International Journal of Software Engineering and Its Applications
Vol. 9, No. 4 (2015), pp. 73-84
http://dx.doi.org/10.14257/ijseia.2015.9.4.09

Remarks on Four-Dimensional Probabilistic Finite Automata

Makoto Sakamoto[1], Yasuo Uchida[2], Makoto Nagatomo[1], Tuo Zhang[1], Hikaru Susaki[1], Takao Ito[3], Tsunehiro Yoshinaga[4], Satoshi Ikeda[1], Masahiro Yokomichi[1] and Hiroshi Furutani[1]

[1]*University of Miyazaki, Miyazaki, Miyazaki 889-2192, Japan*
[2]*National Institute of Technology, Ube College, Ube, Yamaguchi 755-8555, Japan*
[3]*Hiroshima University, Higashi-Hiroshima, Hiroshima 739-8511, Japan*
[4]*National Institute of Technology, Tokuyama College, Shunan, Yamaguchi 745-8585, Japan*
(*Tel: 81-985-58-7392, Fax: 81-985-58-7392*)
(*sakamoto@cs.miyazaki-u.ac.jp*)

Abstract

This paper deals with the study of four-dimensional automata. Recently, due to the advances in many application areas such as dynamic image processing, computer animation, augmented reality (AR), and so on, it is useful for analyzing computation of four-dimensional information processing (three-dimensional pattern processing with time axis) to explicate the properties of four-dimensional automata. From this point of view, we have investigated many properties of four-dimensional automata and computational complexity. On the other hand, the class of sets accepted by probabilistic machines have been studied extensively. As far as we know, however, there is no results concerned with four-dimensional probabilistic machines. In this paper, we introduce four-dimensional probabilistic finite automata, and investigate some accepting powers of them.

Keywords: *Alternation, Chunk, Finite automaton, Four-dimension, Probability*

1. Introduction

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 [31] in 1936 to answer fundamental problems of computer — 'What kind of logical work can we effectively perform ?' 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 importance. Such a study was active in 1950's and 1960's. On the other hand, many researchers have been making their effects to investigate another fundamental problems 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 space complexity is very useful to characterize various types of automata from a point of view of memory requirements [16]. This study was motivated by Stearns, Hartmanis, and Lewis in 1965 [30]. They introduced an L(m) space-bounded one-dimensional Turing machine to formalize the motion of space complexity, and

ISSN: 1738-9984 IJSEIA

investigated its computing ability. Some results were refined by Hopcroft and Ullman [7, 8]. Moreover, Chandra, Kozen, and Stockmeyer introduced an alternating as a theoretical model of parallel computation in 1976 [2]. 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 nondeterministic Turing machine is an alternating Turing machine which has only existential states. After that, the growth 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 and Hewitt first proposed two-dimensional automata — two-dimensional finite automata and marker automata, and investigated their pattern recognition abilities in 1967 [1]. Since then, many researchers in this field have been investigating a lot of properties about automata on a two-dimensional tape [11]. Moreover, 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 study 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 topological 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 thirty-five years, automata on a three-dimensional tape have been proposed and many properties of such automata have been obtained [23].We have also studied about three-dimensional automata, and introduced many computational models on three-dimensional input tapes [23]. By the way, the question of whether processing four-dimensional digital patterns (three-dimensional digital patterns with time axis) such as computer animation, dynamic image processing, and so on is much difficult than two- or three-dimensional ones is of great interest from the theoretical and practical standpoints. From this point of view, we first proposed four-dimensional automata as computational models of four-dimensional pattern processing in 2002, and investigated their several accepting powers [21]. Since then, we have introduced several four-dimensional automata, and investigated their accepting powers and recognizability [22, 24, 26, 27, 32]. On the other hand, the classes of sets recognized by two- or three-dimensional probabilistic finite automata and probabilistic Turing machines have been studied extensively [3-6, 9, 12-15, 17-19, 25, 28, 29, 33-35]. As far as we know, however, there is no results concerning with four-dimensional probabilistic machines. In this paper, we introduce four-dimensional probabilistic finite automata, and investigate some their accepting powers. Especially, we show the relationship of the accepting powers between a probabilistic finite automaton and an alternating finite automaton on four-dimensional input tapes.

2. Preliminaries

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 four-dimensional tape over Σ is denoted by $\Sigma^{(4)}$ (as shown in Figure 1).

