Leaf-Size Hierarchy of Four-Dimensional Alternating Turing Machines

Kouichi Iihoshi[1], Makoto Sakamoto[1], Takao Ito[1], Naoko Tomozoe[1], Yuusuke Nagamizu[1], Hiroshi Furutani[1], Michio Kono[1], Satoshi Ikeda[1], Tatsuhiro Tamaki[2], and Katsushi Inoue[3],

[1]Dept. of Computer Science and Systems Engineering, Miyazaki University, Miyazaki 889-2192, JAPAN

[2]Dept. of Business Administration, Ube National College of Technology, Ube 755-8555, JAPAN

[3]Dept. of Computer Science and Systems Engineering, Yamaguchi University, Ube 755-8611, JAPAN

Abstract

The recent adurances in computer animation, motion image processing and so on prompted us to analyze computational complexity of multi-dimensional information processing to explicate the properties of four-dimensional automata, i.e., three-dimensional automata with the time axis. From this point of view, we first introduced four-dimensional alternating Turing machines 4-ATM's, and investigated leaf-size bounded computation for 4-ATM's in [4,6]. In this paper, we continue the investigations about 4-ATM's, and maily investigate leaf-size bounded computation of 4-ATM's. Basically, the 'leaf-size' is the minimum number of leaves of some accepting computation trees of alternating Turing machines. Leaf-size, in a sense, reflects the minimum number of processors that run in parallel in accepting a given input.

KeyWords **:** alternation, configuration, four-dimensional input tape, leaf-size, space bound, Turing machine.

1 Introduction and Preliminaries

In 1967, the problem of computational complexity was also arisen in the two-dimensional information processing. Blum et al. first proposed two-dimensional automata, and investigated their pattern recognition abilities [1]. Since then, many researchers in this field have been investigating a lot of properties about automata on two- or three-dimensional tapes. In 1976, Chandra et al. introduced the concept of 'alternation'as a theoretical model of parallel computation [2]. After that, Inoue et al. introduced two-dimensional alternating Turing machines as a generalization of two-dimensional nondeterministic Turing machines and as a mechanism to model parallel computation [5]. Moreover, Sakamoto et al. presented three-dimensional alternating Turing machines in [7,9].

On the other hand, recently, due to the advances in many application areas such as computer animation, motion image processing, and so forth, it has become increasingly apparent that the study of four-dimensional pattern processing, i.e., three-dimensional automata with the time axis should be of crucial importance. Thus, we think that it is very useful for analyzing computation of four-dimensional pattern processing to explicate the properties of four-dimensional automata. From this viewpoint, we introduced some four-dimensional automata[6, 10].

In this paper, we continue the investigations about four-dimensional alternating Turing machines [4, 6], and mainly investigate leaf-size hierarchy of four-dimensional alternating Turing machines which each sidelength of each input tape is equivalent. Leaf-size bounded computation was introduced as a simple, natural new complexity measure for alternating Turing machines[5]. Basically, the 'leaf-size' (or 'blanching') is the minimum number of leaves of some accepting computation trees of processors that run in parallel in accepting a given input.

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:

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

(ii) for each r_1, r_2, r_3, r_4 $(1 \leq r_1 \leq l_1(y), 1 \leq r_2 \leq l_2(y), 1 \leq r_3 \leq l_3(y), 1 \leq r_4 \leq l_4(y))$, $y(r_1, r_2, r_3, r_4) = x(r_1 + i_1 - 1, r_2 + i_2 - 1, r_3 + i_3 - 1, r_4 + i_4 - 1)$.

As usual, a four-dimensional input tape x over Σ is surrounded by the boundary symbols #'s ($\# \notin \Sigma$). Furthermore, four-dimensional tape is the sequence of three-dimensional rectangular arrays along the time axis. By $Cube_x(i)$ ($i \geq 1$), we denote the ith three-dimensional rectanglar array along the time axis in $x \in \Sigma^{(4)}$ which each sidelength is equivalent.

