Hardware Hierarchies and Recognizabilities of Four-Dimensional Synchronized Alternating Turing Machines

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

, 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-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: The recent advances in computer animation, motion image processing, robotics and so on prompted us to analyze computational complexity of four-dimensional pattern processing. Thus, the research of four-dimensional automata as a computational model of four-dimensional pattern processing has also been meaningful. From this viewpoint, we introduced a four-dimensional alternating Turing machine (4-*ATM*) operating in parallel. In this paper, we continue the investigations about 4-*ATM*'s, deal with a four-dimensional synchronized alternating Turing machine (4-*SATM*), and investigate some properties of 4-*SATM*'s which each sidelength of each input tape is equivalent. The main topics of this paper are:

- (1) hierarchies based on the number of processes of 4-SATM's, and
- (2) recognizability of connected pictures by 4-SATM's.

Key-Words: alternation, four-dimensional Turing machine, hierarchy, recognizability, synchronization

1 Introduction

The question of whether processing four-dimensional digital patterns is much difficult than two- or threedimensional ones is of great interest from the theoretical and practical standpoints[3,4]. In recent years, due to the advances in many application areas such as motion image processing, computer animation, and so forth, the study of four-dimensional pattern processing has been of crucial importance. Thus, the study of four-dimensional automata as the computatinal model of four-dimensional pattern prcessing has been meaningful. From this viewpoint, we introduced a four-dimensional alternating Turing machine (4-ATM)[18-33,44,53-58,63-67].

On the other hand, Alternating Turing machines were as a model of parallel computation[5,8,12,34-37,45,46,52]. Informally, an alternating Turing ma-

chine 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 [13-15,17] 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 pro-

Makoto Sakamoto, Ryoju Katamune, Tomoya Matsukawa, Hiroshi Furutani, Michio Kono, Satoshi Ikeda, Takao Ito, Yasuo Uchida, Tsunehiro Yoshinaga

cesses enter either an accepting state or a synchronizing state with the same synchronizing symbol. When this happens, all processes are allows 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.

In this paper, we continue the investigations about 4-ATM's, deal with a four-dimensional synchronized alternating Turing machine (4-SATM), and investigate some properties of 4-SATM's which each sidelength of each input tape is equivalent. 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 hierarchies based on the number of processes (hardware hierarchies) of fourdimensional synchronized alternating finite automata, and shows that for four-dimensional synchronized alternating finite automata, k + 1 processes are more powerful than k processes for any $k \ge 1$. Section 4 investigates recognizability of connected pictures by seven-way four-dimensional synchronized alternating Turing machines with only universal states, and shows that (1) the necessary and sufficient space for these machines to accept the complement of T_c (where T_c denotes the set of all the connected pictures) is m^3 , and (2) seven-way four-dimensional synchronized alternating finite automata can accept T_c . Finally, Section 5 concludes this paper by giving several open problems.

2 Preliminaries

Definition 2.1.

Let Σ be a finite set of symbols. A fourdimensional 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 *j*th 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 $1 \le i_j \le i'_j \le l_j(x)$ for each integer $j \ (1 \le j \le 4)$, as the four-dimensional input tape y satisfying the following;

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

(2) for each r_1 , r_2 , r_3 , r_4 $(1 \le r_1 \le l_1(y), 1 \le r_2 \le l_2(y), 1 \le r_3 \le l_3(y), 1 \le r_4 \le l_4(y)), y(r_1, r_2, r_3, r_4) = x(r_1+i_1-1, r_2+i_2-1, r_3+i_3-1, r_4+i_4-1).$ (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 synchro*nized alternating Turing machine.

Definition 2.2.

A four-dimensional synchronized alternating Turing machine (denoted by 4-SATM) is a 10-tuple $M = (Q, q_0, U, E, S, F, \Sigma, \Pi, \Gamma, \delta)$, 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) $\delta \subseteq (Q \times (\Sigma \cup \{\#\}) \times \Gamma) \times (Q \times (\Gamma - \{B\}) \times \{\text{east,west,south,north,up,down,future,past,no move}\} \times \{\text{left,right,no move}\}$ is the next-move relation.

