Synchronized Alternating Turing Machines on Four-Dimensional Input Tapes

Makoto SAKAMOTO, Tomoya MATSUKAWA, Ryoju KATAMUNE,
Hiroshi FURUTANI, Michio KONO, and Satoshi IKEDA
Department of Computer Science and Systems Engineering
University of Miyazaki
1-1 Gakuen Kibanadai Nishi, Miyazaki,
Miyazaki 889-2192, JAPAN
E-mail: $sakamoto@cs.miyazaki\text{-}u.ac.jp$

Takao ITO and Yasuo UCHIDA
Department of Business Administration
Ube National College of Technology
14-1, Tokiwadai 2-chome Ube
Yamaguchi 755-8555, JAPAN
E-mail: $ito@ube\text{-}k.ac.jp$

Tsunehiro YOSHINAGA
Department of Computer Science
and Electronic Engineering
Tokuyama College of Technology
Gakuendai, Shunan, Yamaguchi 745-8585, JAPAN
E-mail: $yosinaga@tokuyama.ac.jp$

Abstract: Synchronized alternating machine is an alternating machine with a special subset of internal states called synchronizing states. This paper introduces a four-dimensional synchronized alternating Turing machine (4-$SATM$), and investigates fundamental properties of 4-$SATM$'s. The main topics of this paper are: (1) a relationship between the accepting powers of 4-$SATM$'s and four-dimensional alternating Turing machines with small space bounds, (2) a relationship between the accepting powers of seven-way and eight-way 4-$SATM$'s, (3) a relationship between the accepting powers of 4-$SATM$'s and four-dimensional nondeterministic Turing machines. In this paper, we let each sidelength of each input tape of these automata be equivalent in order to increase the theoretical interest.

Key–Words: alternation, computational complexity, configuration, four-dimensional Turing machine, synchronization

1 Introduction

Alternating Turing machines were as a model of parallel computation[5,8,29-32,39,43,57,58,62]. Informally, an alternating Turing machine is a generalization of a nondeterministic Turing machine which can, at some point during a computation, split into several processes working in parallel and independently; an input is accepted if all parallel processes finish in accepting configurations. However, the alternating Turing machine is not a realistic model for realworld computers, because it does not allow any communications among its processes.

Synchronized alternating Turing machines were introduced in [12-15] to study the effect of allowing processes of an alternating Turing machine to communicate via synchronization. Informally, a synchronized alternating machine is an alternating machine with a special subset of internal states called synchronizing states. Each of these synchronizing states is associated with a synchronizing symbol. If, during the course of computation, some process enters a synchronizing state, then it has to wait until all other processes enter either an accepting state or a synchronizing state with the same synchronizing symbol. When this happens, all processes are allowed to continue their computation; otherwise, the machine is said to have a deadlock. A computation is successful if no deadlocks occur and all processes terminate in accepting states. It turns out that synchronization significantly increases the computaional power of alternating Turing machines.

On the other hand, Blum and Hewitt first proposed two-dimensional automata as computational models of two-dimensional pattern processing, and investigated their pattern recognition abilities[2-4,6,7,41,42,55,56,59]. Since then, many researchers in this field have been investigating a lot of properties about automata on a two-dimensional tape[16,19-25,34]. Recently, due to the advances

in computer vision, computer animation, moving picture processing, robotics, and so on,the study of multi-dimensional information processing has been of great importance[40]. Thus, the study of three- or four-dimensional automata has been meaningful as the computational model of multi-dimensional information processing[18,26-28,31,35-38,45-54,60,61,63,64]. From this viewpoint, we introduced four-dimensional alternating Turing machine [45,51].

In this paper, we continue the investigations about four-dimensional alternating Turing machines, introduce a four-dimensional synchronized alternating Turing machine (4-$SATM$), and investigate fundamental properties of 4-$SATM$'s.

In this section, we provide a background and a motive for our study of four-dimensional automata. Section 2 summarizes the formal definitions and notations necessary for this paper. Section 3 investigates a relationship between the accepting powers of four-dimensional synchronized alternating machines and four-dimensional nonsynchronized alternating machines. Section 4 investigates a relationship between the accepting powers of seven-way and eight-way synchronized machines. Section 5 imvestigates a relationship between the accepting powers of four-dimensional synchronized alternating machines and four-dimensional nondeterministic machines. Finally, Section 6 concludes this paper by giving some open problems.

2 Preliminaries

This section summarises the formal definitions and notations necessary for the paper.

Definition 2.1.

Let Σ be a finite set of symbols. A *four-dimensional input tape* over Σ is a four-dimensional rectangular array of elements of Σ. The set of all the four-dimensional input tapes over Σ is denoted by $\Sigma^{(4)}$. Given an input 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 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 is 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 $1 \leq i_j \leq i'_j \leq l_j(x)$ for each integer j $(1 \leq j \leq 4)$, as the four-dimensional input tape y satisfying the following;

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

(2) 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.)

We now introduce a four-dimensional synchronized alternating Turing machine.