Let Σ_1, Σ_2 be finite set of symbols. The *projection* is a mapping $\tilde{\tau}$: $\Sigma_1^{(4)} \rightarrow \Sigma_2^{(4)}$ which is obtained by extending a mapping $\tau : \Sigma_1 \rightarrow \Sigma_2$ as follows : $\tilde{\tau}(x)=x'$ if and only if (i)$l_i(x) = l_i(x')$ for each $i(1 \leq i \leq 4)$, and (ii)$\tau(x(i_1, i_2, i_3, i_4))=x'(i_1, i_2, i_3, i_4)$ for each $(i_1, i_2, i_3, i_4)(1 \leq i_1 \leq l_1(x), 1 \leq i_2 \leq l_2(x), 1 \leq i_3 \leq l_3(x), 1 \leq i_4 \leq l_4(x))$. If $T \subseteq \Sigma_1^{(4)}$, we let $\tilde{\tau}(T) = \{\tilde{\tau}(x) \mid x \in T\}$.

We now recall the definition of a *four-dimensional alternating Turing machine* (4-ATM), which can be considered as an alternating version of a four-dimensional Turing machine (4-TM) [6].

4-ATM M is defined by the 7-tuple

$$M = (Q, q_0, U, F, \Sigma, \Gamma, \delta), \text{ where}$$

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

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

(3) $U \subseteq Q$ is the set of *universal states*;

(4) $F \subseteq Q$ is the set of *accepting states*;

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

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

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

A state q in $Q - U$ is said to be *existential.* As shown in Fig. 1, the machine M has a read-only four-dimensional input tape with boundary symbols #'s and one semi-infinite storage tape, initially blank. Of course, M has a finite control, an input head, and a storage-tape head. A *position* is assigned to each cell of the read-only input tape and to each cell of the storage tape, as shown in Fig. 1. The *step* of M is similar to that of a two- or three-dimensional Turing machine [3–5, 7], except that the input head of M can move in eight directions. We say that M *accepts* the tape x if it eventually enters an accepting state. Note that the machine cannot write the blank symbol. If the input head falls off the input tape, or if the storage head falls off the storage tape (by moving left), then the machine M can make no further move.

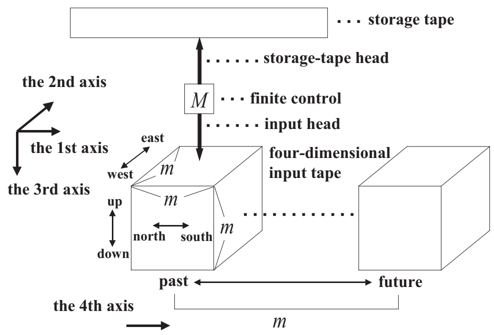

Fig. 1: Four-dimensional alternating Turing machine.

Let $L(m)$: $\mathbf{N} \rightarrow \mathbf{R}$ be a function with one variable m, where $\mathbf{N}$ is the set of all positive integers and $\mathbf{R}$ is the set of all nonnegative real numbers. With each 4-ATM M we associate a space complexity function $SPACE$ that takes configurations to natural numbers. That is, for each configuration $c = (x, (i_1, i_2, i_3, i_4), (q, \alpha, j))$, let $SPACE(c) = |\alpha|$. M is said to be $L(m)$ *space-bounded* if for each $m \geq 1$ and for each x with $l_1(x) = l_2(x) = l_3(x) = l_4(x) = m$, if x is accepted by M, then there is an accepting computation tree of M on input x such that for each node v of the tree, $SPACE(L(v)) \leq \lceil L(m) \rceil$[1]. We denote an $L(m)$ space-bounded 4-ATM by 4-ATM $(L(m))$.

A 4-$ATM(0)$ is called a four-dimensional alternating finite automaton, which can be considered as an alternating version of a four-dimensional finite automaton (4-FA), and is denoted by 4-AFA.