M has a read-only four-dimensional input tape with boundary symbols #'s ($\# \notin \Sigma$) and one semiinfinite 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. 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 $M = (Q, q_0, U, E, S, F, \Sigma, \Pi, \Gamma, \delta)$ 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 $(0 \le i_1 \le l_1(x) + 1, 0 \le i_2 \le l_2(x) + 1, 0 \le i_3 \le l_3(x) + 1), 0 \le i_4 \le l_4(x) + 1)$, 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 $(1 \le k \le |\alpha| + 1)$. 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.

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, $I_0, I_1, \ldots, I_m (m \ge 0)$, 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].

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 vin 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 $\{I \mid I_u \vdash_M I\} = \{I_1, \ldots, I_m\}$, then u has exactly m children v_1, \ldots, v_m , such that $I_{v_i} = I_i, 1 \le i \le m$,

(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 sstates causing the deadlock are called *deadlock states*.

The longest synchronizing sequence of a node in the computation tree T is called the *synchronizing se*quence 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 $T(M) = \{x \in \Sigma^{(4)} \mid M \text{ accepts } x\}$.

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 [44].

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

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

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 pust direction.

Definition 2.6.

Let $L(m) : \mathbf{N} \to \mathbf{N}$ 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) spacebounded" 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 computation 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[9-11,38,40,61,62]. Four-dimensional alternating Turing machines (4-ATM's) and seven-way four-dimensional alternating Turing machines (SV4-ATM's)in [44] 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 By 4-ATM(L(m)) (SV4-ATM(L(m)), states. SV4-SUTM(L(m)),4-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) (sevenway four-dimensional nondeterministic Turing machine (SV4-NTM)) is a 4-ATM which has no universal states. We denote an L(m) spacebounded 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 fourdimensional 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 fourdimensional nondeterministic finite automaton, fourdimensional deterministic finite automaton, sevenway four-dimensional deterministic finite automa-That is, a 4-SAFA (SV4-SAFA, 4ton)[56]. 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 $k \ge$ 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 $k \ge$ 1, 4-NFA(k-heads) (4-DFA(k-heads)) is used to denote a 4-NFA (4-DFA) which has k input heads. For any machine class C, let

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

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

3 Hierarchy Based on the Number of Processes

It is shown in [30] that for two-dimensional alternating finite automata, k+1 processes are more powerful than k processes for any $k \ge 1$. This section shows that a similar result holds also for four-dimensional synchronized alternating finite automata.

We will need the following operation ρ mapping one-dimensional words over an alphabet Σ to fourdimensional input tapes which each sidelength of each input tape is equivalent over $\Sigma \times \Sigma \times \Sigma \times \Sigma$. This operation was first introduced in [47]. Let $w = a_1 a_2 \cdots a_n$ be a word of length n. Then $\rho(w) = x$ where $x(i, j, k, l) = (a_i, a_j, a_k, a_l)$ for $1 \le i \le n, 1 \le j \le n, 1 \le k \le n$ and $1 \le l \le n$. Thus a symbol of x in a certain row, column, plane and cube has the corresponding symbol of w in the first, second, third, and fourth component, respectively. A word $w = a_1 a_2 \cdots a_n$ is mapped to

This operation is extended in the usual way to languages.

For each k > 1, let 1-NFA(k-heads) denote a one-dimensional two-way nondeterministic k-heads finite automaton [39]. **Lemma 3.1.** For each $k \ge 1$, a one-dimensional language L is accepted by a 1-NFA(4k-heads) if ad only if $\rho(L) \in \mathcal{L}$ [4-NFA(k-heads)].

Proof: We only prove the lemma for the case of k = 1. The 1-*NFA*(4-*heads*) simulates the 4-*NFA* by storing the information of row, column, plane and cube in its head positions. It assembles the quartet from the symbols read by the heads.

