Some Results about Hierarchy and Recognizability of Four-Dimensional Synchronized Alternating Turing Machines

MAKOTO SAKAMOTO, RYOJU KATAMUNE, TOMOYA MATSUOKA, HIROSHI FURUTANI, MICHIKO KONO and SATOSHI IKEDA
Department of Computer Science and Systems Engineering
University of Miyazaki
1-1, Gakuen Kibanadai Nishi, Miyazaki City
JAPAN 889-2192
sakamoto@cs.miyazaki-u.ac.jp

TAKAO ITO and YASUO UCHIDA
Ube National College of Technology
Department of Business Administration
2-14-1, Tokiwadai, Ube City
JAPAN 755-8555
ito@ube-k.ac.jp

TSUNEHIRO YOSHINAGA
Department of Computer Science and Electronic Engineering
Tokuyama College of Technology
Gakuendai, Shunan City
JAPAN 745-8585
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 and Preliminaries

The question of whether processing four-dimensional digital patterns is much difficult than two- or three-dimensional ones is of great interest from the theoretical and practical standpoints. 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 research of four-dimensional automata as the computational model of four-dimensional pattern processing has been meaningful. From this viewpoint, we introduced a four-dimensional alternating Turing machine (4-ATM)[6,11,12].

On the other hand, synchronized alternating Turing machines were introduced in [1] 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 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 computational 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. Moreover, this
section summarizes the formal definitions and notations necessary for this paper. Section 2 investigates hierarchies based on the number of processes of four-dimensional 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 \( k \geq 1 \). Section 3 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 \( T_\delta \) (where \( T_\delta \) denotes the set of all the connected pictures) is \( m^3 \), and (2) seven-way four-dimensional synchronized alternating finite automata can accept \( T_\delta \). Finally, Section 4 concludes this paper by giving several open problems.

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

1. For each \( j \) (1 \( \leq j \leq 4 \)), \( l_j(y) = l_j + 1 \);
2. For each \( r_1, r_2, r_3, r_4 \) (1 \( \leq r_1 \leq l_1(y) \), \( 1 \leq r_2 \leq l_2(y) \), \( 1 \leq r_3 \leq l_3(y) \), \( l_4(y) \leq r_4 \leq l_4(y) \)), \( x(r_1,r_2,r_3,r_4) = x(r_1+1,r_2+1,r_3+1,r_4+1) \). (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**. 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. \( \Sigma \) is a finite input alphabet \(# \notin \Sigma \) is the boundary symbol,
8. \( \Pi \) is a finite alphabet of synchronizing symbols,
9. \( \Gamma \) is a finite storage tape alphabet containing the special blank symbol \( B \).
10. \( \delta \subseteq (Q \times (\Sigma \cup \{\#\}) \times \Gamma) \times (Q \times (\Gamma \setminus \{B\}) \times \{east, west, south, north, up, down, future, past, no move\}) \) is the next-move relation.

\( M \) 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. 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 \( \delta \). 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.

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 \( C_M = (N \cup \{0\})^4 \times S_M, S_M = Q \times (\Gamma \setminus \{B\})^* \times N \), where \( N \) denotes the set of all positive integers. The first component of an ID \( I = (x,(i_1,i_2,i_3,i_4),(q_0,\lambda,k)) \) 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 \((q_0,\lambda,k)\) of the second component of \( I \) represents the state of the finite control, nonblank contents of the storage tape, and the storage head position \( 1 \leq k \leq |\lambda| + 1 \). 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 \( \lambda \) is the null word.

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 \( \delta \). 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 \( M \), \( I_0, I_1, \ldots, I_m(m \geq 0) \), 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 \( x \).

The full computation tree of \( M \) on an input tape \( x \) is a (possibly infinite) labeled tree \( \vdash^*_M \) such that

1. each node \( \vdash^*_M \) is labeled by some ID \( I_v \) of \( M \),
2. the root is labeled by \( I_M(x) \),
3. the children of \( I \) are labeled by all \( \vdash_M I \) that can be reached from \( I \).

**RECENT ADVANCES** in **APPLIED MATHEMATICS**
(3) $v_2$ is a direct descendant of $v_1$ iff $I_{v_1} \vdash_M I_{v_2}$. (Each branch of $\vdash_M$ is called a process.) 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 occurring 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 $T^M_x$ 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 $I_{v_i} = I_i$, $1 \leq i \leq m$,

(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 $T^M_x$ 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 $I_D$. 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 [6].

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 past direction.

Let $L(m) : N \to N$ be a function with one variable $m$. With each 4-SATM (or SV4-SATM) $M$ we associate a space complexity function $\text{SPACE}$ which takes $I_D$'s to natural numbers. That is, for each $ID I = (x_i, ((i_1, i_2, i_3, i_4), (q, \alpha, k)))$, let $\text{SPACE}(I)$ be the length of $\alpha$. 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 computation tree of $M$ on input $x$ such that for each node $\pi$ of the tree, $\text{SPACE}(I(\pi)) \leq L(m)$. 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.