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 $\Sigma^{(n1,\ n2,\ n3,n4)}$. 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 $x[(i_1,i_2,i_3,i_4), (i_1',i_2',i_3',i_4')]$-*segment* of x.)

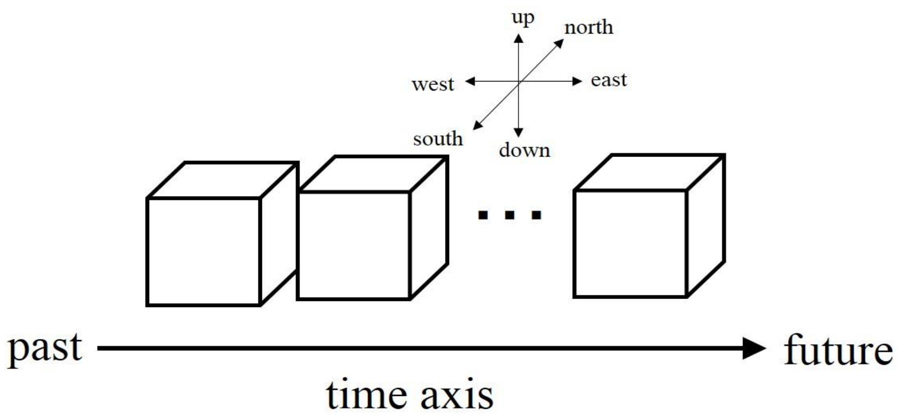

Figure 1. Four-dimensional Input Tape

We next introduce a four-dimensional probabilistic finite automaton which is a natural extension of a three-dimensional probabilistic finite automaton to four dimension [25]. Let S be a finite set. A *coin-tossing distribution on S* is a mapping Ψ from S to $\{0, \frac{1}{2}, 1\}$ such that $\Sigma_{a\in S}\,\Psi(a)=1$. The mapping means "choose a with probability $\Psi(a)$".

Definition 2.2. A *four-dimensional probabilistic finite automaton* (denoted by 4-*PFA*) is the 6-tuple

$$M = (Q, \Sigma, \delta, q_0, q_a, q_r),$$

where

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

(2) Σ is a finite set of *input symbols*;

(3) δ is a *transition function*;

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

(5) $q_a \in Q$ is the *accepting state*; and

(6) $q_r \in Q$ is the *rejecting state*.

An input tape for M is a four-dimensional tape over Σ surrounded by the boundary symbols #'s (not in Σ). The transition function δ is defined on $(Q\text{-}\{q_a, q_r\})\times(\Sigma \cup \{\#\})$ such that for each $q \in Q\text{-}\{q_a, q_r\}$ and each $\sigma \in \Sigma \cup \{\#\}$, $\delta[q, \sigma]$ is a coin-tossing distribution on $Q\times$\{East, West, South, North, Up, Down, Past, Future, Stay\}, where East means 'moving east', West 'moving west', South 'moving south', North 'moving north', Up 'moving up', Down 'moving down', Past 'moving in the past direction', Future 'moving in the future direction' and Stay 'staying there'. The meaning of δ is that if M is in state q with the input head scanning the symbol σ, then with probability $\delta[q, \sigma]$ (q', d) the machine enters state q' and either moves the input head one symbol in direction d if $d \in$ {East, West, South, North, Up, Down, Past, Future} or does not move the input head if d=Stay.

Given an input tape $x \in \Sigma^{(4)}$, M starts in state q_0 with the input head on the upper northwest corner of the first three-dimensional rectangular array of x. The computation of M on x is then governed (probabilistically) by the transition function δ until M either accepts by entering the accepting state q_a or rejects by entering the rejecting state q_r. We assume that δ is denoted so that the input head never falls off an input tape out of the boundary symbols #'s. M halts when it enters state q_a or q_r.

Let $L \subseteq \Sigma^{(4)}$ and $0 \leq \varepsilon < \frac{1}{2}$. A 4-*PFA* M recognizes L with error probability ε if for all $x \in L$, M accepts x with probability at least $1-\varepsilon$, and for all $x \notin L$, M rejects x with probability at least $1-\varepsilon$. Denote by 4-*PFA* the class of sets recognized by 4-*PFA*'s with error probability less than $\frac{1}{2}$.

Finally, we introduce a four-dimensional alternating finite automaton, which can be considered as an alternating version of a four-dimensional finite automaton [22]. In other words, it is a four-dimensional alternating Turing machine which has no storage tape and storage-tape head. Therefore, we begin with the explanation of the definition of four-dimensional alternating Turing machine.

