A Note on Four-Dimensional Finite Automata

Makoto SAKAMOTO, Hidenobu OKABE, Shigehisa NOGAMI, Shinya TANIGUCHI,
Tadayuki MAKINO, Youichirou NAKAMA, Makoto SAITO, Michio KONO
Department of Computer Science and Systems Engineering, Faculty of Engineering,
University of Miyazaki, Miyazaki, 889-2192, Japan

and

Katsushi INOUE
Department of Computer Science and Systems Engineering, Faculty of Engineering,
Yamaguchi University, Ube, 755-8611, Japan

Abstract: - During the past about thirty-five years, many types of two- or three-dimensional automata have been proposed and investigated the properties of them as the computational model of pattern processing. On the other hand, recently, due to the advances in many application areas such as computer animation, motion image processing, and so on, the study of three-dimensional pattern processing with the time axis has been of crucial importance. Thus, we think that it is very useful for analyzing computation of three-dimensional pattern processing with the time axis to explicate the properties of four-dimensional automata. In this paper, we propose a four-dimensional Turing machine and a four-dimensional finite automaton, and show the space complexities necessary and sufficient for seven-way four-dimensional Turing machines to simulate four-dimensional finite automata, where each sidelength of each input tape of these automata is equivalent.

Key-Words: - Turing machine, finite automaton, on-line tessellation acceptor, four-dimensional input tape, determinism, nondeterminism, space-bound, configuration, computation, space complexity

1 Introduction

Since M. Blum, et al. showed the relation of automata and image recognition [1], many researchers investigated a lot of properties about automata on two- and three-dimensional tape [6,8].

By the way, recently, due to the advances in many application areas such as computer animation, virtual reality systems, motion image processing, and so on, the study of three-dimensional pattern processing with the time axis has been of crucial importance. Thus, we think that it is very useful for analyzing computation of three-dimensional pattern processing with the time axis to explicate the properties of four-dimensional automata, i.e., three-dimenisonal automata with the time axis. In [9], we proposed a four-dimensional automaton.

In this paper, we introduce a four-dimensional Turing machine and a four-dimensional finite automaton, and show the space complexities necessary and sufficient for seven-way four-dimensional Turing machines to simulate four-dimensional finite automata, where each sidelength of each input tape of these automata is equivalent in order to increase the theoretical interest.

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 the four-dimensional tapes over Σ is denoted by $\Sigma^{(4)}$. Given a tape $x \in \Sigma^{(4)}$, for each $j (1 \leq j \leq 4)$, we let $l_j(x)$ be the length of x along the j^{th} 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 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 $(1 \leq r_1 \leq l_1(y), 1 \leq r_2 \leq l_2(y),$
$1 \leq r_3 \leq l_3(y), 1 \leq r_4 \leq l_4(y))$,
$y(r_1, r_2, r_3, r_4)$
$= x(r_1 + i_1 - 1, r_2 + i_2 - 1, r_3 + i_3 - 1, r_4 + i_4 - 1)$.

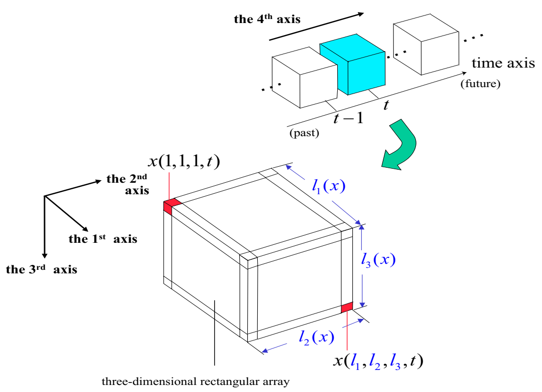

Fig. 2. 1: Four-dimensional tape.

(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.) When a four-dimensional tape x is given to any four-dimensional automaton as an input, we assume that x is surrounded by the boundary symbol $\sharp$.

[Definition 2.2.] A *four-dimensional finite automaton* (4-FA) M has a read-only four-dimensional input tape with boundary symbols $\sharp$'s, a finite control, and an input head. The input head can move in eight directions−east, west, south, north, up, down, future, or past−unless it falls off the input tape.

Formally, M is defined by the 5-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 ($\sharp \notin \Sigma$ is the boundary symbol),

(5) $\delta \subseteq (Q \times (\Sigma \cup \{\sharp\})) \times (Q \times \{$east,west,south,north,up, down,future,past,no move$\})$ is the *next-move relation*.