Four-dimensional alternating Turing machines (4-ATM's) and seven-way four-dimensional alternating Turing machines (SV4-ATM's) in [6] 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-UTM) which has no existential states. By 4-ATM($L(m)$) (SV4-ATM($L(m)$), SV4-SATM($L(m)$), SV4-SUTM($L(m)$)), we denote an $L(m)$ space-bounded 4-ATM (SV4-ATM, 4-SUTM, SV4-SATM, 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 $I_D$'s each have at most one successor, and a four-dimensional non-deterministic Turing machine (4-NTM) (seven-way four-dimensional non-deterministic 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 alternating finite automaton, four-dimensional non-deterministic finite automaton, seven-way four-dimensional non-deterministic finite automaton, seven-way four-dimensional deterministic finite automaton)[15]. 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, SV4-NTM, SV4-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 ≥ 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

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

Thus, for example, \( L[4-SATM(L(m))] \) denotes the class of sets accepted by 4-SATM(L(m))’s.

2 Hierarchy Based on the Number of Processes

It is shown in [3] that for two-dimensional alternating finite automata, \( k + 1 \) processes are more powerful than \( k \) processes for any \( k \geq 1 \). This section shows that a similar result holds also for four-dimensional synchronized alternating finite automata.

We will need the following operation \( \rho \) mapping one-dimensional words over an alphabet \( \Sigma \) to four-dimensional 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 [7]. Let \( w = a_1a_2\cdots a_n \) 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 \( 1 \leq i \leq n, 1 \leq j \leq n, 1 \leq k \leq n \) and \( 1 \leq l \leq n \). 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 \( w = a_1a_2\cdots a_n \) is mapped to

\[
\begin{align*}
(a_1, a_1, a_1, a_1) & (a_1, a_2, a_1, a_1) \cdots (a_1, a_n, a_1, a_1) \\
(a_2, a_1, a_1, a_1) & (a_2, a_2, a_1, a_1) \cdots (a_2, a_n, a_1, a_1) \\
& \cdots \\
(a_{n-1}, a_1, a_n, a_n) & (a_{n-1}, a_2, a_n, a_n) \cdots (a_{n-1}, a_n, a_n, a_n) \\
(a_n, a_1, a_n, a_n) & (a_n, a_2, a_n, a_n) \cdots (a_n, a_n, a_n, a_n)
\end{align*}
\]

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 [5].

**Lemma 2.1.** For each \( k \geq 1 \), a one-dimensional language \( L \) is accepted by a 1-NFA(4k-heads) if and only if \( \rho(L) \in L[4-NFA(k-heads)] \).

**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 \( \Sigma \) read by the head. \( \square \)

It is shown in [5] that the following lemma holds.

**Lemma 2.2.** For each \( k \geq 1 \), \( L[1-NFA(k-heads)] \subseteq L[1-NFA((k+1)-heads)] \).

From Lemmas 2.1 and 2.2, we can get the following theorem.

**Theorem 2.1.** For any integer \( k \geq 1 \),

\[ L[4-NFA(k-heads)] \subseteq L[4-NFA((k+1)-heads)]. \]

**Proof:** Let us suppose that

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

Then we have \( \rho(L) \in L[4-NFA((k+1)-heads)] \) from Lemma 2.1. Now, we assume that \( \rho(L) \in L[4-NFA(k-heads)] \). Then, we would have \( L \in L[1-NFA(4k-heads)] \) from Lemma 2.1. This contradicts (1), and thus we have \( \rho(L) \notin L[4-NFA(k-heads)] \). This completes the proof of the theorem. \( \square \)

From Theorems 5.2 (1) of [4] and Theorem 2.1, we have

**Corollary 2.1.** For any integer \( k \geq 1 \),

\[ L[4-SAFA[k]] \subseteq L[4-SAFA[k+1]]. \]

3 Recognizability of Connected Pictures

There have been many interesting investigations on digital geometry [8-10]. 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. This section investigates the recognizability of connected tapes by SV4-SAFA’s and SV4-SUTM’s.
Definitions 3.1. Let $x$ be in $\{0, 1\}^4$. A maximal subset, $P$ of $N^4$ satisfying the following conditions is called a 1-component of $x$.

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

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

It is shown in [6] that a 4-ATM can accept $T_c$. From this fact and from the fact $\mathcal{L}[SV4-SAFA] = \mathcal{L}[4-SAFA] \supset \mathcal{L}[4-AFA]$ by using a technique similar to that in [2], the following theorem holds.

Theorem 3.1. $T_c \in \mathcal{L}[SV4-SAFA]$.

It is shown in [9] that log $m$ space is necessary and sufficient for SV-4-ATM’s to accept $T_c$. We show the necessary and sufficient space for SV4-SUTM’s to accept $T_c$ (=the complement of $T_c$).

4 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 $L(m) \geq \log m$, $\mathcal{L}[4-ATM(L(m))] \subseteq \mathcal{L}[4-SATM(L(m))]$?

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

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

References:


