ cells, 1 cell and $n-r$ cells each, with no interaction between the parts. The maximum cycle length possible in such a structure is

$$(2^{r-1}-1)(2^1-1)(2^{n-r}-1)$$

which is less than 2^n-1. (See Section7.3.1 for the length of cycles of such "product" CA).

So R_r must depend on exactly one of its neighbours. Let R_r depend on the left neighbour (the other case is similar). Then the CA can be divided into two parts of r cells and $n-r$ cells each, where the first part does not depend on the second part, but the second part depends on the first part. Thus overall the maximum cycle length possible is

$$(2^r-1)(2^{n-r}) < 2^n-1.$$

Hence the result holds. □

The *temporal sequence* of a cell of a CA is the sequence of successive states that the cell passes through in the evolution of the CA. Now we prove the following

PROPOSITION 4.4.1. *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,

$$a_i^t = c_{n-1}a_i^{t-1} + \ldots + c_0a_i^{t-n} \quad for\ t \geq n$$

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 $q(M)\boldsymbol{x} = 0$.

Then $q(x) \mid p(x)$ 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. □

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 synthesize a minimum length LFSR whose output is the given sequence. This is done by the famous Berlekamp-Massey Shift Register Synthesis Algorithm [18, 118]. 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 this 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 4.4.2. *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 4.4.3. *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 CA to have the same characteristic polynomial. If $a_1 \ldots a_n$ encodes a CA, then $K_n(x + a_1, \ldots, x + a_n)$ is its characteristic polynomial and since $K_n(x + a_1, \ldots, x + a_n) = K_n(x + a_n, \ldots, x + a_1)$ (from (IV)), the CA encoded by $a_n \ldots a_1$ also has the same characteristic polynomial. Of course if $a_1 \ldots a_n$ is a palindrome, i.e, $a_i = a_{n-i}$ for all i, then this is trivially true. Otherwise we have two distinct CA with the same characteristic polynomial. It is interesting to note that if the diagonal is a palindrome, then the characteristic polynomial is factorizable.

PROPOSITION 4.4.4. *If the diagonal of the transition matrix for an n-cell null boundary CA is a palindrome, then its characteristic polynomial can be factored.*

Proof : Let $a_1, \ldots, a_n$ be the diagonal. Then the characteristic polynomial $p(x)$ is given by,

$$p(x) \;=\; K_n(a_1 + x, \ldots, a_n + x).$$

Now two cases arise.

<u>n is even and equal to $2r$</u>

$$\begin{aligned} p(x) &= K_{2r}(a_1 + x, \ldots, a_r + x, a_{r+1} + x, \ldots, a_{2r} + x) \\ &= K_r(a_1 + x, \ldots, a_r + x)\, K_r(a_{r+1} + x, \ldots, a_{2r} + x) \\ &\quad + K_{r-1}(a_1 + x, \ldots, a_{r-1} + x)\, K_{r-1}(a_{r+2} + x, \ldots, a_{2r} + x) \\ &= (K_r(a_1 + x, \ldots, a_r + x) \;+\; K_{r-1}(a_1 + x, \ldots, a_{r-1} + x))^2 \end{aligned}$$

In fact in this case $p(x)$ is a perfect square.

<u>n is odd and equal to $2r+1$</u>

$$\begin{aligned} p(x) &= K_{2r+1}(a_1 + x, \ldots, a_r + x, a_{r+1} + x, a_{r+2} + x \ldots, a_{2r+1} + x) \\ &= K_{r+1}(a_1 + x, \ldots, a_{r+1} + x)\, K_r(a_{r+2} + x, \ldots, a_{2r+1} + x) \\ &\quad + K_r(a_1 + x, \ldots, a_r + x)\, K_{r-1}(a_{r+3} + x, \ldots, a_{2r+1} + x) \\ &= K_r(a_1 + x, \ldots, a_r + x)(K_{r+1}(a_1 + x, \ldots, a_{r+1} + x) \\ &\quad + K_{r-1}(a_1 + x, \ldots, a_{r-1} + x)) \end{aligned}$$

□

Let $E^{(n)}$ be the set of all reversible palindromes of length n. Define,

$A_n \;=\; 2^{n-1} \;-\; |E^{(n)}|$

$B_n \;=\; \frac{1}{2}(|S^{(n)}| \;-\; |E^{(n)}|)$

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 $|E^{(n)}|$.

Lemma 4.4.3. *For $n \geq 2$, $|E^{(n)}| \;=\; 2^{\lfloor \frac{n}{2} \rfloor - 1} + |L_0^{(\lfloor \frac{n}{2} \rfloor - 1)}|$, 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,

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

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

b) $a_1 \ldots a_{k-1} \hookrightarrow 0$. 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) $a_1 \ldots a_{k-1} \hookrightarrow 1$. 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
$|L_\epsilon^{(k-1)}| + 2\,|L_0^{(k-1)}| + |L_1^{(k-1)}|$
$= 2^{k-1} + |L_0^{(k-1)}|$. □

Now we can prove

THEOREM 4.4.1. *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 $|E^{(n)}|$ 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 $|E^{(n)}|$, it is easy to find the expressions for A_n and B_n and check that indeed $A_n > B_n$. □

Chapter 5

Two-Dimensional CA on Square Grids

5.1 Introduction

A two-dimensional CA is defined to be a CA on a two-dimensional grid of cells. The grid may be considered to be folded in one or both directions. If it is folded in exactly one direction it is called a cylinder and if it is folded in both directions the structure is called a torus. The neighbours of a cell are the two vertically orthogonal and the two horizontally orthogonal cells. The local rule changes the state of a cell to the sum (modulo 2) of the states of the four orthogonal neighbours. We will consider uniform CA, where each cell evolves under the same local rule. As in the case of one-dimensional CA, one may consider a cell to be a neighbour of itself. In terms of σ-automata, a two-dimensional CA would be a σ-automaton on a product graph $A \times B$, where A and B are suitably chosen to be paths or cycles giving rise to grids, cylinders and tori. The evolution is as before defined by each vertex changing its state to the sum (modulo 2) of the states of its adjacent neighbours. A σ^+-automaton is defined as before by considering a vertex to be a neighbour of itself. The notions of configuration, global map, State Transition Diagram (STD) are simple generalizations of the corresponding concepts for one-dimensional CA (see Chapter 3). Note that the global map of the CA is a linear transformation from the set of all configurations into itself. We will defer to Chapter 6, the task of obtaining a representation of the global linear map. Here we will analyse the reversibility of the global map in terms of the roots of the π-polynomials introduced in Chapter 3.

A $\sigma(\sigma^+)$-automata is reversible iff the corresponding linear transformation is invertible. Reversibility is an important phenomena for this class of automata (see also [161]). It means that the State Transition Diagram consists entirely of cycles, and as a result it is possible to start from one configuration and come back to it after a finite number of steps. The σ-automata on an $m \times n$ grid is reversible iff $\pi_{m+1}(x)$ and $\pi_{n+1}(x)$ are relatively prime iff $m+1$ and $n+1$ are relatively prime. This result has been derived using different methods [15, 166, 161, 168]. The coprimeness of $m+1$ and $n+1$ present a nice characterization of reversibility. Unfortunately, for the σ^+-automata obtaining such a simple characterization seems to be difficult, though it is known [15, 168] that σ^+-automata on an $m \times n$ grid is reversible iff $\pi^+_{m+1}(x)$ and $\pi_{n+1}(x)$ are coprime. The problem has, however, been solved for certain special cases [15, 168].

In this chapter, we study σ^+-automata on square $m \times m$ grids. Our work is motivated by two open problems posed by Sutner in [168]. Before we state them we need to introduce the concept of total irreversibility. In what follows we will denote by $p^+(x)$ the polynomial obtained from $p(x)$ by the map $x \to 1+x$ over $GF(2)$.

The concept of total irreversibility was introduced in [168] for σ^+-automata on product graphs $G = H \times P_n$, where H is an arbitrary graph and P_n is the path graph on n vertices. We however describe the concept only for graphs of the form $P_m \times P_n$, i.e, $m \times n$ grids. The corank (dimension of kernel) of the σ^+-automata on $m \times n$ grid is given by $cork(\pi^+_{m+1}(S_n)) = cork(\pi^+_{n+1}(S_m))$ (see [15, 168]). If the corank is zero then the automaton is reversible and if the corank is positive then it is irreversible. Thus the maximum value of the corank in some sense captures the notion of maximum irreversibility and leads to the following definition of total irreversibility. The σ^+-automata on $m \times n$ grid is totally irreversible if it has the maximum corank, i.e if $cork(\pi^+_{n+1}(S_m)) = n$ iff $\pi^+_{n+1}(S_m) = 0$. But $\pi_{m+1}(x)$ is the minimal polynomial for S_m (by Theorem 3.3.1.) and hence divides $\pi^+_{n+1}(x)$. The least value of n for which this occurs is defined to be the weak period of P_m, the path graph on m vertices. For some interesting results on weak periods see [168]. For the case of square grids, $m = n$ and $\pi_{m+1}(x) \mid \pi^+_{n+1}(x)$ implies $\pi_{m+1}(x) = \pi^+_{m+1}(x)$. So a square grid is totally irreversible under σ^+-automata iff $\pi_{m+1}(x) = \pi^+_{m+1}(x)$. Now we can state two open problems posed by Sutner in [168]. (Refer to Chapter 3 for meaning of the terms ρ_e and $dp(\tau)$. Also certain basic properties of π-polynomials from Chapter 3 will be used.)

1. "For the $m \times m$ grid to be reversible under rule σ^+ we must have $6 \not| m+1$ and for all odd $e > 3$ such that $e \mid m+1$ and $\tau \mid \rho_e$ irreducible : $dp(\tau^+) \not| m+1$. Is there a simple algorithm to test the second property ?"

2. "Are there any totally irreversible squares other than 4×4 ? Equivalently, is there any $m > 4$ such that $\pi_{m+1}(x) = \pi^+_{m+1}(x)$?"

It is conjectured in [168] that the answer to the second question is no and here we prove that indeed it is no. So this also proves that the corank of the σ^+-automata on square $m \times m$ grid is strictly less than m for $m \neq 4$.

As for the first question we derive an alternative equivalent condition for reversibility and use it to obtain several sufficient conditions for both reversibility and irreversibility. The analysis leads us to obtain a complete characterization of irreducible polynomials $\tau(x)$ over $GF(2)$ with $\tau(x) = \tau^+(x)$. It turns out that characterizing the depths of such polynomials is essential for obtaining a simple characterization of reversibility. Our results indicate that this in general is difficult.

5.2 Total Irreversibility

In this section we prove that totally irreversible grids do not exist for $m \neq 4$. We essentially prove that $\pi_{m+1}(x) \neq \pi^+_{m+1}(x)$ for $m \neq 4$. Then the result follows from what has been discussed in the introduction. We start by proving some preliminary results.

The following can easily be proved by induction.

LEMMA 5.2.1. *If m is even, then*

1. *$\pi_{m+1}(x)$ contains only even powers of x, i.e, for odd r the coefficient of x^r in $\pi_{m+1}(x)$ is 0.*

2. *The coefficient of x^{m-2} in $\pi_{m+1}(x)$ is 1.*

Next we require the following crucial result. In what follows, the roots of the polynomials concerned are taken in a suitable extension field.

LEMMA 5.2.2. *If $\pi^+_{m+1}(x) = \pi_{m+1}(x)$, then $m \equiv 4 \bmod 16$.*

Proof : We prove this in four steps.

Step 1: If $\pi^+_{m+1}(x) = \pi_{m+1}(x)$, then m must be even.

If $\alpha_1, \dots, \alpha_m$ are the roots of $\pi_{m+1}(x)$ then $1+\alpha_1, \dots, 1+\alpha_m$ are the roots of $\pi^+_{m+1}(x)$. The coefficient of x^{m-1} in $\pi_{m+1}(x)$ is $\sum_{i=1}^m \alpha_i$ and the coefficient of x^{m-1} in $\pi^+_{m+1}(x)$ is $\sum_{i=1}^m (1+\alpha_i)$.

So if $\pi^+_{m+1}(x) = \pi_{m+1}(x)$, then

$$\sum_{i=1}^m \alpha_i = \sum_{i=1}^m (1+\alpha_i)$$

which gives that $m \bmod 2 = 0$. Hence m must be even.

Step 2: If $\pi^+_{m+1}(x) = \pi_{m+1}(x)$ then $m \equiv 0 \bmod 4$.

By Step 1, we have that m is even ($= 2r$ say). Hence $\pi_{m+1}(x)$ contain only even powers of x (by Lemma 5.2.1.(1)). Let $\alpha_1, \dots, \alpha_m$ be the roots of $\pi_{m+1}(x)$. Then $\sum_{i=1}^m \alpha_i = 0$.

So if $\pi^+_{m+1}(x) = \pi_{m+1}(x)$, equating the coefficient of x^{m-2}, we get,

$$\begin{aligned}
\sum_{i=1}^m \sum_{j=i+1}^m \alpha_i\alpha_j &= \sum_{i=1}^m \sum_{j=i+1}^m (1+\alpha_i)(1+\alpha_j) \\
&= \binom{m}{2} \bmod 2 + (m-1)\sum_{i=1}^m \alpha_i \\
&\quad + \sum_{i=1}^m \sum_{j=i+1}^m \alpha_i\alpha_j \\
&= \binom{m}{2} \bmod 2 + \sum_{i=1}^m \sum_{j=i+1}^m \alpha_i\alpha_j
\end{aligned}$$

$$\Rightarrow \binom{m}{2} \bmod 2 = 0$$

$$\Rightarrow \frac{2r(2r-1)}{2} \bmod 2 = 0$$

$\Rightarrow r$ must be even

$\Rightarrow m \equiv 0 \bmod 4.$

Step 3: If $\pi_{m+1}(x) = \pi^{+}_{m+1}(x)$ *, then* $m \equiv 4 \bmod 8$.

By Step 2, we have $m = 4k$. Since by assumption $\pi^{+}_{m+1}(x) = \pi_{m+1}(x)$ equating coefficient of x^{m-4} on both sides we get,

$$\sum_{i_1,i_2,i_3,i_4} \alpha_{i_1}\alpha_{i_2}\alpha_{i_3}\alpha_{i_4} = \sum_{i_1,i_2,i_3,i_4} (1+\alpha_{i1})(1+\alpha_{i_2})(1+\alpha_{i_3})(1+\alpha_{i_4})$$

Again using the fact that m is even, we know that $\pi_{m+1}(x)$ contain only the even powers of x, hence

$$\sum_i \alpha_i = \sum_{i_1,i_2,i_3} \alpha_{i_1}\alpha_{i_2}\alpha_{i_3} = 0$$

Therefore,

$$\sum_{i_1,i_2,i_3,i_4} \alpha_{i_1}\alpha_{i_2}\alpha_{i_3}\alpha_{i_4} = \begin{pmatrix} m \\ 4 \end{pmatrix} \bmod 2 + \begin{pmatrix} m-2 \\ 2 \end{pmatrix} \bmod 2 \sum_{i_1,i_2} \alpha_{i_1}\alpha_{i_2}$$
$$+ \sum_{i_1,i_2,i_3,i_4} \alpha_{i_1}\alpha_{i_2}\alpha_{i_3}\alpha_{i_4}$$

Now,

$\begin{pmatrix} m-2 \\ 2 \end{pmatrix} \bmod 2 = \begin{pmatrix} 4k-2 \\ 2 \end{pmatrix} \bmod 2$

$= \frac{(4k-2)(4k-3)}{2} \bmod 2$

$= 1$

and by Lemma 5.2.1.(2), $\sum_{i_1,i_2} \alpha_{i_1}\alpha_{i_2} =$ coefficient of $x^{m-2} = 1$.

So we get $\begin{pmatrix} m \\ 4 \end{pmatrix} \bmod 2 = 1$

$\Rightarrow \frac{4k(4k-1)(4k-2)(4k-3)}{1\times2\times3\times4} \bmod 2 = 1$

$\Rightarrow \frac{k(4k-1)(2k-1)(4k-3)}{3} \bmod 2 = 1$

$\Rightarrow k$ must be odd.

$\Rightarrow m \equiv 4 \bmod 8.$

Lastly we prove,

Step 4: If $\pi^{+}_{m+1}(x) = \pi_{m+1}(x)$ *, then* $m \equiv 4 \bmod 16$

By Step 3, we have that $m = 8k+4$.

So assuming $\pi^{+}_{m+1}(x) = \pi_{m+1}(x)$, we equate the coefficient of x^{m-8} on both sides to get,

$\sum_{i_1,\ldots,i_8} \alpha_{i_1}\ldots\alpha_{i_8} = \sum_{i_1,\ldots,i_8} (1+\alpha_{i_1})\ldots(1+\alpha_{i_8})$

$= \begin{pmatrix} m \\ 8 \end{pmatrix} \bmod 2 + \begin{pmatrix} m-2 \\ 6 \end{pmatrix} \bmod 2 \sum_{i_1,i_2} \alpha_{i_1}\alpha_{i_2}$

$+ \begin{pmatrix} m-4 \\ 4 \end{pmatrix} \bmod 2 \sum_{i_1,\ldots,i_4} \alpha_{i_1}\ldots\alpha_{i_4}$

$+\begin{pmatrix} m-6 \\ 2 \end{pmatrix}$ mod 2 $\sum_{i_1,\ldots,i_6} \alpha_{i_1}\ldots\alpha_{i_6}$
$+\sum_{i_1,\ldots,i_8} \alpha_{i_1}\ldots\alpha_{i_8}$
Now for $m = 8k + 4$, we have,

$$\begin{pmatrix} m \\ 8 \end{pmatrix} = \begin{pmatrix} 8k+4 \\ 8 \end{pmatrix} \equiv k \bmod 2$$

$$\begin{pmatrix} m-2 \\ 6 \end{pmatrix} = \begin{pmatrix} 8k+2 \\ 6 \end{pmatrix} \equiv 0 \bmod 2$$

$$\begin{pmatrix} m-4 \\ 4 \end{pmatrix} = \begin{pmatrix} 8k \\ 4 \end{pmatrix} \equiv 0 \bmod 2$$

$$\begin{pmatrix} m-6 \\ 2 \end{pmatrix} = \begin{pmatrix} 8k-2 \\ 2 \end{pmatrix} \equiv 1 \bmod 2$$

So we get,

$$k \bmod 2 + \sum_{i_1,\ldots,i_6} \alpha_{i_1}\ldots\alpha_{i_6} = 0$$

But $c = \sum_{i_1,\ldots,i_6} \alpha_{i_1}\ldots\alpha_{i_6}$ is the coefficient of x^{m-6} in $\pi_{m+1}(x)$ and is determined as follows (see Chapter 3, Equation 3.2),

$$\begin{aligned} c &= \begin{pmatrix} m+1+m-6 \\ 2(m-6)+1 \end{pmatrix} = \begin{pmatrix} 2m-5 \\ 2m-11 \end{pmatrix} = \begin{pmatrix} 2m-5 \\ 6 \end{pmatrix} \\ &= \begin{pmatrix} 16k+8-5 \\ 6 \end{pmatrix} = \begin{pmatrix} 16k+3 \\ 6 \end{pmatrix} \equiv 0 \bmod 2 \end{aligned}$$

Hence it follows that k must be even and so $m \equiv 4 \bmod 16$. □

Let $c(m, i)$ be the coefficient of x^i in $\pi_m(x)$. Then using Chapter 3, Equation 3.2 we can prove the following.

Lemma 5.2.3. *For $m \equiv 4 \bmod 16$,*

1. $c(m+1, m) \equiv 1 \bmod 2$
2. $c(m+1, m-2) \equiv 1 \bmod 2$
3. $c(m+1, m-4) \equiv 1 \bmod 2$
4. $c(m+1, m-6) \equiv 0 \bmod 2$
5. $c(m+1, m-8) \equiv 0 \bmod 2$
6. $c(m+1, m-10) \equiv 1 \bmod 2$

7. $c(m+1, m-12) \equiv 1 \bmod 2$

Proof : The proofs of $1-7$ are similar. We will only prove 7.

Since $m \equiv 4 \bmod 16$ we can write $m = 16k+4$. So from Chapter 3, Equation 3.2 we get,

$$\begin{aligned}
c(m+1, m-12) &= \binom{m+1+m-12}{2(m-12)+1} \bmod 2 \\
&= \binom{2m-11}{2m-23} \bmod 2 \\
&= \binom{2m-11}{12} \bmod 2 \\
&= \binom{32k-3}{12} \bmod 2 \\
&\equiv \frac{32k-4}{4}\frac{32k-6}{6}\frac{32k-8}{8}\frac{32k-10}{10}\frac{32k-12}{12}\frac{32k-14}{2} Y \bmod 2 \\
&\equiv 1 \bmod 2
\end{aligned}$$

where $Y = \frac{32k-3}{3}\frac{32k-5}{5}\frac{32k-7}{7}\frac{32k-9}{9}\frac{32k-11}{11}\frac{32k-13}{13}$

□

Lemma 5.2.4. *For $m \equiv 4 \bmod 16$ the coefficient of x^{m-12} in $\pi^+_{m+1}(x)$ is 0.*

Proof : By Chapter 3, Equation 3.2

$$\begin{aligned}
\pi_{m+1}(x) &= \sum_i \binom{m+1+i}{2i+1} x^i \bmod 2 \\
\Rightarrow \pi^+_{m+1}(x) &= \sum_i \binom{m+1+i}{2i+1} (1+x)^i \bmod 2
\end{aligned}$$

If C be the coefficient of x^{m-12} in $\pi^+_{m+1}(x)$ then

$$C = \sum_{i=0}^{12} \binom{m+1+m-12+i}{2(m-12+i)+1} \binom{m-12+i}{m-12}$$

Using the above Lemma and the fact that $\pi_{m+1}(x)$ contain only even powers of x we can conclude that the first term is non zero only for $i = 0, 2, 4, 10, 12$. Hence,

$$C \equiv \binom{m-12}{m-12} + \binom{m-10}{m-12} + \binom{m-4}{m-12} + \binom{m-2}{m-12} + \binom{m}{m-12}$$

$$\Rightarrow C \equiv 1 + \binom{m-10}{2} + \binom{m-4}{8} + \binom{m-2}{10} + \binom{m}{12}$$

Since $m \equiv 4 \bmod 16$ we can write $m = 16k+4$ and so,

$$\binom{m-10}{2} = \binom{16k-6}{2} \equiv 1 \bmod 2$$

$$\binom{m-4}{8} \equiv \binom{m-2}{10} \equiv \binom{m}{12} \equiv 0 \bmod 2$$

Hence, $C = 1 + 1 \equiv 0 \bmod 2$ □

THEOREM 5.2.1. *$\pi_{m+1}(x) = \pi^+_{m+1}(x)$ iff $m = 4$*

Proof : For $m = 1, 2, 3$ it is easy to verify that $\pi_{m+1}(x) \neq \pi^+_{m+1}(x)$ and for $m = 4$ it is also easy to verify that $\pi_{m+1}(x) = \pi^+_{m+1}(x)$.

If $m > 4$ then assume that $\pi^+_{m+1}(x) = \pi_{m+1}(x)$. Then by Lemma 5.2.2. it follows that $m \equiv 4 \bmod 16$. But by Lemma 5.2.4. it then follows that the coefficient of x^{m-12} in $\pi^+_{m+1}(x)$ is 0 and by Lemma 5.2.3. the coefficient of x^{m-12} in $\pi_{m+1}(x)$ is 1. So this means that $\pi^+_{m+1}(x) \neq \pi_{m+1}(x)$ which is a contradiction to our assumption. Hence the result follows. □

So this proves that totally irreversible grids do not exist for $m \neq 4$.

5.3 Reversibility

In this section we address the problem of characterizing reversible σ^+-automata on square $m \times m$ grids. A necessary and sufficient condition for reversibility is that $\pi_{m+1}(x)$ and $\pi^+_{m+1}(x)$ are relatively prime [15, 168]. For the case of σ-automata on $m \times n$ grid, the analogous condition for reversibility is that $\pi_{m+1}(x)$ and $\pi_{n+1}(x)$ are relatively prime [15, 168]. Thus on a square $m \times m$ grid, σ-automata is always irreversible. For the σ^+-automata on a square grid an equivalent condition for reversibility is stated by Sutner in [168].

"$6 \not| m+1$ and for all odd $e > 3$, such that $e \mid m+1$ and $\tau \mid \rho_e$ irreducible : $\mathrm{dp}(\tau^+) \not| m+1$"

The author asks for a simple algorithm to test for the second property.

Here we view the problem from a different angle. We translate the condition for reversibility into a condition on the roots of $\pi_{m+1}(x)$. From this we are able to derive certain simple sufficient conditions for both reversibility and irreversibility. We also indicate why a simple characterization of reversibility is difficult. Note that reversibility may be determined in $O(N^3)$ steps (where N is the number of cells) by forming the adjacency matrix and reducing it to Hermite Canonical Form (HCF). The HCF will also provide the corank (dimension of the kernel) of the σ^+ operator.

The following result characterizes reversibility of σ^+-automata on a square $m \times m$ grid.

THEOREM 5.3.1. *The σ^+-automata on $m \times m$ grid is irreversible iff there exists roots α and β of $\pi_{m+1}(x)$, such that $\alpha + \beta = 1$. Here α and β belong to a suitable extension field over $GF(2)$.*

Proof : First note that the roots of $\pi^+_{m+1}(x)$ are $1 + \gamma_i$ where γ_i's are roots of $\pi_{m+1}(x)$. Then the result follows simply from the fact that σ^+-automata is irreversible iff $\pi_{m+1}(x)$ and

$\pi^+_{m+1}(x)$ are not relatively prime [15], iff $\pi_{m+1}(x)$ and $\pi^+_{m+1}(x)$ share a common root. Since we are working over a field of characteristic 2, this is the case iff $\alpha + \beta = 1$. □

From the above result we can see that irreversibility can occur in two ways.

1. There exists an irreducible factor $\tau(x)$ of $\pi_{m+1}(x)$, such that $\tau(x)$ has two roots α and β with $\alpha + \beta = 1$. Later, we show that such $\tau(x)$ satisfies $\tau(x) = \tau^+(x)$.

2. There exists two distinct irreducible factors $\tau_1(x)$ and $\tau_2(x)$ of $\pi_{m+1}(x)$ having roots α and β respectively with $\alpha + \beta = 1$. We will prove that under this condition $\tau_2(x) = \tau_1(1+x)$.

PROPOSITION 5.3.1. *If τ is an irreducible polynomial of depth d, then $\tau \,|\, \pi_n$, iff $d \,|\, n$*

Proof : If : $d \,|\, n$ implies $\pi_d \,|\, \pi_n$ (by Lemma 3.2.1.). Since τ is of depth d, $\tau \,|\, \pi_d$ and so $\tau \,|\, \pi_n$.

Only if : This case follows from Claim 1 in the proof of Theorem 2.1 of [168]. Here we reproduce the proof for the sake of completeness. Now $\tau \,|\, \pi_n$ and depth of τ being d, $\tau \,|\, \pi_d$ and so $\tau \,|\, \gcd(\pi_d, \pi_n)$. But by Lemma 3.2.1. $\gcd(\pi_d, \pi_n) = \pi_{\gcd(d,n)}$. Hence $\tau \,|\, \pi_{\gcd(d,n)}$. From the definition of depth it follows $d \le \gcd(d,n)$, which implies $d = \gcd(d,n)$ and so $d \,|\, n$. □

Now it is easy to see why Sutner's condition holds. It essentially says that for any irreducible polynomial $\tau(x)$, both $\tau(x)$ and $\tau^+(x)$ should not divide $\pi_{m+1}(x)$, and so by the above proposition the depths of both $\tau(x)$ and $\tau^+(x)$ should not divide $m+1$. (Note that $6 \,|\, m+1$ means that $\pi_2(x) \,|\, \pi_{m+1}(x)$ and $\pi_3(x) \,|\, \pi_{m+1}(x)$, and $\pi_2(x) = x$ and $\pi_3(x) = (1+x)^2$, see Proposition 5.3.2..) □

REMARK 5.3.1. *One can generate $\pi_{m+1}(x)$ and $\pi^+_{m+1}(x)$ and run the* gcd *algorithm on them to check if they are relatively prime. This procedure will in general be more time efficient than determining the Hermite Canonical Form and will require.less storage space. An interesting related problem is to compute $p^+(x)$ where $p(x)$ is an arbitrary polynomial over $GF(2)$.*

Let $c(m,i)$ and $c^+(m,i)$ denote the coefficient of x^i in $p(x)$ and $p^+(x)$ respectively. Then,

$$c^+(m, m-r) = \sum_{1 \le i_1 < i_2 < \ldots < i_r \le m} (1 + \alpha_{i_1})(1 + \alpha_{i_2}) \ldots (1 + \alpha_{i_r})$$

$$= \begin{pmatrix} m \\ r \end{pmatrix} + \begin{pmatrix} m-1 \\ r-1 \end{pmatrix} \bmod 2\, c(m, m-1)$$

$$+ \begin{pmatrix} m-2 \\ r-2 \end{pmatrix} \bmod 2\, c(m, m-2) + \ldots$$

$$+ \begin{pmatrix} m-r \\ r-r \end{pmatrix} \bmod 2\, c(m, m-r)$$

$$= \textstyle\sum_{i=0}^{r} D(m-i, r-i)\, c(m, m-i)$$

where $D(m-i, r-i) \; = \; \begin{pmatrix} m-i \\ r-i \end{pmatrix} \bmod 2$

So,

$$\begin{aligned} p^+(x) &= \textstyle\sum_{r=0}^{m} c^+(m, m-r)\, x^{m-r} \\ &= \textstyle\sum_{r=0}^{m} (\sum_{i=0}^{r} D(m-i, r-i)\, c(m, m-i))\, x^{m-r} \end{aligned}$$

Therefore if Pascal's triangle (modulo 2) is available upto integer m, then it is easy to compute $p^+(x)$.

