Accepting Powers of
Some Four-Dimensional Automata

(いくつかの4次元オートマトンの受理能力について)

平成 29 年 3 月

宮崎大学大学院

農学工学総合研究科

物質・情報工学専攻

長友　誠

Accepting Powers of Some Four-Dimensional Automata

March, 2017

Makoto Nagatomo

Graduate School of Agriculture and Engineering,
University of Miyazaki

Abstract

This dissertation is a study of four-dimensional automata. In 1936, Turing machine was introduced as a simple mathematical model of computers. In theoretical computer science, Turing machine has played a number of important roles in understanding and exploiting basic concepts and mechanisms in computing and information processing. After that, it has become increasingly apparent that the characterization and classification of powers of the restricted Turing machines should be of great importance. Such a study was called automata theory, and it was active in 1950's and 1960's. In 1967, Blum M. and Hewitt C. first proposed two-dimensional automata as a computational model of two-dimensional pattern processing, and investigated their pattern recognition abilities. Since then, many researchers in this field have been investigating many important properties about automata on a two-dimensional tape. By the way, the question of whether processing three-dimensional digital patterns is much more difficult than two-dimensional ones is of great interest from the theoretical and practical standpoints. Recently, due to the advances in many application areas such as computer vision, robotics, and so forth, it has become increasingly apparent that the study of four-dimensional pattern processing has been of crucial importance. Thus, the research of four-dimensional automata as a computational model of four-dimensional pattern processing has also been meaningful.

The main purpose of this dissertation is to investigate a couple of properties of four-dimensional Turing machine and four-dimensional automata.

This dissertation consists of six chapters.

Chapter 1 provides the background and the motive of the study of four-dimensional

automata, and summarizes the main results in this dissertation.

Chapter 2 summarizes the formal definitions and notations necessary for the studies from Chapters 3 through 6.

Chapter 3 introduces a seven-way four-dimensional Turing machine, and investigates fundamental properties of 4-NMA_1 whose input tapes are restricted to rectangular ones. Necessary space for seven-way four-dimensional Turing machine to simulate four-dimensional one-marker automata are investigated.

Chapter 4 proposes a four-dimensional alternating Turing machine whose input tape restricted to cubic ones, and we presented a technique which we can show a four-dimensional language is not accepted by space-bounded alternating Turing machines.

In Chapter 5, first, we proposed a homogeneous systolic pyramid automata with four-dimensional layers. Second, we compared four-dimensional homogeneous systolic pyramid automata with one-way four-dimensional cellular automata (1-4*CA*).

In Chapter 6, we conclude this dissertation by summarizing the results and discussing the problems which have been argued throughout the dissertation.

Acknowledgements

Over the Ph.D. course of the three years that I was working on this dissertation, a number of researchers and colleagues have made valuable comments and fruitful suggestions on earlier draft. I would like to express my gratitude to all of them, in particular, the members of my doctoral committee at University of Miyazaki, Professor Tohru TSUJIKAWA, Professor Masato IIDA, Professor Takuya TETSUMURA, Associate Professor Makoto SAKAMOTO and Associate Professor Satoshi IKEDA. A very special debt is owed Associate Professor Makoto SAKAMOTO, who suggested a large number of improvement, both in content and exposition, that I feel have greatly increased the allover quality of this dissertation.

I am greatly indebted to Professor Katsushi INOUE at Yamaguchi University, Professor Takao ITO at Hiroshima University, Professor Yasuo UCHIDA at Ube National College of Technology, Tsunehiro YOSHINAGA at National Institute of Technology, Tokuyama College and Masanori SUGISAKA, president of Alife Robotics Co., ltd.

I also gratefully acknowledge the strong support and constant encouragement in the preparation of this dissertation offered by members of SAKAMOTO Laboratory. They gave me a lot of assistance.

Finally, on this occasion I wish to express my thanks to the staffs of interdisciplinary Graduate School of Agriculture and Engineering of for their constant support for my research activities.

Glossary

(1) Symbolic Logic

Term	Interpretation
$\forall x$	for each x
$\exists x$	for some x
$\Rightarrow$	If . . . then
$\Leftrightarrow$	if and only if

(2) Set Theory

Term	Interpretation
φ	empty
∞	infinity
$\|A\|$	cardinality of a set A
$a \in A$	a is an element of A
$a \notin A$	a is not an element of A
$A \subseteq B$	A is a subset of B
*A*Ç*B*	A is a proper subset of B
$A \cup B$	union of sets A and B
$A \cap B$	intersection of sets A and B
$\overline{A}$	complement of a set A
$A - B$	complement of B in A
$A \times B$	Cartesian product of sets A and B
A^k	k-fold Cartesian product of a set A
$P(A) = 2^A$	power set of a set A
$\boldsymbol{N}$	the set of natural numbers

$\boldsymbol{Z}$	the set of integers
$\boldsymbol{R}$	the set of real numbers
$\lceil x \rceil$	greatest integer smaller than or equal to x
$\lfloor x \rfloor$	smallest integer greater than or equal to x
$f : A \mapsto B$	f is a mapping from a set A into a set B
$g : o(f)$	g is a function such that for every r>0 and for all but finitely many m, $g(m) < rf(m)$
$g : O(f)$	g is a function such that for some positive constant r>0 and for all but finitely many m, $g(m) \leq rf(m)$
$g : \Omega(f)$	g is a function such that for some positive constant r>0 and for all but finitely many m, $g(m) > rf(m)$

(3) Symbols for Input Tapes

<u>Term</u>	<u>Interpretation</u>
Σ^*	set of all strings over Σ
Σ^+	set of all nonempty strings over Σ
$\|w\|$	length of a string w
$\Sigma^{(3)}$	set of all three-dimensional tape over Σ
$\Sigma^{(4)}$	set of all four-dimensional tape over Σ
$l_j(x)$	length of three-dimensional tape x along the j-th axis $(1 \leq j \leq 4)$
$x[(i_1, i_2, i_3, i_4), (i'_1, i'_2, i'_3, i'_4)]$	$[(i_1, i_2, i_3, i_4), (i'_1, i'_2, i'_3, i'_4)]$-segment of four-dimensional tape x
$L[M]$	class of the sets of all input tapes accepted by the automata M's

(4) Abbreviations for Three-Dimensional Automata

Term	Interpretation
ATM	three-dimensional alternating Turing machine
3-NTM	three-dimensional nondeterministic Turing machine
3-DTM	three-dimensional deterministic Turing machine
FV 3-ATM	five-way three-dimensional alternating Turing machine
FV 3-NTM	five-way three-dimensional nondeterministic Turing machine
FV 3-DTM	five-way three-dimensional deterministic Turing machine
3-$ATM(L(m))$	$L(m)$ space-bounded three-dimensional alternating Turing machine
3-$NTM(L(m))$	$L(m)$ space-bounded three-dimensional nondeterministic Turing machine
3-$DTM(L(m))$	$L(m)$ space-bounded three-dimensional deterministic Turing machine
FV 3-$ATM(L(m))$	$L(m)$ space-bounded five-way three-dimensional alternating Turing machine
FV 3-$UTM(L(m))$	$L(m)$ space-bounded five-way three-dimensional alternating Truing machine with only universal states
FV 3-$NTM(L(m))$	$L(m)$ space-bounded five-way three-dimensional nondeterministic Turing machine
FV 3-$DTM(L(m))$	$L(m)$ space-bounded five-way three-dimensional deterministic Turing machine
3-AFA	three-dimensional alternating finite automaton

3-*NFA*	three-dimensional nondeterministic finite automaton
3-*DFA*	three-dimensional deterministic finite automaton
FV 3-*AFA*	five-way three-dimensional alternating finite automaton
FV 3-*NFA*	five-way three-dimensional nondeterministic finite automaton
FV 3-*DFA*	five-way three-dimensional deterministic finite automaton
3-*NFA*(*k-heads*)	*k* heads 3-*NFA*
3-*DFA*(*k-heads*)	*k* heads 3-*DFA*
3-NM_k	three-dimensional nondeterministic *k*-marker automaton
3-DM_k	three-dimensional deterministic *k*-marker automaton

(5) Abbreviations for Four-Dimensional Automata

Term	Interpretation
4-*ATM*	four-dimensional alternating Turing machine
4-*NTM*	four-dimensional nondeterministic Turing machine
4-*DTM*	four-dimensional deterministic Turing machine
SV 4-*ATM*	seven-way four-dimensional alternating Turing machine
SV 4-*NTM*	seven-way four-dimensional nondeterministic Turing machine
SV 4-*DTM*	seven-way four-dimensional deterministic Turing machine
4-*ATM*($L(m)$)	$L(m)$ space-bounded four-dimensional alternating Turing machine
4-*NTM*($L(m)$)	$L(m)$ space-bounded four-dimensional nondeterministic

	Turing machine
4-$DTM(L(m))$	$L(m)$ space-bounded four-dimensional deterministic Turing machine
SV4-$ATM(L(m))$	$L(m)$ space-bounded seven-way four-dimensional alternating Turing machine
SV4-$NTM(L(m))$	$L(m)$ space-bounded seven-way four-dimensional nondeterministic Turing machine
SV4-$DTM(L(m))$	$L(m)$ space-bounded seven-way four-dimensional deterministic Turing machine
4-AFA	three-dimensional alternating finite automaton
3-NFA	three-dimensional nondeterministic finite automaton
3-DFA	three-dimensional deterministic finite automaton
FV3-AFA	five-way three-dimensional alternating finite automaton
FV3-NFA	five-way three-dimensional nondeterministic finite automaton
FV3-DF	five-way three-dimensional deterministic finite automaton
4-NM_k	four-dimensional nondeterministic k-marker automaton
4-DM_k	four-dimensional deterministic k-marker automaton
1-4CA	one-way four dimensional cellular automaton

Contents

Chapter 1

Introduction

There are two major components in computer science: The first one is the fundamental mathematics and theories underlying computing, and the second one is engineering techniques for the design of computer systems, including hardware and software.

Theoretical computer science falls under the first area of the two major components. It begins in various fields: physics, mathematics, linguistics, electric and electronic engineering, physiology, and so on. Most of these studies have become important ideas and models that are central to theoretical computer science [1, 12].

In theoretical computer science, the Turing machine has played a number of important roles in understanding and exploiting basic concepts and mechanisms in computing and information processing. It is a simple mathematical model of computers which was introduced by Turing A. M. in 1936 to answer one of the fundamental issues in computer science — "What kind of logical work can we effectively perform?" [81]. If the restrictions in its structure and move are placed on the Turing machine, the restricted Turing machine is less powerful than the original one. However, it has become increasingly apparent that the characterization and classification of powers of the restricted Turing machines should be of great important. Such a study was active in 1950's and 1960's. On the other hand, many researchers have been making their efforts to investigate another fundamental issue of computer science ― "How complicated is it to perform a given logical work?". The concept of computational complexity is a formalization of such difficulty of logical works.

In the study of computational complexity, the complexity measures are of great importance. In general, it is well known that the computational complexity has originated in a study of considering how the computational powers of various types of automata are characterized by the complexity measures such as space complexity, time complexity, or some other related measures. Especially, the concept of complexity is very useful to characterize various types of automata from a viewpoint of memory requirements [32]. This study was motivated by Stearns R. E., Hartmanis J., and Lewis P. M. in 1965 [74]. They introduced an $L(m)$ space-bounded one-dimensional Turing machine to formalize the notion of space complexity, and investigated its computing ability. Some results were refined by Hopcroft J. E. and Ullman J. D. [10-12]. Moreover, Chandra A. K., Kozen D. C., and Stockmeyer L. J. introduced an alternating Turing machine as a theoretical model of parallel computation in 1981 [5]. An alternating Turing machine, whose state set is partitioned into two disjoint sets, the set of universal states and the set of existential states, is a generalization of a non- deterministic Turing machine. A nondeterministic Turing machine is an alternating Turing machine which has only existential states. In related paper, several investigations of these machines have been continued [8, 13, 32, 33, 35, 44, 45, 48, 75].

After that, the development of the processing of pictorial information by computer was rapid in those days. Therefore, the problem of computational complexity was also arisen in the two-dimensional information processing. Blum M. and Hewitt C. first proposed two-dimensional automata ― two-dimensional finite automata and marker automata, and investigated their pattern recognition abilities in 1967 [3]. Since then, many researchers in this field have been investigating properties about automata on a two-dimensional tape. For example, Morita K., Umeo H., and Sugata K. proposed an $L(m, n)$ space-bounded two-dimensional Turing machine and its variants to formalize memory limited computations in the two-dimensional information processing [37-40]. Ito A., Inoue K., Takanami I, and Taniguchi H. introduced two-dimensional alternating Turing machines as a generalization of two-dimensional nondeterministic Turing machines and as a mechanism to model parallel computation. Restricted version of two-dimensional alternating Turing machines were investigated [22, 24-26]. Special types of two-dimensional Turing machines (two-dimensional pushdown automata, stack automata, multicounter automata, multihead automata, and marker automata) were investigated [14, 20, 72, 73, 76]. Moreover, cellular automata on a two-dimensional tape were investigated not only from the viewpoint of formal language theory, but also from the viewpoint of pattern recognition. Cellular automata on a two-dimensional tape can be classified into three types. The first type, called a two-dimensional cellular automaton, is investigated [2, 7, 8]. Especially, many properties of two-dimensional on-line tessellation acceptors, which are restricted type of two-dimensional cellular automata, are investigated [15-17, 19]. The second type of cellular automata on a two-dimensional tape is investigated [18, 46, 47, 78]. Two typical models of this type are parallel / sequential array automata and one-dimensional bounded cellular acceptors. The third type, called a pyramid cellular acceptor, is investigated [21]. More detailed survey of two-dimensional automata theory is done by Inoue K. and Takanami I. [23].

