Procedures for calculating reversible one-dimensional cellular automata

Juan Carlos Seck Tuoh Mora *
Sergio V. Chapa Vergara†
Genaro Juárez Martínez‡
Departamento de Ingeniería Eléctrica, Sección Computación,
Centro de Investigación y de Estudios Avanzados del I.P.N.,
Av. IPN 2508, Col. San Pedro Zacatenco, México 07360, D.F.
Tel: (525) 7476558, 3756 and 3758
Fax: (525) 7477002

Harold V. McIntosh §
Departamento de Aplicación de Microcomputadoras
Instituto de Ciencias, UAP
Apartado Postal 461 (72000) Puebla, Puebla, México

May, 2002

Abstract

Two algorithms for calculating reversible one-dimensional cellular automata of neighborhood size 2 are presented. It is explained how this kind of automata represents all the rest. Using two basic properties of these systems such as the uniform multiplicity of ancestors and Welch indices, these algorithms only require matrix products and the transitive closure of binary relations to yield the calculation of reversible automata. The features, advantages and differences of these algorithms are described and results for reversible automata of 3, 4, 5 and 6 states are comprised.

Key words: cellular automata, theory of matrices, Welch theory, algorithms

1 Introduction

One way of studying the behavior of a given system is understanding the local interactions of its parts. Cellular automata are perhaps the simplest model for this kind of analysis. The concept of cellular automata began with the work of John von Neumann [15] for proving the existence of self-reproductive system. The

*email: seck@computacion.cs.cinvestav.mx, seck@mac.com
†email: schapa@cs.cinvestav.mx
‡email: genaro@enigma.red.cinvestav.mx, genarojm@correo.unam.mx
§email: mcintosh@servidor.unam.mx, Web page: http://delta.cs.cinvestav.mx/~mcintosh

theory of cellular automata followed a gradual development and acquires new interest thanks to the work of John Conway and the automaton called "Life" [3], which is able to yield complex global behaviors in spite of its simple local interactions. Another important work in the field is made by Stephen Wolfram [16] who analyzed the characteristics from one set of one-dimensional rules instead of a particular one.

A special type of cellular automaton is that where the system can return to previous stages, these are reversible automata. Reversible one-dimensional cellular automata were carefully studied by Gustav A. Hedlund as automorphisms of the shift dynamical system [4]. Another important papers in the theory of reversible automata are those made by Masakazu Nasu [12] using a graph-theoretical approach and by Jarkko Kari using block permutations [7].

Reversible cellular automata are systems where the information is conserved during its evolution, for this reason they present a very interesting mathematical theory, and have been used as models for data ciphering, information coding [8] and simulation of reversible physical phenomena [14], [16] among other applications.

In this way, an important work is to enumerate all the possible reversible one-dimensional cellular automata. This search begins with the paper by Amoroso y Patt [1], and has been continued by Tim Boykett [2] using an algebraic approach, by David Hillman [5] using chains of symbols and recently by Hendrik Moraal [11] using graph-theoretical tools.

This paper presents the properties of these systems and proposes two original procedures for calculating and classifying all the possible reversible one-dimensional cellular automata of 3, 4, 5 and 6 states. One algorithm is based in forming the state-pair matrix for reviewing its cycles. The other consists of using the connectivity matrix of each state. In this way, with very simple matrix operations (products and transitive closures) these algorithms detect the invertible behavior of reversible one-dimensional cellular automata.

The paper is organized as follows: section 2 explains the operation of one-dimensional cellular automata with special emphasis in the reversible case. It also shows that every one-dimensional cellular automaton can be simulated by another of neighborhood size 2. In this way, it is enough to analyze this case for understanding the rest. Section 3 presents the detection of reversible automata by means of the state-pair matrix. The operation, advantages and limits of this algorithm are explained. Section 4 illustrates how the connectivity matrices are used for detecting the reversible behavior. It is also explained the advantages and limits of this algorithm with regard of the previous one. Section 5 uses both algorithms for yielding and classifying all the reversible one-dimensional cellular automata of 3, 4, 5 and 6 states. It is described how the automata are generated and classified, showing besides some additional procedures implemented for improving the task. Finally, section 6 provides the concluding remarks of the paper.

2 Basic concepts

A one-dimensional cellular automaton is composed by a one-dimensional array of cells, where each takes one state from a given finite set K. If the array is infinite, then every cell is indexed by an element of $\mathbb{Z}$ and an assignation from each element in $\mathbb{Z}$ into a single element in K is given. Each assignation of the one-dimensional array is a configuration of the automaton. The cardinality of the set K is presented by k, and for $n \in \mathbb{Z}^+$, K^n defines the set of sequences of n states.