Definition 2.3. A *four-dimensional alternating Turing machine* (denoted by 4-*ATM*) 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* ($\# \notin \Sigma$ is the *boundary symbol*);

(6) Γ is a finite *storage-tape alphabet* ($B \in \Gamma$ *is* the *blank symbol*); and

(7) $\delta \subseteq (Q \times (\Sigma \cup \{\#\}) \times \Gamma) \times (Q \times (\Gamma - \{B\}) \times$ {East, West, South, North, Up, Down, Past, Future, No move} × {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 three-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, as shown in Figure 2. 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-tape head falls off the storage tape (by moving left), then machine M can make no further move.

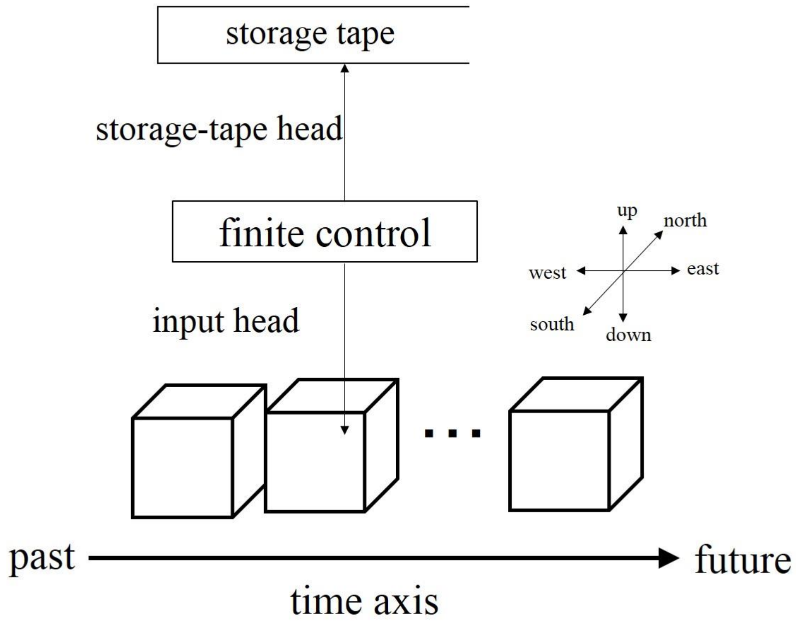

Figure 2. Four-dimensional Alternating Turing Machine

Definition 2.4. 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

$$C_M = (\mathbf{N} \cup \{0\})^4 \times S_M,$$

where $S_M = Q \times (\Gamma\text{-}\{B\})^{*} \times \mathbf{N}$ and **N** denotes the set of all positive integers. The first component x of a *configuration* (A configuration represents an instantaneous description of M at some point in a computation. We note that $0 \leq i_1 \leq l_1(x)+1$, $0 \leq i_2 \leq l_2(x)+1$, $0 \leq i_3 \leq l_3(x)+1$, $0 \leq i_4 \leq l_4(x)+1$, and $1 \leq j \leq +|\alpha|+1$, where for any string w, $|w|$ denotes the length of w (with $|\lambda|=0$, where λ is the *null string*.) $c=(x,((i_1, i_2, i_3, i_4),(q, \alpha, j)))$ 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.

Definition 2.5. 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. $\vdash_M^*$ denotes the reflexive transitive closure of $\vdash_M$. The relation $\vdash_M$ is not necessarily single-valued, because δ is not. A *computation path* of M on x is a sequence $c_0 \vdash_M c_1 \vdash_M \cdots \vdash_M c_n$ $(n \geq 0)$, 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. we define

$$T(M)=\{x\in \Sigma^{(4)} \mid M \text{ accepts } x\}.$$

We next define a restricted type of 4-*ATM*, called a space-bounded 4-*ATM*.

Definition 2.6. Let $L(m)$:**N** $\mapsto$**R** be a function with one variable m, where **N** is the set of all positive integers and **R** is the set of all nonnegative real numbers. With a 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, $SPACE(l(v)) \leq \lceil L(m)\rceil$ ($\lceil r\rceil$ is the smallest integer than or equal to r.) By 4-*ATM*($L(m)$) *we denote an* $L(m)$ *space-bounded* 4-*ATM*.