The action of M is similar to that of one-dimensional finite automaton, except that the input head of M can move in eight directions. That is, when an input tape $x \in \Sigma^{(4)}$ with boundary symbols is presented to M, M starts in its initial state q_0 with the input head on $x(1,1,1,1)$, and determines the next state of the finite control and the move direction of the input head, depending on the present state of the finite control and the symbol read by the input head. We say that M *accepts* the tape x if it eventually enters an accepting state.

[Definition 2.3.] A *seven-way four-dimensional Turing machine* (SV4-TM) M has a read-only four-dimensional input tape with boundary symbols $\sharp$'s and one semi-infinite storage tape, initially blank. Of course, M has a finite control, an input head which can move in seven direction - east, west, south, north, up, down, or future - unless it falls off the input tape, and a storage-tape head. A *position* is assigned to each cell on the read-only input tape and to each cell of the storage tape. Formally, M is defined by the 6-tuple

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

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* ($\sharp \notin \Sigma$ is the *boundary symbol*),

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

(6) $\delta \subseteq (Q \times (\Sigma \cup \{\sharp\}) \times \Gamma) \times (Q \times (\Gamma - \{B\}) \times \{$east, west, south, north, up, down, future, no move$\} \times \{$right, left, no move$\})$.

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. We say that M *accepts* the input tape if it eventually enters an accepting state.

We next consider the another restricted type of SV4-TM, called a *space-bounded* SV4-TM.

[**Definition 2.4.**] Let $L(n)$: $\mathbf{N} \rightarrow \mathbf{R}$ be a function of a variable n, where $\mathbf{N}$ is the set of all positive integers and $\mathbf{R}$ is the set of all nonnegative real numbers. An SV4-TM M is said to be $L(n)$ *space-bounded* if for no input tape $x \in \Sigma^{(4)}$ with $l_1(x)$=$l_2(x)$=$l_3(x)$=$l_4(x)$=n does M scan more than $L(n)$ cells on the storage tape. We denote an $L(n)$ space-bounded SV4-TM by SV4-TM($L(n)$).

A 4-FA [SV4-TM, SV4-TM($L(n)$)] is also nondeterministic in general. In order to distinguish between determinism and nondeterminism, we denote a deterministic 4-FA [nondeterministic 4-FA, deterministic SV4-TM, nondeterministic SV4-TM, deterministic SV4-TM($L(n)$), nondeterministic SV4-TM($L(n)$)] by 4-DFA [4-NFA, SV4-DTM, SV4-NTM, SV4-DTM($L(n)$), SV4-NTM($L(n)$)], respectively.

Let M be an automaton on a four-dimensional tape. We denote by $T(M)$ the set of all the four-dimensional tapes accepted by M. As usual, we denote, for example, by $\mathcal{L}$[4-DFA] the class of sets of all the four-dimensional tapes accepted by 4-DFA's. That is, $\mathcal{L}$[4-DFA]=$\{T | T$=$T(M)$ for some 4-DFA $M\}$. $\mathcal{L}$[4-NFA], $\mathcal{L}$[SV4-DTM], and so on have analogous meanings.

We complete this section by investigating the difference between the accepting powers of 4-DFA's and 4-NFA's. We can get the following lemma by extending Theorem 1 in [7] to four-dimensions. The proof is omitted here, since it is similar to that of Theorem 1 in [7].

[**Lemma 2.1.**] *Let* T_1=$\{x \in \{0, 1\}^{(4)} | ^{\exists}n \geq 1 [l_1(x)$= $l_2(x)$=$l_3(x)$=$l_4(x)$=$2n+1$ & $x(n+1, n+1, n+1, n+1)$=1$]\}$. *Then*

(1) $T_1 \in \mathcal{L}$[4-NFA], *and*

(2) $T_1 \notin \mathcal{L}$[4-DFA].

From Lemma 2.1, we get the following theorem.

[**Theorem 2.1.**] $\mathcal{L}$[4-DFA] $\subsetneq$ $\mathcal{L}$[4-NFA]

3 Main Results

We first investigate the space-bound for SV4-DTM's to simulate 4-DFA's.

[**Lemma 3.1.**] $\mathcal{L}$[4-DFA] $\subseteq$ $\mathcal{L}$[SV4-DTM($n^3 \log n$)].