By the way, the question of whether processing three-dimensional digital patterns is much difficult than two-dimensional ones is of great interest from the theoretical and practical standpoints both. In recent years, due to the advances in many application areas such as computer graphics, computer-aided design / manufacturing, computer vision, image processing, robotics, and so on, the study of three-dimensional pattern processing has been of crucial importance. Thus, the research of three-dimensional automata as the computational model of three-dimensional pattern processing has been meaningful. However, it is conjectured that the three-dimensional pattern processing has its own difficulties not arising in two-dimensional case. One of these difficulties occurs in recognizing topological properties of three-dimensional patterns because the three-dimensional neighborhood is more complicated than two-dimensional case. Generally speaking, a property or relationship is topo- logical only if it is preserved when an arbitrary "rubber-sheet" distortion is applied to the pictures. For example, adjacency and connectedness are topological; area, elongatedness, convexity, straightness, etc. are not.

During the past about thirty years, automata on a three-dimensional tape have been proposed and several properties of such automata have been obtained. Inoue K. and Nakamura I. proposed an n-dimensional on-line tessellation acceptor which can decide whether an n-dimensional tape is accepted or not by the on-line and parallel processing [16]. Blum M. and Sakoda W. J. investigated the capability of finite automata in two-dimensional and three-dimensional space [4]. Yamamoto Y., Morita K., and Sugata K. introduced a three-dimensional k-marker automaton, an $L(m)$ space-bounded three-dimensional Turing machine and an $L(m)$ space-bounded five-way three-dimensional Turing machine [83]. They studied the problem of recognizing connectedness of three-dimensional patterns by these machines. Taniguchi H., Inoue K., and Takanami I. investigated the relationship between the accepting powers of three-dimensional finite automata and five-way three-dimensional Turing machines [77]. They also proposed a k-neighborhood template A-type two-dimensional bounded cellular acceptor which consists of a pair of a converter and a configuration-reader, as the computational model of three-dimensional pattern process. The converter converts the given three-dimensional tape to the two-dimensional configuration, and the configuration-reader determines the acceptance or nonacceptance of given three-dimensional tape, depending on whether or not the derived two-dimensional configuration is accepted [78, 79]. Nakamura A. and Aizawa K. proposed the interlocking component which is a chainlike connectivity a new topological property of three-dimensional digital pictures, and investigated the recognizability of interlocking components [42]. Sakamoto M. et al. proposed several three- or four-dimensional automata, and showed their properties [43, 49-59]. Moreover, Ito T. et al. investigated about synchronized alternation and parallelism for three-dimensional automata [27-30, 59].

In this dissertation, we introduce some four-dimensional automata and investigate their various properties. The dissertation has six chapters in addition to this Introduction. Chapter 2 gives definitions and notations necessary for Chapters 3 through 6.

We show in Chapter 3 many investigations about four-dimensional automata which have been accomplished until now. We can observe the historical review of properties of four-dimensional automata before beginning the main subject.

In Chapter 4, we investigate some accepting powers of four-dimensional alternating Turing machines whose input tapes are restricted to cubic ones and show the space lower bound technique for four-dimensional alternating Turing machines.

In Chapter 5, we deal with four-dimensional homogeneous systolic pyramid automata. We first proposed a four-dimensional homogeneous systolic pyramid automaton. Next, we compared a four-dimensional homogeneous systolic pyramid automaton with one-way a four-dimensional cellular automaton.

In Chapter 6, we summarize the results and discuss the problems which have arisen throughout the study.

It has often been noticed that we can easily get several properties of four-dimensional automata by directly applying the results of one- or two- or three-dimensional case, if the input tapes are not restricted to cubic ones. So we state these subjects only for cubic tapes in Chapters 5.

Chapter 2

Definitions and Notations

This chapter summarizes the formal definitions and notations necessary for this dissertation. We first define a four-dimensional tape which is the input tape of four-dimensional automata. Next, we define four-dimensional Turing machine, and their related notations.

2.1 Four-Dimensional Tape

Definition 2.1. Let Σ be a finite set of symbols. A *four-dimensional tape* over Σ is a four-dimensional rectangular array of elements of Σ. The set of all the four-dimensional tapes over Σ is denoted by $\Sigma^{(4)}$. Given a tape $x \in \Sigma^{(4)}$, for each integer j ($1 \leq j \leq 4$), we let $l_j(x)$ be the length of x along the jth axis. The set of all $x \in \Sigma^{(4)}$ with $l_1(x) = n_1$, $l_2(x) = n_2$, $l_3(x)=n_3$ and $l_4(x)=n_4$ is denoted by $\Sigma^{(n_1,n_2,n_3,\, n_4)}$. When $1 \leq i_j \leq l_j(x)$ for each j ($1 \leq j \leq 4$), let $x(i_1,i_2,i_3,i_4)$ denote the symbol in x with coordinates (i_1,i_2,i_3,i_4) as shown in Fig. 2.1.

Furthermore, we define

$$x[(i_1,i_2,i_3,i_4), (i'_1, i'_2, i'_3, i'_4)],$$

when $1 \leq i_j \leq i'_j \leq l_j(x)$ for each integer j ($1 \leq j \leq 4$), as the four-dimensional input tape satisfying the following (i) and (ii):

(i) for each j $(1 \leq j \leq 3)$, $l_j(y) = i'_j - i_j + 1$;

(ii) for each r_1,r_2,r_3,r_4 $(1 \leq r_1 \leq l_1(y),\ 1 \leq r_2 \leq l_2(y),\ 1 \leq r_3 \leq l_3(y),\ 1 \leq r_4 \leq l_4(y))$, $y(r_1,r_2,r_3,r_4) = x(r_1+i_1-1,\ r_2+i_2-1,\ r_3+i_3-1,\ r_4+i_4-1)$. (We call $x[(i_1,i_2,i_3,i_4), (i'_1, i'_2, i'_3, i'_4)]$ the $[(i_1,i_2,i_3,i_4), (i'_1, i'_2, i'_3, i'_4)]$-*segment* of x.).

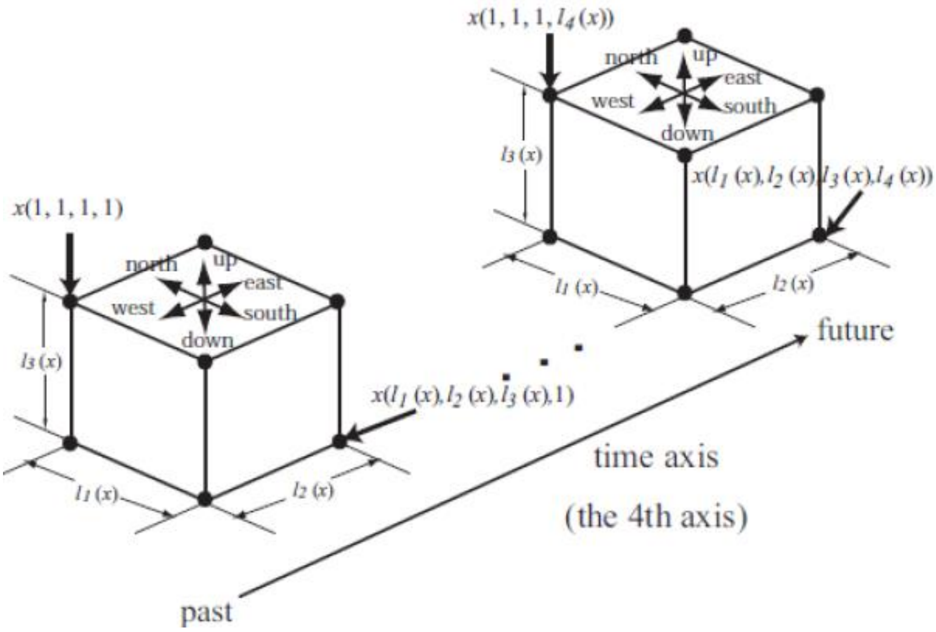

Fig. 2.1: Four-dimensional input tape and coordinates of each cell.

2.2 Four-Dimensional Turing Machine

We now introduce a *four-dimensional Turing machine.*

Definition 2.2. As *four-dimensional Turing machine* has a 6-tuple

$$M = (Q, q_0, F, \Gamma, \Sigma, \delta),$$

where

(1) Q is a finite set of *states*,

(2) $q_0 \in Q$ is the *initial state*,

(3) $F \subseteq Q$ is the set of *accepting states*,

(4) Σ is a finite *input alphabet* ($\# \notin \Sigma$ is the *boundary symbol*),

(5) Γ is a finite *storage tape alphabet* containing the special blank symbol B.

(6) $\delta \subseteq (Q \times (\Sigma \cup \{\#\}) \times \Gamma) \times (Q \times (\Gamma - \{B\}) \times \{east, west, south, north, up, down, no\ move\} \times \{left, right, no\ move\})$ is the *next move relation*.

A *step* of M consists of reading one symbol from each tape, writing a symbol on the storage tape, moving the input and storage tape heads in specified directions, and entering a new state, according to the next move relation δ. When entering accepting state, it stops processes.

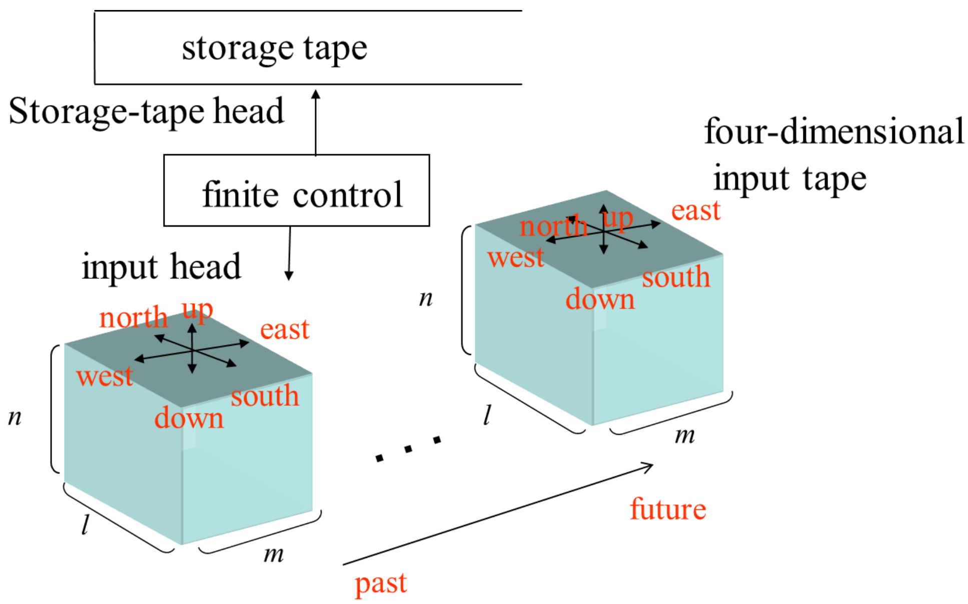

Fig. 2.2 : Four-dimensional Turing machine

Chapter 3

Necessary Spaces for Seven-Way Four-Dimensional Turing Machines to Simulate Four-Dimensional One-Marker Automata

An improvement of picture recognizability of the finite automaton is the reason why the marker automaton was introduced. That is, a two-dimensional one-marker automaton can recognize connected pictures. This automaton has been widely investigated in the two- or three-dimensional case [55]. A multi-marker automaton is a finite automaton which keeps marks as 'pebbles' in the finite control, and cannot rewrite any input symbols but can make marks on its input with the restriction that only a bounded number of these marks can exist at any given time[3].

As is well known among the researchers of automata theory, one-dimensional one-marker automata are equivalent to ordinary finite state automata. In other words, there is no need of working space usage for one- way Turing machines to simulate one-marker automata, as well as finite state automata.

In the two-dimensional case, the following facts are known : the necessary and sufficient space for three-way two-dimensional deterministic Turing machines *TR*2-*DTM*'s to simulate two-dimensional deterministic (nondeterministic) finite automata 2-*DFA*'s (2-

NFA's) is $m\log m$ (m^2) and the corresponding space for three-way two-dimensional nondeterministic Turing machines *TR*2-*NTM*'s is m (m), whereas the necessary and sufficient space for three-way two-dimensional deterministic Turing machines *TR*2-*DTM*'s to simulate two-dimensional deterministic (nondeterministic) one-marker automata 2-DMA_1's (2-NMA_1's) is $2^{m\log m}$ (2^{m^2}) and the corresponding space for *TR*2-*NTM*'s is $m\log m$(m^2), where m is the number of columns of two-dimensional rectangular input tapes.

In the three-dimensional case, the following facts are known : the necessary and sufficient space for five-way three-dimensional deterministic Turing machines *FV*3- *DTM*'s to simulate three-dimensional deterministic (nondeterministic) finite automata 3-*DFA*'s (3-*NFA*'s) is $m^2\log m$ (m^3) and the corresponding space for five-way three-dimensional nondeterministic Turing machines *FV*3-*NTM*'s is m^2 (m^2), whereas the necessary and sufficient space for five-way three-dimensional deterministic Turing machines *FV*3-*DTM*'s to simulate three-dimensional deterministic (nondeterministic) one-marker automata 3-DMA_1's (3-NMA_1's) is $2^{lm\log lm}$ ($2^{l^2m^2}$) and the corresponding space for *FV*3-*NTM*'s is $lm\log lm$ (l^2m^2), where l(m) is the number of rows (columns) on each plane of three-dimensional rectangular input tapes. In the four-dimensional case, we showed the sufficient spaces for four-dimensional Turing machines to simulate four-dimensional one-marker automata [41]. In this paper, we continue the investigations, and deal with the necessary spaces for four-dimensional Turing machines to simulate four-dimensional one-marker automata.

3.1 Preliminaries

An ordinary finite automaton cannot rewrite any symbols on input tape, but a marker automaton can make a mark on the input tape. We can think of the mark as a 'pebble' that M puts down in a specified position. If M has already put down the mark, and wants to put it down elsewhere, M must first go to the position of the mark and pick it up. Formally, we define it as follows.