PROPOSITION 5.3.2. *If any one of the following conditions hold then σ^+-automata on $m \times m$ grid is irreversible.*

1. $6 \mid m+1$

2. $5 \mid m+1$

3. $17 \mid m+1$

Proof :

1. $6 \mid m+1 \Leftrightarrow 2 \mid m+1$ and $3 \mid m+1 \Leftrightarrow \pi_2(x) \mid \pi_{m+1}(x)$ and $\pi_3(x) \mid \pi_{m+1}(x)$ $\Leftrightarrow x \mid \pi_{m+1}(x)$ and $(1+x) \mid \pi_{m+1}(x) \Leftrightarrow (1+x) \mid \pi^+_{m+1}(x)$ and $x \mid \pi^+_{m+1}(x) \Leftrightarrow x(1+x) \mid \gcd(\pi_{m+1}(x)\ , \pi^+_{m+1}(x)\ )$. Therefore $\pi_{m+1}(x)$ and $\pi^+_{m+1}(x)$ are not relatively prime and hence the result follows [15].

2. $\pi_5(x) = x^4 + x^2 + 1 = (x^2 + x + 1)^2$. Let α and β be the roots of $x^2 + x + 1$. Then $\alpha + \beta = 1$ and so if $\pi_5(x) \mid \pi_{m+1}(x)$ then α and β are also roots of $\pi_{m+1}(x)$ and so irreversibility occurs. But $\pi_5(x) \mid \pi_{m+1}(x)$ iff $5 \mid m+1$.

3. Consider $\tau(x) = x^4 + x + 1$. Then τ is an irreducible (in fact primitive) polynomial over $GF(2)$ with roots $\alpha, \alpha^2, \alpha^{2^2}, \alpha^{2^3}$ (see Theorem 5.3.2.). But $\alpha^4 + \alpha = -1 = 1$, since α is a root of τ. So if for some m, $\tau(x) \mid \pi_{m+1}(x)$, then σ^+-automata on $m \times m$ grid is irreversible. Since the depth of τ is 17 (from Appendix A) this can happen iff $17 \mid m+1$ (by Proposition 5.3.1.). □

To extend this idea (of 2 and 3 above) one should be able to

- characterise all irreducible polynomials $\tau(x)$ having two roots α and β with $\alpha + \beta = 1$.
- compute the depths of all such $\tau(x)$.

Next we obtain a complete characterisation of all irreducible polynomials $\tau(x)$, having two roots α and β with $\alpha + \beta = 1$. It turns out that these are the irreducible polynomials which are fixed under the map $x \to 1 + x$, i.e, $\tau(x) = \tau^+(x)$. *In what follows we will use some standard results on irreducible polynomials over finite fields which are all available in [110].* In particular, we will frequently use the following

THEOREM 5.3.2. *Let $\tau(x)$ be an irreducible polynomial of degree n over $GF(2)$ and let α be a root of $\tau(x)$ in its splitting field $GF(2^n)$. Then $\alpha, \alpha^2, \alpha^{2^2}, \ldots, \alpha^{2^{n-1}}$ are all the n roots of $\tau(x)$ in $GF(2^n)$.*

LEMMA 5.3.1. *Let $\tau_1(x)$ and $\tau_2(x)$ be two irreducible polynomials over $GF(2)$. Let α be a root of $\tau_1(x)$ and β be a root of $\tau_2(x)$, where both α and β are in the same suitable extension field over $GF(2)$, with $\beta = 1 + \alpha$. Then $\tau_2(x) = \tau_1(1+x)$.*

Proof : Since τ is irreducible, by Theorem 5.3.2. the roots of $\tau_1(x)$ are $\alpha, \alpha^2, \alpha^{2^2}, \ldots, \alpha^{2^{r_1-1}}$, where r_1 is the degree of $\tau_1(x)$. Similarly, the roots of $\tau_2(x)$ are $\beta, \beta^2, \beta^{2^2}, \ldots, \beta^{2^{r_2-1}}$, where r_2 is the degree of $\tau_2(x)$. Now,

$$\beta^{2^i} = (1 + \alpha)^{2^i} = 1 + \alpha^{2^i}$$

since we are working over a field of characteristic two.

Also $(1 + \alpha^{2^i})$ $(0 \leq i \leq r_1 - 1)$ are the roots of $\tau_1(1 + x)$ (which is also irreducible) and $(1 + \alpha^{2^i})$ $(0 \leq i \leq r_1 - 1)$ are all distinct. So $r_2 \geq r_1$. Now if $r_2 > r_1$, then $\tau_1(1+x)$ properly divides $\tau_2(x)$ which is a contradiction since $\tau_2(x)$ is irreducible. So $r_2 = r_1$ and all the roots of $\tau_1(1+x)$ are the roots of $\tau_2(x)$. Hence $\tau_2(x) = \tau_1(1+x)$. □

LEMMA 5.3.2. *Let $\tau(x)$ be an irreducible polynomial such that it has two roots α and β in its splitting field, with $\alpha + \beta = 1$. Then the degree of τ must be even.*

Proof : As stated above $\beta = \alpha^{2^i}$ for some $i \in \{0, \ldots, r-1\}$, where r is the degree of τ. So $\alpha + \beta = 1$ means

$$\alpha + \alpha^{2^i} = 1$$
$$\Rightarrow (\alpha + \alpha^{2^i})^{2^j} = 1 \quad 0 \leq j \leq r-1$$

This gives r equations,

$\alpha + \alpha^{2^i} = 1$

$\alpha^2 + \alpha^{2^{i+1}} = 1$

...

$\alpha^{2^{r-1}} + \alpha^{2^{i+r-1}} = 1$

Summing up the left and right hand sides we get,

$$\sum_{j=0}^{r-1} \alpha^{2^j} + \sum_{j=0}^{r-1} \alpha^{2^{i+j}} = r \bmod 2$$

But $\sum_{j=0}^{r-1} \alpha^{2^j} = \sum_{j=0}^{r-1} \alpha^{2^{i+j}}$

$\Rightarrow r \bmod 2 = 0$ and so r is even. □

LEMMA 5.3.3. *Let $\tau(x)$ be an irreducible polynomial over $GF(2)$. Then $\tau(x) = \tau^{+}(x)$ iff $\tau(x)$ has two roots α and β taken in a suitable extension field over $GF(2)$, such that $\alpha + \beta = 1$.*

Proof : If $\tau(x) = \tau^{+}(x)$, then the result is easy, so we only prove the other direction. Since, $\alpha + \beta = 1$ we have $\beta = \alpha + 1$, and so $\tau(x)$ has two roots α and $\alpha + 1$. But then $\tau^{+}(x)$ also has the roots α and $\alpha + 1$. This means that $\gcd(\tau(x), \tau^{+}(x))$ is non trivial. But then it must be whole of both $\tau(x)$ and $\tau^{+}(x)$. □

From the above two Lemmata we see that the irreducible polynomials which are fixed under the map $x \to 1 + x$ must have even degree. From the proof of Step 1 of Lemma 5.2.2., it follows that for any polynomial $p(x)$, if $p(x) = p^{+}(x)$, then degree of $p(x)$ must be even. Combined with Lemma 5.3.3., this provides an alternative proof of Lemma 5.3.2.. Next we have the following important result.

THEOREM 5.3.3. *Let $\tau(x)$ be an irreducible polynomial over $GF(2)$. Then $\tau(x) = \tau^{+}(x)$ iff $\tau(x) \,|\, (x^{2^i} + x + 1)$ for some i. In particular, if $\tau(x)$ is an irreducible factor of $x^{2^i} + x + 1$, for some i, then $\tau(x) = \tau^{+}(x)$ and $\tau(x)$ is of even degree.*

Proof : If $\tau(x) = \tau^{+}(x)$ then if α is a root of $\tau(x)$, $\alpha + 1$ is a root of $\tau^{+}(x)$, which must also be a root of $\tau(x)$. But the roots of $\tau(x)$ are of the form α^{2^i}. Thus it follows that $\alpha^{2^i} + \alpha + 1 = 0$ for some i. Since $\tau(x)$ is the minimal polynomial for α, it follows that $\tau(x) \,|\, (x^{2^i} + x + 1)$.

Again if $\tau(x) \,|\, (x^{2^i} + x + 1)$, then $\alpha^{2^i} + \alpha + 1 = 0$ for any root α of $\tau(x)$. Then $1 + \alpha = \alpha^{2^i}$ and hence both α and $\alpha + 1$ are roots of $\tau(x)$ in a suitable extension field. Therefore by the above lemma it follows that $\tau(x) = \tau^{+}(x)$. □

LEMMA 5.3.4. *Let $\tau(x)$ be an irreducible polynomial over $GF(2)$ of degree $2d > 0$, such that $\tau(x) = \tau^{+}(x)$. Then $\tau(x) \,|\, x^{2^d} + x + 1$ and $\tau(x) \not| x^{2^i} + x + 1$ for $i < d$.*

Proof : Let α be a root of $\tau(x)$ in a suitable extension field. Since $\tau(x) = \tau^{+}(x)$, we must have $1 + \alpha = \alpha^{2^k}$ for some $0 < k < 2d$. Hence

$$1 + \alpha^{2^k} = (1 + \alpha)^{2^k} = \alpha^{2^{2k}}$$
$$\Rightarrow \alpha^{2^{2k}} = \alpha$$
$$\Rightarrow 2k \equiv 0 \bmod 2d$$

This along with $0 < k < 2d$ implies $k = d$ and hence α satisfies $x^{2^d} + x + 1$. So $\tau(x)$ being the minimal polynomial for α, divides $x^{2^d} + x + 1$.

If possible, let $\tau(x) \,|\, x^{2^i} + x + 1$ for some $i < d$. Clearly, $i > 0$.
Then since α is a root of $x^{2^i} + x + 1$, $1 + \alpha = \alpha^{2^i}$
$\Rightarrow 1 + \alpha^{2^i} = \alpha^{2^{2i}}$
$\Rightarrow \alpha^{2^{2i}} = \alpha$
$\Rightarrow 2i \equiv 0 \bmod 2d$
$\Rightarrow d \,|\, i$, which is a contradiction. □

Corollary 5.3.1.

1. *The highest degree of all irreducible factors of* $x^{2^n} + x + 1$ *is* $2n$.

2. *If* $\tau(x)$ *of degree* k *is an irreducible factor of* $x^{2^n} + x + 1$, *then* $k \mid 2n$.

The second point of the above corollary is also in [110, pp 146].

Theorem 5.3.4. *Let* $\tau(x)$ *be an irreducible polynomial over* $GF(2)$ *of degree* $2d > 0$, *such that* $\tau(x) = \tau^{+}(x)$. *Then* $\tau(x) \mid x^{2^n} + x + 1$ *iff* $n \equiv d \bmod 2d$.

Proof : Since degree of $\tau(x)$ is $2d$, $\tau(x) \mid x^{2^{2d}} + x + 1$, so any root α of $\tau(x)$ satisfies $x^{2^{2d}} + x + 1$, i.e,

$$\alpha^{2^{2d}} + \alpha + 1 = 0$$

If $n \equiv d \bmod 2d$ then $n = 2dk + d$. So,

$$\begin{aligned} \alpha^{2^n} + \alpha + 1 &= \alpha^{2^{2dk+d}} + \alpha + 1 \\ &= \alpha^{2^d} + \alpha + 1 = 0 \end{aligned}$$

Hence $\tau(x) \mid x^{2^n} + x + 1$.

If $\tau(x) \mid x^{2^n} + x + 1$ then $\alpha^{2^n} + \alpha + 1 = 0$, where α is a root of $\tau(x)$ in a suitable extension field. Also $\tau(x) \mid x^{2^d} + x + 1$ and so $\alpha^{2^d} + \alpha + 1 = 0$. Hence $\alpha^{2^n} = \alpha^{2^d}$ which implies $n \equiv d \bmod 2d$. □

Definition 5.3.1. *Let,*

1. E_{2d} = *Product of all irreducible polynomials* $\tau(x)$ *of degree* $2d$, *such that* $\tau(x) = \tau^{+}(x)$.

2. C_{2d} = *Number of all irreducible polynomials* $\tau(x)$ *of degree* $2d$, *with* $\tau(x) = \tau^{+}(x)$.

Thus we can obtain the factorisation of $x^{2^n} + x + 1$ as

Lemma 5.3.5.

$$x^{2^n} + x + 1 = \prod_{n \equiv d \bmod 2d} E_{2d}$$

In fact we can state the result in a more convenient form.

Theorem 5.3.5.

$$x^{2^n} + x + 1 = \prod_{d|n, 2d \nmid n} E_{2d}$$

The proof of the theorem follows from the following result.

RESULT 5.3.1. *For some $d > 0$, $n \equiv d \bmod 2d$ iff $d \mid n$ and $2d \nmid n$.*

Proof : $d \mid n$ and $2d \nmid n$ implies $n = kd$ with k odd. Then,

$$\begin{aligned} n &= (k-1)d + d \\ &= \tfrac{k-1}{2}\,2d + d \end{aligned}$$

Hence $n \equiv d \bmod 2d$

If $n \equiv d \bmod 2d$ then $n = c\,2d + d = (2c+1)d$. So $d \mid n$ and $2d \nmid n$. □

Having obtained this we can now determine when a trinomial of the form $x^{2^m} + x + 1$ will divide another trinomial of the same form.

THEOREM 5.3.6. $x^{2^m} + x + 1 \mid x^{2^n} + x + 1$ *iff*

1. $D_2(m) = D_2(n)$ *and,*

2. $m \mid n$

where $D_2(m)$ is the greatest integer of the form 2^j that divides m.

Proof : Note that the conditions 1 and 2 are satisfied iff for each d such that $d \mid m$ and $2d \nmid m$, it follows that $d \mid n$ and $2d \nmid n$. Hence by the above theorem it follows that conditions 1 and 2 are satisfied iff $x^{2^m} + x + 1 \mid x^{2^n} + x + 1$. □

Next we count the number of irreducible polynomials $\tau(x)$ of degree $2n$ which satisfy $\tau(x) = \tau^{+}(x)$.

THEOREM 5.3.7. *Let $n = 2^k m$ with m odd and $m \geq 1$ and $k \geq 0$. Then,*

$$C_{2n} = \frac{1}{m} \sum_{e \mid m} \mu(e)\, 2^{\frac{n}{e}-1-k}$$

where $\mu(n)$ is the Mobius function (see [110]).

Proof : Using Theorem 5.3.5. we have,

$$2^n = \sum_{d \mid n, 2d \nmid n} 2dC_{2d}$$

$$\Rightarrow 2^{n-1} = \sum_{d \mid n, 2d \nmid n} dC_{2d}$$

Now the d's which satisfy $d \mid n$ and $2d \nmid n$ are of the form $d = 2^k e$ where $e \mid m$. Therefore,

$$2^{2^k m-1-k} = \sum_{e \mid m} eC_{2(2^k e)}$$

Using Mobius inversion we get,

$$\begin{aligned} mC_{2(2^k m)} &= \sum_{e|m} \mu(e)\, 2^{2^k(\frac{m}{e})-1-k} \\ \Rightarrow\ C_{2(2^k m)} &= \frac{1}{m}\sum_{e|m} \mu(e)\, 2^{2^k(\frac{m}{e})-1-k} \\ \Rightarrow\ C_{2n} &= \frac{1}{m}\sum_{e|m} \mu(e)\, 2^{\frac{n}{e}-1-k} \end{aligned}$$

□

This completes the characterization of the irreducible polynomials over $GF(2)$ which are fixed under the map $x \to 1 + x$. The computation of the depths of irreducible polynomials is in general difficult [168]. In Appendix A we present a complete factorisation of the first ten trinomials of the form $x^{2^i} + x + 1$. From what has been discussed above, it follows that this in effect lists all irreducible polynomials $\tau(x)$ of degree less than or equal to twenty such that $\tau(x) = \tau^+(x)$. The numbers in the first column gives the depth of the corresponding polynomial. So for any m, if any one of these numbers divide $m+1$, then σ^+-automata on $m \times m$ grid is irreversible. There does not seem to be any simple formula for the depth function even for this special class of irreducible polynomials. However, it is interesting to note from the values of depths in Appendix A that if either $2^{2i} - 1$ or $2^{2i} + 1$ $(4 \leq i \leq 10)$ divides $m + 1$, then irreversibility occurs.

The coefficient of x^i for this class of irreducible polynomials show certain interesting regularities. In fact, some of these can also be proved.

PROPOSITION 5.3.3. *For any irreducible polynomial $\tau(x)$ over $GF(2)$, with $\tau(x) = \tau^+(x)$, the following hold.*

1. $c_{2n-1} = n \bmod 2$
2. $c_{2n-2} = 1 + \binom{n}{2} \pmod 2$
3. $c_{2n-3} = \binom{n}{3} + (n-1) \pmod 2$

where $deg(\tau) = 2n$ and c_i is the coefficient of x^i in $\tau(x)$.

Proof : Since $\tau(x) = \tau^+(x)$, the roots of $\tau(x)$ can be written as α_i, $\alpha_i + 1$ $(0 \leq i \leq n-1)$ accounting for $2n$ roots. Then,

1. $$\begin{aligned} c_{2n-1} &= \textstyle\sum_{i=0}^{n-1} \alpha_i + \sum_{i=0}^{n-1} (1+\alpha_i) \\ &= n \bmod 2 \end{aligned}$$

2. Since $deg(\tau) = 2n$, $\tau(x)|x^{2^n} + x + 1$. So for any root α of $\tau(x)$, $\alpha^{2^n} + \alpha + 1 = 0$ which implies $\alpha^{2^n} = \alpha + 1$.

Then it follows that α^{2^i}, $1+\alpha^{2^i}$ $(0 \leq i \leq n-1)$ are all the distinct roots of $\tau(x)$. Let $\alpha_i = \alpha^{2^i}$ for $0 \leq i \leq n-1$. Then, $\sum_{i=0}^{n-1} \alpha_i^2 = 1 + \sum_{i=0}^{n-1} \alpha_i$ (using $\alpha^{2^n} = \alpha + 1$) and so $\sum_{i=0}^{n-1}(\alpha_i + \alpha_i^2) = 0$ Now,

$$c_{2n-2} = \sum_{1 \leq i < j \leq 2n} \beta_i \beta_j$$

where β_is are the $2n$ distinct roots of $\tau(x)$. Therefore,

$$\begin{aligned}
c_{2n-2} &= \textstyle\sum_{0 \leq i<j \leq n-1} \alpha_i\alpha_j + \sum_{0 \leq i<j \leq n-1}(\alpha_i+1)(\alpha_j+1) + \\
&\quad \textstyle\sum_{0 \leq i \leq n-1} \sum_{0 \leq j \leq n-1} \alpha_i(1+\alpha_j) \\
&= \textstyle\sum_{0 \leq i<j \leq n-1} \alpha_i\alpha_j + \begin{pmatrix} n \\ 2 \end{pmatrix} \bmod 2 + \begin{pmatrix} n-1 \\ 1 \end{pmatrix} \bmod 2 \sum_{0 \leq i \leq n-1} \alpha_i + \\
&\quad \textstyle\sum_{0 \leq i<j \leq n-1} \alpha_i\alpha_j + \sum_{i=0}^{n-1} \alpha_i(1+\alpha_i) + \sum_{0 \leq i,j \leq n-1, i \neq j} \alpha_i(1+\alpha_j) \\
&= \begin{pmatrix} n \\ 2 \end{pmatrix} \bmod 2 + \begin{pmatrix} n-1 \\ 1 \end{pmatrix} \bmod 2 \textstyle\sum_{0 \leq i \leq n-1} \alpha_i + \sum_{i=0}^{n-1} \alpha_i(1+\alpha_i) + \\
&\quad \begin{pmatrix} n-1 \\ 1 \end{pmatrix} \bmod 2 \textstyle\sum_{0 \leq i \leq n-1} \alpha_i + 2 \sum_{0 \leq i<j \leq n-1} \alpha_i\alpha_j \\
&= \begin{pmatrix} n \\ 2 \end{pmatrix} \bmod 2 + \textstyle\sum_{i=0}^{n-1}(\alpha_i + \alpha_i^2) \\
&= 1 + \begin{pmatrix} n \\ 2 \end{pmatrix} \pmod 2
\end{aligned}$$

3. The coefficient c_{2n-3} can be written as

$$c_{2n-3} = \sum_{1 \leq i<j<k \leq 2n} \beta_i\beta_j\beta_k$$

where β_is are the $2n$ distinct roots of $\tau(x)$. Then the result follows using a similar, though a bit more tedious, argument as in 2 above. □

Note that irreversibility can also occur in another way, i.e if $\tau(x)$ and $\tau^+(x)$ both divide $\pi_{m+1}(x)$, with $\tau(x) \neq \tau^+(x)$. But this means that the depths of both $\tau(x)$ and $\tau^+(x)$ divides $m+1$. It is also difficult to determine this.

Now we provide sufficient conditions for reversibility. A very easy condition is the following

PROPOSITION 5.3.4. *(cf. [15]) If $m+1 = 2^k$ for some k, then σ^+-automata on $m \times m$ grid is reversible.*

Proof : In this case $\pi_{m+1}(x) = x^m$ (see [15, 168]) and $\pi^+_{m+1}(x) = (1+x)^m$. Therefore the two are relatively prime. □

LEMMA 5.3.6. *If the following conditions hold then the σ^+-automata on a $m \times m$ grid is reversible.*

1. $m+1$ is a prime with $\phi(m+1) = 2\, sord_{m+1}(2)$.

2. $m+1 \equiv 3 \bmod 4$.

Proof : Condition 1 implies that $\pi_{m+1}(x) = \tau^2(x)$ with τ irreducible (see Chapter 3) and condition 2 implies that the degree of τ is odd. Therefore τ and as a result $\pi_{m+1}(x)$ cannot have roots α and β with $\alpha + \beta = 1$ (by Lemma 5.3.2.). Hence the result follows by Lemma 5.3.1.. □

The first ten primes which satisfy the conditions of the above lemma, and the corresponding π-polynomials are given in Table 1.

$m+1$	$\pi_{m+1}(x)$
3	$1+x^2$
7	$1+x^4+x^6$
11	$1+x^2+x^4+x^8+x^{10}$
19	$1+x^2+x^8+x^{10}+x^{12}+x^{16}+x^{18}$
23	$1+x^4+x^6+x^8+x^{16}+x^{20}+x^{22}$
47	$1+x^8+x^{12}+x^{14}+x^{16}+x^{32}+x^{40}+x^{44}+x^{46}$
59	$1+x^2+x^4+x^{32}+x^{34}+x^{36}+x^{48}+x^{50}+x^{52}+x^{56}+x^{58}$
67	$1+x^2+x^{32}+x^{34}+x^{48}+x^{50}+x^{56}+x^{58}+x^{60}+x^{64}+x^{66}$
71	$1+x^4+x^6+x^{32}+x^{36}+x^{38}+x^{48}+x^{52}+x^{54}+x^{56}+x^{64}+x^{68}+x^{70}$
79	$1+x^8+x^{12}+x^{14}+x^{32}+x^{40}+x^{44}+x^{46}+x^{48}+x^{64}+x^{72}+x^{76}+x^{78}$

Table 1: The first ten primes which satisfy Lemma 5.3.6.

Thus we see that reversibility of σ^+-automata on square $m \times m$ grid shows an extremely rich behaviour. It would indeed be very interesting to obtain a full characterisation of reversibility in terms of number theoretic properties of m.

Chapter 6

Multidimensional CA

6.1 Introduction

In this chapter we will study multidimensional CA. The underlying geometry is a multidimensional grid and the local rule is the sum (modulo 2) of the states of all the orthogonal neighbours. One may also consider a cell to be a neighbour of itself. The multidimensional grid may be folded in some or all dimensions. Following the correspondence between CA and σ-automata introduced in the previous chapters, here also we will consider a multidimensional CA to be equivalent to σ-automata on a graph which is suitably defined to be a product of path and cycle graphs. For a σ^+-automaton we will consider self loops to be present at each node in the underlying graph. Later, we make all this precise. In this chapter, we will use terminology and results from Chapter 3.

The class of σ-automata have been studied over arbitrary graphs and were first studied by Lindenmayer [111]. Study of σ-automata is related to the study of σ-game, which is a combinatorial game first introduced by Sutner in [166] and is based on the battery operated toy MERLIN [136]. In [166], Sutner reduces the study of σ-game to that of a suitably constructed σ-automaton. Combinatorial techniques are then used to obtain expressions for the dimension of the kernel of σ-automata on product graphs of the form $G_1 \times G_2$ (see also [161]). For the special case of product graphs of the form $P_m \times P_n$, where P_i is a path graph on i vertices, it was shown that the automaton is invertible iff m+1 and n+1 are relatively prime.

Barua and Ramakrishnan in [15] consider the product graph $P_m \times P_n$ as a two dimensional grid and reduce the σ-game to the study of invertibility of cellular automata on two dimensional array. The global CA rule is considered to be a linear transformation of the form $AX + XB$, where X is a two dimensional CA configuration, regarded as a 0-1 matrix, and A and B are S-matrices. Analysis of this equation provides an algebraic proof for the dimension of the kernel of the linear map.

A natural consequence is to consider σ-games (and hence σ-automata) on multidimensional grids. Sutner in [166, 161] introduced combinatorial techniques to tackle the multidimensional case. For product graphs of the form $G = H \times P_n$, there is an expression relating the coranks (dimension of kernel) of G and H, viz

$$cork\,\sigma(G) = cork\,\pi_{n+1}(\sigma(H))$$

where $\sigma(G)$ denotes the global rule for σ-automaton on graph G and $\pi_{n+1}(x)$ is the characteristic polynomial for the global rule of the σ-automaton on P_n. However analysis of $\pi_{n+1}(\sigma(H))$ seems to be difficult. Though this is a general result, for the special case where G is a multidimensional grid, we use a suitable transformation to obtain a much more elegant representation of the global rule in terms of Kronecker products. Using this representation we attack the question of invertibility of σ-automata.

Finite linear cellular automata on multidimensional grids have been considered before [115]. Martin et al [115], used polynomials of several variables to tackle multidimensional configuration. It is a difficult technique and known results on finite multidimensional cellular automata are few. However, our approach yields interesting results on the invertibility of finite multidimensional linear cellular automata. Using the Kronecker product representation of the global rule we obtain the characteristic roots of the global rule in terms of the roots of π-polynomials. In special cases, this is then related to the number theoretic properties of the number of dimensions and the lengths in each dimension.

6.2 Preliminaries

DEFINITION 6.2.1.

1. *A k-dimensional grid is a multidimensional array $G[0..l_1-1][0..l_2-1]...[0..l_k-1]$, with length l_i in the i^{th} dimension. It will be denoted by $G(l_1, \ldots, l_k)$. Any cell of the array is uniquely identified by a tuple $(i_1, ..., i_k)$, with $0 \le i_j < l_j$ and $1 \le j \le k$ and has a finite set of neighbours as defined below.*

2. *The neighbours of any cell $(i_1, ..., i_k)$ are given by $(i_1, ..., i_j \pm 1, .., i_k)$ with $1 \le j \le k$. If the j^{th} component has a periodic boundary condition, then $i_j \pm 1$ is evaluated modulo l_j. If the j^{th} component has a null boundary condition, then there are no neighbours corresponding to -1 and l_j in the j^{th} component.*

3. *If all dimensions have null boundary condition then the grid is a null boundary grid. If all dimensions have periodic boundary condition then the grid is a folded grid. If some dimensions have null boundary condition and some have periodic boundary condition then we will call the grid a mixed grid.*

4. *The grid is symmetric if the lengths of all dimensions are equal. Else, it is an asymmetric grid. A k-dimensional symmetric grid of length l will be denoted by $G_k(l)$.*

A k-dimensional grid $G(l_1,\ldots,l_k)$ has $\prod_{i=1}^{i=k} l_i$ cells. It is also possible to define a k-dimensional grid (folded, null or mixed) as a finite product of path or cycle graphs (see [161]).

Definition 6.2.2.