(**Proof**) Let M be a 4-DFA. By using the same idea as in the proof of Proposition 1 in [7], we can show that there exists an SV4-DTM(n^3log$n$) M' such that $T(M')$=$T(M)$. (In what follows, the base of logarithms is to be taken as 2.) □

[**Lemma 3.2.**] Let T_2=$\{x \in \{0, 1\}^{(4)} | ^{\exists}n \geq 1$ $[l_1(x)$ =$l_2(x)$=$l_3(x)$=$l_4(x)$=$2n$ & $^{\forall}i_1$, $^{\forall}i_3$ ($1 \leq i_1 \leq 2n$, $1 \leq i_3 \leq n$) [*there exists exactly one* '1' *on* $[(i_1$, 2, i_3, 1), (i_1, $2n$, i_3, 1)]$-$*segment of* x] & $^{\exists}j_1$, $^{\exists}j_3$ ($1 \leq j_1 \leq 2n$, $n+1 \leq j_3 \leq 2n$) [$x(j_1$, 1, j_3, $2n$)=1 & $^{\forall}k_1$, $^{\forall}k_3$ ($1 \leq k_1 \leq 2n$ & $n+1 \leq k_3 \leq 2n$ & (k_1, 1, k_3, $2n$)$\neq$(j_1, 1, j_3, $2n$)) [$x(k_1$, 1, k_3, $2n$)=0] & $^{\exists}r_2$ ($2 \leq r_2 \leq 2n$) [x(j_1, r_2, $j_3 - n$, $2n$)=$x(j_1$, r_2, 1, 1)=1]]]$\}$, *and let* $L(n)$:$\mathbf{N} \rightarrow \mathbf{R}$ *be a function such that* $\lim_{n \to \infty}[L(n)/n^3 \log n]$=0. *Then*

(1) $T_2 \in \mathcal{L}$[4-DFA], *and*

(2) $T_2 \notin \mathcal{L}[SV4\text{-}DTM(L(n))]$.

(**Proof**) (1) We consider the 4-DFA M which acts as follows. Let x be a four-dimensional input tape with each sidelength $2n$ ($n \geq 1$) to be presented to M.

(i) By moving in eight directions (east, west, south, north, up, down, past, and future), M checks, first of all, if there exists one '1' on $x[(i_1$, 2, i_3, 1), (i_1, $2n$, i_3, 1)], for each i_1 ($1 \leq i_1 \leq 2n$), i_3 ($1 \leq i_3 \leq n$). If M

succeeds in this check, then go to (ii). Otherwise, go to (iv).

(ii) M scans $x[(1,\ 1,\ n{+}1,\ 2n),\ (2n,\ 1,\ 2n,\ 2n)]$, and checks if there exists exactly one '1' there. If M succeeds in this check, then go to (iii). Otherwise, go to (iv).

(iii) Let $x\ (j_1,\ 1,\ j_3,\ 2n)$=1 $(1 \leq j_1 \leq 2n, n+1 \leq j_3 \leq 2n)$. M places its input head on $x(j_1,\ 1,\ j_3,\ 2n)$ and continues to move the input head two cells east and one cell up, until the input head reaches the boundary symbol $\sharp$. It is easy to see that the input head reaches the boundary symbol with coordinates $(j_1,\ 2n{+}1,\ j_3{-}n,\ 2n)$. M then continues to move the input head west, until it meets '1' for the first time. From (i), let $x(j_1,\ r_2,\ j_3{-}n,\ 2n)$=1 for some r_2 ($2{\leq}r_2{\leq}2n$). M then places the input head on $x(j_1,\ r_2,\ j_3{-}n,\ 2n)$ and continues to move up against the time axis until the input head reaches the boundary symbol $\sharp$. After that, M moves the input head one cell down and reads the symbol on the cell, that is, $x(j_1,\ r_2,\ 1,\ 1)$. If $x(j_1,\ r_2,\ 1,\ 1)$=1, then M enters an accepting state; otherwise go to (iv).

(iv) M halts in a nonaccepting state.

It is easy to see that the set accepted by M is identical with T_2. This completes the proof of (1).

(2) Suppose that there exists some SV4-DTM($L(n)$) M accepting T_2, and that q is the number of states of its finite control and t is the number of storage symbols. For each $n{\geq}1$, let

$V(n)$=$\{x{\in}\{0,\ 1\}^{(4)}$ | $l_1(x){=}l_2(x){=}l_3(x){=}l_4(x){=}$ $2n$ & $^{\forall}i_1,\ ^{\forall}i_3,\ ^{\forall}i_4$ $(1{\leq}i_1{\leq}2n,\ 1{\leq}i_3{\leq}2n,\ 1{\leq}i_4{\leq}n)$ [there exists extactly one '1' on $x[(i_1,\ 2,\ i_3,\ i_4),\ (i_1,\ 2n,\ i_3,\ i_4)]]\}$.