Definition 3.1.1 We now introduce a space bounded seven-way *four-dimensional Turing machine*.

A space bounded seven-way *four-dimensional Turing machine* (denoted by (SV 4-$TM(L(l,m,n))$) has a 6-tuple

$$M = (Q, q_0, F, \Gamma, \Sigma, \delta),$$

where

(7) Q is a finite set of *states*,

(8) $q_0 \in Q$ is the *initial state*,

(9) $F \subseteq Q$ is the set of *accepting states*,

(10) Σ is a finite *input alphabet* ($\# \notin \Sigma$ is the *boundary symbol*),

(11) Γ is a finite *storage tape alphabet* containing the special blank symbol B.

(12) $\delta \subseteq (Q \times (\Sigma \cup \{\#\}) \times \Gamma) \times (Q \times (\Gamma - \{B\}) \times \{east, west, south, north, up, down, no\ move\} \times \{left, right, no\ move\})$ is the *next move relation*.

A *step* of M consists of reading one symbol from each tape, writing a symbol on the storage tape, moving the input and storage tape heads in specified directions, and entering a new state, according to the next move relation δ. When entering accepting state, it stops processes.

Let $L(l, m, n) : \mathbf{N}^3 \rightarrow \mathbf{R}$ be a function. A seven-way four-dimensional Turing machine M is said to be $L(l, m, n)$ space-bounded if for each $l, m, n \geq 1$ and for each x with $l_1(x) = l$, $l_2(x) = m$, and $l_3(x) = n$, if x is accepted by M.

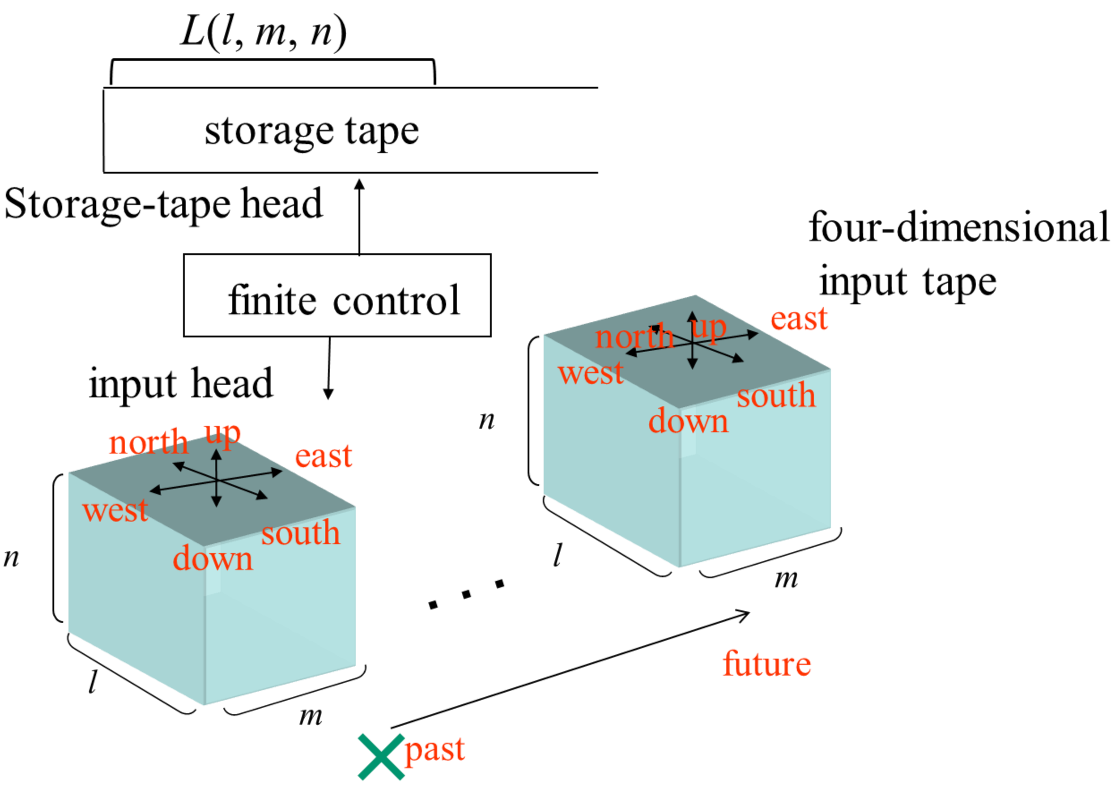

Fig. 3.1 : Space bounded seven-way four-dimensional Turing machine

Definition 3.1.2. A four-dimensional nondeterministic one-marker automaton (4-NMA_1) is defined by the 6-tuple $M = (Q, q_0, F, \Sigma, \{+, -\}, \delta)$, where

(1) Q is a finite set of states ;

(2) $q_0 \in Q$ is the initial state ;

(3) $F \subseteq Q$ is the set of accepting states ;

(4) Σ is a finite input alphabet ($\# \notin \Sigma$ is the boundary symbol);

(5) $\{+,-\}$ is the pair of signs of presence and absence of the marker ; and

(6) $\delta: (Q \times \{+,-\}) \times ((\Sigma \cup \{\#\}) \times \{+,-\}) \rightarrow 2^{Q\times\{+,-\}} \times ((\Sigma \cup \{\#\}) \times \{+,-\}) \times$ {east, west, south, north, up, down, future, past, no move} is the next-move function, satisfying the following : For any $q, q' \in Q$, any $a, a' \in \Sigma$, any $u, u', v, v' \in \{+,-\}$, and any $d \in$ {east, west, south, north, up, down, future, past, no move}, if $((q', u'), (a', v'), d) \in \delta((q, v), (a, v))$ then $a = a'$ and $(u, v, u', v') \in \{(+,-,+,-), (+,-,-,+), (-,+,-,+), (-,+,+,-), (-,-,-,-)\}$.

We call a pair (q, u) in $Q \times \{+,-\}$ an extended state, representing the situation that M holds or does not hold the marker in the finite control according to the sign $u = +$ or $u = -$, respectively. A pair (a, v) in $\Sigma \times \{+,-\}$ represents an input tape cell on which the marker exists or does not exist according to the sign $u = +$ or $u = -$, respectively.

Therefore, the restrictions on δ imply the following conditions. (i) When holding the marker, M can put it down or keep on holding. (ii) When not holding the marker, and ① if the marker exists on the current cell, M can pick it up or leave it there, or ② if the marker does not exist on the current cell, M cannot create a new marker any more.

Definition 3.1.3. Let Σ be the input alphabet of 4-NMA_1 M. An extended input tape $\tilde{x}$ of M is any four-dimensional tape over $\Sigma \times \{+,-\}$ such that for some $x \in \Sigma^{(4)}$,

(i) for each $j (1 \leq j \leq 4)$, $lj(\tilde{x}) = lj(x)$,

(ii) for each $i1 (1 \leq i1 \leq l1(\tilde{x}))$, $i2 (1 \leq i2 \leq l2(\tilde{x}))$, $i3 (1 \leq i3 \leq l3(\tilde{x}))$, and $i4 (1 \leq i4 \leq l4(\tilde{x}))$, $\tilde{x}(i1, i2, i3, i4) = x((i1, i2, i3, i4), u)$ for some $u \in \{+,-\}$.

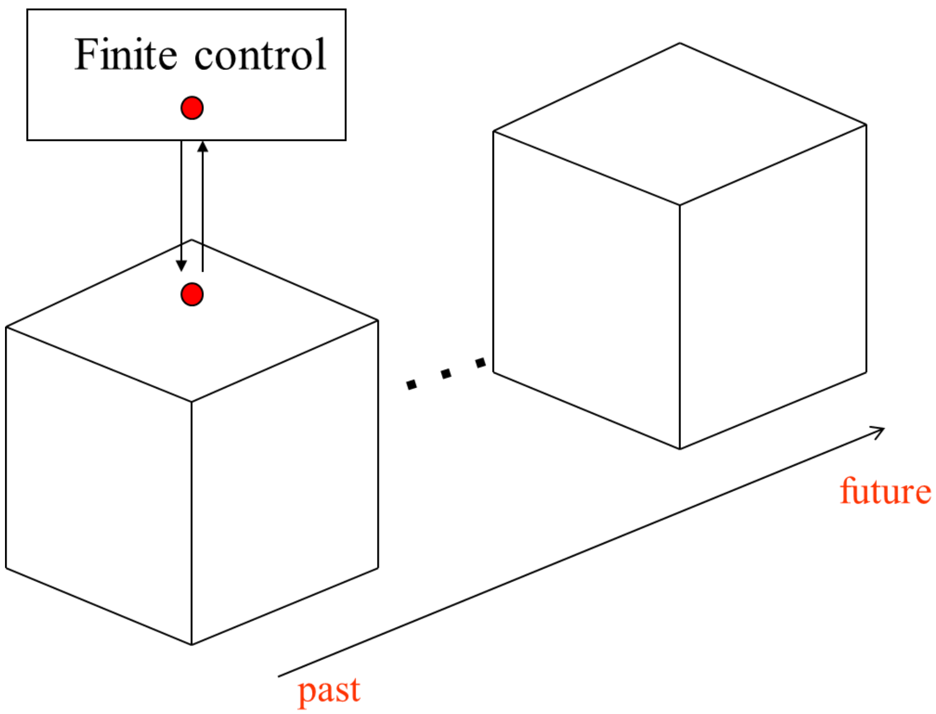

Fig. 3.2 : Four-dimensional one marker automaton

Definition 3.1.4. A *configuration* of 4-NMA_1 $M = (Q, q_0, F, \Sigma, \{+, -\}, \delta)$ is a pair of an element of $((\Sigma \cup \{\#\}) \times \{+, -\}$ and an element of $C = (\mathbf{N} \cup \{0\})^{(4)} \times (Q \times \{+, -\})$. The first component of a configuration $c = (\tilde{x}, ((i1, i2, i3, i4), (q, u)))$ represents the extended input tape of M. The second component (i_1, i_2, i_3, i_4) of c represents the input head position. The third component (q, u) represents the extended state. An element of C_M is called a semi-configuration of M. If q is the state associated with configuration c, then c is said to be an accepting configuration if q is an accepting state. The initial configuration of M on input x is $I_M(x) = (x-, ((1,1,1,1), (q_0, +)))$, where x^- is the special extended input tape of M such that $x^-(i_1, i_2, i_3, i_4) = (x(i_1, i_2, i_3, i_4), -)$ for each i_1, i_2, i_3, i_4 $(1 \leq i_1 \leq l_1(x^-), 1 \leq i_2 \leq l_2(x^-), 1 \leq i_3 \leq l_3(x^-), 1 \leq i_4 \leq l_4(x^-))$. If M moves deterministically, we call M a four-dimensional deterministic one-marker automaton (4-DMA_1).

Definition 3.1.5. Given a 4-NMA_1 $M = (Q, q_0, F, \Sigma, \{+, -\}, \delta)$, we write $c \vdash_M c'$ and say c' is a successor of c if configuration c' follows from configuration c in one step of M, according to the transition rules δ . $\vdash^*$ denotes the reflexive transitive closure of $\vdash_M$. The relation $\vdash M$ is not necessarily single-valued, because is not. A computation path of M on x is a sequence $c_0 \vdash^M c_1 \vdash^M \ldots \vdash^M c_n (n \geq 0)$, where $c_0 = I_M(x)$. An accepting computation path of M on x is a computation path of M on x which ends in an accepting configuration. We say that M accepts x if there is an accepting computation path of M on input x.

Let $L(l, m, n) : \mathbf{N}^3 \rightarrow \mathbf{R}$ be a function. A seven-way four-dimensional Turing machine M is said to be $L(l, m, n)$ space-bounded if for each $l, m, n \geq 1$ and for each x with $l_1(x) = l$, $l_2(x) = m$, and $l_3(x) = n$, if x is accepted by M, then there is an accepting computation path of M on x in which M uses no more than $L(l, m, n)$ cells of the storage tape. We denote an $L(l, m, n)$ space-bounded *SV*4- *DTM* (*SV*4-*NTM*) by *SV*4-*DTM*($L(l, m, n)$) (*SV*4- *NTM*($L(l, m, n)$)).

Definition 3.1.6. For any four-dimensional automaton M with input alphabet Σ, define $T(M) = \{x \in \Sigma^{(4)} \mid M \text{ accepts } x\}$. Furthermore, for each $X \in \{D,N\}$, define L[4-XMA_1] = $\{T \mid T = T(M)$ for some 4-$XMA_1\}$, L[*SV*4-*XTM*($S(l,m,n)$)] = $\{T \mid T = T(M)$ for some *SV*4-*XTM*($S(l,m,n)$) $M\}$, and L[*SV*4-*XTM*($L(l,m)$)] = $\{T \mid T = T(M)$ for some *SV*4- *XTM*((l,m)) $M\}$.

3.2 Necessary Space

In this section, we investigate the necessary spaces (i.e., lower bounds) for seven-way Turing machines to simulate one-marker automata.

Definition 3.2.1. Let x be in $\Sigma^{(4)}$ (Σ is a finite set of symbols) and $l_1(x) = l$, $l_2(x) = m$, $l_3(x) = n$. For each j ($1 \leq j \leq Q[l_4(x)/lmn]$) (where $Q[l_4(x)/lmn]$ denotes the quotient when $l_4(x)$ is divided by lmn), $x[(1, 1, 1, (j-1)lmn+1), (l, m, n, jlmn)]$ is called the jth (l, m, n)-block of x. We say that the tape

x has exactly k (l, m, n)-blocks if $l_4(x)=klmn$, where k is a positive integer.