Conversely the 4-NFA verifies that the first (second, third,fourth) components of input symbols within every row (column, plane, cube) agree. Then the 4-NFA starts a step by step simulation by storing the first head-position as the current row number, the second head-position as the current column number, the third head-position as the current plane number, and the fourth head-position as the current cube number, respectively. The currently scanned symbols are available as the components of the symbol from Σ read by the head.

It is shown in [39] that the following lemma holds.

Lemma 3.2. For each $k \ge 1$, $\mathcal{L}[1-NFA(k-heads)] \subseteq \mathcal{L}[1-NFA((k+1)-heads)].$

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

Theorem 3.1. For any integer $k \ge 1$,

 $\mathcal{L}[4\text{-}NFA(k\text{-}heads)] \subseteq \mathcal{L}[4\text{-}NFA((k+1)\text{-}heads)].$

Proof: Let us suppose that

 $L \in \mathcal{L}[1-NFA((4k+4)-heads)] - \mathcal{L}[1-NFA(4k-heads)] \cdots (1).$

Then we have $\rho(L) \in \mathcal{L}[4-NFA((k + 1)-heads)]$ from Lemma 3.1. Now, we assume that $\rho(L) \in \mathcal{L}[4-NFA(k-heads)]$. Then, we would have $L \in \mathcal{L}[1-NFA(4k-heads)]$ from Lemma 3.1. This contradicts (1), and thus we have $\rho(L) \notin \mathcal{L}[4-NFA(k-heads)]$. This completes the proof of the theorem. \Box

From Theorems 5.2 (1) of [31] and Theorem 3.1, we have

Corollary 3.1.

For any integer $k \geq 1$,

 $\mathcal{L}\left[4\text{-}SAFA[k]\right] \subsetneq \mathcal{L}\left[4\text{-}SAFA[k+1]\right].$

4 Recognizability of Connected Pictures

There have been many interesting investigations on digital geometry [2,6,7,16,41,48-51]. These works form the theoretical foundation of digital image processing. Among them, the problem of recognizability of connectedness is one of the most interesting topics. In [59,60,68], Yamamoto, Morita, and Sugata showed that a three-dimensional nondeterministic one-marker automaton can recognize connected tapes. In the case of L(m) space-bounded five-way three-dimensional deterministic Turing machine, they proved that space $m^2 \log m$ is necessary and sufficient among for recognizing connected tapes of size $m \times m \times m$. In [43], Nakamura and Rosenfeld showed that three-dimensional connected tapes are not recognizable by any three-dimensional deterministic or nondeterministic finite automaton. By the way, it is well known that two-dimensional digital pictures have 4- and 8-connectedness, and three-dimensional digital pictures have 6- and 26-connectedness. It is also known that various topological properties can be defined by making use of these connectedness. For example, Nakamura and Aizawa proposed a new topological property of three-dimensional digital pictures - the interlocking component which is a chainlike connectivity. They showed that three-dimensional deterministic one-marker automaton can not detect interlocking components in a three-dimensional digital picture[42], After that, we have investigated recognizabilities of automata on four-dimensional input tapes, and showed some properties. This section investigates the recognizability of connected tapes by SV4-SAFA's and SV4-SUTM's.

Definition 4.1.

Let x be in $\{0,1\}^{(4)}$. A maximal subset, P of N^4 satisfying the following conditions is called a 1-*component* of x.

