STUDIES ON THE SYSTEMS OF ONE-DIMENSIONAL ITERATIVE AUTOMATA

by

Youichi KOBUCHI

February, 1973

STUDIES ON THE SYSTEMS OF ONE-DIMENSIONAL ITERATIVE AUTOMATA

by

Youichi KOBUCHI

Submitted in partial fulfillment of
the requirement for the degree of
DOCTOR OF ENGINEERING
at
KYOTO UNIVERSITY
KYOTO, JAPAN

February, 1973

PREFACE

More or less systematic treatments of iterative systems, i.e., regularly interconnected aggregates of copies of a machine, began with the development of highly structured information processing machines and of IC or LSI techniques. The regularity and homogeneity of objects are the basis of seemingly wide applicability of the systmes. Though the objects are presented in a rather general form, the existing theories seem to be not so general and powerful as to be able to generate deep and fruitful results.

We treat the systems of one-dimensional iterative automata as a whole through several selected topics. By these efforts, more general ways of treating the systems are hoped to be established.

The contents of this thesis are as follows.

In Chapter 1, a general introduction to the studies of iterative automata or cellular arrays is given by showing the classified problems in the fields with historical developments in mind. The nature of our research and the principles we have adopted are also explained, which we hope will be an Ariadne's thread throughout the studies.

Rigorous definitions of the systems we are going to investigate are given in Chapter 2 with several useful concepts and notations.

Certain state sets which are supposed to be some characteristic sets of states in the systems are treated first. The main results in Chapter 3 say that each set of the treated state configurations constitutes a regular set under some appropriate boundary condition. The proofs are constructive in that corresponding finite state automata accepting each state set are shown. A related decidability problem is also revealed.

Next, we turn to a more specific characteristic of the systems, i.e., information transmission. In Chapter 4, we classify the systems of one-dimensional iterative automata into three categories according to the right to left information transmission capability of the systems and reveal some of the fundamental properties of each class. A related unsolvability result is also shown by embedding Turing machines into the cellular systems.

In Chapter 5, the problems of controlling the systems are treated. The strongly-connectedness, the simply-connectedness, the reachability, and the controllability are fundamental concepts in finite automata theory. We define indices of the systems concerning the above mentioned concepts of a finite automaton. Relations between an index and an initial state, relations among the indices are revealed. As for control steps, several basic properties are shown.

A restricted class of systems are considered in Chapter 6; in case the cardinality of cell state set equals two, and having the cyclic boundary condition. The topic is the information maintence capability in the systems, and the equivalence of the capability and the preservability of the parities of the number of information containing cells is the main result.

In the last Chapter, two examples of algebraic treatments of the systems are shown. One is an analysis method of such systems that are defined via abelian groups, which is a generalization of the analysis method of linear automata. The other is an application and an extension of pair algebra in the area of an error analysis in the systems.

February, 1973
Youichi Kobuchi

ACKNOWLEDGEMENT

The author wishes to thank Prof. T. Sakai of Kyto Univ. for guiding and encouraging him through this work. More than a word of thanks are due to Dr. H. Nishio whose constant guidance and co-operation are sincerely appreaciated. Many of the contents are the outgrowths of the discussions with him, and in particular, Section 3.5 was originally written by Dr. Nishio.

The author is also grateful to Prof. E. Teramoto for providing him fine atmosphere and general support.

At the various stages of this research, Dr. K. Sugata, Mr. M. Iwamoto, Mr. T. Sakaguchi, Mr. H. Takahashi, and many others discussed with the author and gave valued comments.

Last but not least, the author thanks the Data Processing Center of Kyoto Univ. for providing him opportunities of utilizing the facilities.

LIST OF SYMBOLS

A number indicates the page where the Symbol first appears.

S 12

f 12

(S,f) 12

C_i 13

b_l 14

b_r 14

$A_x(S,f,n)$ 15

$A_x(n)$ 15

(b_l,b_r) 15

c 15

o 15

r 15

A_x 15

$F_x(s_1 s_2 \ldots s_n)$ 16

$F_{b_l}(s_1 s_2 \ldots s_n; b_r)$ 16

$P_x(s_1 s_2 \ldots s_n)$ 16

$\approx\!\!>$ 16

$\{f\}$ 17

$P(A_x)$ 19

$I_\alpha(A_x)$ 21

$E(A_x)$ 24

$B(A_x)$ 28

α 39

$P(\alpha)$ 39

C_I 39

C_{II} 39

C_{III} 39

$A_{b_l}(n)$ 47

π 47

$\pi(0)$ 48

$\pi(1)$ 48

$\pi_{\overline{R}}$ 48

$\pi(u^{(1)},u^{(2)},\ldots,u^{(v)})$ 48

$\pi^{(n)}$ 48

$\pi_k^{(n)}$ 49

$\pi_{\widetilde{R}}^{(n)}$ 50

$(\pi_{\overline{R}}^{(n)})$ 51

A_D 53

$P_i(\alpha)$ 54

$R(q)$ 63

R_{st} 63

$m_{st}(M)$ 63
R_{sp} 64
$m_{sp}(M)$ 64
R^n 65
R^n_{st} 65
R^n_{sp} 65
ϕ_{n+1} 66
N_{st} 67
Γ_α 68
$N_r(\Gamma_\alpha)$ 69
$N_c(\Gamma_\alpha)$ 69
$\bar{N}_r$ 70
$\bar{N}_c$ 71
N_{sp} 72
$d^n(\alpha^n, \beta^n)$ 79
$F_{(n_i, n_o)}$ 85
$w(\alpha)$ 85
$W(x)$ 88
T_n 103
π_t 108
μ 108
$(\pi_A\ \pi_B)_{S-S}$ 110
$(\pi_A\ \pi_B)_{I-S}$ 110
$[\pi_A\ \pi_B]$ 110
$(\pi_A\ \pi_B)_{I_r-S}$ 114
$(\pi_A\ \pi_B)_{I_l-S}$ 115
$[\pi_A\ \pi_B\ \pi_C]$ 115

CONTENTS

Chapter 1. Introduction 1

1.1. Iterative Automata 1

1.2. Classification of Problems with Brief Historical Notes 5

1.3. Standpoint of This Research 10

Chapter 2. Definitions of the Systems 12

2.1. Cell Automaton and its Transition Function. 12

2.2. Iterative Automaton and Boundary Conditions 13

2.3. System of One-Dimensional Iterative Automata. 15

2.4. Homomorphism, Isomorphism, and Symmetry 16

Chapter 3. Some Regular State Sets in the Systems 18

3.1. N-Automaton 19

3.2. Set of Passive Configurations 19

3.2.1. Fixed Boundary Condition 20

3.2.2. Cyclic Boundary Condition 22

3.2.3. Other Boundary Conditions 23

3.3. Set of Garden of Eden Configurations 24

3.3.1. Fixed Boundary Condition 24

3.3.2. Cyclic Boundary Condition 27

3.3.3. Other Boundary Conditions 28

3.4. Branching Point Configurations 28

3.5. Iterative Array Viewed as Special Case of Gsm Mapping 31

3.5.1. Gsm Mapping for Fixed Boundary Condition 32

3.5.2. Gsm Mapping for Cyclic Boundary Condition . . .34

3.5.3. Gsm Mapping for Open and Reflexive Boundary Conditions 35

3.6. Concluding Remarks 36

Chapter 4. Information Transmission Capability of the Systems. .38
4.1. Introduction 38
4.2. Lemmas for Classification 40
4.3. Simplification Theorem for Class I 43
4.3.1. Simplification Theorem 43
4.3.2. Systems with External Input 45
4.4. Characterization of Classification via State Partition 47
4.4.1. Definitions on Partitions 47
4.4.2. Characterization of Class I 49
4.4.3. Sufficient Condition for Class II 51
4.5. D-Set and its Decision Problem 53
4.5.1. Classification of Systems of Half-Infinite Iterative Automata 53
4.5.2. Embedding a Turing Machine 55
4.5.3. Halting Problem of TM and ID-Configuration . . .57
4.5.4. Decision Problems 59
4.6. Concluding Remarks 61

Chapter 5. Indices Concerning Controllability of the Systems . .62
5.1. Introduction 62
5.1.1. Definitions .63
5.2. Index of Strongly-Connectedness 65
5.3. Indices of Reachability and Controllability 68
5.4. Index of Simply-Connectedness 71
5.5. Illustrative Examples 72
5.6. Relations Among Indices 78
5.7. Remarks on Control Steps 79
5.8. Concluding Remarks 82

Chapter 6. Information Maintenance Capability of the Systems. . . .84

6.1. Introduction .84

6.2. Definitions 84

6.3. (n_i, n_o)-Preservability. 86

6.4. P-Preservability ($|S|=2$). 87

6.5. (2,2)-Preservability ($|S|=2$). 97

6.6. Concluding Remarks. 100

Chapter 7. Some Algebraic Treatments of the Systems 101

7.1. Analysis of Systems Defined via Abelian Group . . . 101

7.1.1. In Case S is a Cyclic Group of Order p^q. 102

7.1.2. In Case S is a Module of Type $(p,p,\ldots,p)$ . . 104

7.2. Analysis of Error in Terms of Pair Algebra. 108

7.2.1. Applications of Pair Algebra 109

7.2.2. Analysis for Three Neighbor Cells 114

7.3. Concluding Remarks 116

References 117

Appendices 122

CHAPTER 1

INTRODUCTION

1.1. Iterative Automata

An iterative automaton is a collection of copies of a finite state automaton each of which is located at a lattice point of Euclidean space and the limit of interactions between the component automata is homogeneously restricted. It is therefore defined by giving the dimension of the space, a finite state automaton and a neighborhood relation which determines the pattern of interactions between the component automata. The identical component automata are sometimes referred to as 'cell automata' or simply 'cells' and the automaton composed therof as 'cellular automaton'. These names have the origin in studies of iterative automata as models of cellular structure in biology.

The most fundamental feature characterizing the systems of iterative automata seems to be the structural and functional homogeneity. The cell automata located regularly at the lattice points are the same in their functions and the same neighborhood relation is applied to each cell automaton.

Now let us examine several concepts concerning the models of iterative automata. These are related with the models of cell automata and the systems themselves.

As for cell automata, they may be *combinational* or *sequential*. Systems of combinational cellular automata are investigated in great detail by Hennie [16]. A unilateral one-dimensional iterative array of logical circuits can be converted to a sequential machine by a time-space transformation. Accordingly, a unilateral one-dimensional iterative automaton composed of sequential cells may be transformed to a two-dimensional iterative automaton composed of combinational cells. In case cells are sequential machines, they may be of *Moore type* or *Mealy type*. As in the ordinary theory of finite state automata, each of the two types of cells could be converted equivalently to the other type in the sense that one is a simulator of the other. There may exist some differences depending which type of cells is used in considering certain problems. In what follows, Moore type cell is used exclusively, for it seems to the author the handiest to treat.

Further, there may be comparisons of *synchronous* cell to *asynchronous* cell, of *deterministic* to *nondeterministic* and of *deterministic* to *probabilistic*. Most of the works hitherto reported deal with synchronous and deterministic cells, and so does in what follows.

Several important concepts pertaining to the definitions of the systems are in order. First, *finiteness* vs. *infiniteness*. The distinction between finiteness and infiniteness is not for the number of states of a cell (i.e. finite-state or infinite-state of a cell) but is concerned with the number of cells existing in the space ready for operations. That is, whether the number of cells needed for a specified behavior is fixed finite at the beginning of the operation or infinitely many cells are assumed to be existing from the outset. When an infinite cell space is considered, however, it is common to devise a so called 'quiescent state' and assume that all but a finite number of cells are in the quiescent states. [8,35] A cell in the quiescent state stays the same state at the next time step if the states of the neighborhood cells are all quiescent.

Though the infinite cell space appears to yield some systematic treatments of the system, it should be restricted in some sense, e.g. as for computation time, as long as the capability of the whole system is concerned. For even a one-dimensional iterative automaton with infinite cells suffices to simulate a Turing machine.

In cases where a finite number of cells is considered, our interest lies more on a system of iterative automata obtained by collecting arbitrary but finite number of cells than on each of the iterative automata composed of some finitely fixed number of cells. Indeed, it is possible to embed an iterative automaton with finite number of cells into an infinite cell space. For example, an infinite cell space can be constructed by duplicating the state set of the original finite cell space and using the added states as boundary states of the finite part to be simulated.

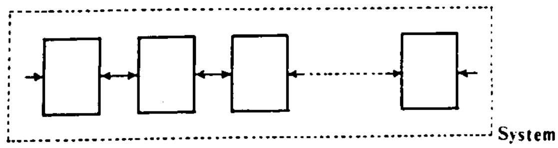
Generally speaking, the finite model seems to be in line with engineering motivation in that a cell may be added one by one, and the infinite model, on the other hand seems to be natural and mathematical owing to the a priori existence of an infinite space.

When systems with finite number of cells are considered, it is necessary to specify *boundary conditions* in order to complete the definitions of the systems. There may be various boundary conditions such as the fixed, the cyclic, the variable boundary conditions and so forth, whose detailed explanations will be given in the following chapter.

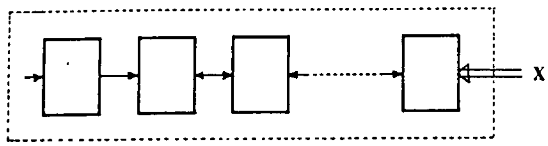
It is a little difficult a problem in general to define rigorously what is the real *dimension* of a cellular structure if a single neighborhood relation is not assumed to be applied to every cell. Thus, the homogeneity of the spacial locations and interconnections (i.e. neighborhood relations) of cells becomes a prerequisite to settle the problem. It seems natural to place cell automata at lattice points of an n-dimensional Euclidean space.

Next, let us consider the types of *input/output* of the iterative automata. It is of fundamental importance to note the dimension of the input(output) in connection with that of the whole system. The concept will be illustrated on the inputs to one-dimensional systems. For the output and higher-dimensional systems, similar considerations may also be applied.

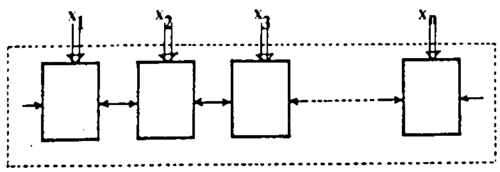
(i) autonomous(i.e. no input from outside of the system)



(ii) input dimension<system dimension



(iii) input dimension=system dimension



The closer the dimension of an input to that of the system, the more easily could the system be controlled from outside of it. Hennie treated mainly systems with input/output of type (iii) and we are going to treat that of type (ii). Cole's problem of language recognition [9] is set up on the n-dimensional system with 0-dimensional input. On the other hand, Fischer [12] constructed an autonomous one-dimensional system with 0-dimensional output which produces prime numbers in real time.

As to input/output, not only the dimension of it but also the location in the system and the amount of information to be injected or drawn out of the system play an important role in handling these systems.

A *neighborhood relation* is, as mentioned before, one of the most important features in the systems of iterative automata in that it specifies by which cells' information a cell's behavior is determined. It seems to be natural that no extra windings should be made between non adjacent cells. That is, any information should be transmitted through adjacent cells, which Neumann utilized in the famous two-dimensional iterative automata for self-reproduction [35]. We will also use the so called 'Neumann neighborhood' in this thesis. Codd [8] presented a simulation method of a system with fewer number of cell states but larger neighborhood. Hennie's unilateral and bilateral cells [16] are but another expressions for typical neighborhood relations.

Last but not least, cells may be designed for *single-purpose* or *multi-purpose*. There are several ways to make a cell operate for multi-purpose and some of them are listed below.

(i) To use variable cell structure.

a) Cutpoint cell [30]

b) Select an operation by the initial state

c) Select an operation by the primary input

(ii) To vary boundary inputs to fit for several purposes.

(iii) To change an output function or the location of an output cell.

(iv) Combinations of the above (i) to (iii).

In case (i), a whole system is often capable of interacting with the outside world (i.e. input dimension=system dimension), whereas in case (ii), the input dimension is apt to be less than the system dimension. To use a system for multi-purpose by changing an initial state ((i) b)) resembles in principle to write a program in order to do a particular job by a general purpose digital computer.

1.2. Classification of Problems with Brief Historical Notes

In this section, we will classify research problems around the theory of iterative automata into five categories and explain them with brief historical notes. They are, though not necessarily mutually exclusive, listed as follows;

(1) Construction Problem :

To construct a cell for a particular purpose.

(2) Decision Problem :

To present and answer, if possible, decision problems concerning cellular systems.

(3) Synthesis Problem :

To show a synthesis method of cellular system for a class of functions.

(4) Analysis Problem :

To analyze some characteristics of cellular systems.

(5) Characterization Problem :

To characterize structures and functions.

(1) Construction Problem : This type of problems can be stated as follows; Is there any iterative automaton which has some specified particular properties ? If the answer is in the affirmative, show it by constructing such an iterative automaton.

Thus, an answer to a construction problem is a kind of an existence proof in mathematics. One of the most outstanding results in this category is the existence of a self-reproducing (and at the same time computation and construction universal) two-dimensional cellular space shown by von Neumann. He constructed a particular cell with 29 states and planned to embed in his cellular space a self-reproducing automaton, which was completed and edited by Burks [35] and was neatly formulated by Thatcher [44]. Then, Codd [8] showed that an 8-state cell is sufficient for the purpose and Banks [3] reduced the number of states of a cell to four.

In this way, researches in this category tend to proceed from any existence proof by construction toward an optimal solution in some sense. Another typical example along this line is the firing squad synchronization problem posed by Myhill [32]. Goto was the first to give the minimal time solution but with a huge number of states and Waksman [47] and Balzer [2] endeavoured to seek a minimal state solution. Then the problem was generalized and some applications were suggested by Varshavsky [46].

The construction problems also exist in the applications of iterative systems to data processing fields. For example, Unger [45] constructed two-dimensional, bilateral iterative combinational switching circuits which detect some of the characteristics of two-dimensional patterns.

Fischer [12] constructed a one-dimensional sequential iterative automaton which generates in real time the binary sequence representing the set of prime numbers.

Construction problems are of value when the answer to the problems itself has any practical or theoretical importance or when the way to the solutions gives rise to such general methods, ideas or concepts that are useful for the studies of iterative systems.

(2) Decision Problem : As is well known [20], Turing machines could be embedded even in one-dimensional unilateral iterative systems. Accordingly, any decision problems concerning the computability of Turing machines may be translated into that for iterative systems. Of course we are concerned not with such kinds of decision problems but with ones which reflect some characteristics of iterative systems.

As to the systems composed of combinational cells with or without intercell delays, Hennie [16] posed several fundamental decision problems relating to the potentiality of the systems and solved them mostly not in the affirmative. That is, he demonstrated that general analysis procedures for the systems in question do not exist, and suggested that attention be concentrated on developing practical synthesis techniques for dealing with various important classes of problems. Kilmer [19] treated stability problems of the systems and solved some of the decision problems posed by Hennie.

There have been not so many works involved in decision problems other than cited above. In the following chapters, we propose some decision problems for the systems of one-dimensional iterative automata (e.g., as to an information transmission capability of systems, controllability of systems and so on).

When a decision problem turns out to be unsolvable, as are the most of the problems related to iterative systems, there remains a way to seek a less general algorithm of the problem for some restricted class of objects.

(3) Synthesis Problem : The term 'synthesis' used here means to show some general procedures for constructing a class of iterative systems. Though it is sometimes difficult to distinguish synthesis problems from construction problems, the former has preference on generality contrasted with speciality of the latter.

One of the most typical examples belonging to this category is the realization problem of logical functions by an iterative array of combinational cells. Maitra [29] considered cascaded switching networks of two-input flexible cells where flexible means that each cell may be adjusted at will to one of the sixteen possible functions of the primary and lateral inputs. Reduction of the number of available cell functions was carried out by Minnick [30], who also treated the problem for two-dimensional systems. Sklansky [40] showed a synthesis procedure realizing a given truth function when no a priori assignment of the variables to input terminals was specified. Maitra (single-rail) cascades were generalized to two-rail cascades of which Short [39] showed the logical completeness for the realization of an arbitrary Boolean function. Yoeli [50] devised group-theoretic approaches for the synthesis of multi-rail cascades. Works and results along this line are summarized and reviewed by Mukhopadhyay and Stone [33], and Elspas [11].

When the problem of realizing Boolean functions is to be solved by seeking a universal cellular logic, it may be classified in the construction problem.

(4) Analysis Problem : The problems in this category include analyses in general of the structure and function of iterative systems and those of the systems for some specified characteristics. There have been little works, except those by Yamada and Amoroso [48], which devote themselves to the analysis of iterative systems in their full generality. One of the reasons why it is so seems to be the

existence of the very difficulty that hinders general treatments of the systems.

On the other hand, there are several topics on which analyses of the systems have been conducted. For example, Бухштаб [6] investigated the real-time computation ability of functions by one-dimensional iterative automata, which was reviewed in Nishio [36].

There are problems concerning the language recognition by cellular automata. Because of the computation universality of the systems of iterative automata in general, most of the problems are set up on the basis of real-time recognition. Cole [9] showed that languages accepted in real-time by the systems of (n+1)-dimensional iterative automata properly include those accepted by the systems of n-dimensional iterative automata. Smith [41] discussed these problems with slightly different model of recognition to that of Cole, and several models of language recognition are compared on their recognition ability by Kasami and Fujii [17].

Several other themes which may deserve special attentions include analyses of the redundancy or stability of the systems, an information transmission through the systems, control of the systems from the outside world, reverberation phenomena occurring in the systems and so forth. Some of these will be treated in later chapters.

(5) Characterization Problem : Though it may be viewed as a part of analysis problems, 'characterization problem' here termed is intended to mean that some characteristics directly related to the global transition diagrams for the systems are treated.

By definition, the global transition diagram of an iterative automaton should completely characterize the behaviors of the automaton. Thus any problems of the systems of iterative automata have something to do with their transition diagrams. But we here focus our attention on the diagrams themselves and try to characterize them. As is well known [14], diagrams for linear automata have been characterized in terms of cycle set, tree height and so on by analyzing the defining matrices. It is difficult, however, to characterize similarly the diagrams for iterative automata mainly because of their non-linearity. And there are few works belonging to this category.

In chapter 3, we are going to characterize the sets of Garden of Eden configurations, Branching Point configurations, and so on in the systems of one-dimensional iterative automata with some appropriate boundary conditions.

1.3. Standpoint of This Research

In the above sections, the concepts and models of iterative automta have been explained with various research topics classified into five categories. Now, our aims and approaches in general follow.

In a usual switching and automata theory, it has been customary to treat the systems from a logical or functional point of view. Indeed there are many topics related with the implementations of the systems but little have concerned with the relations between a structure of the system (e.g., a spacial arrangement of the components, wire connections among the components, and so forth.) and its function or behavior as a whole. Our primary concern is to shed light upon these relations.

A system of iterative automata is chosen because it seems to be the simplest in structure. The uniformity of the structure may serve as a model of some natural framework.

In order to clarify the characteristics of iterative systems and to establish analysis and synthesis methods for them, we have kept in mind several methodological view points which will be outlined below.

As in any other theoretical problems, it seems to be of prime importance to establish how to express the characteristics to be pursued and also the problems themselves. In the studies of iterative automata, the functions of the component cell automaton, which is usually rather simple in structure and has a tractable number of states, may be completely described in many cases. On the other hand, it is sometimes too complex to recognize and express the behavior of the whole iterative systems composed of rather simple cells. Thus, considerable efforts should be paid on finding some general way of expressing the characteristics and behaviors in iterative systems.

In our opinion, some general methods if any may emerge out of solving specific problems which are expected to be characteristic to iterative systems. So, in the following chapters we are going to propose several specific problems not hitherto mentioned elsewhere and endeavor to solve them.

Though specific problems mentioned above may be chosen arbitrarily, we hope they are not such abstract ones that have little connections with real objects like living things and electronic computers etc.. This is one of the points we have in mind when setting up problems in the systems of iterative automata.

A feature of our approach, which seems to distinguish ours from those of the other authors, stems from the fact that we are primarily concerned with a system of iterative automata that is a collection of an iterative automaton of length n where n is an arbitrary positive integer. Thus, our main concern lies on the properties of the whole system rather than that of each iterative automaton of some fixed length n.

Last but not least, the underlying principle of the iterative systems under consideration seems to be that a highly structured aggregate of relatively simple and identical components (cells) does accomplish rather complex tasks by iterations, i.e., some kind of repetition of simple operations eventually results in rather complex outcomes.

CHAPTER 2

DEFINITIONS OF THE SYSTEMS

In this chapter the systems of iterative automata to be treated later are defined rigorously, and some of the useful concepts and notations are introduced. As is mentioned earlier, the researches reported here are mainly concerned with one-dimensional three-neighbor systems, and hence the definitions are so restricted, though easy to make them include broader classes, as to fit for the purposes.

2.1. Cell Automaton and its Transition Function

Let (S,f) denote a cell automaton where S is a finite nonempty set (of internal states) and f is a mapping (transition function) from $S \times S \times S$ into S. (S,f) has the right and the left inputs both of them taken from the set S and behaves as follows: if the state of a cell at a time is β and the left and the right inputs are α and γ respectively, the next state β' of the cell is defined by $\beta'=f(\alpha,\beta,\gamma)$, where $\alpha,\beta,\gamma,\beta' \in S$.

The cell transition function f is sometimes given in the form of tables as shown in Fig. 2.1. Let the cardinality of S equals m.

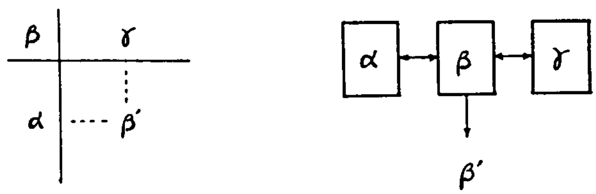


Fig. 2.1. A transition table

Each table is an m×m matrix, and there are m such tables for a fixed f.

Special classes of the transition functions are given special names, some of which will play an important role in later chapters.

RP type : (S,f) is said to be of Right Permutation type (RP type) if $f(\alpha,\beta,\gamma)\neq f(\alpha,\beta,\gamma')$ for any α,β,γ, and γ' in S such that $\gamma\neq\gamma'$. Let us regard $f(\alpha,\beta,\gamma)$ for fixed α and β as a function of γ and denote it as $f_{\alpha\beta}(\gamma)$. If (S,f) is of RP type, then $f_{\alpha\beta}(\gamma)$ is a permutation function on S for fixed α and β.

Similarly, we have the following two types of permutations.

LP type : (S,f) is said to be of Left Permutation type (LP type) if $f(\alpha,\beta,\gamma)\neq f(\alpha',\beta,\gamma)$ for any α,α',β, and γ in S such that $\alpha\neq\alpha'$.

CP type : (S,f) is said to be of Center Permutation type (CP type) if $f(\alpha,\beta,\gamma)\neq f(\alpha,\beta',\gamma)$ for any α,β,β',and γ in S such that $\beta\neq\beta'$.

As the other extreme to the above mentioned permutation type functions, it is sometimes useful to distinguish the following unilateral cases.

LU type : (S,f) is said to be of Left Unilateral type (LU type) if $f(\alpha,\beta,\gamma)=f(\alpha,\beta,\gamma')$ for any α,β,γ, and γ' in S. (S,f) is of LU type iff* the cell automaton is insensitive at all to its right inputs.

RU type : (S,f) is said to be of Right Unilateral type (RU type) if $f(\alpha,\beta,\gamma)=f(\alpha',\beta,\gamma)$ for any α,α',β, and γ in S.

2.2. Iterative Automaton and Boundary Conditions

Copies of a cell automaton (S,f) are assumed to be located at lattice points of a one-dimensional Euclidean space. Each copy of (S,f) will be referred by its position and denoted as C_i where i is the coordinate.

---

* The notation 'iff' will be used throughout as an abbreviation of 'if and only if.'

As shown in Fig. 2.2., an iterative automaton of length n is a collection of n cells located at from 1 to n, where n is a positive integer.

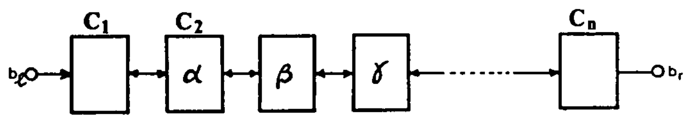


Fig. 2.2. An iterative automaton of length n

The left and right inputs of a cell are assumed to be the states of the left and right neighboring cells respectively.

Let b_l and $b_r (b_l, b_r \in S)$ be boundary input signals to the leftmost cell C_1 and the rightmost one C_n respectively. There may be various kinds of boundary conditions according to the ways how to assign input signals b_l and b_r. Some reasonable boundary conditions are listed below : (See Fig. 2.3.)

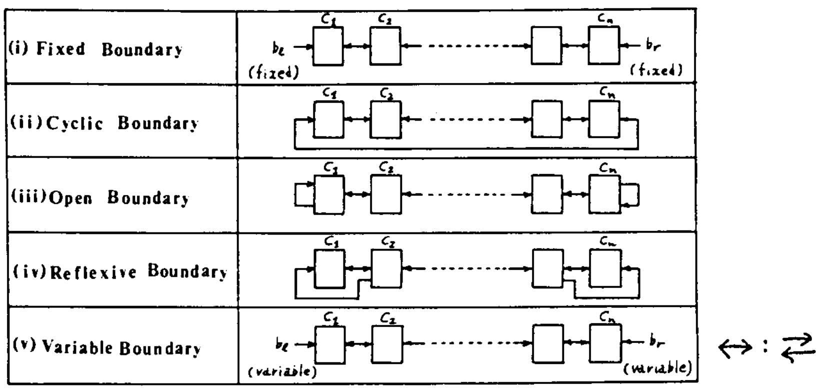


Fig. 2.3. Some boundary conditions

i) *Fixed Boundary Condition*

In this case the input signals b_l and b_r are constant. Behavior of an iterative automaton generally depends rather heavily on the constants chosen as b_l and b_r.

ii) *Cyclic Boundary Condition*

In this case both boundary cells C_1 and C_n are connected side by side, thus making the whole system constitute a "ring."

iii) *Open Boundary Condition*

When b_l equals the present state of C_1 and b_r equals that of C_n, the boundary condition is called open.

iv) *Reflexive Boundary Condition*

The case where b_l equals the present state of C_2 and b_r equals that of C_{n-1} is called reflexive boundary condition. Thus there should be more than one cell in the system when we treat reflexive boundary condition.

v) *Variable Boundary Condition*

It is the case where the boundary input signals b_r and b_l are given at will from outside the system. That is,the iterative automaton has inputs from outside in the usual sense. Contrary to this, the cases i) to iv) make the system autonomous.

It may be possible to consider other boundary conditions than mentioned above, especially the cases(called mixed boundary condition) where the right and the left boundary conditions are of different type. For example, the case where b_l is fixed and b_r is variable will be treated in later chapters.

2.3. System of One-Dimensional Iterative Automata

Let $A_x(S,f,n)$ denote an iterative automaton of length n, with boundary condition x. It is abbreviated simply as $A_x(n)$ when S and f are obvious. In the place of the generic symbol x, we write(b_l,b_r), c,o and r for the fixed, cyclic, open and reflexive boundary condition respectively. Some appropriate symbols will be used for mixed boundary conditions.

Given S and f, $A_x=\{A_x(n)|n=1,2,3...\}$ will be called a *system of one-dimensional iterative automata with boundary condition x.* As is mentioned earlier, we are concerned with A_x as a whole rather than with an individual automaton $A_x(n)$.

The arrangement of the states of n cells in an array at time t is called a *state configuration* (of length n at time t). A typical state configuration of length n is representd by an n-tuple $s_1s_2...s_n$ where $s_i \in S$ is the state of cell C_i. The state configuration $s_1s_2...s_n$ at time t changes into another state configuration $s'_1s'_2...s'_n$ at time t+1, where $s'_i=f(s_{i-1},s_i,s_{i+1})$ (i=2,3...,n-1), $s'_1=f(b_l,s_1,s_2)$ and $s'_n=f(s_{n-1},s_n,b_r)$. The last two equations represent the boundary condition of the system. The global transition function of state configurations under boundary condition x is denoted by F_x. F_x is a mapping from S^n into S^n, i.e. $F_x(s_1s_2...s_n)=s'_1s'_2...s'_n$.

As to the above mentioned mixed boundary condition, $F_x(s_1s_2...s_n)$ may be rewritten as $F_{b_l}(s_1s_2...s_n;b_r)$ especially when b_r is variable.

Given $A_x(S,f,n)$ and an initial state configuration $s_1s_2...s_n$, the state sequence of some specified cell(s) will be denoted by $P_x(s_1s_2...s_n)$. In a fixed boundary condition, the output cell is assumed to be the leftmost cell C_1.

2.4. Homomorphism, Isomorphism, and Symmetry

In this section, some relations between the cell automata are introduced. Let (S_A,f_A) and (S_B,f_B) be cell automata defined in Sec. 2.1.

Definition 2.1.(*Homomorphism*) (S_A,f_A) is said to be *homomorphic* to (S_B,f_B), written $(S_A,f_A) \gtrapprox (S_B,f_B)$, iff there exists a mapping h from S_A onto S_B such that

$$h[f_A(\alpha,\beta,\gamma)]=f_B(h(\alpha),\ h(\beta),\ h(\gamma)\ )$$

holds for any α, β, and γ in S_A.

Definition 2.2. *(Isomorphism)* (S_A,f_A) and (S_B,f_B) are said to be *isomorphic* to each other iff $(S_A,f_A) \gtrapprox (S_B,f_B)$ and $(S_B,f_B) \gtrapprox (S_A,f_A)$.

Let S be an arbitrarily fixed finite set of states. Consider the set of all state transition functions from S×S×S into S, which is denoted by {f}. The above mentioned isomorphism defines an equivalence relation on {f}. In case S={0,1}, possible permutations are the identity and the cyclic permutation (0 1). Thus, let f_A be

$$\begin{array}{c|cc} 0 & 0 & 1 \\ \hline 0 & a & c \\ 1 & b & d, \end{array} \qquad \begin{array}{c|cc} 1 & 0 & 1 \\ \hline 0 & e & g \\ 1 & f & h, \end{array}$$

where a,b,c,d,e,f,and g are in {0,1}. Then (S,f_B) is isomorphic to (S,f_A) iff $f_B=f_A$ or f_B is

$$\begin{array}{c|cc} 0 & 0 & 1 \\ \hline 0 & \bar{h} & \bar{f} \\ 1 & \bar{g} & \bar{e}, \end{array} \qquad \begin{array}{c|cc} 1 & 0 & 1 \\ \hline 0 & \bar{d} & \bar{b} \\ 1 & \bar{c} & \bar{a}\,. \end{array} \qquad \text{(Note: } \bar{x} = 1 - x.\text{)}$$

Another equivalence relation on {f} is defined as follows.