1. *A σ-automaton on a multidimensional grid is a cellular automaton where*

   a) *The state of each cell belongs to $GF(2)$.*

   b) *The next state for any cell is the sum (modulo 2) of the current states of its neighbours. (This specifies the local rule for the σ-automaton).*

2. *A σ^+-automaton is defined similarly, the only difference being the fact that in this case the cell itself is also considered to be its neighbour.*

To keep this chapter self-contained we restate some terminology from Chapter 3 for a σ-automata on a multidimensional grid. An assignment of values 0 or 1 to the cells of a k-dimensional grid is called a *configuration.* We define C to be the set of all configurations. The *global transition rule* for a σ-automaton is a map $T : C \to C$, where $T(c)$ is the configuration obtained from configuration c by applying the local rule to each cell. The global dynamics of a σ-automaton is determined by T and is best expressed in terms of the *State Transition Diagram* (STD), which is a directed graph $D = (V, A)$ where $V = C$ and $(c_1, c_2) \in A$ iff $T(c_1) = c_2$. It is easy to see that the STD for a σ-automaton consists of disjoint components, where each component has a cycle with trees of height ≥ 0 rooted on each cycle vertex [115]. The σ-automaton is said to be *invertible* iff T is a bijection. Also we can consider C to be a vector space over $GF(2)$ and then T is a linear transformation from C to C. So T is invertible iff dim ker $T = 0$. With respect to the standard basis, T is uniquely determined by a matrix M. Then T is invertible iff M is invertible. This M is called the *transition matrix* for the σ-automaton. In this chapter, we will be concerned with the representation and invertibility of M.

6.3 Generalised S-Matrix

In this section, we obtain a representation for the linear transformation defined by the global rule of a σ-automaton on a *null boundary multidimensional grid.* For the one dimensional case this is given by an S-matrix of order n. For the two dimensional case, a representation was obtained in [15] as $AX + XB$, where A and B are S-matrices of proper order. We will show that for a k dimensional grid, the global rule can be represented as a sum of Kronecker product of matrices. We will use this representation in later sections to perform an algebraic analysis of the linear map.

In the following discussion we will consider a multidimensional configuration as a vector in a suitable vector space. To do this, we will need to map a multi dimensional configuration to a one dimensional vector. For this we use the standard one-to-one correspondence used by compilers [10]. Consider a k-dimensional grid $G(l_1, \ldots, l_k)$. Then the coordinate $(i_1, \ldots, i_k)$, $0 \leq i_r \leq l_r - 1$, becomes the j^{th} component of a vector where,

$$\begin{aligned} j &= (\ldots((i_1 l_2 + i_2) l_3 + i_3) \ldots) l_k + i_k \\ &= i_1 l_2 l_3 \ldots l_k + i_2 l_3 \ldots l_k + \ldots + i_{k-1} l_k + i_k \end{aligned} \qquad \text{(I)}$$

In other words, j is the position of $(i_1, \ldots, i_k)$ in the lexicographic ordering of the k-tuples. Thus each such k dimensional configuration is identified with a vector in a vector space of dimension $L = \prod_{i=1}^{i=k} l_i$. Hence we can consider the global rule of a σ-automaton to be represented by a square binary matrix of order L. This matrix we characterise as a sum of Kronecker products and refer to as a *generalised S-matrix.* The name is justified as it turns out that the matrix is block tridiagonal. We obtain the matrix as follows. For each cell $(i_1, \ldots i_k)$ of the array, we consider the j^{th} row in the matrix, where j is given by (I). The L entries in the row correspond to the L cells in the array and the c^{th} column in the j^{th} row is 1 iff the c^{th} cell is a neighbour of the j^{th} cell. It is easy to see that a generalised S-matrix is sparse, symmetric and has all entries on the main diagonal to be zero.

Next we prove a technical lemma, which will be used in this and later sections to express a generalised S-matrix as a sum of Kronecker products. We denote the entry in the r^{th} row and c^{th} column of a matrix A by $A(r, c)$.

LEMMA 6.3.1. *Consider a k-dimensional grid $G(l_1, \ldots, l_k)$ and let d be the map from k dimension to one dimension, i.e*

$$d : \{(i_1, \ldots, i_k) : 0 \leq i_j < l_j,\ 1 \leq j \leq k\} \longrightarrow \{J : 0 \leq J < L\}$$

given by,

$$d(i_1, \ldots, i_k) = (\ldots((i_1 l_2 + i_2) l_3 + i_3) \ldots) l_k + i_k \qquad (II)$$

Consider the matrix $T = A_1 \otimes \ldots \otimes A_k$, where A_i is a square matrix of order l_i.

Let $\{(x_i, y_i) : 1 \leq i \leq k,\ 0 \leq x_i, y_i < l_i\}$ be a set of ordered pairs and let

$$\begin{aligned} X &= d(x_1, \ldots, x_k) \\ Y &= d(y_1, \ldots, y_k) \end{aligned}$$

Then $T(X, Y) = 1$ iff $A_i(x_i, y_i) = 1$, for all i.

Proof : The proof is by induction on r, with $1 \leq r \leq k$. For $r = 1$, the result is trivial. Assume that the result holds for $r - 1$. Then we have to show that the result holds for

$$\begin{aligned} T_r &= A_1 \otimes \ldots \otimes A_r = A_1 \otimes \ldots \otimes A_{r-1} \otimes A_r \\ &= T_{r-1} \otimes A_r \end{aligned}$$

where T_{r-1} is a square matrix of order $L_{r-1} = l_1 \ldots l_{r-1}$. Given r pairs $(x_1, y_1), \ldots, (x_r, y_r)$ with $0 \le x_i, y_i < l_i$, let,
$X_{r-1} = d(x_1, \ldots, x_{r-1})$ and $Y_{r-1} = d(y_1, \ldots, y_{r-1})$. Then,
$X_r = d(x_1, \ldots, x_r) = l_r X_{r-1} + x_r$ and $Y_r = d(y_1, \ldots, y_r) = l_r Y_{r-1} + y_r$. So,

$$T_r = T_{r-1} \otimes A_r = \begin{bmatrix} t_{11}A_r & t_{12}A_r & \ldots & t_{1Y_{r-1}}A_r & \ldots & t_{1L_{r-1}}A_r \\ t_{21}A_r & t_{22}A_r & \ldots & t_{2Y_{r-1}}A_r & \ldots & t_{2L_{r-1}}A_r \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ t_{X_{r-1}1}A_r & t_{X_{r-1}2}A_r & \ldots & t_{X_{r-1}Y_{r-1}}A_r & \ldots & t_{X_{r-1}L_{r-1}}A_r \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ t_{L_{r-1}1}A_r & t_{L_{r-1}2}A_r & \ldots & t_{L_{r-1}Y_{r-1}}A_r & \ldots & t_{L_{r-1}L_{r-1}}A_r \end{bmatrix}$$

where $t_{ij} = T_{r-1}(i,j)$. From this we get that $T_r(X,Y) = 1$ iff $t_{X_{r-1},Y_{r-1}} = 1$ and $A_r(x_r, y_r) = 1$. This is so iff $T_{r-1}(X_{r-1}, Y_{r-1}) = 1$ and $A_r(x_r, y_r) = 1$ iff $A_i(x_i, y_i) = 1, \ \forall\, 1 \le i \le r-1$ (by induction hypothesis) and $A_r(x_r, y_r) = 1$ iff $A_i(x_i, y_i) = 1, \ \forall\, 1 \le i \le r$. □

Now we can present the main result of this section.

THEOREM 6.3.1. *For the σ-automaton on $G(l_1, \ldots, l_k)$ with null boundary condition, the transition matrix is a generalised S-matrix T, defined by,*

$$T = I_{l_1} \otimes I_{l_2} \otimes \ldots \otimes S_{l_k} + I_{l_1} \otimes I_{l_2} \otimes \ldots \otimes S_{l_{k-1}} \otimes I_{l_k} + \ldots + S_{l_1} \otimes I_{l_2} \otimes \ldots \otimes I_{l_k}. \quad (III)$$

Proof : Let $I = (i_1, \ldots, i_k)$ be any cell of the underlying k-dimensional grid. Then its neighbours are given by $(i_1, \ldots, i_j \pm 1, \ldots, i_k) \quad 1 \le j \le k$.

Let T_j be the global transformation corresponding to the local rule where we consider neighbours in the j^{th} dimension only. Then, by linearity, we can write

$$T = \sum_{j=1}^{k} T_j = T_1 + \ldots + T_k$$

If we can show that $T_j = I_{l_1} \otimes \ldots \otimes S_{l_j} \otimes \ldots \otimes I_{l_k}$ then we are done.

Let $X = d(i_1, \ldots, i_j, \ldots, i_k)$ where d is as in (II) above. Let,

$$X_1 = d(i_1, \ldots, i_j - 1, \ldots, i_k)$$

$$X_2 = d(i_1, \ldots, i_j + 1, \ldots, i_k)$$

Here we assume that both $i_j - 1$ and $i_j + 1$ lie between 0 and $l_j - 1$. The other cases are similar. Hence we have,

$$T_j(X,Y) = 1 \text{ iff } Y = X_1 \text{ or } Y = X_2$$

Let the entry (X,C) in $P_j = I_{l_1} \otimes \ldots \otimes S_{l_j} \otimes \ldots \otimes I_{l_k}$ be 1. Then $C = d(x_1,\ldots,x_k)$ for some $x_1,\ldots,x_k$ with $0 \le x_i < l_i$. By the above lemma $P_j(X,C) = 1$ iff
$I_{l_t}(i_t,x_t) = 1 \ \ for \ \ t \neq j \, , \ 1 \le t \le k$ and $S_{l_j}(i_j,x_j) = 1$.
But this happens iff $x_t = i_t$ for $t \neq j \, , \ 1 \le t \le k$ and $x_j = i_j \pm 1$.

Thus $P_j(X,C) = 1$ iff $C = X_1$ or $C = X_2$. But this means that each row of P_j and T_j are equal. Therefore, $T_j = I_{l_1} \otimes \ldots \otimes S_{l_j} \otimes \ldots \otimes I_{l_k}$ and hence the result follows. □

The proof actually provides a recurrence for the generalised S-matrix. This recurrence become particularly interesting when the lengths are equal (a symmetric grid). In this case,

$$T^{(k)} = I_l \otimes I_l \otimes \ldots \otimes S_l + \ldots + S_l \otimes I_l \otimes \ldots \otimes I_l$$

From now on we will follow the convention of dropping the subscript l when the lengths are equal. Also we will denote by $I^{(k)}$ the identity matrix $I_l \otimes \ldots \otimes I_l = I_{l^k}$. Then we can neatly write the recurrence as,

$$T^{(k)} = I \otimes T^{(k-1)} + S \otimes I^{(k-1)} \qquad (IV)$$

Thus our investigation of the invertibility of a symmetric σ-automaton is reduced to the study of non-singularity of $T^{(k)}$ as given by (IV).

In [15], the global transformation of a two dimensional CA is represented in the following way. For an $m \times n$ grid, the global map T is given by $T(X) = S_m X + X S_n$ where X is an $m \times n$ matrix representing a particular configuration of the CA. This matrix equation is completely equivalent to the map $Tx = (S_m \otimes I_n + I_m \otimes S_n)x$, where x is a vector formed from X using the map given in (I). This result can be found in any standard book on matrix algebra [13]. Thus our representation for the multidimensional case is a natural generalisation of the two dimensional case as used in [15]. In what follows we will require some basic results on Kronecker products and resultants. The Appendix contains some preliminary results. The reader is referred to [13] for Kronecker products and to [120] for resultants.

Next, we note several basic properties of generalised S-matrix on symmetric grids.

PROPOSITION 6.3.1. *(a)* $(T^{(k)})^{2^i} = I \otimes (T^{(k-1)})^{2^i} + S^{2^i} \otimes I^{(k-1)}$

(b) $T^{(2k)} = I^{(k)} \otimes T^{(k)} + T^{(k)} \otimes I^{(k)}$

(c) $T^{(2k+1)} = I^{(k)} \otimes T^{(k+1)} + T^{(k)} \otimes I^{(k+1)}$

Proof : (a) For square matrices A, B, C, D we have $(A \otimes B)(C \otimes D) = (AC \otimes BD)$ (see Proposition B.0.1.(8)). Also since we are working over a field of characteristic 2, and multiplication with identity commutes, using (IV), we get,

$$\begin{aligned} (T^{(k)})^{2^i} &= (I \otimes T^{(k-1)} + S \otimes I^{(k-1)})^{2^i} \\ &= (I \otimes T^{(k-1)})^{2^i} + (S \otimes I^{(k-1)})^{2^i} \\ &= I \otimes (T^{(k-1)})^{2^i} + S^{2^i} \otimes I^{(k-1)} \end{aligned}$$

(b) and (c) follow from (IV) by induction on k. □

PROPOSITION 6.3.2. *Let $p(x)$ be an annihilating polynomial for S_l, such that the powers of x are of the form 2^i. Then $p(x)$ annihilates $T^{(k)}$ as given by (IV), unless k is even and $p(x)$ has a constant term, in which case $p(x) - 1$ annihilates $T^{(k)}$.*

Proof : Expanding (IV) we can write,

$$T^{(k)} = I \otimes I \otimes \ldots \otimes S + I \otimes I \otimes \ldots \otimes S \otimes I + \ldots + S \otimes I \otimes \ldots \otimes I$$

which is a sum of k terms. Then using Proposition 6.3.1., we can write,

$$(T^{(k)})^{2^i} = I \otimes I \otimes \ldots \otimes S^{2^i} + I \otimes I \otimes \ldots \otimes S^{2^i} \otimes I + \ldots + S^{2^i} \otimes I \otimes \ldots \otimes I$$

Since the powers of x in $p(x)$ are of the form 2^i, and $p(x)$ annihilates S_l, it follows $p(x)$ also annihilates $T^{(k)}$. To see the special case, just note that when k is odd, $I^{(k)}$ added k times in just $I^{(k)}$. This however is not possible when k is even. □

REMARK 6.3.1. *Using the above proposition it can be shown that for $l = 2, 4, 6$ a σ-automaton on a k-dimensional null boundary grid is invertible iff k is odd. For the case $l = 2$, there is a nice geometric argument. In this case, any cell is identified by a k-tuple $(a_1, \ldots, a_k)$ where each a_i is 0 or 1. Since we are considering null boundary condition any cell has exactly k neighbours. Moreover two cells $v_1 = (x_1, \ldots, x_k)$ and $v_2 = (y_1, \ldots, y_k)$ can either share two neighbours or no neighbours. To see this note that if the Hamming distance between v_1 and v_2 is greater then two, then they share no neighbours and if it is one then they are adjacent cells and hence also do not share any neighbour. Thus v_1 and v_2 share neighbours iff their Hamming distance is two and in this case they share exactly two neighbours. Now if T be the matrix representing the global transformation of the σ-automaton, then T^2 is I or 0 according as k is odd or even. This is because to find T^2 we have to consider the inner product of the i^{th} row $\underset{\sim i}{r}$ and the j^{th} column $\underset{\sim j}{c}$, and by the above discussion and symmetry, this product is $k \bmod 2$ if $i = j$ else it is 0. So if k is odd the STD consists of disjoint cycles each of length one or two and if k is even then the STD consists of a single tree rooted on the null configuration having height 1. Also the structure of the STD in this case is independent of the number of dimensions.*

The above can also be proved using the following result from [161]. For product graphs $G = H \times P_n$, the coranks of rule σ on G and H are related by,

$$cork\ \sigma(G) = cork\ \pi_{n+1}(\sigma(H))$$

Then by induction it can be shown that for a k dimensional structure the corank is 0 or k according as k is odd or even.

Let $T^{(k)}$ be invertible. Then, as we will prove in the next section, it necessarily follows that l is even and k is odd. Since l is even we know from Theorem 3.4.3. that the exponent of S_l divides $2^{1+sord_{l+1}(2)} - 2$ and S_l satisfies

$$p(x) = x^{2^{1+sord_{l+1}(2)}-2} + 1$$

Thus $x^2\,p(x)$ is a polynomial where the powers of x are of the form 2^i (such polynomials are called linearised polynomials [110]). Hence $T^{(k)}$ satisfy $x^2\,p(x)$ and since it is invertible it also satisfies $p(x)$. Thus in this case the exponent of $T^{(k)}$ also divide $2^{1+sord_{l+1}(2)} - 2$. Note that if l is even, then $T^{(k)}$ satisfies $x^2\,p(x)$ whether k is odd or even.

REMARK 6.3.2. *The matrices $T^{(k)}$ have another interesting feature. The above discussion implies that if l is fixed then for infinitely many k, $T^{(k)}$ will have the same minimal polynomial.*

6.4 Symmetric Grids

In this section we consider $\sigma(\sigma^+)$-automata on *symmetric null boundary grids.*

6.4.1 Invertibility of σ-Automata

We obtain necessary and sufficient condition for the invertibility of σ-automata on symmetric, null boundary grids and relate this condition to the number theoretic properties of k, the number of dimensions and l, the length in any dimension.

THEOREM 6.4.1. *For the σ-automaton on $G_k(l)$, the following hold*

(a) If l is odd, then the automaton is non-invertible.

(b) If k is even, then the automaton is non-invertible.

Proof : (a) By induction on k. When $k = 1$, l is odd implies $T^{(k)} = S_l$ is singular. So assume $k > 1$. By (IV) we have,

$$T^{(k)} = I_l \otimes T^{(k-1)} + S_l \otimes I^{(k-1)}$$

By induction hypothesis, $T^{(k-1)}$ is singular and so x divides the characteristic polynomial $p(x)$ for $T^{(k-1)}$. Also, since l is odd $x \mid \pi_{l+1}$. Therefore, $p(x)$ and π_{l+1} share a common root and hence $T^{(k)}$ is non-invertible (see Lemma B.0.2.).

(b) Suppose $k = 2r$. Then,

$$T^{(k)} = T^{(2r)} = I^{(r)} \otimes T^{(r)} + T^{(r)} \otimes I^{(r)}$$

Hence, by Lemma B.0.2. $T^{(k)}$ is non-invertible. □

The case when l is even and k is odd, shows more interesting behaviour. *It is the only case under which $T^{(k)}$ can be invertible.* To analyse the behaviour of $T^{(k)}$ we need the following result.

THEOREM 6.4.2. *Let*

$$T^{(k)} = I_l \otimes I_l \otimes \ldots \otimes S_l + I_l \otimes \ldots \otimes S_l \otimes I_l + \ldots + S_l \otimes I_l \otimes \ldots \otimes I_l$$

Then α is a root of its characteristic polynomial $p(x)$ iff α is of the form

$$\alpha_1 + \ldots + \alpha_k$$

where α_i's are the roots of π_{l+1} over the splitting field of π_{l+1}.

Proof : By induction on k.

For $k = 2$ this follows from Lemma B.0.3..

Assume it to be true for $k-1$ dimensions. Then,

$$T^{(k)} = I_l \otimes T^{(k-1)} + S_l \otimes I^{(k-1)}$$

So α is a root of $p(x)$ iff it is of the form $\beta + \alpha_k$, where β is any root of the characteristic polynomial for $T^{(k-1)}$ and α_k is any root of π_{l+1} (see Lemma B.0.3.). But by induction hypothesis β is of the form $\alpha_1 + \ldots + \alpha_{k-1}$. Hence α is a root of $p(x)$ iff it is of the form $\alpha_1 + \ldots + \alpha_k$. □

COROLLARY 6.4.1. *$T^{(k)}$ given by (IV) is non-invertible iff for some choice of $\alpha_1, \ldots, \alpha_k$ of the roots of π_{l+1}, we have $\alpha_1 + \ldots + \alpha_k = 0$* .

Proof : $T^{(k)}$ is non-invertible iff 0 is a root of the characteristic polynomial for $T^{(k)}$ iff $\alpha_1 + \ldots + \alpha_k = 0$ for some choice of α_i's. □

This corollary provides a necessary and sufficient condition for $T^{(k)}$ to be invertible in terms of the roots of π_{l+1}. We know that invertibility can occur only when l is even and k is odd. Note that the other cases can also be derived by examining the sum $\alpha_1 + \ldots + \alpha_k$. This constitutes an alternative proof to the approach in Theorem 6.4.1..

Before proceeding we restate Lemma 5.2.1.(2) as

LEMMA 6.4.1. *When l is even, π_{l+1} contains both the terms x^l and x^{l-2}.*

REMARK 6.4.1. *Hence for $l = 2r$, $p(x) = \sqrt{\pi_{2r+1}(x)}$ is of degree r and contains the term x^{r-1} and so the sum of the roots of $p(x)$ is 1, since this is the coefficient of x^{r-1}.*

Next we derive sufficient conditions for invertibility in terms of number theoretic properties of k and l.

THEOREM 6.4.3. *Consider the σ-automaton on $G_k(l)$. If the following three conditions hold, then the σ-automaton is invertible.*

1. *k is odd,*
2. *$l+1$ is an odd prime,*
3. *$2\,sord_{l+1}(2) = \phi(l+1) = l$. In this case, $\pi_{l+1} = \rho^2$ with ρ irreducible.*

Proof : The fact $\pi_{l+1} = \rho^2$ follows from Lemma 3.2.2.(b) The roots of the characteristic polynomial $p(x)$ for $T^{(k)}$ are of the form $\alpha_1 + \ldots + \alpha_k$, where α_i's are roots of π_{l+1}. To show that $T^{(k)}$ is invertible we have to show that the sum $\alpha_1 + \ldots + \alpha_k$ cannot be 0 for any choice of α_i and for any odd k. Now,

$$\pi_{l+1} = \rho^2 \text{ , where } \rho \text{ is an irreducible polynomial}$$

Suppose $l = 2r$. Then by the above Lemma, ρ has both the terms x^r and x^{r-1}. Also all the distinct roots of π_{l+1} are given by all the distinct roots of ρ. Since degree of ρ is r, and ρ is irreducible, it has r distinct roots $\alpha_1 \ldots \alpha_r$ (by Theorem 5.3.2.) and the sum

$$\alpha_1 + \ldots + \alpha_r = 1 \text{ , since } \rho \text{ has the term } x^{r-1}$$

When analysing the sum $\alpha_1 + \ldots + \alpha_k$, we can consider all of them to be distinct. Since, in a field of characteristic 2 equal roots cancel in pairs, without disturbing the parity of k.

Thus we have to show that $\alpha_1 + \ldots + \alpha_t$ cannot be 0 for odd $t \leq r$ and for distinct α_i's.

Since ρ is irreducible all its roots are of the form $\beta, \beta^2, \beta^{2^2}, \ldots, \beta^{2^{r-1}}$, where β is any root of ρ. Then it follows that ρ is the minimal polynomial for β.

If possible, let for some odd $t \leq r$,

$$\alpha_1 + \ldots + \alpha_t = 0$$

Then, $\beta^{2^{i_1}} + \ldots + \beta^{2^{i_t}} = 0$

Hence, β satisfies $q(x) = x^{2^{i_1}} + \ldots + x^{2^{i_t}}$ and therefore $\rho \,|\, q(x)$. So all roots of ρ are roots of $q(x)$ and we get the following r relations.

$$\begin{aligned} \beta^{2^{i_1}} + \ldots + \beta^{2^{i_t}} &= 0 \\ \beta^{2^{i_1+1}} + \ldots + \beta^{2^{i_t+1}} &= 0 \\ &\ldots \\ \beta^{2^{i_1+r-1}} + \ldots + \beta^{2^{i_t+r-1}} &= 0 \end{aligned}$$

Summing up left and right hand side we get,

$$1 + 1 + \ldots + 1 = 0$$

Here we use $\beta^{2^{i_j}} + \beta^{2^{i_j+1}} \ldots + \beta^{2^{i_j+r-1}} = 1$

But there are t (odd) 1's on the left hand side and so the sum is 1. This gives us the required contradiction. □.

Note that there exist primes n, such that $\phi(n) > 2\, sord_n(2)$. In fact this will hold for any prime of the form $2^i \pm 1$, $i > 3$. The first ten primes for which conditions 2 and 3 hold are given in Chapter 5, Table 1.

LEMMA 6.4.2. *If for some even length length l, odd dimension k, a σ-automaton is non-invertible, then it is non-invertible for all odd dimensions $\geq k$.*

Proof : It is non-invertible for k implies that there exists roots $\alpha_1, \ldots \alpha_k$ of π_{l+1}, such that $\alpha_1 + \ldots + \alpha_k = 0$.

But then for any odd dimension d greater than k, we know that $d-k$ is even and we can form the sum $\alpha_1 + \ldots + \alpha_k + \alpha_1 + \ldots + \alpha_1 = 0$ where α_1 is repeated $d-k$ times. But this shows that the σ-automaton on d dimensions is also non-invertible. □.

THEOREM 6.4.4. *If for some even length l, $l+1$ has two factors congruent to* 1 mod 4 *and* 3 mod 4, *then there exists an odd integer k, such that the σ-automaton on k dimensions is non-invertible.*

Proof : Let $l+1$ have at least two factors p_1 and p_2, with

$$p_1 \equiv 1 \mod 4 \text{ and } p_2 \equiv 3 \mod 4$$

Then corresponding to these factors p_1 and p_2, $\pi_{l+1}(x)$ has two factors $\pi_{p_1}(x)$ and $\pi_{p_2}(x)$ with $\pi_{p_1} = \rho_1^2$ and $\pi_{p_2} = \rho_2^2$ for some polynomials $\rho_1(x)$ and $\rho_2(x)$. Since $p_1 \equiv 1 \mod 4$, degree of $\rho_1(x)$ is even (say $2r_1$) and since $p_2 \equiv 3 \mod 4$, degree of $\rho_2(x)$ is odd (say $2r_2+1$). Also since p_1 and p_2 are both odd, by Remark 6.4.1., we get

$$\alpha_1 + \ldots + \alpha_{2r_1} = 1$$

$$\beta_1 + \ldots + \beta_{2r_2+1} = 1$$

where α_i's are roots of ρ_1 and β_j's are roots of ρ_2.

Let $k = 2r_1 + 2r_2 + 1$. Then,

$$\alpha_1 + \ldots + \alpha_{2r_1} + \beta_1 + \ldots + \beta_{2r_2+1} = 0$$

and hence by Corollary 6.4.1., the σ-automaton on k dimensions is non-invertible. □

THEOREM 6.4.5. *If for some even length l, $l+1$ has two relatively prime factors both congruent to* 3 mod 4, *then there exists an odd integer k, such that the σ-automaton on k dimensions is non-invertible.*

Proof : Let $p_1 \,|\, l+1$ and $p_2 \,|\, l+1$, with p_1 and p_2 both congruent to 3 mod 4.

Let $n = p_1 p_2 \equiv 1 \bmod 4$. Since $\gcd(p_1, p_2) = 1$, we can write,

$\pi_n(x) = \pi_{p_1}(x)\, \pi_{p_2}(x)\, (p(x))^2$ for some polynomial $p(x)$.

Now degrees of both $\sqrt{\pi_{p_1}}$ & $\sqrt{\pi_{p_2}}$ are odd, so $p(x)$ must be an even degree polynomial. (Since $\sqrt{\pi_n}$ is of even degree)

Let the degrees of $\sqrt{\pi_{p_1}}$, $\sqrt{\pi_{p_2}}$ & $p(x)$ be r_1, r_2, r_3 respectively with $r_1 = \frac{p_1-1}{2}$, $r_2 = \frac{p_2-1}{2}$ and $r_1 + r_2 + r_3 = \frac{n-1}{2}$. By Lemma 6.4.1. $\sqrt{\pi_{p_1}}$, $\sqrt{\pi_{p_2}}$ and $\sqrt{\pi_n}$ contain the terms x^{r_1-1}, x^{r_2-1}, $x^{\frac{n-3}{2}}$ respectively. But this implies that $p(x)$ has the term x^{r_3-1}.

Let $\alpha_1, \ldots, \alpha_{r_1}$ be the roots of $\sqrt{\pi_{p_1}}$ and $\beta_1, \ldots, \beta_{r_3}$ be the roots of $p(x)$.

Then for $k = r_1 + r_3$,

$$\sum \alpha + \sum \beta = 1 + 1 = 0$$

and k is odd. □

REMARK 6.4.2. *By Lemma 6.4.2. it follows that such σ-automata are also non-invertible for all odd dimension $\geq k$ and hence for all dimension $\geq k$ (since if k is even it is in any case non-invertible). This however does not preclude the fact that it may be invertible for some lower odd dimension. Thus in these cases, invertibility has to be checked only for finitely many dimensions. From the proofs of the theorems it follows that k can be chosen $< l$.*

This method of obtaining invertibility does not work if all prime factors of $l+1$ are congruent to 1 mod 4.

EXAMPLE 6.4.1. *We present examples of the cases considered in this section.*

1. *l odd, k even. $l = 3$ $k = 4$, σ-automata non-invertible.*
2. *l odd, k odd. $l = 3$. $k = 5$, σ-automata non-invertible.*
3. *l even, k even. $l = 4$ $k = 8$, σ-automata non-invertible.*
4. *l even, k odd.*
   - (a) *$l = 10$, $l+1 = 11$, $\phi(11) = 10 = 2 \times 5 = 2\,sord_{11}(2)$. Hence σ-automata invertible for all odd dimensions.*
   - (b) *$l = 34$, $l+1 = 35 = 5 \times 7$. $5 \equiv 1 \bmod 4$ and $7 \equiv 3 \bmod 4$. Then for $k = 2+3 = 5$ dimensions σ-automata is non-invertible.*
   - (c) *$l = 76$, $l+1 = 77 = 7 \times 11$. $7 \equiv 3 \bmod 4$ and $11 \equiv 3 \bmod 4$. Then for $k = 3+30 = 33$ dimensions σ-automata is non-invertible.*

6.4.2 Invertibility of σ^+-Automata

In this subsection we will consider σ^+-automaton on a k-dimensional *symmetric* orthogonal grid $G_k(l)$. The analysis is similar to that in the case of σ-automaton. We start with the following

THEOREM 6.4.6. *The global transformation of a σ^+-automaton on $G(l_1, \dots, l_k)$, is given by a generalised S^+-matrix written as*

$$T^+ = T + I_{l_1} \otimes I_{l_2} \otimes \dots \otimes I_{l_k},$$

where T is the matrix representing the global transformation of a σ-automaton on $G(l_1, \dots, l_k)$.