For a given configuration, every cell evolves depending on its current state and the state of its r neighbor cells at each side; r is called the neighborhood radius and $2r+1$ is called the neighborhood size, where $2r+1 \in \mathbb{Z}^+$ and K^{2r+1} is the set of neighborhoods of the automaton. The mapping $\varphi : K^{2r+1} \to K$ is called the evolution rule of the automaton. All the cells in the configuration update their values at the same time according to the evolution rule, and the whole configuration evolves in a new one, yielding the global

evolution of the system. Thus, the global evolution of the automaton depends on the local interaction of its parts.

Sequences of $2r + 1$ cells generate individual states and sequences of $n + 2r$ states generate sequences of n states, in this way, a sequence of $n + 2r$ cells contains n neighborhoods where each overlaps with its contiguous neighborhoods in $2r$ cells. For $n \in \mathbb{Z}^+$, a sequence $w \in K^{n+2r}$ is ancestor of $v \in K^n$ if w evolves in v applying the evolution rule φ to all the neighborhoods in w.

Given some evolution rule, a sequence of states evolves in a single way, but it can be generated by one, many, or no ancestors. The set of sequences without ancestors is the Garden of Eden of the automaton defined by this rule. A cellular automaton is reversible if each configuration has a single ancestor and the Garden of Eden does not exist.

These systems were widely studied by the papers of Gustav A Hedlund y Masakazu Nasu, which show that these systems hold the following properties:

1. Every finite sequence of states has the same number of ancestors that all the others, and this number is equal to k^{2r}, which is the average of ancestors per state. This property is called the principle of uniform multiplicity of ancestors [9].
2. For $n_0 \in \mathbb{Z}^+$ and $n \geq n_0$, every sequence in K^n holds that its ancestors have three parts (left, central and right) in their structure, each indexed respectively by L, M and R, where $LR = k^{2r}$ and $M = 1$. In other words, the ancestors share the same central part and differ at the ends.
3. For $n_0 \in \mathbb{Z}^+$ and $n \geq n_0$, the ancestors of each sequence in K^n fulfill that one of their L left endings is equal to one and only one of their R right endings.

For a reversible automaton with evolution rule φ, property 2 implies that there is another inverse rule φ^{-1} which makes invertible the evolution of the automaton. For every finite configuration with at least n_0 cells, property 3 says that it has a single ancestor, assuring the reversibility of the automaton.

Before presenting the algorithms for detecting the reversible behavior in one-dimensional cellular automata, it will be explained how any cellular automaton can be simulated by another with neighborhood size 2, which makes simpler the detection [2] [6]. Given a cellular automaton of k states, neighborhood radius r and evolution rule φ, for $w \in K^{2r}$, the ancestors of w has $4r$ cells. Take a new set S of states such that its cardinality $\mid S \mid$ is equal to k^{2r}. Then there is a bijection from K^{2r} to S and $\mid S^2 \mid = k^{4r}$. In this way, the mapping $K^{4r} \to K^{2r}$ induced by φ can be also presented by $\tau : S^2 \to S$, but τ is the evolution rule of a cellular automaton with neighborhood size 2. With this procedure, any automaton is simulated by another of neighborhood size 2 and it is enough to detect reversible automata of neighborhood size 2 for including all the other cases.

For reversible one-dimensional cellular automata of k states and neighborhood size 2, the properties previously explained are now as follows:

1. Each sequence has k ancestors.
2. For $n_0 \in \mathbb{Z}^+$ and $n \geq n_0$, the ancestors of each sequence in K^n begin with L distinct states and finish in R different states, with $LR = k$, sharing at least one common internal state.
3. For $n_0 \in \mathbb{Z}^+$ and $n \geq n_0$, the ancestors of each sequence in K^n have one initial state equal to one final state.

Now the algorithms for detecting the reversible behavior of one-dimensional cellular automata with neighborhood size 2 will be presented.

3 Algorithm using the state-pair matrix

A fast and effective algorithm for reviewing the reversibility of a given evolution rule is provided by the state-pair matrix [10]. For automata with neighborhood size 2, the evolution rule is presented by a matrix, where the row indices are the left cells of the neighborhoods and the column indices are the right cells of them. Each entry in the matrix takes the value of the state in which the neighborhood represented by its indices evolves. For constructing the state-pair matrix take all the ordered pairs of states in the automaton, these pairs will be the indices by rows and columns of the state-pair matrix and its order is k^2.