Definition 2.3. (*Symmetry*) (S,f_A) and (S,f_B) are said to be *symmetric* to each other iff $f_A(\alpha,\beta,\gamma)=f_B(\gamma,\beta,\alpha)$ for any α, β, and γ in S.

Let f_A be as before, f_B is symmetric to f_A iff f_B is .

$$\begin{array}{c|cc} 0 & 0 & 1 \\ \hline 0 & a & b \\ 1 & c & d, \end{array} \qquad \begin{array}{c|cc} 1 & 0 & 1 \\ \hline 0 & e & f \\ 1 & g & h. \end{array}$$

CHAPTER 3

SOME REGULAR STATE SETS IN THE SYSTEMS [24,25]

Consider $A_x=\{A_x(n)|n=1,2,3,......\}$, a system of one-dimensional iterative automata with boundary condition x. Since an iterative automaton $A_x(n)$, where $x=(b_l,b_r)$, c, o or r, operates autonomously, its transition diagram has $|S|^n$ nodes and is of the form, for example, like Fig. 3.1, where each node represents a state configuration of length n.

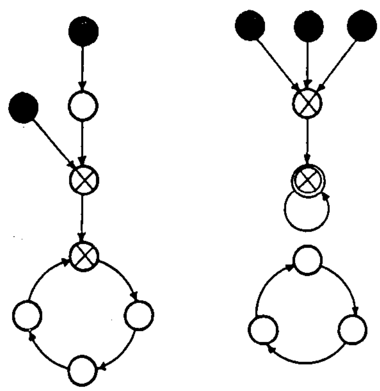


Fig. 3.1. An example of state transition diagram

In that figure, nodes marked by ●, ◎, and ⊗ are called *Garden-of-Eden configuration, passive configuration,* and *branching point configuration* respectively. In this chapter, we are going to characterize those configurations. That is, each set of those configurations in A_x for given S and f is shown to be regular. As is mentioned earlier, we are concerned with A_x as a whole rather than with an individual automaton $A_x(n)$.

Before going into the details, we introduce a nondeterministic finite state automaton associated with a cell (S,f), which will be utilized throughout this chapter in establishing theorems.

3.1. N-Automaton

A nondeterministic finite state automaton (abbreviated non-deterministic fsa) is a 5-tuple $A=(Q,\Sigma,\delta,S_o,F)$, where Q is a finite set (of states), Σ is a finite set (of inputs), S_o and F are designated subsets of Q(sets of initial and final states respectively) and δ is a mapping from $Q\times\Sigma$ into 2^Q.

Given S and f, we associate the cell structure with a special incompletely-specified non-deterministic fsa as follows.

Let $Q=S^2$ and $\Sigma=S$. For any pair $(\alpha,\beta)\in Q$ and $\beta'\in\Sigma$, δ is defined by $\delta((\alpha,\beta),\beta')=\{(\beta,\gamma)\mid(\beta,\gamma)\in Q,\ f(\alpha,\beta,\gamma)=\beta'\}$.
If the right hand side of the above equality is the empty set, input β' can't be applied to state (α,β). The automaton will be referred to as an N-automaton and denoted by N(S,f) or $N(S,f,S_o,F)$. Though the construction has not yet been completed in that S_o and F are not specified, it may be completed when the machine is utilized in later sections.

In fact, S_o and F will be determined according to a boundary condition x of A_x. For example, when we treat a fixed boundary condition, $S_o=\{(b_l,\alpha)\mid(b_l,\alpha)\in Q\}$ and $F=\{(\alpha,b_r)\mid(\alpha,b_r)\in Q\}$.
Thus, a sequence $\beta_1\beta_2\ldots\beta_n\in\Sigma^*$ may be accepted by the N-automaton if there exists a sequence of states $(\alpha_0,\alpha_1)(\alpha_1,\alpha_2)\ldots(\alpha_n,\alpha_{n+1})$, where $\alpha_0=b_l$, $\alpha_{n+1}=b_r$ and $\delta((\alpha_{i-1},\alpha_i),\beta_i)\ni(\alpha_i,\alpha_{i+1})$, $(i=1,2,\ldots n)$.
In such a case, notice that $F_{(b_l,b_r)}(\alpha_1\alpha_2\ldots\alpha_n)=\beta_1\beta_2\ldots\beta_n$ holds in $A_{(b_l,b_r)}(n)$.

Σ^* is the set of all finite or null sequences on Σ.

3.2. Set of Passive Configurations

A state configuration $s_1s_2\ldots s_n$ of $A_x(n)$ is called passive if its next state configuration equals itself; i.e. $F_x(s_1s_2\ldots s_n)=s_1s_2\ldots s_n$.
Given S and f, let $P(A_x)$ denote the set of all passive state configurations of A_x, i.e. $P(A_x)=\bigcup_{n=1}^{\infty}P(A_x(n))$, where $P(A_x(n))$ is the set of all passive configurations of $A_x(n)$. Then the following theorems can

be proved.

3.2.1. Fixed Boundary Condition

Theorem 3.1.

$P(A_{(b_l,b_r)})$ is a regular set over the alphabet S.

Proof

Consider the N-automaton $N(S,f,S_o,F)$, and branches leaving state (α,β) and entering state (β,γ) with labels β for every (α,β) and (β,γ) in Q. Such branches are called p-branches. Delete from the N-automaton all branches but for p-branches, the resulting automaton (isolated states are also deleted) is called an N_p-automaton and designated by $N_p(S,f,S_o,F)$. The state transition diagram of the N_p-automaton is composed of transitions like $(\alpha,\beta) \xrightarrow{\beta} (\beta,\gamma)$.

Now, if we let $S_o=\{(b_l,\alpha) \mid (b_l,\alpha)$ is a state of N_p-automaton$\}$ and $F=\{(\alpha,b_r) \mid (\alpha,b_r)$ is a state of N_p-automaton$\}$, then $N_p(S,f,S_o,F)$ accepts just the sequences contained in $P(A_{(b_l,b_r)})$. $P(A_{(b_l,b_r)})$ is empty when N_p-automaton does not exist. Thus we have established Theorem 3.1. (Q.E.D.)

The method of constructing $N_p(S,f,S_o,F)$ used for the proof of Theorem 3.1 is illustrated in Example 3.1.

Example 3.1.

Let $S=\{0,1\}$, $(b_l,b_r)=(1,1)$ and f is given as

$$\begin{array}{c|cc} 0 & 0 & 1 \\ \hline 0 & 1 & 0 \\ 1 & 0 & 1, \end{array} \qquad \begin{array}{c|cc} 1 & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 1 & 0 \end{array}.$$

The N_p-automaton under consideration has four states, (0,0), (0,1), (1,0), and (1,1), and $S_o=\{(1,1),(1,0)\}$, $F=\{(0,1),(1,1)\}$ as shown in Fig. 3.2.

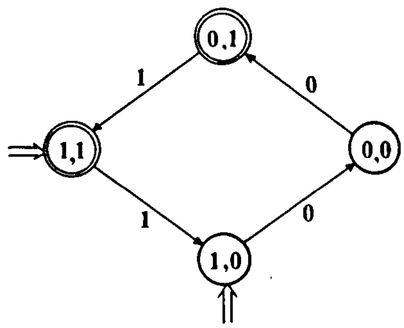


Fig. 3.2. N_p-automaton used in Example 3.1

The sequences accepted by the automaton are expressed in the form

$$P(A_{(1,1)})=(10\cup 0)0\ \{(1100)^*\cup 1(1001)^*\}.$$

Comment 3.1.

In an N-automaton, a branch is called α-branch if it has label α. For arbitrarily fixed $\alpha \in S$, construct an N_α-automaton composed of α-branches and corresponding states as was done in construction of an N_p-automaton.

Let $I_\alpha(A_{(b_l,b_r)})=\{s_1s_2\ldots s_n | F_{(b_l,b_r)}(s_1s_2\ldots s_n)=\alpha^n,\ n=1,2,\ldots\}$, i.e. $I_\alpha(A_{(b_l,b_r)})$ denotes the set of state configurations whose next state configuration equals α^n, where $\alpha^n=\overbrace{\alpha\alpha\ldots\alpha}^{n}$. It is easily seen that $I_\alpha(A_{(b_l,b_r)})$ is accepted by the above defined N_α and therefore is a regular set for arbitrary α of S.

Given any pair of $A_{(b_l,b_r)}(S,f)$ and $\alpha \in S$, there exists f' such that

$P(A_{(b_l,b_r)}(S,f))=I_\alpha(A_{(b_l,b_r)}(S,f'))$, and there exists also f" such that

$I_\alpha(A_{(b_l,b_r)}(S,f))=P(A_{(b_l,b_r)}(S,f''))$.

Comment 3.2.

When S,b_l, and b_r are given, let P_f and $P(A_{(b_l,b_r)},f)$ denote the set of p-branches and the set of passive state configurations, respectively, corresponding to a transition function f. Then it can be easily seen from the process of proving Theorem 3.1 that

$$P_f \supset P_{f'} \rightarrow P(A_{(b_l,b_r)},f) \supset P(A_{(b_l,b_r)},f').$$

Therefore a partial ordering $\geq$ may be defined for the set of the local transition functions by putting $f \geq f'$ if $P_f \supset P_{f'}$.

3.2.2. Cyclic Boundary Condition

Theorem 3.2.

$P(A_c)$ is a regular set and represented in the form $P(A_c) = \bigcup_i Q_i Q_i^*$, where Q_i designates some regular set and i is an element of some finite set.

Proof

Consider an N_p-automaton with $S_o = F = \{(\alpha,\beta)\}$, where (α,β) is any one of its states. The set of input sequences accepted by the automaton $N_p(S,f,S_o,F)$ may be represented in a form $Q_{(\alpha,\beta)}Q^*_{(\alpha,\beta)}$ where $Q_{(\alpha,\beta)}$ is the set of sequences which leave state (α,β) and return to the same state for the first time. It is obvious that $Q_{(\alpha,\beta)}$ is a regular set. Now $P(A_c) = \bigcup_{(\alpha,\beta)} Q_{(\alpha,\beta)}Q^*_{(\alpha,\beta)}$, where the set union exhausts all states of N_p-automaton. (Q.E.D.)

Example 3.2.

Assume S and f to be the same as those used in Example 3.1. The way of getting $P(A_c)$ is illustrated here. The N_p-automaton used for this purpose is shown in Fig. 3.3.

In this case, $Q_{(10)} = \{0011\}$, $Q_{(01)} = \{1100\}$, $Q_{(00)} = \{0110\}$, and $Q_{(11)} = \{1001\}$.
Thus,

$$P(A_c) = 0011(0011)^* \cup 0110(0110)^* \cup 1100(1100)^* \cup 1001(1001)^*.$$

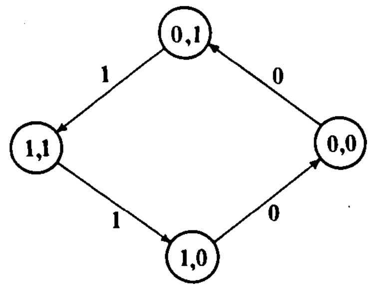


Fig. 3.3. N_p-automaton used in Example 3.2

Comment 3.3.

Under the cyclic boundary condition, it can be easily seen that if a state configuration $s_1 s_2 \ldots s_n$ belongs to $P(A_c)$, then the state configuration $s_n s_1 s_2 \ldots s_{n-1}$ also does, and vice versa.

Let R denote a binary relation defined in S* by $s_1 s_2 \ldots s_n$ R $s_n s_1 \ldots s_{n-1}$ and $\widetilde{R}$ denote the equivalence relation naturally derived from R. Using $\widetilde{R}$, the representation of $P(A_c)$ can be reduced to a simpler form.

In the case of Example 3.2, we can express, for example, $P(A_c)/\widetilde{R}=\{0011(0011)^*\}$. It seems rather difficult to obtain a simplified expression $P(A_c)/\widetilde{R}$ directly or without enumerating $P(A_c)$ itself.

3.2.3. Other Boundary Conditions

As to the systems with open or reflexive boundary conditions, similar reasoning as that of Sec.3.2.1. leads to the conclusions that $P(A_x)$ (x=o or r.) is also a regular set.

In the case of open boundary condition, let $S_o=F=\{(\alpha,\alpha)\mid(\alpha,\alpha)$ is a state of N_p-automaton$\}$, then $N_p(S,f,S_o,F)$ accepts $P(A_o)\cup\{\varepsilon\}$ where ε is a sequence of length 0.

In the case of reflexive boundary condition, let $F=\{(\beta,\alpha)\mid(\beta,\alpha)$ is a state of N_p-automaton and $f(\alpha,\beta,\alpha)=\beta\}$, and $S_o=\{(\beta,\alpha)\}\subset F$. The set of sequences accepted by $N_p(S,f,S_o,F)$ except for null sequence will be denoted by $R_{(\beta,\alpha)}$. $P(A_r)$ equals $\bigcup_{(\beta,\alpha)\in F} \beta R_{(\beta,\alpha)}$, which is composed of sequences whose lengths are more than one.

For the cell shown in Example 3.1, $P(A_x)$ (x=o or r.) can be written as

$P(A_o)=(1001 \cup 01)(1001)^* \cup (0110 \cup 10)(0110)^*$,

$P(A_r)=\phi$.

Furthermore the set of the passive state configurations for the system with a mixed boundary condition is also seen to be regular.

3.3. Set of Garden-of-Eden Configurations

A state configuration $s_1 s_2 \ldots s_n$ of $A_x(n)$ is said to precede a state configuration $s_1' s_2' \ldots s_n'$ if $F_x(s_1 s_2 \ldots s_n)=s_1' s_2' \ldots s_n'$.

A state configuration is called a *Garden-of-Eden configuration* if it has no preceding state configurations.

Let $E(A_x)$ denote the set of all Garden-of-Eden configurations in A_x where S and f are assumed to be understood.

3.3.1. Fixed Boundary Condition

Theorem 3.3.

$E(A_{(b_l,b_r)})$ is a regular set over the alphabet S.

Proof

It will suffice to show that $S^*-E(A_{(b_l,b_r)})=\overline{E(A_{(b_l,b_r)})}$ constitutes a regular set. $\overline{E(A_{(b_l,b_r)})}$ is the set of all state configurations that have preceding state configurations.

Consider N-automaton $N(S,f,S_o,F)$ with $S_o=\{(b_l,\alpha)\mid(b_l,\alpha)\in Q\}$ and $F=\{(\alpha,b_r)\mid(\alpha,b_r)\in Q\}$. As is noted in Sec. 3.1., a sequence $\beta_1\beta_2\ldots\beta_n$ is accepted by the N-automaton iff $F_{(b_l,b_r)}(\alpha_1\alpha_2\ldots\alpha_n)$

$=\beta_1\beta_2\cdots\beta_n$ for some $\alpha_1\alpha_2\cdots\alpha_n \in S^*$. Thus, $N(S,f,S_o,F)$ accepts just all the sequences that belong to $\overline{E(A_{(b_l,b_r)})}$. (Q.E.D.)

In order to construct an automaton which accepts $E(A_{(b_l,b_r)})$ directly, one should in the first place transform the non-deterministic fsa $N(S,f,S_o,F)$ into its equivalent deterministic fsa $N_d(S,f,S_o,F)$ (called N_d-automation) as usual. We add a new state μ (dummy state) whenever necessary to make N_d-automaton completely specified. Then an automaton which accepts $E(A_{(b_l,b_r)})$ is obtained by taking the complement of the existing final states of N_d-automaton.

The process is illustrated by

Example 3.3.

Let $S=\{0,1\}$, $(b_l,b_r)=(1,1)$, and f be

0	0	1
0	1	0
1	1	0,

1	0	1
0	0	1
1	0	0.

Figure 3.4 shows the corresponding N-automaton, which may be transformed into an N_d-automaton as shown in Fig. 3.5. Thus an automaton accepting $E(A_{(1,1)})$ can be obtained as in Fig. 3.6.

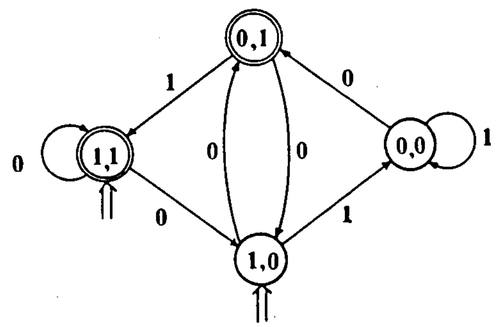


Fig. 3.4. N-automaton used in Example 3.3

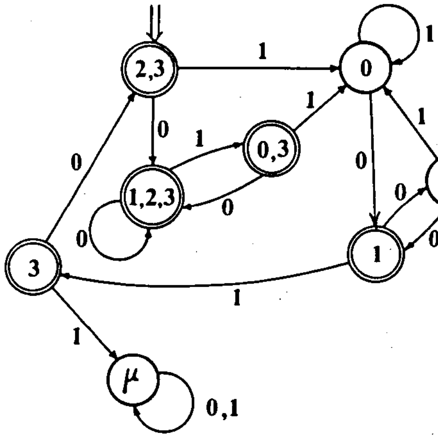


Fig. 3.5. N_d-automaton corresponding to the N-automaton in Fig.3.4

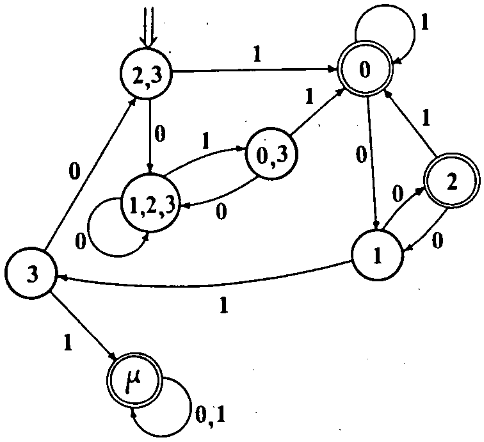


Fig. 3.6. Automaton used in Example 3.3

The set of the sequences accepted by the automaton is represented as follows.

$$E(A_{(1,1)})=[1\cup 0(0\cup 10)^*11][1\cup 0(00)^*(01\cup 101\cup 100(0\cup 10)^*11]^* \cdot[\varepsilon\cup 0(00)^*(0\cup 11(0\cup 1)^*].$$

Comment 3.4.

As is easily seen, if an N_d-automaton has the dummy state μ, the set of the final states of the automaton accepting $E(A_{(b_l,b_r)})$ includes the state μ. Thus, in general $E(A_{(b_l,b_r)})$ may be written in a form $R_1\cup R_2S^*$, where R_1 and R_2 are both regular sets and the former is dependent on the boundary condition b_r whereas the latter is not. Further, we have $\bigcap_{b_r \in S} E(A_{(b_l,b_r)}) = R_2S^*$.

Now, that there are no Garden-of-Eden configurations in $A_{(b_l,b_r)}$ means that the state transition diagrams of $A_{(b_l,b_r)}$ are composed only of cyclic parts. As is well known, it is decidable whether $\overline{E(A_{(b_l,b_r)})}$ equals S^*, or whether $E(A_{(b_l,b_r)})$ is empty, or not. So, as a corollary to Theorem3.3 we get

Theorem 3.4.

It is decidable whether every state transition diagram of $A_{(b_l,b_r)}$ is composed only of cyclic parts or not.

3.3.2. Cyclic Boundary Condition

Theorem 3.5.

$E(A_c)$ is a regular set and can be written in the form $E(A_c)= S^* - \bigcup_i Q_i Q_i^*$ where Q_i designates a regular set and i is an element of some finite set.

Proof

As in the case of Theorem3.3, $S^* - E(A_c) = \overline{E(A_c)}$ is shown to be regular. Consider an N-automaton and let $Q_{(\alpha,\beta)}$ denote the set of all input sequences that start from a state (α,β) in the N-automaton and return to the state for the first time. $Q_{(\alpha,\beta)}$ is, of course, a regular set. Then, as before,

$$\overline{E(A_c)} = \bigcup_{(\alpha,\beta)\in Q} Q_{(\alpha,\beta)} Q^*_{(\alpha,\beta)}$$

holds, establishing the theorem. (Q.E.D.)

Example 3. 4.

For cells defined by the same S and f as in Example3.3, $Q_{(\alpha,\beta)}$'s are given as follows;

$Q_{(0,1)}=(0\cup 10^*0)(0\cup 11^*0)$,
$Q_{(1,0)}=(0\cup 11^*0)(0\cup 10^*0)$,
$Q_{(1,1)}=0\cup 0(00\cup 11^*00)^*(0\cup 11^*0)1$,
$Q_{(0,0)}=1\cup 0(00\cup 10^*00)^*(0\cup 10^*0)1$.

As in the case of passive state configurations, it can also be said that if $s_1s_2\ldots s_n \in E(A_c)$, then $s_ns_1\ldots s_{n-1}\in E(A_c)$ and vice versa. Thus one may express $E(A_c)$ in a simpler form by dividing the original $E(A_c)$ by $\tilde{R}$ used in Sec.3.2.2.

The same reasoning as in the derivation of Theorem3.4 leads us to

Theorem 3.6.

It is decidable whether every state transition diagram of A_c is composed only of cyclic parts or not.

3.3.3. Other Boundary Conditions

$E(A_x)$ is shown to constitute a regular set under the open or the reflexive boundary condition and even under mixed boundary conditions. Proofs can be carried out much the same as in the proof of Theorem3.3. Alterations should be made on the definitions S_o and F of the N-automaton, as we have done for passive state configurations in Sec.3.2.3.

Similar theorems as Theorems3.4 and 3.6 hold under the above mentioned several boundary conditions.

3.4. Branching Point Configurations

A state configuration $s_1s_2\ldots s_n$ of $A_x(n)$ is called a *branching point configuration* if it has more than one preceding state configuration. Let $B(A_x)$ denote the set of all branching point configurations of A_x. In what follows, though only the fixed boundary condition is treated, similar considerations can be applied to the other boundary conditions.

Theorem 3.7.

$B(A_{(b_l,b_r)})$ is a regular set over the alphabet S.

Proof

In proving Theorem3.3, we have constructed an automaton which acceptes $\overline{E(A_{(b_l,b_r)})}$, where $\overline{E(A_{(b_l,b_r)})}$ means the set of all state configurations having preceding state configurations. Therefore, the branching point configuration is a configuration accepted by this automaton via more than one path.

Consider the following nondeterministic fsa $N_b(Q,\Sigma,\delta,S_o,F)$.

State set Q consists of the following;

(1) $[(\alpha,\beta)\ (\alpha',\beta')]$:

The order of (α,β) and (α',β') is immaterial, i.e. $[(\alpha,\beta)\ (\alpha',\beta')]$equals $[(\alpha',\beta')\ (\alpha,\beta)]$. Elements (α,β) and (α',β') may be identical.

The states in (1) are meant to designate the cases, where each sequence that leads to a state of this type has passed more than one distinct state pair of the N-automaton.

(2) (α,β):

These states are used to indicate that each sequence leading to one of these states intrinsically passes a single route in the N-automaton.

(3) σ:

This is the initial state. I.e., $S_o=\{\sigma\}$.

$\Sigma=S$ and δ is defined as follows;

(1) $\delta([(\beta,\gamma)\ (\beta',\gamma')],\alpha)=[(\gamma,\eta)\ (\gamma',\eta')]$ when $\alpha=f(\beta,\gamma,\eta)$ $=f(\beta',\gamma',\eta')$. If $(\beta,\gamma)=(\beta',\gamma')$ then $\eta=\eta'$.

(2) $\delta((\beta,\gamma),\alpha)=(\gamma,\eta)$ when $\alpha=f(\beta,\gamma,\eta)$.

$\delta((\beta,\gamma),\alpha)=[(\gamma,\eta)\ (\gamma,\eta')]$ if $\alpha=f(\beta,\gamma,\eta)=f(\beta,\gamma,\eta')$ and $\eta\neq\eta'$.

(3) $\delta(\sigma,\alpha)=(\beta,\gamma)$ when $\alpha=f(b_l,\beta,\gamma)$ and $\delta(\sigma,\alpha)=[(\beta,\gamma)\ (\beta',\gamma')]$ if $\alpha=f(b_l,\beta,\gamma)=f(b_l,\beta',\gamma')$ and $(\beta,\gamma)\neq(\beta',\gamma')$.

Function δ is not defined for the arguments not listed above and not defined also for the case where right hand side of δ is empty.

The set of accepting states is composed of $[(\alpha,\beta)\ (\alpha',\beta')]$'s where (α,β) and (α',β') are both accepting states of the automaton which accepts $\overline{E(A_{(b_l,b_r)})}$. I.e., $F=\{[(\alpha,b_r)\ (\beta,b_r)]\}$.

$B(A_{(b_l,b_r)})$ is the set of sequences accepted by the above mentioned N_b-automaton. (Q.E.D.)

Example 3.5.

An N_b-automaton which accepts $B(A_{(1,1)})$ for the same cell as in Example3.3 is shown in Fig3.7. In order to simplify the diagram, state (α,β) is designated by a decimal number corresponding to a binary number $\alpha\beta$.

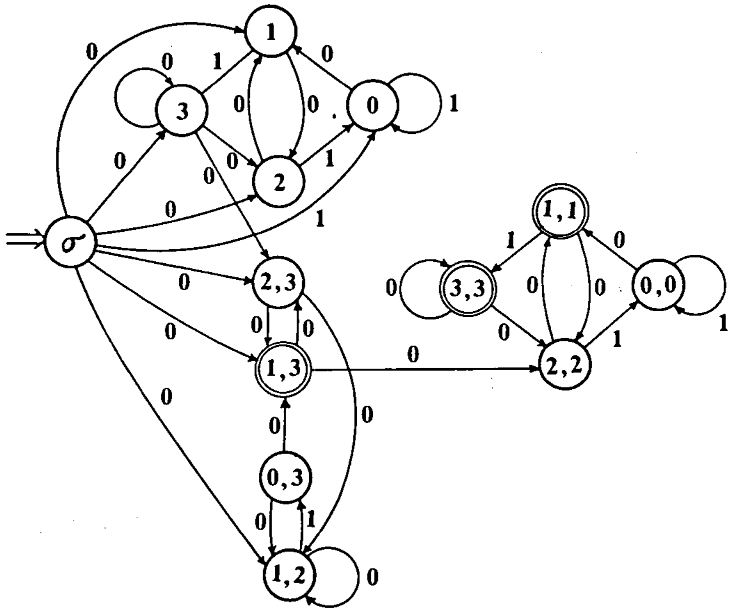


Fig. 3.7. N_b-automaton used in Example 3.5

If we consider only, for example, the sequences of length 4, 0000, 0010, 0001, and 0100 are the branching point configurations.

It can also be verified that $B(A_x)$ (x=c, o or r) is a regular set.

3.5. Iterative Array Viewed as Special Case of GSM Mapping

We can consider our system A_x as a special case of gsm (generalized sequential machine) mapping. Using established results on the gsm mapping, some theorems of Sections 3.2 and 3.3 are proved here again.

The behavior of a one-dimensional iterative array is simulated by a gsm as roughly illustrated in Fig. 3.8.

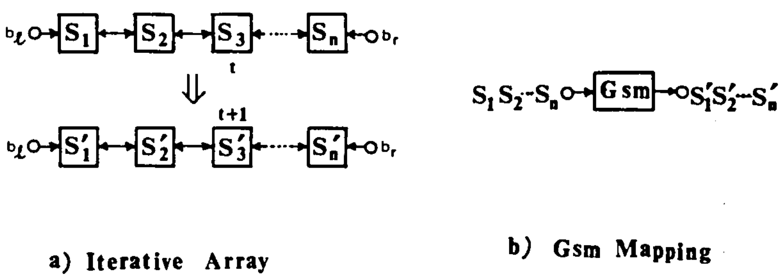


Fig. 3.8. Simulation of one-dimensional iterative array by a gsm

In words, the state configuration $s_1s_2 \ldots s_n$ is interpreted as the input string of length n to a gsm, whose output is $s_1's_2' \ldots s_n'$. To make it precise and settle problems depending on each boundary condition, we need some elaboration.

A gsm (mapping) is formulated generally as follows:

$g=(Q,\Sigma,\Delta,\delta,\lambda, q_0)$, where

Q is a finite set of states,

Σ is a finite alphabet of input symbols,

Δ is a finite alphabet of output symbols,

δ is a mapping $Q\times\Sigma\to Q$ (the transition function),

λ is a mapping $Q\times\Sigma\to\Delta^*$(the output function)

and q_o is an element of Q, the initial state.

To apply this formulation to our problem, let

(3.1) $\Sigma=\Delta=S\cup\{E\}$,

where E is the distinguished symbol denoting the end marker, which does not belong to S.

3.5.1. GSM Mapping for Fixed Boundary Condition

$$(3.2)\qquad Q=\{q_o\}\cup\{q_\alpha|\alpha\in S\}\cup\{q_{\alpha\beta}|\alpha,\beta\in S\}.$$

$$(3.3)\quad\begin{cases}\delta(q_o,\alpha)=q_\alpha, & \alpha\in S;\\ \delta(q_\alpha,\beta)=q_{\alpha\beta}, & \alpha,\beta\in S;\\ \delta(q_{\alpha\beta},\gamma)=q_{\beta\gamma}, & \alpha,\beta,\gamma\in S;\\ \delta(q,E)=q_o, & q\in Q.\end{cases}$$

$$(3.4)\quad\begin{cases}\lambda(q_o,\alpha)=\varepsilon, & \alpha\in S;\\ \lambda(q_\alpha,\beta)=f(b_l,\alpha,\beta), & \alpha,\beta\in S;\\ \lambda(q_{\alpha\beta},\gamma)=f(\alpha,\beta,\gamma), & \alpha,\beta,\gamma\in S;\\ \lambda(q_\alpha,E)=f(b_l,\alpha,b_r)E, & \alpha\in S;\\ \lambda(q_{\alpha\beta},E)=f(\alpha,\beta,b_r)E, & \alpha,\beta\in S;\\ \lambda(q_o,E)=\varepsilon.\end{cases}$$

where b_l and b_r are the left and the right boundary conditions respectively.

Equations (3.1) - (3.4) define our gsm mapping $g:\Sigma^*\to\Sigma^*$, where $\Sigma=S\cup\{E\}$ is the common alphabet of the domain and the range and $\delta(q,\varepsilon)=q$ and $\lambda(q,\varepsilon)=\varepsilon$ for every q in Q. Figure3.9 shows our gsm for the iterative array of Example3.3 with fixed boundary condition (1,1).

In using this mapping, we restrict the domain to the set S*E or to the input strings having one and only one E at their ends.

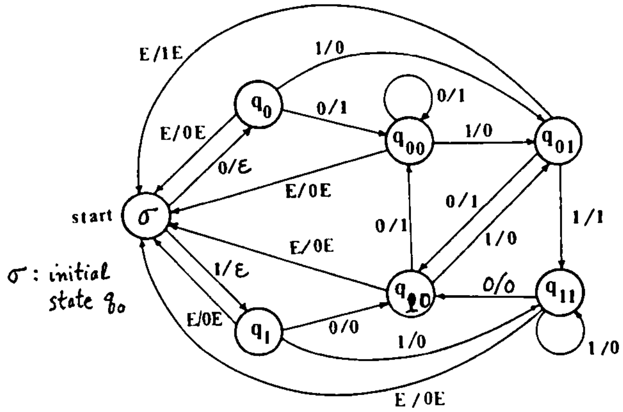


Fig. 3.9. Gsm for iterative array of Example 3.3 with x=(1,1)

Then from Eqs.(3.3) and (3.4) the set of outputs for inputs S*E is a subset of S*E, or $g(S^*E)\subset S^*E$. Since each gsm mapping preserves regular sets in general, g(S*E) is a regular set.

From Lemma 3.1 below, which we do not prove here, it is seen that $G=\{x|xE\in g(S^*E), x\in S^*\}$ is a regular set.

Lemma 3.1.

Let A be a subset of S* and E be a symbol not belonging to S. Denote $A_E=\{x|x=yE,\ y\in A\}$. Then, if A_E is a regular set over $S\cup\{E\}$, A is regular over S.

Now it is easily seen that $S^*-G=E(A_{(b_l,b_r)})$, where S*-G means the set of all strings of S* which do not belong to G. Since the complement of a regular set is also a regular set, another proof of Theorem3.3 has been obtained.

Secondly, let $R_\alpha=\alpha^*$ for any α of S. Since each inverse gsm mapping preserves regular sets, $g^{-1}(R_\alpha E)$ is regular. On the other hand, it is seen from the definition of I_α in Comment3.1 of Sec.3.2 that

$$I_\alpha(A_{(b_l,b_r)})=\{x|xE\in g^{-1}(R_\alpha E)\}.$$

With Lemma 3.1, this gives another proof of the regularity of I_α.

By Comment3.1 on the relationship between $I_\alpha(A_{(b_l,b_r)})$ and $P(A_{(b_l,b_r)})$, Theorem3.1 is proved again.

Comment 3.5.

Though general results on the gsm mapping were thus useful in proving the regularity of a set of state configurations, they alone are not sufficient for the study of iterative arrays. It is a special case, where special treatments are required. An N-automaton has been devised for this study and shown to be more powerful than the gsm in obtaining the concrete forms of the above mentioned regular sets. In higher dimensional cases, the ordinary language theory is obviously not of use.