For the special case of symmetric σ^+-automaton, this reduces to,

$$\begin{aligned} T^{(k)+} &= T^{(k)} + I^{(k)} \\ &= I \otimes T^{(k-1)} + S^+ \otimes I^{(k-1)} \end{aligned} \qquad \text{(V)}$$

From this we get a result similar to that in Theorem 6.4.2.. However, in this case the recurrence itself is difficult to analyze because of the asymmetry in the expression.

THEOREM 6.4.7. *The σ^+-automaton on $G_k(l)$ is non-invertible iff*

$$\alpha_1 + \dots + \alpha_k = 1$$

for some choice of $\alpha_1, \dots \alpha_k$, where α_i's are roots of π_{l+1} over its splitting field.

Proof : The proof is similar to that of Theorem 6.4.2.. The right hand side is 1 because of S^+ in equation (V). Since the characteristic polynomial for S^+ is $\pi_{l+1}(x+1)$, its roots are of the form $\alpha + 1$ where α is any root of $\pi_{l+1}(x)$. □

REMARK 6.4.3. *Analogous to Lemma 6.4.2. we can deduce for the σ^+-automaton that if it is non-invertible for k dimensions, it is also non-invertible for $k + 2i$ dimensions. ($i = 1, 2, \dots$)*

LEMMA 6.4.3. *If $l+1$ has a divisor congruent to* 3 mod 4,, *then there exists an odd $k \leq \frac{l}{2}$ such that the σ^+-automaton on k dimensions is non-invertible.*

Proof : Let $a \mid l+1$ and $a \equiv 3 \bmod 4$. Then $\pi_a \mid \pi_{l+1}$ and so the roots of π_a are the roots of π_{l+1}. Also $\pi_a = p^2(x)$, where $p(x)$ has odd degree $d = \frac{a-1}{2}$ and sum of roots of $p(x)$ is 1 (by Remark 6.4.1.). Then the σ^+-automaton on d dimensions is non-invertible. □

Arguing similarly, we have

LEMMA 6.4.4. *If $l+1$ has a divisor congruent to* 1 mod 4, *then there exists an even $k \leq \frac{l}{2}$ such that σ^+-automaton on k dimensions is non-invertible.*

The above two lemmata and the remark yield

LEMMA 6.4.5. *If $l+1$ has two divisors a and b with $a \equiv 1 \bmod 4$ and $b \equiv 3 \bmod 4$, then there exists an integer $k \leq \frac{l}{2}$ such that σ^+-automaton on i dimensions is non-invertible for all $i \geq k$.*

REMARK 6.4.4. *Thus in these cases invertibility has to be checked only for finitely many dimensions.*

LEMMA 6.4.6. *If l is of the form $2^n - 1$ for some n, then the σ^+-automaton is invertible for all dimensions.*

Proof : In this case, $\pi_{l+1} = x^{2^n-1}$ and hence the only root of π_{l+1} is 0, so it is impossible to have a subset sum of roots to be 1. □

The following is an analogue of Theorem 6.4.3..

THEOREM 6.4.8. *Let $l+1$ be a prime such that $\pi_{l+1}(x)$ has only one irreducible factor (i.e, $\phi(l+1) = 2\, sord_{l+1}(2)$).*

If $l+1 \equiv 3 \bmod 4$, then σ^+-automaton is invertible for all even dimensions.

If $l+1 \equiv 1 \bmod 4$, then σ^+-automaton is invertible for all odd dimensions.

Proof : Let $\pi_{l+1} = \rho^2$ with ρ irreducible and of degree $r = \frac{l}{2}$.

Then there are r distinct roots $\alpha_1, \ldots, \alpha_r$ of π_{l+1}, and by Remark 6.4.1.,

$$\alpha_1 + \ldots + \alpha_r = 1$$

Since ρ is irreducible (by Theorem 5.3.2.) its roots are of the form $\beta, \beta^2, \ldots, \beta^{2^{r-1}}$ and so

$$\beta + \beta^2 + \ldots + \beta^{2^{r-1}} = 1$$

Let if possible for some $k < r$ such that $k \bmod 2 \neq r \bmod 2$,

$$\alpha_1 + \ldots + \alpha_k = 1$$

Then for some $i_1, \ldots, i_k$

$$\beta^{2^{i_1}} + \ldots + \beta^{2^{i_k}} = 1 \qquad (VI)$$

and by an argument similar to the one in the proof of Theorem 6.4.3., (VI) will be satisfied by all roots of ρ and hence we will get the r equations.

$$\beta^{2^{i_1}} + \ldots + \beta^{2^{i_k}} = 1$$
$$\beta^{2^{i_1+1}} + \ldots + \beta^{2^{i_k+1}} = 1$$

$$\cdots$$

$$\beta^{2^{i_1}+r-1} + \ldots + \beta^{2^{i_k}+r-1} = 1$$

Summing up we get $k \mod 2 = r \mod 2$ which is a contradiction. Hence for dimension k such that, $k \mod 2 \neq r \mod 2$, it is not possible to obtain $\alpha_1, \ldots, \alpha_k$ (which are roots of ρ and hence of π_{l+1}) such that $\alpha_1 + \ldots + \alpha_k = 1$. But this means that the σ^+-automaton on $G_k(l)$ is invertible. Now $k \mod 2 \neq r \mod 2$ means that if $l+1 \equiv 3 \mod 4$, then r is odd and k must be even. And if $l+1 \equiv 1 \mod 4$, then r is even and k must be odd. Hence the result. □

Example 6.4.2. *We present examples of the conditions covered in this section.*

1. *If* $l = 6$, $l+1 = 7 \equiv 3 \mod 4$.
   Then for $k = 3 + 2i$ *dimensions* σ^+*-automata is non invertible.*

2. *If* $l = 8$, $l+1 = 9 \equiv 1 \mod 4$.
   Then for $k = 4 + 2i$ *dimensions* σ^+*-automata is non invertible.*

3. *If* $l = 134$, $l+1 = 135 = 9 \times 15$ *with* $9 \equiv 1 \mod 4$ $15 \equiv 3 \mod 4$.
   Then for $k \geq 7$ *dimensions* σ^+*-automata is non invertible.*

4. $l = 7 = 2^3 - 1$. $\pi_8 = x^7$ *and* σ^+*-automata is invertible for all dimensions* k.

5. (a) $l = 6$, $l+1 = 7 \equiv 3 \mod 4$, $\phi(l+1) = 2\,sord_{l+1}(2)$. *So* σ^+*-automata is invertible for all even dimensions.*

   (b) $l = 4$, $l+1 = 5 \equiv 1 \mod 4$, $\phi(l+1) = 2\,sord_{l+1}(2)$. *So* σ^+*-automata is invertible for all odd dimensions.*

Some more results on σ^+-automata are obtained in the next subsection.

6.4.3 Characteristic Polynomial of Generalised S-Matrix

We now derive an expression for the characteristic polynomial of a generalised S-matrix in terms of resultant of two polynomials. First we need the following which can easily be proved using the identity 3.6(v) of [120] for the resultant of two polynomials (see also Appendix C).

Lemma 6.4.7. *If* $P(x)$ *and* $Q(x)$ *are two nonconstant polynomials with coefficients in a field* K *and with roots* $\alpha_1, \ldots, \alpha_m$ *and* $\beta_1, \ldots, \beta_n$ *respectively, then the roots of the polynomial*

$$R(y) = Res_x(P(x+y), Q(-x))$$

are the elements $\alpha_i + \beta_j$, $1 \leq i \leq m$, $1 \leq j \leq n$.

THEOREM 6.4.9. *For a fixed length l, define a sequence of polynomials by the following recurrence,*

$$\begin{aligned} Q_1(x) &= \pi_{l+1}(x) \\ Q_k(x) &= Res_y(Q_1(x+y), Q_{k-1}(y)), \quad k > 1 \end{aligned}$$

Then $Q_k(x)$ is the characteristic polynomial for the transition matrix $T^{(k)}$ of the σ-automaton on $G_k(l)$.

Proof : By induction on k we prove that α is a root of $Q_k(x)$ iff α is of the form $\alpha_1 + \ldots + \alpha_k$, where α_i's are roots of $\pi_{l+1}(x)$. Then using Theorem 6.4.2. we are done.

For $k = 1$ the result is easy.

So assume the result to be true for $k-1$.

Then $Q_k(x) = Res_y(Q_1(x+y), Q_{k-1}(x))$, and α is a root of $Q_k(x)$ iff it is of the form $\beta + \alpha_k$, where β is any root of $Q_{k-1}(x)$ and α_k is any root of $Q_1(x)$. But, by induction hypothesis, β is of the form $\alpha_1 + \ldots + \alpha_{k-1}$. Hence the result follows. □

COROLLARY 6.4.2. *$Q_k(1+x)$ is the characteristic polynomial for $T^{(k)} + I^{(k)}$, the matrix for σ^+-automaton on $G_k(l)$.*

We will write $T^{(k)+}$ for $T^{(k)} + I^{(k)}$ and $Q_k^+(x)$ for $Q_k(1+x)$. The characteristic polynomial can be used to settle a few more cases for the non-invertibility of σ^+-automata.

THEOREM 6.4.10. *If $l \equiv 2 \bmod 3$ and σ-automaton on $(k-1)$ dimensions is non-invertible then so is σ^+-automaton on k dimensions.*

Proof : Since $l \equiv 2 \bmod 3$, $l+1 \equiv 0 \bmod 3$ and so $3 \mid l+1$. Hence noting that $\pi_3(x) = 1 + x^2$ we get, $(x+1)^2 \mid \pi_{l+1}(x)$ (by Lemma 3.2.1.) and so we can write $\pi_{l+1}(x) = (x+1)^2 \pi'_{l+1}(x)$. So, noting that all operations are over $GF(2)$, we have,

$$\begin{aligned} Q_k(y) &= Res_x((x+y+1)^2 \pi'_{l+1}(x+y), Q_{k-1}(x)) \\ &= Q_{k-1}^2(1+y)\, Res_x(\pi'_{l+1}(x+y), Q_{k-1}(x)) \text{ (by Lemma C.0.5.(a)))} \end{aligned}$$

But this shows $Q_{k-1}(y) \mid Q_k(1+y)$. Thus if $T^{(k-1)}$ is non-invertible then $y \mid Q_{k-1}(y)$. Hence $y \mid Q_k(1+y)$ and so $T^{(k)+}$ is non-invertible. □

COROLLARY 6.4.3. *If $l \equiv 2 \bmod 3$, then σ^+-automaton is non-invertible for all odd dimensions k.*

Proof : Follows from the above theorem and the fact that σ-automaton is non-invertible for $k-1$ (since $k-1$ is even). □

THEOREM 6.4.11. *If $l \equiv 1 \bmod 2$ and if σ^+-automaton on k dimensions is non-invertible then so is σ^+-automaton on $k+1$ dimensions.*

Proof : Since $l \equiv 1 \mod 2$ we have $\pi_{l+1}(x) = x\,\pi'_{l+1}(x)$. Hence, since all operations are over $GF(2)$,

$$\begin{aligned} Q_{k+1}(y) &= Res_x((x+y)\,\pi'_{l+1}(x+y), Q_k(x)) \\ &= Q_k(y)\,Res_x(\pi'_{l+1}(x+y), Q_k(x)) \text{ (by Lemma C.0.5.)} \end{aligned}$$

But then $Q_k(1+y) \,|\, Q_{k+1}(1+y)$. Hence if $T^{(k)+}$ is non invertible then so is $T^{(k+1)+}$. □

COROLLARY 6.4.4. *If $l+1 \equiv 0 \mod 6$, then σ^+-automaton is non-invertible for all dimensions.*

Proof : Since $l+1 \equiv 0 \mod 6$, it follows that l is odd and hence by the above theorem it is sufficient to prove that $x \,|\, \pi_{l+1}(1+x)$. But this happens iff $(1+x) \,|\, \pi_{l+1}(x)$. Again, since $l+1 \equiv 0 \mod 6$ we have $3 \,|\, l+1$, so $(1+x)^2 \,|\, \pi_{l+1}(x)$. This proves the result. □

EXAMPLE 6.4.3. *We provide examples of the results settled in this section.*

1. $l = 83 \equiv 2 \mod 3$, *and so σ^+-automata is non-invertible for all odd dimensions.*

2. $l = 11$, $l+1 \equiv 0 \mod 6$, *and so σ^+-automata is non-invertible for all dimensions.*

6.5 Generalisations

6.5.1 Asymmetric Grids

In this subsection, we extend the results of previous sections to cover σ-automata on *null boundary asymmetric grids.* Most of the proofs are plain generalisations and will be omitted.

THEOREM 6.5.1. *A σ-automaton (resp. σ^+-automaton) on $G(l_1, \ldots, l_k)$ is non-invertible iff*

$$\alpha_1 + \ldots + \alpha_k = 0 \text{ (resp. 1)}$$

for some choice of α_i's, where α_i is any root of π_{l_i+1} over a field in which all π_{l_i+1}'s split.

In [15], this result is obtained for two dimensions by showing that σ-automaton is invertible iff $\pi_{l_1+1}(x)$ and $\pi_{l_2+1}(x)$ are relatively prime and for the σ^+-automaton $\pi_{l_1+1}(x)$ and $\pi_{l_2+1}(1+x)$ must be relatively prime. It turns out that $\pi_{l_1+1}(x)$ and $\pi_{l_2+1}(x)$ are relatively prime iff l_1+1 and l_2+1 are so (see also [166, 168]). For the σ^+-automaton, such complete result could not be obtained. For certain special cases, sufficiency conditions for invertibility based on number theoretic properties of l_1 and l_2 can be derived. But a general characterisation of this nature seems to be difficult. The above theorem indicates the cause for this difficulty. To obtain a characterisation of invertibility in terms of number theoretic properties we have to characterize in terms of number theoretic properties when a subset sum of roots will lie in the base field. Since the roots in general lie in an extension field, answering this question in general will be difficult.

LEMMA 6.5.1.

(a) If $l_1, \ldots, l_k$ are all odd, then σ-automaton of k dimensions is non-invertible.

(b) If for even k , $\gcd(l_1 + 1, \ldots, l_k + 1) > 1$, then σ-automaton on such a grid is non-invertible.

(c) If the l_i's are of the form $2^{n_i} - 1$ for some n_is, then the σ^+-automaton on such a grid is invertible.

6.5.2 Folded and Mixed Grids

Here we will allow some or all dimensions to have periodic boundary condition. The following is similar to Theorem 6.3.1..

THEOREM 6.5.2. *Consider a k-dimensional grid $G(l_1, \ldots, l_k)$ with periodic boundary condition in some r $(0 \le r \le k)$ dimensions. Then the transition matrix of the σ-automaton on this grid is given by,*

$$T = I_{l_1} \otimes I_{l_2} \otimes \ldots \otimes A_{l_k} + I_{l_1} \otimes I_{l_2} \otimes \ldots \otimes A_{l_{k-1}} \otimes I_{l_k} + \ldots + A_{l_1} \otimes I_{l_2} \otimes \ldots \otimes I_{l_k}$$

where,

$$\begin{aligned} A_{l_i} &= S_{l_i}, \quad \textit{if there is null boundary condition in the } i^{th} \textit{ dimension;} \\ &= C_{l_i}, \quad \textit{if there is periodic boundary condition in } i^{th} \textit{ dimension.} \end{aligned}$$

The matrix for the σ^+-automaton is given by,

$$T^{(+)} := T + I_{l_1} \otimes I_{l_2} \otimes \ldots \otimes I_{l_k}$$

THEOREM 6.5.3. *Consider a mixed grid as in the above theorem. The σ-automaton (resp. σ^+-automaton) on such a grid is non-invertible iff for some $\alpha_1, \ldots, \alpha_k$*

$$\alpha_1 + \ldots + \alpha_k = 0 \ (\textit{resp. } 1)$$

where α_i is any root of $p_i(x)$, the characteristic polynomial for A_{l_i} and so,

$$\begin{aligned} p_i(x) &= \pi_{l_i+1}(x), \quad \textit{if the } i^{th} \textit{ dimension has null boundary condition;} \\ &= x\pi_{l_i}(x), \quad \textit{if the } i^{th} \textit{ dimension has periodic boundary condition.} \end{aligned}$$

Note that in the above theorem, we can replace the characteristic polynomial for A_{l_i} by the minimal polynomial for A_{l_i}. This is because the minimal and characteristic polynomials have the same set of distinct roots. Thus $p_i(x)$ can be written as,

$$\begin{aligned} p_i(x) &= \pi_{l_i+1}(x), && \text{if the } i^{th} \text{ dimension has null boundary condition;} \\ &= x\,\pi_{\frac{l_i}{2}}(x), && \text{if the } i^{th} \text{ dimension has periodic boundary condition and } l_i \text{ is even;} \\ &= x\sqrt{\pi_{l_i}}(x), && \text{if the } i^{th} \text{ dimension has periodic boundary condition and } l_i \text{ is odd.} \end{aligned}$$

LEMMA 6.5.2. *In the underlying grid, if a dimension has length $2^{r_1}-1$ with null boundary condition or length 2^{r_2} with periodic boundary condition, then we can ignore the effect of this dimension on the invertibility of σ or σ^+-automaton.*

LEMMA 6.5.3. *For a mixed asymmetric grid, if all dimensions with null boundary condition have lengths of the form $2^{r_1}-1$ and all dimensions with periodic boundary condition have lengths of the form 2^{r_2}, then σ-automaton on such a grid is non-invertible and σ^+-automaton is invertible.*

Similar to Theorem 6.4.9., one gets

THEOREM 6.5.4. *Consider a k dimensional mixed grid on $G(l_1,\ldots,l_k)$. Then the characteristic polynomial $Q_k(x)$ for the transition matrix of σ-automaton on such a grid is given by,*

$$\begin{aligned} Q_1(x) &= p_1(x) \\ Q_i(x) &= Res_y(p_i(x+y), Q_{i-1}(y)), \quad 1 < i \leq k, \end{aligned}$$

where $p_i(x)$ is as in Theorem 6.5.3..

6.5.3 Other Neighbourhoods

We generalise the concept of nearest neighbourhood to higher dimensions. For the two dimensional case there are two kinds of nearest neighbourhood condition - the orthogonal neighbourhood and the diagonal neighbourhood. Our generalisation is based upon the following observation. The orthogonal neighbourhood correspond to taking one step in one dimension. The diagonal neighbourhood correspond to taking one step each in two dimensions. Generalizing, for a cell in a k-dimensional grid, we let its r-dimensional set of neighbours be the cells which are reachable by taking one step each in exactly r-dimensions. Since in any dimension we do not allow more than one step the notion of nearest neighbourhood is preserved. Any neighbour of a cell J can also be visualized to be lying on some hyperplane unit distance away from J. We formally express this idea in the following

DEFINITION 6.5.1. *For a cell $(i_1,\ldots,i_k)$ in a k-dimensional grid, the set of r-dimensional (r-D) nearest neighbours is given by,*

$$N_r(i_1,\ldots,i_k) = \{(i_1,\ldots,i_{j_1}\pm 1,\ldots,i_{j_r}\pm 1,\ldots,i_k) : 1 \leq j_1 < \ldots < j_r \leq k\}$$

where $i_j \pm 1$ is taken modulo l_j if the j^{th} dimension has a periodic boundary condition. If the j^{th} dimension has a null boundary condition, then the values -1 and l_j are ignored for the j^{th} dimension.

It is easy to see that the definition exactly correspond to the idea described above. Also it is clear that $|N_r(i_1,\ldots,i_k)| \leq 2^r\binom{k}{r}$ where equality holds for all cells iff all $l_i > 2$ and all dimensions have periodic boundary condition. Such neighbourhoods for multidimensional CA have not been considered before. Martin et al [115] introduced Type I and Type II neighbourhoods for multidimensional CA. Type I neighbourhood correspond to our 1-D neighbourhood. Type II neighbours of a cell $J = (i_1,\ldots,i_k)$ are given by the set $\{J\} \bigcup \cup_{1\leq r\leq k} N_r(J)$. Thus our definition captures a finer sense of multidimensional neighbourhood.

We now obtain a characterisation of the global rule of an r-D neighbourhood σ-automaton in terms of Kronecker product.

THEOREM 6.5.5. *Consider an r-d neighbourhood σ-automaton on a k dimensional mixed grid $G(l_1,\ldots,l_k)$. Then the global rule is given by the following matrix.*

$$T_r^{(k)} = \sum_{1\leq j_1<\ldots<j_r<\ldots\leq k} R_1 \otimes \ldots \otimes R_k$$

where,

$$\begin{aligned} R_i &= I_{l_i} && \textit{if } i \notin \{j_1,\ldots j_r\} \\ &= S_{l_i} && \textit{if } i \in \{j_1,\ldots j_r\} \textit{ and the } i^{th} \textit{ dimension has null boundary condition.} \\ &= C_{l_i} && \textit{if } i \in \{j_1,\ldots j_r\} \textit{ and the } i^{th} \textit{ dimension has periodic boundary condition.} \end{aligned}$$

For the σ^+-automaton the corresponding global rule is $T_r^{(k)+} = T_r^{(k)} + I_{l_1\ldots l_k}$.

Proof : Let $T^{(k)}_{j_1,\ldots,j_r}$ be the matrix which correspond to the local rule which consider neighbours only in the dimensions $j_1,\ldots,j_r$. Then by linearity it follows that,

$$T_r^{(k)} = \sum_{1\leq j_1<\ldots<j_r\leq k} T^{(k)}_{j_1,\ldots,j_r}$$

Using Lemma 6.3.1., we can construct a proof similar to that of Theorem 6.3.1. to show that

$$T^{(k)}_{j_1,\ldots,j_r} = R_1 \otimes \ldots \otimes R_k$$

where R_i is as defined in the theorem. Hence the result follows. □

Analogous to Theorem 6.4.2., we have

THEOREM 6.5.6. *Consider an r-D neighbourhood σ-automaton on a k-dimensional mixed grid $G(l_1,\ldots,l_k)$. Then α is a root of the characteristic polynomial of the transition matrix of the σ-automaton iff α is of the form*

$$\sum_{1\leq j_1<\ldots<j_r<\ldots\leq k} \alpha_{j_1}\ldots\alpha_{j_r}, \quad \textit{for some choice of } \alpha_1,\ldots,\alpha_k.$$

. *Here α_i is a root of $p_i(x)$, where $p_i(x)$ is as in Theorem 6.5.3..*

Proof : Let ϵ be an arbitrary scalar and consider the product

$(I_{l_1} + \epsilon\, A_{l_1}) \otimes (I_{l_2} + \epsilon\, A_{l_2}) \otimes \ldots \otimes (I_{l_k} + \epsilon\, A_{l_k})$

$= I_{l_1} \otimes \ldots \otimes I_{l_k} + \epsilon(I_{l_1} \otimes I_{l_2} \otimes \ldots \otimes A_{l_k} + \ldots + A_{l_1} \otimes I_{l_2} \otimes \ldots \otimes I_{l_k}) + \ldots + \epsilon^k A_{l_1} \otimes \ldots \otimes A_{l_k}$

where A_{l_i} is S_{l_i} or C_{l_i} according as the i^{th} dimension has null or periodic boundary condition. The characteristic roots of the left hand side are,

$$(1 + \epsilon\,\alpha_1)(1 + \epsilon\,\alpha_2) \ldots (1 + \epsilon\,\alpha_k)$$

$$= 1 + \epsilon(\alpha_1 + \ldots + \alpha_k) + \ldots + \epsilon^k \alpha_1 \ldots \alpha_k$$

where α_i is a root of $p_i(x)$.

Let $\underset{\sim}{u}$ be an eigen vector corresponding to a root. Then,

$(\underset{\sim}{u} - (I_{l_1} \otimes \ldots \otimes I_{l_k})\underset{\sim}{u}) + \epsilon((\alpha_1 + \ldots + \alpha_k)\underset{\sim}{u} - (I_{l_1} \otimes I_{l_2} \otimes \ldots \otimes A_{l_k} + \ldots + A_{l_1} \otimes I_{l_2} \otimes \ldots \otimes I_{l_k})\underset{\sim}{u}) + \ldots + \epsilon^k((\alpha_1 \ldots \alpha_k)\underset{\sim}{u} - (A_{l_1} \otimes \ldots \otimes A_{l_k})\underset{\sim}{u}) = 0$

Since ϵ is arbitrary, all coefficients of ϵ^i are 0 $(0 \leq i \leq k)$. From this the result follows. □

COROLLARY 6.5.1. *An r-D neighbourhood σ(resp. σ^+)-automaton on a k-dimensional mixed grid is non-invertible iff for some choice of $\alpha_1, \ldots, \alpha_k$ we have,*

$$\sum_{1 \leq j_1 < \ldots < j_r < \ldots \leq k} \alpha_{j_1} \ldots \alpha_{j_r} = 0(resp.\ 1)$$

where α_i's are as described in the above theorem.

In particular, we have

PROPOSITION 6.5.1. *For $G_l(l)$ with null boundary condition, an r-D neighbourhood σ(resp. σ^+)-automaton is non-invertible if the coefficient of x^{l-r} in $\pi_{l+1}(x)$ is 0(resp. 1).*

Proof : Here we have to consider only $\pi_{l+1}(x) = x^l + a_{l-1}x^{l-1} + \ldots + a_0$ with $\sum \alpha_1 = a_{l-1}$, $\sum \alpha_1\alpha_2 = a_{l-2}$, ... , $\alpha_1 \ldots \alpha_l = a_0$, where α_i's are roots of $\pi_{l+1}(x)$. From this using Corollary 6.5.1., the result follows. □

REMARK 6.5.1. *If in the above proposition all dimensions have periodic boundary condition, then we will have to consider the characteristic polynomial for C_l instead of $\pi_{l+1}(x)$.*

6.6 Conclusion

In this chapter we have developed necessary and sufficient conditions for the invertibility of σ (σ^+)-automata on multidimensional orthogonal grids with different boundary conditions. These conditions have been obtained in terms of the roots of π-polynomials. Also we have tried to relate this to the number theoretic properties of the number of dimensions and lengths of the dimensions.

For symmetric (all dimensions having equal lengths l) σ-automata, we have to consider only one π-polynomial (π_{l+1}). In this case, the invertibility is directly related to a sum of subset of the roots of π_{l+1}. In trying to relate this to number theoretic properties, we are able to settle for k (dimension) even or l odd. The case for k odd, l even could not be settled completely. See Example 6.4.1. for the cases which could be settled. This is intimately related to the subset sum of roots of π_{l+1} and settling the invertibility question will also settle the question of when such an arbitrary subset sum will take values in the base field.

For symmetric σ^+-automata, we could obtain similar results, though a few cases remain unsettled. See Example 6.4.2. and Example 6.4.3. for the cases which could be settled. We were able to extend the subset sum necessary and sufficient condition to asymmetric as well as folded and mixed grids. Also for these grids we have been able to point out special cases where the invertibility can be settled in terms of number theoretic properties. Other cases which remain unsettled can form the subject of further research. Table 1 lists the cases which have been settled for symmetric grids.

The concept of non-orthogonal nearest neighbourhood have been generalised. Invertibility of σ-automata with such neighbourhood have been characterised in terms of the roots of π-polynomials. However, in this case number theoretic characterisation of invertibility remains open.

Chapter 7

A CA Based Private Key Cryptosystem

In the previous chapters we had concentrated more on theoretical aspects of CA. We have developed algebraic techniques to analyse several kinds of finite linear CA. However a large amount of work has been done in finding applications of CA, since CA based architectures are easy to implement in VLSI. In this chapter, we present a private key cryptosystem based upon composite (or products of) linear hybrid 90/150 CA. While providing adequate security, the system is easy to implement in VLSI.

7.1 Introduction

In the basic model of a private (or secret) key cryptosystem the sender and the receiver both share a secret key. The sender uses the key to encrypt the message he wants to send. The actual message is called the clear text or the message text, whereas the text obtained after encipherment is called the cipher text. The sender transmits the cipher text over a public channel. The receiver deciphers the message text from the cipher text using the common key. The attacker has access to the public channel and can obtain copies of the cipher text. His task is to recover the message text without knowing the secret key (see Figure 7.1). The strength of a cryptosystem lies in its invulnerability against different classes of attacks. Private key cryptosystems are widely used in defence data communications. The most popular private key encryption scheme is the Data Encryption Standard (DES) [135].

There are two classes of cryptosystems. In block cipher cryptography, the message text is divided into blocks of fixed length, and each block is encrypted separately. In stream cipher cryptography, the cipher text is obtained through bitwise exclusive OR of the message text with a cryptographically strong pseudo random bit sequence. Decryption is done by another bitwise exclusive OR operation. Here, we will present a block cipher scheme.