Definition 3.2.2. Let (l_1, m_1, n_1), (l_2, m_2, n_2), … , be a sequence of points (i.e., pairs of three natural numbers), and let $\{(l_i, m_i, n_i)\}$ denote this sequence. We call a sequence $\{(l_1, m_1, n_1)\}$ the regular sequence of points if $(l_i, m_i, n_i) \neq (l_j, m_j, n_j)$ for $i \neq j$.

Lemma 3.2.1. Let $T_1=\{x \in \{0, 1\}(4) \mid \exists\ l \geq 1, \exists m\geq 1, \exists n\geq 1[l_1(x)=l$ and $l_2(x)=m$ and $l_3(x)=n$ and (each cuboid of x contains exactly one '1') and $\exists d \geq 2$ [(x has exactly $d(l, m, n)$-blocks, i.e., $l_4(x)=dlmn$) and (the last (l, m, n)-block is equal to some other (l, m, n)-block)]]$\}$. Then,

(1) $T_1 \in L[4\text{-}DMA_1]$, but

(2) $T_1 \notin L[SV4\text{-}DTM(2^{L(l, m, n)})]$ (so, $T_1 \notin L[SV4\text{-}NTM(L(l, m, n))]$) for any function $L(l, m, n)$ such that

$$\lim_{i\to\infty}[L(l_i, m_i, n_i)/(l_i m_i n_i \log l_i m_i n_i)] = 0.$$

for some regular sequence of points $\{(l_i, m_i, n_i)\}$.

Proof: (1): We construct a $4\text{-}DMA_1$ M accepting T_1 as follows. Given an input x with $l_1(x) = l$, $l_2(x) = m$, and $l_3(x) = n$, M first checks, by sweeping cuboid by cuboid, that each cuboid of x contains exactly one '1', and M then checks, by making a zigzag of 45°-direction from top cuboid to bottom cuboid, that x has exactly d (l, m, n)-blocks for some integer $d \geq 2$. After that,

M tests by utilizing its own marker whether the last (l, m, n)-block is identical to some other (l, m, n)-block. M then finds the '1' position on the cuboid and move up vertically from this position. In this course, each time M meets a '1' position, it checks whether or not there is a marker on the cuboid (containing the '1' position).

(i):If there is a marker on the cuboid, M knows that the kth cuboids of the hth and the last (l, m, n)-blocks are identical, and so M then tries to check whether the next (k+1)th cuboids of the hth and the last (l, m, n)-blocks are identical.

(ii):If there is no marker on the cuboid, M goes back to the '1' position on the cuboid, and vertically moves up again to find the next '1' position. In this case, if M eventually encounters the top boundary, M knows that the kth cuboids of the hth and the last (l, m, n)-blocks are different (thus, the hth (l, m, n)-block is not identical to the last (l, m, n)-block), and so M then tries to check whether the next (h+1)th (l, m, n)-block is identical to the last (l, m, n)-block.

In this way, M enters an accepting state just when it finds out some (l, m, n)-block, each of whose cuboids is identical to the corresponding cuboid of the last (l, m, n)-block. It will be obvious that $T(M)= T_1$.

(2):Suppose to the contrary that there exists an $SV4\text{-}DTM(2^{L(l,\,m,\,n)})$ M accepting T_1, where $L(l, m, n)$ is a function such that

$$\lim_{i\to\infty}[L(l_i,m_i,n_i)/(l_i m_i n_i \log l_i m_i n_i)] = 0.$$

For some regular sequence of points $\{(l_i, m_i, n_i)\}$. Let s and t be the numbers of states in the finite control and storage tape symbols of M, respectively. We assume without loss of generality that if M accepts an input, then M enters an accepting state on the bottom boundary. For each $l \geq 1$, $m \geq 1$, $n \geq 1$, let $V(l, m, n)= \{x \in T_1 \mid l_1(x)=l$ and $l_2(x)=m$ and $l_3(x)=n$ and (x has exactly $((lmn)lmn+1)$ (l, m, n)-blocks)$\}$. For each $x \in V(l, m, n)$, let $B(x)=\{b \in \{0, 1\}^{(4)}$

| $\exists h(1 \leq h \leq (lmn)lmn)$ [b is the hth (l, m, n)-block of x] }, and let $S(l, m, n)$= {$B(x)$ | $x \in V(l, m, n)$}. Note that for each $x \in (l, m, n)$, there is a sequence of configurations of M which leads M to an accepting state. Let $conf(x)$ be the semi-configuration just after M leaves the second-to-last (l, m, n)-block of x. Then, we get following proposition.

Proposition 3.2.1. For any two tapes $x, y \in V(l, m, n)$, if $B(x) \neq B(y)$, then $conf(x) \neq conf(y)$.

Proof of Lemma 3.2.1(continued): There are at most $E(l, m, n)=(l+2)(m+2)(n+2)s2^{L(l, m, n)}t^{2^{L(l,m,n)}}$ different semi-configurations of M just when M enters the last (l, m, n)-block of tapes in $V(l, m, n)$. On the other hand, $|S(l, m, n)|=2r-1$ ($r=(lmn)^{lmn}$).

Thus, from the assumption concerning the function $L(l, m, n)$, it follows that there exists a point (l_i, m_i, n_i) such that $|S(l_i, m_i, n_i)| > E(l_i, m_i, n_i)$. For such (l_i, m_i, n_i), there exist two tapes x, y in $V(l_i, m_i, n_i)$ such that $B(x) \neq B(y)$ and $conf(x) = conf(y)$. This contradicts Proposition 3.2.1. This completes the proof of (2). □

From Lemma 3.1, we can conclude as follows.

Theorem 3.2.1. To simulate 4-DMA_1's,

(1) $SV4$-NTM's require $\Omega(lmn(\log lmn))$ space, and

(2) $SV4$-DTM's require $2^{\Omega(lmn(\log lmn))}$ space.

By using the same technique as in the proof of Theorem 3.2.1, we can get as follows.

Theorem 3.2.2. To simulate 4-NMA_1's,

(1) $SV4$-NTM's require $\Omega(l^2m^2n^2)$ space, and

(2) $SV4$-DTM's require $2^{\Omega(l^2m^2n^2)}$ space.

3.3 Concluding Remarks

In this chapter, we showed the necessary spaces for four-dimensional Turing machines to simulate four-dimensional one-marker automata. It will be interesting to investigate how much space is necessary and sufficient for seven-way four-dimensional deterministic or nondeterministic Turing machines to simulate four-dimensional 'alternating' one-marker automata.

Chapter 4

A Space Lower-Bound Technique for Four-Dimensional Alternating Turing Machines

Alternating Turing machines were introduced in 1981 as a generalization of nondeterministic Turing machines and as a mechanism to model parallel computation. On the other hand, we have no enough techniques which we can show that some concrete four-dimensional language is not accepted by any space-bounded four-dimensional alternating Turing machines. The main purpose of this paper is to present a technique which we can show that some four-dimensional language is not accepted by any space-bounded four-dimensional alternating Turing machines.

Concretely speaking, we show that the set of all four-dimensional input tapes over {0,l}, which each top half part is equal to each bottom half part, is not accepted by any $L(m)$ space-bounded four-dimensional alternating Turing machines for any function $L(m)$ smaller than $\log m$.

4.1 Preliminaries

We have no enough techniques which we can show that some concrete four-dimensional language is not accepted by any space-bounded four-dimensional alternating Turing machines [5, 50]. The main purpose of this paper is to present a technique which we can show that some four-dimensional language is not accepted by any space-bounded four-dimensional alternating Turing machines. Concretely speaking, we show that the set of all four-dimensional input tapes over {0,l}, which each top half part is equal to each bottom half part, is not accepted by any $L(m)$ space-bounded four-dimensional alternating Turing machines for any function $L(m)$ such that $\lim_{m\to\infty}[L(m)/\log m] = 0$. We let each side-length of each four-dimensional input tape of these automata be equivalent in order to increase the theoretical interest.

Let Σ be a finite set of symbols. A four-dimensional tape over Σ is a four-dimensional rectangular array of elements of Σ. The set of all four-dimensional tapes over Σ is denoted by $\Sigma^{(4)}$. Given a tape $x \in \Sigma^{(4)}$, for each integer $j(1 \leq j \leq 4)$, we let $m_j(x)$ be the length of x along the jth axis. The set of all $x \in \Sigma^{(4)}$ with $l_1(x) = m_1$, $l_2(x)=m_2$, $l_3(x)=m_3$, and $l_4(x)=m_4$ denoted by $\Sigma^{(m_1,m_2,m_3,m_4)}$. If $1 \leq i_j \leq l_j(x)$ for each $j(1 \leq j \leq 4)$, let $x(i_1,i_2,i_3,i_4)$ denote the symbol in x with coordinates (i_1,i_2,i_3,i_4).

Furthermore, we define $x[(i_1,i_2,i_3,i_4),(i'_1,i'_2,i'_3,i'_4)]$, when $1 \leq i_j \leq i'_j(x)$ for each integer $j(1 \leq j \leq 4)$, as the four-dimensional tape y satisfying the following (i) and (ii):

(i) for each $j(1 \leq j \leq 4)$, $l_j(y)=i'_j-i_j+1$;

(ii) for each $r_1, r_2, r_3, r_4 (1 \leq r_1 \leq l_1(y), 1 \leq r_2 \leq l_2(y),$

$1 \leq r_3 \leq l_3(y), 1 \leq r_4 \leq l_4(y))$, $y(r_1, r_2, r_3, r_4) = x(r_1+i_1-1, r_2+i_2-1, r_3+i_3-1, r_4+i_4-1)$. (We call $x[(i_1, i_2, i_3, i_4), (i'_1, i'_2, i'_3, i'_4)]$-segment of x.);

A four-dimensional alternating Turing machine (4-*ATM*) M is defined by the 7-tuple $M = (Q, q_0, U, F, \Sigma, \Gamma, \delta)$, where (1) Q is a finite set of states; (2) $q_0 \in Q$ is the initial state; (3) U

$\subseteq Q$ is the set of universal states; (4) $F \subseteq Q$ is the set of accepting states; (5) Σ is a finite input alphabet ($\# \in \Sigma$ is the boundary symbol); (6) Γ is a finite storage-tape alphabet ($B \in \Gamma$ is the boundary symbol), and (7) $\delta \subseteq (Q \times \{\#\}) \times \Gamma) \times (Q \times (\Gamma - \{B\}) \times$ {east, west, south, north, up, down, past, future, no move} $\times$ {right, left, no move}) is the *next-move relation.*

A state q in $Q - U$ is said to be *existential*. The machine M has a read-only four-dimensional input tape with boundary symbols # 's and one semi-infinite storage tape, initially blank. Of course, M has a finite control, an input head, and a storage-tape head. A position is assigned to each cell of the read-only input tape and to each cell of the storage tape. A step of M consists of reading one symbol from each tape, writing a symbol on the storage tape, moving the input and storage heads in specified directions, and entering a new state, in accordance with the next-move relation δ. Note that the machine cannot write the blank symbol. If the input head falls off the input tape, or if the storage head falls off the storage tape (by moving left), then the machine M can make no further move.

A *configuration* of a 4-*ATM* $M = (Q, q_0, U, F, \Sigma, \Gamma, \delta)$ is a pair of an element of $\Sigma^{(4)}$ and an element of $C_M = (\mathrm{N} \cup \{0\})^{(3)} \times S_M$, where $S_M = Q \times (\Gamma\text{-}\{B\})^* \times N$ and N denotes the set of all the positive integers. The first component x of a configuration $c = (x, ((i_1, i_2, i_3, i_4), (q, \alpha, j)))$ represents the input to M. The second component (i_1, i_2, i_3, i_4) of c represents the input-head position. The third component (q, α, j) of c represents the state of the finite control, nonblank contents of the storage tape, and the storage-head position. An element of C_M is called a semi-configuration of M and an element of S_M is called a storage state of M. If q is the state associated with configuration c, then c is said to be a universal (existential, accepting) configuration if q is a universal(existential, accepting) state. The initial configuration of M on input x is $I_M(x) = (x, (1, 1, 1, 1), (q_0, \lambda, 1))$, where λ is the null string.

Given $M = (Q, q_0, U, F, \Sigma, \Gamma, \delta)$, we write $c \vdash_M c'$ and say c' is a successor of c if configuration c' follows from configuration c in one step of M, according to the transition rules δ. The relation $\vdash_M$ is not necessarily single-valued, because δ is not. A computation path of M on x is a sequence. $C_0 \vdash_M C_1 \vdash_M \ldots \vdash_M C_n$ ($n \geq 0$), where $C_0 = I_M(x)$. A computation tree of M is a finite, nonempty labeled tree with the following properties: (1) Each node v of the tree is labeled with a configuration $l(v)$, (2) If v is an internal node (a nonleaf) of the tree, $l(v)$ is universal and $\{c \mid l(v) \vdash_M c\} = \{c_1, \ldots, c_k\}$, then v has exactly k children $v_1, \ldots, v_k$ such that $l(v_i) = c_i$ ($1 \leq i \leq k$), and (3) If v is an internal node of the tree and $l(v)$ is existential, then v has exactly one child u such that $l(v) \vdash_M l(u)$. A computation tree of M on input x is a computation tree of M whose root is labeled with $I_M(x)$. An accepting computation tree of M on x is a computation tree of M on x whose leaves are all labeled with accepting configurations. We say that M accepts x if there is an accepting computation tree of M on input x. Define $T(M) = \{x \in \Sigma^{(4)} \mid M \text{ accepts } x\}$.

In this paper, we shall concentrate on investigating the properties of 4-*ATM*'s whose each side-length of each four-dimensional input tape is equivalent and whose storage tapes are bounded (in length) to use.