In order to distinguish among determinism, nondeterminism, alternation with only universal states, and alternation, we denote a deterministic 4-TM [nondeterministic 4-TM, 4-ATM with only universal states, deterministic 4-$TM(L(m))$, nondeterministic 4-$TM(L(m))$, 4-$ATM(L(m))$ with only universal states, deterministic 4-FA, nondeterministic 4-FA, 4-AFA with only universal states] by 4-DTM [4-NTM, 4-UTM, 4-$DTM(L(m))$, 4-$NTM(L(m))$, 4-$UTM(L(m))$, 4-DFA, 4-NFA].

Let M be an automaton on a three-dimensional tape. We denote by $T(M)$ the set of all three-dimensional tapes accepted by M. As usual, for each $X \in \{D, N, U, A\}$, we denote, for example, by £[3-XTM] the class of sets of all the four-dimensional tapes accepted by 4-XTM's. That is, £[4-XTM] = $\{T \mid T = T(M)$ for some 4-XTM $M\}$. £[4-XTM

[1] $\lceil r \rceil$ means the smallest integer greater than or equal to r.

$(L(m))]$, and $\pounds[4\text{-}XFA]$ also have analogous meanings.

Let $L(m)$: $\mathbf{N} \to \mathbf{R}$ be a function. For each tree t, let $LEAF(t)$ denote the leaf-size of t (i.e., the number of leaves of t). We say that a 4-ATM M is $Z(m)$ *leaf-size bounded* if for all x with $l_1(x)$=$l_2(x)$=$l_3(x)$=$l_4(x)$=m and for each computation tree t of M on x, $LEAF(t) \leq \lceil Z(m) \rceil$.

By 4-$ATM(L(m)$,$Z(m))$, we denote a $Z(m)$ leaf-size bounded 4-$ATM(L(m))$. Especially, a 4-$ATM(0$,$Z(m))$ is denoted by 4-$AFA(Z(m))$. Define $\pounds[4\text{-}ATM(L(m), Z(m))] = \{T \mid T = T(m)$ for some 4-$ATM(L(m), Z(m))$ $M\}$. We use 4-$AFA(k)$ (4-$UFA(k)$, 4-DFA) to denote a 4-$ATM(0$,$k)$ (4-$UTM(0$,$k)$, 4-$DTM(0)$).

2 Unbounded Leaf-Size Hierarchy

A function $L(m)$: $\mathbf{N} \to \mathbf{R}$ is called *four-dimensionally space constructible* if there is a *strongly* 4-$ATM(L(m))$ M such that for each $m \geq 1$, there exists some input tape x with l_1(x) = l_2(x) = l_3(x) = l_4(x) = m on which M halts after its storage head has marked off exactly $\lfloor L(m) \rfloor^2$ cells of the storage tape.(In this case, we say that M constructs the function L.)

We first show a hierarchy of complexity classes based on leaf-size bounded computations.

The main theorem is

Theorem 2.1. *Let $k \geq 1$ be a positive integer. Let L* : $\mathbf{N} \to \mathbf{N}$ and L' : $\mathbf{N} \to \mathbf{N}$ *be any functions such that*

(1) *L is a four-dimensional space-constructible function such that $L(m)^{k+1} \leq m$ $(m \geq 1)$,*

(2) $\lim_{m\to\infty} L(m)L'(m)^k/\log m = 0$, *and*

(3) $\lim_{m\to\infty} L'(m)/L(m) = 0$.

Then there is a set in $\pounds[4\text{-}ATM(L(m),L(m)^k)]$, but not in $\pounds[4\text{-}ATM(L(m),L'(m)^k)]$.

Proof: Let M be a four-dimensional deterministic Turing machine which constructs the function L. Let $T_k[L,M]$ be the following set, which depends on k, L and M :

$$T_k[L,M] = \{x \in (\textstyle\sum \times \{0,1\})^{(4)} \mid \exists m \geq 2\ [l_1(x) = l_2(x) = l_3(x) = l_4(x) = m \& \exists r (r \leq L(m)\ [(\text{when the tape } \tilde{h_1}(x) \text{ is presented to } M, \text{ its read-write head marks off r cells of the storage tape and then halts}) \ \&\ \exists i\ (1 \leq i \leq m-1)\ [\tilde{h_2}(x[(1,1,m,1),r^{k+1}, r^{k+1},m,1)]) = [\tilde{h_2}(x[(1,1,i,1),(r^{k+1}, r^{k+1},i,1)])]]]\},$$

where $\sum$ is the input alphabet of M, and $\tilde{h_1}(\tilde{h_2})$ is the *projection* which is obtained by extending the mapping $h_1 : \sum \times \{0,1\} \to \sum$ (h_2:$\sum\times\{0,1\}\to\{0,1\}$) such that for any $c = (a,b) \in \sum \times \{0,1\}$, $h_1(c) = a(h_2(c) = b)$.