There are two notions about proving the strength or security of a cryptosystem. A

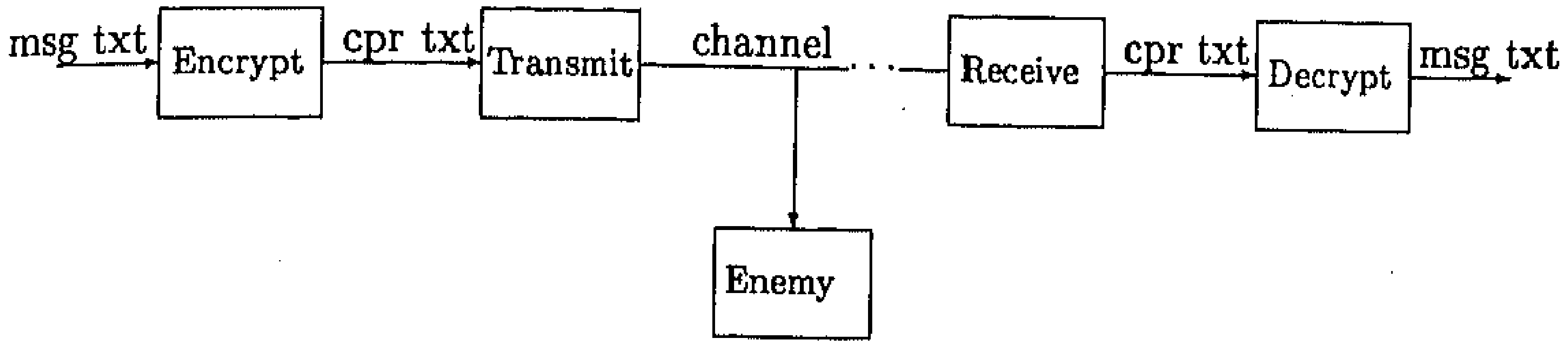

Figure 7.1: Basic model of a private key cryptosystem

system is said to possess *perfect secrecy* if the entropy of the message is equal to the entropy of the message given the cipher. This is an information theoretic notion of secrecy and was introduced by Shannon in [153]. The one-time pad is a system, where a string of true random bits is generated and used to encrypt the message only once. The one-time pad can be proved to possess perfect secrecy but is not suitable for practical implementation. A different notion of secrecy is to show that any cryptanalytic attack on the system is computationally infeasible. This is usually achieved by showing that cryptanalysing the system would amount to providing an algorithm for some computationally hard problem. The system is then secure modulo the intractibility assumption of the hard problem. One such system based upon the intractibility of the quadratic residuacity problem is presented in [20].

The use of CA in cryptography was suggested by Wolfram in [188]. Based on the study of random sequence generation by uniform periodic boundary CA with rule 30 [192], Wolfram proposed a stream cipher using the temporal sequence of a particular cell of the CA as a pseudorandom sequence. Damgard [51] later proposed a secure hash function based on Wolfram's generator. However, cryptanalytic attacks have been reported on Wolfram's original scheme [119] and Damgard's construction [50]. In [50] itself a CA based method is proposed for the secure design of computationally collision free hash function. A public key cryptosystem based on CA has also been proposed in [70]. A cellular automata based cryptosystem has been patented by Gutowitz [73, 74]. More, recently CA based block and stream ciphers have been reported [128]. In [128], additive CA rules are used to generate a set of fundamental transformations. These fundamental transformations are used as the block ciphering functions. The fundamental transformations are self inverses so that the deciphering process is the same as the enciphering process. The scheme suffers from the disadvantage of having a complicated key management procedure, which increases the hardware complexity, and also require synchronization between the sender and the receiver. The stream cipher in [128] uses two coupled CA to generate the pseudo-random sequence. Since the block cipher in [128] is based on affine transformation, it is open to cryptanalytic attacks based on algebraic techniques. This is highlighted in [19] where an attack on the stream cipher is also outlined.

Any CA based scheme will have the advantage of providing an efficient VLSI hardware for implementation of enciphering and deciphering functions. The use of CA in private key cryptosystem springs from the fact that the State Transition Diagram for a reversible n-cell CA is a permutation of the integers 0 to $2^n - 1$. By constructing special purpose CA it is possible to create desirable permutations which can be used as the secret key in data enciphering. We first introduce the concept of composite CA and provide complete characterisation of such composite CA in terms of cycle lengths. Given an irreducible polynomial p(x) over GF(2) it is possible to design a hybrid 90/150 CA such that the transition matrix T for the CA has p(x) as its characteristic polynomial (see [152]). We use this technique to design maximal length CA. (A CA is maximal length CA if $p(x)$ is a primitive polynomial). Cascading such maximal length CA leads to a composite CA which has one cycle of length one and all other cycles of equal length L. To encrypt, the message text is divided into blocks of bits. Each such block is loaded as initial state into a composite CA with cycle length L as described above. Here $L = 2^k - 1$, where for practical implementation k should be between 10 and 20. A random integer i is chosen in the range 1 to $L - 1$ and the CA is evolved for i steps. The output of the CA is used as input to a non-linear bijective transformation whose output is passed through a transformation which rearranges the order of the bits. Finally this bit string along with the integer i is sent to the receiver. The bits of the integer i is placed in-between the bits of the enciphered message. At the receiver's end a sequence of inverse transformations are applied to get back the original message. We prove that the cipher satisfies the perfect secrecy condition of [153].

7.2 Preliminaries

Here we will consider only one dimensional null boundary condition 90/150 CA, where each cell can assume values from GF(2). Figure 7.2 shows a 4-cell CA with *global rule vector* < 9015090150 > (see Section 7.2.1).

Suppose we have a n-cell linear CA. The global transition of the CA over one time period is given by,

$$y = Tx$$

where x and y are n-bit CA configuration vectors, and T is an $n \times n$ transition matrix which correspond to the CA evolution over one time step.

Consider the 4-cell CA of Figure 7.2. Then,

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

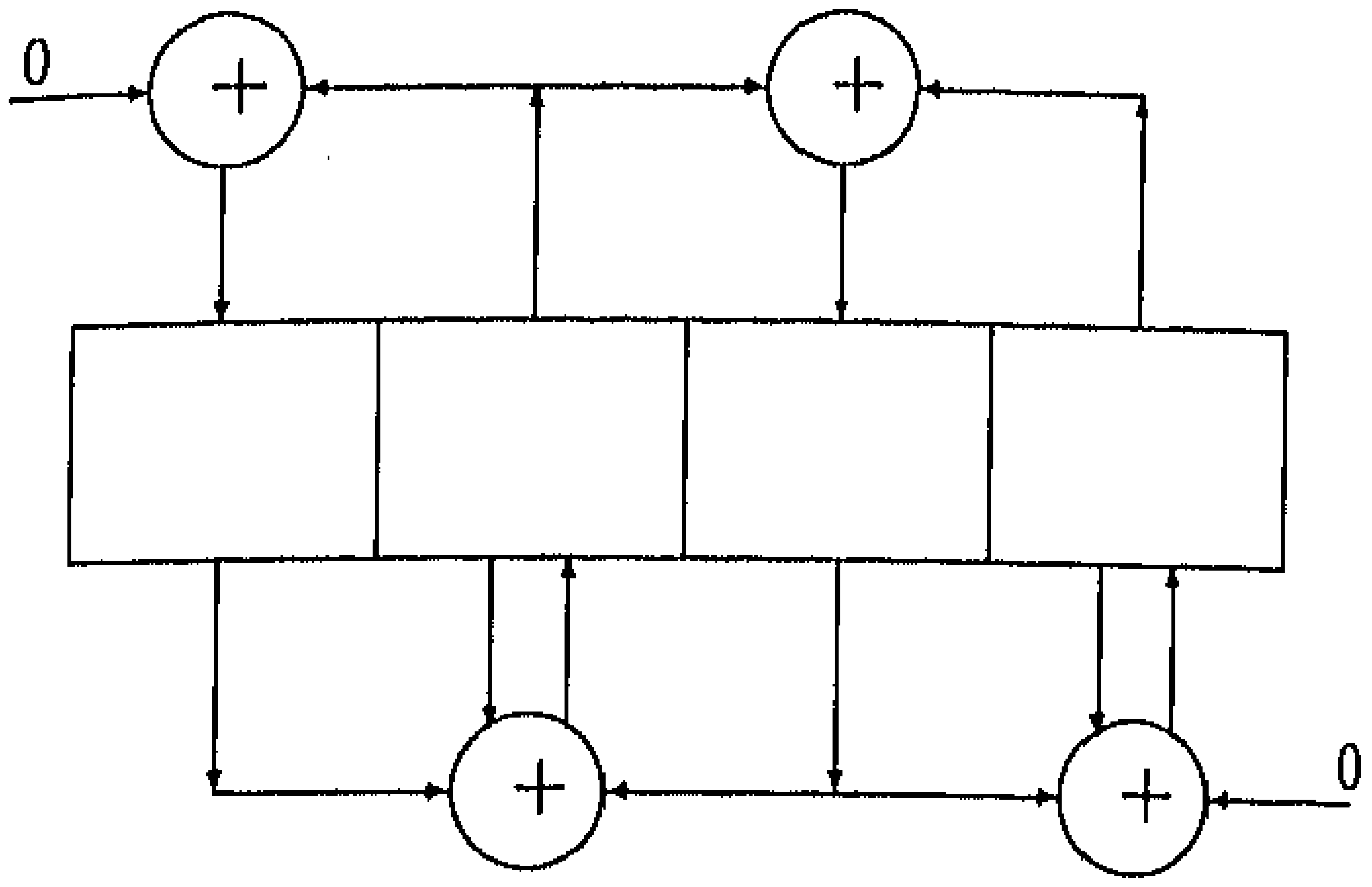

Figure 7.2: A $< 90, 150, 90, 150 >$ CA

Note that any such T will be a tridiagonal matrix (see Chapter 4). The STD for the CA in Figure 7.2 is given in Figure 7.3.

The following proposition relates reversible CA to the permutation it generates.

PROPOSITION 7.2.1. *The transition matrix T defines a map G from the set of n bit 0/1 vectors to itself. If the CA is reversible then T is invertible and G is a bijection which defines a permutation $\mathcal{P}$ of the set of integers $\{i : 0 \leq i \leq 2^n - 1\}$. The STD for such a CA represent the cycles of $\mathcal{P}$.*

Note that if the constant term in the characteristic polynomial $p(x)$ is one, then G is a bijection. This is so since the constant term is the determinant of the matrix.

We call a CA *maximal length*, if its STD consists of two cycles, with the null configuration on a cycle of length one and all other configurations on the other cycle. A CA is maximal length iff the characteristic polynomial of its global transition matrix is primitive. If the CA is maximal length then it is easy to see that the characteristic polynomial must be primitive. The other way is also well known (see for example [110]), but for the sake of completeness we provide a proof.

THEOREM 7.2.1. *If the characteristic polynomial $p(x)$ for the matrix T corresponding to an n-cell CA is a primitive polynomial, then the CA is an maximal length CA.*

Proof : The proof follows from the following observations.

1. For any non-null vector y, if $\phi(x)$ be the polynomial of least degree such that, $\phi(T)y = 0$, then for any polynomial $\psi(x)$ such that, $\psi(T)y = 0$, $\phi(x)|\psi(x)$
2. By Cayley Hamilton theorem $p(T) = 0$, hence, $p(T)y = 0$ and so $\phi(x)|p(x)$ which implies $p(x) = \phi(x)$.

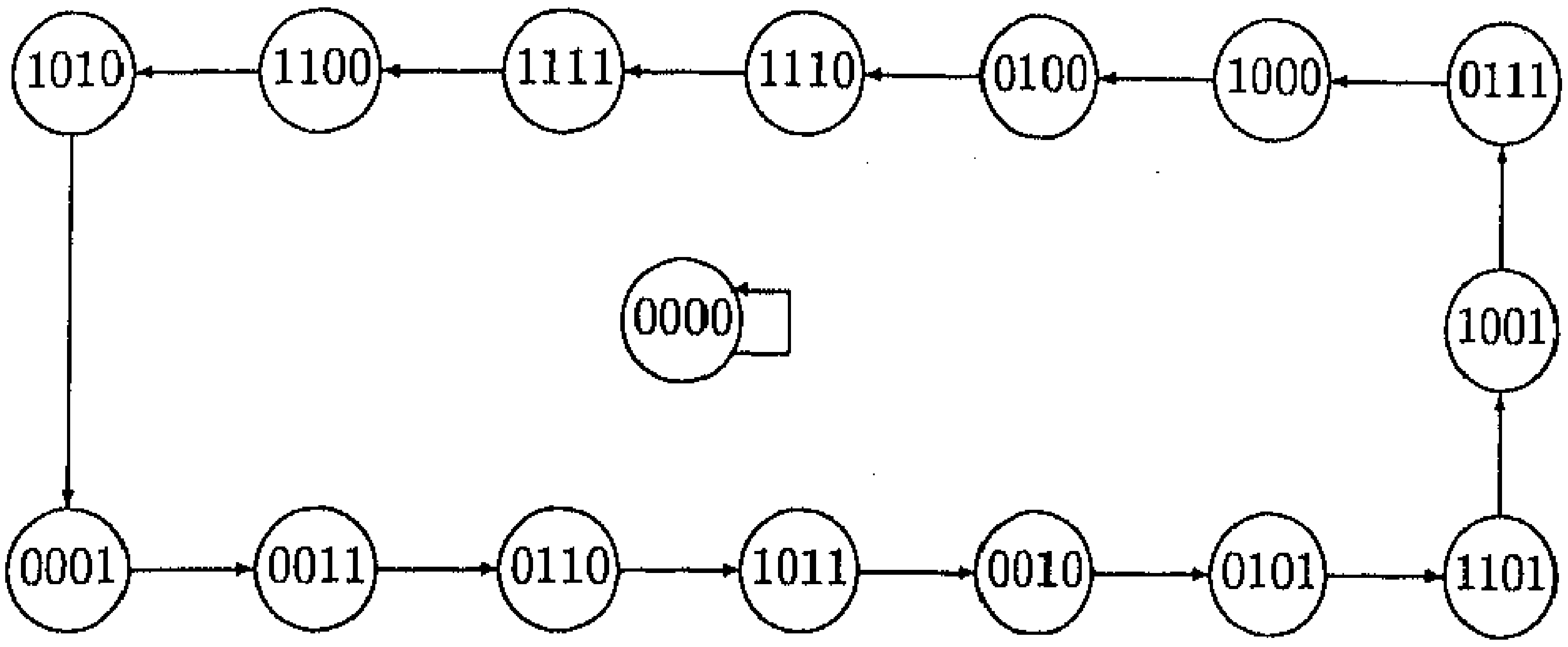

Figure 7.3: STD for the CA in Figure 2

3. If k be the least positive integer, such that, $T^k y = y$, then $p(x)|x^k - 1$.

4. By primitiveness of $p(x)$, it follows that $k = 2^n - 1$. □

There is also an algorithm to construct a 90/150 CA from an irreducible or primitive polynomial [152]. The following lemma gives an idea of the number of distinct maximal length CA over n cells.

LEMMA 7.2.1. *If $p_1(x)$ and $p_2(x)$ are two distinct n-degree primitive polynomials over $GF(2)$, then there are two distinct 90/150 CA corresponding to these two polynomials.*

For each 90/150 matrix constructed, reversal of the diagonal will lead to another 90/150 matrix with the same characteristic polynomial provided the diagonal is not a palindrome. If the diagonal is a palindrome then the characteristic polynomial can be factored (by Lemma 4.4.4.) and hence cannot be primitive.

Thus there are at least twice as many n-cell maximal length CA as there are n-degree primitive polynomials over GF(2). This lower bound will be important in assessing the complexity of cryptanalytic attack. The number of n-degree primitive polynomials over GF(2) is equal to $\frac{\phi(2^n-1)}{n}$ [110], where $\phi(n) = |\{m : 1 \leq m < n, \gcd(m,n) = 1\}|$ is the Euler's totient. Table 1 lists the number of primitive polynomials for some values of N. Tables for primitive polynomials are to be found in [138]. Moreover complete tables of maximal length 90/150 CA upto degree 32 are available in [155].

N	$\frac{\phi(2^N-1)}{N}$
10	60
11	176
12	144
13	630
14	756
15	1800
16	2048
17	7710
18	7776
19	27594
20	24000

Table 1

LEMMA 7.2.2. *Let T_1 and T_2 be two 90/150 matrices corresponding to two maximal length CA on n cells. Then $\exists x \in GF(2^n)$, such that, $T_1x \neq T_2x$.*

Proof : $(T_1 - T_2)$ is a diagonal matrix $\neq 0_{n\times n}$. Therefore $\exists r$ such that, $t_{rr} = 1$. Then any $x \in GF(2^n)$ having its rth bit 1 will satisfy $T_1x \neq T_2x$. □

LEMMA 7.2.3. *Let T_1 and T_2 be two 90/150 matrices corresponding to two maximal length CA on n cells. Let $T_1 - T_2 = (t_{ij})$ and $\sum_{i=1}^{i=n} t_{ii} = k$. Then there are $2^k - 1$ vectors x such that $T_1x \neq T_2x$.*

These two lemmata prove that the STDs' of two distinct maximal length CA are different (though isomorphic).

7.2.1 Programmable CA

Here we introduce the notion of Programmable Cellular Automata (PCA) (see [128]). By a *global rule vector* we will mean the specification of a local rule for each cell of a CA. The global rule vector of a CA completely defines the evolution of the CA over successive time steps. This can be changed at successive time steps, that is the CA is evolved at time step 0 using rule vector α_0, is evolved at time step 1 using rule vector α_1 and so on. Such a CA is called a Programmable CA. If the local rules are restricted to be only 90/150, then a rule vector for an n-cell CA can be uniquely specified by an n-bit vector, with a 0 representing rule 90 and 1 representing rule 150. Hence the global rule vectors that are to be applied on successive time step can be stored as ROM words. In each time step the local rule for each cell is configured according to the instruction in the ROM word for that time step. PCA are used as the basic building block in [128].

7.3 Composite CA

By a composite CA we will mean a set of non-interacting CA evolving under a common clock. The notion of composite CA is quite general though here we only use composite CA constructed out of 90/150 CA. The idea is to place the elementary CA side by side and consider them as one composite CA. We say that the elementary CA are cascaded to form a composite CA. A simple characterisation of the cycle lengths of composite CA is the following

LEMMA 7.3.1. *If we cascade two maximal length linear CA on m and n cells, then the composite CA will have an STD consisting of*

1. *One isolated point cycle.*

2. *One cycle of length* $2^n - 1$.

3. *One cycle of length* $2^m - 1$

4. $2^{\gcd(n,m)} - 1$ *cycles of length* $lcm(2^n - 1, 2^m - 1)$ *each.*

Proof : First note that the point cycle will be the null configuration of the composite CA.

Let T_i be the transition matrix for CA C_i, $1 \leq i \leq 2$, and let T be the transition matrix for the composite CA.

Let c be any n-cell CA configuration. Then c is represented by an n-bit binary number a. If in one time step the CA evolves to a configuration d represented by a binary number b, then we can say that integer a in one time step evolves to the integer b.

Let b_0 and b_1 be two integers representing any two configurations of the first (m-cell) and second (n-cell) CA respectively. Then the integer $b = 2^n \times b_0 + b_1$ represent the configuration $c = (c_0\ c_1)$ of the composite CA. We will use the pair (b_0,b_1) to represent b. Then,

$$T^k(b_0,b_1) = (T_0^k(b_0),T_1^k(b_1))$$

For any integer $k \geq 0$.

1. If $b_0 = b_1 = 0$, c is the null configuration and lies on a point cycle.

2. If $b_0 = 0$ and $b_1 \neq 0$, then the minimum k for which $T^k c = c$ is $k = 2^m - 1$, and configurations of the form $(0,b_1)$ lie on this cycle.

3. If $b_0 \neq 0$ and $b_1 = 0$, then the minimum k for which $T^k c = c$ is $k = 2^n - 1$, and configurations of the form $(b_0,0)$ lie on this cycle.

4. If $b_0 \neq 0$ and $b_1 \neq 0$, then the minimum k for which $T^k c = c$ is $\mathrm{lcm}(2^n - 1, 2^m - 1)$. The number of such configurations c is $2^{m+n} - 1 - (2^m - 1) - (2^n - 1)$ and the number of such cycles is $\frac{2^{m+n}-1-(2^m-1)-(2^n-1)}{lcm(2^m-1,2^n-1)} = \frac{(2^m-1)(2^n-1)}{lcm(2^m-1,2^n-1)} = \gcd(2^m - 1, 2^n - 1) = 2^{\gcd(n,m)} - 1$. □

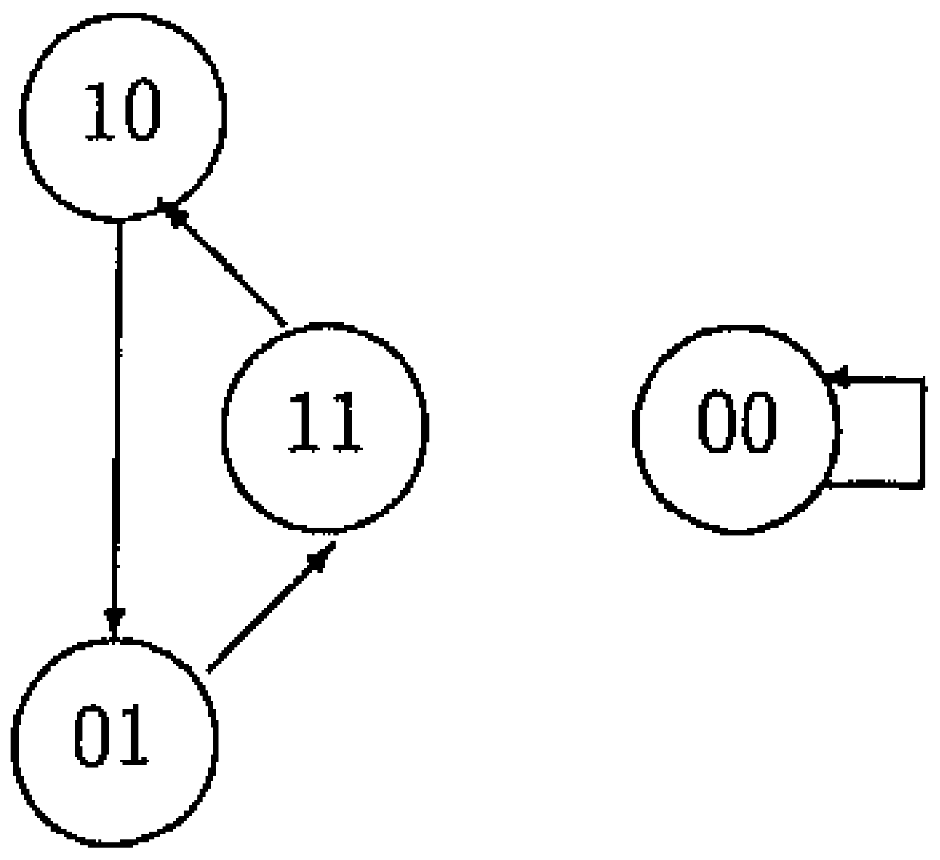

Figure 7.4: STD for $< 90, 150 >$ CA.

Similarly, we can prove the following

THEOREM 7.3.1. *If we cascade b linear and maximal length CA having n cells each, then the STD of the composite CA consists of*

1. *The null configuration on a point cycle.*
2. $\frac{(2^{nb}-1)}{(2^n-1)}$ *cycles of length* (2^n-1) *each.*

We give an example to illustrate the theorem. Consider two maximal length CA on 2 cells.

$$\text{CA1} = < 90150 >, \text{CA2} = < 90150 >$$

The cycle structure for any of these CA is shown in Figure 7.4. The cycle structure for the composite CA is shown in Figure 7.5.

This gives us a method to construct a permutation of the integers 0 to $2^{bn}-1$, such that the permutation can be decomposed into a cycle of length one having the configuration 0, and all other cycles of equal length 2^n-1. The cyclic property of the type of permutation described above is summarized in the following

LEMMA 7.3.2. *Let P be the permutation generated by the composite CA as described in Theorem 7.3.1.. Then* $\{I, P, P^2, ..., P^{2^N-2}\}$ *is a cyclic subgroup of* S_{2^m} *of order* 2^n-1. *Here* S_{2^m} *is the symmetric group of all permutations of the integers* $\{0, ..., 2^m-1\}$, *where* $m = b \times n$.

Using Lemma 7.2.2. and Lemma 7.2.3. we can state the following

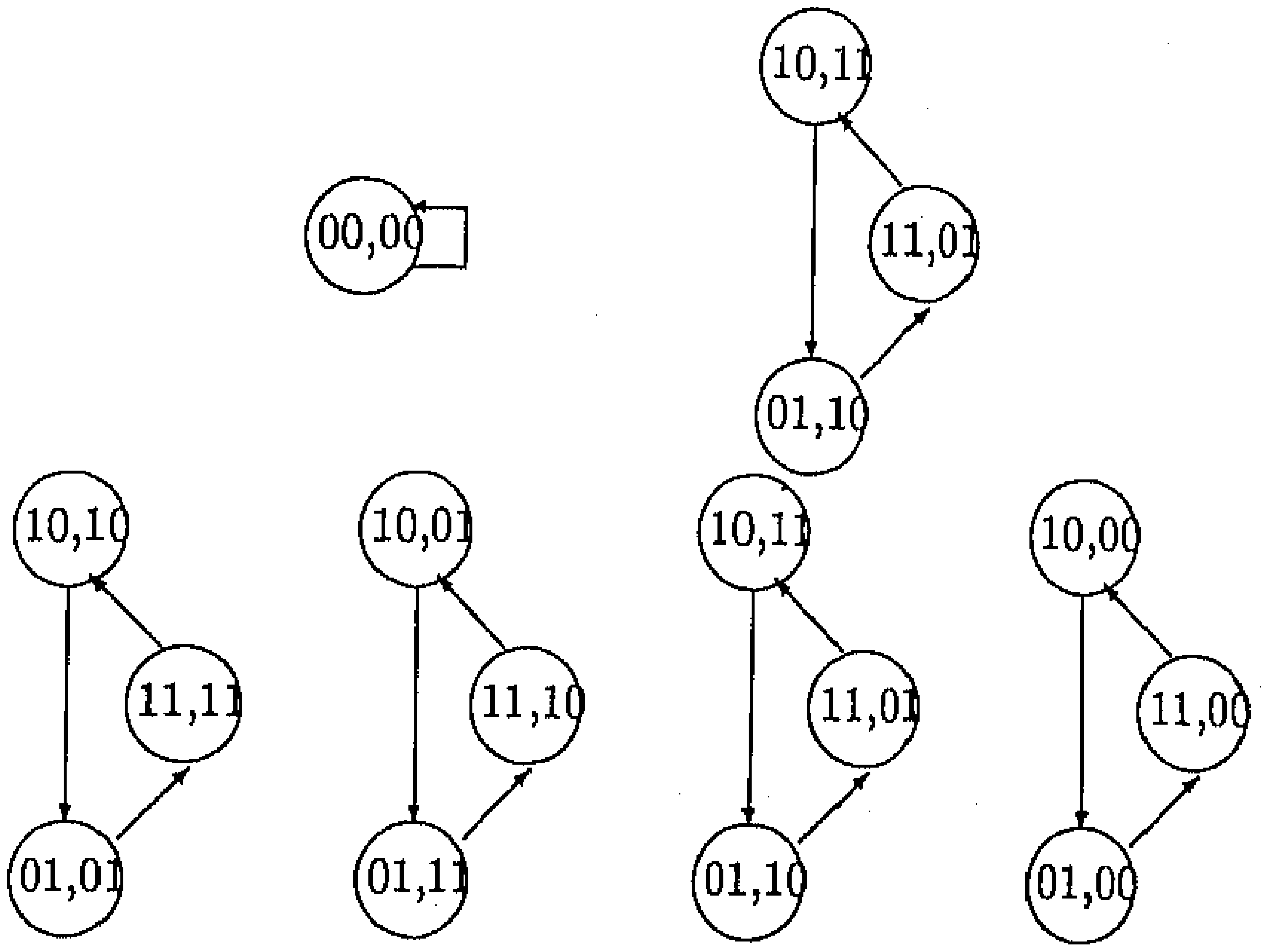

Figure 7.5: STD for composite $(< 90, 150 >, < 90, 150 >)$ CA.

LEMMA 7.3.3. *Each arrangement of the composite CA creates a permutation distinct from the permutation created by any other arrangement.*

In the next subsection we present a complete characterisation of the STD of composite CA in a more formal setup.

7.3.1 STD of Composite CA

Here we present a complete characterisation of the STD of a composite CA in terms of the STDs' of the elementary CA. In fact we take a more abstract approach and consider products of autonomous automata, that is, automata which do not take any input. Since a CA is an autonomous automata and the product that we define corresponds to the informal notion of compositeness introduced in the text, *all our results also hold for composite CA.*

An autonomous or "self-generating" automata is a dynamical system (Q, T), where Q is a finite set of states (or configurations) and T is a function from Q to Q. In effect the system evolves from an initial state in discrete time steps in a completely deterministic manner. The change of state depends totally on the present state and does not involve any "input". The dynamical behaviour of a such a system is completely described by its State Transition Diagram (STD) which is a directed graph $G = (V, E)$ where $V = Q$ and $(q_1, q_2) \in E$ iff

$T(q_1) = q_2$. It is easy to see that the STD for an autonomous automata will consist of components where each component consists of a cycle with trees of height ≥ 0 rooted on each cycle vertex. The trees represent the initial "transient" behaviour of the system while the cycles represent the "steady state" behaviour of the system and are sometimes called attractors. A number of important dynamical parameters like the number of cycles, the heights of the trees can be obtained from the STD.