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

(2)For any (i_1,i_2,i_3,i_4) , $(i'_1,i'_2,i'_3,i'_4) \in P$, there exists a sequence $(i_{1,0}, i_{2,0}, i_{3,0},i_{4,0})$, $(i_{1,1}, i_{2,1}, i_{3,1},i_{4,1}),\ldots$, $(i_{1,n}, i_{2,n}, i_{3,n}, i_{4,n})$ of elements in Psuch that $(i_{1,0}, i_{2,0}, i_{3,0}, i_{4,0}) = (i_1, i_2, i_3,i_4)$, $(i_{1,n}, i_{2,n}, i_{3,n}, i_{4,n}) = (i'_1,i'_2,i'_3,i'_4)$, and $|i_{1,j} - i_{1,j-1}| + |i_{2,j} - i_{2,j-1}| + |i_{3,j} - i_{3,j-1}| + |i_{4,j} - i_{4,j-1}| \leq 1$ $(1 \leq j \leq n)$. A tape $x \in \{0, 1\}^4$ is called *connected* if there exists exactly one 1-compnent of x. We denote the set of all the four-dimensinal connected tapes by T_c . It is shown in [44] that a 4-ATM can accept T_c . From this fact and from the fact $\mathcal{L}[SV4\text{-}SAFA] = \mathcal{L}[4\text{-}SAFA] \supseteq \mathcal{L}[4\text{-}AFA]$ by using a technique similar to that in Ref.[4], the following theorem holds.

Theorem 4.1. $T_c \in \mathcal{L}[SV4\text{-}SAFA]$.

It is shown in [44] that log m space is necessary and sufficient for SV4-ATM's to accept T_c . We below show the necessary and sufficient space for SV4-SUTM's to accept \overline{T}_c (=the complement of T_c).

Theorem 4.2. m^3 space is necessary and sufficient for FV3-SUTM's to accept \overline{T}_c .

Proof: (The proof of sufficiency) It is shown in [50] that T_c is accepted by a deterministic one-way parallel/sequential array acceptor (DOWPS), and it is shown in [25] that $\mathcal{L}[\text{DOWPS}] = \mathcal{L}[TR2-DTM(m)]$ (TR2-DTM(m)) means m space-bounded three-way two-dimensional deterministic Turing machine). From these facts and the fact [22,23] that $\mathcal{L}[TR2-DTM(m)]$ is closed under complementation, it follows that \overline{T}_c is in $\mathcal{L}[TR2-DTM(m)]$, and thus in $\mathcal{L}[TR2-SUTM(m)]$. By applying the same idea of such a two-dimensional case, we can easily get the fact that \overline{T}_c is in $\mathcal{L}[SV4-SUTM(m^3)]$.

(The proof of necessity) Suppose that there is an SV4-SUTM(L(m)) M accepting \overline{T}_c , where $L(m) = o(m^3)$. We assume without loss of generality that M enters an accepting state only on the bottom boundary.

Let

$$\begin{split} T_c' &= \{ x \in \{0,1\}^{(4m+1,4m+1,4m+1,4m+1)} \mid m \geq 1 \\ \& \; \forall_{i_1}(1 \leq i_1 \leq m+1) \; \forall_{i_2}(1 \leq i_2 \leq 2m+1) \; \forall_{i_3}(1 \leq i_3 \leq 4m+1) [x[(2i_2-1,1,2i_1-1,i_3),(2i_2-1,4m-2i_1+3,2i_1-1,i_3)],x[(2i_2-1,4m-2i_1+3,i_3)],x[(2i_2-1,4m-2i_1+3,4m-2i_1+3,i_3)],x[(2i_2-1,4m-2i_1+3,2i_1-1,i_3),(2i_2-1,4m-2i_1+3,4m-2i_1+3,i_3)] \in \{1\}^{(3)}] \; \& \; \forall_{i_2}(1 \leq i_2 \leq 2m) \; \forall_{i_3}(1 \leq i_3 \leq 4m+1) [\; x[(2i_2,1,2m+1,i_3),(2i_2,2m+1,2m+1,i_3)] \in \{1\}^{(3)}] \& \; \forall_{i_1}(1 \leq i_1 \leq 2m) \; \forall_{i_2}(1 \leq i_2 \leq 2m+1) \forall_{i_3}(1 \leq i_3 \leq 4m+1) [\; x(2i_2-1,1,2i_1,i_3)] = x(2i_2-1,1,4m-2i_1+2,i_3)] \; \& (\text{the other part of x consists of 0's) }, \end{split}$$

where we define $\overline{0} = 1$ and $\overline{1} = 0$. Furter, for each $w \in \{0,1\}^{(3)}, \overline{w}$ denotes the tape w' such that (i) $l_j(w) = l_j(w'), 1 \le j \le 4$, and (ii) $w'(i_1, i_2, i_3, i_4) = w(i_1, i_2, i_3, i_4)$ for each $i_1, i_2, i_3, i_4[1 \le i_1 \le l_1(w'), 1 \le i_2 \le l_2(w'), 1 \le i_3 \le l_3(w'), 1 \le i_4 \le l_4(w')]$ (See Fig. 1.).

Clearly $T'_c \subseteq T_c$. Let s and t be the numbers of states (of the finite control) and storage tape symbols of M, respectively. For each $m(m \ge 1)$, let

$$V(m) = \{ x \in T'_c \mid l_1(x) = l_2(x) = l_3(x) = l_4(x) = 4m+1 \}.$$



Fig. 1: A tape in T'_c .

For each x in V(m), let S(x) and C(x) be sets of configurations of M defined as follows:

 $S(x) = \{((i_1, i_2, 2m+1, i_3), (q, \alpha, k)) | \text{there exists a computation path } I_M(x) \vdash_M^* (x, ((i_1, i_2, 2m, i_3), (q', \alpha', k'))) \vdash_M (x, ((i_1, i_2, 2m+1, i_3), (q, \alpha, k))) \text{ of } M \text{ on } x \text{ (that is, } (x, ((i_1, i_2, 2m+1, i_3), (q, \alpha, k))) \text{ is an } ID \text{ of } M \text{ just after the point where the input head left the } (2m+1)\text{th plane of each cube of } x \},$

 $C(x) = \{\{\rho_1, \rho_2\} \mid \rho_1 \text{ and } \rho_2 \text{ are configurations }$ in S(x) such that

(1) in case of $\rho_1 = \rho_2$, there exists a sequential computation of M which starts with $ID(x,\rho_1)$ and either terminates in a rejecting ID, or enters an infinite loop, and

(2) in case of $\rho_1 \neq \rho_2$, there exist two sequential computations of M which start with ID's (x,ρ_1) and (x,ρ_2) , respectively, and terminate in sync ID's with different sync elements}.

(Note that, for each x in V(m), C(x) is not empty, since x is not in \overline{T}_c , and so not accepted by M.) Then the following proposition must hold.

Proposition 4.1. For any two different tapes $x, y \in V(m)$,

$$C(x) \cap \mathbf{C}(y) = \phi.$$

[Proof: For otherwise, suppose that $x \neq y(x,y \in V(m))$, $C(x) \cap C(y) \neq \phi$, and $\{\rho_1, \rho_2\} \in C(x) \cap C(y)$. Let z (with $l_1(x)=l_2(x)=l_3(x)=l_4(x)=4m+1$) be the tape such that

 $(1)z \ [(1,1,1,1), \ (4m+1,4m+1,4m+1,2m+1)] = x[(1,1,1,1), (4m+1,4m+1,4m+1,2m+1)], and$

 $\begin{array}{l} (2)z \ [(1,1,1,2m+2), \ (4m+1,4m+1,4m+1,4m+1)] \\ = y[(1,1,1,2m+2), \ (4m+1,4m+1,4m+1,4m+1)]. \end{array}$

Since $\{\rho_1,\rho_2\} \in C(x)$, there exist computation paths $I_M(z) \vdash_M^* (z,\rho_1)$ and $I_M(z) \vdash_M^* (z,\rho_2)$. Since $\{\rho_1,\rho_2\} \in C(y)$, in case of $\rho_1=\rho_2$, there exists a sequential computation of M which starts with the $ID(z,\rho_1)$ and either terminates in a rejecting ID, or enters an infinite loop, and in case of $\rho_1 \neq \rho_2$, there exist two sequential computations of M which start with ID's (z, ρ_1) and (z, ρ_2) , respectively, and terminate in sync ID's with different sync elements. This means that z is not accepted by M. This contradicts the fact that z is in $\overline{T}_c = T(M)$.