3.5.2. GSM Mapping for Cyclic Boundary Condition

$$(3.5)\quad Q=\{q_o\}\cup\{q_\alpha|\alpha\in S\}\cup\{q_{\alpha\beta}|\alpha,\beta\in S\}\cup\{q_{\alpha'\beta',\alpha\beta}|\alpha',\beta',\alpha,\beta\in S\}.$$

$$(3.6)\begin{cases}\delta(q_o,\alpha)=q_\alpha, & \alpha\in S;\\ \delta(q_\alpha,\beta)=q_{\alpha\beta}, & \alpha,\beta\in S;\\ \delta(q_{\alpha\beta},\gamma)=q_{\alpha\beta,\beta\gamma}, & \alpha,\beta,\gamma\in S;\\ \delta(q_{\alpha'\beta',\alpha\beta},\gamma)=q_{\alpha'\beta',\beta\gamma}, & \alpha,\alpha',\beta,\beta',\gamma\in S;\\ \delta(q,E)=q_o, & q\in Q;\\ \delta(q,\varepsilon)=q, & q\in Q.\end{cases}$$

$$(3.7)\begin{cases}\lambda(q_o,\alpha)=\varepsilon, & \alpha\in S;\\ \lambda(q_\alpha,\beta)=\varepsilon, & \alpha,\beta\in S;\\ \lambda(q_{\alpha\beta},\gamma)=f(\alpha,\beta,\gamma), & \alpha,\beta,\gamma\in S;\\ \lambda(q_{\alpha'\beta',\alpha\beta},\gamma)=f(\alpha,\beta,\gamma), & \alpha,\alpha',\beta,\beta',\gamma\in S;\\ \lambda(q_{\alpha\beta},E)=f(\alpha,\beta,\alpha)f(\beta,\alpha,\beta)E, & \alpha,\beta\in S;\\ \lambda(q_{\alpha'\beta',\alpha\beta},E)=f(\alpha,\beta,\alpha')f(\beta,\alpha',\beta')E, & \alpha,\alpha',\beta,\beta'\in S;\\ \lambda(q_\alpha,E)=f(\alpha,\alpha,\alpha)E, & \alpha\in S;\\ \lambda(q_o,E)=\varepsilon. & \end{cases}$$

If $F_c(s_1s_2...s_n)=s_1's_2'...s_n'$, then from Eqs.(3.1) and (3.5)-(3.7) $g(s_1s_2...s_nE)=s_2's_3'..s_n's_1'E$.

We define the cyclic shift operation t for each nonnull element of S* as follows: $t(s_1s_2...s_n)=s_2s_3..s_ns_1$,

$$t^{-1}(s_1s_2...s_n)=s_ns_1...s_{n-1}.$$

Lemma 3.2.

The cyclic shift operation t and t^{-1} preserve regular sets. That is, let A be a subset of S* and denote $t(A)=\{x|x=t(y),y\in A\}$ and $t^{-1}(A)=\{x|y=t(x),y\in A\}$. If R is a regular set, then t(R) and $t^{-1}(R)$ are regular sets.

This lemma is proved by constructing a sequential transducer, which realizes the cyclic shift operation using an end marker.

Now $E(A_c)=S^*-t^{-1}(\{x|xE\in g(S^*E)\})$. By Lemmas 3.1 and 3.2, $E(A_c)$ is regular, which is another proof of Theorem3.4.

Secondly, Theorem3.2 is proved again by $I_\alpha(A_c)=\{x|xE\in g^{-1}(R_\alpha E)\}$. Note that $t(R_\alpha)=R_\alpha$.

3.5.3. GSM Mapping for Open and Reflexive Boundary Conditions

Regularity of passive and Garden-of-Eden configurations for the open and the reflexive boundary conditions can be proved by the following equations through similar reasoning.

Open Boundary

Q and δ are defined by Eqs.(3.2) and (3.3) respectively.

$$(3.8)\quad \begin{cases} \lambda(q_0,\alpha)=\varepsilon, & \alpha\in S; \\ \lambda(q_\alpha,\beta)=f(\alpha,\alpha,\beta), & \alpha,\beta\in S; \\ \lambda(q_{\alpha\beta},\gamma)=f(\alpha,\beta,\gamma), & \alpha,\beta,\gamma\in S; \\ \lambda(q_\alpha,E)=f(\alpha,\alpha,\alpha)E, & \alpha\in S; \\ \lambda(q_{\alpha\beta},E)=f(\alpha,\beta,\beta)E, & \alpha,\beta\in S. \end{cases}$$

Reflexive Boundary

Q and δ are defined by Eqs.(3.2) and (3.3) respectively.

$$(3.9)\quad \begin{cases} \lambda(q_o,\alpha)=\varepsilon, & \alpha\in S; \\ \lambda(q_\alpha,\beta)=f(\beta,\alpha,\beta), & \alpha,\beta\in S; \\ \lambda(q_{\alpha\beta},\gamma)=f(\alpha,\beta,\gamma), & \alpha,\beta,\gamma\in S; \\ \lambda(q_{\alpha\beta},E)=f(\alpha,\beta,\alpha)E, & \alpha,\beta\in S. \end{cases}$$

Note that the gsm mapping for this case is defined for the input string of length more than one.

3.6. Concluding Remarks

It has been shown that each set of all passive, Garden-of-Eden and branching point state configurations constitutes a regular set and this regularity holds for various kinds of boundary conditions.

In order to characterize the autonomous behavior of A_x in more detail, we should investigate the cyclic and the transient part of the transition diagram. Let $C(A_x)$ and $T(A_x)$ denote the set of all state configurations that belong to the cyclic and the transient part respectively. ($C(A_x)\cup T(A_x)=S^*$ and $C(A_x)\cap T(A_x)=\phi$.) Then $E(A_x)$ and $P(A_x)$, which have been proved to be regular, are special subsets of $T(A_x)$ and $C(A_x)$ respectively. To determine the language classes of $C(A_x)$, $T(A_x)$, and $C(A_x)\cap B(A_x)$ is a challenging problem but is still unsettled. Mr. H.Takahashi of our group suggested that $C(A_x)$ equals essentially the "maximal invariant set of the gsm mapping", or the largest set $M\subset S^*$ such that $g(M)=M$ and found a nonregular $C(A_x)$ for a unilateral fixed boundary array, which can be simulated by a complete finite automaton instead of a gsm. That is, $C(A_x)$ and $T(A_x)$ are not always regular sets. About the periods of $C(A_x)$, little has been known.

While we have restricted ourselves to the one-dimensional case, our problem to characterize the sets of state configurations may be naturally extended to higher dimensional cases. In fact, the passive and the Garden-of-Eden configurations were treated on the two-dimensional cellular automata.[8,31] The main difficulty is to obtain the notion of the *regular set* for the higher dimensional case. In connection with the picture processing algorithm, several grammatical methods have been developed by many researchers.[7,13] We are interested in such a grammatical definition, if any, of the *regular-like* set for the two-dimensional configurations that makes regular the above mentioned state sets.

CHAPTER 4

INFORMATION TRANSMISSION CAPABILITY OF THE SYSTEMS [21,26,27]

4.1. Introduction

In a system of iterative automata, a cell communicates only with its neighborhood cells which are usually located at the nearest neighbor to the cell in question. As such, it is important as well as interesting to ask how far a system of iterative automata could transmit information through the system via local communications defined by the neighborhood relation and a local transition function. Right to left information transmission through a system of one-dimensional iterative automata is treated in this chapter.

In the beginning,there are two points that must be clarified in considering the problems of information transmission: the first is how to represent "information" itself and the second is to clarify the meaning of "transmission" of information. In our problem, "information" is given as an initial state setting to the rightmost part of cells, and "transmission" of (particular) information to the leftmost cell (called an output cell) means that one could detect any difference of initial state settings to the rightmost part of cells (i.e. information) by observing state transition sequences of the output cell.

A classification of the systems of one-dimensional iterative automata with respect to the above mentioned information transmission capability is proposed. We divide the set of all systems into three classes, reveal the characteristics of each class, and show several lemmas for the classification. Further, a related undecidability result is also shown.

As in before, we are interested not in an automaton composed of a fixed number of cells, but rather in a system of such automata that is obtained by varying the number of cells.

When a state configuration $\alpha=\alpha_1\alpha_2\ldots\alpha_n(\alpha_i\in S,\ i=1,2,\ldots n)$ is concerned, it is assumed that there exists an iterative automaton of length n and that a state of cell C_i corresponds to α_i. Given a state configuration α, let $P(\alpha)$ denote a time sequence of states of the leftmost cell (C_1) which is considered to be an output cell, when a system in question operates autonomously starting with initial configuration α. $F_{b_l,b_r}(\alpha)=\alpha'$ means that a state configuration α becomes α', where $\alpha'=\alpha_1'\alpha_2'\ldots\alpha_n'(\alpha_i'\in S\ i=1,2,\ldots n)$, at the next time step under boundary condition (b_l,b_r). In such situation, we can write $P(\alpha)=\alpha_1P(\alpha')$ by the definition of $P(\alpha)$.

Now, let us consider the following classification of the set of all systems of one-dimensional iterative automata $\{A_{b_l,b_r}(S,f)\}$ according to their (right to left) information transmission capability.

Class I (C_I)

$A_{b_l,b_r}(S,f)$ belongs to class I iff there exists a positive integer k such that for $^\forall\alpha_i\in S(i=1,2,\ldots,k)$ and $^\forall\beta\in S^*$, $P(\alpha_1\alpha_2\ldots\alpha_k)$ $=P(\alpha_1\alpha_2\ldots\alpha_k\beta)$ (S^* is a free monoid, S being a generator).

When a system belongs to Class I, the minimum of k satisfying the above relations will be referred to as k-value of the system.

Class II (C_{II})

$A_{b_l,b_r}(S,f)$ belongs to Class II iff it does not belong to C_I and there exist a positive integer k and $\alpha_i\in S$ $(i=1,2,\ldots k)$ such that

$$P(\alpha_1\alpha_2\ldots\alpha_k)=P(\alpha_1\alpha_2\ldots\alpha_k\beta)\text{ for }^\forall\beta\in S^*.$$

Class III (C_{III})

$A_{b_l,b_r}(S,f)$ belongs to Class III iff for any positive integer k and $^\forall\alpha_i\in S(i=1,2,\ldots k)$, there exists $\beta_1\beta_2\ldots\beta_l\in S^*$ such that

$$P(\alpha_1\alpha_2\ldots\alpha_k)\neq P(\alpha_1\alpha_2\ldots\alpha_k\beta_1\beta_2\ldots\beta_l).$$

Note that the above mentioned three classes constitute a partition (i.e., mutually exclusive and exhaustive) of $\{A_{b_l,b_r}(S,f)\}$.

If $A_{b_l,b_r}(S,f)$ belongs to, for example, Class I, we say that f belongs to Class I (write $f \in C_I$) for brevity when S, b_l, and b_r are understood.

4.2. Lemmas for Classification

Several lemmas for classification of the systems, showing such and such f's do or do not belong to a particular class, are listed here. Most of them are straightforward and will be given without proof.

Lemma 4.1.

If $f(\alpha,\beta,\gamma)=f(\alpha,\beta,\delta)$ for $^\forall\alpha,\beta,\gamma,\delta \in S$, then $f \in C_I$ under arbitrary boundary values (b_l,b_r). In this case, k-value equals 1.

Lemma 4.1 notes that there exists a (unilateral) iterative automaton where states of a cell do not affect those of a cell to the left, which belongs to Class I.

Lemma 4.2.

If for $^\forall\gamma \in S$ and $^\exists\beta \in S$, $f(b_l,\beta,\gamma)=\beta$ and if there exists $\alpha \in S$ such that $f(b_l,\alpha,\alpha)=f(\alpha,\alpha,\alpha)=\alpha$ and $f(b_l,\alpha,\beta)=f(\alpha,\alpha,\beta)=\beta$, then $f \in C_{II}$.

The first condition guarantees the existence of such a particular state to be used at the leftmost cell (C_1) that could not be changed by any states of the remaining cells and the second, that of a specific state configuration through which the above mentioned state propagates itself from right to left.

Proof

It is clear that there exists f which satisfies conditions of the lemma. Note that $P(\beta)=P(\beta\beta')=(\beta)$ for $\beta' \in S^*$ and $P(\alpha^n\beta)=\alpha^n\beta\ldots$ (n is any positive integer), where (β) means an infinite repetition of β, and α^n means $\overbrace{\alpha\alpha\ldots\alpha}^{n}$. These two relations satisfy the requirments for f to be in Class II.(Q.E.D.)

Lemma 4.3.

If $f(\alpha,\beta,\gamma)\neq f(\alpha,\beta,\delta)$ for $^\forall\alpha,\beta,\gamma,\delta\in S(\gamma\neq\delta)$, then $f\in C_{III}$ for $^\forall(b_l,b_r)$.

The condition of Lemma 4.3 assures any information in a cell to be transmitted to the left side cell, which means that f belongs to Class III.

It can be seen by the above mentioned three lemmas that each of C_I, C_{II} and C_{III} is not an empty set.

Lemma 4.4.

If $f(b_l,\beta,\gamma)=f(b_l,\beta,\delta)$ for $^\forall\beta,\gamma,\delta\in S$, then $f\in C_I$ and whose k-value equals 1.

Lemma 4.5.

For any $\gamma\in S$, if there exists $\beta\in S$ such that $f(b_l,\beta,\gamma)=\beta$, then $f\notin C_{III}$.

Lemma 4.6.

If there exist β and β' in S such that $f(b_l,\beta,\beta')=\beta$ and $f(\beta,\beta',\gamma)=\beta'$ for $^\forall\gamma\in S$, then $f\notin C_{III}$.

Lemma 4.6 could be easily generalized to cases where condition part reads "if there exist $\beta_1\beta_2\ldots$, and β_n".

Lemma 4.7.

Let $b_l=b_r=\beta$ and if there exists $\alpha\in S(\alpha\neq\beta)$ such that a part of the transition table corresponding to f includes

α	α	β
α	α	α
β	α	β

and

β	α
α	α
β	α

, then $f\in C_I$.

In fact, $P[(\alpha\beta)^n\alpha\alpha]=(\alpha\beta)^n\alpha....$ for the cells mentioned in Lemma 4.7.

Lemma 4.8.

If there exist distinct states α and β in S such that a part of the transition table corresponding to f includes

$$\begin{array}{c|cc} \alpha & \alpha & \beta \\ \hline \alpha & \alpha & \beta \\ b_l & \alpha & \beta \end{array}\quad,\ \text{then } f\in C_I.$$

In this case, $P(\alpha^n\beta)=\alpha^n\beta.....$

All of the above mentioned lemmas are valid for cells with the number of states m being greater than or equal to 2. Now, let us see some cases where S is fixed to be {0,1}.

Lemma 4.9.

If $b_l=1$ and if f is given by

$$\begin{array}{c|cc} 0 & 0 & 1 \\ \hline 0 & * & 1 \\ 1 & * & 1 \end{array}\quad \text{and} \quad \begin{array}{c|cc} 1 & 0 & 1 \\ \hline 0 & * & 1 \\ 1 & 0 & 1 \end{array}$$ (* means 0 or 1), then $f\in C_{III}$.

Proof

Let $P(\alpha_1\alpha_2....\alpha_n\ 111)=p_1p_2....p_t....$ The (n+3)rd output P_{n+3} equals 1 for $^\forall\alpha_1\alpha_2....\alpha_n\in S^*$. On the other hand, the (n+3)rd output of $P(\alpha_1\alpha_2...\alpha_n 110)$ equals 0 for $^\forall\alpha_1\alpha_2....\alpha_n\in S^*$. Thus $P(\alpha_1\alpha_2....\alpha_n 110)$ $\neq P(\alpha_1\alpha_2....\alpha_n 111)$, which guarantees f to be in Class III. The lemma holds whether b_r equals 0 or 1.

Lemma 4.10.

If $b_l=1$ and if f is given by

$$\begin{array}{c|cc} 0 & 0 & 1 \\ \hline 0 & * & * \\ 1 & 1 & 1 \end{array}\quad \text{and} \quad \begin{array}{c|cc} 1 & 0 & 1 \\ \hline 0 & 0 & 0 \\ 1 & 0 & 1 \end{array}\quad,\ \text{then } f\in C_{II}.$$

Proof

Letting α=1 and β=0 in Lemma 4.8, we see that $f\notin C_I$. On the other hand, $P(0\ 1\ \beta)=(0\ 1)$ for $^\forall\beta\in S^*$, which means $f\notin C_{III}$. Thus, we have $f \in C_{II}$.

4.3. Simplification Theorem for Class I

4.3.1. Simplification Theorem

Lemmas listed in the preceding section determine that f belongs to or does not belong to a particular Class if the function f satisfies conditions of one of them. But it is clear that we can't classify an arbitrarily given f by these lemmas even if the number of states m is restricted.

So, in this section, we investigate the characteristics of our classification mainly through a "simplification theorem" for Class I.

By definition, a system of Class I with k-value k is that output sequences thereof do not depend on the initial states of cells located at the right of the k-th cell (i.e. C_i, $i \geqq k+1$). The definition can be reduced considerably, yielding a finitary conditions. That is, we have the following simplification theorem.

Theorem 4.1. (*simplification theorem*)

If there exists an integer $k>0$, and if $P(\alpha_1\alpha_2 \cdots \alpha_k)=P(\alpha_1\alpha_2 \cdots \alpha_k\beta)$ holds for $^\forall\alpha_i \in S (i= 1,2,\ldots,k)$ and $^\forall\beta \in S$, then $P(\alpha_1\alpha_2 \cdots \alpha_k)=P(\alpha_1 \alpha_2 \cdots \alpha_k\beta)$ for $^\forall\beta \in S^*$.

The theorem says that for any system under investigation to belong to Class I, it is not necessary to compare all possible output sequences adding an infinite number of cells, but adding only one cell to the original system.

Proof

It suffices to show that $P(\alpha_1\alpha_2 \cdots \alpha_k\beta_1\beta_2)=P(\alpha_1\alpha_2 \cdots \alpha_k\beta_1)$ for

$\forall\beta_1,\beta_2 \in S$ when the hypothesis of the theorem holds. Let $F_{b_l,b_r}(\alpha_1\alpha_2 \ldots\alpha_k\beta_1\beta_2)=\alpha_1'\alpha_2'\ldots\alpha_k'\beta_1'\beta_2'$, and we have $P(\alpha_1\alpha_2\ldots\alpha_k\beta_1\beta_2)=\alpha_1 P(\alpha_1'\alpha_2'\ldots\alpha_k'\beta_1'\beta_2')$. On the other hand, $F_{b_l,b_r}(\alpha_1\alpha_2\ldots\alpha_k\beta_1)$ can be written as $\alpha_1'\alpha_2'\ldots\alpha_k'\tilde{\beta}_1'$ by an appropriate choice of $\tilde{\beta}_1 \in S$. Then, $P(\alpha_1\alpha_2\ldots\alpha_k\beta_1)=\alpha_1 P(\alpha_1'\alpha_2'\ldots\alpha_k'\tilde{\beta}_1')$, where $P(\alpha_1'\alpha_2'\ldots\alpha_k'\tilde{\beta}_1')$, in turn, equals $P(\alpha_1'\alpha_2'\ldots\alpha_k'\beta_1')$ by the hypothesis. Thus, we have $P(\alpha_1\alpha_2\ldots\alpha_k\beta_1)=\alpha_1 P(\alpha_1'\alpha_2'\ldots\alpha_k'\beta_1')$.

The above process shows that the first output states of $P(\alpha_1\alpha_2\ldots\alpha_k\beta_1\beta_2)$ and $P(\alpha_1\alpha_2\ldots\alpha_k\beta_1)$ are identical and that the remaining part of the output sequences are represented by $P(\alpha_1'\alpha_2'\ldots\alpha_k'\beta_1'\beta_2')$ and $P(\alpha_1'\alpha_2'\ldots\alpha_k'\beta_1')$ respectively. So, the same procedure can be applied to the latter pair of output sequences, ensuring the validity of the theorem. (Q.E.D.)

Comment 4.1.

Notice that the hypothesis of the theorem contains a clause "if $P(\alpha_1\alpha_2\ldots\alpha_k)=P(\alpha_1\alpha_2\ldots\alpha_k\beta)$ holds for $\forall\alpha_i \in S(i=1,2,\ldots k)$ and ....". If we adopt $\exists\alpha_i \in S$ instead of $\forall\alpha_i \in S$ in this clause, the corresponding proposition "If there exists an integer $k>0$, and if $P(\alpha_1\alpha_2\ldots\alpha_k)=P(\alpha_1\alpha_2\ldots\alpha_k\beta)$ holds for $\exists\alpha_i \in S(i=1,2,\ldots k)$ and $\forall\beta \in S$, then $P(\alpha_1\alpha_2\ldots\alpha_k)=P(\alpha_1\alpha_2\ldots\alpha_k\beta)$ for $\forall\beta \in S^*$" is not always true. That is, no simplification procedure similar to that of Theorem 4.1 could be generally applied for Class II-like systems.

A counterexample exhibiting the above mentioned situation is shown as follows:

Counterexample Let $S=\{0,1\}$ and $b_l=b_r=1$. Let f be given by

0	0	1
0	1	1
1	1	1

and

1	0	1
0	0	1
1	0	1

.

Then, P(10)=P(100)=P(101)=10(1) and P(1010)=1010(1). Thus P(10)$\neq$ P(1010). In fact, the system belongs to C_{III}.

Comment 4.2.

As the converse to Theorem 4.1 obviously holds, $A_{b_l,b_r}(S,f)$ belongs to Class I iff the if part of Theorem 4.1 is true. The equivalent condition for a system to belong to Class I, however, could be replaced by a weaker one, if the process of the proof is carefully followed. That is to say, it suffices to demand that for all β in S, $P(\alpha_1\alpha_2....\alpha_k\beta)$'s are identical. That is, the following proposition holds. If there exists an integer k>0, and if $P(\alpha_1\alpha_2....\alpha_k\beta)=P(\alpha_1\alpha_2....\alpha_k\beta')$ for $^\forall\alpha_i \in S(i=1,2,....k)$ and $^\forall\beta,\beta' \in S$, then $P(\alpha_1\alpha_2....\alpha_k)=P(\alpha_1\alpha_2....\alpha_k\beta)$.

Comment 4.3.

Further, the following characteristics can be observed. If a system belongs to Class I with k-value more than or equal to 2, there exist $\alpha_i \in S$ (i=1,2,....k-1), $\alpha_k \in S$ and $\alpha_k' \in S(\alpha_k \neq \alpha_k')$ such that $P(\alpha_1\alpha_2....\alpha_{k-1}\alpha_k)=P(\alpha_1\alpha_2....\alpha_{k-1}\alpha_k')$. For, there exist some elements $\gamma,\delta,\varepsilon,\varepsilon'(\varepsilon\neq\varepsilon')$ in S such that $f(\gamma,\delta,\varepsilon)\neq f(\gamma,\delta,\varepsilon')$. (If this is not true, the system under consideration becomes Class I with k-value 1. See Lemma 4.4.) Further, as the system belongs to Class I, $P(\beta_1\beta_2....\beta_{k-2}\gamma\ \delta\ \varepsilon)=P(\beta_1\beta_2....\beta_{k-2}\gamma\ \delta\ \varepsilon')$ for $^\forall\beta_i$(i=1,2,....,k-2). Let $F_{b_l,b_r}(\beta_1\beta_2....\beta_{k-2}\ \gamma\ \delta\ \varepsilon)=\alpha_1\alpha_2....\alpha_{k-2}\alpha_{k-1}\alpha_k\tilde{\varepsilon}$ and $F_{b_l,b_r}(\beta_1\beta_2....\beta_{k-2}\gamma\ \delta\ \varepsilon')=\alpha_1\alpha_2....\alpha_{k-1}\alpha_k'\tilde{\varepsilon}'$. Then, $P(\alpha_1\alpha_2....\alpha_{k-1}\alpha_k\tilde{\varepsilon})=P(\alpha_1\alpha_2\alpha_{k-1}\alpha_k'\tilde{\varepsilon}')$ holds, which assures that $P(\alpha_1\alpha_2....\alpha_{k-1}\alpha_k)=P(\alpha_1\alpha_2...\alpha_{k-1}\alpha_k')$ where $\alpha_k\neq\alpha_k'$.

4.3.2. Systems with External Input

In this subsection, let us consider a system with external input; we assume that b_l is fixed as before, while b_r may vary at discrete time step as $x_1\ x_2....x_t....$, i.e., we have an external input to the rightmost cell.

Let $P(\alpha_1\alpha_2....\alpha_n)_{x_1x_2....x_t....}$ denote a state sequence of the leftmost cell (C_1), when the system under consideration is fed an external input sequence $x_1x_2....x_t....$, starting with an initial state configuration $\alpha_1\alpha_2....\alpha_n$.

Systems with and without external input (the latter means a fixed boundary condition here) are compared concerning the above mentioned simplification theorem, obtaining the following

Theorem 4.2.

Let k be a positive integer and if $P(\alpha_1\alpha_2....\alpha_k)=P(\alpha_1\alpha_2....\alpha_k)_{x_1x_2....x_t....}$ for $^\forall\alpha_i\in S(i=1,2,....,k)$ and $^\forall x_1x_2....x_t....$, then $P(\alpha_1\alpha_2....\alpha_k)=P(\alpha_1\alpha_2....\alpha_k\beta)$ for $^\forall\alpha_i\in S,^\forall\beta\in S$, and vice versa.

Proof

First, the converse part of the theorem is shown.

Let $F_{b_l,x_1}(\alpha_1\alpha_2....\alpha_k)=\alpha_1'\alpha_2'....\alpha_k'$. Then, $P(\alpha_1\alpha_2....\alpha_k)_{x_1x_2...x_t....}=\alpha_1\underline{P(\alpha_1'\alpha_2'...\alpha_k')_{x_2...x_t....}}$.

By assumption, $P(\alpha_1\alpha_2....\alpha_k)=P(\alpha_1\alpha_2....\alpha_kx_1)$ holds, and the right-hand side expression equals $\alpha_1P(\alpha_1'\alpha_2'....\alpha_k'x_1')$ for an appropriate $x_1'\in S$. Then, applying the assumption once again, $P(\alpha_1\alpha_2....\alpha_k)=\alpha_1\underline{P(\alpha_1'\alpha_2'....\alpha_k')}$. One step of $P(\alpha_1\alpha_2....\alpha_k)_{x_1x_2....x_t....}$ and $P(\alpha_1\alpha_2....\alpha_k)$ proceeds as such, $P(\alpha_1\alpha_2....\alpha_k)=P(\alpha_1\alpha_2....\alpha_k)_{x_1x_2...x_t....}$ is seen to hold by noting that the underlined expressions preserve the similar pattern as before and that the same process may be taken recursively.

Next, let us assume $P(\alpha_1\alpha_2....\alpha_k)=P(\alpha_1\alpha_2....\alpha_k)_{x_1x_2....x_t....}$ for any sequence $x_1x_2....x_t....$. Let $F_{b_l,b_r}(\alpha_1\alpha_2....\alpha_k\beta)=\alpha_1'\alpha_2'........\alpha_k'\beta'$ for arbitrarily fixed $\beta\in S$. Then $P(\alpha_1\alpha_2....\alpha_k\beta)=\alpha_1\underline{P(\alpha_1'\alpha_2'....\alpha_k'\beta')}$. On the other hand,

$$P(\alpha_1\alpha_2....\alpha_k)=P(\alpha_1\alpha_2....\alpha_k)_{\beta x_2x_3....x_t....}$$

$$=\alpha_1 P(\alpha_1'\alpha_2'\ldots\alpha_k')x_2x_3\ldots x_t\ldots$$
$$=\alpha_1 \underline{P(\alpha_1'\alpha_2'\ldots\alpha_k')}.$$

Comparing the underlined two expressions, a similar reasoning as before leads us to the conclusion that $P(\alpha_1\alpha_2\ldots\alpha_k)=P(\alpha_1\alpha_2\ldots\alpha_k\beta)$. (Q.E.D.)

By Theorem 4.1 and Theorem 4.2, $A_{b_l,b_r}(S,f)$ belongs to Class I iff for some positive integer k, $P(\alpha_1\alpha_2\ldots\alpha_k)=P(\alpha_1\alpha_2\ldots\alpha_k)x_1x_2\ldots x_t\ldots$ for $^\forall\alpha_i$ (i=1,2,.... k) and $^\forall x_1x_2\ldots x_t\ldots$. This means that if $A_{b_l,b_r}(S,f)$ belongs to Class I, then a modified system with external input does not change the validity of k-value, that is, it is useless to consider more than k cells (as long as the leftmost cell is regarded as an output cell).

As a corollary to Theorem 4.2, we have

Corollary 4.1.

If $A_{b_l,b_r}(S,f)$ belongs to Class I, then $A_{b_l,b_r'}(S,f)$ also belongs to Class I for any $b_r'\in S$, and has the same k-value.

4.4. Characterization of Classification via State Partition

Let $A_{b_l}(n)$ denote an iterative automaton of length n, whose left boundary input is fixed to b_l and has external inputs to the rightmost cell. Several state partitions of the state set of $A_{b_l}(n)$, which is composed of m^n elements, are utilized to characterize our classification of $\{A_{b_l,b_r}(S,f)\}$.

Some definitions and notations are given first.

4.4.1. Definitions on Partitions

Let $\pi=\{B_1,B_2,\ldots,B_r\}$ denote a partition of the state set of some $A_{b_l}(n)$. Assume that $F_i(B_j)$ is the set of all states into which a state in B_j transits by an application of input i, i.e.,

$$F_i(B_j) \triangleq \{q_i \mid F_{b_l,i}(p)=q_i,\ \forall p \in B_j\}.$$

A partition π is called a preserved partition (P.P.) if for any block B_j of π, there exists a block B_s of π such that $\bigcup_{i \in S} F_i(B_j) \subset B_s$*.

An ordering is given naturally in the set of all partitions of a set. That is, for partitions $\pi=\{B_1,B_2,\ldots,B_r\}$ and $\pi'=\{B_1',B_2',\ldots,B_t'\}$,

$\pi \leq \pi' \Leftrightarrow$ For any B_j of π, there exists B_i' of π' such that $B_j \subset B_i'$.

The minimum and the maximum partitions are written $\pi(0)$ and $\pi(1)$, respectively.

A P.P. derived from a partition π is the one that is the smallest of the preserved partitions greater than π. It is clear that such a P.P. is uniquely determined from π. The P.P. derived from $\pi(0)$ is denoted by $\pi_{\bar{R}}$. By definition, an element of the state set of $A_{b_l}(n)$ is a string composed of n elements of S. Let $\pi(u^{(1)},u^{(2)},\ldots,u^{(v)})$ denote a partition every block of which contains strings whose left side v components are identical.

An operator a is a function from the set of partitions on the state set of $A_{b_l}(n)$ into that of $A_{b_l}(n+1)$ as follows. Let $\pi=\{B_1,B_2,\ldots,B_r\}$ be a partition on the state set of $A_{b_l}(n)$. Then $a\pi=\{B_1',B_2',\ldots,B_r'\}$, where B_i' is composed of any strings of the form $x\alpha$ such that x is a string (of length n) in B_i and α is an element of S. Thus, the cardinality of B_i' is m times that of B_i.

In what follows, a partition on the state set of $A_{b_l}(n)$ is distinguished from the others by superscript n, as $\pi^{(n)}$.

---

* Note that a usual definition of a P.P. is less restrictive than ours. For example, see [5].

4.4.2. Characterization of Class I

When $n \geq 2$, a relation K is defined on the state set of $A_{b_l}(n)$, that is on S^n, as follows:

$$\alpha_1\alpha_2\ldots\alpha_n \; K \; \beta_1\beta_2\ldots\beta_n \Leftrightarrow \begin{cases} \alpha_1=\beta_1, \text{ and} \\ f(b_l,\alpha_1,\alpha_2)=f(b_l,\beta_1,\beta_2). \end{cases}$$

K is obviously an equivalence relation, and the partition defined by K is written as $\pi_k^{(n)}$. When n=1, $\pi_k^{(1)}$ is defined to be $\pi^{(1)}(0)$.

Then, we can characterize Class I of our classification in terms of partitions by the following

Theorem 4.3.

$A_{b_l, b_r}(S,f)$ belongs to Class I with k-value less than or equal to n iff $\pi_{\bar{R}}^{(n)}$ is a refinement of $\pi_k^{(n)}$, i.e., $\pi_{\bar{R}}^{(n)} \leq \pi_k^{(n)}$.

Proof

This proof is divided into two parts according to whether n=1 or not.

I) n=1; It is clear from Theorem 4.2 that $A_{b_l, b_r}(S,f)$ belongs to Class I with k-value 1 iff $\pi_{\bar{R}}^{(1)}=\pi^{(1)}(0)$. This is the case corresponding to Lemma 4.4.

II) $n \geq 2$; Let $\pi_{\bar{R}}^{(n)} \leq \pi_k^{(n)}$. Given any initial state configuration $\alpha_1\alpha_2\ldots\alpha_n$ and external input sequence $x_1x_2\ldots x_t\ldots$, there exists a sequence $B^0, B^1, B^2,\ldots,B^t,\ldots$ of blocks of $\pi_{\bar{R}}^{(n)}$, such that

$$\alpha_1\alpha_2\ldots\alpha_n \in B^0$$
$$F(\alpha_1\alpha_2\ldots\alpha_n)_{x_1} \in B^1$$
$$\vdots$$
$$F^t(\alpha_1\alpha_2\ldots\alpha_n)_{x_1x_2\ldots x_t} \in B^t \; ,$$
$$\vdots$$

where $F^t(\alpha_1\alpha_2\ldots\alpha_n)_{x_1x_2\ldots x_t}$ means the latest state configuration reached by an application of external input sequence $x_1x_2\ldots x_t$ to the initial state configuration $\alpha_1\alpha_2\ldots\alpha_n$. As long as the initial state configuration equals $\alpha_1\alpha_2\ldots\alpha_n$, the sequence of blocks of $\pi_{\bar{R}}^{(n)}$ mentioned above remains the same for any external input sequence $x_1'x_2'\ldots x_t'\ldots$, because $\pi_{\bar{R}}^{(n)}$ is a P.P.. As $\pi_{\bar{R}}^{(n)} \leq \pi_k^{(n)}$, there exists a sequence $B^{0'}$, $B^{1'},\ldots B^{t'},\ldots$ of blocks of $\pi_k^{(n)}$, corresponding to the sequence $B^0, B^1,\ldots B^t\ldots$ of $\pi_{\bar{R}}^{(n)}$ such that $B^t \subset B^{t'}$ $(t=0,1,2,\ldots)$. Thus, we have for an arbitrary input sequence $x_1x_2\ldots x_t\ldots$,

$$\begin{aligned}
&\alpha_1\alpha_2\ldots\alpha_n \in B^{0'},\\
&F(\alpha_1\alpha_2\ldots\alpha_n)_{x_1} \in B^{1'},\\
&\quad\vdots\\
&F^t(\alpha_1\alpha_2\ldots\alpha_n)_{x_1x_2\ldots x_t} \in B^{t'},
\end{aligned}$$

which means, as the construction method of $\pi_k^{(n)}$ indicates, that $P(\alpha_1\alpha_2\ldots\alpha_n)_{x_1x_2\ldots x_t\ldots} = P(\alpha_1\alpha_2\ldots\alpha_n)_{x_1'x_2'\ldots x_t'\ldots}$ for any $x_1'x_2'\ldots x_t'\ldots$. Therefore, $A_{b_l,b_r}(S,f)$ belongs to Class I with k-value less than or equal to n by Theorem 4.2.