Next we define product of autonomous automata. Let $\mathcal{A}_i = (Q_i, T_i)$, $1 \leq i \leq n$, be a set of autonomous automata. We define the product in the most obvious way as an automaton $\mathcal{A} = (Q, T)$, where $Q = Q_1 \times \ldots \times Q_n$ and $T : Q \to Q$ is given by $T(q_1, \ldots, q_n) = (T_1(q_1), \ldots, T_n(q_n))$. This is symbolically written as $\mathcal{A} = \mathcal{A}_1 \times \ldots \times \mathcal{A}_n$.

The justification behind this definition is to study the behaviour of the "composite" system, where each automata is evolved individually. We obtain results that describe the STD of a product automata in terms of the STDs' of the elementary automata. The proofs are purely combinatorial in nature and we provide only the main ideas.

We describe the STD for a product automaton in terms of the STDs of the elementary automaton. This description consists of two parts. The description of the cycles and the description of the trees. In what follows we will use $\mathcal{A}$ with or without subscript to denote an autonomous automaton.

The description of how cycles of the elementary automata give rise to cycles in the product automaton is given in the following

THEOREM 7.3.2. *Let $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$. Let C_i be a cycle of $\mathcal{A}_i$ of length π_i $(1 \leq i \leq 2)$. Then $\mathcal{A}$ contains $\gcd(\pi_1, \pi_2)$ cycles of length $lcm(\pi_1, \pi_2)$. Moreover, all cycles in $\mathcal{A}$ are obtained in this manner.*

Proof : Let $\mathcal{A}_i = (Q_i, T_i)$, $1 \leq i \leq 2$ and $\mathcal{A} = (Q_1 \times Q_2, T)$. Then, $T^k(q_1, q_2) = (T_1^k(q_1), T_2^k(q_2))$. Let c_i be any state on cycle C_i (i = 1,2). Then, (c_1, c_2) is on a cycle of length $\text{lcm}(\pi_1, \pi_2)$ in $\mathcal{A}$. Since there are $\pi_1\pi_2$ configurations of the form (c_1, c_2) and any configuration of this form is on a cycle of length $lcm(\pi_1, \pi_2)$, there are $\frac{\pi_1\pi_2}{lcm(\pi_1,\pi_2)} = \gcd(\pi_1, \pi_2)$ such cycles. □

This theorem should be compared to Lemma 7.3.1..

COROLLARY 7.3.1. *Let $\mathcal{A} = \mathcal{A}_1 \times \ldots \times \mathcal{A}_N$. Let π_i be the length of cycle C_i in $\mathcal{A}_i$ $(1 \leq i \leq N)$. Then there are $\frac{\pi_1 \ldots \pi_N}{lcm(\pi_1,\ldots,\pi_N)}$ cycles of length $lcm(\pi_1, \ldots, \pi_N)$ each in $\mathcal{A}$, accounting for $\pi_1 \ldots \pi_N$ configurations of the form $(c_1, \ldots, c_N)$, where c_i are configurations on cycles C_i.*

COROLLARY 7.3.2. *In the above corollary, there is exactly one cycle of length $\pi_1 \ldots \pi_N$ in $\mathcal{A}$ iff $\gcd(\pi_i, \pi_j) = 1$, for $i \neq j$, $1 \leq i, j \leq N$, that is, iff the π_is are pairwise relatively prime.*

THEOREM 7.3.3. *Let $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$. Let C_i be a cycle of length π_i in $\mathcal{A}_i$, $(i = 1,2)$. Fix a configuration a of $\mathcal{A}_1$ on cycle C_1 and a configuration b of $\mathcal{A}_2$ on cycle C_2. Then,*

1. *Configurations* $(a, T_2^i(b))$ *and* $(a, T_2^j(b))$ *are on the same cycle iff* $i \equiv j \bmod \gcd(\pi_1, \pi_2)$.

2. *Configurations* $(T_1^i(a), b)$ *and* $(T_1^j(a), b)$ *are on the same cycle iff* $i \equiv j \bmod \gcd(\pi_1, \pi_2)$.

Proof : We will prove 1. Proof of 2 is similar.
Let $g = \gcd(\pi_1, \pi_2)$. First we assume that g is distinct from π_1 and π_2, since if g is equal to either π_1 or π_2 then the proof is easy.

If: This is proved if we can prove that for $c = T_2^g(b)$, (a, c) is reachable from (a, b) or vice versa, which is proved if we can show that there exists positive integers k_1 and k_2, such that,

$$k_1\pi_1 \;=\; k_2\pi_2 \;+\; g \;=\; k \tag{1}$$

$$\text{or} \;\; k_1\pi_1 \;=\; k_2\pi_2 \;-\; g \;=\; k \tag{2}$$

since then $T^k(a, b) = (T_1^k(a), T_2^k(b)) = (a, c)$ or $T^k(a, c) = (T_1^k(a), T_2^k(c)) = (a, b)$.
To see this note that since $g = \gcd(\pi_1, \pi_2)$, $\exists$ integers k_1, k_2, such that, $\pi_1 k_1 + \pi_2 k_2 = g$. Now k_1, k_2 both cannot be positive or both cannot be negative. If k_1 ¿ 0, then, $\pi_1 k_1 - \pi_2(-k_2) = g$ and 1 holds else if $k_2 > 0$, then 2 holds.

Only If : This is proved by showing that if $i \not\equiv j \bmod g$ then $(a, T_2^i(b))$ and $(a, T_2^j(b))$ are on different cycles. We prove this by showing that no configuration of the form $(a, T_2^r(b))$ is reachable from (a, b) if $0 < r < g$.

Suppose not. Then there exists positive integers k_1, k_2, such that, $k_1\pi_1 = k_2\pi_2 + r$. Since $g|\pi_1$ and $g|\pi_2$, it follows $g|r$. But this contradicts $0 < r < g$. □

THEOREM 7.3.4. *Let* $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$. *Let* a_1 *be reachable from* x *in* $\mathcal{A}_1$ *in* p_1 *steps and* a_2 *be reachable from* y *in* $\mathcal{A}_2$ *in* p_2 *steps. Then* (a_1, a_2) *is reachable from* (x, y) *in* $\mathcal{A}$ *iff* $\exists$ *non-negative integers* k_1, k_2, *such that,* $p_1 + \pi_1 k_1 = p_2 + \pi_2 k_2$, *where* π_i *is 0 if* a_i *is not a cycle vertex, else* π_i *is the length of the cycle on which* a_i *is present* $(i = 1, 2)$. *Consequently, if* (a_1, a_2) *is a cycle vertex, then* (a_1, a_2) *is reachable from* (x, y) *iff* $\gcd(\pi_1, \pi_2) \,|\, (p_1 - p_2)$.

Proof : Let $p = p_1 + \pi_1 k_1 = p_2 + \pi_2 k_2$. Then

$$T^p(x, y) = (T_1^p(x), T_2^p(y)) = (T_1^{p_1 + \pi_1 k_1}(x), T_2^{p_2 + \pi_2 k_2})(y)) = (a_1, a_2).$$

The last statement follows from the fact that if $g = \gcd(\pi_1, \pi_2)$, then for any integer x, $g|x$ iff $x = \pi_1 k_1 + \pi_2 k_2$ for some integers k_1, k_2. □

We next present results on trees in the STD for the product automaton.

THEOREM 7.3.5. *Let* $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$. *If* B_i *is a branch of length* b_i *in* $\mathcal{A}_i$, $(i = 1, 2$ *,assume* $b_1 \geq b_2)$, *having* u_i *and* c_i *as the unreachable and cycle configurations, then corresponding to these two branches there are a total of* $(b_1 + b_2 + 1)$ *branches in* $\mathcal{A}$. *Let* c_i *be on a cycle of length* π_i. *Then in* $\mathcal{A}$,

1. *There are $(b_1 - b_2 + 1)$ branches starting with configurations of the form (x, u_2), where, $x \in \{c : c = T_1^i(u_1),\ 0 \leq i \leq b_1 - b_2\}$, having length $(b_1 - i)$, where $x = T_1^i(u_1)$ and is rooted on the cycle configuration $(c_1, T_2^{b_1-b_2-i}(c_2))$, which is on the same cycle as (c_1, c_2), iff $(b_1 - b_2 - i)$ mod π_2 is a multiple of $\gcd(\pi_1, \pi_2)$.*

2. *There are b_2 branches starting with configurations of the form (x, u_2), where $x \in \{c : c = T_1^i(u_1),\ b_1 - b_2 < i \leq b_1\}$ having length b_2 and is rooted on the cycle configuration $(T_1^{b_1-b_2-i}(c_1), c_2)$, where $x = T_1^i(u_1)$ and is on the same cycle as (c_1, c_2) iff $(b_1 - b_2 - i)$mod π_1 is a multiple of $\gcd(\pi_1, \pi_2)$.*

3. *There are b_2 branches starting with configurations of the form (u_1, x), where $x \in \{c : c = T_2^i(u_2),\ 1 \leq i \leq b_2\}$, having length b_1, and is rooted on the cycle configuration $(c_1, T_2^{b_1-b_2-i}(c_2))$, where $x = T_2^i(u_2)$, which is on the same cycle as (c_1, c_2) iff $(b_1 - b_2 - i)$ mod π_2 is a multiple of $\gcd(\pi_1, \pi_2)$.*

Proof : Only configurations of the form (x, u_2) and (u_1, x) are unreachable, and there are $(b_1 + 1) + (b_2 + 1) - 1$ such configurations, since for a branch of length b, there are $b + 1$ configurations on it.

The rest is routine verification. □

REMARK 7.3.1. *The case for $b_1 < b_2$ is similar.*

The proof of the following is similar to the above.

THEOREM 7.3.6. *Let $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$. Let B_1 be a branch of length b_1 in $\mathcal{A}_1$, with u_1 as the unreachable configuration and c_1 as the cycle configuration on a cycle of length π_1. Let C_2 be a cycle of length π_2 in $\mathcal{A}_2$. Then there are π_2 branches of length b_1 rooted on cycles of length lcm(π_1, π_2), with the unreachable configurations as (u_1, x) with x on C_2.*

REMARK 7.3.2. *This completes the cycle-cycle, branch-branch and cycle-branch combinations. Thus we can conclude*

- *Any cycle in the product automaton arises only from cycles of the elementary automaton, whereas branches in the product automaton arise from both cycles and branches of the elementary automaton.*

- *Forming the product does not increase the height of the trees beyond the maximum of all the elementary trees.*

- *The lengths of the cycles on the other hand gets multiplied. If the cycle lengths are pairwise relatively prime, then there is a "large" cycle of length equal to the product of the lengths of the cycles.*
- *The branching degree (or the indegree) of the configurations increase as indicated by the next theorem.*

THEOREM 7.3.7. *Let $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$. Let c_i be any configuration of $\mathcal{A}_i$ $(i = 1, 2)$, having indegree d_i. Then indegree of (c_1, c_2) is $d_1 d_2$. Consequently if either c_1 or c_2 is unreachable, then indegree of $(c_1, c_2) = 0$, that is (c_1, c_2) is unreachable .*

The next result describes the balance condition for a tree in a product automaton.

THEOREM 7.3.8. *Let $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$. Let (c_1, c_2) be any configuration of $\mathcal{A}$. Then the tree $\mathcal{T}$ of $\mathcal{A}$ rooted at (c_1, c_2) is balanced iff the trees $\mathcal{T}_i$ of $\mathcal{A}_i$ rooted at c_i are balanced $(i = 1, 2)$ and have the same height.*

Proof : If the heights of $\mathcal{T}_1$ and $\mathcal{T}_2$ are different or they are not balanced then quite clearly $\mathcal{T}$ cannot be balanced.

So suppose that the heights are the same and the trees $\mathcal{T}_i$ are both balanced. Then all unreachable vertices in $\mathcal{T}_i$ are at the same height h. Let (c_1, c_2) be reachable from unreachable configuration (x, y) in $\mathcal{T}$. Then the following cases may arise.

1. The path p_1 from x to c_1 in $\mathcal{T}_1$ and the path p_2 from y to c_2 in $\mathcal{T}_2$ contain no cycle vertices. Then the length of p must be h.
2. If at least one of the paths p_1 or p_2 contain a cycle vertex, then either c_1 is the first cycle vertex in p_1 (in $\mathcal{T}_1$) or c_2 is the first cycle vertex in p_2 (in $\mathcal{T}_2$). If this does not hold, then the path p from (x, y) to (c_1, c_2) in $\mathcal{T}$ does not belong to the tree rooted at (c_1, c_2). In either case the length of p is h.

By 1 and 2 it follows that $\mathcal{T}$ must be balanced. □

7.4 The Basic Cryptosystem

In this section we describe the cryptosystem and in the next section we will discuss implementation using composite CA. We describe a block cipher scheme, where each message is divided into blocks of a fixed size (possibly padding the last block with additional bits) and each block is encrypted separately. Let m be the number of bits in each message block and let $\mathcal{M}$ be the set of all possible message blocks, i.e, the set of all possible bit strings of length m. Let C be the set of all possible cipher blocks where $C \in \mathcal{C}$ consists of $m + n$ bits. Let k be an n-bit number and $M \in \mathcal{M}$ be a message (henceforth by a message we will mean a message block). We use M and k to obtain a cipher $C \in \mathcal{C}$ as follows.

Step 1: $(Q \circ NL \circ P(k))M \;=\; B.$

Step 2: $Mix(B,k) \;=\; C.$

where

(a) Q is a secret bit permutation of a m-bit string, i.e. given an m-bit string as input it will output an m-bit string which is a bit permutation of the input string.

(b) $P(k)$ is a secret linear bijective map from $\mathcal{M}$ into itself. For each k, we have a different linear transformation. This component will be constructed out of composite CA. The composite CA will provide a linear transformation from $\mathcal{M}$ to itself, and $P(k)$ is the application of this linear transformation to the message k times.

(c) NL is a secret non-linear bijective map from $\mathcal{M}$ to itself.

(d) The integer k is a random integer in the range 1 to $2^n - 2$.

(e) Mix is a secret function which places the n bits of k at fixed positions between the bits of B.

First note that Q and $P(k)$ are linear maps and NL is a non-linear map. Hence the composition of the three maps is a non-linear map. The reason for adopting such a scheme is twofold. Linear or affine transformations are not suitable for cryptographic use, since they are susceptible to attacks using algebraic techniques. Non-linear maps are more resistant to such attacks. On the other hand it is difficult to design non-linear maps having "nice" properties. However one can design linear maps having desired properties. So using a composition of linear and non-linear maps we hope to get both advantages. Next we briefly outline the intuitive reasons for each of the functions.

1. The function $P(k)$ introduces a randomisation element in the entire scheme and forms the core of the encryption system. It is generally felt that randomisation increases the security of a system, since it increases the equivocation of a message. This function has nice properties and is easily realised using a composite CA.

2. The function NL is introduced to make the whole transformation non-linear and prevent attacks based on algebraic techniques. This function ensures that the system satisfies the perfect secrecy condition of Shannon [153].

3. The function Q destroys local properties. In our realisation of $P(k)$ and NL, the value of each bit will depend only on a local neighbourhood. Hence Q will thwart attempts by a cryptanalyst to attack the system by choosing messages having local structural properties.

4. The function Mix makes it difficult to obtain the integer k.

The secret key of the system consists of the following

1. The set of functions $\{P(k) : 1 \leq k \leq 2^n - 2\}$.
2. The function NL, Q and Mix.

Given a cipher $C \in \mathcal{C}$ decryption is done as follows.

Step 1: $Extract(C) = (B, k)$.

Step 2: $(P^{-1}(k) \circ NL^{-1} \circ Q^{-1})(B) = M$.

Here *Extract* recovers the bits of k from C and constructs the pair (B, k). Since all the original functions are bijective, one applies their inverses in the reverse order to B to get back M. We will construct $P(k)$ so that $P^{-1}(k) = P(2^n - 1 - k)$.

Next we discuss the implementation of the scheme and in the last two sections we discuss security and flexibility of the system.

7.5 CA Implementation

Using the idea of composite CA, it is easy to set up the scheme. The steps are.

1. Choose $m = b \times n$, where $10 \leq n \leq 20$ and $50 \leq b \leq 100$.
2. From tables of 90/150 maximal length n-cell CA [155], select b CA. Alternatively one may construct the b many n-cell CA using the algorithm in [152].
3. Create an arbitrary arrangement of the b CA selected in step 2.
4. Cascade the individual CA to form a composite CA. Then from Theorem 7.3.1., we know that the composite CA has the all zero configuration on a cycle.of length one and $\frac{2^{bk}-1}{2^k-1}$ cycles of length $2^k - 1$ each. Evolving the CA k times $(1 \leq k \leq 2^n - 2)$ gives rise to the function $P(k)$. Since evolving the CA $2^n - 1$ times is the identity operation, we have $P^{-1}(k) = P(2^n - 1 - k)$. The composite CA thus constructed is required at both the sender's and the receiver's end.
5. Choose the bit permutation Q. To do this one has to randomly choose a permutation of the integers $\{1, \ldots, m\}$. At the sender's end we require Q and at the receiver's end we require Q^{-1}, which is easily constructed from Q. The functions Q and Q^{-1} can be very easily implemented using no additional hardware. Alternatively, they can be implemented using microprogram. This will allow the functions to be changed, leading to a possibility of changing the key.
6. The function NL is constructed in the following way. Divide the m-bit input into contiguous bits of l bits each (with possible adjustment at the end). Here l is a small integer not greater than 10. Create an arbitrary arrangement of the numbers 0 to $2^l - 1$ on the nodes of a directed cycle. Let f be the function from $\{0, \ldots, 2^l - 1\}$ to itself

whose graph is the created cycle. Then it is easy to design a synchronous circuit which realizes f. Alternatively f may also be implemented using table look-up. Later we discuss why the table look-up implementation may be preferable. Clearly f is a non-linear bijective map. For each block of l bits we choose a (possibly) different function. The function NL is the application of the corresponding f to each of the substrings of l bits that the message block has been divided into. Once NL is constructed, one can construct NL^{-1} similarly. The function NL is required at the sender's end and the function NL^{-1} is required at the receiver's end.

7. The functions *Mix* and *Extract* are easy to design. Let,

$$C = c_0, ..., c_{m-1} \text{ and } k = k_0, ..., k_{n-1},$$

Choose integers $r_1, ...r_n$, such that, $r_i \geq 1$ and $\sum_{i=1}^{n} r_i < m$. Then form the string,

$$S = c_0...c_{r_1-1}k_0c_{r_1}...c_{r_1+r_2-1}k_1c_{r_1+r_2}...c_{m-1}.$$

Then $Mix(C, k) = S$ and $Extract(S) = (C, k)$. As in the case of Q, the function Mix can be easily implemented using little additional hardware, or it can be implemented using microprogram, leading to a possibility of key change.

The following additional step is required only at the sender's end.

8. Set up a random number generator, which will produce random integers between 1 and $2^n - 2$ inclusive. This can be a true random number generator, since the integers will not be required to be regenerated. Thus for example one may use radioactive decay process to generate random numbers.

Next we present the encoding and decoding algorithms.

Encoding Algorithm E

1. Divide the message M into blocks M_i of m bits each. Pad heroes or ones at the end if required.
2. For each i do
   (a) Generate a random number k_i in the range 1 to $2^n - 2$ inclusive.
   (b) Load the composite CA with M_i.
   (c) Evolve the CA for k_i steps.
   (d) Apply NL to the output of the CA.
   (e) Apply Q to the output of NL to get B_i.
   (f) Apply Mix to (B_i, k_i) to get C_i.
   (g) Transmit C_i.

```
3. od.
```

Decoding Algorithm D

```
1. For each $C_i$ received do
    (a) $Extract(C_i) = (B_i, k_i)$.
    (b) Apply $Q^{-1}$ to $B_i$.
    (c) Apply $NL^{-1}$ to the output of $Q^{-1}$.
    (d) Load the composite CA with the output of $NL^{-1}$ and evolve for $2^n - 1 - k_i$ steps.
    (e) The output of the CA is $M_i$.
2. od
```

Note that $2^n - 1 - k_i$ is the one's complement of k_i and can be simply obtained by complementing (or inverting) the bits of k_i.

REMARK 7.5.1. *The total number of steps to encode and decode each message block is* $(2^n - 1)$. *This is same for each block, and is independent of the number of bits in a message block but depends on the number of cells of any one CA. Thus the total time to process a message varies linearly with the number of message blocks, that is, the size of the message.*

The size of the message block is m bits and each message block expands by n bits, hence the data expansion factor D_f is

$$D_f = \frac{n}{n+m} = \frac{1}{1+b}.$$

The value of D_f should be small for the scheme to be practicable. However, it has been observed that any randomisation scheme leads to some data expansion. The parameters n and b are chosen with the following constraints in mind.

1. The total time for encoding and decoding each message block is $2^n - 1$, that is, it grows exponentially with n.
2. Since the total time for encoding and decoding is independent of the length of the message block, b is to be as large as possible. Choosing a large b also ensures that D_f is small. This is however limited by hardware constraints. Since the length of a message block is $m = n \times b$, this will require a register of size m. Hardware cost will limit the size of this register and hence limit the value of b.

With $b = 50$ and $n = 16$, 800 bits will be processed in 0.3 msecs (with a clock of 200 MHz) with a data expansion factor of $\frac{16}{800} = 0.02$. (We consider only the time required for CA evolution). Thus in one second about 2.7×10^6 bits $\simeq$ 333 kbytes will be processed, which is good enough for on line processing to be possible on most communication channels. With $n = 20$ and $b = 50$, 125 bytes can be processed in 5 msecs (data expansion factor $D_f = 0.02$), leading to rate of 25 kbytes/sec. The complexity of cryptanalytic attack on the resulting set up will however be much higher. Note that from Table 1, there are more primitive polynomials for $n = 19$ than for $n = 20$. Since for a fixed length, the number of maximal length CA is at least twice the number of primitive polynomials, the security of the system is higher for $n = 19$ than for $n = 20$. Also the time required for encryption and decryption for $n = 19$ is approximately half that for $n = 20$. Thus we gain on both time and security by choosing $n = 19$ rather than $n = 20$. This anamolous situation arises because of the irregular behaviour of the number of primitive polynomials of degree n. Table 2 provides a comparative study of cryptosystems with different values of n and b.

		$b = 50$			$b = 100$		
n (bits)	T_n (msecs)	m (bits)	D_f	R_t (Mbytes/sec)	m (bits)	D_f	R_t (Mbytes/sec)
10	0.005	500	0.02	12.75	1000	0.01	25.00
11	0.010	550	0.02	7.01	1100	0.01	13.80
12	0.020	600	0.02	3.80	1200	0.01	7.50
13	0.040	650	0.02	2.10	1300	0.01	4.10
14	0.080	700	0.02	1.10	1400	0.01	2.10
15	0.160	750	0.02	0.53	1500	0.01	1.20
16	0.330	800	0.02	0.31	1600	0.01	0.61
17	0.660	850	0.02	0.20	1700	0.01	0.32
18	1.300	900.	0.02	0.09	1800	0.01	0.17
19	2.600	950	0.02	0.05	1900	0.01	0.09
20	5.200	1000	0.02	0.03	2000	0.01	0.05

Table 2

Note:

1. $T_n = \frac{(2^n-1)}{2} \times 10^{-8}$ is the total time for encryption and decryption at 200 MHz clock.

2. $m = n \times b$ is the size of the message block.

3. $D_f = \frac{n}{m+n} = \frac{1}{1+b}$ is the data expansion factor.

4. $R_t = \frac{m+n}{T_n}$ is the transmission rate. Note that the rate of encryption of message is $R_e = \frac{m}{T_n}$ and is almost equal to R_t.

7.6 Security

Let us first calculate the size of the keyspace. Fix the value of parameters n, b, $m = n \times b$ and l and let

$\mathcal{K}$ = set of all possible keys.

$\mathcal{K}_1$ = set of all possible composite CA with parameters n and b.

$\mathcal{K}_2$ = set of all possible functions NL with parameters l and m.

$\mathcal{K}_3$ = set of all possible bit permutations Q of m bit strings.

$\mathcal{K}_4$ = set of all possible functions Mix with parameters n and m.

Any key $K \in \mathcal{K}$ is a composition of functions $k_i \in \mathcal{K}_i$, $(1 \leq i \leq 3)$ along with a function Mix chosen from $\mathcal{K}_4$. In any specific implementation, for each message block there are a choice of $2^n - 2$ possible keys. It is easy to see that

$$|\mathcal{K}| = |\mathcal{K}_1| \times |\mathcal{K}_2| \times |\mathcal{K}_3| \times |\mathcal{K}_4|.$$

Let us now find the individual cardinalities.

Proposition 7.6.1. $|\mathcal{K}_1| \geq (2\frac{\phi(2^n-1)}{n})^b$.

Proof: There are a total of $\frac{\phi(2^n-1)}{n}$ primitive polynomials of degree n and hence at least twice as many maximal length CA. Out of these we create an arrangement of b (not necessarily distinct) CA. □

Proposition 7.6.2. $|\mathcal{K}_2| = ((2^l)!)^{\frac{m}{l}}$.

Proof: Each individual function f of NL has a graph where all l-bit strings are on a single directed cycle. Thus each f can be constructed in $(2^l)!$ ways and we choose a (not necessarily distinct) f for each l-bit block of the m-bit input. Thus we choose a total of $\frac{m}{l}$ such f's and hence the result follows. □

Proposition 7.6.3. $|\mathcal{K}_3| = m!$

Proof: One has to choose a permutation of the integers 1 to m.

Proposition 7.6.4. $|\mathcal{K}_4| = \begin{pmatrix} m+1 \\ n \end{pmatrix}$

Proof: Out of a total of $m+1$ positions inbetween and at the end of the m-bit message block, one has to choose n positions. □

Thus we get

Theorem 7.6.1.

$$|\mathcal{K}| \geq (2\frac{\phi(2^n-1)}{n})^b \times ((2^l)!)^{\frac{m}{l}} \times m! \times \begin{pmatrix} m+1 \\ n \end{pmatrix}$$

It is easy to see that $\mathcal{K}$ is very large for the proposed ranges of n, b and l, i.e., $10 \leq n \leq 20$, $50 \leq b \leq 100$ and $4 \leq l \leq 10$. Of course if n, b and l are not known then one has to sum over the proper ranges. It is clear that exhaustive search will fail.

Next we prove that given a cipher, a cryptanalyst will not gain any information about the actual message or the key used. This is proved by showing that $H(\underset{\sim}{M}) = H(\underset{\sim}{M}|\underset{\sim}{C})$ and $H(\underset{\sim}{K}) = H(\underset{\sim}{K}|\underset{\sim}{C})$, where $H(.)$ is the entropy function and $\underset{\sim}{M}$, $\underset{\sim}{K}$ and $\underset{\sim}{C}$ are random variables ranging over the set of messages, the set of keys and the set of ciphers respectively. Here $H(\underset{\sim}{M}|\underset{\sim}{C})$ and $H(\underset{\sim}{K}|\underset{\sim}{C})$ represent the message and key equivocation respectively. These parameters were introduced by Shannon in his seminal paper [153].

Assume a uniform probability distribution over the set of messages and the set of keys. Let $Pr(\underset{\sim}{X} = X)$ or $Pr(X)$ be the probability that the random variable $\underset{\sim}{X}$ takes the value X. Then by our assumption

$$Pr(\underset{\sim}{M} = M) = \frac{1}{|\mathcal{M}|} = \frac{1}{2^m}$$

for each $M \in \mathcal{M}$.

Theorem 7.6.2.

$$H(\underset{\sim}{M}|\underset{\sim}{C}) = H(\underset{\sim}{M})$$

Proof : This is proved by showing that $Pr(\underset{\sim}{M} = M|\underset{\sim}{C} = C) = Pr(\underset{\sim}{M} = M)$ for all $M \in \mathcal{M}$ and $C \in \mathcal{C}$. We prove this by showing that given a cipher C, the set of possible messages which could have led to this cipher is the entire set of messages. This can be proved by showing that given any $C \in \mathcal{C}$ and $M \in \mathcal{M}$, there exists $K \in \mathcal{K}$ such that applying the encoding algorithm E to the message M with K as the key will give the cipher C. We proceed as follows. Choose any function Mix from $\mathcal{K}_4$ and let $Extract(C) = (B, k)$, where B is an m-bit string and k is the integer by which the composite CA has been evolved. Let the Hamming weight of B be w_B. Choose any composite CA, load it with the message M and evolve for k steps to get an output A of Hamming weight w_A. Now we have to choose the functions NL and Q properly to map A to B. Note that the function Q being a bit permutation preserves the Hamming weight of its input. So if we can construct a function NL which maps A of weight w_A to a string T of weight w_B, then we can get a function Q in $\mathcal{K}_3$, which maps T to B, and we are done. We proceed to construct NL as follows. Let,

$$A = a_0 \ldots a_{m-1} \quad \text{and} \quad T = t_0 \ldots t_{m-1}.$$

Assume that both A and T are partitioned into blocks of l bits each,i.e.,

$$A = A_0 \dots A_{g-1} \quad \text{and} \quad T = T_0 \dots T_{g-1}$$

where $g = \frac{m}{l}$ and

$$A_i = a_{li+0} \dots a_{li+l-1}.$$
$$T_i = t_{li+0} \dots t_{li+l-1}.$$

For each i $(0 \leq i \leq g-1)$ we construct a function f_i as follows. The 2^l many l-bit configurations are arbitrarily arranged on a directed cycle (which represents the graph of f_i), with the only restriction that $f_i(A_i) = T_i$. The function NL is the application of the individual functions f_i to A_i. Clearly NL maps A to T and hence the result follows.