Proof of Theorem 4.2(*continued*): Let p(m) denote the number of pairs of possible configurations of M just after the point where the input head left the (2m+1)th cube of tapes in V(m). Then

$$p(m) = \binom{K}{2} + K$$

where $K = s(4m + 3)^3 L(4m+1)t^{L(4m+1)}$. On the other hand, $|V(m)| = 2^{m(2m+1)(4m+1)}$. Since L(m) = o(m), we have $|V(m)| \ge p(m)$ for large m. Therefore, it follows that for large m there must be two different tapes x, y in V(m) such that $C(x) \cap$ $C(y) \ne \phi$. This contradicts Proposition 4.1 and completes the proof of necessity. \Box

5 Conclusion

This paper dealt with two topics concerning 4-SATM's. We mainly investigated about hierarchies based on the number of processes of 4-SATM's and recognizability of connected pictures by 4-SATM's, and showed some properties of 4-SATM's.

In this section, we conclude this paper by giving several open problems.

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

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

$$(3)T_c \in \mathcal{L}[4\text{-}SUFA]?$$
 and $T_c \in \mathcal{L}[SV4\text{-}SUFA]?$

References:

- 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 twodimensional tape", *Proceedings of IEEE Symposium on Switching and Automata Theory*, 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 The*ory, 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 communication among parallel processes 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, and 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 spacebounded synchronizing alternating Turing machines", *Theoretical Computer Science*, Vol.99, pp.243-264,1992.
- [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, "Three-way tapebounded two-dimensional Turing machines", *Information Sciences*, Vol.17, No.3, pp.195-220,1979.
- [23] K. Inoue, I. Takanami, and H. Taniguchi, "Three-way two-dimensional simple multihead finite automata -Closure properties-", *The Transactions of the IECE*, Vol.J54-D, No.4, pp.273-280, 1979.
- [24] K. Inoue and I. Takanami, "A note on bottomup pyramid acceptors", *Information Processing Letters*, Vol.8, No.1, pp.34-37, 1979.
- [25] K. Inoue and I. Takanami, "A note on deterministic three-way tape-bounded two-dimensional Turing machines", *Information Sciences*, Vol.20, pp.41-55,1980.
- [26] K. Inoue and I. Takanami, "A survey of twodimensional automata theory", *Information Sciences*, Vol.55, pp.99-121, 1991.

- [27] A. Ito, K. Inoue, I. Takanami, and H. Taniguchi, "Two-dimensional alternating Turing machines with only universal states", *Information and Control*, Vol.55, No.1-3, pp.193-221, 1982.
- [28] 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.
- [29] 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.
- [30] A. Ito, K. Inoue, I. Takanami, and Y. Inagaki, "Constant leaf-size hierarchy of twodimensional alternating Turing machines", *International Journal of Pattern Recognization and Artificial Intelligence.*, Vol.8, No.2, pp.509-524,1994.
- [31] T. Ito, M. Sakamoto, M. Saito, K.Iihoshi, H. Furutani, M. Kono, and K.Inoue, "Threedimensional synchronized alternating Turing machines", *Proceedings of the 11th International Symposium on Artificial Life and Robotics, Oita, Japan*, OS 9-4, 2006.
- [32] 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.
- [33] 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.
- [34] K. N. King, "Measures of parallelism in alternating computation trees", *Proceedings of the 13th ACM Symposium on Theory of Computing*, pp.189-201, 1981.
- [35] K. N. King, "Alternating multihead finite automata", *Theoretical Computer Science*, Vol.61, pp.149-174, 1985.
- [36] 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.
- [37] 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.