A 4-*ATM*(0) is called a four-dimensional alternating finite automaton, as shown in Figure 3, and denoted by 4-*AFA*.

Let $\mathcal{L}$[4-*PFA*] = $\{T \mid T = T(M)$ for some 4-*PFA M* $\}$. $\mathcal{L}$[4-*AFA*] is defined in the same way as $\mathcal{L}$[4-*PFA*].

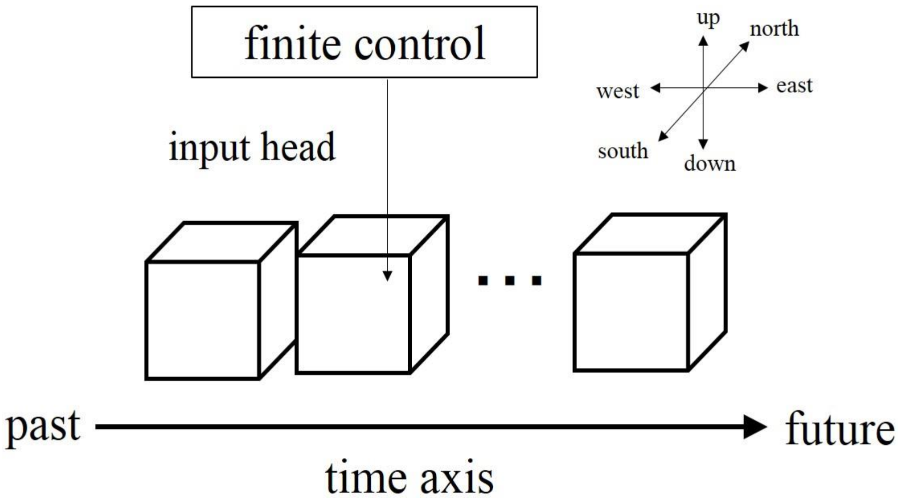

Figure 3. Four-dimensional Alternating Finite Automaton

3. Main Results

This section shows that the 4-*PFA* is incomparable with 4-*AFA*. We first give several preliminaries to get our desired results. Let M be a 4-*PFA* and Σ be the input alphabet of M. For each k, m, n ($k \geq 1$, $m \geq k+1$, $1 \leq n \leq m-1$), an (m, n, k)-*chunk* over Σ is a four-dimensional object obtained from a four-dimensional tape in $\Sigma^{(m, m, m, m)}$ by cutting off the [(m-k+1, 1,1,1), (m, n, 1, 1)]-*segment*. We show the image of the first three-rectangular array of an (m, n, k)-*chunk* as shown in Figure 4.

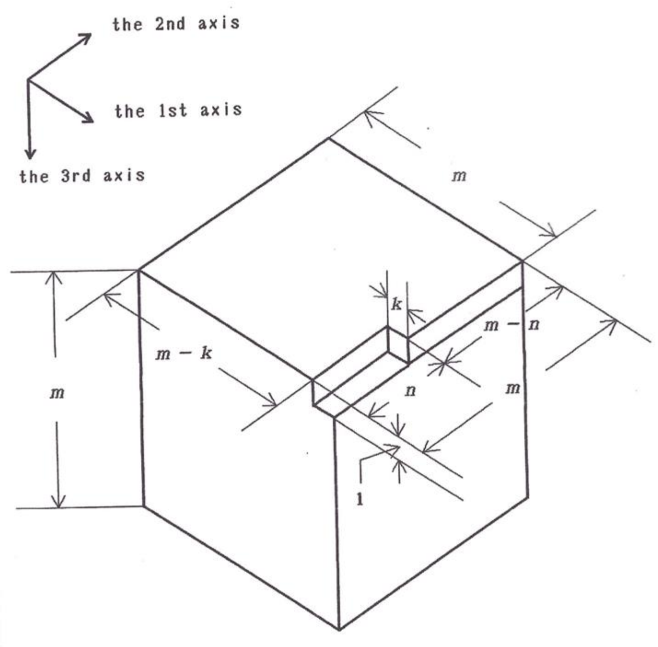

Figure 4. The First Three-Dimensional Rectangular Array of an (*m*, *n*, *k*)-chunk

Especially, an (m, n, 1)-chunk over Σ is called an (m, n)-*chunk* over Σ. For an (m, n, 1)-chunk x, we denote by x(#) the object (obtained from x by surrounding x by the boundary symbols #'s) . We show the image of the first three-rectangular array of x(#) as shown in Figure 5.

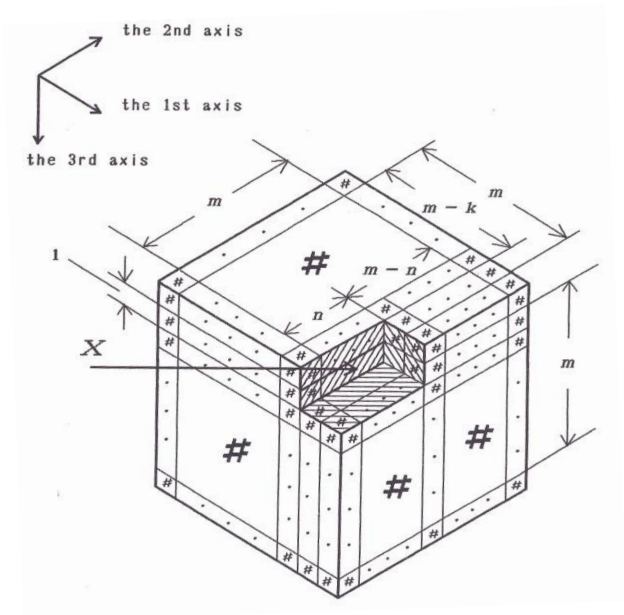

Figure 5. The First Three-Dimensional Rectangular Array of *x*(#)

Suppose that a four-dimensional automaton enters or exits the object $x(\#)$ only at the face designated by the bold line in Figure 6.

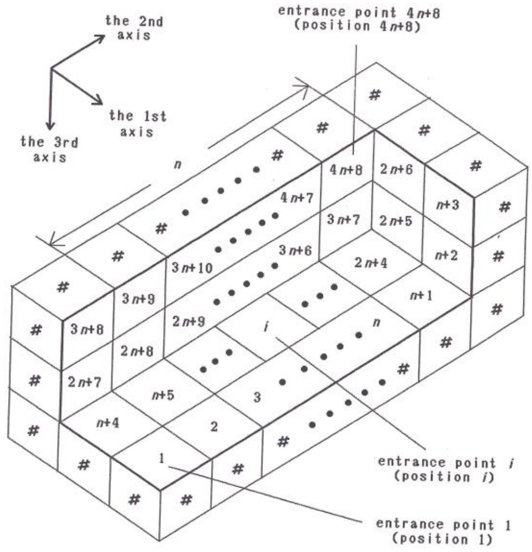

Figure 6. Entrance Points to *x*(#) and Positioning of the Voxel of *x*(#)

Let M be a 4-PFA. Below we assume without loss of generality that for any (m, n)-chunk x ($m \geq 2$, $1 \leq n \leq m$-1) over the input alphabet Σ of M, M has the following property.

M enters or exits the object $x(\#)$ only at the shaded face as shown in Figure 5, and M never enters an accepting state in $x(\#)$. Thus, the number of the entrance points to $x(\#)$ [or the exit points from $x(\#)$] for M is 4n+8. We suppose that these entrance points (or exit points) are numbered 1, 2, … , 4n+8 in an appropriate way as shown in Figure 6. Let M be a 4-PFA with accepting state q_a and rejecting state q_r, and x be an (m, n, k)-chunk (or an (m, n)-chunk) over the input alphabet of M ($k \geq 1$, $m \geq k+1$, $1 \leq n \leq m$-1). We define the *chunk probabilities* of M on x as follows. A *starting condition* for the chunk probability is a pair (q, l), where q is a state of M and $l \in PT(x(\#))$, where $PT(x(\#))$ is the set of these entrance points (or exit points); its intuitive meaning is "M has just entered $x(\#)$ in state q from entrance point l of $x(\#)$". A *stopping condition* for the chunk probability is either:

(i) a pair (q, l) as above, meaning that M exits from $x(\#)$ in state q at exit point l,

(ii) "Loop" meaning that the computation of M loops forever within $x(\#)$,

(iii) "Accpt" meaning that M halts in the accepting state q_a before exiting from $x(\#)$ at an exit point of $x(\#)$, or

(iv) "Reject" meaning that M halts in the rejecting state q_r before exiting from $x(\#)$ at an exit point of $x(\#)$.