Definition 2.2.

A four-dimensional synchronized alternating Turing machine (denoted by 4-$SATM$) is a 10-tuple M = (Q, q_0, U, E, S, F, Σ, Π, Γ, δ), where

(1) $Q = U \cup E \cup S$ is a finite set of *states*,

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

(3) U is the set of *universal states*,

(4) E is the set of *existential states*,

(5) $S \subseteq \{(q, s) : q \in U \cup E, s \in \Pi\}$ is the set of *synchronizing states* (*s-states*),

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

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

(8) Π is a finite *alphabet of synchronizing symbols*,

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

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

As shown in Fig.1, *M* has a read-only four-dimensional input tape with boundary symbols #'s ($\# \notin \Sigma$) and one semi-infinite storage tape, initially filled with the blank symbols. *M* begins in state q_0. A *position* is assigned to each cell of the input tape and the storage tape, as shown in Fig.1. 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 a process P enters a synchronizing state,

it stops and waits until all the parallel processes either enter the states with the same synchronizing element or stop in accepting states.

Definition 2.3.

An *instantaneous description* (ID) of a 4-$SATM$ M = (Q, q_0, U, E, S, F, Σ, Π, Γ, δ) is a pair of an element of $\Sigma^{(4)}$ and an element of

$$C_M = (N \cup \{0\})^4 \times S_M, S_M = Q \times (\Gamma - \{B\})^* \times \mathbf{N},$$

where **N** denotes the set of all positive integers. The first component of an ID $I = (x, ((i_1, i_2, i_3, i_4), (q, \alpha, k)))$ represents the input to M, and the first component (i_1, i_2, i_3, i_4) of the second component of I represents the input head position $(0 \leq i_1 \leq l_1(x) + 1, 0 \leq i_2 \leq l_2(x) + 1, 0 \leq i_3 \leq l_3(x) + 1, 0 \leq i_4 \leq l_4(x) + 1)$, and the second component (q, α, k) of the second component of I represents the state of the finite control, nonblank contents of the storage tape, and the storage head position $(1 \leq k \leq |\alpha| + 1)$. An element of C_M is called a *configuration* of M, and an element of S_M is called a *storage state* of M.

An ID is *universal* (*existential, synchronizing, accepting*) depending on the type of the state of the finite control. The *initial* ID of M on input x is $I_M(x) = (x, ((1, 1, 1, 1), (q_0, \lambda, 1)))$, where λ is the null word.

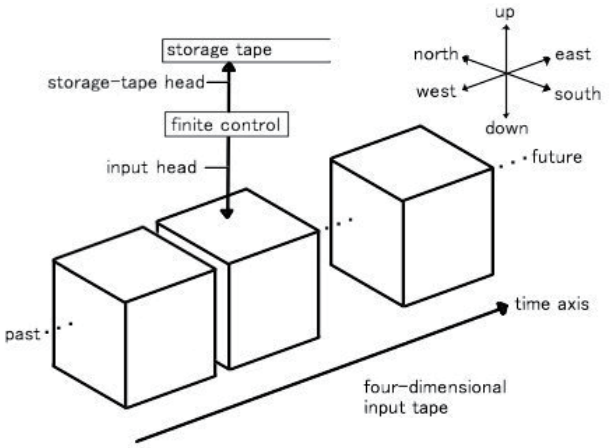

Fig. 1: Four-dimensional synchronized alternating Turing machine.

Definition 2.4.

Suppose I_1 and I_2 are two ID's of M and I_2 follows from I_1 in one step according to the next-move relation δ. Then we write $I_1 \vdash_M I_2$ and say that I_2 is a *successor* of I_1. The reflexive transitive closure of $\vdash_M$ is denoted by $\vdash_M^*$.

A sequence of ID's of M, $I_0, I_1, ..., I_m (m \geq 0)$, is called a *sequential computation* of M if $I_0 \vdash_M I_1 \vdash_M \cdots \vdash_M I_M$. If $I_0 = I_M(x)$ for some x, we call this sequence a $computation\ path$ of M on x[1,9-11].

The *full computation tree* of M on an input tape x is a (possibly infinite) labeled tree $\vdash_x^M$ (Each branch of $\vdash_x^M$ is called a *process*.) such that

(1) each node v is labeled by some ID I_v of M,

(2) the root is labeled by $I_M(x)$,

(3) v_2 is a direct descendant of v_1 iff $I_{v_1} \vdash_M I_{v_2}$.

The *synchronizing sequence* (*s-sequence*) of a node v in a full computation tree T with root v_0 is the sequence of synchronizing symbols occuring in labels of the nodes on the path from v_0 to v. Two s-sequences are *compatible* if one is a prefix of the other. If s_1 and s_2 are two compatible s-sequences, and s_2 is longer than s_1, then we use $s_2 - s_1$ to denote their difference.