Let $L(m) : \mathbf{N} \rightarrow \mathbf{N}$ be a function with one variable m. With each 4-*ATM* M we associate a space complexity function SPACE that takes configurations to natural numbers. That is, for each configuration $c = (x, ((i_1, i_2, i_3, i_4), (q, \alpha, j)))$, let SPACE$(c) = |\alpha|$. We say that M is $L(m)$ space-bounded if for all $m \geq 1$ and for each x with $l_1(x) = l_2(x) = l_3(x) = l_4(x) = m$, if x is accepted by M, then there is an accepting computation tree of M on input x such that for each node v of the tree, SPACE$(l(v)) \leq L(m)$. We denote an $L(m)$ space-bounded 4-*ATM* by 4-*ATM* $(L(m))$. L[4-*ATM* $(L(m))$] $= \{T \mid T = T(M)$ for some 4-*ATM* $(L(m))$ $M\}$.

4.2 Main Result

Theorem 4.2.1 Let $T=\{x \in \{0, 1\}^{(4)} \mid \exists m \leq 1 \, [l_1(x) = l_2(x) = l_3(x) = l_4(x) = 2m \,\&\, x[(l, 1, 1, 1), (2m, 2m, 2m, m)] = x[(l, 1, 1, m + 1), (2m, 2m, 2m, 2m)]]\}$. Then, $T \notin L[4\text{-}ATM\,(L(m))]$ for any $L(m) = o(\log m)$.

(Proof) Suppose that there exists a 4-ATM $(L\,(m))$ M accepting T, where $L(m) = o(\log m)$. We assume, without loss of generality, that M moves a storage-tape head after changing its state and writing a new symbol on the storage tape, and moves an input head finally.

For each $m \leq 1$, let $V(m) = \{x \in T \mid l_1(x) = l_2(x) = l_3(x) = l_4(x) = 2m\}$. For each x in $V(m)$, let $t(x)$ be one fixed accepting computation tree of M on x such that each node v of the tree satisfies SPACE$(l(v)) \leq L(2m)$, where for each node v of $t(x)$, $l(v)$ represents the label of v. Without loss of generality, we assume that for any $t(x)$; (i) any two different nodes on any path of $t(x)$ are labeled by different configurations, and, (ii) if any different nodes of $t(x)$ have the same label, then the subtrees [of $t(x)$] with these nodes as the roots are identical.

For each x in $V(m)$, let $t(m)$, which we call the reduced accepting computation tree of M on x, be a tree obtained from $t(x)$ by the following procedure [for each node v of $t(x)$, we denote by $d(v)$ the length of the path from the root of $t(x)$ to v (i.e., the number of edges from the root of $t(x)$ to v)]:

Begin

1. $T_r = t(x)$

2. $i = 1$

3. Let $N(i) \triangleq \{v \mid v$ is node of T_r and $d(v) \leq i\}$. Divide $N(i)$ as follows: $N(i) = P(l) \cup P(2) \cup \ldots \cup P(j_i)$, where: (1) if $i_a = i_b (l \leq i_a, i_b \leq j_i)$, then $P(i_a) \cap P(i_b) = \varphi$, and (2) for each $i_a (1 \leq i_a \leq j_i)$ and for each v_a, $V_b \in P(i_a)$, $l(v_a) = l(v_b)$ (i.e., the labels of v_a and v_b are identical). For each i_a (1 ≤

$i_a \leq j_i$), let dis(i_a) = min{$d(v) \mid v \in P(i_a)$} and let $n(i_a)$ be the leftmost node among those nodes v in $P(i_a)$ such that $d(v)$ = dis(i_a). Further, let $N'(i) = N(i)$ - {$n(1), n(2), \ldots, n(j_i)$}. By removing from T_r all the subtrees whose roots are included in $N'(i)$, we make the new T_r. 4. If the height of T_r (i.e., the length of the longest path of T_r) is less than or equal to i, then we let $t'(x) = T_r$. Otherwise, we let $i = i + 1$ and go to step 3.

end

[Example 1] Let $x \in V(m)$ and $t(x)$ be a tree. Here, suppose that nodes A and D have the same label, nodes B and C have the same label, and other nodes each have different labels. [From the preceding assumption (ii) concerning $t(x)$, identical.] Then, $t'(x)$ is a tree. That is, $t'(x)$ is obtained from $t(x)$ by moving the subtree with nodes C and D as the roots from $t(x)$.

It is easily seen that for each x in $V(m)$, all the nodes of $t'(x)$ have labels different from one another, and the set of all the paths from root of $t'(x)$ to the leaves of $t'(x)$ represents necessary and sufficient accepting computations of M on x. From $t'(x)$, we now define an extended crossing sequence (ECS) at the boundary between the top and bottom halves of x. The concept of ECS was first introduced in [84]. We relabel each node v of $t'(x)$, as follows. (We denote this new labeling by l'.) For each node v of t'(x), let $f(v)$ denote the father node of v. Then, for each node v of $t'(x)$, where $x \in V(m)$, let if, for some storage states (q, α, j) and (q', α', j'),

(i) $l(f(v)) = (x, (i_1, i_2, i_3, m), (q', \alpha', j'))$ and $l(v) = (x, (i_1, i_2, i_3, m+1), (q, \alpha, j))$, or

(ii) $l(f(v)) = (x, (i_1, i_2, i_3, m+1), (q', \alpha', j'))$ and $l(v) = (x, (i_1, i_2, i_3, m), (q, \alpha, j))$, then $l'(v) = ((i_1, i_2, i_3), (q, \alpha, j))$

else

$l'(v) = *$.

That is, if the movement of M from $f(v)$ to v represents the action of crossing the boundary between the top and bottom halves of x, then v is newly labeled by $(i_1, i_2, i_3), (q, \alpha, j))$, where

(q, α, j) is the storage state component of $l(v)$. Otherwise, v is newly labeled by *. From the newly labeled $t'(x)$, we extract those nodes v such that $l'(r_n) = *$, and by using these nodes, we construct a tree t''(x) satisfying the following condition:

(A) For any node v of $t''(x)$, nodes $v_1, v_2, \ldots, v_S$ are children of v if and only if $v_1, v_2, \ldots, v_S$ are descendants of v in $t'(x)$ and $l'(u) = *$ for each node u on the path rom v to each v_i. In general, there can be two or more uth trees $t''(x)$. Let these trees be $t_1''(x), \ldots, t_n''(x)$. For each node v of each $t_i''(x)$ $(1 \leq i \leq n)$, we now define an element of *ECS* (*EECS*) inductively as follows: Let $l'(v) = ((i_1, i_2, i_3), (q, \alpha, j))$.

(1) If v is a leaf, then $[((i_1, i_2, i_3), (q, \alpha, j))]$ is an EECS of v.

(2) If v has only one child v_1 and $Q_1 = [((i_{11}, i_{21}, i_{31}), (q_1, \alpha_1, j_1))P]$ is an *EECS* of v_1, then $[((i_1, i_2, i_3), (q, \alpha, j))\ ((i_{11}, i_{21}, i_{31}), (q_1, \alpha_1, j_1))P]$ is an *EECS* of v.

(3) If v has $d(\geq 2)$ children $v_1, \ldots, v_d$ and $Q_1, \ldots, Q_d$ are *EECS*'s of $v_1, \ldots, v_d$, respectively, then $[((i_1, i_2, i_3), (q, \alpha, j))\ Q_\sigma(1)\ldots Q_\sigma(d)]$ is an *EECS* of v for any permutation $\sigma : \{1, \ldots, d\} \rightarrow \{1, \ldots, d\}$.

(4) An *EECS* of v is defined only by the preceding statements (1), (2), and (3).

Now, let $Q_1, \ldots, Q_n$ be *EECS*'s of the root nodes of $t_1''(x), \ldots, t_n''(x)$, respectively. Then, for any permutation $\sigma : \{1, \ldots, n\} \rightarrow \{1, \ldots, n\}$, we call $Q_\sigma(1), \ldots, Q_\sigma(n)$ an *ECS* of x. As is easily seen from the definitions, there can be two or more *EECS*'s of each node v of each $t''(x)$, and there can be two or more *ECS*'s of x. Let Q_1 and Q_2 be any two *EECS*'s. If the following condition (B) is satisfied, we say Q_1 and Q_2 are equivalent and write $Q_1 \equiv Q_2$:

(B) Let $Q_1 = [((i_{11}, i_{21}, i_{31}), (q_1, \alpha_1, j_1)) \ldots ((i_{1n}, i_{2n}, i_{3n}), (q_n, \alpha_n, j_n))\ P_1 \ldots P_S]$, $Q_2 = [((i'_{11}, i'_{21}, i'_{31}), (q'_1, \alpha'_1, j'_1)) \ldots ((i'_{1n'}, i'_{2n'}, i'_{3n'}), (q'_{n'}, \alpha'_{n'}, j'_{n'}))\ P'_1 \ldots P'_{S'}]$. Then $n = n'$, $s = s'$, and $((i_{1k}, i_{2k}, i_{3k}), (q_k, \alpha_k, j_k)) = ((i'_{1k}, i'_{2k}, i'_{3k}), (q'_k, \alpha'_k, j'_k))$ for each k $(1 \leq k \leq n)$, and there exists a permutation $\sigma : \{1, \ldots, s\} \rightarrow \{1, \ldots, s\}$ such that $P_i \equiv P'_{\sigma(i)}$ for each $i (1 \leq i \leq s)$, where $n, s \geq 0$, and $((i_1, i_2, i_3), (q, \alpha, j))$'s and $((i'_1, i'_2, i'_3), (q', \alpha', j'))$'s are pairs (coordinates along the fourth axis, storage state), and further P, P' are *EECS*'s.

Let $Q = Q_1 \ldots Q_n$, $Q' = Q'_1 \ldots Q'_n$ be any two *ECS*'s. We say that Q and Q' are equivalent if $n = n'$ and there exists a permutation $\sigma : \{1, \ldots, n\} \rightarrow \{1, \ldots, n\}$ such that $Q_i \equiv Q'_{\sigma(i)}$ for each i ($1 \leq i \leq n$). [As is easily seen from the definition, any two *ECS*'s of x are equivalent for any x in $V(m)$.] For any *ECS* Q, the length of Q is the number of pairs (coordinates along the fourth axis, storage state) in Q, and is denoted by $|Q|$. For each $m \geq 1$, let $E(m) = \{Q \mid Q$ is an *ECS* of x for some x in $V(m)\}$. Then, the following two propositions must hold :

Proposition 4.2.1 $|E(m)| = Z(m)^{dZ(m)}$, where $Z(m) = (2m + 2)3rL(2m)s^{L(2m)}$, r and s are the numbers of states (of the finite control) and storage-tape symbols of M, and d
is a positive constant.

Proposition 4.2.2 Let x and y be any two different tapes in $V(m)$, and let Q_x and Q_y be any *ECS*'s of x and y, respectively. Then, Q_x and Q_y are not equivalent.

Clearly, $|V(m)| = 2^{8t}$ $(t=m^4)$. Because $L(m) = o(\log m)$, it follows from Proposition 1 that $|V(m)| > |E(m)|$ for large m. For such a large m, there must exist two different tapes $x, y \in V(m)$ such that some *ECS* of x and some *ECS* of y are equivalent, which contradicts Proposition 2. This completes the proof. □

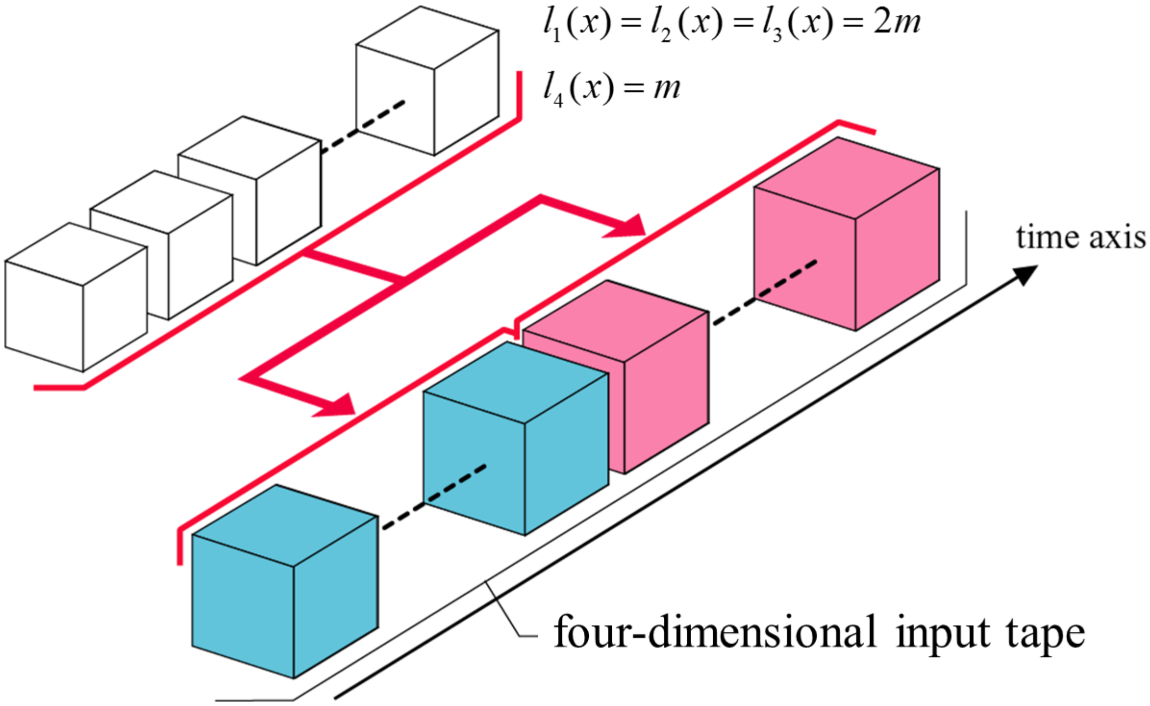

Fig. 4.1 : Four-dimensional input tape which we use in the main theorem

4.3 Concluding Remarks

In this chapter, we presented a technique which we can show that a four-dimensional language is not accepted by any space-bounded alternating Turing machines. It will be interesting to investigate infinite space hierarchy properties of the classes of sets accepted by 4-*ATM*'s with spaces of size smaller than log m.