For each starting condition σ and each stopping condition τ, let $p(x, \sigma, \tau)$ be the probability that stopping condition τ occurs given that M is started in starting condition σ on an (m, n, k)-chunk (or an (m, n)-chunk) x.

Computations of a 4-PFA are modeled by Markov chains [28] with finite state space, say {1, 2, … , s} for some s. A particular Markov chain is completely specified by its matrix $R=\{r_{ij}\}1 \leq i, j \leq s$ of transition probabilities. If the Markov chain is in state i, then it next moves to state j with probabilities r_{ij}. The chains we consider have the designated starting state, say, state 1, and some set T_r of trapping states, so r_{ij}=1 for all $t \in T_r$. For $t \in T_r$, let p^* $[t, R]$ denote the probability that Markov chain R is trapped in state t when started in state 1.

We are now ready to prove our key lemma.

Lemma 3.1. *Let* $L_1 = \{x \in \{0, 1\}^{(4)} | \ l_4(x) \geq 2 \ \& \ \exists k \ (2 \leq r \leq l_4(x)) \ [x[(1, 1, 1, 1), (l_1(x), l_2(x), l_3(x), 1)] = x[(1, 1, 1, r), (l_1(x), l_2(x), l_3(x), r)]]$ *(i.e., the top three-dimensional rectangular array of x is identical with some another three-dimensional rectangular array of x)]. Then,* $L_1 \in \mathcal{L}[4\text{-}AFA]\text{-} \ \mathcal{L}[4\text{-}PFA]$.

Proof: L_1 is accepted by the 4-*AFA* M with acts as follows. Given an input tape x with $l_4(x) \geq 2$, M existentially tries to check that, for each $i, j, k(1 \leq i \leq l_1(x), 1 \leq j \leq l_2(x), 1 \leq k \leq l_3(x))$, $x(i, j, k, r) = x(i, j, k, 1)$. That is, on the rth three-dimensional rectangular array of $x(1 \leq i \leq l_1(x), 1 \leq j \leq l_2(x), 1 \leq k \leq l_3(x), 1 \leq r \leq l_4(x))$, M enters a universal state to choose one of two further actions. One action is to pick up the symbol $x(i, j, k, r)$, move up the symbol store in the finite control, compare the stored with the symbol $x(i, j, k, 1)$, and enter an accepting state if both symbols are identical. The other action is to continue to move next tape cell (in order to pick up the symbol $x(i + 1, j + 1, k+1, r)$ and compare it with the symbol $x(i + 1, j + 1, k+1, r)$ and compare it with the symbol $x(i + 1, j + 1, k+1, 1)$. It will be obvious that M accepts L_1. We next show that $L_1 \notin$ 4-*PFA*. Suppose to the contrary that there exists a 4-*PFA* M' recognizing L_1 with error probability ϵ <1/2. For large n, let $V(n)$ be the set of all the $(2^n, n)$-chunks over {0, 1}. We shall below consider the computations of M' on the input tapes x with $l_1(x) = l_2(x) = l_3(x) = l_4(x) = 2^n$. Let c be the number of states of M'. Consider the chunk probabilities $p(v, \sigma, \tau)$ defined above. For each $(2^n, n)$-chunk v in $V(n)$, there are a table of

$$d(n) = c \times | PT(v(\#)) | \times (c \times | PT(v(\#)) | + 3) = O(n^2)$$

chunk probabilities, where for any S, $|S|$ denotes the number of elements of S. Fix some ordering of the pairs (σ, τ) of starting and stopping conditions and let $\boldsymbol{p}(v)$ be the vector of these $d(n)$ probabilities according to this ordering. By using the counting argument and reduction to absurdity, we can derive the lemma [17].

Lemma 3.2. *Let* $L_2 = \{x \in \{0, 1\}^{(4)} | \ l_4(x) = 1 \& \ (x$ *is of the form* $0^n 1^n$ *for some* $n \geq 1)\}$. *Then,* $L_2 \in \mathcal{L}[4\text{-}PFA]\text{-} \ \mathcal{L}[4\text{-}AFA]$.

Proof: It is shown the L_2 is recognized by a two-way probabilistic finite automaton with error probability ϵ < 1/2 [1, 3]. On the other hand, it is showed that alternating finite automata accept only regular sets. Thus $L_2 \in \mathcal{L}[4\text{-}PFA]$ - $\mathcal{L}[4\text{-}AFA]$ by using the same technique.

From Lemmas 3.1 and 3.2, we have the following theorem.