- [38] P. Michel, "A survey of space complexity", *Theoretical Computer Science*, Vol.101, pp.99-132, 1992.
- [39] B. Monien, "Two-way multihead automata over a one-leter alphabet", *RAIRO Information Theory*, Vol.14, pp.67-82, 1980.
- [40] K. Morita, "Computational complexity in oneand two-dimensional tape automata", *Ph.D. Thesis, Osaka University*, 1978.
- [41] 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.
- [42] A. Nakamura and K. Aizawa, "Detection of interlocking components in three-dimensional digital pictures", *Information Sciences*, Vol.40, pp.143-153, 1986.
- [43] A. Nakamura and A. Rosenfeld, "Threedimensional connected pictures are not recognizable by finite-state acceptors", *Technical Report TR-566, Computer Science Center, University of Maryland*, 1991.
- [44] H. Okabe, S. Taniguchi, T. Makino, S. Nogami, Y.Nakama, M.Sakamoto, and K.Inoue, "Computation for motion image processing by 4dimensional alternating Turing machines", *Proceedings of the 7th World Multiconference on SCI, Orland, USA*, pp.241-246, 2003.
- [45] W. J. Paul, E. J. Prauss, and R. Reischuk, "On alternation", *Acta Informatica*, Vol.14, pp.243-255, 1980.
- [46] W. J. Paul, E. J. Prauss, and R. Reischuk, "On alternation", *Acta Informatica*, Vol.14, pp.391-403, 1980.
- [47] H. Petersen, "Some results concerning twodimensional Turing machines and finite automata", *Lecture Notes on Computer Science* 965, *FCT*'95, pp.394-382, 1995.
- [48] A. Rosenfeld and D. L. Milgram, "Parallel / sequential array automata", *Information Processing Letters*, Vol.2, pp.43-46, 1973.
- [49] A. Rosenfeld and A. C. Kak, "*Digital Picture Processing*", Academic Press, New York, 1976.
- [50] A. Rosenfeld, "Picture Languages (Formal Models for Picture Recognition)", Academic Press, New York, 1979.
- [51] A. Rosenfeld, "Three-dimensional digital topology", *Information and Control*, Vol.50, pp.119-127, 1981.
- [52] W. L. Ruzzo, "Tree-size bounded alternation", Journal of Computer and System Sciences, Vol.21, pp.218-235, 1980.

- [53] M. Sakamoto, K. Inoue, and I. Takanami, "A note on three-dimensional alternating Turing machines with space smaller than log m", *Information Sciences* 72, pp.225-249, 1993.
- [54] M. Sakamoto, T. Okazaki, and K. Inoue, "Threedimensional multicounter automata", *Proceedings of the 6th International Workshop on Parallel Image Processing and Analysis—Theory and Applications*, pp.267-280, 1999.
- [55] M. Sakamoto, "Three-dimensional alternating Turing machines", *Ph.D. Thesis, Yamaguchi University*, 1999.
- [56] 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.
- [57] 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.
- [58] 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.
- [59] A. N. Shah, "Pebble automata on arrays", *Computer Graphics Image Processing*, Vol.3, pp.236-246, 1974.
- [60] A. N. Shah, "Pushdown automata on arrays", *In-formation Sciences*, Vol.25, pp.175-193, 1981.
- [61] R. E. Stearns, J. Hartmanis, and P. M. Lewis , "Hierarchies of memory limited computations", *Proceedings of the IEEE Conference on Switching Circuit Theory and Logical Design*, pp.179-190, 1965.
- [62] 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.
- [63] A. Szepietowski, "On three-way twodimensional multicounter automata", *Information Sciences*, Vol.55, pp.35-47, 1991.
- [64] H. Taniguchi, K. Inoue, and I. Takanami, "A note on three-dimensional finite automata", *Information Sciences*, Vol.26, pp.65-85, 1982.
- [65] H. Taniguchi, K. Inoue, and I. Takanami, "kneigh-borhood template A-type 2-dimensional bounded cellular acceptor (in Japanese)", *The Transactions of the IEICE*, Vol.J69-D, No.3, pp.291-301, 1986.

- [66] 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.
- [67] Y. Uchida, T. Ito, H. Okabe, M. Sakamoto, H. Furutani, M. Kono, "Four-Dimensional Multi-Inkdot Finite Automata", WSEAS Transactions on Computers, Issue 9, Vol.7, pp.1437-1446, 2008.
- [68] 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.