Chapter 5

Four-Dimensional Homogeneous Systolic Pyramid Automata

Cellular automaton is famous as a kind of the parallel automaton. Cellular automata were investigated not only in the viewpoint of formal language theory, but also in the viewpoint of pattern recognition. Cellular automata can be classified into some types. A systolic pyramid automata is also one parallel model of various cellular automata. A homogeneous systolic pyramid automaton with four-dimensional layers (4-*HSPA*) is a pyramid stack of four-dimensional arrays of cells in which the bottom four-dimensional layer (level 0) has size an (a≥1), the next lowest 4(a-1), and so forth, the (a-1)st four-dimensional layer (level (a-1)) consisting of a single cell, called the root. Each cell means an identical finite-state machine. The input is accepted if and only if the root cell ever enters an accepting state. A 4-*HSPA* is said to be a real-time 4-*HSPA* if for every four-dimensional tape of size 4a (a≥1) it accepts the four-dimensional tape in time a-1. Moreover, a one-way four-dimensional cellular automaton (1-4*CA*) can be considered as a natural extension of the one-way two-dimensional cellular automaton to four-dimension. The initial configuration is accepted if the last special cell reaches a final state. A 1-4*CA* is said to be a real- time 1-4*CA* if when started with four-dimensional array of cells in nonquiescent state, the special cell reaches a final state. In this paper, we propose a homogeneous systolic automaton with four-dimensional layers (4-*HSPA*), and investigate some properties of real-time 4-*HSPA*. Specifically, we first investigate a relationship between the accepting powers of real-time 4-*HSPA*'s and real-time 1-4*CA*'s. We next show the recognizability of four-dimensional connected tapes by real-time 4-*HSPA*'s.

5.1 Introduction

In recent years, due to the advances in many application areas such as dynamic image processing, computer animation, VR(virtual reality), AR (augmented reality), and so on, the study of four-dimensional pattern processing has been of crucial importance. And the question of whether processing four-dimensional digital patterns is much more difficult than three-dimensional ones is of great interest from the theoretical and practical standpoints. Thus, the study of four-dimensional automata as a computational model of four-dimensional pattern processing has been meaningful[34, 55-71, 80, 82]. On the other hand, cellular automata were investigated not only in the viewpoint of formal language theory, but also in the viewpoint of pattern recognition. Cellular automata can be classified into some types [31]. A systolic pyramid automaton is also one parallel model of various cellular automata. In this chapter, we propose a homogeneous systolic pyramid automaton with four-dimensional layers (4-*HSPA*) as a new four-dimensional parallel computational model, and investigate some properties of real-time 4-*HSPA*.

5.2 Preliminaries

Let Σ be a finite set of symbols. A four-dimensional tape over Σ is a four-dimensional rectangular array of elements of Σ. The set of all four-dimensional tapes over $\sum$ is denoted by

$\Sigma^{(4)}$. Given a tape $x \in \Sigma^{(4)}$, for each integer j ($1 \leq j \leq 4$), we let $l_j(x)$ be the length of x along the jth axis. The set of all $x \in \Sigma^{(4)}$ with $l_1(x) = n_1$, $l_2(x) = n_2$, $l_3(x) = n_3$, and $l_4(x) = n_4$ is denoted by $\Sigma^{(n_1,n_2,n_3,n_4)}$. When $1 \leq i_j \leq l_j(x)$ for each j($1 \leq j \leq 4$), let $x(i_1, i_2, i_3, i_4)$ denote the symbol in x with coordinates (i_1, i_2, i_3, i_4). Furthermore, we define

$x\,[(i_1, i_2, i_3, i_4), (i'_1, i'_2, i'_3, i'_4)]$,

when $1 \leq i_j \leq i'_j \leq l_j(x)$ for each integer j($1 \leq j \leq 4$), as the four-dimensional input tape y satisfying the following conditions :

(i)for each j($1 \leq j \leq 4$), $l_j(y) = i_j - i_j + 1$;

(ii)for each r_1, r_2, r_3, r_4 ($1 \leq r_1 \leq l_1(y)$, $1 \leq r_2 \leq l_2(y)$, $1 \leq r_3 \leq l_3(y)$, $1 \leq r_4 \leq l_4(y)$), $y(r_1, r_2, r_3, r_4) = x(r_1 + i_1 - 1, r_2 + i_2 - 1, r_3 + i_3 - 1, r_4 + i_4 - 1)$. (We call $x[(i_1, i_2, i_3, i_4), (i'_1, i'_2, i'_3, i'_4)]$ the $[(i_1, i_2, i_3, i_4), (i'_1, i'_2, i'_3, i'_4)]$-segment of x.)

For each $x \in \Sigma^{(n_1,n_2,n_3,n_4)}$ and for each $1 \leq i_1 \leq n_1$, $1 \leq i_2 \leq n_2$, $1 \leq i_3 \leq n_3$, $1 \leq i_4 \leq n_4$, $x[(i_1, 1, 1, 1), (i_1, n_2, n_3, n_4)$, $x[(1, i_2, 1, 1), (n_1, n_2, n_3, n_4)]$, and $x[(1, 1, i_3, 1), (n_1, n_2, n_3, n_4)]$ are called the i_1th (2-3) plane, the i_2th (1-3) plane, and the i_3th (1-2) plane of each time of x, and are denoted by $x[i_1, *, *, *]$, $x[*, i_2, *, *]$, and $x[*, *, i_3, *]$, respectively. $x[*, *, *, i_4]$ also has analogous meaning.

A four-dimensional homogeneous systolic pyramid automaton (4-*HSPA*) is a pyramidal stack of four-dimensional arrays of cells in which the bottom four-dimensional layer (level 0) has size $4a$ ($a \geq 1$), the next lowest $4(a - 1)$, and so forth, the $(a-1)$st four-dimensional layer (level ($a-1$)) consisting of a single cell, called the root . Each cell means an identical finite-state machine, $M = (Q, \Sigma, \delta, \#, F)$, where Q is a finite set of states, $\Sigma \subseteq Q$ is a finite set of input states, $\# \in Q-\Sigma$ is the *quiescent state*, $F \subseteq Q$ is the set of *accepting states*, and δ: $Q^{17} \rightarrow Q$ is the *state transition function*, mapping the current states of M and its 16 son cells in a $2 \times 2 \times 2 \times 2$ block on the four-dimensional layer below into M's next state. The input is accepted if and only if the root cell ever enters an accepting state. A 4-*HSPA* is said to be a real-time 4-*HSPA* if for every four-dimensional tape of size $4a$ ($a \geq 1$) it accepts the four-dimensional tape in time $a-1$. By $£^R$[4-*HSPA*] we denote the class of the sets of all the four-dimensional tapes accepted by a real-time 4-*HSPA*[6].

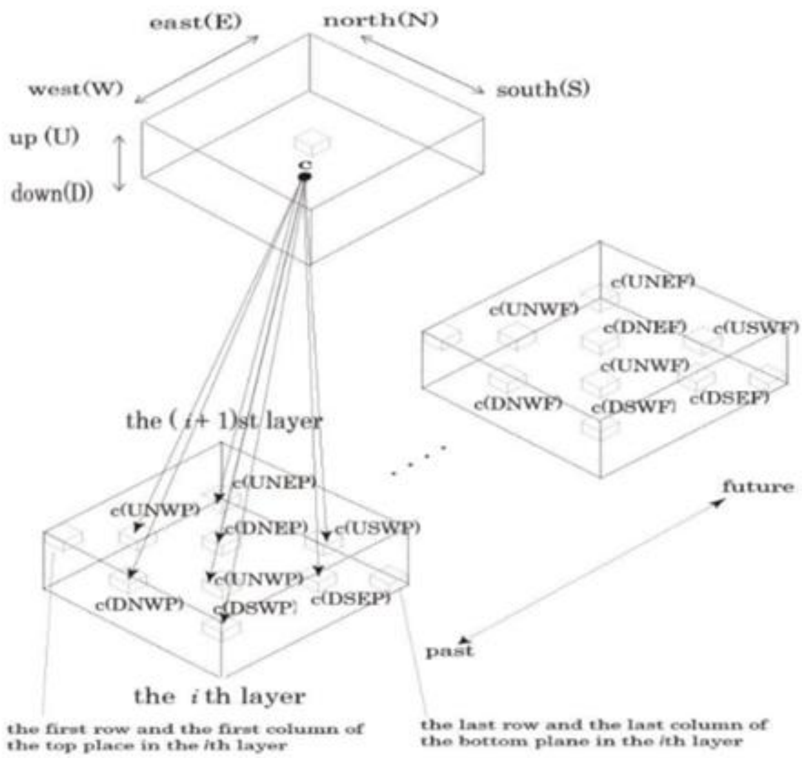

Fig.5.1 : Four-dimensional homogeneous systolic pyramid automaton.

A one-way four-dimensional cellular automaton (1-4CA) can be considered as a natural extension of the one-way two-dimensional cellular automaton to four dimensions [31]. The initial configuration of the cellular automaton is taken to be an $l_1(x) \times l_2(x) \times l_3(x) \times l_4(x)$ array of cells in the nonquiescent state. The initial configuration is accepted if the last special cell reaches a final state.

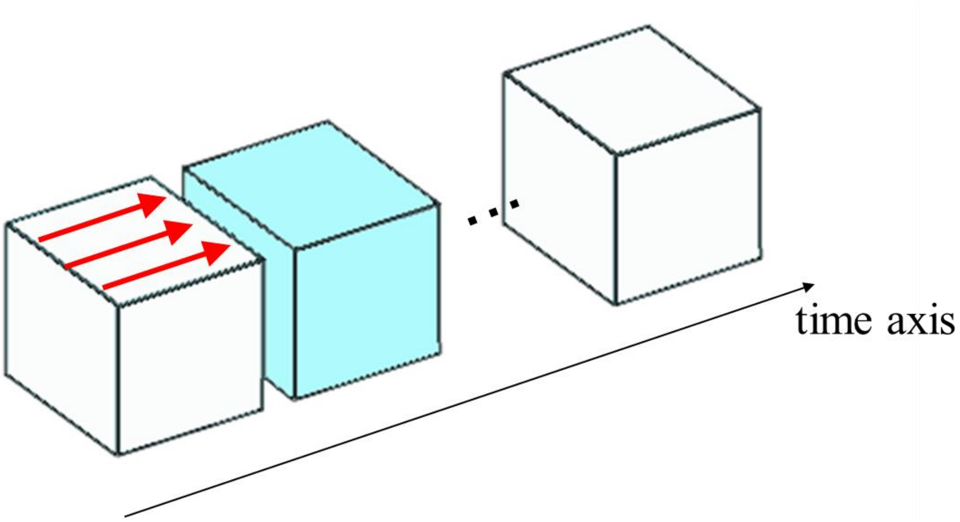

Fig. 5.2 : One-way four-dimensional cellular automaton

A 1-4*CA* is said to be a real-time 1-4CA if when started with an $l_1(x) \times l_2(x) \times l_3(x) \times l_4(x)$ array of cells in the nonquiescent state, the special cell reaches a final state in time $l_1(x) + l_2(x) + l_3(x) \times l_4(x) - 1$. By $£_R$[1-4*CA*] we denote the class of the sets of all the four-dimensional tapes accepted by a real-time 1-4*CA* [31].

5.3 Results

We mainly investigate a relationship between the accepting powers of real-time 4-*HSPA*'s and real-time 1-4*CA*'s. The following theorem implies that real-time 4-*HSPA*'s are less powerful than real-time 1-4*CA*'s.

Theorem 5.3.1. $£^R[4\text{-}HSPA] \subsetneq £^R[1\text{-}4CA]$ if input tape V is as follows.

$$V = \{ x \in \{0,1\}^{(4)} \mid l_1(x) = l_2(x) = l_3(x)= l_4(x) \ \& \ \forall i_1, \forall i_2, \forall i_3, \forall i_4 \ (1 \le i_1 \le l_1(x),\ 1 \le i_2 \le l_2(x),\ 1 \le i_3 \le l_3(x),\ 1 \le i_4 \le l_4(x))[x(i_1, i_2, i_3, 1) = x(i_1, i_2, i_3, l_4(x))]\}.$$

Proof : It is easily shown that $V \in £^R[1\text{-}4CA]$. Below, we show that $V \notin £^R[4\text{-}HSPA]$. Suppose that there exists a real-time 4-*HSPA* accepting V. For each $t \ge 4$, let

$$W(n) = \{ x \in \{0,1\}^{(4)} \mid l_1(x) = l_2(x) = l_3(x) = l_4(x) \ \& \ x[(1, 1, 1, 2), (t, t, t, t\text{-}1)] \in \{0\}^{(4)}\}.$$