Conversely, let us assume that $A_{b_l,b_r}(S,f)$ belongs to Class I with k-value less than or equal to $n (n \geq 2)$. Then, by Theorem 4.2, an output sequence of cell C_1 cannot be changed by varying the external input sequence of $A_{b_l}(n)$. Thus, state configurations in a block of $\pi_{\bar{R}}^{(n)}$ should produce the same output sequence, hence $\pi_{\bar{R}}^{(n)} \leq \pi_k^{(n)}$. (Q.E.D.)

Theorems hitherto mentioned dealt with an iterative automaton of length n and gave sufficient conditions for the corresponding system to belong to Class I with k-value *less than or equal to n*. Another sufficient condition applying to an automaton of length n is provided here that decides whether the system in question belongs to Class I with k-value *less than or equal to* $(n-1)$, and with Theorem 4.3, is given a characterization of Class I with k-value *equal to n*.

Consider $A_{b_l}(n)$, and let $\pi_{\bar{R}}^{(n)}$ denote a P.P. derived from $\pi^{(n)}(u^{(1)}, u^{(2)},\ldots u^{(n-1)})$. Then, we have the following

Theorem 4.4.

$A_{b_l,b_r}(S,f)$ belongs to Class I with k-value less than or equal to (n-1) iff $\pi_{\tilde{R}}^{(n)} \leq \pi_k^{(n)}$, where $n \geq 3$.

Proof

Assume a system belong to Class I with k-value less than or equal to (n-1). Then, $\pi_{\bar{R}}^{(n-1)} \leq \pi_k^{(n-1)}$ by Theorem 4.1. If $\pi_1 \leq \pi_2$, then $a\pi_1 \leq a\pi_2$ by the definition of a, thence $a\pi_{\bar{R}}^{(n-1)} \leq a\pi_k^{(n-1)}$.

As $a\pi_{\bar{R}}^{(n-1)}$ is a P.P. and $a\pi_{\bar{R}}^{(n-1)} \geq \pi^{(n)}(u^{(1)}, u^{(2)}, \ldots. u^{(n-1)})$, $a\pi_{\bar{R}}^{(n-1)} \geq \pi_{\tilde{R}}^{(n)}$ is obtained. On the other hand, $a\pi_k^{(n-1)} = \pi_k^{(n)}$ in case $n \geq 3$. Thus $\pi_k^{(n)} \geq \pi_{\tilde{R}}^{(n)}$.

Conversely, let $\pi_{\tilde{R}}^{(n)} \leq \pi_k^{(n)}$. For any $\alpha_1\alpha_2 \ldots. \alpha_{n-1} \in S^{n-1}$ and for any β and β' in S, $\alpha_1\alpha_2 \ldots. \alpha_{n-1}\beta$ and $\alpha_1\alpha_2 \ldots. \alpha_{n-1}\beta'$ rest in the same block of $\pi_{\tilde{R}}^{(n)}$, and so do of $\pi_k^{(n)}$. Thus, $P(\alpha_1\alpha_2 \ldots. \alpha_{n-1}\beta)$ $= P(\alpha_1\alpha_2 \ldots. \alpha_{n-1}\beta')$ holds, which assures us that the system under consideration belongs to Class I with k-value less than or equal to (n-1). (See Comment 4.2 of Theorem 4.1.) (Q.E.D.)

Theorems 4.3 and 4.4 are combined to the following

Theorem 4.5.

$A_{b_l,b_r}(S,f)$ belongs to Class I with k-value n iff $\pi_{\bar{R}}^{(n)} \leq \pi_k^{(n)}$ and $\pi_{\tilde{R}}^{(n)} \nleq \pi_k^{(n)}$, where $n \geq 3$.

4.4.3. Sufficient Condition for Class II

Here a sufficient condition for $A_{b_l,b_r}(S,f)$ to belong to Class II is given.

A subset $\bar{\pi} = \{B_{i_1}, B_{i_2}, \ldots., B_{i_s}\}$ of a partition $\pi = \{B_1, B_2, \ldots. B_r\}$ on some state set of an automaton, where $\{i_1, i_2, \ldots, i_s\} \subset \{1, 2, \ldots, r\}$, is called a partial P.P. of π if $\bar{\pi}$ constitutes a P.P. disregarding the states not appearing in the blocks of $\bar{\pi}$. Let $(\pi_{\bar{R}}^{(n)})$ denote a partition on a subset of S^n that is composed of blocks of $\pi_{\bar{R}}^{(n)}$ contained in any one of the blocks of $\pi_k^{(n)}$. Then, we have a sufficient condition for a system to belong to Class II as follows.

Theorem 4.6.

If $(\pi_{\bar{R}}^{(n)}) \neq \pi_{\bar{R}}^{(n)}$ for any positive integer n, and if there exists a positive integer n such that $(\pi_{\bar{R}}^{(n)})$ contains a partial P.P. of $\pi_{\bar{R}}^{(n)}$, then A_{b_l,b_r} (S,f) belongs to Class II. Further, if A_{b_l,b_r} (S,f) belongs to Class II satisfying these conditions, then $A_{b_l,b_r'}$ (S,f) also belongs to Class II for any $b_r' \in S$.

Proof

Let $(\pi_{\bar{R}}^{(n)})=\{B_1,B_2\ldots,B_e\}$ denote a partial P.P. of $\pi_{\bar{R}}^{(n)}$ contained in $(\pi_{\bar{R}}^{(n)})$. Consider $F(\alpha_1\alpha_2\ldots\alpha_n)x_1x_2\ldots x_t\ldots$, where $\alpha_1\alpha_2\ldots\alpha_n$ is assumed to be in some $B_i (1 \leq i \leq e)$ and $x_1x_2\ldots x_t\ldots$ is an arbitrary external input sequence. Then, there exists a sequence $B^0, B^1,\ldots B^t\ldots$ of blocks of $(\pi_{\bar{R}}^{(n)})$ such that

$$\begin{gathered} \alpha_1\alpha_2\ldots\alpha_n \in B^0=B_i, \\ F(\alpha_1\alpha_2\ldots\alpha_n)x_1 \in B^1, \\ \vdots \\ F^t(\alpha_1\alpha_2\ldots\alpha_n)x_1x_2\ldots x_t \in B^t, \\ \vdots \end{gathered}$$

As each block of $\{B_1,B_2,\ldots,B_e\}$ is contained in some block of $\pi_k^{(n)}$, there exists a sequence $B^{0\prime},B^{1\prime},\ldots,B^{t\prime}\ldots$ of blocks of $\pi_k^{(n)}$ such that $B^t \subset B^{t\prime}$ $(t=0,1,2,\ldots)$. This is equivalent to say that all of the $P(\alpha_1\alpha_2\ldots\alpha_n)x_1x_2\ldots x_t\ldots$'s for any $x_1x_2\ldots x_t\ldots$ are identical, which in turn implies $P(\alpha_1\alpha_2\ldots\alpha_n)=P(\alpha_1\alpha_2\ldots\alpha_n\beta)$ for $\forall\beta \in S^*$. Besides, the result obtained is valid for any $b_r \in S$. On the other hand, $\pi_{\bar{R}}^{(n)} \nsubseteq \pi_k^{(n)}$ for any n by the fact that $(\pi_{\bar{R}}^{(n)}) \neq \pi_{\bar{R}}^{(n)}$. Thus, by Theorem 4.3, the system in question does not belong to Class I. (Q.E.D.)

4.5. D-Set and its Decision Problem

A configuration $\alpha_1\alpha_2\ldots\alpha_k$ is called a *D-configuration* if $P(\alpha_1\alpha_2\ldots\alpha_k)=P(\alpha_1\alpha_2\ldots\alpha_k\beta)$ for $^\forall\beta\in S^*$. D stands for definite in that an output sequence is determined solely by that part of an initial configuration. Given $A_{b_l,b_r}(S,f)$, the set of all D-configurations in $A_{b_l,b_r}(S,f)$ will be denoted by $A_D(S,f,b_l,b_r)$ or simply A_D when S,f,b_l , and b_r are understood.

A_D, which is called a *D-set*, characterizes our classification of $\{A_{b_l,b_r}(S,f)\}$ as follows;

Class I $\Leftrightarrow {}^\exists k>0,\ A_D\supset S^k$.
Class II $\Leftrightarrow A_D\neq\phi$ and there exists no $k>0$ such that $A_D\supset S^k$.
Class III $\Leftrightarrow A_D=\phi$.

It seems to be an interesting problem to investigate properties of D-sets. In this section, it is shown to be recursively unsolvable whether arbitrarily given $\alpha_1\alpha_2\ldots\alpha_k$ belongs to the D-set of an arbitrarily given $A_{b_l,b_r}(S,f)$. The undecidability result is obtained by embedding a Turing machine (TM) into a system of iterateve automata.

As a step to the goal, a classification of the systems of iterative automata with an infinite number of cells in regard to the right to left information transmission capability is defined, and the relations between this new classification and that of $\{A_{b_l,b_r}(S,f)\}$ are mentioned.

4.5.1. Classification of Systems of Half-Infinite Iterative Automata

A system of half-infinite iterative automata is a collection of an infinite number of cells arranged as in Fig.4.1, where the transition function f of each cell is a mapping from S^3 into S as before. In order to assure finiteness of an operation, it is assumed that there exists a special state $s_o \in S$, called a quiescent state, such that $f(s_o, s_o, s_o)=s_o$ in any (S,f).

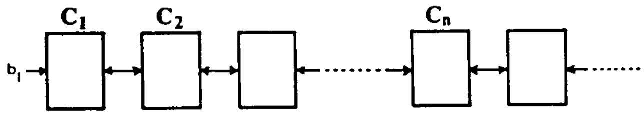


Fig. 4.1. System of half-infinite iterative automata

As shown in Fig. 4.1, the left boundary condition is fixed to $b_l \in S$. When an initial state configuration $\alpha_1\alpha_2....\alpha_n$ is concerned, it is assumed that the state of cell C_i corresponds to $\alpha_i (i=1,2,...,n)$, and the other cells $C_i (i>n)$ are given the quiescent state s_o. Let $P_i(\alpha)$ denote an output sequence of C_1, given an initial state configuration $\alpha=\alpha_1\alpha_2....\alpha_n$.

Being connected with a D-configuration, let us define an ID-configuration in the system of half-infinite iterative automata as follows;

$$\alpha_1\alpha_2....\alpha_k \text{ is an ID-configuration} \Leftrightarrow P_i(\alpha_1\alpha_2....\alpha_k)=P_i(\alpha_1\alpha_2....\alpha_k\beta) \quad \text{for } \beta \in S^*.$$

Given S,f, and b_l, the set of all ID-configurations $A_{ID}(S,f,b_l)$ or shortly A_{ID} may be used to define a classification of the systems of half-infinite iterative automata. That is, replace A_D by A_{ID} in the reformulation of the classification of $\{A_{b_l,b_r}(S,f)\}$ cited in this section. The classification thus obtained again represents the right to left information transmission capability of the systems of half-infinite iterative automata, whose classes are distinguished by subscript i, as Classes I_i, II_i and III_i.

As a result of Theorems 4.1 and 4.2, (S,f) with quiescent state belongs to Class I_i for some $b_l \in S$ iff $A_{b_l,b_r}(S,f)$ belongs to Class I.

The following lemma plays an important role in the rest of this section.

Lemma 4.11.

For any (S,f) with quiescent state, if $\alpha_1\alpha_2 \cdots\cdots \alpha_k \in S^*$ is a D-configuration for some $b_l, b_r \in S$, then it is also an ID-configuration for the same b_l.

Proof

Assume that $\alpha_1\alpha_2 \cdots\cdots \alpha_k$ is not an ID-configuration. That is, assume that there exists a configuration $\beta_1\beta_2 \cdots\cdots \beta_m$ such that $P_i(\alpha_1\alpha_2 \cdots\cdots \alpha_k) \neq P_i(\alpha_1\alpha_2 \cdots\cdots \alpha_k\beta_1\beta_2 \cdots\cdots \beta_m)$. Let t_o be the time when $P_i(\alpha_1\alpha_2 \cdots\cdots \alpha_k)$ and $P_i(\alpha_1\alpha_2 \cdots\cdots \alpha_k\beta_1\beta_2 \cdots\cdots \beta_m)$ produce different outputs for the first time. Then, $P(\overbrace{\alpha_1\alpha_2 \cdots\cdots \alpha_k\, s_o\, s_o \cdots\cdots s_o}^{t_o}) \neq P(\overbrace{\alpha_1\alpha_2 \cdots\cdots \alpha_k\beta_1\beta_2 \cdots\cdots \beta_m\, s_o \cdots\cdots s_o}^{t_o})$, which contradicts the hypothesis that $\alpha_1\alpha_2 \cdots\cdots \alpha_k$ is a D-configuration. Hence the lemma holds. (Q.E.D.)

It can be easily seen by the above mentioned lemma that if (S,f) belongs to Class III_i, then it also belongs to Class III. Whether the converse of Lemma 4.11 holds or not is open.

4.5.2. Embedding a Turing Machine

It is well known that any TM could be simulated by an equivalent TM which operates on a potentially half-infinite tape. Now, let us consider a system of half-infinite iterative automata embedding such a TM that operates on a half-infinite tape.

Following Davis [10], let us define a (simple) TM by a finite nonempty set P of quadruples of the forms (1) $q_i S_j S_k q_l$ (2) $q_i S_j R q_l$ (3) $q_i S_j L q_l$, where q_i and q_l are in the set $Q = \{q_1, q_2, \ldots, q_r\}$ of *internal states*, and S_j is an element of the set $K = \{B, S_1, \ldots, S_t\}$ of tape *alphabet*. As for further details of TM, see Davis [10].

Let us consider a cell (S,f) arranged as in Fig.4.2, where

B : blank tape

e : absence of the head of TM

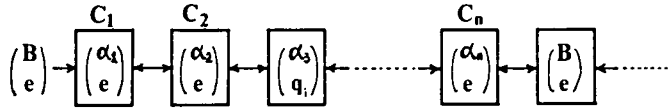


Fig. 4.2. System of half-infinite iterative automata simulating a Turing machine

$S=K\times(Q\cup\{e\})$, $e\notin Q$, and f is a mapping from S^3 into S, simulating essentially the behavior of the above mentioned TM as follows;

(1) In case $q_i S_j S_k q_l \in P$,

$f(\binom{\alpha}{e}, \binom{S_j}{q_i}, \binom{\beta}{e}) = \binom{S_k}{q_l}$ for $\forall\alpha,\beta\in K$.

(2) In case $q_i S_j R q_l \in P$,

$f(\binom{\alpha}{e}, \binom{S_j}{q_i}, \binom{\beta}{e}) = \binom{S_j}{e}$ and $f(\binom{S_j}{q_i}, \binom{\alpha}{e}, \binom{\beta}{e}) = \binom{\alpha}{q_l}$

for $\forall\alpha,\beta\in K$.

(3) In case $q_i S_j L q_l \in P$,

$f(\binom{\alpha}{e}, \binom{S_j}{q_i}, \binom{\beta}{e}) = \binom{S_j}{e}$ and $f(\binom{\alpha}{e}, \binom{\beta}{e}, \binom{S_j}{q_i}) = \binom{\beta}{q_l}$

for $\forall\alpha,\beta\in K$.

(4) In the other case where at least one of x and y both in $Q\cup\{e\}$ equals e,

$f(\binom{\alpha}{x}, \binom{\beta}{e}, \binom{\gamma}{y}) = \binom{\beta}{e}$ for $\forall\alpha,\beta\in K$.

If there exists, in an initial state configuration, one and only one element of Q(in a form $\binom{\alpha}{q}$, $q\in Q$) then the system of half-infinite iterative automata composed of the above mentioned (S,f) with $b_l=\binom{B}{e}$ simulates step by step the behavior of the given TM.

Though the system of infinite iterative automata thus constructed is indeed a simulator of a TM under the above mentioned restriction, what happens if the restriction is removed? That is, we should consider a case where there are many elements of Q in an initial state configuration, which is necessary in treating the system generally. The situation in question corresponds to the case where there are many TM's defined by the same P scanning a tape at different squares independently. In such a case, the T's may collide each other, which will result in confusion.

In order to avoid the circumstances, it is assumed a priori that the actions of the left side cell are dominant when two TM's are side by side in our system of half-infinite iterative automata. For example, if $q_i \alpha R q_l$ and $q_j \beta L q_k$ are in P, f should be defined as $f((\underset{q_i}{\alpha}), (\underset{q_j}{\beta}), (\underset{e}{\gamma}))=(\underset{q_l}{\beta})$ and $f((\underset{e}{\gamma}), (\underset{q_i}{\alpha}), (\underset{q}{\beta})=(\underset{e}{\alpha})$ for $^{\forall}\gamma \in K$.

A cell (S,f) constructed in such a manner is said to correspond to a TM.

4.5.3. Halting Problem of TM and ID-Configuration

Here we explain the relation between the halting problem of TM and the decision problem whether given $\alpha_1\alpha_2 \ldots . \alpha_k$ is an ID-configuration or not.

First, let us construct a TM T' from an arbitrarily given TM T as follows:

Behavior of T':

T' is assumed to be scanning the leftmost symbol at the start of its action, and then moves to the right until the first block between the symbol λ's is found. After finding the block, T' acts like T regarding the block as its initial tape. While simulating the behavior of T, if T' happens to scan the left side λ, T' shifts the contents of the block (including the right side λ) one square to the right. Thus T' does not go to the left of the leftmost λ in simulating T. If T' happens to scan the right side λ, it only shifts the λ one square to the right ignoring what happens to the right part of the tape. If and only if T halts, then T' moves to the left through the leftmost λ and halts at the leftmost square of the tape.

When T' is given an initial tape containing more than one λ at anywhere except the leftmost square of the tape, then the following lemma holds.

Lemma 4.12.

Under the assumption of the dominating lefthand side head of TM, when there are more than one head of TM, the following relation holds.

T halts for some initial tape. $\Leftrightarrow$ T' returns back to the leftmost square for some initial tape.

Proof

$\Rightarrow$ If T halts for an initial tape $\beta_1\beta_2\ldots\beta_l$, then T' with initial tape $\alpha_1\alpha_2\ldots\alpha_k\lambda\beta_1\beta_2\ldots\beta_l\lambda$ where $\alpha_i \neq \lambda$ $(i=1, 2, \ldots, k)$, returns back to the leftmost square by the construction of T'.

$\Leftarrow$ Let there be in general more than one head of T'. The head scanning the leftmost square of the initial tape moves to the right erasing the encountered heads if any (by the assumption of the dominating lefthand side head) until the first λ is found. After finding the first λ, it still moves to the right erasing the encountered heads if any until the second λ is found, and then it returns to one square right to the first λ, beginning the simulation of T. Accordingly, the head returns back to the leftmost square of the tape only if T halts for the tape contained between the two λ's. The behavior of heads if any scanning the right part to the second λ does not influence the simulation of T because the heads in question can move to the left through the second λ only in halt state of T which merely propagates to the left without rewriting the symbols on the tape and eventually vanishes by encountering the above mentioned head simulating T. (Q.E.D.)

Given any TM T, let us construct a TM T' from T and consider a cell (S,f) corresponding to T'. Let $\alpha_1\alpha_2\ldots\alpha_k$ denote an initial state configuration for systems composed of the above mentioned cells, where $\alpha_1 \in (K-\{\lambda\})\times q_1$ (q_1 is the initial state of T') and $\alpha_i \in (K-\{\lambda\})\times e$

(i=2,3,....,k). Then, we have the following

Lemma 4.13.

$P_i(\alpha_1\alpha_2....\alpha_k)=P_i(\alpha_1\alpha_2....\alpha_k\beta)$ for $^\forall\beta\in S^*$ iff any head of T' never returns back to the leftmost square of a tape with initial configurations corresponding to $\alpha_1\alpha_2....\alpha_k\beta$ for $^\forall\beta\in S^*$.

Proof

Let $\alpha_1'\alpha_2'....\alpha_k'$ be an initial tape for T' where α_i' corresponds to the symbol part of α_i(i.e., $\alpha_i=\binom{\alpha_i'}{x}$, $x\in Q\cup\{e\}$). By the assumption, α_i' does not equal to λ for i=1,2,...,k, and the head of T' goes on moving to the right. Accordingly, $P_i(\alpha_1\alpha_2....\alpha_k)=\alpha_1\binom{\alpha_1'}{e}\binom{\alpha_1'}{e}....\binom{\alpha_1'}{e}.....$

On the other hand, $P_i(\alpha_1\alpha_2....\alpha_k\beta)=\alpha_1\binom{\alpha_1'}{e}\binom{\alpha_1'}{e}....\binom{\alpha_1'}{e}....$ iff any head of T' never returns to the leftmost square of the tape. (Q.E.D.)

By Lemmas 4.12 and 4.13, we have

Theorem 4.7.

Let (S,f) and $\alpha_1\alpha_2....\alpha_k$ be as in Lemma 4.13. $P_i(\alpha_1\alpha_2....\alpha_k)=P_i(\alpha_1\alpha_2....\alpha_k\beta)$ for $^\forall\beta\in S^*$ iff T does not halt for any initial tape.

4.5.4. Decision Problems

First, for (S,f) and $\alpha_1\alpha_2....\alpha_k$ used in Theorem 4.7, the converse to Lemma 4.11 is shown to hold.

Lemma 4.14.

Let (S,f) and $\alpha_1\alpha_2....\alpha_k$ be as in Theorem 4.7. $\alpha_1\alpha_2....\alpha_k$ is a D-configuration if $P_i(\alpha_1\alpha_2....\alpha_k)=P_i(\alpha_1\alpha_2....\alpha_k\beta)$ for $^\forall\beta\in S^*$.

Proof

Assume that $\alpha_1\alpha_2....\alpha_k$ is not a D-configuration. Then, there exists a configuration $\beta_1\beta_2....\beta_n$ such that $P(\alpha_1\alpha_2....\alpha_k)\neq P(\alpha_1\alpha_2....\alpha_k\beta_1\beta_2....\beta_n)$, where b_r is assumed to be $\binom{B}{e}$. By a similar reasoning as in the proof of Lemma 4.13, $P(\alpha_1\alpha_2....\alpha_k)=\alpha_1\binom{\alpha_1'}{e}\binom{\alpha_1'}{e}....\binom{\alpha_1'}{e}.....$ Thus, $P(\alpha_1\alpha_2....\alpha_k)\neq P(\alpha_1\alpha_2....\alpha_k\beta_1\beta_2....\beta_n)$ iff given $\alpha_1'\alpha_2'....\alpha_k'\beta_1'\beta_2'....\beta_n'$ as an initial tape for T', the head never goes off to the right of the square containing β_n', and does return back to the leftmost square of the tape. For the head to return back to the leftmost square, it is necessary that there exist at least two λ's in $\beta_1'\beta_2'....\beta_n'$ and that T halts for the leftmost part of $\beta_1'\beta_2'....\beta_n'$ packed between two λ's as an initial tape. If this condition holds, then $P_i(\alpha_1\alpha_2....\alpha_k)\neq P_i(\alpha_1\alpha_2....\alpha_k\beta_1\beta_2....\beta_n)$, which contradicts the hypothesis of the lemma. (Q.E.D.)

By Lemmas 4.11 and 4.14, we have

Theorem 4.8.

Let (S,f) and $\alpha_1\alpha_2....\alpha_k$ be as in Theorem 4.7. Then, $\alpha_1\alpha_2....\alpha_k$ is a D-configuration iff it is an ID-configuration.

As is well known, given an arbitrary TM T, it is undecidable whether or not there exists a tape for which T halts, that is, whether T does not halt for all initial tapes. Accordingly, we are in a position to state the following results.

Theorem 4.9.

Given an arbitrary cell (S,f) corresponding to a TM, there exists a configuration $\alpha_1\alpha_2....\alpha_k \in S^*$ such that it is undecidable whether $P(\alpha_1\alpha_2....\alpha_k)=P(\alpha_1\alpha_2....\alpha_k\beta)$ for $^\forall\beta \in S^*$.

Theorem 4.10.

Given an arbitrary cell (S,f) and a configuration $\alpha_1\alpha_2....\alpha_k \in S^*$, it is undecidable whether $\alpha_1\alpha_2....\alpha_k$ is a D-configuration or not.

4.6. Concluding Remarks

We proposed a classification method of the systems of one-dimensional iterative automata with respect to their (right to left) information transmission capability and revealed some characteristics of each class.

Although generalizations and variations of the problem are possible (e.g., as to the higher dimensional systems, boundary conditions, locations of output cells, and so forth), we confined ourselves to the model presented in view of a clear presentation of the concepts and methods of our problem.

It should be noted that the unsolvability result in 4.5 does not directly imply the unsolvability of the classification problem, which is shown elsewhere [42]. In passing, it is decidable by Theorem 4.1 that given (S,f), b_l, b_r, and a positive integer n, whether or not the system belongs to Class I with k-value n.

We have been trying the classification of $\{A_{(b_l,b_r)}(S,f)\}$ for $S=\{0,1\}$ and $b_l=b_r=1$. It is rather hard to accomplish the task even for this simplest case. By virtue of the lemmas and the theorems in this chapter, and of some ingenious calculations, we have classified 200 of the total 256 kinds of 2-state cells; 124 cells are in C_I, 47 cells are in C_{II}, and 29 cells in C_{III}. And of the cells which are known to be in C_I, the maximum k-value equals 7. The detailed classification results are listed in the Appendix A.

CHAPTER 5

INDICES CONCERNING CONTROLLABILITY OF THE SYSTEMS [28]

5.1. Introduction

In the studies of the systems of iterative automata, analysis and synthesis of the systems have the two main features; the former problem is to seek what kinds of characteristics the systems have or how a system behaves when the component cell automaton is given, while the latter is to show how to construct a component cell automaton when some characteristics or behaviors of the systems are specified. In both cases, the key point is what should be treated as "characteristics or behaviors of the systems" and how to represent them. For example, we have adopted in the previous chapters 'regularity of certain state sets' and 'information transmission capability' of the systems as such features. In this chapter, we investigate the problem of controlling the systems from the outside of them.

It is fundamental in view of control as well as finite automata theory, whether an automaton is strongly-connected or simply-connected. And if a particular state is specified, whether every state may be reached from the state (reachability) and whether the state may be reached from every state (controllability) are also the important marks for an automaton.

Here in this chapter we define indices of the systems of one-dimensional iterative automata, concerning strongly- and simply-connectedness, reachability, and controllabitity by the limits how these properties are maintained when a cell automaton is added one by one to a system. Relations between an index and a particular state, relations among the indices and the time steps needed for the control are revealed. Examples of some values of the indices are also shown.

5.1.1. Definitions

Let $M=(X,Q,\lambda)$ denote a finite state automaton (without outputs), where X and Q are finite nonempty sets (of inputs and of states, respectively) and λ is a mapping from $Q \times X$ into Q (state transition function). Let X^* be the free semigroup with identity generated by X. Then λ may be easily extended to a mapping from $Q \times X^*$ into Q. That is, by repeated applications of $\lambda(q,x'x'')=\lambda(\lambda(q,x'),x'')$, where $q \in Q$ and $x',x'' \in X^*$.

A state $q' \in Q$ is said to be *reachable* from a state $q \in Q$ (written $qR'q'$) if there exists an input sequence $x \in X^*$ that carries q to q', i.e., $\lambda(q,x)=q'$. Let R(q) denote the set of all reachable states from q, i.e., $R(q)=\{q' \mid qR'q'\}$.

Definition 5.1. (*Strongly-connectedness of M*)
M is strongly-connected iff $q' \in R(q)$ for $^{\forall}q \in Q$ and $^{\forall}q' \in Q$.

The strongly-connectedness is also defined for a pair of states in Q. The states q and q' are said to be strongly-connected if $qR'q'$ and $q'R'q$, which is denoted by $q\ R_{st}\ q'$. R_{st} is an equivalence relation on Q, and whose index (the cardinality of the quotient set Q/R_{st}) will be written as $m_{st}(M)$. The above mentioned definition of the strongly-connectedness of M is equivalent to $m_{st}(M)=1$.

Definition 5.2. (*Simply-connectedness of M*)
M is simply-connected iff for $^{\forall}q \in Q$ and $^{\forall}q' \in Q$, there exists a sequence of states $q_1 q_2 \ldots q_p$ ($q_1=q$, $q_p=q'$) such that $q_{i+1} \in R(q_i)$ or $q_i \in R(q_{i+1})$ ($i=1,2,\ldots,p-1$) holds.

Now, the relation R' is reflexive and transitive, but not symmetric. So, let R_{sp} be the equivalence relation on Q derived from R', and let the index of R_{sp} be $m_{sp}(M)$. The definition of the simply-connectedness of M is equivalent to $m_{sp}(M)=1$.

In the above definitions, each state of M has equal weight in that no specific state plays a particular role. But in the following definitions of reachability and controllability, they are made with respect to some particular state.

Definition 5.3. *(Reachability of M with respect to q_s)*

M is reachable with respect to q_s for a given state $q_s \in Q$ iff $R(q_s)=Q$. Concerning the reachability of M, the state q_s is referred to as an initial state of M.

Definition 5.4. *(Controllability of M with respect to q_d)*

M is controllable with respect to q_d for a given state $q_d \in Q$ iff $q_d \in R(q)$ for $\forall q \in Q$. Concerning the controllability of M, the state q_d is referred to as a target state.

Now, let us consider an iterative automaton of length n with a mixed boundary condition. Each cell and its behavior are defined by (S,f) as usual. The left boundary condition is fixed to b_l and the right boundary condition is variable, which is considered to be an input. (See Fig.5.1.)

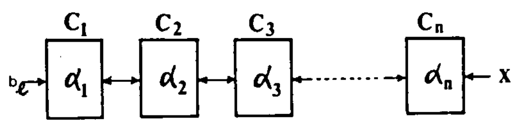


Fig. 5.1. Iterative automaton of length n with a mixed boundary condition

For fixed n, the automaton will be denoted by $A_{b_l}(S,f,n)$ or simply $A_{b_l}(n)$ when (S,f) is understood. Formally, $A_{b_l}(S,f,n)$ is a finite state automaton represented by a triple (S^n,S,F_{b_l}) where

F_{b_l} is a global transition function of an iterative automaton composed of cells (S,f). Thus, we are able to discuss the strongly-connectedness of $A_{b_l}(S,f,n)$, the controllability of $A_{b_l}(S,f,n)$ with respect to some target state, and so forth. The system of iterative automata under consideration is $A_{b_l}(S,f)=\{A_{b_l}(S,f,n)\,|\,n=1,2,3,\ldots.\}$. We are concerned how the above mentioned four properties of a finite state automaton reveal themselves if we vary n, the number of cells in the system of iterative automata.

When we are dealing with $A_{b_l}(n)$, the corresponding R', R, R_{st}, and R_{sp} are distinguished by a superscript n, as R'^n, R^n, $R_{st}{}^n$, and $R_{sp}{}^n$, respectively.

5.2. Index of Strongly-Connectedness

Regarding the strongly-connectedness of a system of iterative automata, the following simple theorem is of fundamental importance.

Theorem 5.1.

For any given (S,f,b_l), the following relation holds.

$$1\leqq m_{st}(A_{b_l}(n))\leqq m_{st}(A_{b_l}(n+1)) \qquad n=1,2,\ldots..$$

As $A_{b_l}(n)$ is not generally equal to a homomorphic image of $A_{b_l}(n+1)$, Theorem 5.1 seems to be not self-evident.

Proof

The left hand side of the inequality is obvious.

Let $[\alpha^{n+1}]_{R_{st}{}^{n+1}}$ be such a block of the partition defined by $R_{st}{}^{n+1}$ on S^{n+1} of $A_{b_l}(n+1)$ that contains $\alpha^{n+1}=\alpha_1\alpha_2\ldots\alpha_{n+1}(\alpha_i\in S$, i=1,2,....,n+1). Let d denote an operator which acts on a configuration of any length and produces a shorter configuration by deleting the rightmost element of the original configurations For example, $d(\alpha_1\alpha_2\ldots\alpha_n\alpha_{n+1})=\alpha_1\alpha_2\ldots\alpha_n$. For a set of configurations B, the operator is defined by $dB=\{x\,|\,dy=x,\ {}^{\exists}y\in B\}$.

Let $\alpha^{n+1}=\alpha_1\alpha_2\ldots\alpha_{n+1}$ and $\beta^{n+1}=\beta_1\beta_2\ldots\beta_{n+1}$ (α_i, $\beta_i\in S$, $i=1,2,\ldots,$ $n+1$). In a system of iterative automata, if $\alpha^{n+1}R'^{n+1}\beta^{n+1}$, then $d\alpha^{n+1}R'^{n}d\beta^{n+1}$ holds. Thus, if $\beta^{n+1}\in[\alpha^{n+1}]_{R_{st}}n+1$, which is equivalent to $\alpha^{n+1}R'^{n+1}\beta^{n+1}$ and $\beta^{n+1}R'^{n+1}\alpha^{n+1}$, then $d\alpha^{n+1}R'^{n}d\beta^{n+1}$ and $d\beta^{n+1}R'^{n}d\alpha^{n+1}$ hold. So we have $d\beta^{n+1}\in[d\alpha^{n+1}]_{R_{st}}n$, which means that $d[\alpha^{n+1}]_{R_{st}}n+1$ $\subset[d\alpha^{n+1}]_{R_{st}}n$. Now we can define a mapping ϕ_{n+1} from $A_{b_l}(n+1)/R_{st}n+1$ into $A_{b_l}(n)/R_{st}^{\ n}$ by $\phi_{n+1}([\alpha^{n+1}]_{R_{st}}n+1)=[d\alpha^{n+1}]_{R_{st}}n$.