We first show that $T_k[L,M] \in \pounds[4\text{-}ATM(L(m), L(m)^k)]$. Suppose that an input x with l_1(x) = l_2(x) = l_3(x) = l_4(x) = m $(m \geq 2)$ is presented to M_1. M_1 directly simulates the action of M on $\tilde{h_1}$. If M does not halt, then M_1 also does not halt, and will not accept x. If M_1 finds out that M halts (in this case, note that M_1, has marked off at most cells of the storage tape because M constructs the function L), then M_1 existentially chooses some $i(1 \leq i \leq m-1)$ and moves its input tape head on x(1,1,i,1). After that, M_1 universally tries to check that, for each $1 \leq j \leq r^k$, where r is the length of the non-blank part of the storage tape just after M_1 has found out that M halts,
$\tilde{h_2}(x[((j-1)r+1,(j-1)r+1,i,1),(jr,jr,i,1)])$
$=\tilde{h_2}(x[((j-1)r+1,(j-1)r+1,m,1),(jr,jr,m,1)])$.
That is, on $x((j-1)r+1,(j-1)r+1,i,1)$ $(1 \leq j \leq r^k)$, M_1 enters a universal state to choose one of two further actions. One action is to pick up and store the segment $\tilde{h_2}(x[((j-1)r+1,(j-1)r+1,i,1),(jr,jr,i,1)])$ on some track of the storage tape, to compare the segment stored above with the segment $\tilde{h_2}(x[((j-1)r+1,(j-1)r+1,m,1),(jr,jr,m,1)])$, and to enter an accepting state only if both segments are identical. The other action is to continue moving to $x(jr+1,jr+1,i,1)$ (in order to pick up the next segment $\tilde{h_2}(x[jr+1,jr+1,i,1),((j+1)r,(j+1)r,i,1)]))$ and compare it with the corresponding segment $\tilde{h_2}(x[(jr+1,jr+1,m,1),((j+1)r,(j+1)r,m,1)])$.
Note that the number of pairs of segments which should be compared with each other in the future can be easily seen by using r cells of the storage tape. It will be obvious that the input x is in $T_k[L,M]$ if and only if there is an accepting computation three of M_1 on x with at most $L(m)^k$ leaves. Thus $T_k[L,M] \in \pounds[4\text{-}ATM(L(m), L(m)^k)]$.

On the other hand, we can next show that $T_k[L,M] \notin \pounds[4\text{-}ATM(L(m), L'(m)^k)]$ by using the well-known counting argument [3,7]. This completes the proof of the theorem. □

Corollary2.1. *Let $k \geq 1$ be a positive integer. Let L*

[2] $\lfloor r \rfloor$ means the greatest integer smaller than or equal to r.

: $\mathbf{N} \to \mathbf{N}$ *and* $L' : \mathbf{N} \to \mathbf{N}$ *be any functions satisfying the condition that* $L'(m) \leq L(m)(m \geq 1)$ *and satisfying conditions* (1), (2) *and* (3) *described in Theorem* 2.1. *Then,*
$\pounds[4\text{-}ATM(L(m),L'(m)^k)] \subsetneq \pounds[4\text{-}ATM(L(m),L(m)^k)]$.