COROLLARY 7.6.1. *Given any $M \in \mathcal{M}$ and $C \in \mathcal{C}$, the number of keys which map M to C is given by*

$$X_{n,b,l} \geq (2\frac{\phi(2^n-1)}{n})^b \times ((2^l-2)!)^{\frac{m}{l}} \times (2^{w_B}) \times (w_B!(m-w_B)!) \times \begin{pmatrix} m+1 \\ n \end{pmatrix}$$

where w_B is the Hamming weight of B, and $Extract(C) = (B, k)$.

Proof : Note that one is free to choose the composite CA and the function Mix. There are a total of 2^{w_B} configurations of Hamming weight equal to that of B. One can choose any of these to play the role of T in the above theorem. The function NL can be chosen in $((2^l-2)!)^{\frac{m}{l}})$ ways and for each T and B there are $w_B!(m-w_B)!$ bit permutations Q which maps T of Hamming weight w_B to B also of weight w_B. □

COROLLARY 7.6.2.

$$Pr(\underset{\sim}{K} = K | \underset{\sim}{M} = M \textit{ and } \underset{\sim}{C} = C) = \frac{1}{X_{n,b,l}}$$

if K maps M to C, else it is 0.

One can similarly prove that given any cipher C and key K one gets a message M such that K maps M to C. Hence we get

THEOREM 7.6.3.

$$H(\underset{\sim}{K}|\underset{\sim}{C}) = H(\underset{\sim}{K})$$

REMARK 7.6.1. *The above two theorems show that the message and the key equivocation is the maximum possible.*This proves that the system is secure against a ciphertext only attack. *In fact, Theorem 7.6.2. shows that the system satisfies Shannon's perfect secrecy condition [153] (see also [183, Page 113]). A necessary condition for this to hold is that the number of keys is at least as large as the number of messages [183, Page 114], which is true for our system. However, one should point out that Theorem 7.6.2. and Theorem 7.6.3. actually proves a limited form of security. They show that one cipher block will not provide the attacker with any additional information regarding the message sent or the key used. This is also true for one-time pads. Moreover, in one-time pads since the key is changed for each block, the result is true for each message block. In our system, successive message blocks are encrypted with different but related keys. Thus one has to argue that given a sequence of cipher blocks, the attacker gains no knowledge of the set of keys used, or the messages sent. Proving this or its converse, that one gains information about the key or message given a sequence of cipher blocks seems to be difficult. The main disadvantage of one-time pads is that they require the generation of a completely random key string for each message. For our system, successive message blocks are encrypted with a* different *but related key. The keys are different since the number of steps for which the composite CA is evolved is different for each message block. Thus a practical compromise between implementation difficulty and provable security is achieved.*

From the proof of Theorem 7.6.2. it is clear that the perfect secrecy condition arises due to the function NL. Hence one might feel that the other constituents are really not necessary. However, it is important to note that perfect secrecy proves security against ciphertext only attack. A system with only the function NL can easily be broken if 2^l pairs of message and cipher are known. This is feasible only if l is small. Increasing l will certainly provide more security, but the hitch is that this will increase the implementation difficulty of the function NL by the same extent that it increases the security. This is clearly an undesirable situation. In our case, resistance against known plaintext attack is provided by the other constituents of the secret key.

To prove security against any known plaintext attack, one has to show that for an arbitrary subset $\mathcal{A} \subset \mathcal{M} \times \mathcal{C}$, with $|\mathcal{A}| \leq A(n,b,l)$, where $A(n,b,l)$ is an integer valued function of n, b and l, the following holds

$$H(\underset{\sim}{K}) = H(\underset{\sim}{K}|\mathcal{A}).$$

Then the system is $A(n,b,l)$-secure. Such notion of security is difficult to prove. However Corollary 7.6.2. indicates that for a single message cipher pair, the key equivocation does not substantially decrease. It is difficult to prove such results for arbitrary $\mathcal{A}$. Hence this leaves open the possibility of attacking the system with properly chosen pairs of plaintext and ciphertext. However, we believe it will not be easy to design such attacks.

7.7 Flexibility

Since the secret key consists of the secret functions Q, NL, Mix and the set of functions $\{P(k) : 1 \leq k \leq 2^k - 2\}$, to allow for change of key we must have a reconfigurable system. This can be achieved in the following way.

1. The CA is to be implemented as a PCA (see Section 7.2.1). This will allow the CA to be changed as required. See [128] for details of implementing PCA.

2. The function NL consists of a set of functions f_i, $1 \leq i \leq \frac{m}{l}$, where each f_i takes an l-bit input and produces an l-bit output. For a reconfigurable system the f_i's are to implemented using table look-up and the look-up table to be realised by a EPROM. To change the f_i's all one would need to do is to change the values in the look-up table.

3. The functions Q and Mix and their inverses are to be implemented using microprograms. This will allow these functions to be changed as and when required.

7.8 Conclusion

In this chapter we have introduced the notion of composite CA and have presented a characterisation of the STD of such CA in a more abstract setting of products of autonomous automata. A block cipher private key cryptosystem have been proposed based on composite CA. The system is easy to implement in VLSI and satisfies the perfect secrecy condition of Shannon. This *proves* that the system is secure against ciphertext only attack. Though we have not being able to prove the system secure against plaintext only attack, we believe that it will be difficult to design such attacks on the system.

Bibliography

[1] C. Adami. On modeling life. *Artificial Life*, 1:429–438, 1994.

[2] P. Aigrain and D. Beauquier. Polyomino tilings, cellular automata and codicity. *Theoretical Computer Science*, 147:165–180, 1995.

[3] V. Aladyev. Survey of research in the theory of homogeneous structures and their applications. *Mathematical Biosciences*, 15:121–154, 1974.

[4] J. Albert and K. Culik II. A simple universal cellular automaton and its one-way and totalistic version. *Complex Systems*, 1:1–16, 1987.

[5] S. Amoroso and G. Cooper. Tessellation structures for reproduction of arbitrary patterns. *Journal of Computer and System Sciences*, 5:455–464, 1971.

[6] S. Amoroso and Y.N. Patt. Decision procedures for surjectivity and injectivity of parallel maps for tessellation structures. *Journal of Computer and System Sciences*, 6:448–464, 1972.

[7] M.A. Arbib. Simple self-reproducing universal automata. *Information and Control*, 9:177–189, 1966.

[8] H. Aso and N. Honda. Dynamical characteristics of linear cellular automata. *Journal of Computer and System Sciences*, 30:291–317, 1985.

[9] A.J. Atrubin. A one dimensional real time iterative multiplier. *IEEE Transactions on Electronic Computers*, 41:394–399, 1965.

[10] Aho A.V. and Ullman J.D. *Principles of Compiler Design*. Addison Wesley, 1977.

[11] E.R. Banks. Cellular automata. AI Memo 198, MIT Artificial Intelligence Lab., 545 Technology Square - Room 821, Cambridge, Massachusetts 02139, 1970.

[12] P.H. Bardell. Analysis of cellular automata used as pseudorandom pattern generators. In *Proceedings of International Test Conference*, 1990.

[13] S. Barnett. *Matrices, Methods and Applications*. Clarendon Press, Oxford, 1990.

[14] R. Barua. Additive cellular automata and matrices over finite fields. Technical Report 17/91, Indian Statistical Institute, Stat-Math Division, 1991.

[15] R. Barua and S. Ramakrishnan. σ-game, σ^+-game, and two dimensional cellular automata. *Theoretical Computer Science*, 154:349–366, 1996.

[16] R. Barua and S. Sengupta. Architectures for arithmetic over $GF(2^m)$. In 10^{th} *International Conference on VLSI Design*, pages 465–468. IEEE, 1997.

[17] M-P. Beal and D. Perrin. Symbolic dynamics and finite automata. In *Handbook of Formal Languages, Volume 2*, pages 463–506. Springer, 1997.

[18] E.R. Berlekamp. *Algebraic Coding Theory.* McGraw Hill, 1968.

[19] S.R. Blackburn, S. Murphy, and K.G. Paterson. Comments on "Theory and Applications of Cellular Automata in Cryptography". *IEEE Transactions on Computers*, 46(5):637–638, 1997.

[20] L. Blum, M. Blum, and M. Shub. A simple unpredictable pseudo random number generator. *SIAM Journal on Computing*, 15(2), 1986.

[21] S. Boubezari and B. Kaminska. A deterministic built-in self-test generator based on cellular automata structures. *IEEE Transactions on Computers*, 44(6):805–816, 1995.

[22] G. Braga et al. Pattern growth in elementary cellular automata. *Theoretical Computer Science*, 145:1–26, 1995.

[23] W. Bucher and K. Culik II. On real-time and linear-time cellular automata. *R.A.I.R.O. Informatique Theoretique*, 18(4):307–325, 1984.

[24] A.W. Burks, editor. *Essays on Cellular Automata.* University of Illinois Press, Urbana, Illinois, 1970.

[25] Jon T. Butler. A note on cellular automata simulations. *Information and control*, 1974.

[26] Choffrut C. and K. Culik II. On real-time cellular automata and trellis automata. *Acta Informatica*, 21:393–409, 1984.

[27] H.C. Card et al. Analysis of bounded linear automata based on a method of image charges. *Journal of Computer and System Sciences*, 33:473–480, 1986.

[28] S. Chakraborty, D.R. Chowdhury, and P. Pal Chaudhuri. Theory and applications of non-group cellular automata for synthesis of easily testable finite state machines. *IEEE Transactions on Computers*, 45(4):769–781, 1996.

[29] J.H. Chang, O.H. Ibarra, and A. Vergis. On the power of one-way communication. *Journal of the ACM*, 35(3):697–726, 1988.

[30] P. Chaudhuri et al. *Additive cellular automata theory and applications, volume 1.* IEEE Press, 1997.

[31] D.R. Chowdhury et al. Design of CAECC - cellular automata based error correcting codes. *IEEE Transactions on Computers*, 43(6):759–764, 1994.

[32] D.R. Chowdhury, I. Sen Gupta, and P. Pal Chaudhuri. Cellular automata based byte error correcting codes. *IEEE Transactions on Computers*, 44(3):371–382, 1995.

[33] L.O. Chua and L. Yang. Cellular neural networks: Applications. *IEEE Transactions on Circuits and Systems*, pages 1273–1290, 1988.

[34] L.O. Chua and L. Yang. Cellular neural networks: Theory. *IEEE Transactions on Circuits and Systems*, pages 1257–1272, 1988.

[35] E.F. Codd. *Cellular Automata.* ACM Monograph Series. Academic Press, New York, 1968.

[36] S.N. Cole. Real-time computation by n-dimensional iterative arrays of finite state machines. *IEEE transactions on Computers*, 18:349–365, 1969.

[37] K. Culik II. On invertible cellular automata. *Complex Systems*, 1:1035–1044, 1987.

[38] K. Culik II. Variations of the firing squad problem and applications. *Information Processing Letters*, 30:153–157, 1989.

[39] K. Culik II and S. Dube. An efficient solution to the firing mob problem. *Theoretical Computer Science*, 91:57–69, 1991.

[40] K. Culik II, J. Gruska, and A. Salomaa. Systolic trellis automata, part 1. *International Journal of Computer Mathematics*, 15:195–212, 1984.

[41] K. Culik II, J. Gruska, and A. Salomaa. Systolic trellis automata, part 2. *International Journal of Computer Mathematics*, 16:3–22, 1984.

[42] K. Culik II, J. Gruska, and A. Salomaa. Systolic trellis automata: stability, decidability and complexity. *Information and Control*, 71:218–230, 1986.

[43] K. Culik II, L.P. Hurd, and S. Yu. Computation theoretic aspects of cellular automata. *Physica D*, 45:357–378, 1990.

[44] K. Culik II, L.P. Hurd, and S. Yu. Formal languages and global cellular automaton behaviour. *Physica D*, 45:396–403, 1990.

[45] K. Culik II, J. Pachl, and S. Yu. On the limit sets of cellular automata. *SIAM Journal on Computing*, 18:831–842, 1989.

[46] K. Culik II and S. Yu. Undecidability of CA classification schemes. *Complex Systems*, 2(2):177–190, 1988.

[47] K. Culik II and S. Yu. Cellular automata, $\omega\omega$-regular sets, and sofic systems. *Discrete Applied Mathematics*, 32:85–101, 1991.

[48] Physica D. 1984, vol. 10.

[49] Physica D. 1990, vol. 45.

[50] J. Daemen, R. Govaerts, and J. Vandewalle. A framework for the design of one-way hash functions including cryptanalysis of Damgard's one-way function based on cellular automata. In *ASIACRYPT'91*, Lecture Notes in Computer Science, 739, pages 82–96. Springer-verlag.

[51] I.B. Damgard. A design principle for hash functions. In *CRYPTO'89*, Lecture Notes in Computer Science, 435, pages 416–427. Springer-verlag.

[52] A.K. Das and P. Pal Chaudhuri. Vector space theoretic analysis of additive cellular automata and its application for pseudo-exhaustive test pattern generation. *IEEE Transactions on Computers*, 42(3):340–352, 1993.

[53] R. Das et al. Evolving globally synchronized cellular automata. In *Proceedings of the Sixth Int. Conf. on Genetic Alg.*, 1995.

[54] Chowdhury D.R. *Theory and Application of Additive Cellular Automata for Reliable and Testable VLSI Circuit Design.* PhD thesis, Dept. of Comp. Sc. and Engg. , Indian Institute of Technology, Kharagpur,INDIA, 1992.

[55] J.C. Dubacq. How to simulate turing machines by invertible one-dimensional cellular automata. *International Journal of Foundations of Computer Science*, 6(4):395–402, 1995.

[56] B. Durand. Inversion of 2d cellular automata: some complexity results. *Theoretical Computer Science*, 134:387–401, 1994.

[57] B. Durand. A random NP-complete problem for inversion of cellular automata. *Theoretical Computer Science*, 148:19–32, 1995.

[58] C. Dyer. One-way bounded cellular automata. *Information and Control*, 44:261–281, 1980.

[59] B. Elspas. The theory of autonomous linear sequential networks. *IRE Transactions on Circuits*, CT-6:45–60, 1959.

[60] P. Gacs. Reliable computation with cellular automata. *Journal of Computer and System Sciences*, 32:15–78, 1968.

[61] P. Gacs. Reliable cellular automata with self-organization. In *Proceedings of FOCS'97.* IEEE, 1997.

[62] M. Gardner. The fantastic combinations of John Conway's new solitaire game "Life". *Scientific American*, 223:120–123, 1970.

[63] M. Gardner. On cellular automata, self-reproduction, the garden of eden and the game of "Life". *Scientific American*, 224:112–117, 1971.

[64] M. Garzon. *Models of massive parallelism : Analysis of cellular automata and neural networks.* EATCS monographs. Springer Verlag, 1995.

[65] B. Gilman. Classes of linear cellular automata. *Ergodic Theory and Dynamical Systems*, 7:105, 1987.

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

[67] U. Golze. Differences between 1- and 2- dimensional cell spaces. In A. Lindenmayer and G. Rozenberg, editors, *Automata, Languages, Development*, pages 369–384. North-Holland, 1976.

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

[69] F. Green. NP-complete problems in cellular automata. *Complex Systems*, 1:453–474, 1987.

[70] P. Guan. Cellular automata based public key cryptosystem. *Complex Systems*, 1:51–56, 1987.

[71] P. Guan and Y. He. Exact results for deterministic cellular automata with additive rules. *Journal of Statistical Physics*, 43:463–478, 1986.

[72] H.A. Gutowitz. A hierarchical classification of cellular automata. *Physica D*, 45:136–156, 1990.

[73] H.A. Gutowitz. Cryptography with dynamical systems. In *Cellular Automata and Cooperative Systems.* Kluwer Academic Publishers, 1993.

[74] H.A. Gutowitz. Method and apparatus for encryption, decryption and authentication using dynamical systems, 1994. US Patent 5,365,589.

[75] V.C. Hamacher. Machine complexity versus interconnection complexity in iterative arrays. *IEEE Transactions on Computers*, 20:321–323, 1971.

[76] M. Harao and S. Noguchi. Fault tolerant cellular automata. *Journal of Computer and System Sciences*, 11:171–185, 1975.

[77] M. Harao and S. Noguchi. On some dynamical properties of finite cellular automata. *IEEE Transactions on Computers*, 27(1), 1978.

[78] G.A. Hedlund. Endomorphisms and automorphisms of the shift dynamical systems. *Mathematical Systems Theory*, 4(3):320–375, 1969.

[79] G.T. Herman et al. Synchronization of growing cellular arrays. *Information and Control*, 25(2):103–122, 1974.

[80] J.H. Holland. Universal embedding spaces for automata. In N. Wiener and J.P. Schade, editors, *Cybernetics of the Nervous System (Progress in Brain Research 17)*, pages 223–243. Elsevier, 1965. (Essay 15 of Burks).

[81] J.H. Holland. Studies of the spontaneous emergence of self-replicating systems using cellular automata and formal grammers. In A. Lindenmayer and G. Rozenberg, editors, *Automata, languages and Development*, pages 385–404. North-Holland Publishing Company, 1976.

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

[83] P.D. Hortensius et al. Cellular automata based pseudorandom number generators for built-in self-test. *IEEE Transactions on Computers Aided Design of Circuits and Systems*, 8(8):842–859, 1989.

[84] P.D. Hortensius et al. Importance sampling for Ising computers using one-dimensional cellular automata. *IEEE Transactions on Computers*, 38(6):769–774, 1989.

[85] P.D. Hortensius et al. Parallel pseudo-random number generation for VLSI systems using cellular automata. *IEEE Transactions on Computers*, 38(10):1466–1473, 1989.

[86] P.D. Hortensius, R.D. McLeod, and B.W. Podaima. Cellular automata circuits for built-in self-test. *IBM Journal of Research and Development*, 34:389–405, 1990.

[87] L.P. Hurd. Formal language characterizations of cellular automata limit sets. *Complex Systems*, 1:69–80, 1987.

[88] O.H. Ibarra and T. Jiang. On one-way cellular arrays. *SIAM Journal on Computing*, 16(6):1135–1154, 1987.

[89] O.H. Ibarra and S.M. Kim. Characterizations and computational complexity of systolic trellis automata. *Theoretical Computer Science*, 29:123–153, 1984.

[90] O.H. Ibarra, M. Palis, and S.M. Kim. Some results concerning linear iterative (systolic) arrays. *Journal of Parallel and Distributed Computing*, 2:182–218, 1985.

[91] T. Ikegami and T. Hashimoto. Active mutation in self-reproducing networks of machines and tapes. *Artificial Life*, 2:305–318, 1995.

[92] M. Ito, N. Osato, and M. Nasu. Linear cellular automata over Z_m. *Journal of Computer and System Sciences*, 27:125–140, 1983.

[93] E. Jen. Linear cellular automata and recurring sequences in finite fields. *Communications in Mathematical Physics*, 119:13–28, 1988.

[94] J. Jump and J.S. Kirtane. On the interconnection structure of cellular networks. *Information and Control*, 24:74–91, 1974.

[95] J. Kari. Reversibility of 2D cellular automata is undecidable. *Physica D*, 45:379–385, 1990.

[96] J. Kari. The nilpotency problem of one-dimensional cellular automata. *SIAM Journal on Computing*, 21:571–586, 1992.

[97] J. Kari. Reversibility and surjectivity problems of cellular automata. *Journal of Computer and System Sciences*, 48:149–182, 1994.

[98] L. Kari, G. Rozenberg, and A. Salomaa. L Systems. In *Handbook of Formal Languages, Volume 1*, pages 253–328. Springer, 1997.

[99] Y. Kawahara et al. Period lengths of cellular automata on square lattices with rule 90. *Journal of Mathematical Physics*, 36(3):1435–1456, 1995.

[100] A. Khan Raouf et al. VLSI architecture of a cellular automata machine. *Computers and Mathematics with Applications*, 33(5):79–94, 1997.

[101] T. Killingback and M. Doebeli. Self-organized criticality in spatial evolutionary game theory. *Journal of Theoretical Biology*, 191(3):335–340, 1998.

[102] D.E. Knuth. *The Art of Computer Programming, Vol. 2.* Reading, MA. Addison-Wesley, 1973.

[103] S. R. Kosaraju. Speed of recognition of context-free languages on cellular arrays. *SIAM Journal on Computing*, 1974.

[104] V.B. Kudravcev, A.C. Podkolzin, and A.A. Bolotov. *Osnovy Teorii Odnoroznych Struktur.* Nauka, Moscow, 1990.

[105] S.Y. Kung. *VLSI Array Processors.* Prentice-Hall, Englewood Cliffs, NJ, 1988.

[106] L. Le Bruyn and M. Van den Bergh. Algebraic properties of linear cellular automata. *Linear Algebra and its Applications*, 157:217–234, 1991.

[107] H.Y. Lee and Y. Kawahara. Transition diagrams of finite cellular automata. *Bulletin of Informatics and Cybernetics*, 28(1), 1996.

[108] B.K. Lemont and C. Cheng. Cellular automata model of membrane permeability. *Journal of Theoretical Biology*, 186(1):75–80, 1997.

[109] L. Levin. Average case complete problems. *SIAM Journal on Computing*, 15(1):285–286, 1986.

[110] R. Lidl and H. Niederreiter. *Encyclopedia of Mathematics, Finite Fields.* Cambridge: Cambridge University Press, 1986.

[111] A. Lindenmayer. Mathematical models for cellular interactions in development. parts I and II. *Journal of Theoretical Biology*, 18:280–315, 1968.

[112] B. Litow and Ph. Dumas. Additive cellular automata and algebraic series. *Theoretical Computer Science*, 119:345–354, 1993.

[113] A. Maass. On the sofic limit sets of cellular automata. *Ergodic Theory and Dynamical Systems*, 15:663–684, 1995.

[114] M. Mahajan. *Studies in language classes defined by different types of time-varying cellular automata.* PhD thesis, Dept. of Comp. Sc. and Engg. , Indian Institute of Technology, Madras, INDIA, 1992.

[115] O. Martin, A.M. Odlyzko, and S. Wolfram. Algebraic properties of cellular automata. *Communications in Mathematical Physics*, 93:219–258, 1984.

[116] A. Maruoka and M. Kimura. Completeness problem of one-dimensional binary scope-3 tessellation automata. *Journal of Computer and System Sciences*, 9(1):31–47, 1974.

[117] A. Maruoka and M. Kimura. Completeness problem of multi-dimensional tessellation automata. *Information and Control*, 35(1):52–86, 1977.

[118] J.L. Massey. Shift register synthesis and BCH decoding. *IEEE Transactions on Information Theory*, IT-15:122–127, 1969.

[119] W. Meier and O. Staffelbach. Analysis of pseudo random sequences generated by cellular automata. In *EUROCRYPT'91*, Lecture Notes in Computer Science, 547, pages 186–199. Springer-Verlag, 1991.

[120] M. Mignotte. *Mathematics for Computer Algebra.* Springer Verlag, 1991.

[121] M. Mitchell, J.P. Crutchfield, and P.T. Hraber. Evolving cellular automata to perform computations: mechanisms and impediments. *Physica D*, 75:361–391, 1994.

[122] M. Mitchell, P.T. Hraber, and J.P. Crutchfield. Revisiting the edge of chaos: evolving cellular automata to perform computations. *Complex Systems*, 7:89–130, 1993.

[123] E.F. Moore. *Sequential Machines. Selected Papers*, pages 213–214. Reading, MA. Addison-Wesley, 1964.

[124] E.F. Moore and G.C. Langdon. A generalized firing squad problem. *Information and control*, 12:212–220, 1968.

[125] K. Morita. Computation-universality of one-dimensional one-way reversible cellular automata. *Information Processing Letters*, 42:325–329, 1992.

[126] K. Morita. Reversible simulation of one-dimensional irreversible cellular automata. *Theoretical Computer Science*, 148:157–163, 1995.

[127] J. Myhill. The converse of Moore's Garden-of-Eden theorem. *Proceedings of the American Mathematical Society*, 14:685–686, 1963. (Essay 7 of Burks).

[128] S. Nandi, B.K. Kar, and P. Pal Chaudhuri. Theory and applications of cellular automata in cryptography. *IEEE Transactions on Computers*, 43(12), 1994.

[129] S. Nandi and P. Pal Chaudhuri. Analysis of periodic and intermediate boundary 90/150 cellular automata. *IEEE Transactions on Computers*, 45(1):1–11, 1996.

[130] H.B. Nguyen and V.C. Hamacher. Pattern synchronization in two-dimensional cellular spaces. *Information and Control*, 1974.

[131] H. Nishio and Y. Kobuchi. Fault tolerant cellular spaces. *Journal of Computer and System Sciences*, 11:150–170, 1975.

[132] T.J. Ostrand. Pattern recognition in tessellation automata of arbitrary dimensions. *Journal of Computer and System Sciences*, 5:623–628, 1971.

[133] N.H. Packard and S. Wolfram. Two-dimensional cellular automata. *Journal of Statistical Physics*, 30:901–942, 1985.

[134] P. Pal Choudhury and R. Barua. Cellular automata based VLSI architecture for computing multiplication and inverses in GF(2^m). In *Proc. 7^{th} Int. Conf. on VLSI Design*, pages 279–282. IEEE Press, 1994.

[135] W. Patterson. *Mathematical Cryptography.* Rowman and Littlefield, 1987.

[136] D.H. Pelletier. Merlin's magic square. *American Mathematical Monthly*, 94:143–150, 1987.

[137] U. Pesavento. An implementation of von Neumann's self-reproducing machine. *Artificial Life*, 2:337–354, 1995.

[138] W.W. Peterson and E.J. Weldon. *Error Correcting Codes.* MIT Press, Cambridge, Ma, 1972.

[139] G. Pighizzini. Asynchronous automata versus asynchronous cellular automata. *Theoretical Computer Science*, 132:179–207, 1994.

[140] A.S. Podkolzin. On the behaviour of homogeneous structures. *Problemy Kibernetiky*, 31:133, 1976.

[141] W. Pries, A. Thanailakis, and H.C. Card. Group properties of cellular automata and VLSI applications. *IEEE Transactions on Computers*, C-35:1013–1024, 1986.

[142] D. Richardson. Tessellations with local transformations. *Journal of Computer and Systems Sciences*, 6:373–388, 1972.

[143] S. Richter and R.F. Werner. Ergodicity of quantum cellular automata. *Journal of Statistical Physics*, 82:963–998, 1996.

[144] Z. Roka. One-way cellular automata on Cayley graphs. *Theoretical Computer Science*, 132:259–290, 1994.

[145] P. Sarkar. σ^+-automata on square grids. *Complex Systems*, 10:121–141, 1996.

[146] P. Sarkar. Products of finite autonomous automata. In *Proceedings of the 7^{th} National Seminar on Theoretical Computer Science*, pages C–45 to C–49. Indian Association for Research in Computer Science, 1997.

[147] P. Sarkar and R. Barua. Multidimensional σ-automata, π-polynomials and generalised S-matrices. *Theoretical Computer Science*, 197:111–138, 1998.

[148] P. Sarkar and R. Barua. The set of reversible 90/150 cellular automata is regular. *Discrete Applied Mathematics*, 84:199–213, 1998.

[149] T. Sato. Ergodicity of linear cellular automata over Z_m. *Information Processing Letters*, 61:169–172, 1997.

[150] J.I. Seiferas. Observations on nondeterministic multidimensional iterative arrays. In *Proceedings of the 6th Annual ACM Symposium on Theory of Computing*, pages 276–289, 1974.

[151] M. Serra et al. The analysis of one dimensional cellular automata and their aliasing properties. *IEEE Transactions on Computer Aided Design of Circuits and Systems*, 9(7):767–778, 1990.

[152] M. Serra and T. Slater. A Lanczos algorithm in a finite field and its applications. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 1990.

[153] C.E. Shannon. Communication theory of secrecy systems. *Bell Systems Technical Journal*, 28:657–715, 1949.

[154] I. Shinahr. Two- and three-dimensional firing-squad synchronization problems. *Information and Control*, 24:163–180, 1974.

[155] T. Slater and M. Serra. Tables of linear hybrid 90/150 cellular automata. Technical Report DCS-105-IR, Univ. of Victoria, Dept. of Comp. Sc., B.C., 1989.

[156] A.R. Smith III. Cellular automata complexity trade-offs. *Information and Control*, 18:466–482, 1971.

[157] A.R. Smith III. Real-time language recognition by one-dimensional cellular automata. *Journal of Computer and System Sciences*, 6:233–253, 1972.

[158] A.R. Smith III. Introduction to and survey of polyautomata theory. In A. Lindenmayer and G. Rozenberg, editors, *Automata, Languages and Development*. North-Holland Publishing Company, 1976.