A *computation tree* of M on an input x is a (possibly infinite) subtree T' of the full computation tree T_x^M satisfying the following conditions:

(1) if u is an internal (non-leaf) node of the tree T', I_u is universal and $\{I \mid I_u \vdash_M I\} = \{I_1, ..., I_m\}$, then u has exactly m children $v_1, ..., v_m$, such that $I_{v_i} = I_i, 1 \leq i \leq m$,

(2) if u is an internal node of the tree and I_u is existential, then u has exactly one chiled v such that $I_u \vdash I_v$,

(3) For arbitrary nodes u and v of T', the *s-sequences* of u and v are compatible.

If M on input x has no computation trees, then any subtree of T_x^M that satisfies the first two conditions above must have two processes with incompatible s-sequences. In this case, we say M *deadlocks* on x. The two processes with incompatible s-sequences are called *deadlock processes* and the nonmatching s-states causing the deadlock are called *deadlock states*.

The longest synchronizing sequence of a node in the computation tree T is called the *synchronizing sequence of the computation tree* T.

An *accepting computation tree* of M on an input x is a finite computation tree of M on x such that each leaf node is labeled by an accepting ID. We say that

M accepts x if there is an accepting computation tree of M on x. Let $T(M) = \{x \in \Sigma^{(4)} \mid M\ accepts\ x\}$.

We next introduce a seven-way four-dimensional synchronized alternating Turing machine which can be considered as a synchronized version of seven-way four-dimensional alternating Turing machine [12,14].

Definition 2.5.

A *seven-way four-dimensional synchronized alternating Turing machine* (denoted by SV4-$SATM$) is a 4-$SATM$ $M = (Q, q_0, U, E, S, F, \Sigma, \Pi, \Gamma, \delta)$, such that

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

That is, an SV4-$SATM$ is a 4-$SATM$ whose input head can move east, west, south, north, up, down, or in the future direction, but not in the past direction.

Definition 2.6.

Let $L(m)$: **N** $\rightarrow$ **N** be a function with one variable m. With each 4-$SATM$ (or SV4-$SATM$) M we assosiate a *space complexity function* SPACE which takes ID's to natural numbers. That is, for each ID $I = (x, ((i_1, i_2, i_3, i_4), (q, \alpha, k)))$, let SPACE($I$) be the length of α. We say that M is "$L(m)$ space-bounded" if for all m and for all 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 omputa tion tree of M on input x such that for each node π of the tree, SPACE $(I(\pi)) \leq L(m)$. By "4-$SATM(L(m))$" ("SV4-$SATM(L(m))$") we denote an $L(m)$ space-bounded 4-$SATM$ (SV4-$SATM$) which each sidelength of each input tape is equivalent[33].

Four-dimensional alternating Turing machines (4-ATM's) and seven-way four-dimensional alternating Turing machines (SV4-ATM's) in [38] are 4-$SATM$'s and SV4-$SATM$'s, respectively, which have no synchronizing states. We use 4-$SUTM$ (SV4-$SUTM$, 4-UTM, SV4-UTM) to denote a 4-$SATM$ (SV4-$SATM$, 4-ATM, SV4-ATM) which has no existential states. By 4-$ATM(L(m))$ (SV4-$ATM(L(m))$, 4-$SUTM(L(m))$, SV4-$SUTM(L(m))$, 4-$UTM(L(m))$, SV4-$UTM(L(m))$), we denote an $L(m)$ space-bounded 4-ATM (SV4-ATM, 4-$SUTM$, SV4-$SUTM$, 4-UTM, SV4-UTM).

A *four-dimensional deterministic Turing machine* (4-DTM) (*seven-way four-dimensional deterministic Turing machine* (SV4-DTM)) is a 4-ATM (SV4-ATM) whose ID's each have at most one successor, and a *four-dimensional nondeterministic Turing machine* (4-NTM) (*seven-way four-dimensional nondeterministic Turing machine* (SV4-NTM)) is a 4-ATM which has no universal states. We denote an $L(m)$ space-bounded 4-DTM(4-NTM, SV4-DTM, SV4-NTM) by 4-$DTM(L(m))$ (4-$NTM(L(m))$, SV4-$DTM(L(m))$, SV4-$NTM(L(m))$). We use 4-$SAFA$ (SV4-$SAFA$, 4-AFA, SV4-AFA, 4-NFA, SV4-NFA, 4-DFA,SV4-DFA) to denote a *four-dimensional synchronized alternating finite automaton* (*seven-way four-dimensional synchronized alternating finite automaton, four-dimensional alternating finite automaton, seven-way four-dimensional alternating finite automaton, four-dimensional nondeterministic finite automaton, seven-way four-dimensional nondeterministic finite automaton, four-dimensional deterministic finite automaton, seven-way four-dimensional deterministic finite automaton*). That is, a 4-$SAFA$ (SV4-$SAFA$, 4-AFA, SV4-AFA, 4-NFA, SV4-NFA, 4-DFA, SV4-DFA) is a 4-SATM (SV4-$SATM$, 4-ATM, SV4-ATM, 4-NTM, SV4-NTM, 4-DTM, SV4-DTM) which doesn't have storage tape. Similarly, we use 4-$SUFA$ (SV4-$SUFA$, 4-UFA, SV4-UFA) to denote a 4-$SUTM$ (SV4-$SUTM$, 4-UTM, SV4-UTM) which doesn't have the storage tape. Furthermore, for any integer $k \geq 1$, 4-$SATM(L(m))[k]$ is used to denote a 4-$SATM(L(m))$ such that any computation tree of M on any input x has at most k leaves. SV4-$SATM(L(m))[k]$, 4-$SUTM(L(m))[k]$,...,4-$SAFA(L(m))[k]$, etc. have the similar meaning. For any integer $k \geq 1$, 4-$NFA(k\text{-}heads)$ (4-$DFA(k\text{-}heads)$) is used to denote a 4-NFA (4-DFA) which has *k* input heads. For any machine class C, let