Theorem 3.1. 4-*PFA is incomparable with* 4-*AFA.*

4. Conclusion

It was introduced four-dimensional probabilistic finite automata 4-*PFA*'s and shown their some properties in this paper. Especially, we showed that the accepting powers of 4-*PFA*'s were incomparable with the accepting powers of 4-*AFA*'s. We conclude this paper by giving the following some open problems.

(1) Let 4-PFA^c (resp. 4-AFA^c) be the class of sets of cu-bic tapes recognized by 4-PFA^c's with error probability less than (resp., accepted by 4-AFA^c's). Is 4-PFA^c incomparable with 4-AFA^c ?

(2) Let T^c be all the four-dimensional connected tapes. Is Tc recognized by 4-PFA^c's ?

(3) It will be interesting to investigate the properties of various four-dimensional probabilistic Turing machines.

(4) It will be also interesting to deal with the closure properties of 4-PFA^c's.

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

References

[1] M. Blum and C. Hewitt, "Automata on a two-dimensional tape", IEEE Symposium on Switching and Automata Theory, **(1967)**, pp. 155-160.

[2] A. K. Chandra, D. C. Kozen and L. J. Stockmeyer, "Alternation", Journal of the ACM, vol. 28, no. 1, **(1981)**, pp. 114-133.

[3] C. Dwork and L. J. Stockmeyer, "Finite state verifier I: the power of interaction", Journal of the ACM Proceedings of the International Symposium on Mathematical Foundations of Computer Science, Lecture Notes in Computer Science 118, Springer, New York, vol. 39, no. 4, **(1992)**, pp. 800-828.

[4] R. Freivalds, "Probabilistic two-way machines", Proceedings of the International Symposium on Mathematical Foundations of Computer Science, Lecture Notes in Computer Science, Springer, New York, 118, **(1981)**.

[5] J. Gill, "Computational complexity of probabilistic Turing machines", Journal of the SIAM, Comput., vol. 6, no. 4, **(1977)**, pp. 675-695.

[6] A. G. Greenberg and A. Weiss, "A lower bound for probabilistic algorithms for finite state machines", Journal of Computer and System Sciences, vol. 33, **(1986)**, pp. 88-105.

[7] J. E. Hopcroft and J. D. Ullman, "Formal languages and their relation to automata", Addison-Wesley, Reading, Mass, **(1967)**.

[8] J. E. Hopcroft and J. D. Ullman, "Some results on tape-bounded Turing machines", Journal of the Association for Computer Machinery, vol. 16, **(1967)**, pp. 168-177.

[9] K. Inoue, I. Takanami and A. Nakamura, "A note on two-dimensional finite automata", Information Processing Letters, vol. 7, no. 1, **(1978)**, pp. 49-52.

[10] K. Inoue, I. Takanami and H. Taniguchi, "Two-dimensional alternating Turing machines", Theoretical Computer Science, vol. 27, **(1983)**, pp. 61-83.

[11] K. Inoue and I. Takanami, "A survey of two-dimensional automata theory", Information Sciences, vol. 55, **(1991)**, pp. 99-121.

[12] A. Ito, K. Inoue and Y. Wang, "Nonclosure properties of two-dimensional one-marker automata", International Journal of Pattern Recognition and Artificial Intelligence, vol. 11, no. 7, **(1997)**, pp. 1025-1050.

[13] J. Kaneps, "Regulality of one-letter languages acceptable by 2-way finite probabilistic automata", Proceedings of the Fundamentals of Computation Theory, Lecture Notes in Computer Science, Springer, New York, vol. 529, **(1991)**, pp. 287-296.

[14] M. Karpinski and R. Verbeek, "On the Monte Carlo space constructible functions and separation results for probabilistic complexity classes", Information and Computation, vol. 75, **(1987)**, pp. 178-189.

[15] Macarie II, "Multihead two-way probabilistic finite automata", Theoretical Computer Science, vol. 30, **(1997)**, pp. 91-109.

[16] P. Michel, "A survey of space complexity", Theoretical Computer Sciences, vol. 101, **(1992)**, pp. 99-132.

[17] T. Okazaki, K. Inoue, A. Ito and Y. Wang, "Closure properties of the classes of sets recognized by space-bounded two-dimensional probabilistic Turing machines", Information Sciences, vol. 115, **(1999)**, pp. 61-81.