Given an evolution rule φ and two ordered pairs $(i_1, j_1), (i_2, j_2)$ where $i_m, j_m \in K,\ m = 1, 2$, the entry $((i_1, j_1), (i_2, j_2))$ at the state-pair matrix is defined as follows:

$$((i_1, j_1), (i_2, j_2)) = \begin{cases} 1 & if\ \varphi(i_1, i_2) = \varphi(j_1, j_2) \\ \\ 0 & \text{in other case} \end{cases} \tag{1}$$

The state-pair matrix is a $0-1$ matrix and shows what pairs of states evolve in the same state. In this way a binary relation is established between ordered pairs of states. With the transitive closure of this relation, its connected components are obtained. Every connected component represents two cyclic ancestors of the same sequence.

If a given connected component contains an ordered pair where both elements are different one another, then there is a sequence of states with two different cyclic ancestors and the automaton is not reversible. Then for detecting if a given one-dimensional cellular automaton is reversible, the state-pair matrix is formed, its transitive closure is calculated and its connected components are verified. If the only connected component so formed is by ordered pairs where both elements are equal then the automaton is reversible, in other case the rule is discarded (Table 1).

Evolution rule	State-pair matrix	Transitive closure
$\begin{array}{c} \\ 0 \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{ccc} 0 & 1 & 2 \end{array} \\ \begin{pmatrix} 2 & 2 & 0 \\ 0 & 0 & 2 \\ 1 & 1 & 1 \end{pmatrix} \end{array}$	$\begin{array}{c} \\ 00 \\ 01 \\ 02 \\ 10 \\ 11 \\ 12 \\ 20 \\ 21 \\ 22 \end{array} \begin{array}{c} \begin{array}{ccccccccc} 00 & 01 & 02 & 10 & 11 & 12 & 20 & 21 & 22 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \end{array}$	$\begin{array}{c} \\ 00 \\ 01 \\ 02 \\ 10 \\ 11 \\ 12 \\ 20 \\ 21 \\ 22 \end{array} \begin{array}{c} \begin{array}{ccccccccc} 00 & 01 & 02 & 10 & 11 & 12 & 20 & 21 & 22 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$

Table 1: Reversible one-dimensional cellular automata of 3 states, neighborhood size 2, and Welch indices $L = 1$ and $R = 3$. The state-pair matrix and its transitive closure are presented, note that the only connected component is composed by $\{(0, 0), (1, 1), (2, 2)\}$.

The number of necessary steps to form the state-pair matrix is $O(k^4)$ and the transitive closure of each matrix is calculated by Warshall's algorithm [13] which is $O(k^6)$. The procedure for checking the connected

components is $O(k^4)$ so the algorithm is polynomial with regard to the number of states, but with a very big exponent. Besides, the algorithm is identically applied for all the evolution rules to review, in this way a little increment in the number of states also increases considerably the calculation time. Another observation is that the algorithm gives no information about the size of the inverse rule.

4 Algorithm using connectivity matrices

Another way of detecting the reversibility of a given evolution rule is using the connectivity matrix of each state [10]. For any $s \in K$, form a new matrix indexing both rows and columns with the elements of K. In the matrix, the entry (i, j) is defined as follows:

$$(i,j) = \begin{cases} 1 & if\ \varphi(ij) = s \\ \\ 0 & \text{in other case} \end{cases} \tag{2}$$

Then each state has a $0-1$ matrix representing its ancestors. For a bigger sequence take the product of the connectivity matrices following the order of the states forming this sequence. The product yields a new matrix showing the ancestors of the sequence. Using the properties exposed in Section 2, the conditions that a connectivity matrix must hold for presenting the ancestors of a reversible automaton are as follows:

1. The matrix must be a $0-1$ matrix.
2. The sum of elements of each matrix must be equal to k.
3. Each matrix must have one and only one diagonal element equal to 1.
4. Each matrix must have L identical rows, each with R entries equal to 1, the other rows are zero.
5. Each matrix must have R identical columns, each with L entries equal to 1, the other columns are zero.

Condition 1 assures that there is at most a single form of connecting a state i with another j. Since there is another form of returning from j to i specified in another connectivity matrix, if there are more ways for connecting i with j in the matrix then the undefined product alternating both matrices yields a growing number of ancestors, which contradicts the uniform multiplicity principle.

Condition 2 verifies that the uniform multiplicity of ancestors is fulfilled and conditions 3, 4 and 5 review the behavior of the matrix with regard of the Welch indices.