Sixteen sons of the root cell $A(t-1,1,1,1,1)$ of M $A_{(t-2,1,1,2,1)}$, $A_{(t-2,1,2,2,1)}$, $A_{(t-2,2,1,2,1)}$, $A_{(t-2,2,2,2,1)}$, $A_{(t-2,1,1,3,1)}$, $A_{(t-2,1,2,3,1)}$, $A_{(t-2,2,1,3,1)}$, $A_{(t-2,2,2,3,1)}$, $A_{(t-2,1,1,2,2)}$, $A_{(t-2,1,2,2,2)}$, $A_{(t-2,2,1,2,2)}$, $A_{(t-2,2,2,2,2)}$, $A_{(t-2,1,1,3,2)}$, $A_{(t-2,1,2,3,2)}$, $A_{(t-2,2,1,3,2)}$, $A_{(t-2,2,2,3,2)}$ are denoted by $C^{(UNWP)}$, $C^{(USWP)}$, $C^{(USEP)}$,$C^{(UNEP)}$, $C^{(DNWP)}$, $C^{(DSWP)}$, $C^{(DSEP)}$, $C^{(DNEP)}$, $C^{(UNWF)}$, $C^{(USWF)}$, $C^{(USEF)}$,$C^{(UNEF)}$, $C^{(DNWF)}$, $C^{(DSWF)}$, $C^{(DSEF)}$, $C^{(DNEF)}$, respectively. For each x in $W(n)$, $x(UNWP)$, $x(USWP)$, $x(USEP)$, $x(UNEP)$, $x(DNWP)$, $x(\mathrm{USWP})$, $x(USEP)$, $x(UNEP)$, $x(UNWF)$, $x(USWF)$, $x(USEF)$, $x(UNEF)$, $x(DNWF)$, $x(\mathrm{USWF})$, $x(USEF)$, $x(UNEF)$ are the states of $C^{(UNWP)}$, $C^{(USWP)}$, $C^{(USEP)}$,$C^{(UNEP)}$, $C^{(DNWP)}$, $C^{(DSWP)}$, $C^{(DSEP)}$, $C^{(DNEP)}$, $C^{(UNWF)}$, $C^{(USWF)}$, $C^{(USEF)}$,$C^{(UNEF)}$, $C^{(DNWF)}$, $C^{(DSWF)}$, $C^{(DSEF)}$, $C^{(DNEF)}$, at time t-2, respectively. Let $\sigma(x)$ = ($x(UNWP)$, $x(USWP)$, $x(USEP)$, $x(UNEP)$, $x(DNWP)$, $x(\mathrm{USWP})$, $x(USEP)$, $x(UNEP)$), $\gamma(x)$=($x(UNWF)$, $x(USWF)$, $x(USEF)$, $x(UNEF)$, $x(DNWF)$, $x(\mathrm{USWF})$, $x(USEF)$, $x(UNEF)$), and $\rho(x)$ = ($x(UNWP)$, $x(USWP)$, $x(USEP)$, $x(UNEP)$, $x(DNWP)$, $x(\mathrm{USWP})$, $x(USEP)$, $x(UNEP)$, $x(UNWF)$, $x(USWF)$, $x(USEF)$, $x(UNEF)$, $x(DNWF)$, $x(\mathrm{USWF})$, $x(USEF)$, $x(UNEF)$). Then, the following two propositions must hold.

Proposition 5.3.1. (i) *For any two tapes* $x, y \in W(n)$ *whose* 1*st cubes are same*, $\sigma(x) = \sigma(y)$.

(ii) *For any two tapes* $x, y \in W(n)$ *whose nth cubes are same*, $\gamma(x) = \gamma(y)$.

[**Proof :** From the mechanism of each cell, it is easily seen that the states of $C^{(UNWP)}$, $C^{(USWP)}$, $C^{(USEP)}$,$C^{(UNEP)}$, $C^{(DNWP)}$, $C^{(DSWP)}$, $C^{(DSEP)}$, $C^{(DNEP)}$ are not influenced by the information of 1st cube. From this fact, we have (i). The proof of (ii) is the same as that of (i). □]

Proposition 5.3.2. *For any two tapes* $x, y \in W(t)$ *whose* 1*st cube are different*, $\sigma(x) \neq \sigma(y)$.

[**Proof :** Suppose to the contrary that $\sigma(x) = \sigma(y)$. We consider two tapes $x', y' \in W(t)$ satisfying the following :

(i) 1st cube of x and nth cube of x are equal to 1st cube of x', respectively

(ii) 1st cube of y' is equal to 1st cube of y, and nth cube of y' is equal to 1st cube of x.

As is easily seen, $x' \in V$ and so x' is accepted by M. On the other hand, from Proposition 2.1(ii), $\gamma(x') = \gamma(y')$. From Proposition 2.1(i), $\sigma(x) = \sigma(x')$, $\sigma(y) = \sigma(y')$. It follows that y' must be also accepted by M. This contradicts the fact that y' is not in V . □]

Proof of Theorem 5.3.1 (*continued*) : Let $p(t)$ be the number of tapes in $W(t)$ whose 1st cubes are different, and let $Q(t) = \{ \sigma(x) \mid x \in W(t)\}$, where k is the number of states of each cell of M. Then, $p(t) = 2^{t^2}$, and $Q(t) \leq k^4$. It follows that $p(n) > Q(t)$ for large t. Therefore, it follows that for large t, there must be two tapes x, y in $W(t)$ such that their 1st cubes are different and $\sigma(x) = \sigma(y)$. This contradicts Proposition 2.2, so we can conclude that $V \notin £^R$[4-*HSPA*]. In the case of four-dimension, we can show that $V \notin £^R$[4-*HSPA*] by using the same technique. This completes the proof of Theorem 2.1. □

We next show the recognizability of four-dimensional connected tapes by real-time 4-*HSPA*'s by using the same technique of Ref.[31]. Let x in $\{0,1\}^{(4)}$. A maximal subset P of $\boldsymbol{N}^4$ satisfying the following conditions is called a 1-*component* of x.

(i)For any $(i_1, i_2, i_3, i_4 \in P$, we have $1 \leq i_1 \leq l_1(x)$, $1 \leq i_2 \leq l_2(x)$, $1 \leq i_3 \leq l_3(x)$, $1 \leq i_4 \leq l_4(x)$, and $x(i_1, i_2, i_3, i_4) = 1$.

(ii) For any (i_1, i_2, i_3, i_4), $(i_1', i_2', i_3', i_4') \in P$, there exists a sequence $(i_{1},0,i_{2},0,i_{3},0,i_{4},0)$,$(i_{1},1,i_{2},1, i_{3},1,i_{4},1)$, . . . , $(i_{1},4,i_{2},4,i_{3},4,i_{4},4)$ of elements in P such that $(i_{1},0,i_{2},0,i_{3},0,i_{4},0) = (i_1,i_2,i_3,i_4)$, $(i_{1},4,i_{2},4, i_{3},4, i_{4}, 4) = (i'_1, i'_2, i'_3, i'_4)$, and $|i_1, j - i_1, j{-}1| + |i_2, j - i_2, j{-}1| + |i_3, j - i_3, j{-}1| + |i_4, j - i_4, j{-}1| \leq 1 (1 \leq j \leq 4)$. A tape $x \in \{0, 1\}^{(4)}$ is called *connected* if there exists exactly one 1-component of x.

Let Tc be the set of all the four-dimensional connected tapes. Then, we have

Theorem 5.3.2. $Tc \notin £^R[4\text{-}HSPA]$.

5.4 Concluding Remarks

The technique of AR (augmented reality) or VR(virtual reality) progresses like the Pokemon GO and the Virtual Cinema in the world. The virtual technique will spread steadily among our societies in future. Thus, we think that the study of four-dimensional automata is very meaningful as a computational model of four-dimensional pattern processing. In this paper, we proposed a homogeneous systolic pyramid automaton with four-dimensional layers, and investigated a relationship between the accepting powers of homogeneous systolic pyramid automaton with four-dimensional layers(4-*HSPA*) and one-way four-dimensional cellular automata (1-4*CA*) in real time, and showed that real-time 4-*HSPA*'s are less powerful than real time 1-4*CA*'s.

It will be interesting to investigate about an alternating version or synchronized alternating version of homogeneous systolic pyramid automaton with four-dimensional layers. Moreover, we think that there are many interesting open problems for digital geometry. Among them, the problem of connectedness, especially topological component is one of the most interesting topics[67].

Finally, we would like to hope that some unsolved problems concerning this paper will be explicated in the near future.

Chapter 6

Conclusion

In recent years, due to the advances in computer animation, dynamic image processing, and so forth, the study of four-dimensional information processing has been very important. For example, in the Internet environment, new protocols have been proposed for virtual reality communication on the WWW. In the medical field, we can easily get the precise three-dimensional volumetric images along the axis of time about a human body by excellent equipments such as X-ray CT scanner and MRT scanner. Thus, the study of four-dimensional automata has meaningful as the computational model of four-dimensional information processing.

In this dissertation, we have mainly studied several properties of four-dimensional alternating automata and four-dimensional parallel Turing machines, and have also studied about recognizability of topological components in three-dimensional input tapes. We believe that it is useful for analyzing four-dimensional images to explicate the properties of four-dimensional automata.

In Chapter 3, we investigated the accepting powers of necessary space for seven-way four-dimensional Turing machines to simulate four-dimensional one marker automata. We first define four-dimensional deterministic or nondeterministic one-marker automata and second, investigated necessary space for seven-way Turing machines to one-marker automata. In results, we proved seven-way four-dimensional nondeterministic Turing machine require $\Omega(lmn\log lmn)$ space, and in case of deterministic, require $2^{\Omega(lmn\log lmn)}$ space to simulate four-dimensional one-marker automata.

In Chapter 4, we showed a space lower-bound technique for four-dimensional alternating Turing machines. First, we introduced space-bounded four-dimensional alternating Turing machines. Second, we proved that when each side-length of input tape is equivalent and we limited space of storage tape to $\log m$, four-dimensional alternating Turing machine cannot accept this input tape.

In Chapter 5, we investigated the recognizability of four-dimensional homogeneous systolic pyramid automata. First, we introduced preliminaries for four-dimensional homogeneous systolic pyramid automata and one-way four-dimensional cellular automaton. Second, we investigated a relationship between those automata.

Finally, we would like to hope that some unsolved questions concerning this dissertation will be explicated in the near future. Moreover, theory of four-dimensional automata has a lot of potential because this study is fundamental study. We would like to continue, sophisticate, and apply this study to another theoretical study and practical study. In the case of theoretical study, we can apply four-dimensional automata theory to various mathematical feature such as topological and differential features. On the other hand, in the case of practical study, we can apply four-dimensional automata theory to the high-dimensional information processing such as VR, AR, CV (computer vision) and so on. For example, by taking aim at complexity in programming of VR, AR or CV using four-dimensional automata theory, we can make it easy to create a computer program in VR, AR or CV, because we can understand how computer move by using four-dimensional automata theory.

Bibliography

[1] Aho A., Hopcroft J. E., and Ullman J. D., *The design and analysis of computer algorithms*; Addison-Wesley, Reading, Mass., 1974.

[2] Beyer T., Recognition of topological invariants by interactive arrays, *Ph. D. Thesis, MIT*, 1970.

[3] Blum M. and Hewitt C., Automata on a two-dimensional tape, *Proceedings of IEEE Symposium on Switching and Automata Theory*, pp.155-160, 1967.

[4] Blum M. and Sakoda W. J., On the capability of finite automata in 2 and 3 dimensional space, *Proceedings of the 18th Annual Symposium on Foundations of Computer Science*, pp.147-161, 1977.

[5] Chandra A. K., Kozen D. C., and Stockmeyer L. J., Alternation, *Journal of the ACM*, Vol.28, No.1, pp.114-133, 1981.

[6] Culik K. II and Gruska J. Salomaa, Sytolic trellis automata, Part I, *International Journal of Computer Mathmatics*, 55, pp.99-121, 1991.

[7] Deyer C. R. and Rosenfeld A., Parallel image processing by memory-augmented cellular automata; *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.PAMI-3, No.1, pp.29-41, 1981.

[8] Dietz P. and Kosaraju S. R., Recognition of topological equivalence of patterns by array automata, *Journal of Computer and System Sciences*, Vol.2, pp.111-116, 1980.

[9] Gurari E. M. and Ibarra O. H., (Semi-) alternating stack automata, *Mathematical Systems Theory*, Vol.15, pp.211-224, 1982.

[10] Hopcroft J. E. and Ullman J. D., Some results on tape-bounded Turing machines, *Journal of the ACM*, Vol.16, pp.168-177, 1969.

[11] Hopcroft J. E. and Ullman J. D., Some results on tape-bounded Turing machines, *Journal of the ACM*, Vol.19, No.2, pp.283-295, 1972.

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

[13] Hromkovič J., On the power of alternation in automata theory, *Journal of Computer and System Sciences*. Vol.31, No.1, pp.28-39, 1985.

[14] Ibarra O. H. and Melson R. T., Some results concerning automata on two-dimensional tapes, *International Journal of Computer Mathematics*, Vol.4, No.A, pp.269-279, 1974.

[15] Inoue K. and Nakamura A., Some notes on *n*-dimensional on-line tessellation acceptors, *Proceedings of the International Symposium on Uniformly Structured Automata and Logic*, pp.103-112, 1975.

[16] Inoue K. and Nakamura A., An *n*-dimensional on-line tessellation accepter (in Japanese), *The Transactions of the IECE,* Vol.J59-D, No.4, pp.229-236, 1976.

[17] Inoue K., Investigations of two-dimensional on-line tessellation acceptors (in Japanese), *Ph.D. Thesis, Nagoya University*, 1977.

[18] Inoue K. and Nakamura A., Some properties of two-dimensional nondeterministic finite automata and parallel sequential array acceptors (in Japanese), *The Transactions*

of the IECE, Vol.J60-D, No.11, pp.990-997, 1977.

[19] Inoue K. and Nakamura A., Some properties of two-dimensional on-line tessellation accepters, *Information Sciences*, Vol.13, pp.95-121, 1977.

[20] Inoue K. and Takanami I., Some properties of three-way two-dimensional multicounter automata over square tapes (in Japanese), *The Transactions of the IECE*, Vol.J62-D, No.10, pp.673-680, 1979.

[21] Inoue K. and Takanami I., A note on bottom-up pyramid acceptors, *Information Processing Letters*, Vol.8, No.1, pp.34-37, 1979.

[22] Inoue K., Takanami I., and Taniguchi H., The accepting powers of two-dimensional automata over square tapes, *The Transactions of the IECE*, Vol.J63-D, No.2, pp.113-120, 1980.

[23] Inoue K. and Takanami I., A survey of two-dimensional automata theory, *Information Sciences*, Vol.55, pp.99-121, 1991.