[18] M. O. Rabin, "Probabilistic automata", Information and Control, vol. 6, **(1963)**, pp. 230-245.

[19] A. Rosenfeld, "Picture Language (Formal Models for Picture Recognition)", Academic Press, New York, **(1979)**.

[20] M. Sakamoto, "Three-dimensional alternating Turing machines", Ph.D. Thesis, Yamaguchi University, **(1999)**.

[21] M. Sakamoto, H. Okabe and K. Inoue, "Some properties of four-dimensional finite automata", Chugoku-Section Joint Convention Record of Institutes of Electrical and Information Engineering, Shimane, Japan, vol. 351, **(2002)**.

[22] M. Sakamoto, H. Okabe, S. Nogami, S. Taniguchi, T. Makino, Y. Nakama, M. Saito, M. Kono and K. Inoue, "A note on four-dimensional finite automata", WSEAS Transactions on Computers, vol. 5, no. 3, pp. 1651-1656.
[23] M. Sakamoto, N. Tomozoe, H. Furutani, M. Kono, T. Ito, Y. Uchida and H. Okabe, "A Survey of Automata on Three-Dimensional Input Tapes", WSEAS Transactions on Computers, vo. 10, no. 7, pp. 1638-1647.
[24] M. Sakamoto, T. Matsukawa, R. Katamune, H. Furutani, M. Kono, S. Ikeda, T. Ito, Y. Uchida and T. Yoshinaga, "Synchronized alternating Turing machines on four-dimensional input tapes", WSEAS Transactions on Computers, vol. 4, no. 9, **(2010)**, pp. 319-328.
[25] M. Sakamoto, T. Ito, X. Qingquan, Y. Uchida, T. Yoshinaga, M. Yokomichi, S. Ikeda and H. Furutani, "A note on three-dimensional probabilistic finite automata", Proceedings of the 17th International Symposium on Artificial Life and Robotics, Oita, Japan OS-22-4, **(2012)**, pp. 492-495.
[26] M. Sakamoto, Y. Uchida, M. Nagatomo, T .Ito, T. Yoshinaga, S. Ikeda, M. Yokomichi and H. Furutani, "Hierarchies based on the number of cooperating systems of finite automata on four-dimensional input tapes", World Academy of Science, Engineering and Technology, vol. 72, **(2012)**, pp. 1068-1073.
[27] M. Sakamoto, M. Nagatomo, T. Zhang, X. Feng, T. Kurogi, S. Ikeda, M. Yokomichi, H. Furutani, T. Ito, Y. Uchida and T. Yoshinaga, "Hierarchy based on configuration-reader about k-neighborhood template *A*-type three-dimensional bounded cellular acceptor", Journal of Robotics, Networks and Artificial Life, vol. 1, no. 1, **(2014)**, pp. 80-84.
[28] E. Seneta, "Non-negative matrices and Markov Chains", 2nd ed., Springer, New York, **(1981)**.
[29] G. Siromoney, R. Siromoney and K. Krithivasan, "Picture languages with array rewriting rules", Information and Control, vol. 22, **(1973)**, pp. 447-470.
[30] R. E. Stearns, J. Hartmanis and P. M. Lewis II, "Hierarchies of memory limited computations", IEEE Conference on Switching Circuits Theory and Logical Design, **(1965)**, pp. 179-190.
[31] A. M. Turing, "On computable numbers, with an application to the Entscheidungs problem", Proceedings of the London Mathematical Society, vol. 2, no. 42, **(1936)**, pp. 230-265.
[32] Y. Uchida, T. Ito, M. Sakamoto, R. Katamune, K. Uchida, H. Furutani, M. Kono, S. Ikeda and T. Yoshinaga, "A relationship between marker and inkdot for four-dimensional automata", International Journal Computers, NAUN, vol. 1, no. 5, **(2011)**, pp. 66-73.
[33] J. Wang, "A note on two-way probabilistic automata", Information Processing Letters, vol. 43, **(1992)**, pp. 321-326.
[34] Y. Yamamoto and S. Noguchi, "Time-and-leaf bounded 1-tape alternating Turing machines (in Japanese)", The Transactions of the IECE J68-D(10):1719-1726, **(1985)**.
[35] Y. Yamamoto, K. Morita and K. Sugata, "Space complexity for recognizing connected patterns in a three-dimensional tape", (in Japanese), Technical Report of the IECE AL79-104, pp. 91-96.