Next, let us consider $[\alpha^n]_{R_{st}}n$ for an arbitrarily fixed $\alpha^n=\alpha_1\alpha_2$ $\ldots\alpha_n$. By the afore mentioned result, $d[\alpha^n\alpha]_{R_{st}}n+1\subset[\alpha^n]_{R_{st}}n$ holds for ${}^{\forall}\alpha\in S$. Thus, ϕ_{n+1} is an onto mapping and the theorem is established. The equality of the right hand side of the relation holds iff ϕ_{n+1} is one to one. (Q.E.D.)

As corollaries to Theorem 5.1, we have the followings.

Corollary 5.1.

Given (S,f,b_l), if $A_{b_l}(n)$ is strongly-connected, then so is $A_{b_l}(n-1)$, where $n\geqq 2$.

Corollary 5.2.

Given (S,f,b_l), if $A_{b_l}(n)$ is strongly-connected, then so is $A_{b_l}(k)$ $(1\leqq k\leqq n)$, and if $A_{b_l}(n)$ is not strongly-connected, then $A_{b_l}(k)$ $(k\geqq n)$ is not so either.

By this corollary, we are able to see that there may exist a critical number of cells such that the iterative automata of length up to the number are all strongly-connected and if we add still one cell to the automaton composed of the critical number of cells, the automaton thus constructed is no longer strongly-connected and never afterwards. The number is defined as the index of strongly-connectedness of a system of iterative automata as follows.

Definition 5.5. (*Index of strongly-connectedness*)

In a system of iterative automata $A_{b_l}(S,f)$, if $A_{b_l}(n)$ is strongly-connected and $A_{b_l}(n+1)$ is not, then n is called the index of strongly-connectedness of the system, and is written $N_{st}(S,f,b_l)$ or simply N_{st}. $N_{st}=0$ iff $A_{b_l}(1)$ is not strongly-connected, and $N_{st}=\infty$ iff $A_{b_l}(n)$ is strongly-connected for any positive integer n.

$N_{st}=\infty$ iff $A_{b_l}(1)$ is strongly-connected and ϕ_{n+1} is one to one $(n=1,2,3,\ldots)$. It can be easily seen that ϕ_{n+1} is one to one iff $\alpha^n\alpha$'s for all $\alpha \in S$ fall into the same block of $A_{b_l}(n+1)/R_{st}^{n+1}$. Though a more general characterization of a state transition function f that meets the condition is open, we have the following sufficient condition.

Theorem 5.2.

Given (S,f), if $f(\alpha,\beta,\gamma)\neq f(\alpha,\beta,\gamma')$ for any α,β,γ, and $\gamma'(\gamma\neq\gamma')$ in S, then $N_{st}(S,f,b_l)=\infty$ for any $b_l \in S$.

Proof

For any (S,f) satisfying the condition of the theorem, it suffices to show that $A_{b_l}(n)$ is strongly-connected for any positive integer n, which will be done by mathematical induction..

$A_{b_l}(1)$ is obviously strongly-connected.

Let $A_{b_l}(n)$ be strongly-connected. To show that $A_{b_l}(n+1)$ is strongly-connected, it suffices to see that ϕ_{n+1} is one to one. Consider arbitrarily fixed $\alpha^n \in S^n$ and any α, $\beta \in S$. By the hypothesis of the induction, there exists a sequence $x=x_1x_2\ldots x_t \in S^*$ such that $F_{b_l}(\alpha^n, \alpha x)=\alpha^n$. Then, by the condition of the theorem, there exists a sequence $y=y_1y_2\ldots y_ty_{t+1}$ such that $f(\alpha_n{}^i, x_i, y_{i+1})=x_{i+1}$ $(i=0,1,2,$

$\ldots .t)$, where $x_0=\alpha$, $x_{t+1}=\beta$, and $\alpha_n^{\ i}$ is the rightmost state of $F_{b_l}(\alpha^n, \alpha x_1 x_2 \ldots x_{i-1})$. Thus, $F_{b_l}(\alpha^n\alpha, y)=\alpha^n\beta$, i.e., $\alpha^n\alpha R'^{n+1}\alpha^n\beta$. As α and β are arbitrarily chosen, $\alpha^n\beta R'^{n+1}\alpha^n\alpha$ also holds, and $\alpha^n\alpha$ and $\alpha^n\beta$ belong to the same block of $A_{b_l}(n+1)/R_{st}^{n+1}$, which shows that ϕ_{n+1} is one to one. (Q.E.D.)

5.3. Indices of Reachability and Controllability

In the previous section, an index of strongly-connectedness for $A_{b_l}(S,f)$ has been defined. Can we validly define an index of reachability or controllability similar to that of strongly-connectedness for a system of iterative automata? That is, can a theorem similar to Theorem 5.1 be established?

The question can not be answered generally without referring to some particular (initial or target) state configurations. First, we make the following definition.

Definition 5.6. (*Admissible state sequence*)

Consider a sequence of state configurations $\Gamma_\alpha=\{\alpha_1,\alpha_2,\ldots.\alpha_n,\ldots\}$, where α_n is a state configuration of length n. For a given $A_{b_l}(S,f,n)$, let α_n be an initial (a target) state configuration. Γ_α is said to be admissible for reachability (controllability) iff reachability (controllability) of $A_{b_l}(n+1)$ implies that of $A_{b_l}(n)$.

By means of the above mentioned admissible state sequences for reachability and controllability, we are in a position to define the corresponding indices as follows;

Definition 5.7. (*Indices of reachability and controllability*)

In a system of iterative automata $A_{b_l}(S,f)$ with an admissible state sequence $\Gamma_\alpha=\{\alpha_1,\alpha_2,\ldots,\alpha_n,\ldots.\}$, if $A_{b_l}(n)$ is reachable (controllable) with respect to α_n and $A_{b_l}(n+1)$ is not so with respect to α_{n+1}, then n is called the index of reachability (controllability) of the system with respect to Γ_α, and is written $N_r(\Gamma_\alpha;S,f,b_l)$ ($N_c(\Gamma_\alpha;S,f,b_l)$) or simply $N_r(\Gamma_\alpha)$ ($N_c(\Gamma_\alpha)$). $N_r(\Gamma_\alpha)=0$ ($N_c(\Gamma_\alpha)=0$) iff $A_{b_l}(1)$ is not reachable (controllable) with respect to α_1, and $N_r(\Gamma_\alpha)=\infty$ ($N_c(\Gamma_\alpha)=\infty$) iff $A_{b_l}(n)$ is reachable (controllable) with respect to α_n for any positive integer n.

For a given $A_{b_l}(S,f)$, there exist various kinds of admissible state sequences. For example, there is such a sequence that is admissible for reachability but not for controllability. So, in what follows, we restrict ourselves to the 'standard' sequences instead of taking into account all the possible admissible sequences for reachability or controllability. As is shown later, the restriction loses almost nothing essential.

Definition 5.8. *(Standard state sequence)*

A sequence of state configurations $\Gamma_\alpha=\{\alpha_1,\alpha_2,\dots,\alpha_n,\dots\}$ is standard iff $\alpha_{n+1}=\alpha_n\alpha^{(n+1)}$ ($\alpha^{(n+1)}\in S$, $n=1,2,\dots$).

In a standard sequence, a state configuration of length n+1 is obtained by adding any one of the cell states to the right of a configuration of length n. In other words, $d\alpha_{n+1}=\alpha_n$ for $n=1,2,\dots$

One of the typical standard sequences used in later examples is $\Gamma_0=\{0,00,000,\dots\dots, 000\dots0,\dots\}$.

As to the standard sequences, the following theorem holds.

Theorem 5.3.

For any $A_{b_l}(S,f)$, a standard sequence is admissible for both reachability and controllability.

Proof

It is shown that a standard sequence $\Gamma_\alpha=\{\alpha_1,\alpha_2,\dots,\alpha_n,\dots\}$

of $A_{b_l}(S,f)$ is admissible for reachability. Let $\alpha_n=\alpha_1\alpha_2\ldots\alpha_n$ and $\alpha_{n+1}=\alpha_1\alpha_2\ldots\alpha_n\alpha_{n+1}$. Assume that $A_{b_l}(n+1)$ is reachable with respect to α_{n+1}. Consider any configurations $\beta_n=\beta_1\beta_2\ldots\beta_n$ and $\beta_{n+1}=\beta_1\beta_2\ldots\beta_n\beta_{n+1}$. By the assumption, $\alpha_{n+1}\ R'^{n+1}\ \beta_{n+1}$ holds, hence $d\alpha_{n+1}\ R'^n\ d\beta_{n+1}$ i.e., $\alpha_n\ R'^n\ \beta_n$. Thus, $A_{b_l}(n)$ is reachable with respect to α_n, which is to be shown.

As for controllability, the theorem can be proved similarly. (Q.E.D.)

Note that for a fixed $A_{b_l}(S,f)$, the converse of Theorem 5.3, i.e., "a sequence admissible for both reachability and controllability is standard" is not correct. See the following counterexample where Γ_α is admissible for both reachability and controllability but not standard.

Counterexample

Let $S=\{0,1\}$ and $b_l=1$. Consider

$$f:\quad \begin{array}{c|cc} 0 & 0 & 1 \\ \hline 0 & 1 & 1 \\ 1 & 1 & 1 \end{array}, \qquad \begin{array}{c|cc} 1 & 0 & 1 \\ \hline 0 & 1 & 1 \\ 1 & 0 & 0 \end{array}, \text{ and } \Gamma_\alpha=\{0,\ 11,\ 000,\ 0000,\ldots\}.$$

Given $A_{b_l}(S,f)$, $N_r(\Gamma_\alpha)$ may be determined for each admissible sequence for reachability (not necessarily standard) Γ_α. Let $\overline{N}_r$ be the maximum of such $N_r(\Gamma_\alpha)$'s, and let $\Gamma_{\overline{\alpha}}$ be a sequence such that $N_r(\Gamma_{\overline{\alpha}})=\overline{N}_r$. Then, we have the following

Theorem 5.4.

There is a standard sequence that meets the requirements of $\Gamma_{\overline{\alpha}}$ for reachability.

Proof

Let $\overline{N}_r=n$. Then, there exists at least one configuration $\alpha_n=\alpha_1\alpha_2\ldots\alpha_n$ with respect to which $A_{b_l}(n)$ is reachable. Construct

$\Gamma_{\bar{\alpha}}=\{\alpha_1,\alpha_2,\ldots\ldots,\alpha_n,\ldots\ldots\}$ by letting $\alpha_i=\alpha_1\alpha_2\ldots\alpha_i(i=1,2,\ldots\ldots,n)$ and for $\alpha_i(i>n)$ $d\alpha_{i+1}=\alpha_i$. Thus, $\Gamma_{\bar{\alpha}}$ is standard and gives the maximum value $\overline{N}_r$. (Q.E.D.)

A similar argument is possible for controllability. Let $\overline{N}_c$ denote the maximum value of $N_c(\Gamma_\beta)$'s for a given $A_{b_l}(S,f)$ and for all admissible sequences for controllability Γ_β. $\Gamma_{\bar{\beta}}$ denotes a sequence such that $N_c(\Gamma_{\bar{\beta}})=\overline{N}_c$. Then, we have

Theorem 5.5.

There is a standard sequence that meets the requirements of $\Gamma_{\bar{\beta}}$ for controllability.

5.4. Index of Simply-Connectedness

An index of simply-connectedness may also be defined by means of the following fundamental

Theorem 5.6.

For any given (S,f,b_l), the following relation holds.

$$1\leqq m_{sp}(A_{b_l}(n))\leqq m_{sp}(A_{b_l}(n+1)),\quad n=1,2,\ldots\ldots.$$

Proof

The argument here parallels that used in the proof of Theorem 5.1. First, the left hand side of the relation is obvious. For the right hand side of the relation, we are going to define a mapping ψ_{n+1} from $A_{b_l}(n+1)/R_{sp}{}^{n+1}$ into $A_{b_l}(n)/R_{sp}{}^{n}$, and to show that it is a surjection. Let $[\alpha^{n+1}]R_{sp}{}^{n+1}$ be a block of $A_{b_l}(n+1)/R_{sp}{}^{n+1}$ containing α^{n+1}. If $\beta^{n+1}\in[\alpha^{n+1}]R_{sp}{}^{n+1}$, then there exists a sequence of states $\gamma_1,\gamma_2,\gamma_3,\ldots\ldots\gamma_p(\gamma_1=\alpha^{n+1},\ \gamma_p=\beta^{n+1})$ such that $\gamma_i R'^{n+1}\gamma_{i+1}$ or $\gamma_{i+1}R'^{n+1}\gamma_i$ holds for $i=1,2,\ldots\ldots,$ p-1. So, there exists a sequence of states $d\gamma_1, d\gamma_2, \ldots, d\gamma_p$ $(d\gamma_1 = d\alpha^{n+1}, d\gamma_p = d\beta^{n+1})$ such that $d\gamma_i R'^n d\gamma_{i+1}$ or $d\gamma_{i+1} R'^n d\gamma_i$ holds for $i=1,2,..,p-1$. Thus, $d\beta^{n+1} \in [d\alpha^{n+1}]_{R_{sp}} n$ and hence, $d[\alpha^{n+1}]_{R_{sp}} n+1 \subset [d\alpha^{n+1}]_{R_{sp}} n$.

A mapping ψ_{n+1} can be defined by $\psi_{n+1}([\alpha^{n+1}]_{R_{sp}} n+1) = [d\alpha^{n+1}]_{R_{sp}} n$.

Next, consider $[\alpha^n]_{R_{sp}} n$ for an arbitrarily fixed α^n. Then, $d[\alpha^n\alpha]_{R_{sp}} n+1 \subset [\alpha^n]_{R_{sp}} n$ for $\forall \alpha \in S$, which assures that ψ_{n+1} is an onto mapping. (Q.E.D.)

Corollary 5.3.

Given (S,f,b_l), if $A_{b_l}(n)$ is simply-connected, then so is $A_{b_l}(n-1)$, where $n \geqq 2$.

Now, we are in a position to define the index of simply-connectedness.

Definition 5.9. (*Index of simply-connectedness*)

In a system of iterative automata $A_{b_l}(S,f)$, if $A_{b_l}(n)$ is simply-connected and $A_{b_l}(n+1)$ is not, then n is called the index of simply-connectedness of the system, and is written $N_{sp}(S,f,b_l)$ or simply N_{sp}. $N_{sp}=0$ iff $A_{b_l}(1)$ is not simply-connected, and $N_{sp}=\infty$ iff $A_{b_l}(n)$ is simply-connected for any positive integer n.

5.5. Illustrative Examples

In the previous sections, we defined four indices concerning the controllability of the systems from outside world; they are of the strongly-connectedness, the reachability, the controllability, and the simply-connectedness. As an illustrative example, we use the following cell (S,f), and see how large these indices are for this case.

Let S={0,1} and b_l=1. Consider a transition function f defined by

0	0	1
0	0	1
1	1	0

and

1	0	1
0	1	1
1	1	0

A standard sequence Γ_0={0, 00, 000,......} is adopted as the initial and the target states. Figures 5.2 (1) - (6) show the transition diagrams of the systems A_{b_l} (S,f,n) for n=1 to 6, respectively. By these diagrams, the indices for the system are determined as

N_{st}=1, N_c=1, N_r=3, and N_{sp}=5.

As for the other cells with S={0,1}, the indices are listed in the Appendix B.

n=1

	0	1
0	1	0
1	1	0

St.C.
C.
R.
Sp.C.

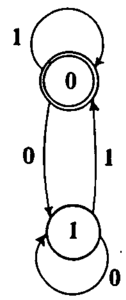
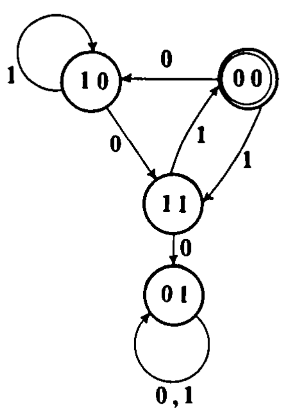


Fig.5.2(1)

n=2



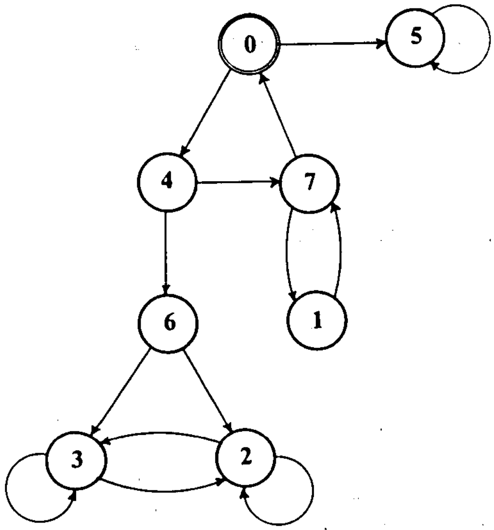
	0	1
00	10	11
01	01	01
10	11	10
11	01	00

notSt.C.
not. C.
R.
Sp.C.

Fig5.2(2)

Transition diagrams for a system of length n

n=3



	0	1
000	100	101
001	111	111
010	011	010
011	011	010
100	110	111
101	101	101
110	011	010
111	001	000

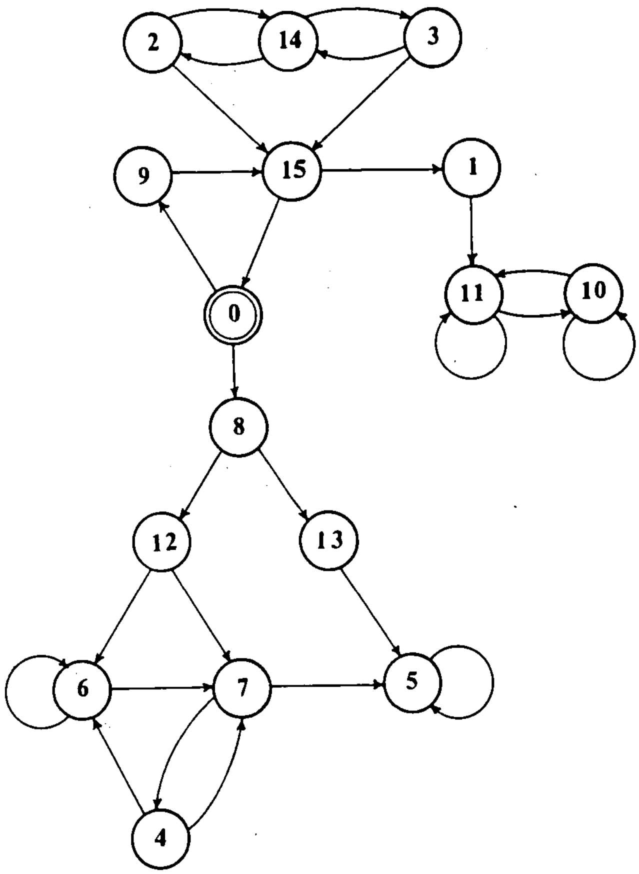
not St.C.
not C.
R.
Sp.C.

Fig.5.2 (3)

Transition diagrams for a system of length n

	0	1
0	8	9
1	11	11
2	15	14
3	15	14
4	6	7
5	5	5
6	7	6
7	5	4
8	12	13
9	15	15
10	11	10
11	11	10
12	6	7
13	5	5
14	3	2
15	1	0

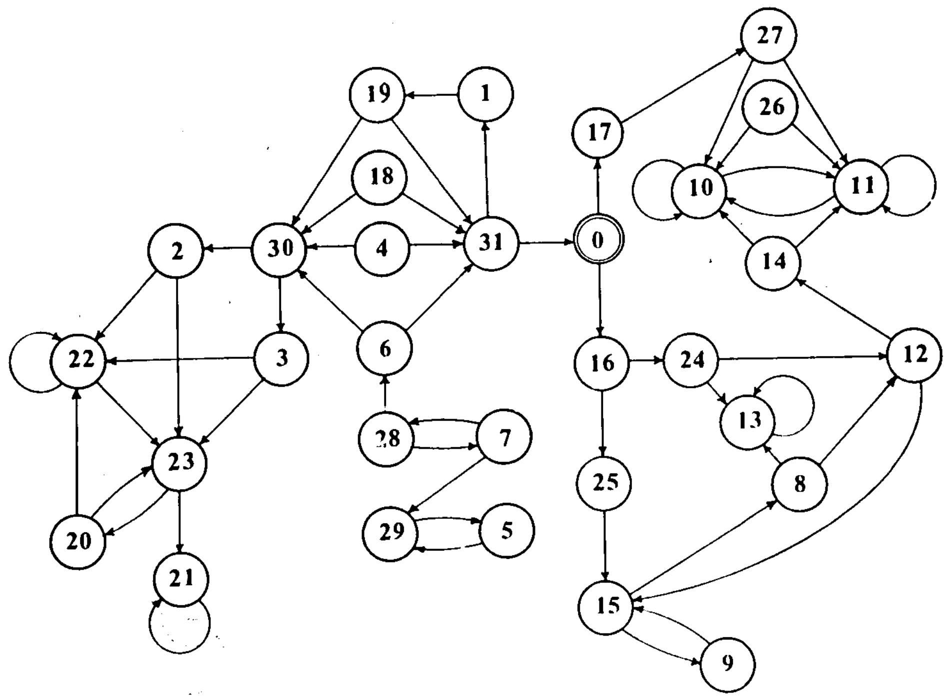
n=4



not St.C.
not C.
not R.
Sp.C.

Fig. 5.2(4)
Transition diagrams for a system of length n

n=5



not St.C.

not C.

not R.

Sp.C.

Fig.5.3(5)

Transition diagrams for a system of length n

n=6

Not St.C.
not C.
not R
not Sp.C.

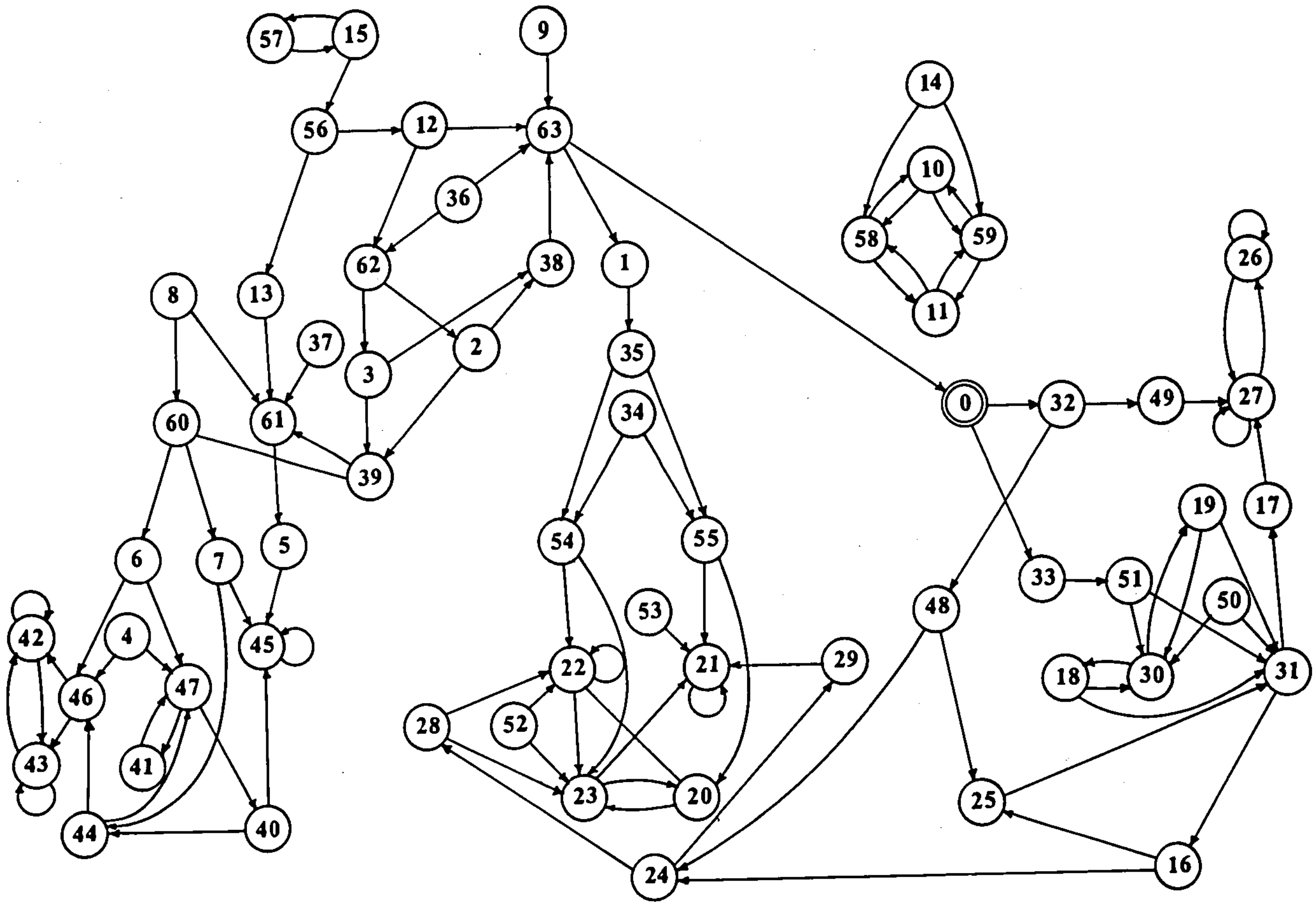


Fig.5.2(6)

Transition diagrams for a system of length n

5.6. Relations Among Indices

Let Γ_α be an arbitrary standard sequence of configurations. For any fixed $A_{b_l}(S,f)$, the following relations are self-evident.

$$N_{sp} \geqq \overline{N}_r \geqq N_r(\Gamma_\alpha) \geqq N_{st} \cdot \tag{5.1}$$

$$N_{sp} \geqq \overline{N}_c \geqq N_c(\Gamma_\alpha) > N_{st} \cdot \tag{5.2}$$

If an automaton $A_{b_l}(S,f,n)$ is reachable and controllable with respect to the same state configuration α_n, then it is at once strongly-connected. Thus,

$$N_{st} \geqq \min(N_r(\Gamma_\alpha),\ N_c(\Gamma_\alpha)). \tag{5.3}$$

Let $N_c(\Gamma_\alpha) \geqq N_r(\Gamma_\alpha)$. By (5.1) and (5.3), we have

$$N_r(\Gamma_\alpha) \geqq N_{st} \geqq \min(N_r(\Gamma_\alpha),\ N_c(\Gamma_\alpha)) = N_r(\Gamma_\alpha).$$

Therefore, $N_r(\Gamma_\alpha) = N_{st}$.

In case $N_r(\Gamma_\alpha) \geqq N_c(\Gamma_\alpha)$, we obtain by (5.2) and (5.3)

$$N_c(\Gamma_\alpha) \geqq N_{st} \geqq \min(N_r(\Gamma_\alpha), N_c(\Gamma_\alpha)) = N_c(\Gamma_\alpha).$$

So in this case, $N_c(\Gamma_\alpha) = N_{st}$. Thus, the following theorem is established.

Theorem 5.7.

For any $A_{b_l}(S,f)$ and any standard sequence Γ_α, one of the following two cases occurs:

$$N_c(\Gamma_\alpha) \geqq N_r(\Gamma_\alpha) = N_{st} \text{ or } N_r(\Gamma_\alpha) \geqq N_c(\Gamma_\alpha) = N_{st}.$$

As a corollary to Theorem 5.7, we have the following

Corollary 5.4.

$\overline{N}_r = N_r(\Gamma_{\overline{\alpha}}) \Longrightarrow N_r(\Gamma_\alpha) = N_{st}$ for any standard Γ_α or $N_c(\Gamma_{\overline{\alpha}}) = N_{st}$.

$\overline{N}_c = N_c(\Gamma_{\overline{\alpha}}) \Longrightarrow N_c(\Gamma_\alpha) = N_{st}$ for any standard Γ_α or $N_r(\Gamma_{\overline{\alpha}}) = N_{st}$.

Next, let us consider two standard sequences Γ_α and Γ_β, for which $N_r(\Gamma_\alpha) > N_r(\Gamma_\beta)$ holds. If $N_r(\Gamma_\alpha) > N_r(\Gamma_\beta) > N_{st}$, then $N_c(\Gamma_\alpha) =$

$N_c(\Gamma_\beta)=N_{st}$ by Theorem 5.7. If $N_r(\Gamma_\alpha)>N_r(\Gamma_\beta)=N_{st}$, then $N_c(\Gamma_\alpha)=N_{st}$. In both cases, $N_c(\Gamma_\beta)\geqq N_c(\Gamma_\alpha)=N_{st}$.

Similarly, if we have $N_c(\Gamma_\alpha)>N_c(\Gamma_\beta)$, then $N_r(\Gamma_\beta)\geqq N_r(\Gamma_\alpha)=N_{st}$. These relations are summerized in

Theorem 5.8.

For any $A_{b_l}(S,f)$ and any standard sequences Γ_α and Γ_β,

$$N_r(\Gamma_\alpha)>N_r(\Gamma_\beta) \Longrightarrow N_c(\Gamma_\beta)\geqq N_c(\Gamma_\alpha)=N_{st},$$

$$N_c(\Gamma_\alpha)>N_c(\Gamma_\beta) \Longrightarrow N_r(\Gamma_\beta)\geqq N_r(\Gamma_\alpha)=N_{st}.$$

5.7. Remarks on Control Steps

Hitherto, we have not touched on a problem: how many time steps does it take to transit from state α to β ? In this section, a pseudo-distance function $d(\alpha,\beta)$ is defined and some of its elementary properties are discussed.

Definition 5.10. (*Control step function*)

For a given iterative automaton $A_{b_l}(S,f,n)$, $d^n(\alpha^n,\beta^n)$ is the minimum time steps to go from α^n to β^n. In other words, $d^n(\alpha^n,\beta^n)=\min\{|x| \mid F_{b_l}(\alpha^n,x)=\beta^n,\ x\in S^*\}$. If β^n is not reachable from α^n, $d^n(\alpha^n,\beta^n)$ is assumed to be infinity. $d^n(.,.)$ is called a control step function of $A_{b_l}(S,f,n)$, several properties of which are listed below.

P.1. $d^n(.,.)$ is a pseudo-distance function:

- o $d^n(\alpha^n,\beta^n)\geqq 0$ for $\forall\alpha^n,\beta^n\in S^n$;
- o $d^n(\alpha^n,\beta^n)=0 \Longrightarrow \alpha^n=\beta^n$;
- o $d^n(\alpha^n,\tau^n)\leqq d^n(\alpha^n,\beta^n)+d^n(\beta^n,\tau^n)$ for $\forall\alpha^n,\beta^n,\tau^n\in S^n$;
- o In general, $d^n(\alpha^n,\beta^n)\neq d^n(\beta^n,\alpha^n)$.

P.2. $d^n(\alpha^n,\beta^n)<\infty \Rightarrow d^n(\alpha^n,\beta^n)\leqq m^n$ $(m=|S|)$.

P.3. $d^{n-1}(d\alpha^n,d\beta^n)\leqq d^n(\alpha^n,\beta^n)$.

P.4. For any $\alpha^n\in S^n$, there exists $\beta^n\in S^n$ such that $d^n(\alpha^n,\beta^n)\geqq n$.

Proof

Let $N^i(\alpha^n)$ denote the number of reachable states from α^n by the inputs of length i. Generally, $N^i(\alpha^n) \leqq m^i$, where m^i equals the number of possible inputs. Thus,

$$\sum_{i=0}^{n-1} N^i(\alpha^n) \leqq \sum_{i=0}^{n-1} m^i = \frac{m^n-1}{m-1} < m^n \quad (m \geqq 2).$$

Thus, α^n can not reach all the states of $A_{b_l}(n)$ whose number equals m^n, by up to (n-1) steps. (Q.E.D.)

P.5. If (S,f) is of RP type (See 2.1.), then for $^\forall\alpha^n$ and β^n there exists $\alpha \in S$ such that $d^{n+1}(\alpha^n\alpha, \beta^n\beta) = d^n(\alpha^n, \beta^n)$ for $^\forall\beta \in S$.

Proof

By P.3, $d^{n+1}(\alpha^n\alpha, \beta^n\beta) \geqq d^n(\alpha^n, \beta^n)$. To complete the proof, it is necessary to show that $d^{n+1}(\alpha^n\alpha, \beta^n\beta) \leqq d^n(\alpha^n, \beta^n)$. Assume that $F_{b_l}(\alpha^n, x) = \beta^n$, $x = x_1x_2\ldots x_t$ and $t = d^n(\alpha^n, \beta^n)$. Let $\alpha = x_1$. Then, as in the proof of Theorem 5.2, there exists a sequence $y = y_1y_2\ldots y_t$ for each $\beta \in S$ such that $F_{b_l}(\alpha^n\alpha, y) = \beta^n\beta$, which implies $d^{n+1}(\alpha^n\alpha, \beta^n\beta) \leqq d^n(\alpha^n, \beta^n)$. (Q.E.D.)

P.6. If (S,f) is of RP type, $d^n(\alpha^n, \beta^n) \leqq n$ for $^\forall\alpha^n, \beta^n \in S^n$.

Proof

Let $\alpha^n = \alpha_1\alpha_2\ldots\alpha_{n-1}\alpha_n$ and $\alpha^{n-1'} = F_{b_l}(d\alpha^n, \alpha_n)$.

First, let us show that $d^n(\alpha^n, \beta^n) \leqq d^{n-1}(\alpha^{n-1'}, d\beta^n)+1$. There exists α such that $d^n(\alpha^{n-1'}\alpha, \beta^n) = d^{n-1}(\alpha^{n-1'}, d\beta^n)$ by P.5, and $d^n(\alpha^n, \beta^n) \leqq d^n(\alpha^n, \alpha^{n-1'}\alpha) + d^n(\alpha^{n-1'}\alpha, \beta^n)$ by P.1. As (S,f) is of RP type, $d^n(\alpha^n, \alpha^{n-1'}\alpha) = 1$ and the above mentioned inequality is shown to hold.

As to the proposition that $d^n(\alpha^n, \beta^n) \leqq n$, it can be proved by the mathematical induction on n, noting $d^1(\alpha^1, \beta^1) = 1$ for $^\forall\alpha^1, \beta^1 \in S$. (Q.E.D.)

P.7. If (S,f) is not of RP type, there exist α^n and β^n for $n \geqq 2$

such that $d^n(\alpha^n,\beta^n)>n$.