For each x in $V(n)$, let

conf(x) $\triangleq$ The configuration of M just after the input head left $x(4)$ along the time axis. (For any Turing machine M', we define the configuration of M' to be a combination of the (1) state of the finite control, (2) position of the input head, (3) position of the storage head within the nonblank portion of the storage tape, and (4) contents of the storage tape.)

Then the following proposition must hold.

[Proposition 3.1.] *For any two tapes x, y in $V(n)$ such that their* $[(1,\ 2,\ 1,\ 1),\ (2n,\ 2n,\ 2n,\ n)]$*−segments are different,*

$$conf(x) \neq conf(y).$$

(**Proof of Lemma 3.2** (*continued*)) Let $p(n)$ be the number of tapes in $V(n)$ such that their $[(1,\ 2,\ 1,\ 1),\ (2n,\ 2n,\ 2n,\ n)]$-segment are different. Clearly, $p(n){=}(2n-1)^{4n^3}$. On the other hand, let $c(n)$ be the number of possible configuration of M just after the input head left the n^{th} three-dimensional rectangualr arrays of tapes in $V(n)$. Then we get the inequality

$$c(n) \leq q(2n+2)^3 L(2n) t^{L(2n)}$$

Since $\lim_{n\to\infty}[L(2n)/8n^3 \log 2n]$=0 (by the assumption of the lemma), it follows that $\lim_{n\to\infty}$ $[L(2n)/4n^3 \log(2n-1)]$=0, and thus $p(n){>}c(n)$ for large n. Therefore, it follows that for large n there must be two tapes x, y in $V(n)$ such that their $[(1,\ 2,\ 1,\ 1),\ (2n,\ 2n,\ 2n,\ n)]$-segments are different and $conf(x){=}conf(y)$. This contradicts Proposition 3.1, and thus part (2) of the lemma also holds. □

From Lemmas 3.1 and 3.2, we can get the following theorem.

[Theorem 3.1.] *$n^3 \log n$ space is necessary and sufficient for $SV4-DTM's$ to simulate $4-DFA's$.*

We next investigate the space bound for SV4-DTM's to simulate 4-NFA's. By using the same ideas in [5,10], we can get the following propositions.

[Proposition 3.2.] *For any function $L(n) : \mathbf{N}{\to}\mathbf{R}$ such that $L(n){\geq}n^4$* ($n{\geq}1$),

$$\mathcal{L}[SV4-DTM(L(n))] = \mathcal{L}[4-DTM(L(n))].$$

[Proposition 3.3.] *For any function $L(n) : \mathbf{N}{\to}\mathbf{R}$ such that $L(n){\geq}\log n$* ($n{\geq}1$),

$$\mathcal{L}[4-NTM(L(n))] \subseteq \mathcal{L}[4-DTM([L(n)]^2)].$$

From Propositions 3.2 and 3.3, we can get the following lemma.

[**Lemma 3.3.**] $\mathcal{L}$[4-NFA] $\subseteq \mathcal{L}$[SV4-DTM(n^4)].

(**Proof**) From the definitions, $\mathcal{L}$[4-NFA] $\in \mathcal{L}$[4-NTM(0)] $\in \mathcal{L}$[4-NTM($\log n$)], and by Proposition 3.3, $\mathcal{L}$[4-NTM($\log n$)] $\subseteq \mathcal{L}$[4-DTM($[\log n]^2$)] $\subseteq \mathcal{L}$[4-DTM(n^4)]. Furthermore , it follows by Proposition 3.2 that $\mathcal{L}$[4-DTM(n^4)]=$\mathcal{L}$ [SV4-DTM(n^4)], and thus the lemma holds. □

[**Lemma 3.4.**] *Let* T_3=$\{x{\in}\{0,\ 1,\ 2\}^{(4)}\ |\ {}^{\exists}n{\ge}1\ [l_1(x){=}l_2(x){=}l_3(x){=}l_4(x){=}2n$ & $[x(1, 1, 2n, 2n){=}2$ & ${}^{\forall}(q, r, s, t)\ ({\ne}(1, 1, 2n, 2n))[x(q, r, s, t) \in \{0, 1\}]$ & $x[(2n, 2, 1, 1), (2n, 2n, 1, 1)]{\ne}x[(1, 2, 2n, 2n), (1, 2n, 2n, 2n)]]]\}$, *and let* $L(n) : \mathbf{N}{\to}\mathbf{R}$ *be a function such that* $\lim_{n\to\infty}[L(n)/n^4]$=0. *Then*

(1) $T_3 \in \mathcal{L}[4-NFA]$, *and*

(2) $T_3 \notin \mathcal{L}[SV4-DTM(L(n))]$.