$$\mathcal{L}\ [\mathrm{C}] = \{\ T \mid T = T(M) \text{ for some } M \text{ in } C\ \}.$$

Thus, for example, $\mathcal{L}[$4-$SATM(L(m))]$ denotes the class of sets accepted by 4-$SATM(L(m))$'s.

3 Synchronization versus Non-Synchronization

This section investigates a relationship between the accepting powers of 4-ATM's and 4-$SATM$'s.

Lemma 3.1. *Let* $T_1 = \{x \in \{0,1\}^{2m\times 2m\times 2m\times 2m} \mid m \geq 1 \&\ x[(1,1,1,1),(2m,2m,2m,m)] = x[(1,1,1,m+1),(2m,2m,2m,2m)]\}$*(See Fig.2.). Then,*

(1) $T_1 \in \mathcal{L}[SV$4-$SUFA[2]]$, *and*

(2) $T_1 \notin \mathcal{L}[$4-$ATM(L(m))]$ *for any* L : $\boldsymbol{N} \rightarrow \boldsymbol{N}$ *such that* $L(m) = o(\log m)$.

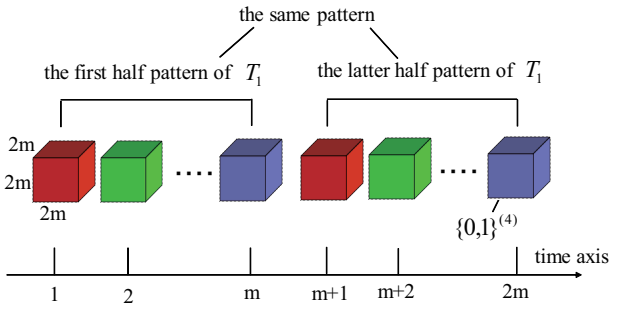

Fig. 2: A tape in T_1.

Proof: (1) We can construct an $SV4$-$SUFA$[2] M accepting T_1 as follows: Given x with $l_1(x) = l_2(x) = l_3(x) = l_4(x) = 2m$ ($m \geq 1$), starting on position (1,1,1,1) of x, M first splits universally into two processes p_1 and p_2. Process p_2 moves its head to $(1, 1, 1, m+1)$ and then synchronizes with process p_1 to compare $x(i_1, i_2, i_3, i_4)$ and $x(i_1, i_2, i_3, i_4+m)$ for each i_1, i_2, i_3, i_4 ($1 \leq i_1 \leq 2m$, $1 \leq i_2 \leq 2m$, $1 \leq i_3 \leq 2m$, $1 \leq i_4 \leq m$). M accepts x iff $x(i_1, i_2, i_3, i_4) = x(i_1, i_2, i_3, i_4 + m)$ for each i_1, i_2, i_3, i_4 ($1 \leq i_1 \leq 2m$, $1 \leq i_2 \leq 2m$, $1 \leq i_3 \leq 2m$, $1 \leq i_4 \leq m$).

(2) This proof is the same as that of Theorem 1 in [44]. □

From this lemma, we have

Theorem 3.1. *For any function* $L(m) = o(\log m)$,

(1) $\mathcal{L}[SV4\text{-}UTM(L(m))] \subsetneq \mathcal{L}[SV4\text{-}SUTM(L(m))]$,

(2) $\mathcal{L}[SV4\text{-}ATM(L(m))] \subsetneq \mathcal{L}[SV4\text{-}SATM(L(m))]$,

(3) $\mathcal{L}[4\text{-}UTM(L(m))] \subsetneq \mathcal{L}[4\text{-}SUTM(L(m))]$, *and*

(4) $\mathcal{L}[4\text{-}ATM(L(m))] \subsetneq \mathcal{L}[4\text{-}SATM(L(m))]$.

Corollary 3.1. *(1)* $\mathcal{L}[SV4\text{-}UFA] \subsetneq \mathcal{L}[SV4\text{-}SUFA]$,

(2) $\mathcal{L}[SV4\text{-}AFA] \subsetneq \mathcal{L}[SV4\text{-}SAFA]$,

(3) $\mathcal{L}[4\text{-}UFA] \subsetneq \mathcal{L}[4\text{-}SUFA]$, *and*

(4) $\mathcal{L}[4\text{-}AFA] \subsetneq \mathcal{L}[4\text{-}SAFA]$.