For any r in $\mathbf{N}$, $\log^{(r)} m$ be the function defined as follows :

$$log^{(1)}m = \begin{cases} 0 & (m = 0) \\ logm & (m \geq 1), \end{cases}$$

$$log^{(r+1)}m = \Big\{ log^{(1)}(log^{(r)}m).$$

As shown in Theorem2.1 of [8], the function $\log^{(r)} m(r \geq 1)$ is three-dimensionally space-constructible, and thus four-dimensionally space-constructible. It is easy to see that for each $r \geq 1$, $\log^{(r+1)} m \leq \log^{(r)} m (m \geq 1)$ and $\lim_{m \to \infty} \log^{(r+1)} m / \log^{(r)} m = 0$. Further, for each $r \geq 2$ and each $k \geq 1$, $\lim_{m \to \infty} \log^{(r)} m (\log^{(r+1)} m)^k / \log m = 0$. From these facts, we have the following.

Corollary2.2. *For any* $r \geq 2$ *and any* $k \geq 1$,
$\pounds[4\text{-}ATM(\log^{(r)} m,(\log^{(r+1)} m)^k) \subsetneq$
$\pounds[4\text{-}ATM(\log^{(r)} m,(\log^{(r)} m)^k)$.

3 Constant Leaf-Size Hierarchy

We next investigate a constant leaf-size hierarchy : Are $k + 1$ leaves better then k? We first show that in the case of an alternating Turing machine with only universal states, no hierarchy exists for any space bound.

Theorem 3.1. *For any* $k \in \mathbf{N}$ *and any function* $L(m)$,

$$\pounds[4\text{-}UTM\ (L(m),\ k)] = \pounds[4\text{-}DTM\ (L(m))].$$

Proof: Given a k leaf-size bouded 4-UTM M and an input tape x, a 4-DTM M' performs a *depth-first-search* on the computation tree of M on x without any extra cells of the working tape : Normal tree-search method needs one stack for backtracking, Instead, M' adopts only the forward tracking from the root to each leaf and uses finite internal memories in the finite control. Note that since M has constant leaves, the branching structure of universal configurations of M on x is also constant. After each traversal of a path and finding out its leaf is labeled with an accepting configuration M' adds the newly obtained information about the tree structure into a memory cell of the finite control. Then M begins to walk from the root to the next leaf, whose route can be specified by referring the memories of the finite control. When the whole travel have been done and if M is surely k leaf-size bouded, M' enters an accepting state. Note that M' accepts exactly $T(M)$ and that M' is L space-bouded if and only if M is L space-bouded. □

Corollary3.1. *For any* $k \in \mathbf{N}$,

$$\pounds[4\text{-}UFA\ (k)] = \pounds[4\text{-}DFA].$$

In contrast to universal machines, we can show that there exists an infinite hierarchy of $o(\log m)$ space-bouded four-dimentional alternating Turing machines based on leaf-size by using the block of input tape and the counting argument[3,7].

Theorem3.2. *For each* $k \in \mathbf{N}$,if $L(m)$=$o(\log m)$, *then*

$$\pounds[4\text{-}ATM\ (L(m),\ k)] \subsetneq \pounds[4\text{-}ATM\ (L(m), k+1)].$$

Corollary3.2. *For each* $k \in \mathbf{N}$,

$$\pounds[4\text{-}AFA\ (k)] \subsetneq \pounds[4\text{-}AFA\ (k+1)].$$

References

[1] M. Blum et al., *IEEE Symposium on Switching and Automata Theory* :155-160(1967).

[2] A. K. Chandra et al., *J. ACM* 28(1):114-133 (1981).

[3] J. E. Hopcroft et al., *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, Reading, Mass. (1979).

[4] K.Iihoshi et al., *Proc. 11th AROB*(2006).

[5] K. Inoue et al., *Theoret. Comput. Sci.* 27:61-83 (1983).

[6] H. Okabe et al., *Proc. 7th Words Multiconf. on SCI* :241-246 (2003).

[7] M. Sakamoto et al., *Inform. Sci.* 72:225-249 (1993).

[8] M.Sakamoto, et al., *Trans. IEICE Jpn.* E77-D(6):723-725(1994).

[9] M. Sakamoto, *Ph.D. Thesis* (1999).

[10] M. Sakamoto et al., *WSEAS Trans. on Computers*, Issue 5, Vol. 3 :1651-1656 (2004).