(**Proof**) (1) We consider the 4-NFA M which acts as follows. Let x be a four-dimensional input tape with sidelength $2n$ ($n{\ge}1$) to be presented to M.

(i) M checks first of all if there exists exactly one '2' on the last plane of the last three-dimensional rectangular array. If M succeeds in this check, then go to (ii), and otherwise, go to (iii).

(ii) Let $x(1,\ 1,\ 2n,\ 2n){=}2$. M places its input head on $x(1,\ 1,\ 2n,\ 2n)$ and continues to move the input head one cell east and one cell up, until the input head reaches the boundary symbol ♯, and M then continues to move the input head one cell west and one cell south after moving one cell down, until the input head reaches the boundary symbol ♯. (This action is making a zigzag of 45°-direction from $(1,\ 1,\ 2n,\ 2n)$ to $(1,\ 2n,\ 1,\ 2n)$.) Similarly, M continues to move, by making a zigzag of 45°-direction from bottom three-dimensional rectangualr array to top three-dimensional rectangular array, that x has exactly $2n$ three-dimensional rectangular arrays. It is easy to see that the input head reaches the boundary symbol with coordinates $(2n,\ 0,\ 1,\ 1)$. M then continues to move the input head east. During this action, M chooses some i ($2{\le}i{\le}2n$) nondeterministically, picks up $x(2n,\ i,\ 1,\ 1)$, and stores it in the finite control. Then M continues to move input head down and north. Each time the input head reads a northmost symbol of each plane which is different from the symbol $x(2n,\ i,\ 1,\ 1)$ stored in the finite control, M nondeterministically chooses the action (a) or action (b) below :

(a) M continues to move the input head down and north.

(b) M continues to move input head west, and checks that input head meets the symbol '2'. If M succeeds in this check (note that $x[(2n,\ 2,\ 1,\ 1),\ (2n,\ 2n,\ 1,\ 1)]$ is not identical with $x[(1,\ 2,\ 2n,\ 2n),\ (1,\ 2n,\ 2n,\ 2n)]$ in this case). M enters an accepting state. Otherwise, go to (iii). If M continues to choose the action (a) and the input head reaches the boundary symbol, then go to (iii).

(iii) M halts in a nonaccepting state. It will be obvious that $T(M){=}T_3$. This completes the proof (1).

(2) : Suppose that there exists some SV4-DTM($L(n)$) M accepting T_3, and q is the number of states of its finite control and t is the number of storage symbols. For each $n{\ge}1$, let

$V(n){=}\{x \in \{0,\ 1,\ 2\}^{(4)}\ |\ l_1(x){=}l_2(x){=}l_3(x){=}l_4(x){=}2n$ & $x[(1,\ 1,\ 1,\ 1),\ (2n,2n,\ 2n,\ n)] \in \{0,\ 1\}^{(4)}\}$.

and for each x in $V(x)$, let

conf(x) ≜ the configuration of M just after the input head left the nth three-dimensional rectangular array.

Then the following proposition must hold.

[**Proposition 3.4.**] *For any two tapes* x, y *in* $V(n)$ *such that their* $[(1,\ 2,\ 1,\ 1),\ (2n,\ 2n,\ 2n,\ n)]-$*segments are different,*

$$conf(x) \neq conf(y).$$

(**Proof of Lemma 3.4** (continued)) Now let $p(n)$ be the number of tapes in $V(n)$ such that their [(1, 2, 1, 1), $(2n,\ 2n,\ 2n,\ n)$]-segments are different. It is clear that $p(n) = 2^{4n^2(2n-1)n}{=}2^{8n^4-4n^3}$. On the other hand, let $c(n)$ be the number of possible configurations of M just after the input head left the n^{th} three-dimensional rectangular arrays of tapes in $V(n)$. Thus we get the inequality

$$c(n) \le q(2n+2)^3 L(2n) t^{L(2n)}.$$

Since $\lim_{n\to\infty}[L(2n)/16n^4]$=0 (by the assumption of the lemma), it follows that $\lim_{n\to\infty}\ [L(2n)/(8n^4 - 4n^3)]$=0, and thus $p(n){>}c(n)$ for large n. Therefore, it follows that for large n there must be two tapes x, y in $V(n)$ such that their $[(1, 2, 1, 1), (2n, 2n, 2n, n)]$-segments are different and conf(x)=conf(y). This contradicts Proposition 3.4, and thus part (2) of the lemma also holds. □

From lemmas 3.3 and 3.4, we can get the following theorem.