[24] Ito A., Inoue K., Takanami I., and Taniguchi H., Two-dimensional alternating Turing machines with only universal states, *Information and Control*, Vol.55, No.1-3, pp.193-221, 1982.

[25] Ito A., Inoue K., and Takanami I., A note on three-way two-dimensional alternating Turing machines, *Information Sciences*, Vol.45, pp.1-22, 1988.

[26] Ito A., Inoue K., and Takanami I., Complexity of acceptance problems for two-dimensional automata, *a perspective in Theoretical Computer Science, Series in Computer Science*, Vol.16, pp.70-94, 1989.

[27] Ito T., Sakamoto M., Saito M., Iihoshi K., Furutani H., Kono M., and Inoue K., Three-dimensional synchronized alternating Turing machines, *Proceedings of the 11th International Symposium on Artificial Life and Robotics, Oita, Japan*, OS 9-4, 2006.

[28] Ito T., Sakamoto M., Tomozoe N., Iihoshi K., Furutani H., Kono M., Tamaki T., and Inoue K., Some properties of three-dimensional synchronized alternating Turing machines, *Proceedings of the 10th WSEAS International Conference on COMPUTERS, Athens, Greece*, pp.665-670, 2006.

[29] Ito T., Sakamoto M., Tomozoe N., Iihoshi K., Furutani H., Kono M., Tamaki T., and Inoue K., Two topics concerning three-dimensional synchronized alternating Turing machines, *WSEAS Transactions on Computers*, Issue 10, Vol.5, pp.2149-2155, 2006.

[30] Ito T., Sakamoto M., Nagamizu Y., Iihoshi K., Tomozoe N., Furutani H., Kono M., Ikeda S., Tamaki T., and Inoue K., Three-dimensional parallel Turing machines, *Proceedings of the 12th International Symposium on Artificial Life and Robotics, Oita, Japan*, pp.131-134, 2007.

[31] Iwamoto C., Inoue K., and Takanami I., Some Properties of Homogeneous Pyramid Automata, *The IEICE Transactions on Information and Systems (Japanese Edition)*, J73-D-I(9), pp.778-780, 1990.

[32] King K. N., Measures of parallelism in alternating computation trees, *Proceedings of the 13th ACM Symposium on Theory of Computing*, pp.189-201, 1981.

[33] King K. N., Alternating multihead finite automata, *Theoretical Computer Science*, Vol.61, pp.149-174, 1985.

[34] Krithivassan K., Mahajan M., Systolic pyramid automata, cellular automata and array languages, *International Journal of Pattern Recognition and Artificial Intelligence*, Vol3, Issue 3-4, pp405-433.

[35] Ladner R. E., Lipton R. J., and Stockmeyer L. J., Alternating pushdown automata, *Proceedings of the 19th IEEE Symposium on Foundations of Computer Science*, pp.92-106, 1978.

[36] Michel P., A survey of space complexity, *Theoretical Computer Science*, Vol.101, pp.99-132, 1992.

[37] Morita K., Umeo H., and Sugata K., Computational complexity of L (m, n) tape-bounded two-dimensional tape Turing machines (in Japanese), *The Transactions of the IEICE*, Vol.J60-D, No.11, pp.982-989, 1977.

[38] Morita K., Computational complexity in one- and two-dimensional tape automata, *Ph.D Thesis Osaka University*, 1978.

[39] Morita K., Umeo H., and Sugata K., Language recognition abilities of several two-dimensional tape automata and their relation to tape complexities (in Japanese), *The Transactions of the IEICE*, Vol.J60-D, No.12, pp.1077-1084, 1977.

[40] Morita K., Umeo H., and Sugata K., Accepting ability of offside-free two-dimensional marker automata−The simulation of four-way automata by three-way tape-bounded Turing machines, *Technical Report of the IEICE*, No.AL79-2, 1979.

[41] Nagatomo M., Sakamoto M., Susaki H., Zhang S., Ikeda S., Furutani H., Ito T., Uchida Y., and Yoshinaga T., Sufficient spaces for seven-way four-dimensional Turing machines to simulate four-dimensional one-marker automata, *Proceedings of the International Conference on Artificial Life and Robotics*, *Oita*, *Japan*, OS6-1, pp.340-343, 2015.

[42] Nakamura A. and Aizawa K., Detection of interlocking components in three-

dimensional digital pictures, *Information Sciences*, Vol.40, pp.143-153, 1986.

[43] Okabe H., Sakamoto M., and Inoue K., Computational complexity for the simulation of four-dimensional marker automata by seven-way four-dimensional Turing machines, *Proceedings of the 8th international symposium on artificial life and robotics, Oita, Japan*, Vol.1, pp.114-117, 2003.

[44] Paul W. J., Prauss E. J., and Reischuk R., On alternation; *Acta Informatica*, Vol.14, pp.243-255, 1980.

[45] Paul W. J., Prauss E. J., and Reischuk R., On alternation; *Acta Informatica*, Vol.14, pp.391-403, 1980.

[46] Rosenfeld A. and Milgram D. L., Parallel / sequential array automata, *Information Processing Letters*, Vol.2, pp.43-46, 1973.

[47] Rosenfeld A., *Picture Languages−Formal Models for Picture Recognition*, Academic Press, 1979.

[48] Ruzzo W. L., Tree-size bounded alternation, *Journal of Computer and System Sciences*, Vol.21, pp.218-235, 1980.

[49] Sakamoto M., Sakuramoto I., Inoue K., and Takanami I., A space lower-bound technique for two-dimensional alternating Turing machines, *IEICE Trans.*, D-I, J74-DI(4), pp.311-314, 1991.

[50] Sakamoto M., Inoue K., and Takanami I., A note on three-dimensional alternating Turing machines with space smaller than *log*(*m*), *Information Sciences*, Vol.72, pp.225-249, 1993.

[51] Sakamoto M., Ito A., Inoue K., and Takanami I., Simulation of three-dimensional one-marker automata by five-way Turing machines, *Information Sciences*, Vol.77, pp.77-99,

1994.

[52] Sakamoto M., Inoue K., and Takanami I., Three-dimensionally fully space constructible functions, *IEICE Transactions on Information and Systems*, Vol.E77-D, No.6, pp.723-725, 1994.

[53] Sakamoto M. and Inoue K., Three-dimensional alternating Turing machines with only universal states, *Information Sciences−Information and Computer Science*, Vol.95, pp.155-190, 1996.

[54] Sakamoto M., Okazaki T., and Inoue K., Three-dimensional multicounter automata, *Proceedings of the 6th International Workshop on Parallel Image Processing and Analysis−Theory and Applications*, *Madras*, *India*, pp.267-280, 1999.

[55] Sakamoto M., Three-dimensional alternating Turing machines, *Ph.D Thesis Yamaguchi University*, 1999

[56] Sakamoto M., Okabe H., and Inoue K., Some properties of four-dimensional finite automata, *2002 Chugoku-Section Joint Convention Record of Institutes of Electrical and Information Engineerings, Japan*, pp.351, 2002.

[57] Sakamoto M., Nogami S., Inoue K., and Kono M., A relationship between the accepting powers of three-dimensional finite automata and time-bounded bottom-up pyramid cellular acceptors with three-dimensional layers, *Trans. of SCI* (*Japan*), Vol.17, No.10, pp.451-458, 2004.

[58] Sakamoto M., Ito T., Tomozoe N., and Furutani H., A survey of three-dimensional automata, *The papers of Technical Meeting on Information Systems, IEE, Japan*, IS-07-12, pp.11-17, 2007.

[59] Sakamoto M., Ito T., Furutani H., and Kono M., Some accepting powers of three-dimensional parallel Turing machines, *1st European Workshop on Artificial Life and Robotics, Vienna, Austria*, pp.73-79, 2007.

[60] Sakamoto M., Okabe H., Nagami S., Taniguchi S., Maki T., Nakama Y., Saito M., Kono M., and Inoue K., A note on four-dimensional finite automata, *WSEAS Transactions on Computers*, Vol.3, Issue 5, pp.1651-1656, 2004.

[61] Sakamoto M., Tomozoe N., Furutani H., Kono M., Ito T., Uchida Y., and Okabe H., A survey of automata on three-dimensional input tapes, *WSEAS Transactions on Computers*, Vol.7, Issue 10, pp.1638-1647, 2008.

[62] Sakamoto M., Ito T., Furutani H., and Kono M., Some accepting powers of three-dimensional parallel Turning machines, I*nternational Journal of Artificial Life and Robotics*, *Springer*, Vol.13, No.1, pp.27-30, 2008.

[63] Sakamoto M., Fukuda M., Okatani S., Ito T., Furutani H., and Kono M., Path-bouded Three-dimensional finite automata, *International Journal of Artificial life and Robotics, Springer*,Vol.13, No.1, pp.54-57, 2008.

[64] Sakamoto M., Okatani S., Fukuda M., Ito T., Furutani H., and Kono M., A relationship between Turning machines and finite automata on four-dimensional input tapes, *International Journal of Artificial life and Robotics, Springer*, Vol.13, No.1, pp.58-60, 2008.

[65] Sakamoto M., Okatani S., Kajisa K., Fukuda M., Matsuawa T., Taniue A., Ito T., Furutani H., and Kono M., Hierarchies based on the number of cooperating systems of three-dimensional finite automata, *International Journal of Artificial Life and Robotics,*

Springer, Vo.4, No.3, pp.425-428, 2009.

[66] Sakamoto M., Matsukawa T., Katamune R., Furutani H., Kono M., Ikeda S., Ito T., Uchida Y., and Yoshinaga T., Four-dimensional synchronized alternating Turning machines, *Proceedings of the 10th American Conference on Applied Mathmatics, Harvard University, Cambridge, USA*, pp.195-200, 2010.

[67] Sakamoto M., Katamune R., Matsukawa T., Furutani H., Kono M., Ikeda S., Ito T., Uchida Y., and Yoshinaga T., Some result about hierarchy and recognizability of four-dimensional synchronized alternating Turning machines, *Proceedings of the 10th American Conference on Applied Mathematics*, *Harvard University*, *Cambridge*, *USA*, pp.201-205, 2010.

[68] Sakamoto M., Matsukawa T., Katamune R., Furutani H., Kono M., Ikeda S., Ito T., Uchida Y., and Yoshinaga T., Synchronized Alternating Turning Machines on Four-Dimensional Input Tapes, *WSES TRANSACTIONS on COMPUTERS,* Issue 4,Vol.9, pp.319-328, 2010.

[69] Sakamoto M., Katamune R., Matsukawa T., Furutani H., Kono M., Ikeda S., Ito T., Uchida Y., and Yoshinaga T., Hardware Hierarchies and Recognizabilities of Four-Dimensional Synchronized Alternating Turning Machines, *WSEAS TRANSACTIONS on COMPUTERS*, Issue 4, pp.329-338, 2010.

[70] Sakamoto M., Ito T., Qingquan X., Uchida Y., Yochinaga T., Yokomichi M., Ikeda S., and Furutani H., A Note on Three-dimensional Probabilistic Finite Automata, *The Seventeenth International Symposium on Artificial Life and Robotics 2012*, *Oita*, *Japan*, pp.492- 495, 2012.

[71] Sakamoto M., Uchida Y., Hamada T., Ito T., Yoshinaga T., Yokomichi M., Ikeda S., and Furutani H., A Relationship between the Accepting Powers of Nondeterministic Finite Automata and Probabilistic Finite Automata on Three-Dimensional Input Tapes, *the 2012 IEICE General Conference, Okayama University*, p.4, 2012.

[72] Shah A. N., Pebble automata on arrays, *Computer Graphics Image Processing*, Vol.3, pp.236-246, 1974.

[73] Shah A. N., Pushdown automaton arrays, *Information Sciences*, Vol.25, pp.175-193, 1981.

[74] Stearns R. E., Hartmanis J., and Lewis P. M., Hierarchies of memory limited computations, *Proceedings of the IEEE Conference on Switching Circuit Theory and Logical Design*, pp.179-190, 1965.

[75] Sudborough I. H., Efficient algorithms for paths system programs and applications to alternating and time-space complexity classes, *Proceedings of the 21st IEEE Symposium on Foundations of Computer Science*, pp.62-73, 1980.

[76] Szepietowski A., On three-way two-dimensional multicounter automata, *Information Sciences*, Vol.55, pp.35-47, 1991.

[77] Taniguchi H., Inoue K., and Takanami I., A note on three-dimensional finite automata, *Information Sciences*, Vol.26, pp.65-85, 1982.

[78] Taniguchi H., Inoue K., and Takanami I., *k*-neighborhood template *A*-type 2-dimensional bounded cellular acceptor (in Japanese), *The Transactions of the IEICE*, Vol.J69-D, No.3, pp.291-301, 1986.

[79] Taniguchi H., Inoue K., and Takanami I., Hierarchical properties of the *k*- neighborhood

template *A*-type 2-dimensional bounded cellular acceptor (in Japanese), *The Transactions of the IEICE*, Vol.J69-D, No.7, pp.1025-1034, 1986.

[80] Toffoli T. and Margolus N., Cellullar automaton machines − new environment for modeling, MIT Press, 1987.

[81] Turing A. M., On computable number, with an application to the Entscheidungsproblem, *Proceedings of the London Math. Soc.*, Vol.2, No.42, pp.230-265, 1936.

[82] Wiedermann J., Parallel Turing machines, *Technical Report RUU-CS-84-11, Department of Computer Science, University of Utrecht, the Netherlands*, 1984.

[83] Yamamoto Y., Morita K., and Sugata K., Space complexity for recognizing connected patterns in a three-dimensional tape (in Japanese), *Technical Report of the IEICE*, No.AL79-104, pp.91-96, 1979.

[84] Yamamoto H. and Noguchi S.: Time-and-leaf bounded 1-tape alternating Turing machines, *The Trans. of IECE, J68-D*, pp.1719-1726, 1985.