In this way a recursive algorithm is defined. The algorithm forms all the possible sequences of states up to some prefixed length and the connectivity matrix of every sequence is obtained in each step. If the matrix holds all the conditions previously described, then the extensions of the sequence are not studied and another different sequence is now analyzed (Table 2).

$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$	$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \end{array}$	$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix} \end{array}$	$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \end{array}$

Table 2: Instances of valid connectivity matrices for reversible one-dimensional cellular automata of 4 states, neighborhood size 2 and Welch indices $L = 2$ and $R = 2$.

If the connectivity matrix has any of the following features:

- The sum of elements is different from k.
- The matrix has an element greater than 1.
- The trace of the matrix is different from 1.
- A row has more than R entries distinct from 0.
- A column has more than L entries distinct from 0.
- The matrix presents the ancestors of a sequence with greater length than the prefixed one.

Then the automaton is not reversible and the rule is discarded (Table 3).

Sum greater than 4	Row with 3 positive entries	2 positive diagonal entries	An entry greater than 1
$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$	$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \end{array}$	$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$	$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 1 \end{pmatrix} \end{array}$

Table 3: Some forbidden connectivity matrices for reversible automata of 4 states, neighborhood size 2 and Welch indices $L = 2$ and $R = 2$.

In other case where it is not resolved whether the sequence is reversible or not, then the extensions of the sequence are studied. Thus with simple matrix products the algorithm detects if a given automaton is reversible or not.

A problem with the algorithm is the selection of the maximum length for reviewing. This problem was resolved independently by Nasu [12] and Kari [6] with a length at most of $k - 1$ if a Welch index is equal to 1. But the question stills open when both indices are different from 1. In this case one bound is provided by the state-pair matrix reviewing how long a sequence can be without repeating any pair, which yields a bound equal to k^2.

In this way, the prefixed length for stopping the procedure depends on the number of states, and the algorithm is exponential in the worst case. However, it allows to know the length of the inverse rule, observing the maximum length where a given connectivity matrix acquires the reversible form. The algorithm also yields the Welch subsets defining the reversible behavior of a given evolution rule.

In opposite case to the algorithm using the state-pair matrix, this one does not need to run completely for discarding some evolution rule, it is enough that the ancestors of a single sequence fail the reversible conditions for stopping the process.

5 Experimental results

Both process were implemented for calculating reversible one-dimensional cellular automata of 3, 4, 5 and 6 states; for the second and the fourth case there are two types of automata, one with a Welch index 1 and another with a Welch index 2.

An fundamental point is the generation of the evolution rules to review. The matrix presenting some evolution rule must hold some special conditions before analyzing its reversibility:

- Each state must appear the same number of times that all the others in the matrix, fulfilling the uniform multiplicity principle.
- Any state can not appear in the same row more than R times or in the same column more than L times. Thus, in the case of L or R equal to 1, every column or row respectively is a permutation of K.
- The entry (i, j) must be different from the entry (j, i) and if a given state is placed in the entries (i, j) and (k, m), then the entries (j, i) and (m, k) must be different as well. These restrictions avoid the formation of different cyclic ancestors for some sequence of two cells.

With the previous conditions, the number of rules to generate and studying has a very good decrement, which improves the calculation of reversible one-dimensional cellular automata.

Each rule will be identified by a single number in a suitable base as Stephen Wolfram did assigning a binary number to every rule in one-dimensional cellular automata of 2 states and neighborhood size 3 [16]. In the case of automata with neighborhood size 2, every evolution rule is ordered in a descendent way from the neighborhood $((k-1)(k-1))$ to the neighborhood (00).

For instance, a one-dimensional cellular automaton of 3 states has nine different neighborhoods. Once ordered these neighborhoods, they are grouped in blocks of three neighborhoods and each block is identified by a single number in base 27, taking the leftmost element of the block as the most significative one. For automata of 4 states the same process is applied grouping the evolution rule in blocks of two cells and using numbers in base 16. For automata of 5 states, the first ordered neighborhood is not grouped and the rest is grouped in blocks of two cells, numbering each block with a number in base 25.

For a given evolution rule, equivalent evolution rules are yielded reflecting, applying symmetry, permuting states or a combination of these operations in the initial evolution rule [17]. In this way, the evolution rules will be clustered, where two rules in a cluster are equivalent. Every cluster is identified by its rule with the smallest number in the corresponding base.

All these features and both algorithms for proving the reversibility of possible invertible evolution rules were implemented in a standard C program over a Macintosh G4 Dual, and results for one-dimensional cellular automata of 3, 4, 5 and 6 were obtained (Table 4).