Theorem 3.2. *For any function* $L(m) \geq \log m$,
$\mathcal{L}[4\text{-}SUTM(L(m))] = \mathcal{L}[4\text{-}UTM(L(m))]$.

Proof: Given a 4-$SUTM(L(m))$ M where $L(m) \geq \log m$, we construct a 4-$UTM(L(m))$ M' to accept the same set as follows. On input x of sidelength $m \geq 1$, M' simulates each process of M with a process of its own. When some process p of M enters an s-state, the corresponding process p' of M' spawns off a process c whose worktape contains the s-symbol associated with the s-state and the number of s-states p has entered so far. Since each process makes at most $d^{L(m)}$ moves (d is a constant), and $L(m) \geq \log m$, there is enough space to store them. Process c restars the computation of M on x and verifies that the corresponding s-symbols in other processes match with the one stored on its worktape. If a discrepancy occurs, M' rejects. It is easy to see that M and M' accept the same set. □

By using a technique similar to that in the proof of Theorem 3.2, we have

Theorem 3.3. *For any function* $L(m) \geq \log m$,
$\mathcal{L}[SV4\text{-}SUTM(L(m))] = \mathcal{L}[SV4\text{-}UTM(L(m))]$.

4 Seven-Way versus Eight-Way

This section investigates a relationship between the accepting powers of seven-way and eight-way synchronized machines.

It is shown in [14] that three-way two-dimensional synchronized alternating Turing machine are equivalent to two-dimensional synchronized alternating Turing machines. By using the same idea as in the proof of this fact, we can easily show that the following theorem holds.

Theorem 4.1. *For any function* $L : \boldsymbol{N} \to \boldsymbol{N}$,

$$\mathcal{L}[SV4\text{-}SATM(L(m))] = \mathcal{L}[4\text{-}SATM(L(m))].$$

Below,we investigate a difference betwecn the accepting powers of space-bounded 4-$SUTM$'s and $SV4$-$SUTM$'s.

Lemma 4.1. *Let* $T_2 = \{x \in \{0,1\}^{m\times m\times m\times m} \mid m \geq 2,\ \& \ x[(1,1,1,1),(m,m,m,1)] \neq x[(1,1,1,2),(m,m,m,2)] \ \& \ x[(1,1,1,3),(m,m,m,3)] \in \{0\}^{(4)}\}$ *(See Fig.3.). Then,*

(1) $T_2 \in \mathcal{L}[4\text{-}DFA] (= \mathcal{L}[4\text{-}SUTM(0)[1]])$, *and*

(2) $T_2 \notin \mathcal{L}[SV4\text{-}SUTM(L(m))]$ *for any* $L : \boldsymbol{N} \to \mathbf{N}$ *such that* $L(m) = o(m^3)$.

Proof: (1) We can construct a 4-DFA M accepting T_2 as follows: Given x with $l_1(x) = l_2(x) = l_3(x) =$

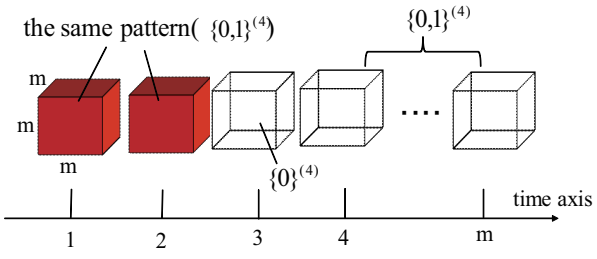

Fig. 3: A tape in T_2.

$l_4(x) = m$ ($m \geq 2$), starting on position (1,1,1,1) of x, M first checks that $x[(1,1,1,3),(m,m,m,3)] \in \{0\}^{(4)}$. Then, M repeats the following process from $j = 1$ to m; M records the input symbol $x[(1,1,1,1),(m,m,m,1)]$ in the finite control and checks that two symbols $x[(1,1,1,1),(m,m,m,1)] \neq x[(1,1,1,2),(m,m,m,2)]$. If so, M enters an accepting state. It is clear that $T(M) = T_2$.

(2) Suppose that there exists a $SV4\text{-}SUTM(L(m))$ M accepting T_2, where $L(m) = o(m^3)$. By using the technique of counting argument[14,44], we can get the desired result. □

Lemma 4.2. *Let* $T_3 = \{x \in \{0,1\}^{2m\times 2m\times 2m\times 2m} \mid m \geq 1, \ \& \ x[(1,1,1,1),(2m,2m,2m,m)] \neq x[(1,1,1,m+1),(2m,2m,2m,2m)]\}$ *(See Fig.4.). Then,*

(1) $T_3 \in \mathcal{L}[4\text{-}DTM(\log m)]$ $(= \mathcal{L}[4\text{-}SUTM(\log m)[1]])$, *and*

(2) $T_3 \notin \mathcal{L}[SV4\text{-}SUTM(L(m))]$ *for any* $L : \mathbf{N} \to \mathbf{N}$ *such that* $L(m) = o(m^4)$.