Proof

As (S,f) is not of RP type, there exist α,β,γ, and $\gamma'(\gamma\neq\gamma')$ in S such that $f(\alpha,\beta,\gamma)=f(\alpha,\beta,\gamma')$. Let $\alpha^n=\alpha_1\alpha_2\ldots\ldots\alpha_{n-1}\alpha_n$ where $\alpha_{n-1}=\alpha$ and $\alpha_n=\beta$. A state configuration reached from α^n by the application of an input sequence $x_1x_2\ldots x_i$ will be denoted as $\alpha_1^{(i)}\alpha_2^{(i)}\ldots\alpha_n^{(i)}$ (See Fig.5.3).

$$
\begin{array}{c}
\alpha_1\ \alpha_2\ \cdots\cdots\ \alpha_{n-1}\ \alpha_n\ x_1 \\[1em]
\alpha_1'\alpha_2'\cdots\cdots\ \alpha_{n-1}'\ \alpha_n'\ x_2 \\[1em]
\vdots \\[1em]
\alpha_1^{(n)}\ \alpha_2^{(n)}\cdots\cdots\ \alpha_{n-1}^{(n)}\ \alpha_n^{(n)}
\end{array}
$$

Fig. 5.3. State configurations reached from α^n

Consider $\alpha_k^{(l)}$ $(k+l=n+1)$ $l=1,2,\ldots,n$, which is determined uniquely by α^n and x_1. By the choice of α^n, there exists at least one state (let it be β_n) that could not appear as α_n' through any choice of x_1. As for α''_{n-1}, there exists some state β_{n-1} which could not appear at the place, because α_n' may take at most (m-1) kinds of states. By a similar reasoning, there exists β_k which could not appear as $\alpha_k^{(l)}$ $(k+l=n+1)$ for $k=n, n-1,\ldots 2,1$. Let $\beta^n=\beta_1\beta_2\ldots\beta_n$, then it can't be included in the set of states that are reachable from α^n by up to n steps. (Q.E.D.)

By P.6 and P.7, we obtain a characterization of RP type cell in terms of a control step function. That is, (S,f) is of RP type iff there exists an integer $n\geq 2$ such that $d^n(\alpha^n,\beta^n)\leq n$ for $^\forall\alpha^n,\beta^n\in S^n$.

P.8. If (S,f) is of RP type, for $\forall\alpha^n$ and β^n, there exists exactly one input sequence $x^n=x_1x_2\dots x_n$ of length n such that $F_{b_l}(\alpha^n,x^n)=\beta^n$.

Proof

As in the proof of P.7, let $\alpha^n=\alpha_1\alpha_2\dots\alpha_n$ and define $\alpha_1^{(i)}\alpha_2^{(i)}\dots\alpha_n^{(i)}$ as before. The $\alpha_k^{(l)}$'s $(l+k\leqq n)$ are determined solely by α^n. In order to reach $\beta^n=\beta_1\beta_2\dots\beta_n$ at step n, $\alpha_2^{(n-1)}$ should satisfy $f(b_l,\alpha_1^{(n-1)},\alpha_2^{(n-1)})=\beta_1$, which is possible by the condition that (S,f) is of RP type. Further, such $\alpha_2^{(n-1)}$ is determined uniquely. Similarly, $\alpha_i^{(n-1)}$'s$(i=3,4,\dots,n)$ are uniquely determined one by one, and x_n is also determined uniquely so as to satisfy $f(\alpha_{n-1}^{(n-1)}, \alpha_n^{(n-1)}, x_n)=\beta_n$. Continuing these processes upward (see Fig.5.4), a unique input sequence $x^n=x_1x_2\dots x_n$ is obtained. It is evident that by this x^n, α^n goes to β^n by exactly n steps. (Q.E.D.)

$$
\begin{array}{ccccccc}
\alpha_1 & \alpha_2 & \cdots\cdots & \alpha_{n-1} & \alpha_n & & x_1 \\
\alpha_1' & \alpha_2' & \cdots\cdots & \alpha_{n-1}' & \alpha_n' & & x_2 \\
 & & \vdots & & & & \\
\alpha_1^{(n-1)} & \alpha_2^{(n-1)} & \cdots\cdots & \alpha_{n-1}^{(n-1)} & \alpha_n^{(n-1)} & & x_n \\
\beta_1 & \beta_2 & \cdots\cdots & \beta_{n-1} & \beta_n & &
\end{array}
$$

Fig. 5.4. α^n goes to β^n by n steps

5.8. Concluding Remarks

In order to investigate characteristic behaviors of the systems of one-dimensional iterative automata, we proposed four quantities which reveal how much an input sequence can control the state transitions of the systems. Those indices defined in this chapter not only seem to be interesting for its own sake but also have some connections with the other problems. For example, the indices may be helpful when we consider a realization problem of a finite state automaton in the form of our one-dimensional iterative automaton, and want to select the possible candidates of cells. The classification problem treated in the previous chapter also is, though not so direct, relevant to the indices; a relation between the information transmission capability and the controllability of the systems.

Further details of the relations among the indices, the relations between an initial state sequence and the indices, and such application topics that are mentioned above are left to be solved.

CHAPTER 6

INFORMATION MAINTENANCE CAPABILITY OF THE SYSTEMS

6.1. Introduction

It seems interesting and important to treat the behaviors of cellular systems in the presence of noise or error in some sense. There are several works which deal with fault-detection experiments in cellular logic, where two-dimensional (unilateral) combinational cells are considered. [18,49] And there has been little work concerning noises, errors, or information in the systems of iterative automata.

In this chapter, we analyze the effect of noises appeared in one-dimensional iterative automata, where the term 'noise' means that a state of a cell has suddenly been changed to another one, though the functions of each cell are normal. We are concerned with a comparison between the behavior of the system with noise and that of the system which ought to be.

As is clear from the above context, the term noise may be read as the injection of information. The difference is whether we regard the effect as desirable or not. So, in later sections, the problem will be formalized as that of information maintenance.

6.2. Definitions

Let $A_x(S,f)$ denote a system of one-dimensional iterative automata composed of cell (S,f) with an appropriate boundary condition x. S is a finite nonempty set of cell states and f is a state transition function from S^3 into S as before. The cardinality of S(denoted $|S|$)

is assumed to be m.

Let $S' \subset S$. A function g from S' into S is said to be (n_i,n_o), where n_i and n_o are positive integers such that $|S'|=n_i$ and $|g(S')| =n_o$. The set of (n_i,n_o)-functions will be denoted by $F_{(n_i,n_o)}$.

When we speak of an (n_i,n_o)-function, $m \geq n_i \geq n_o$ is evident by the definition. An $(n_i,1)$-function is a constant function and an (m,m)-function is a permutation function.

Consider an iterative automaton, belonging to a system $A_x(S,f)$, whose initial state configuration contains an (n_i,n_o)-function g at some fixed cell. The states of the other cells are assumed to be elements of S, i.e., $(n_i,1)$-functions.[37,38] The behavior of an iterative automaton in such a situation may be interpreted as follows. The cell containing g acts as if it had n_o different kinds of initial states $\{g(S')\}$. The other cells behave as usual, and information g may propagate through cells in a suitably modified form.

It is clear that if we assign an (n_i,n_o)-function as an initial state, there appears no (n'_i,n'_o)-function in the system, where $n'_i>n_i$.

There may occur two cases, which dichotomizes the systems of iterative automata according to their information maintenance capability. They are : (1) at any time step, there remains at least one (n_i,n_o)-function in the system, and (2) at some time step, there are no (n_i,n_o)-funtions in the system. If the case (1) occurs, for every (n_i,n_o)-function g and every combination of $(n_i,1)$-functions as an initial state, the system $A_x(S,f)$ is said to be *(n_i,n_o)-preserving*.

Next, we define parity-preserving characteristics. Let us assume $n_o=2$. Consider a mapping w from $S \cup F_{(n_i,2)}$ into {0,1} defined as

$$w(\alpha)=\begin{cases} 0 & \text{if } \alpha \in S \\ 1 & \text{if } \alpha \in F_{(n_i,2)}. \end{cases}$$

The function w is called a parity function when the domain is extended to $\{S \cup F_{(n_i,2)}\}^+$ as follows :

$w(\alpha\alpha)=w(\alpha)+w(\alpha) \qquad \pmod 2$, where $\alpha\in\{SUF_{(n_i,2)}\}^+$.

Let $F_x(\cdot)$ denote a global transition function of $A_x(S,f)$. If $w(\alpha)=w(F_x(\alpha))$ holds for every state configuration $\alpha\in\{SUF_{(n_i,2)}\}^+$, we say that the system is *p-preserving* (parity-preserving). The main objective of this chapter is to establish the relations between the (n_i,n_o)-preservability and the p-preservability.

First, we summarize the properties of the (n_i,n_o)-preservability.

6.3. (n_i,n_o)-Preservability

The following lemmas are easily obtained.

Lemma 6.1. Any system $A_x(S,f)$ is $(n_i,1)$-preserving.

Lemma 6.2. If a system $A_x(S,f)$ is (n_i,n_o)-preserving, it is (n_i,n'_o)-preserving where $n'_o\le n_o$.

Lemma 6.3. If a system $A_x(S,f)$ is (n_i,n_o)-preserving, it is (n'_i,n_o)-preserving where $n_i\ge n'_i\ge n_o$.

From Lemma 6.2 and Lemma 6.3, we have

Theorem 6.1. If a system $A_x(S,f)$ is (n_i,n_o)-preserving, it is (n'_i,n'_o)-preserving where $n_i\ge n'_i\ge n'_o$.

Lemma 6.4. If a system $A_x(S,f)$ is (n_i,n_o)-preserving, it is (n'_i,n_o)-preserving where $m\ge n'_i\ge n_i$.

Theorem 6.2. If a system $A_x(S,f)$ is (n_i,n_o)-preserving, it is (n'_i,n'_o)-preserving when $n_i\le n'_i\le m$ and $n'_o\le n_o$.

The results of Theorem 6.1 and Theorem 6.2 are diagramatically illustrated in Fig.6.1.

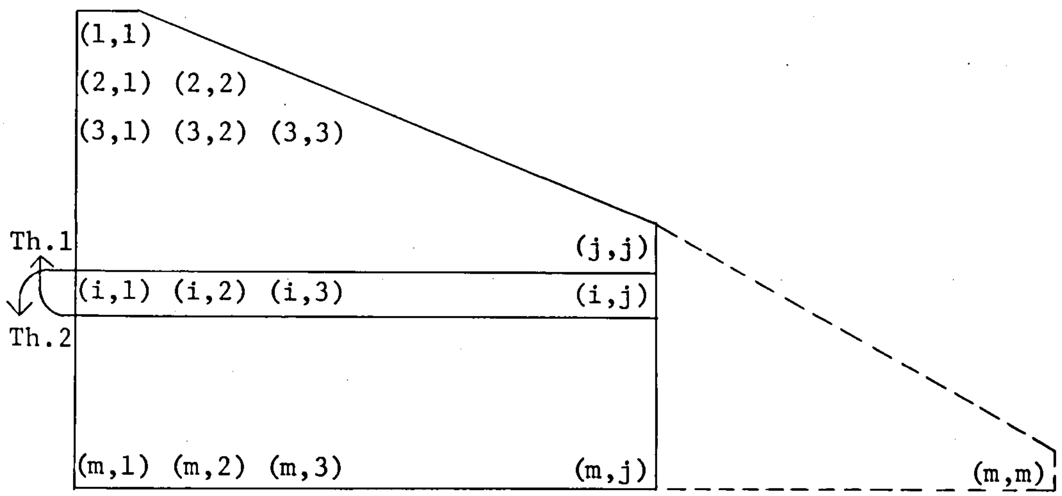


Fig.6.1. If $A_x(S,f)$ is (i,j)-preserving, it is (i',j')-preserving if (i',j') is within the border line.

6.4. P-Preservability ($|S|=2$)

In this section, we examine p-preservability when $|S|=2$, and extract necessary and sufficient conditions for $A_c(S,f)$ to be p-preserving where c denotes the cyclic boundary condition.

Let $S=\{0,1\}$. A transition function f may be represented by a table as follows :

0	0	1
0	a	c
1	b	d

,

1	0	1
0	e	g
1	f	h

$a,b,c,d,e,f,g,h \in S.$

An extended transition table is a transition table of $\begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1\\1\end{bmatrix}$, where the first and the second row of each state vector act independently under the rule f. $\begin{bmatrix}0\\1\end{bmatrix}$ and $\begin{bmatrix}1\\0\end{bmatrix}$ may be identified as x and $\overline{x}$, respectively where $x, \overline{x} \in F_{(2,2)}$. $\begin{bmatrix}0\\0\end{bmatrix}$ and $\begin{bmatrix}1\\1\end{bmatrix}$ may also be identified as constant functions 0 and 1, respectively where

$0,1\in F_{(2,1)}$. The vector notation and functional notation of the extended state will be used interchangeably.

As an example, a general extended transition table for x and $\bar{x}$ is shown in Table 6.1.

x	0	1	x	$\bar{x}$
0	$[\genfrac{}{}{0pt}{}{a}{e}]$	$[\genfrac{}{}{0pt}{}{c}{g}]$	$[\genfrac{}{}{0pt}{}{a}{g}]$	$[\genfrac{}{}{0pt}{}{c}{e}]$
1	$[\genfrac{}{}{0pt}{}{b}{f}]$	$[\genfrac{}{}{0pt}{}{d}{h}]$	$[\genfrac{}{}{0pt}{}{b}{h}]$	$[\genfrac{}{}{0pt}{}{d}{f}]$
x	$[\genfrac{}{}{0pt}{}{a}{f}]$	$[\genfrac{}{}{0pt}{}{c}{h}]$	$[\genfrac{}{}{0pt}{}{a}{h}]$	$[\genfrac{}{}{0pt}{}{c}{f}]$
$\bar{x}$	$[\genfrac{}{}{0pt}{}{b}{e}]$	$[\genfrac{}{}{0pt}{}{d}{g}]$	$[\genfrac{}{}{0pt}{}{b}{g}]$	$[\genfrac{}{}{0pt}{}{d}{e}]$

$\bar{x}$	0	1	x	$\bar{x}$
0	$[\genfrac{}{}{0pt}{}{e}{a}]$	$[\genfrac{}{}{0pt}{}{g}{c}]$	$[\genfrac{}{}{0pt}{}{e}{c}]$	$[\genfrac{}{}{0pt}{}{g}{a}]$
1	$[\genfrac{}{}{0pt}{}{f}{b}]$	$[\genfrac{}{}{0pt}{}{h}{d}]$	$[\genfrac{}{}{0pt}{}{f}{d}]$	$[\genfrac{}{}{0pt}{}{h}{b}]$
x	$[\genfrac{}{}{0pt}{}{e}{b}]$	$[\genfrac{}{}{0pt}{}{g}{d}]$	$[\genfrac{}{}{0pt}{}{e}{d}]$	$[\genfrac{}{}{0pt}{}{g}{b}]$
$\bar{x}$	$[\genfrac{}{}{0pt}{}{f}{a}]$	$[\genfrac{}{}{0pt}{}{h}{c}]$	$[\genfrac{}{}{0pt}{}{f}{c}]$	$[\genfrac{}{}{0pt}{}{h}{a}]$

Table 6.1. An extended transition table for x and $\bar{x}$.

We denote by $W(x)$ a matrix whose i-j component is $w(\alpha)$ where α is the i-j component of an extended transition table for x.

$$W(x)=[w_{ij}]_4^4,\ \text{i.e.,}\ w[\genfrac{}{}{0pt}{}{a}{e}]=w_{11},\ w[\genfrac{}{}{0pt}{}{c}{g}]=w_{12}\quad \text{etc.}$$

The following four lemmas can be easily obtained.

Lemma 6.5.

If $A_c(S,f)$ is p-preserving, $w_{11}=w_{22}$.

Proof

From $F_c(x)=[\genfrac{}{}{0pt}{}{a}{h}]$ and $F_c(x1)=[\genfrac{}{}{0pt}{}{d}{h}]\ [\genfrac{}{}{0pt}{}{e}{h}]$, we get $a\neq h$ and $d\neq e$ by the p-preserving condition. Then, $w_{11}=w[\genfrac{}{}{0pt}{}{a}{e}]=w[\genfrac{}{}{0pt}{}{\bar{h}}{\bar{d}}]=w[\genfrac{}{}{0pt}{}{d}{h}]=w_{22}$.

Lemma 6.6.

If $A_c(S,f)$ is p-preserving, then

$$w_{11}=w_{22}=1 \Longrightarrow [\genfrac{}{}{0pt}{}{a}{e}]=[\genfrac{}{}{0pt}{}{d}{h}]\ ;$$

$$w_{11}=w_{22}=0 \Longrightarrow [{a \atop e}]\neq[{d \atop h}].$$

Proof

When $w_{11}=w_{22}=1$ holds, $a\neq e$, and $d\neq h$. Further, we have $a\neq h$ and $d\neq e$ from the proof of Lemma 6.5. Then, $a=\bar{e}=d$ and $e=\bar{a}=h$. The second case can be proved similarly.

Lemma 6.7.

If $A_c(S,f)$ is p-preserving, $w_{12}=w_{21}$.

Proof

Note that $F_c(1 \times 0)=[{e \atop g}]\ [{b \atop f}]\ [{c \atop d}]$ and $F_c(0 \times 1)=[{b \atop d}]\ [{c \atop g}]\ [{e \atop f}]$.
If $b\neq f$ then ($e=g$ and $c=d$), or ($e\neq g$ and $c\neq d$) holds.

In both cases, $c\neq g$ can be established by the fact $d\neq e$ shown in the proof of Lemma 6.5. If $b=f$ then $c=g$ can be obtained similarly.

Lemma 6.8.

If $A_c(S,f)$ is p-preserving, then

$$w_{11}=w_{22}=1 \Longrightarrow [{c \atop g}]=[{b \atop f}] \ ;$$
$$w_{11}=w_{22}=0 \Longrightarrow [{c \atop g}]\neq[{b \atop f}] \ .$$

Proof

Note that $F_c(0 \times 0)=[{a \atop c}]\ [{a \atop e}]\ [{a \atop b}]$ and $F_c(1 \times 1)=[{f \atop h}]\ [{d \atop h}]\ [{g \atop h}]$.

In case $W_{11}=w_{22}=1$, $a\neq e$ and $d\neq h$ assure that $b=c$ and $f=g$.

The other case can be shown to hold similarly.

Thus, we obtained four necessary conditions for $A_c(S,f)$ to be p-preserving. They are

C.1. $\quad w_{11}=w_{22}$;

C.2. $\quad w_{11}=w_{22}=1 \Longrightarrow [{a \atop e}]=[{d \atop h}]$,

$\quad\quad w_{11}=w_{22}=0 \Longrightarrow [{a \atop e}]\neq[{d \atop h}]$;

C.3. $w_{12}=w_{21}$;

C.4. $w_{11}=w_{22}=1 \Longrightarrow \begin{bmatrix} c \\ g \end{bmatrix} = \begin{bmatrix} b \\ f \end{bmatrix}$,

$w_{11}=w_{22}=0 \Longrightarrow \begin{bmatrix} c \\ g \end{bmatrix} \neq \begin{bmatrix} b \\ f \end{bmatrix}$.

Lemma 6.5 to Lemma 6.8 are summarized as

Theorem 6.3.

If $A_c(S,f)$ is p-preserving, then C.1, C.2, C.3, and C.4 hold.

In the latter part of this section, we will show that the converse of Theorem 6.3 is also true.

Let M be a 4×4 matrix as shown in Fig.6.2.

$$M=\left(\begin{array}{cc:cc} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ \hdashline m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{array}\right)$$

Fig.6.2. Diagonal pairs of M

A pair of elements of M connected by a right (left) upward line is called a right (left) diagonal pair of M. For example, m_{11} and m_{22} is a left diagonal pair and m_{43} and m_{34} is a right diagonal pair of M. A pair of right and left diagonal pairs whose connecting lines intersect is said to correspond to each other.

Lemma 6.9.

If C.1, C.2, C.3, and C.4 hold, then the elements of each (right and left) diagonal pair of W(x) are the same.

Proof

From the conditions C.1 and C.3, we have $w_{11}=w_{22}$ and $w_{12}=w_{21}$.

As for the other pairs, see Table 6.2 below.

	In case $w_{11}=w_{22}=1$, i.e. $\left[\begin{smallmatrix}a\\e\end{smallmatrix}\right]=\left[\begin{smallmatrix}d\\h\end{smallmatrix}\right],\left[\begin{smallmatrix}c\\g\end{smallmatrix}\right]=\left[\begin{smallmatrix}b\\f\end{smallmatrix}\right]$ $a\neq e$ $d\neq h$	In case $w_{11}=w_{22}=0$, i.e. $\left[\begin{smallmatrix}a\\e\end{smallmatrix}\right]\neq\left[\begin{smallmatrix}d\\h\end{smallmatrix}\right],\left[\begin{smallmatrix}c\\g\end{smallmatrix}\right]\neq\left[\begin{smallmatrix}b\\f\end{smallmatrix}\right]$ $a=e$ $d=h$
$w_{13}=w_{24}$	$a=g \rightarrow d=a=g=f$ $d=f \rightarrow a=d=f=g$	$a=g \rightarrow d=\bar{a}=\bar{g}=f$ $d=f \rightarrow a=\bar{d}=\bar{f}=g$
$w_{31}=w_{42}$	$a=f \rightarrow d=a=f=g$ $d=g \rightarrow a=d=g=f$	$a=f \rightarrow d=\bar{a}=\bar{f}=g$ $d=g \rightarrow a=\bar{d}=\bar{g}=f$
$w_{33}=w_{44}$	$a=h \rightarrow d=a=h=e$ $d=e \rightarrow a=d=e=h$	$a=h \rightarrow d=\bar{a}=\bar{h}=e$ $d=e \rightarrow a=\bar{d}=\bar{e}=h$
$w_{14}=w_{23}$	$c=e \rightarrow b=c=e=h$ $b=h \rightarrow c=b=h=e$	$c=e \rightarrow b=\bar{c}=\bar{e}=h$ $b=h \rightarrow c=\bar{b}=\bar{h}=e$
$w_{32}=w_{41}$	$c=h \rightarrow b=c=h=e$ $b=e \rightarrow c=b=e=h$	$c=h \rightarrow b=\bar{c}=\bar{h}=e$ $b=e \rightarrow c=\bar{b}=\bar{e}=h$
$w_{34}=w_{43}$	$c=f \rightarrow b=c=f=g$ $b=g \rightarrow c=b=g=f$	$c=f \rightarrow b=\bar{c}=\bar{f}=g$ $b=g \rightarrow c=\bar{b}=\bar{g}=f$

Table 6.2. Proof of Lemma 6.9.

Lemma 6.10.

If C.1, C.2, C.3, and C.4 hold, then the equality and inequality of the elements of a right diagonal pair means the equality and inequal ity, respectively of the elements of the corresponding left diagonal pair in the extended transition table for x.

Proof

From the conditions C.2 and C.4, the Lemma is true for the upper-left part of the table for x. As to the other parts of the table, c=b holds when the elements of a right diagonal pair are the same. Then, $w_{11}=w_{22}=1$ must hold from C.1, C.3, and C.4, ensuring a=d, e=h, and g=f. Thus the elements of a left diagonal pair are the same. Conversely, if the elements of a left diagonal pair are the same, those of a right diagonal pair are shown to be the same by a similar reasoning.

Lemma 6.11.

If C.1, C.2, C.3, and C.4 hold, then the equality and inequality of the left diagonal pair of the upper-left part of the table for x means the equality and inequality, respectively of the other diagonal pairs in the table for x.

Proof

Let $[{a \atop e}]=[{d \atop h}]$. From C.2 and C.4, $[{c \atop g}]$ equals $[{b \atop f}]$.

If $[{a \atop e}]\neq[{d \atop h}]$, then $a\neq d$ by C.2.

Lemma 6.12.

$$w_{i1}=w_{i2} \iff w_{i3}=3_{i4}, \quad (i=1,2,3,4) \ ;$$

$$w_{1j}=w_{2j} \iff w_{3j}=w_{4j}, \quad (j=1,2,3,4) \quad .$$

Proof

We here show that $w_{11}=w_{12} \Longrightarrow w_{13}=w_{14}$.

As $w[{a \atop e}]=w[{c \atop g}]$, two cases must be considered; 1) $a=e$, $c=g$, and 2) $a\neq e$, $c\neq g$.

In case 1), $w[{a \atop g}]=w[{e \atop c}]=w[{c \atop e}]$. In case 2), $w[{a \atop g}]=w[{\bar{e} \atop \bar{c}}]=w[{e \atop c}]=w[{c \atop e}]$.
The other relations are similarly proved.

Lemma 6.13.

Let 2×2 matrices W_i $(i=1,2,3,4)$ be minors of a matrix $W(x)$ arranged as $W(x)=\begin{pmatrix}W_1W_2\\W_3W_4\end{pmatrix}$. If a pair of the four minor matrices is the same, the remaining pair is also the same.

Proof

Let $W_1=W_2$. Then $e=g$ and $f=h$ hold by the relations $w[{a \atop e}]=w[{a \atop g}]$ and $w[{b \atop f}]=w[{b \atop h}]$, respectively. Thus $w[{a \atop f}]=w[{a \atop h}]$, $w[{b \atop e}]=w[{b \atop g}]$, $w[{c \atop h}]=w[{c \atop f}]$, and $w[{d \atop g}]=w[{d \atop e}]$, i.e., $W_3=W_4$. The other cases can be proved similarly.

Lemma 6.15.

If C.1, C.2, C.3, and C.4 hold, possible patterns of W(x) are limited to the following 8 types.

T_1

f_9		f_{246}	
1	0	0	1
0	1	1	0
0	1	1	0
1	0	0	1

T_2

f_{111}		f_{144}	
1	0	1	0
0	1	0	1
1	0	1	0
0	1	0	1

T_3

f_{53}		f_{202}	
0	1	0	1
1	0	1	0
1	0	1	0
0	1	0	1

T_4

f_{83}		f_{172}	
0	1	1	0
1	0	0	1
0	1	1	0
1	0	0	1

T_5

f_{105}		f_{150}	
1	1	0	0
1	1	0	0
0	0	1	1
0	0	1	1

T_6

f_{15}		f_{240}	
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

T_7

f_{51}		f_{204}	
0	0	1	1
0	0	1	1
0	0	1	1
0	0	1	1

T_8

f_{85}		f_{170}	
0	0	0	0
0	0	0	0
1	1	1	1
1	1	1	1

(As for the numbers accompanying f, see Appendix.)

Proof

Utilize the results of Lemmas 6.9 and 6.12. Notice that $w_{33}=1$ (i.e., $a \neq h$) is assured by C.1 and C.3.

We have hitherto considered several properties of the extended transition table for x and W(x), under conditions C.1 to C.4. Matrices $W(\overline{x})$, W(0), and W(1) are related to W(x) by

Lemma 6.16. (i)

$$W(\overline{x}) = \begin{pmatrix} w_{11} & w_{12} & w_{14} & w_{13} \\ w_{21} & w_{22} & w_{24} & w_{23} \\ w_{41} & w_{42} & w_{44} & w_{43} \\ w_{31} & w_{32} & w_{34} & w_{33} \end{pmatrix} .$$

(ii) Under conditions C.1, C.2, C.3, and C.4 ;

$$W(0)=\begin{pmatrix} 0 & 0 & \bar{w}_{14} & \bar{w}_{14} \\ 0 & 0 & \bar{w}_{14} & \bar{w}_{14} \\ \bar{w}_{14} & \bar{w}_{14} & 0 & 0 \\ \bar{w}_{14} & \bar{w}_{14} & 0 & 0 \end{pmatrix}, \quad W(1)=\begin{pmatrix} 0 & 0 & \bar{w}_{13} & \bar{w}_{13} \\ 0 & 0 & \bar{w}_{13} & \bar{w}_{13} \\ \bar{w}_{13} & \bar{w}_{13} & 0 & 0 \\ \bar{w}_{13} & \bar{w}_{13} & 0 & 0 \end{pmatrix}.$$

(in case $w_{11}=w_{22}=1$)

$$W(0)=W(1)=\begin{pmatrix} 0 & 0 & w_{13} & w_{13} \\ 0 & 0 & w_{13} & w_{13} \\ \bar{w}_{13} & \bar{w}_{13} & 0 & 0 \\ \bar{w}_{13} & \bar{w}_{13} & 0 & 0 \end{pmatrix}.$$

(in case $w_{11}=w_{22}=0$)

Proof Omitted.

Now we are in a position to show the converse of Theorem 6.3. That is, we have

Theorem 6.4.

If C.1, C.2, C.3, and C.4 hold, then $A_c(S,f)$ is p-preserving.

Proof

An outline of the proof is shown as follows ; By Lemma 6.15, we know that possible patterns of W(x) are limited only to the 8 types, if the conditions hold. Further, we can get the corresponding $W(\bar{x})$, W(0), and W(1) by Lemma 6.16.

Thus, it suffices to see the p-preserving property for all the 8 types of W(x). We show it by mathematical induction with respect to the number of cells in a system. As a typical example, the type T_1 of W(x) listed in Lemma 6.15 is used. That is, we treat the following four matrices;

$$W(x)=\begin{pmatrix}1&0&0&1\\0&1&1&0\\0&1&1&0\\1&0&0&1\end{pmatrix},\qquad W(\bar{x})=\begin{pmatrix}1&0&1&0\\0&1&0&1\\1&0&1&0\\0&1&0&1\end{pmatrix},$$

$$W(0)=\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix},\ \text{and}\qquad W(1)=\begin{pmatrix}0&0&1&1\\0&0&1&1\\1&1&0&0\\1&1&0&0\end{pmatrix}.$$

Let $f(\alpha,\beta,\gamma)=\beta'$. The weight of β' is calculated by

$w(\beta')=w(\beta)$ when $\beta=0$,

$w(\beta')=w(\alpha)+w(\gamma)+w(\beta)$ when $\beta=1$,

$w(\beta')=b(\alpha)+b(\gamma)+w(\beta)$ when $\beta=x$, and

$w(\beta')=a(\alpha)+a(\gamma)+w(\beta)$ when $\beta=\bar{x}$,

where a and b are characteristic functions for $\{0,x\}$ and $\{0\ \bar{x}\}$, respectively. For example, $a(0)=1$, $a(\bar{x})=0$ etc.. Note that $a(\alpha)+b(\alpha)=w(\alpha)$ for any $\alpha\in\{0,1,x,\bar{x}\}$. Then we can see that $w(\beta')$ is represented by a single formula as

$$\begin{aligned}w(\beta')&=(1+a(\beta))\ (a(\alpha)+a(\gamma))+(1+b(\beta))(b(\alpha)+b(\gamma))+w(\beta)\\&=w(\alpha)+w(\beta)+w(\gamma)+a(\beta)(a(\alpha)+a(\gamma))+b(\beta)(b(\alpha)+b(\gamma)).\end{aligned}$$

If there exists only one cell in the system, the p-preservability is assured by the 3-3 elements of $W(x)$ and $W(\bar{x})$. Assume there are r cells, and the p-preservability holds. In other words, $w(\alpha_1\alpha_2\ldots\alpha_r)=w(\alpha'_1\alpha'_2\ldots\alpha'_r)$ for any $\alpha_1\alpha_2\ldots\alpha_r$ and $\alpha'_1\alpha'_2\ldots\alpha'_r$ in $\{S\cup F_{(2,2)}\}^+$ such that $F_c(\alpha_1\alpha_2\ldots\alpha_r)=\alpha'_1\alpha'_2\ldots\alpha'_r$. Now, let us consider (r+1)-cell system and any configuration $\alpha_1\alpha_2\ldots\alpha_r\alpha_{r+1}$ in $\{S\cup F_{(2,2)}\}^+$. Let $F_c(\alpha_1\alpha_2\ldots\alpha_r\alpha_{r+1})=\alpha'_1\alpha'_2\ldots\alpha'_r\alpha'_{r+1}$. If $\alpha_1\alpha_2\ldots\alpha_r\alpha_{r+1}\in S^+$, it is obvious that the p-preservability holds as $w(\alpha_1\alpha_2\ldots\alpha_r\alpha_{r+1})=w(\alpha'_1\alpha'_2\ldots\alpha'_r\alpha'_{r+1})=0$. The remaining case is that there exists at least one cell containing an element of $F_{(2,2)}$ (i.e.,x or $\bar{x}$). Let

$\alpha_i \in F_{(2,2)}$. Drop the i-th cell and let $F_c(\alpha_1\alpha_2\ldots\alpha_{i-2}\alpha_{i-1}\alpha_{i+1}\alpha_{i+2}\ldots\alpha_{r+1})=\alpha_1'\,\alpha_2'\ldots\alpha_{i-2}'\,\alpha_{i-1}''\,\alpha_{i+1}''\,\alpha_{i+2}'\ldots\alpha'_{r+1}$. Then $w(\alpha_1\alpha_2\ldots\alpha_{i-2}\alpha_{i-1}\alpha_{i+1}\alpha_{i+2}\ldots\alpha_{r+1})=w(\alpha_1'\,\alpha_2'\ldots\alpha_{i-2}'\,\alpha_{i-1}''\,\alpha_{i+1}''\,\alpha_{r+2}'\ldots\alpha'_{r+1})$ by the induction hypothesis. Observe the following two reductions.

$$\begin{aligned}
w(\alpha_1\alpha_2\ldots\alpha_r\alpha_{r+1}) &= w(\alpha_1\alpha_2\ldots\alpha_{i-1}\alpha_{i+1}\ldots\alpha_{r+1})+w(\alpha_i)\\
&= w(\alpha_1'\,\alpha_2'\ldots\alpha_{i-2}'\,\alpha_{i-1}''\,\alpha_{i+1}''\,\alpha_{i+2}'\ldots\,\alpha_{r+1}')+w(\alpha_i)\\
&= w(\alpha_1'\,\alpha_2'\ldots\alpha_{i-2}'\,\alpha_{i+2}'\ldots\alpha'_{r+1})\\
&\quad +w(\alpha_{i-1}''\,\alpha_{i+1}'')+w(\alpha_i).
\end{aligned}$$

$$\begin{aligned}
w(\alpha_1'\,\alpha_2'\ldots\alpha_r'\,\alpha'_{r+1}) &= w(\alpha_1'\,\alpha_2'\ldots\alpha_{i-2}'\,\alpha'_{i+2}\ldots\alpha_{r+1}')\\
&\quad +w(\alpha_{i-1}'\,\alpha_i'\,\alpha_{i+1}').
\end{aligned}$$

As $w(\alpha_i)=1$, it is sufficient to show $w(\alpha_{i-1}''\,\alpha_{i+1}'')+1=w(\alpha_{i-1}'\,\alpha_i'\,\alpha_{i+1}')$ if we are to see $w(\alpha_1\alpha_2\ldots\alpha_r\alpha_{r+1})=w(\alpha_1'\,\alpha_2'\ldots\alpha_r'\,\alpha_{r+1}')$. Recall the above mentioned formula for $w(\beta')$, and we are able to calculate $w(\alpha_{i-1}''\,\alpha_{i+1}'')+1$ and $w(\alpha_{i-1}'\,\alpha_i'\,\alpha_{i+1}')$ as follows;

$$\begin{aligned}
w(\alpha_{i-1}'') &= w(\alpha_{i-2})+w(\alpha_{i-1})+w(\alpha_{i+1})+a(\alpha_{i-1})(a(\alpha_{i-2})+a(\alpha_{i+1}))\\
&\quad +b(\alpha_{i-1})(b(\alpha_{i-2})+b(\alpha_{i+1}))\\
w(\alpha''_{i+1}) &= w(\alpha_{i-1})+w(\alpha_{i+1})+w(\alpha_{i+2})+a(\alpha_{i+1})(a(\alpha_{i-1})+a(\alpha_{i+2}))+b(\alpha_{i+1})\\
&\quad \times(b(\alpha_{i-1})+b(\alpha_{i+2}))\\
+)\quad 1 &= 1\\
\hline
w(\alpha_{i-1}''\,\alpha_{i+1}'')+1 &= 1+w(\alpha_{i-2})+w(\alpha_{i+2})+a(\alpha_{i-1})a(\alpha_{i-2})+a(\alpha_{i+1})a(\alpha_{i+2})\\
&\quad +b(\alpha_{i-1})b(\alpha_{i-2})+b(\alpha_{i+1})b(\alpha_{i+2}).
\end{aligned}$$

$$w(\alpha'_{i-1})=w(\alpha_{i-2})+w(\alpha_{i-1})+1+a(\alpha_{i-1})(a(\alpha_{i-2})+a(\alpha_i))+b(\alpha_{i-1})(b(\alpha_{i-2})+b(\alpha_i))$$

$$w(\alpha'_i)=w(\alpha_{i-1})+1+w(\alpha_{i+1})+a(\alpha_i)(a(\alpha_{i-1})+a(\alpha_{i+1}))+b(\alpha_i)(b(\alpha_{i-1})+b(\alpha_{i+1}))$$

$$w(\alpha'_{i+1})=1+w(\alpha_{i+1})+w(\alpha_{i+2})+a(\alpha_{i+1})(a(\alpha_i)+a(\alpha_{i+2}))+b(\alpha_{i+1})(b(\alpha_i)+b(\alpha_{i+2}))$$

$$+)\overline{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}$$

$$w(\alpha'_{i-1}\ \alpha'_i\ \alpha'_{i+1})=1+w(\alpha_{i-2})+w(\alpha_{i+2})+a(\alpha_{i-1})a(\alpha_{i-2})+a(\alpha_{i+1})a(\alpha_{i+2})$$
$$+b(\alpha_{i-1})b(\alpha_{i-2})+b(\alpha_{i+1})b(\alpha_{i+2}).$$

As $w(\alpha''_{i-1}\ \alpha''_{i+1})+1=w(\alpha'_{i-1}\ \alpha'_i\ \alpha'_{i+1})$, the p-preservability is established for (r+1)-cell system. The other seven cases (T_2—T_8) are shown to hold similarly.