[**Theorem 3.2.**] n^4 *space is necessary and sufficient for* $SV4-DTM's$ *to simulate* $4-NFA's$.

We next investigate the space bound for SV4-NTM's to simulate 4-DFA's or 4-NFA's.

[**Lemma 3.5.**] $\mathcal{L}$[4-NFA] $\subseteq$ $\mathcal{L}$[SV4-NTM(n^3)].

(**Proof**) As shown in Lemma 4.1 in [4], the class of sets accepted by two-dimensional nondeterministic on-line tessellation acceptors. By using the same idea as in the proof of this fact, we can show that $\mathcal{L}$[4-NFA] $\subseteq$ $\mathcal{L}$[4-OTA], where 4-OTA denotes the four-dimensional nondeterministic on-line tessellation acceptor (see [2,3] for the definition of this acceptor), and $\mathcal{L}$[4-OTA] denotes the class of sets accepted by 4-OTA's on four-dimensional input tapes. On the other hand, it is easy to see from the definition of 4-OTA's that $\mathcal{L}$[4-OTA] $\subseteq$ $\mathcal{L}$[SV4-NTM(n^3)]. Therefore, it follows that $\mathcal{L}$[4-NFA] $\subseteq$ $\mathcal{L}$[SV4-NTM(n^3)], and thus the lemma holds. □

The following lemma is an extension of lemma 3.1 in [5] to four dimensions. The proof is omitted here, since it is similar to that of lemma 3.1 in [5].

[**Lemma 3.6.**] *Let* T_4={x *in* {0, 1}$^{(4)}$|$^{\exists}n$≥2 [$l_1(n)$=$l_2(n)$=$l_3(n)$= $l_4(n)$=n & $x(4)_1$=$x(4)_2$}, *and let* $L(n)$: $\mathbf{N}\rightarrow\mathbf{R}$ *be a function such that* $\lim_{n\rightarrow\infty}[L(n)/n^3]$=0. *Then*

(1) $T_4 \in \mathcal{L}$[4-DFA], *and*

(2) $T_4 \notin \mathcal{L}$[SV4-NTM($L(n)$)].

From Lemmas 3.5 and 3.6, we can get the following theorem.

[**Theorem 3.3.**] n^3 *space is necessary and sufficient for* $SV4-NTM's$ *to simulate* $4-DFA's$ *or* $4-NFA's$.

4 Conclusion

In this paper, we investigated the space complexities for simulation of four-dimensional finite automata by seven-way four-dimensional Turing machines, where each sidelength of each four-dimensional input tape is equivalent. We conclude this paper by giving Table 4.1, which show the space complexities for simulation of 4-FA's by SV4-TM's.

X \ Y	*SV4-DTM*	*SV4-NTM*
4-*DFA*	$\Theta(n^3 \log n)$	$\Theta(n^3)$
4-*NFA*	$\Theta(n^4)$	$\Theta(n^3)$

Table 4. 1: Necessary and sufficient space for Y's to simulate X's, where each sidelength of each four-dimensional input tape is n.

References:

[1] M. Blum, et al., *IEEE Symp.* : 1967.

[2] K. Inoue, et al., *Int. Symp. on Uniformly Structured Automata and Logic* : 1975.

[3] K. Inoue, et al., *Trans. of the IECE* J59-D(4) : 1976.

[4] K. Inoue, et al., *Inform. Sci.* 13 : 1977.

[5] K. Inoue, et al., *Inform. Sci.* 17 : 1979.

[6] K. Inoue, et al., *Inform. Sci.* 55 : 1991.

[7] K. Morita, et al., *Tech. Rep. of the IEICE* AL79-2 : 1979.

[8] M. Sakamoto, *Ph.D Thesis* : 1999.

[9] M. Sakamoto, et al., *2002 Chugoku-Section Joint Convention Rec. of Institutes of Electrical and Inform. Eng., Japan* : 2002.

[10] W. J. Savitch, *J. of Comput. and Syst. Sci.* 4 : 1970.