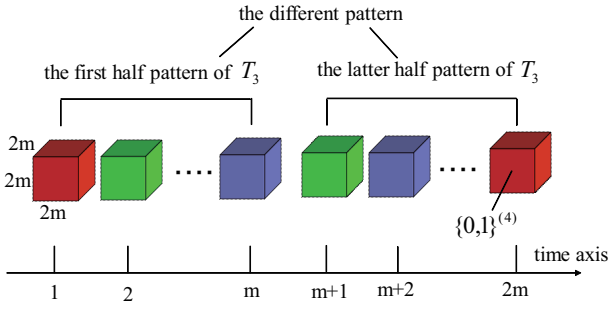

Fig. 4: A tape in T_3.

Proof: (1) We can construct a $4\text{-}DTM(\log m)$ M accepting T_3 as follows: Given x with $l_1(x) = l_2(x) = l_3(x) = l_4(x) = 2m$ ($m \geq 1$), starting on position (1,1,1,1) of x for all i_1,i_2,i_3,i_4 ($1 \leq i_1 \leq 2m$, $1 \leq i_2 \leq 2m$, $1 \leq i_3 \leq 2m$, $1 \leq i_4 \leq m$), M repeats the following process; M records the input symbol $x(i_1,i_2,i_3,i_4)$ in the finite control and checks that two symbols $x(i_1,i_2,i_3,i_4) \neq x(i_1,i_2,i_3,i_4+m)$. (This can be easily done by using $\log m$ cells of the storage tape.) If so, M and enters an accepting state. It is clear that $T(M) = T_3$.

(2) The idea is almost the same as in the proof of Lemma 4.1 (2). □

From Lemmas 4.1 and 4.2 we can get the following theorem.

Theorem 4.2. *Let* $L : \mathbf{N} \to \mathbf{N}$ *be a function such that (1)* $L(m) = o(m^3)$, *or (2)* $L(m) \geq \log m$ *and* $L(m) = o(m^4)$. *Then,*

$$\mathcal{L}[SV4\text{-}SUTM(L(m))] \subsetneq \mathcal{L}[4\text{-}SUTM(L(m))].$$

Corollary 4.1. $\mathcal{L}[SV4\text{-}SUFA] \subsetneq \mathcal{L}[4\text{-}SUFA]$.

It is easy to show that the following theorem holds.

Theorem 4.3. *For any function* $L(m) \geq m^4$,

$$\mathcal{L}[SV4\text{-}SUTM(L(m))] = \mathcal{L}[4\text{-}SUTM(L(m))].$$

5 Nondeterminism versus Synchronized Alternation

This section investigates a relationship between the accepting powers of four-dimensional synchronized alternating machines and four-dimensional nondeterministic machines.

Let $L : \mathbf{N} \to \mathbf{N}$ be a function. The function L is said to be *three-dimensionally fully space constructible* if there is a $4\text{-}DTM$ which for any input tape x with $l_1(x) = l_2(x) = l_3(x) = l_4(x) = m$ ($m \geq 1$) makes use of exactly $L(m)$ cells of the storage tape and halts.

Theorem 5.1. *For any function* $L(m) \geq \log m$,

$$\mathcal{L}[4\text{-}SATM(L(m))] = U_{c\geq 0}\mathcal{L}[4\text{-}NTM(m^4c^{L(m)})].$$

Proof: We first show that $\mathcal{L}[4\text{-}SATM(L(m))] \subseteq U_{c>0}\mathcal{L}[4\text{-}NTM(m^4c^{L(m)})]$. Given a $4\text{-}SATM(L(m))$ M, we can construct a $4\text{-}NTM(m^4c^{L(m)})$ M' to simulate M by doing a breadth-first-like traversal of the computation tree of M on input x of sidelength m. Each process of M is simulated until it enters an s-state; M' will compare the corresponding s-states to make sure that no deadlock occurs before continuing the simulation. Since there are at most $m^4d^{L(m)}$ distinct

configurations of M on an input x of sidelength m, M' needs at most $m^4e^{L(m)}$ space, for some constants d and e, at any time to maintain the current ID's of all processes of M on x. Then on any input x of sidelength m, M uses at most $L(m)$ space iff M' uses at most $m^4e^{L(m)}$ space.

On the other hand, by using same idea described in Lemma 3.4 in [17], we can show that $\bigcup_{c\geq 0}\mathcal{L}[4\text{-}NTM(m^4c^{L(m)})] \subseteq \mathcal{L}[4\text{-}SATM(L(m))$]. □

Theorem 5.2. *For any integer $k \geq 1$,*

(1) $\mathcal{L}[4\text{-}SAFA[k]] = \mathcal{L}[4\text{-}NFA(k\text{-}heads)]$, and

(2) $\mathcal{L}[SV4\text{-}SAFA[k]] = \mathcal{L}[4\text{-}NFA(k\text{-}heads)]$.