6.5. (2,2)-Preservability ($|S|=2$)

In this section, we are going to show that if $A_c(S,f)$ is (2,2)-preserving, then conditions C.1, C.2, C.3, and C.4 hold. To establish such a theorem, we collect several lemmas which comprise the contrapositive proposition to the theorem as follows.

Lemma 6.17.

If C.1 does not hold, then $A_c(S,f)$ is not (2,2)-preserving.

Proof

By the assumption $w[{a \atop e}]\neq w[{d \atop h}]$, we have two cases to be considered, i.e., 1) a=e and d≠h, and 2) a≠e and d=h. As $F_c(x)=[{a \atop h}]$, $A_c(S,f)$ is not (2,2)-preserving when a=h. So let us assume a≠h. In case 1), observe $F_c(x0)=[{a \atop e}][{a \atop d}]$, and note that $a=\overline{h}=d$. In case 2), observe $F_c(x1)=[{d \atop h}][{e \atop h}]$, and note that $e=\overline{a}=h$. Thus, in both cases $A_c(S,f)$ is not (2,2)-preserving.

Lemma 6.18.

If C.2 does not hold, then $A_c(S,f)$ is not (2,2)-preserving.

Proof

Let $w_{11}=w_{22}=1$, and assume $\genfrac{[}{]}{0pt}{}{a}{e}\neq\genfrac{[}{]}{0pt}{}{d}{h}$. Then $a\neq e$, $d\neq h$, and $a\neq d$ mean $a=h$. If $w_{11}=w_{22}=0$ and $\genfrac{[}{]}{0pt}{}{a}{e}=\genfrac{[}{]}{0pt}{}{d}{h}$, then $a=e$, $d=h$, and $a=d$ also imply $a=h$. The relation $a=h$ in turn means that $A_c(S,f)$ is not (2,2)-preserving as cited in the previous proof.

Lemma 6.19.

If C.1, C.2, and C.4 hold, and C.3 does not, then $A_c(S,f)$ is not (2,2)-preserving.

Proof

If $w_{11}=w_{22}=1$, then $\genfrac{[}{]}{0pt}{}{c}{g}=\genfrac{[}{]}{0pt}{}{b}{f}$ by C.4, and à fortiori $w_{12}=w_{21}$. As $w_{12}\neq w_{21}$, w_{11} and w_{22} equal 0. Thus, $\genfrac{[}{]}{0pt}{}{a}{e}\neq\genfrac{[}{]}{0pt}{}{d}{h}$ and $\genfrac{[}{]}{0pt}{}{c}{g}\neq\genfrac{[}{]}{0pt}{}{b}{f}$ by C.2 and C.4, respectively. In view of the isomorphism and symmetry mentioned earlier, we have to consider the following two cases, (1) $\{\overline{aecgf}, \overline{bdh}\}$ and (2) $\{\overline{aecgb}, \overline{fdh}\}$, where the elements are in the same block of a partition if and only if they have the same value. Observe $F_c(0 \times 1)=\genfrac{[}{]}{0pt}{}{b}{d}\ \genfrac{[}{]}{0pt}{}{c}{g}\ \genfrac{[}{]}{0pt}{}{e}{f}$ for case (1), and $F_c(0 \times 0)=\genfrac{[}{]}{0pt}{}{a}{c}\ \genfrac{[}{]}{0pt}{}{a}{e}\ \genfrac{[}{]}{0pt}{}{a}{b}$ for case (2), which assure that $A_c(S,f)$ is not (2,2)-preserving.

Lemma 6.20.

If C.1 and C.2 hold, and C.3 and C.4 do not, then $A_c(S,f)$ is not (2,2)-preserving.

Proof

In case $w_{11}=w_{22}=0$, C.4 for this part should hold because if not, $\genfrac{[}{]}{0pt}{}{c}{g}=\genfrac{[}{]}{0pt}{}{b}{f}$ which contradicts the assumption $w_{12}\neq w_{21}$. So this case is included in Lemma 6.19, and we have to consider only the case $w_{11}=w_{22}$ $=1$ and $\genfrac{[}{]}{0pt}{}{c}{g}\neq\genfrac{[}{]}{0pt}{}{b}{f}$. As in the proof of Lemma 6.19, there are two typical cases to be treated; (1) $\{\overline{abcdg}, \overline{efh}\}$ and (2) $\{\overline{acdgf}, \overline{ebh}\}$. As for case (1), see that $F_c(0\ x\ 1)=[{b \atop d}]\ [{c \atop g}]\ [{e \atop f}]$. The case (2) is further divided into two cases according to the assignment of the values 0 and 1 to each block. If a=0, note that $F_c(1\ x\ 0)=\bar{x}\ \bar{x}\ 0$, and $F_c(\bar{x}\ \bar{x}\ 0)=0\ 0\ 0$. While a=1, note that $F_c(0\ 1\ x\ 0)=1\ x\ x\ \bar{x}$, and $F_c(1\ x\ x\ \bar{x})=1\ 0\ 1\ 1$.

Lemma 6.21.

If C.1, C.2, and C.3 hold and C.4 does not, then $A_c(S,f)$ is not (2,2)-preserving.

Proof

Let $w_{11}=w_{22}=0$ and $[{c \atop g}]\ [{b \atop f}]$. As $[{a \atop e}]\neq[{d \atop h}]$ holds in this situation by C.2, there are three typical cases to be considered; (1) $\{\overline{aebcgf}, \overline{dh}\}$ (2) $\{\overline{aebc}, \overline{gfdh}\}$ and (3)$\{\overline{aegf}, \overline{bcdh}\}$. In cases (1) and (2), note that $F_c(0\ x\ 0)=[{a \atop c}]\ [{a \atop e}]\ [{a \atop b}]$. As for case (3), $F_c(1\ \ 0\ x\ 0)=e[{b \atop d}]\ [{a \atop e}]\ [{c \atop d}]$ assures that $A_c(S,f)$ is not (2,2)-preserving. Let $w_{11}=w_{22}=1$ and $[{c \atop g}]\neq[{b \atop f}]$. As $[{a \atop e}]=[{d \atop h}]$ holds by C.2 and $w_{12}=w_{21}$ by C.3, there are two typical cases to be considered; (1) $\{\overline{adcg}, \overline{bfeh}\}$ (2) $\{\overline{adcf}, \overline{ehgb}\}$. For case (1), note that $F_c(0\ 0\ x\ 1)=b[{a \atop c}]\ [{c \atop g}]\ [{e \atop f}]$. As for case (2), if a=0, observe that $F_c(1\ \ 1\ x\ 0)=1\ x\ \bar{x}\ 0$, $F_c(1\ x\ \bar{x}\ 0)=1\ 0\ 1\ 0$, and if a=1, $F_c(1\ 0\ x\ 1)=1\ x\ \bar{x}\ 0$, $F_c(1\ x\ \bar{x}\ 0)=0\ 1\ 0\ 1$, which show that $A_c(S,f)$ is not (2,2)-preserving.

From Lemmas 6.17-6.21, we have the following

Theorem 6.5.

If $S_c(S,f)$ is (2,2)-preserving, then C.1, C.2, C.3, and C.4 hold.

As it can be easily recognized that if $A_c(S,f)$ is p-preserving, then it must be (2,2)-preserving, we have as a conclusion of this chapter,

Theorem 6.6.

For a given (S,f) the following three propositions are equivalent:

(1) $A_c(S,f)$ is (2,2)-preserving.

(2) $A_c(S,f)$ is p-preserving.

(3) (S,f) satisfies C.1, C.2, C.3, and C.4.

Proof　　See Fig. 6.3.

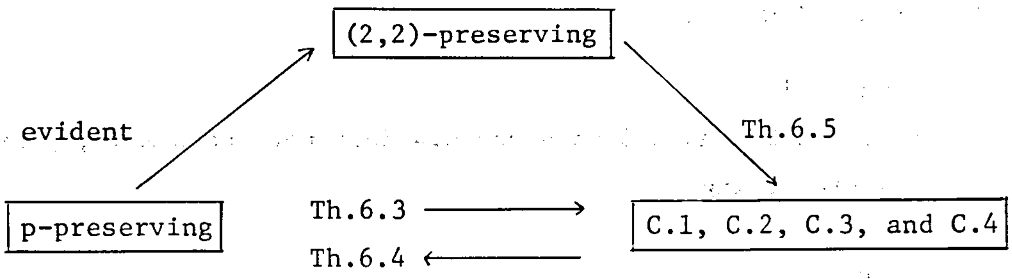


Fig.6.3. Relations between three propositions of Theorem.6.6

6.6. Concluding Remarks

We proposed a method to represent noises or information appeared in a cellular space and discussed several properties concerning the conservation of the effects in the system. Most of our effort is paid to the systems of one-dimensional two-state iterative automata with cyclic boundary condition.

In particular, we showed that the parity-preservability is a necessary and sufficient condition for such a system to be information preserving. In order to show the equivalence, we utilized some conditions for the extended transition table.

As for p-preservability, it is not so clear what is meant by such property, though it is a handier tool to treat our problem. As is felt by the process of the proofs in this chapter, it seems essential for the validity of our method that systems under consideration are with cyclic boundary condition.

Generalizations of our method and propositions to the systems of a higher dimensional cellular space, and of more than two-state-cells are left untouched.

CHAPTER 7

SOME ALGEBRAIC TREATMENTS OF THE SYSTEMS

As is mentioned earlier, it seems to be hard to establish some general method for the analysis and synthesis of iterative systems. In this chapter, we are going to present some algebraic treatments of the systems, which may be a step to a unified treatment for a restricted class of the objects or for some specified characteristics.

First, an algebraic method of analyzing an iterative automata composed of cells whose transition function is defined via an abelian group is proposed. The approach resembles that for linear systems, and in fact, makes itself a generalization of the latter.

In the latter half of this chapter, the pair algebra of Hartmanis and Stearns [15] is utilized for analysis and synthesis of the systems.

7.1. Analysis of Systems Defined via Abelian Group [23]

Let $S = \{0,1,2,\ldots,m-1\}$ be a finite state set of a cell automaton. The set S with an additive operation (denoted by +) is assumed to form an abelian group in which 0 is the neutral element. A cell automaton (S,f) is said to be *defined via abelian group* if S is an abelian group and f is defined by the operation of the group as $f(\alpha,\beta,\gamma)=\alpha+\beta+\gamma$, $f(\alpha,\beta,\gamma)=2\alpha+\beta+\gamma$, etc., where 2α means $\alpha+\alpha$.

As shown below, systems of iterative automata composed of cells defined via abelian group may be divided into two classes; one is the systems that can be analyzed as that of linear automata, and the other that can not.

It is well known [4] that a finite abelian group S of order m may be represented as a direct product of subgroups as follows; Let $m = p_1^{r_1} p_2^{r_2} \ldots p_t^{r_t}$ where p_1 , p_2 ,....p_t are the distinct primes and r_i's are positive integers. Then S has a subgroup P_i of order $p_i^{r_i}$ for each i (i=1,2,...,t) and is represented as a direct product $P_1 \times P_2 \times \ldots \times P_t$.

Further, let the invariant of an abelian group of order p^r be $(p^{q_1}, p^{q_2}, \ldots, p^{q_k})$ where p is a prime number, and r, $q_1, \ldots, q_k$ are positive integers such that $r = q_1 + q_2 + \ldots + q_k$. If $q_1 = q_2 = \ldots = q_k = 1$, then an iterative automaton defined via the group P can be analyzed as a usual linear automaton regarding P as GF(p^r). If otherwise, there remains cyclic groups to be analyzed whose orders are of the form p^q. An iterative automaton defined via cyclic group of order p^q may be analyzed much the same as a linear automaton as shown below.

7.1.1. In Case S is a Cyclic Group of Order p^q

Let S be a cyclic module of order p^q where p is prime and q is a positive integer greater than one. By definition, a transition function of a cell automaton may be written in its most general form as $f(\alpha, \beta, \gamma) = n_\alpha \alpha + n_\beta \beta + n_\gamma \gamma + n_c$, where n_α, n_β, and n_γ are integers, and $n_c \in S$.

An essential point of our method is that S should be regarded as a factor ring $\mathbf{Z}/(p^q)$ of the integral ring $\mathbf{Z}$.

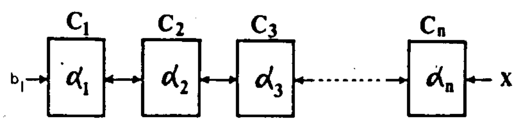


Fig. 7.1. Iterative automaton of length n with a mixed boundary condition

As shown in Fig. 7.1, iterative automata with a mixed boundary condition (i.e., b_l being fixed, and b_r variable) are treated in this section. Let $F_{b_l}(\alpha_1\alpha_2\ldots.\alpha_n;\ x) = \alpha_1'\ \alpha_2'\ldots\alpha_n'$ be the global transition function for the system composed of the above mentioned cell (S,f). Then, the mapping of $\alpha_1\alpha_2\ldots.\alpha_n$ to $\alpha_1'\ \alpha_2'\ldots.\alpha_n'$ can be written in the form

$$\begin{pmatrix} \alpha_1' \\ \alpha_2' \\ \vdots \\ \\ \alpha_n' \end{pmatrix} = \begin{pmatrix} n_\beta & n_\gamma & 0 & \cdots & & 0 \\ n_\alpha & n_\beta & n_\gamma & 0 & \cdots & 0 \\ & & \vdots & & & \\ 0 & 0 & \cdots & n_\alpha & n_\beta & n_\gamma \\ 0 & 0 & \cdots & n_\alpha & n_\beta & n_\gamma \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \\ \alpha_n \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{pmatrix} n_c + \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} b_l + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} x \qquad (7.1)$$

where additions and multiplications are thought to be those of $\mathbf{Z}/(p^q)$.

Let $\alpha_n(t)$ denote a state of an iterative automaton of length n at time t, and e_n^i the unit column vector all of its n elements equal 0 except the i-th element. If an n×n matrix of the above equation is written as T_n, Eq. 7.1 may be rewritten as

$$\alpha_n(t+1)=T_n\alpha_n(t)+1n_c+e_n^1 b_l+e_n^n x(t) \qquad (7.2)$$

Solving Eq. 7.2, we get

$$\alpha_n(t)=T_n^t\alpha(0)+\left(\sum_{k=0}^{t-1} T_n^k\right)(1n_c+e_n^1 b_l)+\sum_{k=0}^{t-1} T_n^{t-k-1}e_n^n x(k), \qquad (7.3)$$

where $T_n^k = T_n \cdot T_n^{k-1}$ and T_n^0 is the n×n unit matrix.

The three terms of Eq. 7.3 from left to right represent the effects of an initial state configuration, the constant inputs n_c and b_l, and an input sequence to the rightmost cell, respectively. And an autonomous behavior is determined by the matrix T_n.

Although analyses of the system parallel those for linear systems, it should be borne in mind that a matrix T_n is defined on a ring.

For example, the non-singularity of a T_n is determined depending on whether the determinant of T_n (denoted as $|T_n|$) is a unit element or not.

Let us examine, as an illustration, a case in which $n_\alpha = n_\beta = n_\gamma = 1$. Then we have the following recursive relation and the initial conditions;

$$|T_n| = |T_{n-1}| - |T_{n-2}| \qquad (n \geqq 3),$$
$$|T_1| = 1, \text{ and } |T_2| = 0.$$

Solving the above equation, we get

$$|T_{3k-1}| = 0 \qquad (k=1,2,3,\ldots\ldots\ldots),$$
$$|T_{3k}| = |T_{3k+1}| \qquad = \begin{cases} 1 & (k=2,4,6,\ldots\ldots) \\ p^q-1 & (k=1,3,5,\ldots\ldots). \end{cases}$$

As p^q-1 and 1 are the unit elements of $Z/(p^q)$, the T_n under consideration is non-singular for all $n \geqq 1$, except the case $n = 3k-1$ $(k=1,2,\ldots\ldots)$. Accordingly, the autonomous transition diagrams for the system with $n_c = 0$, $b_l = 0$, and $x(t) = 0$ $(t=0,1,2,\ldots\ldots)$ are composed only of the cyclic parts if the length of the system is not equal to $3k-1$ $(k=1,2,\ldots\ldots)$.

7.1.2. In Case S is a Module of Type $(p,p,\ldots\ldots,p)$

When S is a module of order p^q whose invariant is $(p,p,\ldots,p)$, then it may be treated as $GF(p^q)$. Equation 7.1 is also valid in this case with T_n being regarded as defined on $GF(p)$. This is a usual linear automaton.

Some different aspects between the above mentioned two cases are illustrated by the following examples.

Let $S = \{0,1,2,3\}$, and consider $Z/(2^2)$ and $GF(2^2)$, whose additive operations are shown below.

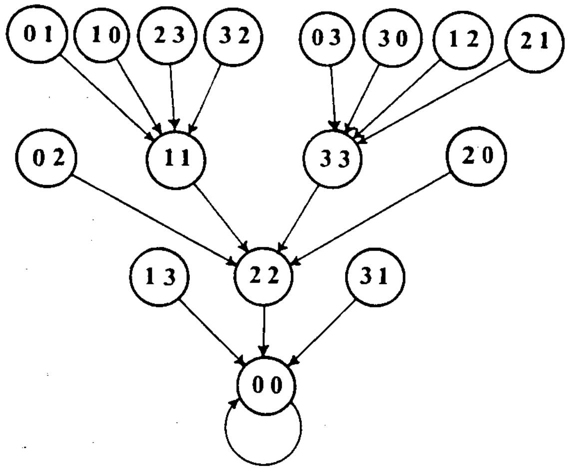
+	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

$Z/(2^2)$

+	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

$GF(2^2)$

For each above mentioned abelian group, assume $f(\alpha,\beta,\gamma)=\alpha+\beta+\gamma$ and $b_l=0$. The autonomous behaviors (i.e., in case $x(t)=0$ for $t=0, 1,2,\ldots..$) of the systems of length two and three with corresponding T_n^k's are shown by Figs. 7.2 (1)-(4).

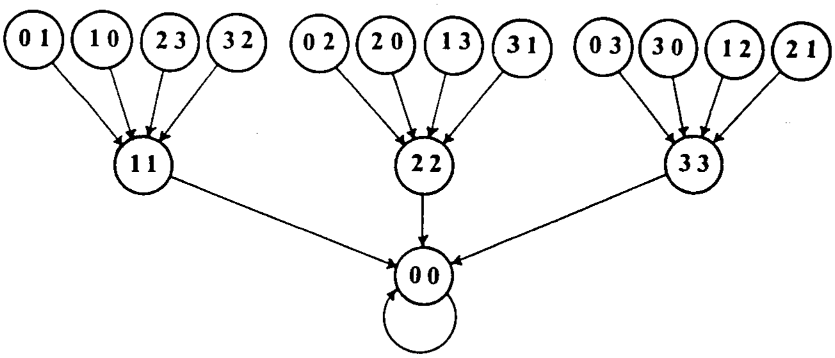


$T=\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix}$

$T^2=\begin{bmatrix}2 & 2\\ 2 & 2\end{bmatrix}$

$T^3=\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}$

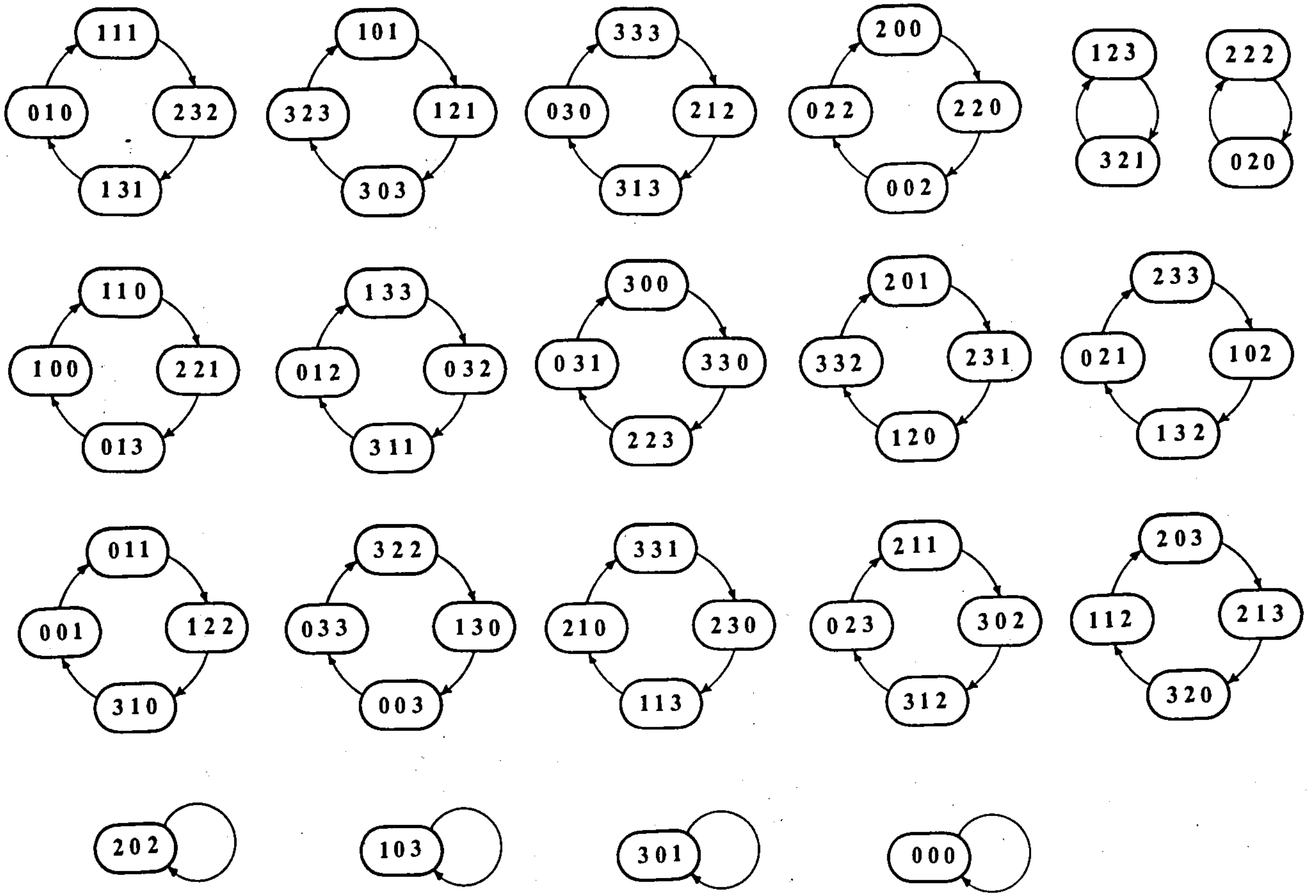
Fig. 7.2(1) $Z/(2^2)$ n = 2



$T=\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix}$

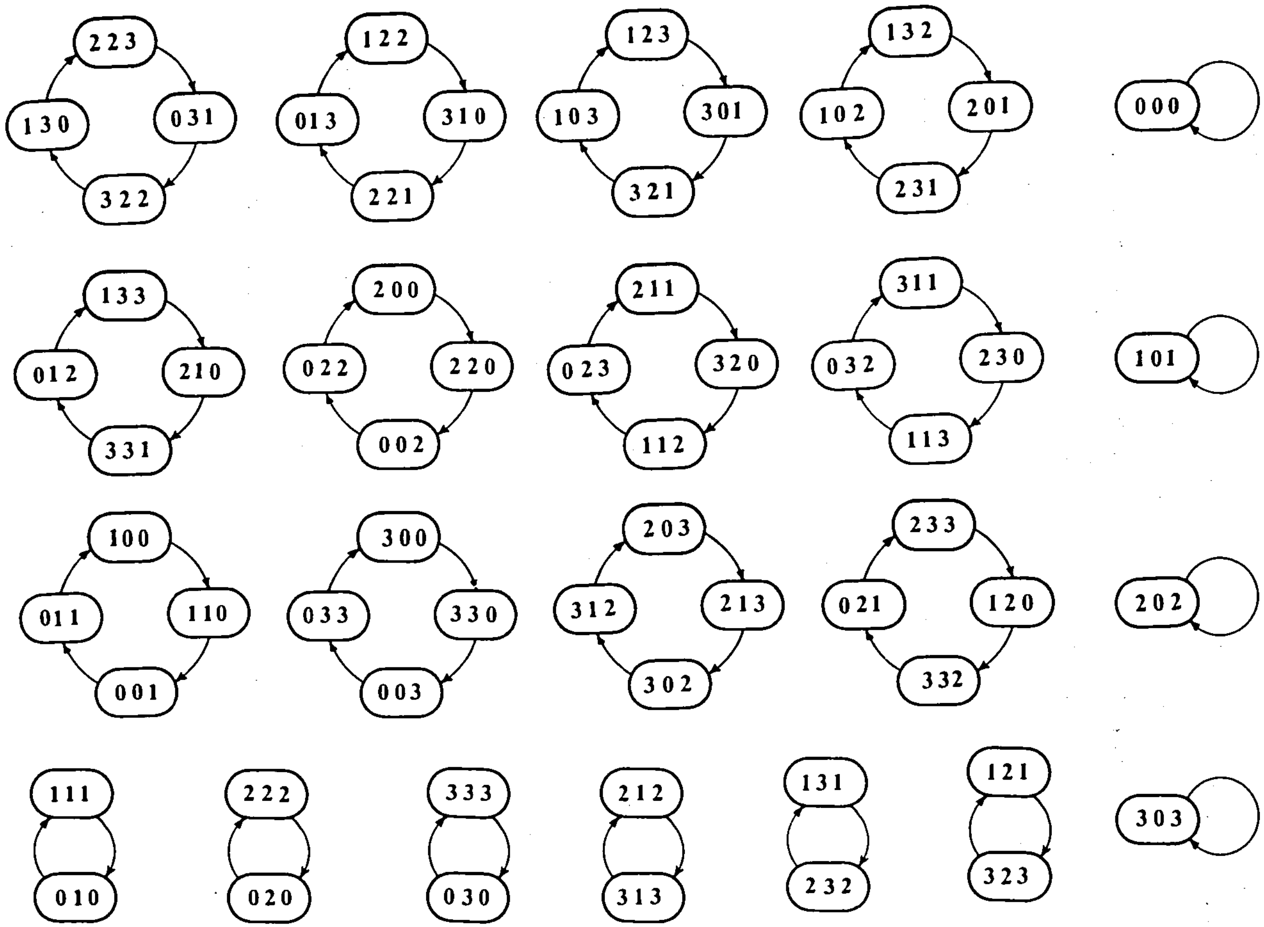
$T^2=\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}$

Fig. 7.2(2) $GF(2^2)$ n = 2



$$T=\begin{bmatrix}1&1&0\\1&1&1\\0&1&1\end{bmatrix}\quad T^2=\begin{bmatrix}2&2&1\\2&3&2\\1&2&2\end{bmatrix}\quad T^3=\begin{bmatrix}0&1&3\\1&3&1\\3&1&0\end{bmatrix}\quad T^4=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}$$

Fig.7.2(3) $Z/(2^2)$ n=3 {3[1] 2[2] 14[4]}



$$T=\begin{bmatrix}1&1&0\\1&1&1\\0&1&1\end{bmatrix}\quad T^2=\begin{bmatrix}0&0&1\\0&1&0\\1&0&0\end{bmatrix}\quad T^3=\begin{bmatrix}0&1&1\\1&1&1\\1&1&0\end{bmatrix}\quad T^4=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}$$

Fig.7.2(4) GF(2^2) n=3 {3[1] 6[2] 12[4]}

7.2. Analysis of Error in Terms of Pair Algebra [22]

There have been several approaches appeared in the literatures [1, 34, 43] for treating the problems of errors or mal-functions of an automaton. One of them is to seek effectively some error detecting input sequences. Another approach concerns itself with constructing easily diagnosable circuits. Still another is related with probabilistic treatments in such a way as to design a highly reliable circuit from less reliable components.

An analysis method of a system of iterative automata composed of cells with redundancies is presented in this section. First, a tolerable error partition on a state set S is assumed to be given. That is, we are assumed to know in advance the types of errors tolerated at the output cell. Then the problem treated here is to limit the types of errors that may occur at the other cells and that are tolarated at the output cell.

In what follows, a system of iterative automata composed of RU type cells is treated first ($f(\cdot, \alpha, \beta)$ will be denoted as $g(\alpha,\beta)$.), and then extended to the cases of ordinary three neighbor cells. The leftmost cell (C_1) is assumed to be an output cell as usual.

Several definitions are now in order.

Definition 7.1. (*tolerance partition* ; π_t)

A tolerance partition π_t is a partition on S with the following interpretation : only an error that is within a block of π_t is assumed to be tolerated. That is, we are able to observe only $\lambda(s)$'s, $s \in S$, where λ is a mapping from S onto $\pi_t(S)$ (Moore type output function), and an operation of λ is assumed to be error free.

Definition 7.2. (*error partition* ; μ)

An error partition μ is also a partition on S which specifies a pattern of errors supposed to happen in a cell. That is, errors are assumed to be reflexive and transitive, and an occurence of an error means that a state α of a cell at a time is not equal to what it ought to be (let the state be β). And states α and β are in the same block of an error partition μ . An error partition of cell C_k is denoted by $\mu(k)$.

Definition 7.3. (*similarity on* μ ; $\tilde{\mu}$)

Let $A = (S, g_A)$ and $B = (S, g_B)$ be two cell automata defined on the same state set S. Then A is similar to B on an error partition μ (written $A \underset{\mu}{\sim} B$) iff $g_A(\alpha,\beta) = \alpha'$ implies $\alpha' \equiv g_B(\alpha,\beta)(\mu)$ for any α and β in S. The relation $\underset{\mu}{\sim}$ is an equivalence relation and defines classes of cell automata induced by μ. Thus an error partition specifies structural as well as behavioral error types.

Definition 7.4. ($\mu(k)$-*tolerable*)

Given (S, g), π_t, and $\mu(k)$ a system of iterative automata composed of the cells (S, g) is said to be $\mu(k)$-tolerable if any error in $\mu(k)$ is tolerated by π_t at the output cell.

7.2.1. Applications of Pair Algebra

Hartmanis and Stearns utilize a pair algebra of the partitions on the state set and/or the input output symbol set of a sequential machine in order to establish a structural decomposition theory of machines [15]. The pair algebra used by them essentially describes various information flows in a sequential machine, and some applications to an error analysis are also mentioned in the cited reference.

Here we define a pair algebra and give in terms of it a necessary and sufficient condition for given (S, g) and π_t to be $\mu_i(i)$-tolerable $(i=1,2,\ldots.n)$.

First, let us define an S-S pair and an I-S pair following Hartmanis and Stearns [15]. As for notations and further details, see the reference.