[159] K. Sutner. σ-automata and Chebyshev-polynomials. *Theoretical Computer Science*, to appear.

[160] K. Sutner. Additive automata on graphs. *Complex Systems*, 2:649–661, 1988.

[161] K. Sutner. On σ-automata. *Complex Systems*, 2(1):1–28, 1988.

[162] K. Sutner. The computation complexity of cellular automata, 1989.

[163] K. Sutner. Linear cellular automata and the garden-of-eden. *Mathematical Intelligencer*, 11(2):49–53, 1989.

[164] K. Sutner. A note on the Culik-Yu classes. *Complex Systems*, 3(1):107–115, 1989.

[165] K. Sutner. Classifying circular cellular automata. *Physica D*, 45:386–395, 1990.

[166] K. Sutner. The σ-game and cellular automata. *Amer. Math. Monthly*, pages 24–34, 1990.

[167] K. Sutner. De Bruijn graphs and linear cellular automata. *Complex Systems*, 5:19–30, 1991.

[168] K. Sutner. σ-automata and π-polynomials. Technical Report CS-9408, Stevens Institute of Technology, December 1994.

[169] K. Sutner. On the computational complexity of finite cellular automata. *Journal of Computer and System Sciences*, 50:87–97, 1995.

[170] S. Takahashi. Self-similarity of linear cellular automata. *Journal of Computer and System Sciences*, 44:114–140, 1992.

[171] J.W. Thatcher. Universality in von Neumann cellular model. In A.W. Burks, editor, *Essays on Cellular Automata*, chapter 5. University of Illinois Press, 1970.

[172] T. Toffoli. Computation and construction universality of reversible cellular automata. *Journal of Computer and System Sciences*, 15:213–231, 1977.

[173] T. Toffoli and N. Margolus. *Cellular Automata Machines - A New Environment for Modeling.* MIT Press, Cambridge, MA, 1987.

[174] T. Toffoli and N.H. Margolus. Invertible cellular automata : a review. *Physica D*, 45:229–253, 1990.

[175] V.I. Varshavsky, V.B. Marakhovsky, and V.A. Pechansky. Synchronization of interacting automata. *Mathematical Systems Theory*, 4:212–230, 1970.

[176] P.M.B. Vitanyi. Sexually reproducing cellular automata. *Mathematical Biosciences*, 18:23–54, 1973.

[177] T. Vollmar. Cellular spaces and parallel algorithms, an introductory survey. In M. Feilmeier, editor, *Parallel Computation - Parallel Mathematics*, pages 49–58. North-Holland, 1977.

[178] J. von Neumann. The general and logical theory of automata. In Taub A.H., editor, *J. von Neumann Collected Works, vol 5.*, page 288. 1963.

[179] J. von Neumann. Probabilistic logics and the synthesis of relaible organisms from unreliable components. In Taub A.H., editor, *J. von Neumann Collected Works, vol 5.*, page 329. 1963.

[180] J. von Neumann. *Theory of Self-Reproducing Automata (Burks, A.W. ed.).* Univ. of Illinois Press, 1966.

[181] A. Waksman. An optimum solution to the firing squad synchronization problem. *Information and Control*, 9:67–78, 1966.

[182] J. Watrous. On one-dimensional quantum cellular automata. In *Proceedings of FOCS'96*, pages 528–537, 1996.

[183] D. Welsh. *Codes and Cryptography.* Oxford University Press, 1988.

[184] S.J. Willson. On convergence of configurations. *Discrete Mathematics*, 23:279–300, 1978.

[185] S.J. Willson. Growth patterns of ordered cellular automata. *Journal of Computer and System Sciences*, 22:29–41, 1981.

[186] S.J. Willson. Cellular automata can generate fractals. *Discrete Applied Mathematics*, 8:91–99, 1984.

[187] S.J. Willson. Growth rates and fractional dimensions in cellular automata. *Physica D*, 10:69–74, 1984.

[188] S. Wolfram. Cryptography with cellular automata. In *CRYPTO'85*, Lecture Notes in Computer Science, 218, pages 429–432. Springer-verlag.

[189] S. Wolfram. Statistical mechanics of cellular automata. *Reviews of Modern Physics*, 55:601–644, 1983.

[190] S. Wolfram. Computation theory of cellular automata. *Communications in Mathematical Physics*, 96:15–57, 1984.

[191] S. Wolfram. Universality and complexity in cellular automata. *Physica D*, 10:1–35, 1984.

[192] S. Wolfram. Random sequence generation by cellular automata. *Advances in Applied Mathematics*, 7:123–169, 1986.

[193] S. Wolfram. *Theory and Applications of Cellular Automata: including selected papers 1983-1986.* World Scientific, 1986.

[194] T. Yaku. The constructibility of a configuration in a cellular automata. *Journal of Computer and System Sciences*, 7:481–496, 1973.

[195] H. Yamada and S. Amoroso. Tessellation automata. *Information and Control*, 14:299–317, 1969.

[196] H. Yamada and S. Amoroso. A completeness problem for pattern generation in tessellation automata. *Journal of Computer and System Sciences*, 4:137–176, 1970.

[197] H. Yamada and S. Amoroso. Structural and behavioural equivalences of tessellation automata. *Information and Control*, 18:1–31, 1971.

Appendix A

Factorisation tables

Here we present complete factorisation of the first ten trinomials of the form $T_i(x) = x^{2^i} + x + 1$. Note that such trinomials are square free. The value in the first column is the depth of the corresponding irreducible factor $\tau(x)$ in the second column.

$i = 1, T_i(x) = x^2 + x + 1$

Depth	$\tau(x)$
5	$1 + x + x^2$

$i = 2, T_i(x) = x^4 + x + 1$

Depth	$\tau(x)$
17	$1 + x + x^4$

$i = 3, T_i(x) = x^8 + x + 1$

Depth	$\tau(x)$
5	$1 + x + x^2$
63	$1 + x^2 + x^3 + x^5 + x^6$

$i = 4, T_i(x) = x^{16} + x + 1$

Depth	$\tau(x)$
255	$1 + x^3 + x^5 + x^6 + x^8$
257	$1 + x + x^3 + x^4 + x^5 + x^6 + x^8$

$i = 5, T_i(x) = x^{32} + x + 1$

Depth	$\tau(x)$
5	$1 + x + x^2$
205	$1 + x + x^2 + x^3 + x^8 + x^9 + x^{10}$
1023	$1 + x^2 + x^3 + x^4 + x^8 + x^9 + x^{10}$
1025	$1 + x + x^5 + x^6 + x^8 + x^9 + x^{10}$

$i = 6, T_i(x) = x^{64} + x + 1, 2^{12} = 4096$

Depth	$\tau(x)$
17	$1 + x + x^4$
1365	$1 + x^2 + x^3 + x^6 + x^8 + x^9 + x^{12}$
4095	$1 + x^2 + x^5 + x^9 + x^{12}$
4095	$1 + x^5 + x^8 + x^9 + x^{12}$
4097	$1 + x + x^4 + x^5 + x^8 + x^9 + x^{12}$
4097	$1 + x + x^2 + x^4 + x^5 + x^9 + x^{12}$

$i = 7, T_i(x) = x^{128} + x + 1, 2^{14} = 16384$

Depth	$\tau(x)$
5	$1 + x + x^2$
5461	$1 + x^4 + x^5 + x^7 + x^8 + x^{10} + x^{11} + x^{13} + x^{14}$
5461	$1 + x^2 + x^4 + x^5 + x^7 + x^{10} + x^{11} + x^{13} + x^{14}$
16383	$1 + x^6 + x^7 + x^9 + x^{11} + x^{13} + x^{14}$
16383	$1 + x^2 + x^6 + x^7 + x^8 + x^9 + x^{11} + x^{13} + x^{14}$
16383	$1 + x^2 + x^3 + x^4 + x^6 + x^7 + x^8 + x^{10} + x^{11} + x^{13} + x^{14}$
16385	$1 + x + x^5 + x^7 + x^8 + x^{10} + x^{11} + x^{13} + x^{14}$
16385	$1 + x + x^3 + x^6 + x^7 + x^{10} + x^{11} + x^{13} + x^{14}$
16385	$1 + x + x^2 + x^3 + x^4 + x^5 + x^7 + x^9 + x^{11} + x^{13} + x^{14}$
16385	$1 + x + x^2 + x^4 + x^6 + x^7 + x^8 + x^9 + x^{11} + x^{13} + x^{14}$

$i = 8, T_i(x) = x^{256} + x + 1, 2^{16} = 65536$

Depth	$\tau(x)$
13107	$1 + x^3 + x^6 + x^7 + x^{10} + x^{11} + x^{13} + x^{14} + x^{16}$
21845	$1 + x^4 + x^5 + x^6 + x^7 + x^8 + x^{11} + x^{12} + x^{13} + x^{14} + x^{16}$
21845	$1 + x^2 + x^3 + x^4 + x^7 + x^8 + x^{11} + x^{12} + x^{13} + x^{14} + x^{16}$
21845	$1 + x^3 + x^4 + x^7 + x^{11} + x^{12} + x^{13} + x^{14} + x^{16}$
65535	$1 + x^2 + x^5 + x^7 + x^{10} + x^{11} + x^{13} + x^{14} + x^{16}$
65535	$1 + x^2 + x^3 + x^5 + x^6 + x^7 + x^9 + x^{10} + x^{11} + x^{12} + x^{13} + x^{14} + x^{16}$
65535	$1 + x^2 + x^3 + x^4 + x^5 + x^7 + x^9 + x^{11} + x^{13} + x^{14} + x^{16}$
65537	$1 + x + x^4 + x^5 + x^7 + x^8 + x^{10} + x^{11} + x^{13} + x^{14} + x^{16}$
65537	$1 + x + x^6 + x^7 + x^9 + x^{11} + x^{13} + x^{14} + x^{16}$
65537	$1 + x + x^2 + x^6 + x^7 + x^8 + x^9 + x^{11} + x^{13} + x^{14} + x^{16}$
65537	$1 + x + x^4 + x^7 + x^9 + x^{10} + x^{11} + x^{12} + x^{13} + x^{14} + x^{16}$
65537	$1 + x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7 + x^9 + x^{10} + x^{11} + x^{12} + x^{13} + x^{14} + x^{16}$
65537	$1 + x + x^3 + x^5 + x^7 + x^8 + x^9 + x^{11} + x^{13} + x^{14} + x^{16}$
65537	$1 + x + x^3 + x^4 + x^5 + x^6 + x^7 + x^8 + x^9 + x^{10} + x^{11} + x^{12} + x^{13} + x^{14} + x^{16}$
65537	$1 + x + x^2 + x^5 + x^6 + x^7 + x^{11} + x^{12} + x^{13} + x^{14} + x^{16}$
65537	$1 + x + x^2 + x^4 + x^5 + x^7 + x^{10} + x^{11} + x^{13} + x^{14} + x^{16}$

$i = 9, T_i(x) = x^{512} + x + 1, 2^{18} = 262144$

Depth	$\tau(x)$
5	$1+x+x^2$
63	$1+x^2+x^3+x^5+x^6$
7085	$1+x+x^2+x^3+x^4+x^5+x^7+x^8+x^{10}+x^{11}+x^{13}+x^{14}+x^{16}+x^{17}+x^{18}$
12483	$1+x^2+x^3+x^5+x^7+x^8+x^{10}+x^{11}+x^{13}+x^{14}+x^{16}+x^{17}+x^{18}$
13797	$1+x^2+x^3+x^6+x^{10}+x^{12}+x^{16}+x^{17}+x^{18}$
20165	$1+x+x^3+x^5+x^6+x^7+x^{11}+x^{12}+x^{13}+x^{14}+x^{16}+x^{17}+x^{18}$
20165	$1+x+x^3+x^6+x^9+x^{12}+x^{16}+x^{17}+x^{18}$
29127	$1+x^2+x^9+x^{10}+x^{16}+x^{17}+x^{18}$
37449	$1+x^3+x^7+x^8+x^9+x^{10}+x^{11}+x^{12}+x^{13}+x^{14}+x^{16}+x^{17}+x^{18}$
37449	$1+x^4+x^5+x^6+x^{16}+x^{17}+x^{18}$
52429	$1+x+x^2+x^5+x^6+x^8+x^{16}+x^{17}+x^{18}$
52429	$1+x+x^2+x^5+x^7+x^8+x^9+x^{11}+x^{13}+x^{14}+x^{16}+x^{17}+x^{18}$
87381	$1+x^8+x^9+x^{10}+x^{16}+x^{17}+x^{18}$
87381	$1+x^3+x^4+x^5+x^9+x^{12}+x^{16}+x^{17}+x^{18}$
87381	$1+x^2+x^4+x^7+x^{11}+x^{12}+x^{13}+x^{14}+x^{16}+x^{17}+x^{18}$
262143	$1+x^3+x^4+x^6+x^7+x^8+x^9+x^{11}+x^{13}+x^{14}+x^{16}+x^{17}+x^{18}$
262143	$1+x^2+x^6+x^7+x^{10}+x^{11}+x^{13}+x^{14}+x^{16}+x^{17}+x^{18}$
262143	$1+x^2+x^5+x^6+x^7+x^8+x^9+x^{10}+x^{11}+x^{12}+x^{13}+x^{14}+x^{16}+x^{17}+x^{18}$
262143	$1+x^3+x^5+x^7+x^{10}+x^{11}+x^{13}+x^{14}+x^{16}+x^{17}+x^{18}$
262143	$1+x^5+x^6+x^7+x^9+x^{10}+x^{11}+x^{12}+x^{13}+x^{14}+x^{16}+x^{17}+x^{18}$
262143	$1+x^5+x^{10}+x^{12}+x^{16}+x^{17}+x^{18}$
262145	$1+x+x^4+x^6+x^7+x^8+x^{10}+x^{11}+x^{13}+x^{14}+x^{16}+x^{17}+x^{18}$
262145	$1+x+x^2+x^3+x^6+x^7+x^9+x^{11}+x^{13}+x^{14}+x^{16}+x^{17}+x^{18}$
262145	$1+x+x^2+x^3+x^{16}+x^{17}+x^{18}$
262145	$1+x+x^2+x^4+x^9+x^{10}+x^{16}+x^{17}+x^{18}$
262145	$1+x+x^3+x^4+x^6+x^8+x^{10}+x^{12}+x^{16}+x^{17}+x^{18}$
262145	$1+x+x^2+x^3+x^5+x^8+x^9+x^{12}+x^{16}+x^{17}+x^{18}$
262145	$1+x+x^3+x^4+x^5+x^6+x^9+x^{10}+x^{16}+x^{17}+x^{18}$
262145	$1+x+x^2+x^4+x^5+x^8+x^{10}+x^{12}+x^{16}+x^{17}+x^{18}$
262145	$1+x+x^2+x^3+x^4+x^6+x^{10}+x^{12}+x^{16}+x^{17}+x^{18}$

$i = 10,\ T_i(x) = x^{1024} + x + 1,\ 2^{20} = 1048576$

Depth	$\tau(x)$
17	$1 + x + x^4$
13981	$1 + x^2 + x^3 + x^4 + x^6 + x^8 + x^{16} + x^{17} + x^{20}$
25575	$1 + x^2 + x^5 + x^6 + x^{10} + x^{12} + x^{16} + x^{17} + x^{20}$
61681	$1 + x + x^3 + x^5 + x^6 + x^8 + x^9 + x^{12} + x^{16} + x^{17} + x^{20}$
61681	$1 + x + x^2 + x^5 + x^{16} + x^{17} + x^{20}$
69905	$1 + x^2 + x^3 + x^4 + x^6 + x^7 + x^8 + x^9 + x^{10} + x^{11} + x^{12} + x^{13} + x^{14} + x^{17} + x^{20}$
95325	$1 + x^2 + x^3 + x^5 + x^7 + x^{11} + x^{12} + x^{13} + x^{14} + x^{17} + x^{20}$
209715	$1 + x^2 + x^3 + x^4 + x^5 + x^6 + x^9 + x^{12} + x^{16} + x^{17} + x^{20}$
209715	$1 + x^3 + x^5 + x^8 + x^9 + x^{10} + x^{16} + x^{17} + x^{20}$
209715	$1 + x^6 + x^7 + x^{11} + x^{12} + x^{13} + x^{14} + x^{17} + x^{20}$
209715	$1 + x^2 + x^4 + x^5 + x^6 + x^{10} + x^{12} + x^{17} + x^{20}$
349525	$1 + x^3 + x^6 + x^7 + x^9 + x^{10} + x^{11} + x^{12} + x^{13} + x^{14} + x^{16} + x^{17} + x^{20}$
349525	$1 + x^2 + x^3 + x^5 + x^9 + x^{10} + x^{16} + x^{17} + x^{20}$
349525	$1 + x^2 + x^5 + x^6 + x^7 + x^9 + x^{11} + x^{13} + x^{14} + x^{17} + x^{20}$
349525	$1 + x^2 + x^4 + x^7 + x^8 + x^{10} + x^{11} + x^{13} + x^{14} + x^{17} + x^{20}$
349525	$1 + x^2 + x^8 + x^9 + x^{12} + x^{17} + x^{20}$
349525	$1 + x^4 + x^5 + x^6 + x^8 + x^{10} + x^{12} + x^{17} + x^{20}$
349525	$1 + x^4 + x^6 + x^9 + x^{10} + x^{17} + x^{20}$
1048575	$1 + x^2 + x^6 + x^8 + x^9 + x^{10} + x^{16} + x^{17} + x^{20}$
1048575	$1 + x^4 + x^9 + x^{12} + x^{16} + x^{17} + x^{20}$
1048575	$1 + x^4 + x^5 + x^6 + x^7 + x^8 + x^9 + x^{11} + x^{13} + x^{14} + x^{16} + x^{17} + x^{20}$
1048575	$1 + x^2 + x^3 + x^5 + x^6 + x^7 + x^{10} + x^{11} + x^{13} + x^{14} + x^{16} + x^{17} + x^{20}$
1048575	$1 + x^3 + x^{10} + x^{12} + x^{16} + x^{17} + x^{20}$
1048575	$1 + x^5 + x^6 + x^8 + x^{10} + x^{12} + x^{16} + x^{17} + x^{20}$
1048575	$1 + x^2 + x^4 + x^6 + x^7 + x^8 + x^{11} + x^{12} + x^{13} + x^{14} + x^{16} + x^{17} + x^{20}$
1048575	$1 + x^3 + x^6 + x^{17} + x^{20}$
1048575	$1 + x^7 + x^{10} + x^{11} + x^{13} + x^{14} + x^{16} + x^{17} + x^{20}$
1048575	$1 + x^3 + x^4 + x^7 + x^9 + x^{11} + x^{13} + x^{14} + x^{16} + x^{17} + x^{20}$
1048575	$1 + x^2 + x^3 + x^7 + x^8 + x^9 + x^{11} + x^{13} + x^{14} + x^{17} + x^{20}$
1048575	$1 + x^3 + x^4 + x^5 + x^6 + x^7 + x^8 + x^{10} + x^{11} + x^{13} + x^{14} + x^{17} + x^{20}$
1048577	$1 + x + x^2 + x^3 + x^5 + x^7 + x^{11} + x^{12} + x^{13} + x^{14} + x^{16} + x^{17} + x^{20}$
1048577	$1 + x + x^2 + x^4 + x^8 + x^9 + x^{12} + x^{17} + x^{20}$
1048577	$1 + x + x^2 + x^3 + x^5 + x^9 + x^{10} + x^{17} + x^{20}$
1048577	$1 + x + x^3 + x^4 + x^6 + x^{17} + x^{20}$
1048577	$1 + x + x^4 + x^7 + x^{10} + x^{11} + x^{13} + x^{14} + x^{16} + x^{17} + x^{20}$
1048577	$1 + x + x^2 + x^3 + x^4 + x^8 + x^{10} + x^{12} + x^{16} + x^{17} + x^{20}$
1048577	$1 + x + x^2 + x^3 + x^6 + x^8 + x^{16} + x^{17} + x^{20}$
1048577	$1 + x + x^3 + x^5 + x^8 + x^9 + x^{10} + x^{17} + x^{20}$
1048577	$1 + x + x^2 + x^5 + x^6 + x^{10} + x^{12} + x^{17} + x^{20}$
1048577	$1 + x + x^9 + x^{12} + x^{16} + x^{17} + x^{20}$
1048577	$1 + x + x^2 + x^4 + x^5 + x^7 + x^9 + x^{10} + x^{11} + x^{12} + x^{13} + x^{14} + x^{16} + x^{17} + x^{20}$
1048577	$1 + x + x^3 + x^4 + x^5 + x^7 + x^8 + x^{11} + x^{12} + x^{13} + x^{14} + x^{17} + x^{20}$
1048577	$1 + x + x^2 + x^3 + x^4 + x^6 + x^7 + x^8 + x^9 + x^{10} + x^{11} + x^{12} + x^{13} + x^{14} + x^{16} + x^{17} + x^{20}$
1048577	$1 + x + x^3 + x^4 + x^6 + x^7 + x^9 + x^{10} + x^{11} + x^{12} + x^{13} + x^{14} + x^{16} + x^{17} + x^{20}$
1048577	$1 + x + x^4 + x^6 + x^9 + x^{10} + x^{16} + x^{17} + x^{20}$
1048577	$1 + x + x^5 + x^6 + x^8 + x^{10} + x^{12} + x^{17} + x^{20}$
1048577	$1 + x + x^2 + x^4 + x^7 + x^8 + x^{10} + x^{11} + x^{13} + x^{14} + x^{16} + x^{17} + x^{20}$
1048577	$1 + x + x^3 + x^4 + x^7 + x^9 + x^{11} + x^{13} + x^{14} + x^{17} + x^{20}$
1048577	$1 + x + x^4 + x^5 + x^6 + x^7 + x^8 + x^9 + x^{11} + x^{13} + x^{14} + x^{17} + x^{20}$
1048577	$1 + x + x^2 + x^4 + x^6 + x^7 + x^8 + x^{11} + x^{12} + x^{13} + x^{14} + x^{17} + x^{20}$
1048577	$1 + x + x^5 + x^7 + x^8 + x^9 + x^{10} + x^{11} + x^{12} + x^{13} + x^{14} + x^{17} + x^{20}$
1048577	$1 + x + x^2 + x^6 + x^8 + x^9 + x^{10} + x^{17} + x^{20}$

Appendix B

Kronecker Product

In this section we present basic results on Kronecker product of two matrices. These will be required for the analysis of generalized S-matrix, which is shown to be a sum of Kronecker products. More elaborate discussion on Kronecker product can be found in [13].

DEFINITION B.0.1. *The Kronecker (or direct) product of an $m \times n$ matrix A and an $p \times q$ matrix B is defined to be the $mp \times nq$ matrix*

$$A \otimes B \;=\; ((a_{ij}B))$$

PROPOSITION B.0.1. *1. $\otimes$ is in general not commutative.*

2. $A \otimes (\alpha B) \;=\; \alpha A \otimes B \;=\; \alpha(A \otimes B)$

3. $A \otimes (B \otimes C) \;=\; (A \otimes B) \otimes C$

4. $A \otimes (B + C) \;=\; (A \otimes B) + (A \otimes C)$

5. $(B + C) \otimes A \;=\; B \otimes A + C \otimes A$

6. $I \otimes A \;=\; diag(A, A, \ldots, A)$ *and* $I_n \otimes I_p \;=\; I_{np}$

7. $\mathbf{0} \otimes A = A \otimes \mathbf{0} = \mathbf{0}$

8. $(AB \otimes CD) \;=\; (A \otimes C)(B \otimes D)$

9. $(A \otimes B)^T \;=\; A^T \otimes B^T$

LEMMA B.0.1. *If A and B are two square matrices, then c is an eigen value of $A \otimes B$ iff it is of the form $a_i b_j$, where a_i and b_j are eigen values of A and B respectively.*

LEMMA B.0.2. *Let A be an $m \times m$ matrix and B be an $n \times n$ matrix. Then the matrix equation $AX + XB = Y$ where X, Y are $m \times n$ matrices, is completely equivalent to the system of linear equations $(A \otimes I_n + I_m \otimes B)vec(X) = vec(Y)$. Here $vec(X)$ is the matrix X written out in row major form. The equation has an unique solution iff the characteristic polynomials of A and B are relatively prime. In this case the transformation $T(X) = (A \otimes I_n + I_m \otimes B)vec(X)$ is invertible.*

LEMMA B.0.3. *Let A be an $m \times m$ matrix having characteristic polynomial $p(x)$. Let B be an $n \times n$ matrix having characteristic polynomial $q(x)$. Let α_i, $1 \leq i \leq m$ and β_j, $1 \leq j \leq n$ be the roots of $p(x)$ and $q(x)$ respectively. Then γ is a root of the characteristic polynomial for the matrix*

$$T = I_m \otimes B + A \otimes I_n$$

iff γ is of the form $\alpha_i + \beta_j$ for some i, j.

The matrix T is also called the Kronecker sum of A and B.

Appendix C

Resultant

We introduce the concept of resultant of two polynomials and state some of its elementary properties. A more detailed discussion can be found in [120].

DEFINITION C.0.2. *The resultant of two polynomials*

$$f(x) = \sum_{i=0}^{m} a_i x^i \quad g(x) = \sum_{j=0}^{n} b_j x^j$$

where a_i and b_j are elements of a ring R, is given by the determinant of the following $(m+n) \times (m+n)$ matrix (called the matrix of Sylvester).

$$Res_x(f,g) = det[c_{ij}] = det \begin{bmatrix} a_m & a_{m-1} & \cdots & a_1 & a_0 & 0 & \cdots & 0 \\ 0 & a_m & \cdots & & a_1 & a_0 & \cdots & 0 \\ \cdot & \cdot & \cdot & & & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & & & \cdot & \cdot & \cdot \\ b_n & b_{n-1} & \cdots & b_1 & b_0 & 0 & \cdots & 0 \\ \cdot & \cdot & \cdot & & & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & & & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdots & b_n & b_{n-1} & \cdot & \cdots & b_0 \end{bmatrix}$$

where c_{ij} is precisely given by,

$$c_{i,j} = a_{m-j+i} \quad for \quad 1 \leq i \leq n,$$
$$c_{n+i,j} = b_{n-j+i} \quad for \quad 1 \leq i \leq m$$

where

$$a_i = 0 \quad for \quad i \notin \{0,\ldots,m\} \; and$$
$$b_j = 0 \quad for \quad j \notin \{0,\ldots,n\}$$

Thus $Res_x(f,g)$ is an element of R.

PROPOSITION C.0.2. *1.* $Res(f,0) \ = \ 0$

2. $Res(g,f) \ = \ (-1)^{mn} Res(f,g)$

3. If $deg(f) \ = \ m \leq n \ = \ deg(g)$ *and if* $g = fq + h$, *then* $Res(f,g) \ = \ a_m^{n-m+1} Res(f,h)$

4. $Res(f,g) \ = \ 0$ *iff* f *and* g *have a common nontrivial factor.*

5. Let R *be an integral domain. Then if* $\alpha_i, i = 1,\ldots,m$ *are roots of* f *and* $\beta_j, j = 1,\ldots,n$ *are roots of* g,

$$\begin{aligned} Res(f,g) \ &= \ a_m^n \textstyle\prod_{i=1}^m g(\alpha_i) \\ &= \ (-1)^{mn} b_n^m \textstyle\prod_{j=1}^n g(\beta_j) \\ &= \ a_m^n b_n^m \textstyle\prod_{i=1}^m \prod_{j=1}^n (\alpha_i - \beta_j) \end{aligned}$$

When the coefficients are taken over $GF(2)$, the sign does not matter and all a_i's and b_j's are either 0 or 1.

LEMMA C.0.4. *If* $P(x)$ *and* $Q(x)$ *are two nonconstant polynomials with coefficients in a field* K *and with roots* $\alpha_1,\ldots,\alpha_m$ *and* $\beta_1,\ldots,\beta_n$ *respectively. Then the roots of the polynomial*

$$R(y) \ = \ Res_x(P(x+y), Q(-x))$$

are the elements $\alpha_i + \beta_j$, $1 \leq i \leq m$, $1 \leq j \leq n$.

Proof : Let $P_1(x) \ = \ P(x+y)$. Then the coefficients of $P_1(x)$ belong to $K[y]$. Since $Q(x)$ is a polynomial over K, it is certainly a polynomial over $K[y]$.

Now the coefficient of x^m in $P_1(x)$ is equal to the coefficient of x^m in $P(x+y)$. Also if α is a root of $P(x)$, $\alpha - y$ is a root of $P_1(x)$. Similarly if β is a root of $Q(x)$, $-\beta$ is a root of $Q(-x)$. Hence by proposition 2.6(5),

$$\begin{aligned} R(y) \ &= \ Res_x(P(x+y), Q(-x)) \\ &= \ a_m^n b_n^m (-1)^{mn} \textstyle\prod_{i=1}^m \prod_{j=1}^n (\alpha_i + \beta_j - y) \end{aligned}$$

Hence the result follows. □

LEMMA C.0.5. *Let* A, B *and* C *be polynomials with coefficients in a ring* R *and let* $a \in R$. *Then,*

(a) $Res((x-a)A(x), B(x)) \ = \ B(a).Res(A(x), B(x))$

(b) $Res(A(x), B(x)C(x)) \ = \ Res(A(x), B(x)).Res(A(x), C(x))$