Proof: We only prove (1). Given a 4-NFA(*k*-heads) M where $k \geq 1$, we can construct a 4-$SAFA[k]$ M'. Let $H_1, H_2, ..., H_k$ denote the input heads of M. These heads are simulated by a single input head of 4-$SAFA[k]$ M' in the following way. The computation of M' branches from the initial configuration in a universal manner into k processes. Note that, the initial configuration is the only universal configuration which occurs in the computation. The states of M' (in all processes of M') store the simulated state of M. If the stored state is an accepting (rejecting) state, then the state of M is also an accepting (rejecting) one. In the ith process, for any i ($1 \leq i \leq k$), the input head of M' is at the same position as H_i.

One step of M is simulated by two steps of M'. Besides the state of M the symbols scanned by all heads of M have to be known to M'. Every process in the computation of M' has only a part of the necessary information. The processes can share this information via the synchronization. The synchronizing element consists of k components and represents the symbols scanned by the input heads of M. The ith process, for $1 \leq i \leq k$, sets the ith component according to the symbol scanned by the input head of M'. The other components are set nondeterministically. The synchronization is successful only in the case when every process has correctly guessed the remaining components.

The next synchronization is necessary because of nondeterminism. One configuration of M has several potential successors. All of the processes of M' must agree on the next step of M (they must simulate the same successor of the currently simulated configuration). The synchronizing element represents the new state of M and the actions of the heads of M. The successful synchronization means that all processes choose the same element of the next move relation of M. After that, M' moves its heads and enters a new state in accordance with the synchronizing element (i.e., in the ith process (for $1 \leq i \leq k$), M' moves by its input head like M by H_i). It will be obvious that M' can simulate M.

Conversely, given a 4-$SAFA[k]$ M, we can construct a 4-NFA(k-heads) M' such that $T(M) = T(M')$. The proof is omitted here. □

We next investigate a relationship betwen the accepting powers of seven-way nondeterministic machines and seven-way synchronized machines with only universal states.

Theorem 5.3.

(1)$\mathcal{L}[SV4\text{-}SUFA[2]] - \mathcal{L}[SV4\text{-}NTM(o(m^4))] \neq \phi$,

(2) $\mathcal{L}[SV4\text{-}NFA] - \mathcal{L}[SV4\text{-}SUTM(o(m^3))] \neq \phi$, and

(3) $\mathcal{L}[SV4\text{-}NTM(\log m)] - \mathcal{L}[SV4\text{-}SUTM(o(m^4)] \neq \phi$.

Proof: (1) Let T_1 be the set described in Lemma 3.1. By using the technique of counting argument, we can show that $T_1 \notin \mathcal{L}[SV4\text{-}NTM(o(m^4))]$. (1) follows from this fact and Lemma 3.1 (1).

(2) Let T_2 be the set of described in Lemma 4.1. It is easy to see that $T_2 \in \mathcal{L}[SV4\text{-}NFA]$. (2) follows from this fact and Lemma 4.1 (2).

(3) Let T_3 be the set of described in Lemma 4.2. It is easy to see that $T_3 \in \mathcal{L}[SV4\text{-}NTM(\log m)]$. (3) follows from this fact and Lemma 4.2 (2). □

Corollary 5.1. *For any function $L(m) = o(m^4)$, $\mathcal{L}[SV4\text{-}SUTM(L(m))]$ and $\mathcal{L}[SV4\text{-}NTM(L(m))]$ are incomparable.*

6 Conclusion

In this paper, we introduced a four-dimensional synchronized alternating Turing machine and investigated basic several accepting powers. In this section, we conclude this paper by giving two open problems.

(1) For any function $L(m) \geq \log m$, $\mathcal{L}[4\text{-}ATM(L(m))] \subsetneq \mathcal{L}[4\text{-}SATM(L(m))]$? and $\mathcal{L}[SV4\text{-}ATM(L(m))] \subsetneq \mathcal{L}[SV4\text{-}SATM(L(m))]$?

(2) For any integer $k \geq 1$, $\mathcal{L}[4\text{-}SUFA[k]] \subsetneq \mathcal{L}[4\text{-}SUFA[k+1]]$? and $\mathcal{L}[SV4\text{-}SUFA[k]] \subsetneq \mathcal{L}[SV4\text{-}SUFA[k+1]]$?

References:

[1] A. Aho, J. E. Hopcroft, and J. D. Ullman, *"The Design and Analysis of Computer Algorithms"*, Addison-Wesley, Reading, Mass., 1974.

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

[3] M. Blum and C. Hewitt, "Automata on a two-dimensional tape", *Proceedings of the 8th IEEE Symposium on Switching and Automata Theory, Texas, USA*, pp.155-160, 1967.

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

[6] C. R. Deyer and A. Rosenfeld, "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.

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

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

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

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

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

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

[13] J. Hromkovič, "How to organize the commmunication among parallel proceses in alternating computations", *manuscript*, 1986.