No. states	Welch indices	Total clusters	Inverse size	No. clusters
3	L=1,R=3	2	2	2
4	L=1,R=4	8	2	5
			3	3
	L=2,R=2	3	2	3
			3	0
5	L=1,R=5	132	2	17
			3	50
			4	65
6	L=1,R=6	14570	2	83
			3	1216
			4	5631
			5	7640
	L=2,R=3	3079	2	920
			3	1651
			4	508
			5	0

Table 4: Results of the processes for calculating reversible automata.

The program for 3 and 4 states runs in a time smaller than one second with both algorithms. For 5 states the search takes thirty seconds by the algorithm using connectivity matrices and forty-five seconds by the algorithm using the state-pair matrix. For 6 states the algorithm using connectivity matrices was only applied by its better performance, and it takes eleven hours for $L = 1$ and nine hours for $L = 2$. For 7 states, both algorithms are impractical by the huge number of rules to generate.

6 Concluding remarks

The algorithms for calculating reversible one-dimensional cellular automata have allowed to know and classify these systems both for their behavior and for their complexity. It is one of the problems where increasing the number of states also increments the run time in a huge amount, still for small values as it was seen in the previous results.

Another observation is that the algorithm using connectivity matrices which has the greatest order in the worst case is the faster one. In this way the order of its average case must be smaller than the order of the algorithm using state-pair matrices. To know the order of the average case of the algorithm using connectivity matrices is an open question, it is the same that knowing the average of the number of sequences to review for detecting if a given evolution rule is irreversible, and the average length of the sequences where it happens. These questions are not easily solved for evolution rules like the ones generated by the previous program.

Finally, for the case of reversible automata with both Welch indices different from 1, evolution rules with an inverse neighborhood longer than $k-1$ were not found, different from the case of reversible automata with a Welch index 1. This result suggests that $k-1$ can be a general bound for all reversible one-dimensional cellular automaton, which is an important property to formalize.

References

[1] Serafino Amoroso and Yale Patt. Decision procedures for surjectivity and injectivity of parallel maps for tessellation structures. *Journal of Computer ans System Sciences*, 6:448–464, 1972.

[2] Tim Boykett. Combinatorial construction of one-dimensional reversible cellular automata. *Contributions to general algebra*, 9:81–90, 1994.

[3] Martin Gardner. The fantastic combinations of john conway's new solitaire game "life". *Scientific American*, 223(4):120–123, 1970.

[4] G. A. Hedlund. Endomorphisms and automorphisms of the shift dynamical system. *Mathematical Systems Theory*, 3:320–375, 1969.

[5] David Hillman. The structure of reversible one-dimensional cellular automata. *Physica D*, 52:277–292, 1991.

[6] Jarkko J. Kari. On the inverse neighborhoods of reversible cellular automata. In *Lindenmayer Systems*, pages 477–495. Springer-Verlag, Berlin, 1992.

[7] Jarkko J. Kari. Representation of reversible cellular automata with block permutations. *Mathematical Systems Theory*, 29:47–61, 1996.

[8] Douglas Lind and Brian Marcus. *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, Cambridge, 1995.

[9] Harold V. McIntosh. *Linear Cellular Automata*. Universidad Autonoma de Puebla, Apartado Postal 461 (72000) Puebla, Puebla, Mexico, 1990. also available in http://delta.cs.cinvestav.mx/~mcintosh.

[10] Harold V. McIntosh. Reversible cellular automata. *http:// delta. cs. cinvestav. mx/ ~mcintosh*, 1991.

[11] Hendrik Moraal. Graph-theoretical characterization of invertible cellular automata. *Physica D*, 141:1–18, 2000.

[12] Masakazu Nasu. Local maps inducing surjective global maps of one dimensional tesellation automata. *Mathematical Systems Theory*, 11:327–351, 1978.

[13] Harold S. Stone. *Discrete mathematical structures and their applications*. Chicago : Science research associates, 1973.

[14] Tommaso Toffoli and Norman Margolus. *Cellular Automata Machines*. MIT Press, London, 1987.

[15] John von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, Urbana and London, 1966. edited by Arthur W. Burks.

[16] S. Wolfram, editor. *Theory and Applications of Cellular Automata*. World Scientific, Singapore, 1986.

[17] Andrew Wuensche and Mike Lesser. *The global dynamics of cellular automata : an atlas of basin of attraction fields of one-dimensional cellular automata*. Addison-Wesley, 1992.