Definition 7.5. (*S-S pair, I-S pair*)

Given a cell (S, g). Let π_A and π_B be partitions on S.

S-S pair

An ordered pair π_A and π_B is an S-S pair (written $(\pi_A\ \pi_B)_{S-S}$) iff $s \equiv t\ (\pi_A)$ implies $g(s,\alpha) \equiv g(t,\alpha)\ (\pi_B)$ for $\forall \alpha \in S$ and s, $t \in S$.

I-S pair

An ordered pair π_A and π_B is an I-S pair (written $(\pi_A\ \pi_B)_{I-S}$) iff $u \equiv v\ (\pi_A)$ implies $g(\alpha, u) \equiv g(\alpha, v)(\pi_B)$ for $\forall \alpha \in S$ and u, v $\in$ S.

Now we introduce another partition pair called an array pair as follows.

Definition 7.6. (*array pair* ; $[\pi_A\ \pi_B]$)

Let (S, g), π_A, and π_B be as before. An ordered pair π_A and π_B is an array pair (written $[\pi_A\ \pi_B]$) iff $u \equiv v\ (\pi_A)$ and $s \equiv t\ (\pi_B)$ imply $g(s,u) \equiv g(s,v) \equiv g(t,u) \equiv g(t,v)\ (\pi_B)$ where u, v, s, and t are in S.

We can easily verify the following

Lemma 7.1.

Two partitions π_A and π_B constitute $[\pi_A\ \pi_B]$ iff $(\pi_B\ \pi_B)_{S-S}$ and $(\pi_A\ \pi_B)_{I-S}$.

Proof

The 'only if' part of the lemma is obvious. The remainder is shown as follows. Let $u \equiv v\ (\pi_A)$ and $s \equiv t\ (\pi_B)$. Then $g(s,u) \equiv g(t,u)\ (\pi_B)$ and $g(s,v) \equiv g(t,v)\ (\pi_B)$ hold from $(\pi_B\ \pi_B)_{S-S}$. By $(\pi_A\ \pi_B)_{I-S}$, $g(s,u) \equiv g(s,v)\ (\pi_B)$ holds to establish $[\pi_A\ \pi_B]$.

Let L denote the set of all partitions on S, and L' the set of all partitions π such that π and π is an S-S pair for given (S,g). That is, $L' = \{\pi \mid (\pi\ \pi)_{S-S}\}$.

Given (S,g) and let Δ be the set of all array pairs for (S,g). Then we have

Lemma 7.2.

$\Delta = \{ [\pi_A \ \pi_B] \}$ is a pair algebra on $L \times L'$.

Proof

First, note that Ω_{S-S} (the set of all S-S pairs) and Ω_{I-S} (the set of all I-S pairs) are both pair algebras.

Let $[\pi_A \ \pi_B] \in \Delta$ and $[\pi'_A \ \pi'_B] \in \Delta$. By Lemma 7.1, we have $(\pi_B \ \pi_B)_{S-S}$, $(\pi_A \ \pi_B)_{I-S}$, $(\pi'_B \ \pi'_B)_{S-S}$, and $(\pi'_A \ \pi'_B)_{I-S}$. Noting the structure of Ω_{S-S} and Ω_{I-S}, we get $(\pi_B\pi'_B \ \ \pi_B\pi'_B)_{S-S}$ and $(\pi_A\pi'_A \ \pi_B\pi'_B)_{I-S}$, which is equivalent to $[\pi_A\pi'_A \ \pi_B\pi'_B] \in \Delta$. Similarly, $[\pi_A+\pi'_A \ \pi_B+\pi'_B] \in \Delta$ is established from $(\pi_B+\pi'_B \ \ \pi_B+\pi'_B)_{S-S}$ and $(\pi_A+\pi'_A \ \pi_B+\pi'_B)_{I-S}$. The next step is to see that $[\pi \ \ I] \in \Delta$ and $[0 \ \pi'] \in \Delta$ for $\forall \pi \in L$ and $\forall \pi' \in L'$ where I and 0 are the maximum and the minimum elements of L' and L respectively. By definition, $(I \ \ I)_{S-S}$ and $(\pi \ \ I)_{I-S}$ for $\forall \pi \in L$, which means that $[\pi \ \ I] \in \Delta$. The relation $[0 \ \pi'] \in \Delta$ holds as $(\pi' \ \pi')_{S-S}$ and $(0 \ \pi')_{I-S}$ for $\forall \pi' \in L'$. (Q.E.D.)

Now we are in a position to state the following

Theorem 7.1.

An iterative automaton of length n, A(S, g, n) is $\mu_1(1)$-tolerable, $\mu_2(2)$-tolerable, ..., and $\mu_n(n)$-tolerable with respect to given π_t iff $\mu_1 \leqq \pi_t$, $[\mu_2 \ \mu_1]$, $[\mu_3 \ \mu_2]$,...., $[\mu_n \ \mu_{n-1}]$, and $(\mu_n \ \mu_n)_{S-S}$.

Proof

The "if" part of the theorem may be obvious. The "only if" part can be established step by step as follows. As A(S,g,n) is $\mu_1(1)$-tolerable, it is necessary that $\mu_1 \leqq \pi_t$ for the cell C_1 is assumed to be an output cell. At the same time, it is $\mu_2(2)$-tolerable, which necessitates $[\mu_2 \ \mu_1]$. Similarly, the $\mu_i(i)$ and $\mu_{i+1}(i+1)$ tolerability demands that $[\mu_{i+1}\ \mu_i]$ where i=2,3,.....,n-1. As the rightmost cell (C_n) receives inputs from the outside world and no errors are assumed in this step, $(\mu_n\ \mu_n)_{S-S}$ is the only requisite for C_n to be $\mu_n(n)$-tolerable. (Q.E.D.)

(S,g) is said to have h-property with respect to π, a partition on S, iff there exist $h: S \to \pi(S)$ and $g_h : \pi(S)\times\pi(S) \to \pi(S)$ such that

$$g_h(\pi(\alpha),\ \pi(\beta)) = h(g(\alpha,\beta)) \text{ for } {}^\forall\alpha,\beta \in S.$$

As to the h-property, we have the following

Lemma 7.3.

(S,g) has h-property with respect to π iff $[\pi\ \pi] \in \Delta$.

Proof

Let (S,g) has h-property with respect to π. Let $u \equiv v(\pi)$ and $s \equiv t\ (\pi)$, i.e., $\pi(u) = \pi(v)$ and $\pi(s) = \pi(t)$ respectively. By definition of h-property, we have

$$\pi(g(s,u)) = h(g(s,u)) = g_h(\pi(s),\ \pi(u)),$$

$$\pi(g(s,v)) = h(g(s,v)) = g_h(\pi(s),\ \pi(v)),$$

$$\pi(g(t,u)) = h(g(t,u)) = g_h(\pi(t),\ \pi(u)), \text{ and}$$

$$\pi(g(t,v)) = h(g(t,v)) = g_h(\pi(t),\ \pi(v)).$$

As the rightmost terms in the above equations are the same, we obtain $g(s,u) \equiv g(s,v) \equiv g(t,u) \equiv g(t,v)\ (\pi)$, which means that $[\pi\ \pi] \in \Delta$.

Conversely, let $[\pi\ \pi] \in \Delta$. Define h and g_h as follows;

$$h(\alpha) = \pi(\alpha) \qquad \alpha \in S,$$

$$g_h(\pi(\alpha),\ \pi(\beta)) = \pi(g(\alpha,\beta)) \quad \alpha,\beta \in S.$$

The remaining thing to do is to see that g_h is well defined. Consider any α' and β' in S such that $\pi(\alpha) = \pi(\alpha')$ and $\pi(\beta) = \pi(\beta')$.

By the fact that $[\pi\ \pi]\in\Delta$, $g(\alpha,\beta) \equiv g(\alpha',\beta')\ (\pi)$, i.e., $\pi(g(\alpha,\beta)) = \pi(g(\alpha',\beta'))$. This completes the proof. (Q.E.D.)

Noting Lemma 7.3, the above mentioned Theorem 7.1 may be regarded as a generalization of the concept of h-property.

Now, an algorithm of deriving μ_i's satisfying Theorem 7.1 is in order.

For given (S,g) and π_t,

(1) Find the elements of L' and draw an ordering diagram of the lattice L'.

(2) Add the diagram an arrow from π_B to π_A iff $[\pi_A\ \pi_B]\in\Delta$ where π_A and π_B are in L'.

(3) Let μ_1 be a maximal point in L' such that $\mu_1 \leqq \pi_t$.

(4) As μ_1 being a starting point, go upward along an arrow (i.e., to larger partitions), if possible, in the diagram of L'. The sequence of points in L' traversed constitutes μ_i's.

An illustrative example of the above process is given as follows;

Let S = {0,1,2,3} and let g be

α \ β	0	1	2	3
0	1	1	3	1
1	0	1	3	0
2	1	0	3	2
3	3	3	1	3

$g(\alpha,\beta)$

The elements of L' are

$0=(\overline{0}\ \overline{1}\ \overline{2}\ \overline{3})$,

$\pi_1=(\overline{01}\ \overline{2}\ \overline{3})$,

$\pi_2=(\overline{012}\ \ \overline{3})$,

$\pi_3=(\overline{013}\ \ \overline{2})$, and

$I=(\overline{0123})$.

A diagram of L' is shown in Fig. 7.3.

The set of array pairs Δ is given by

$\Delta=\{[0,x],\ x\in L',\ [\pi_1\ \pi_1],\ [\pi_1\ \pi_2],\ [\pi_1\ \pi_3],\ [\pi_2\ \pi_3],\ [\pi_3\ \pi_2]\}$,

which is also shown by arrows in Fig. 7.3. (Arrows corresponding to [0,x]'s are omitted.)

Then, for example, let $\pi_t = (\overline{012}\ \overline{3})$. The sequence μ_1, μ_2, ... satisfying Theorem 7.1 will be $\pi_2, \pi_3, \pi_2, \pi_3, \ldots$.

If $\pi_t = (\overline{01}\ \overline{2}\ \overline{3})$, the only possibility of the μ_i sequence is $\pi_1, \pi_1, \pi_1 \cdots\cdots$.

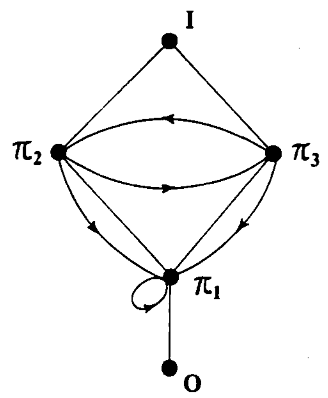


Fig. 7.3. A diagram of L'

7.2.2. Analysis for Three Neighbor Cells

Though we have hitherto treated a system of iterative automata composed of RU type cells, it is easy to extend the theory to fit a system composed of ordinary three neighbor cells. Some necessary modifications on the above mentioned definitions are explained here.

The S-S pair and I-S pair in Definition 7.5 are modified as follows:

Definition 7.6. (*S-S pair, I_r-S pair, I_l-S pair*)
Given a cell (S,f). Let π_A and π_B be partitions on S.

S-S pair

An ordered pair π_A and π_B is an S-S pair (written $(\pi_A\ \pi_B)_{S-S}$) iff $s \equiv t\ (\pi_A)$ implies $f(\alpha,s,\gamma) \equiv f(\alpha,t,\gamma)(\pi_B)$ for $\forall\alpha,\gamma \in S$ and $s, t \in S$.

I_r-S pair

An ordered pair π_A and π_B is an I_r-S pair (written $(\pi_A\ \pi_B)_{I_r-S}$) iff $u \equiv v\ (\pi_A)$ implies $f(\alpha,\beta,u) \equiv f(\alpha,\beta,v)(\pi_B)$ for $\forall\alpha,\beta \in S$ and $u, v \in S$.

I_l-S pair

An ordered pair π_A and π_B is an I_l-S pair (written $(\pi_A\ \pi_B)_{I_l\text{-}S}$) iff $p \equiv q\ (\pi_A)$ implies $f(p,\beta,\gamma) \equiv f(q,\beta,\gamma)(\pi_B)$ for $\forall\beta,\gamma \in S$ and $p, q \in S$.

As for the array pair in Definition 7.6, we have

Definition 7.7. *(array triple ; $[\pi_A\ \pi_B\ \pi_C]$)*

Consider (S,f) and let π_A, π_B, and π_C be partitions on S. An ordered triple π_A, π_B, and π_C is an array triple (written $[\pi_A\ \pi_B\ \pi_C]$) iff $p \equiv q\ (\pi_A)$, $s \equiv t\ (\pi_B)$, and $u \equiv v\ (\pi_C)$ imply $f(p,s,u) \equiv f(p,s,v) \equiv f(p,t,u) \equiv f(p,t,v) \equiv f(q,s,u) \equiv f(q,s,v) \equiv f(q,t,u) \equiv f(q,t,v)$ (π_B) where p,q,s,t,u, and v are in S.

Corresponding to Lemma 7.1, we have

Lemma 7.4.

Three partitions π_A, π_B, and π_C constitute $[\pi_A\ \pi_B\ \pi_C]$ iff $(\pi_B\ \pi_B)_{S\text{-}S}$, $(\pi_C\ \pi_B)_{I_r\text{-}S}$, and $(\pi_A\ \pi_B)_{I_l\text{-}S}$.

The proof is omitted as it could be carried out like in that of Lemma 7.1.

With these modifications in the definitions, Theorem 7.1 may be generalized to

Theorem 7.2.

An iterative automaton of length n, A(S,f,n) is $\mu_1(1)$-tolerable, $\mu_2(2)$-tolerable, ..., and $\mu_n(n)$-tolerable with respect to given π_t iff $\mu_1 \leq \pi_t$, $[\mu_1\ \mu_2\ \mu_3]$, $[\mu_2\ \mu_3\ \mu_4]$, ..., $[\mu_{n-2}\ \mu_{n-1}\ \mu_n]$, $(\mu_{n-1}\ \mu_n)_{I_l\text{-}S}$, and $(\mu_n\ \mu_n)_{S\text{-}S}$.

7.3. Concluding Remarks

The two approaches treated in this chapter are not so general in that the former is applicable only for a limited class of cells and the latter is concerned with a specified feature of the systems. Regrettably, however, there seems to be no general enough and at the same time fruitful ways of handling iterative systems as yet. Thus our approaches should be regarded as a step toward such a unified theory.

In this chapter, already existing mathematical concepts like an abelian group, a factor ring, a lattice, a pair algebra, and so forth are utilized. In order to accomplish the above mentioned tasks, however, such novel mathematical concepts and tools if any should be invented that reflect some essential structures of the systems.

REFERENCES

1 Armstrong,D.B.(1966), On finding a nearly minimal set of fault detection tests for combinational logical nets. *IEEE Trans. Electronic Computers* EC-15, 66-73.

2 Balzer,R.(1967), An 8-state minimal time solution to the firing squad synchronization problem. *Inform. Control.* 10, 22-42.

3 Banks,E.R.(1971), Information processing and transmission in cellular automata. MAC TR-81,MIT.

4 Birkhoff,G. and S.MacLane(1965) "A Survey of Modern Algebra." The Macmillan Company, Inc.

5 Booth,T.L.(1967) "Sequential Machines and Automata Theory." John Wiley & Sons, Inc.

6 Бухштаб,Ю.А.(1970), Реализуемость функций на одномерных итеративных сетях в реальное время. Проблемы кибернетики, Выпуск 22, 85-94.

7 Chan,S.K.(1970), The analysis and synthesis of two-dimensional patterns using picture-processing grammars. Second ACM Symposium on Theory of Computing, 206-216.

8 Codd,E.F.(1968) "Cellular Automata." ACM Monograph. Academic Press, Inc.

9 Cole,S.N.(1969), Real-time computation by n-dimensional iterative arrays of finite state machines. *IEEE Trans. Computers.* C-18, 349-365.

10 Davis,M.(1958) "Computability and Unsolvability." McGraw-Hill Book Company, Inc.

11 Elspas,B. (1971), The theory of multirail cascades. in "Recent Developments in Switching Theory."(ed. Mukhopadhyay,A.) 315-367. Academic Press,Inc.

12 Fischer,P.C.(1965), Generation of primes by a one-dimensional real time iterative array. J.ACM, 12, 3, 388-394.

13 Furuta,S., Y.Inagaki, and T.Fukumura(1969), Figure generating rules and two-dimensional tape automata. Technical Report A69-5, *IECEJ*. (in Japanese)

14 Gill,A.(1967) "Linear Sequential Circuits : Analysis, Synthesis and Applications." McGraw-Hill Book Company, Inc.

15 Hartmanis,J. and R.E.Stearns(1966) "Algebraic Structure Theory of Sequential Machines." Prentice-Hall, Inc.

16 Hennie,F.C.(1961) "Iterative Arrays of Logical Circuits." MIT Press.

17 Kasami,T. and M.Fujii(1968), On powers of the systems of one-dimensional iterative logical circuits. *IECEJ Trans.* 51-C, No.6, 275-282.

18 Kautz,W.H.(1967), Testing for faults in combinational cellular logic arrays. Proc. 8th Annu. Symp. on Automata and Switching Theory, 161-174.

19 Kilmer,W.L.(1964), Topics in the theory of one-dimensional iterative networks. *Inform. Control.* 7, 180-199.

20 Kobuchi,Y.(1967) "Studies on One-Dimensional Iterative Circuits." Master Thesis, Dept. of Electrical Engineering , Kyoto Univ.

21 Kobuchi,Y.(1971), A classification of systems of 1-dimensional iterative automata (2). Technical Report A71-3, *IECEJ*.(in Japanese)

22 Kobuchi,Y.(1972), Analysis of errors in the system of one-dimensional iterative automata. 1972 Convention Record of *IECEJ*. 168(in Japanese)

23 Kobuchi,Y. and H.Nishio(1970), Analysis of the system of one-dimensional iterative automata defined via abelian group. 1970

Joint Convention Record 2798 (in Japanese)

24 Kobuchi,Y. and H.Nishio(1971), Properties of certain state sets in the system of one-dimensional iterative automata. *IECEJ Trans.* 54-C, No. 5, 396-403.(in Japanese)

25 Kobuchi,Y. and H.Nishio, Some regular state sets in the system of one-dimensional iterative automata. to appear in *Information Sciences*.

26 Kobuchi,Y. and H.Nishio, A classification of systems of one-dimensional iterative automata by its information transmission capability. to appear in *IECEJ Trans.*

27 Kobuchi,Y., H.Nishio and T.Sakai(1970), A classification of systems of 1-dimensional iterative automata (1). Technical Report A69-61, *IECEJ*. (in Japanese)

28 Kobuchi,Y., H.Nishio and T.Sakai(1970), On indices concerning control of systems of 1-dimensional iterative automata. Technical Report A69-62, *IECEJ*. (in Japanese)

29 Maitra,K.K.(1962), Cascaded switching networks of two-input flexible cells. *IRE Trans. Electronic Computers* EC-11, 136-143.

30 Minnick,R.C.(1964), Cutpoint cellular logic. *IEEE Trans. Electronic Computers* EC-13, 685-698.

31 Moore,E.F. (1962), Machine models of self-reproduction. Proc.Symp. Appl. Math. 14, 17-33.

32 Moore,E.F.(1964) (ed.) "Sequential Machines : Selected Papers." Addison-Wesley, Inc.

33 Mukhopadhyay,A. and H.S.Stone(1971), Cellular logic. in "Recent Developments in Switching Theory."(ed. Mukhopadhyay,A.) 255-313. Academic Press,Inc.

34 von Neumann,J.(1956), Probabilistic logics and the synthesis of reliable organisms from unreliable components. Automata Studies

(ed. Shannon,C.E. and J.McCarthy) 43-98.

35 von Neumann,J.(1966) "Theory of Self-Reproducing Automata." (edited and completed by A.W.Burks.) Univ. of Illinois Press, Urbana.

36 Nishio,H.(1971), Uniformly structured machines : Cellular automata. 1971 Joint Convention Record 159 (in Japanese)

37 Sakaguchi,T.(1971) "Information-Maintenance Capability in One-Dimensional Cyclic Cellular Automata." Master Thesis, Dept. of Electrical Engineering, Kyoto Univ.

38 Sakaguchi,T., T.Sakai, Y.Kobuchi, and H.Nishio(1971), Information maintenance capability of 1-dimensional cyclic cellular automata. Technical Report A70-85, *IECEJ*. (in Japanese)

39 Short,R.A.(1965), Two-rail cellular cascades. AFIPS Conference Record 27, part 1, 355-369.

40 Sklansky,J.(1963), General synthesis of tributary switching networks. *IEEE Trans. Electronic Computers* EC-12, 464-469.

41 Smith III,A.R.(1969), Cellular automata theory. Technical Report No.2, Digital Systems Laboratory, Stanford Univ.

42 Takahashi,H.(1972), Decision problems about the maximum invariant set. Technical Report AL72-2, *IECEJ*.(in Japanese)

43 Takaoka,T.(1970) "Algebraic Theory of Automata and its Application to Fail-Safe Systems." Doctor Thesis, Dept. of Applied Mathematics and Physics, Kyoto Univ.

44 Thatcher,J.W.(1964), Universality in the von Neumann cellular model. I.B.M. Research Report RC-1255.

45 Unger,S.H.(1963), Pattern recognition using two-dimensional, bilateral, iterative, combinational switching circuits. Proc. Symp. on Mathematical Theory of Automata, Microwave Research Institute Symposia Series, Vol.12, Polytechnic Press

of the Polytechnic Institute of Brooklyn, N.Y. 577-591.

46 Varshavsky,V.I.(1969), The organization of interaction in collection of automata. in "Machine Intelligence, 4."(ed.Michi), 285-311.

47 Waksman,A.(1966), An optimum solution to the firing squad synchronization problem. *Inform.Control*. 9, 67-78.

48 Yamada,H. and S.Amoroso(1969), Tessellation automata. *Inform. Control*. 14, 299-317.

49 Yau,S.S. and M.Orsic(1970), Fault diagnosis and repair of cutpoint cellular arrays. *IEEE Trans. Computers* C-19, 259-262.

50 Yoeli,M.(1965), A group-theoretical approach to two rail cascades. *IEEE Trans. Electronic Computers* EC-14, 815-822.

IECEJ = Institute of Electronics and Communication Engineers of Japan.

Appendix A.

Classification of $\{A_x(S,f)\}$ by their right to left information transmission capability, where $S = \{0,1\}$ and $x = (1,1)$. (See Chapter 4.)

Let f be given by
$$\begin{array}{c|cc} 0 & 0 & 1 \\ \hline 0 & a & c \\ 1 & b & d \end{array} \quad \text{and} \quad \begin{array}{c|cc} 1 & 0 & 1 \\ \hline 0 & e & g \\ 1 & f & h \end{array}, \text{ where}$$
a,b,c,d,e,f,g,and h are in S. With each f is associated an integer which is the decimal equivalent of a binary number abcdefgh. In the following table, numbers 0 to 255 represent the transition functions, and I, II, and III in the entries signify Classes I, II, and III, respectively. The k-values are indicated in brackets.

The results are summed up as follows.

Number of Classified Cells									Number of Unclassified Cells			
I (k-value)							II	III	not I	not III	no information	total
1	2	3	4	5	6	7						
64	34	19	5	1	0	1	47	29	29	3	24	56

f	Class	f	Class	f	Class	f	Class
0	I (1)	32	I (1)	64	I (3)	96	
1	II	33	II	65	II	97	not I
2	I (1)	34	I (1)	66	I (4)	98	
3	II	35	II	67	not I	99	III
4	I (2)	36	I (2)	68	I (3)	100	
5	I (1)	37	I (1)	69	I (2)	101	I (2)
6	I (2)	38	I (2)	70		102	III
7	I (1)	39	I (1)	71	I (3)	103	I (3)
8	I (1)	40	I (1)	72	not I	104	not I
9	II	41	II	73	not I	105	III
10	I (1)	42	I (1)	74	I (4)	106	I (7)
11	II	43	II	75	II	107	II
12	I (2)	44	I (2)	76	III	108	III
13	I (1)	45	I (1)	77	II	109	II
14	I (2)	46	I (2)	78	I (4)	110	not III
15	I (1)	47	I (1)	79	I (2)	111	I (2)
16	II	48	III	80	I (1)	112	I (1)
17	II	49	III	81	II	113	II
18	II	50	III	82	I (1)	114	I (1)
19	II	51	III	83	not I	115	III
20	II	52		84	I (2)	116	
21	I (2)	53	II	85	I (1)	117	I (1)
22	not III	54	III	86		118	not I
23	I (2)	55	II	87	I (1)	119	I (1)
24	I (3)	56		88	I (1)	120	I (1)
25	II	57	III	89	not I	121	not I
26	I (3)	58		90	I (1)	122	I (1)
27	II	59	III	91	not I	123	III
28	II	60	III	92	I (3)	124	
29	I (2)	61	II	93	I (1)	125	I (1)
30	I (4)	62		94		126	not I
31	I (2)	63	II	95	I (1)	127	I (1)

f	Class	f	Class	f	Class	f	Class
128	I (1)	160	I (1)	192	III	224	I (3)
129	II	161	II	193	not I	225	not I
130	I (1)	162	I (1)	194		226	I (3)
131	II	163	II	195	III	227	not I
132	I (2)	164	I (2)	196		228	not I
133	I (1)	165	I (1)	197	I (2)	229	I (2)
134	I (2)	166	I (2)	198	III	230	
135	I (1)	167	I (1)	199	I (3)	231	I (3)
136	I (1)	168	I (1)	200	not I	232	II
137	II	169	II	201	III	233	not I
138	I (1)	170	I (1)	202	I (3)	234	I (3)
139	II	171	II	203	II	235	II
140	I (2)	172	I (2)	204	III	236	
141	I (1)	173	I (1)	205	II	237	II
142	I (2)	174	I (2)	206	II	238	I (3)
143	I (1)	175	I (1)	207	I (2)	239	I (2)
144	II	176		208	I (1)	240	I (1)
145	not I	177	not I	209	II	241	II
146		178	not I	210	I (1)	242	I (1)
147	III	179	III	211	not I	243	III
148		180	I (5)	212		244	
149	I (3)	181	I (2)	213	I (1)	245	I (1)
150	III	182		214		246	I (3)
151	I (4)	183	I (2)	215	I (1)	247	I (1)
152		184	I (3)	216	I (1)	248	I (1)
153	III	185	not I	217	not I	249	not I
154	not III	186	I (3)	218	I (1)	250	I (1)
155	II	187	III	219	not I	251	III
156	III	188		220		252	III
157	II	189	I (2)	221	I (1)	253	I (1)
158	II	190		222		254	I (2)
159	I (3)	191	I (2)	223	I (1)	255	I (1)

Appendix. B.

Indices concerning controllability of $\{A_{b_l}(S,f)\}$, where $S = \{0,1\}$ and $b_l = 1$, and the initial and target state sequence being $\Gamma_0 = \{0, 00, 000, \ldots\ldots\}$. (See Chapter 5.)

In the following tables, N_{st}, N_r, N_c, and N_{sp} stand for the indices of the strongly-connectedness, the reachability, the controllability, and the simply-connectedness, respectively. * is a symbol signifying 'more than or equal to 10.'

The results are summed up as follows.

Number of cells having a specified index											
	0	1	2	3	4	5	6	7	8	9	*	∞
N_{st}	112	72	44	11	1	/	/	/	/	/	/	16
N_r	64	86	68	21	1	/	/	/	/	/	/	16
N_c	64	56	31	16	12	3	2	/	/	2	39	31
N_{sp}	16	27	15	8	9	2	2	/	2	/	141	34

f	N_{st}	N_r	N_c	N_{sp}	f	N_{st}	N_r	N_c	N_{sp}	f	N_{st}	N_r	N_c	N_{sp}	f	N_{st}	N_r	N_c	N_{sp}
0	0	0	∞	∞	32	0	0	∞	∞	64	1	1	3	*	96	2	2	*	*
1	0	0	∞	∞	33	0	0	∞	∞	65	1	1	3	*	97	2	2	*	*
2	0	0	∞	∞	34	0	0	∞	∞	66	1	1	4	*	98	3	3	*	*
3	0	0	∞	∞	35	0	0	∞	∞	67	1	1	4	*	99	∞	∞	∞	∞
4	0	0	1	1	36	0	0	1	1	68	2	2	3	*	100	3	3	4	*
5	0	0	0	0	37	0	0	0	0	69	0	1	0	*	101	0	1	0	*
6	0	0	1	1	38	0	0	1	1	70	2	2	9	*	102	∞	∞	∞	∞
7	0	0	0	0	39	0	0	0	0	71	0	1	0	2	103	0	1	0	2
8	0	0	2	2	40	0	0	2	*	72	1	1	4	*	104	3	3	*	*
9	0	0	2	2	41	0	0	2	*	73	1	1	5	*	105	∞	∞	∞	∞
10	0	0	1	1	42	0	0	1	*	74	1	1	1	3	106	1	2	1	*
11	0	0	1	1	43	0	0	1	*	75	1	1	1	*	107	1	3	1	*
12	0	0	1	1	44	0	0	1	1	76	2	2	3	*	108	∞	∞	∞	∞
13	0	0	0	0	45	0	0	0	0	77	0	1	0	*	109	0	1	0	*
14	0	0	1	1	46	0	0	1	1	78	1	2	1	3	110	1	3	1	5
15	0	0	0	0	47	0	0	0	0	79	0	1	0	1	111	0	1	0	1
16	1	1	∞	∞	48	1	1	∞	∞	80	1	1	1	*	112	1	2	1	2
17	1	1	∞	∞	49	1	1	*	*	81	1	1	1	*	113	1	2	1	4
18	1	1	∞	∞	50	2	2	*	*	82	2	2	2	*	114	3	3	3	*
19	1	1	∞	∞	51	∞	∞	∞	∞	83	1	2	1	*	115	1	2	1	*
20	1	1	2	2	52	2	2	2	*	84	2	2	2	*	116	2	2	3	*
21	0	1	0	1	53	0	2	0	*	85	0	1	0	*	117	0	1	0	*
22	1	1	2	6	54	∞	∞	∞	∞	86	2	2	*	*	118	2	2	*	*
23	0	1	0	1	55	0	3	0	*	87	0	1	0	*	119	0	1	0	*
24	1	1	4	4	56	2	2	*	*	88	2	2	4	*	120	2	2	*	*
25	1	1	4	4	57	∞	∞	∞	∞	89	1	2	1	*	121	1	2	1	*
26	1	1	2	4	58	1	1	*	*	90	2	2	2	2	122	3	3	3	*
27	1	1	2	2	59	2	2	*	*	91	1	2	1	*	123	1	2	1	*
28	1	1	2	5	60	∞	∞	∞	∞	92	2	2	3	*	124	2	2	4	*
29	0	1	0	1	61	0	2	0	*	93	0	1	0	*	125	0	1	0	*
30	1	1	2	2	62	1	1	*	*	94	2	2	*	*	126	2	2	*	*
31	0	1	0	1	63	0	1	0	∞	95	0	1	0	*	127	0	1	0	*

f	N_{st}	N_r	N_c	N_{sp}	f	N_{st}	N_r	N_c	N_{sp}	f	N_{st}	N_r	N_c	N_{sp}	f	N_{st}	N_r	N_c	N_{sp}
128	0	0	4	4	160	0	0	*	*	192	2	2	∞	∞	224	1	1	3	4
129	0	0	*	*	161	0	0	4	*	193	3	3	5	*	225	2	2	*	*
130	0	0	6	*	162	0	0	3	*	194	3	3	*	*	226	1	1	*	*
131	0	0	*	*	163	0	0	3	*	195	∞	∞	∞	∞	227	3	3	4	*
132	0	0	1	1	164	0	0	1	1	196	2	2	*	*	228	1	1	*	*
133	0	0	0	0	165	0	0	0	0	197	0	1	0	*	229	0	1	0	*
134	0	0	1	1	166	0	0	1	1	198	∞	∞	∞	∞	230	1	1	9	*
135	0	0	0	0	167	0	0	0	0	199	0	1	0	2	231	0	1	0	2
136	0	0	2	4	168	0	0	2	4	200	3	3	*	*	232	1	1	3	4
137	0	0	2	8	169	0	0	2	*	201	∞	∞	∞	∞	233	2	2	*	*
138	0	0	1	3	170	0	0	1	*	202	1	2	1	2	234	1	1	1	2
139	0	0	1	*	171	0	0	1	*	203	1	3	1	*	235	1	3	1	*
140	0	0	1	1	172	0	0	1	1	204	∞	∞	∞	∞	236	1	1	*	*
141	0	0	0	0	173	0	0	0	0	205	0	1	0	*	237	0	1	0	*
142	0	0	1	1	174	0	0	1	1	206	1	2	1	*	238	1	1	1	2
143	0	0	0	0	175	0	0	0	0	207	0	1	0	1	239	0	1	0	1
144	1	1	6	8	176	1	1	2	*	208	1	3	1	*	240	1	1	1	1
145	1	1	*	*	177	1	1	2	*	209	1	2	1	*	241	1	2	1	3
146	2	2	*	*	178	2	2	3	*	210	2	3	2	*	242	1	1	*	*
147	∞	∞	∞	∞	179	2	3	2	*	211	1	2	1	*	243	1	2	1	∞
148	2	2	2	*	180	2	2	2	*	212	2	2	*	*	244	2	2	*	*
149	0	2	0	*	181	0	2	0	*	213	0	1	0	*	245	0	1	0	*
150	∞	∞	∞	∞	182	3	3	*	*	214	2	2	*	*	246	2	2	4	6
151	0	2	0	3	183	0	2	0	*	215	0	1	0	*	247	0	1	0	*
152	2	2	*	*	184	2	2	3	*	216	3	3	*	*	248	1	1	2	3
153	∞	∞	∞	∞	185	2	3	2	*	217	1	2	1	*	249	1	2	1	*
154	1	1	2	*	186	1	1	3	*	218	2	2	2	*	250	1	1	*	*
155	2	3	2	*	187	2	2	2	*	219	1	2	1	*	251	1	2	1	*
156	∞	∞	∞	∞	188	4	4	*	*	220	2	2	*	*	252	2	2	∞	∞
157	0	2	0	*	189	0	2	0	*	221	0	1	0	*	253	0	1	0	*
158	1	1	2	3	190	1	1	5	*	222	2	2	*	*	254	1	1	3	3
159	0	1	0	2	191	0	1	0	*	223	0	1	0	*	255	0	1	0	∞