[14] J. Hromkovič, J. Karhumäki, B. Rován, and A. Slobodová, "On the power of synchronization in parallel computations", *Technical Report, Comenius University, Bratislava, Czechoslovakia, Dept. of Theoretical Cybernetics and Institute of Computer Science*, 1989.

[15] J. Hromkovič, K. Inoue, B. Rovan, A. Slobodová, I. Takanami, and K. W. Wagner, "On the power of one-way synchronized alternating machines with small space", *International Journal of Foundations of Computer Science*, Vol.3, No.1, pp.65-79, 1992.

[16] O. H. Ibarra and R. T. Melson, "Some results concerning automata on two-dimensional tapes", *International Journal of Computer Mathematics*, Vol.4, Section A, pp.269-279, 1974.

[17] C. M. Ibarra and N. Q. Trân, "On space-bounded synchronizing alternating Turing machines",*Theoretical Computer Science*, Vol.99, pp243-264, 1992.

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

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

[20] K. Inoue and A. Nakamura, "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.

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

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

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

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

[25] A. Ito, K. Inoue, and I. Takanami, "Complexity of acceptance Problems for two-dimensional automata", *A Perspective in Theoretical Computer Science, Series in Computer Science*, Vol.16, pp.70-94, 1989.

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

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

[28] T. Ito and M. Sakamoto, "Some accepting powers of three-dimensional synchronized alternating Turing machines", *Journal of Engineering, Computing and Architecture, Scientific Journals International*, Issue 1, Vol.1, pp.1-11, 2007.

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

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

[31] T. Makino, H. Okabe ,S. Taniguchi, M. Sakamoto, and K.Inoue, "Three-dimensional multi-inkdot automata", *International Journal of Artificial Life and Robotics*, Vol.9, No.2, pp.99-101, 2005.

[32] H. Matsuno, "Multihead automata and alternation", *Ph.D. Thesis, Kyusyu University*, 1994.

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

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

[35] A. Nakamura and K. Aizawa, "On the recognition of properties of three-dimensional pictures", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.PAMI-7, pp.708-713, 1985.

[36] A. Nakamura and K. Aizawa, "Detection of interlocking components in three-dimensional digital pictures", *Information Sciences*, Vol.40, pp.143-153, 1986.

[37] A. Nakamura and A. Rosenfeld, "Three-dimensional connected pictures are not recognizable by finite-state acceptors", *Technical Report TR-566, Computer Science Center, University of Maryland*, 1991.

[38] H. Okabe, S. Taniguchi, T. Makino, S. Nogami, Y. Nakama, M. Sakamoto, and K. Inoue, "Computation for motion image processing by 4-dimensional alternating Turing machines", *Proceedings of the 7th World Multiconference on SCI, Orlando, USA*, pp.241-246, 2003.

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

[40] E. Peng and L. Li, "Modeling human skeleton proportions from monocular data", *WSEAS Transactions on Computers*, Issue 3, Vol.5, pp.681-687, 2006.

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

[42] A. Rosenfeld, *"Picture Languages−Foumal Models for Picture Recognition"*, Academic Press, New York, 1979.

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

[44] M. Sakamoto, I. Sakuramoto, K. Inoue, and I. Takanami, "A space lower-bound technique for two-dimensional alternating Turing machines (in Japanese)", *The Transactions of the IEICE D-I*, Vol.J74-DI, No.4, pp.311-314, 1991.

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

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

[47] M. Sakamoto, A. Ito, K. Inoue, and I. Takanami, "Simulation of three-dimensional one-marker automata by five-way Turing machines", *Information Sciences*, Vol.77, pp.77-99, 1994.

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

[49] M. Sakamoto, T. Okazaki, and K. Inoue, "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.

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

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

[52] 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*, Issue 5, Vol.3, pp.1651-1656, 2004.

[53] M. Sakamoto, S. Nogami, K. Inoue, and M. Kono, "A relationship between the accepting powers of three-dimensional finite automata and time-bounded bottom-up pyramid cellular acceptors with three-dimensional layers", *The Transactions of SCI*, Vol.17, No.10, pp.451-428, 2004.

[54] 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*, Issue 10, Vol.7, pp.1638-1647, 2008.

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

[56] A. N. Shah, "Pushdown automata on arrays", *Information Sciences*, Vol.25, pp.175-193, 1981.

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

[58] I. H. Sudborough, "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.

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

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

[61] H. Taniguchi, K. Inoue, and I. Takanami, "k-neigh-borhood template $\mathcal{A}$-type 2-dimensional bounded cellular acceptor (in Japanese)", *The Transactions of the IEICE*, Vol.J69-D, No.3, pp.291-301, 1986.

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

[63] Y. Uchida, T. Ito, H. Okabe, M. Sakamoto, H. Furutani, and M. Kono, "Four-Dimensional Multi-Inkdot Finite Automata", *WSEAS Transactions on Computers*, Issue 9, Vol.7, pp.1437-1446, 2008.

[64] Y. Yamamoto, K. Morita, and K. Sugata, "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.