Constant Leaf-Size Hierarchy of Three-Dimensional Alternating Turing Machines

Makoto SAKAMOTO
(Dept. of Shipping Technology, Oshima National College of Maritime Technology)
Katsushi INOUE
(Dept. of Computer Science and Systems Engineering)

Abstract

'Leaf-size' (or 'branching') is the minimum number of leaves of some accepting computation trees of alternating devices. For example, one leaf corresponds to nondeterministic computation. In this paper, we investigate the effect of constant leaves of three-dimensional alternating Turing machines, and show the following facts : (1) For cubic input tapes, k leaf- and $L(m)$ space-bounded three-dimensional alternating Turing machines with only universal states are equivalent to the same space-bounded three-dimensional deterministic Turing machines for any integer $k \geq 1$ and any function $L(m)$. (2) For cubic input tapes, $k+1$ leaf- and $o(\log m)$ space-bounded three-dimensional alternating Turing machines are more powerful than k leaf-bounded ones for each $k \geq 1$.

Keywords **:** *Leaf-size, alternation, space complexity, three-dimensional Turing machine, three-dimensional finite automaton.*

1. Introduction

Inoue and Takanami [5] introduced a three-way two-dimensional leaf-size bounded computation was introduced as a simple, natural new complexity measure for alternating Turing machines*. Basically, the 'leaf-size' (or 'branching') 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. After that, several interesting facts concerning the computational complexity based on this measure has been revealed. For instance, Yamamoto shows the leaf compression theorem for time-bounded alternating Turing machines [15]. Matsuno *et al.* and Hromkovic applies the concept of leaf-size to alternating multihead automata [4, 10]. Moreover, due to the advances in the processing of pictorial information by computer, it has become increasingly apparent that the study of two- or three-dimensional pattern processing should be very important. Thus, the research of multi-dimensional automata as the computational model of two- or three-dimensional pattern processing has also been meaningful [2, 3, 14, 16]. Ito *et al.* investigated several properties of leaf-size bounded two-dimensional Turing machines. In [7], they showed that a parallel two-dimensional machine with cooperative processors is more powerful than a two-dimensional mechanism with the same number of processors which run independently. In [6, 8], they established a hierarchy of complexity classes besed on leaf-size bounded computations for two-dimensional alternating Turing machines, and the constant leaf-size hierarchy of two-dimensional alternating Turing machines using small space. On the other hand, we introduced a three-dimensional alternating Turing machine, and investigated its several properties [11, 12]. In [13], we provided an unbounded leaf-size hierarchy of three-dimensional alternating Turing machines.

In this paper, we continue the investigations about a leaf-size hierarchy of three-dimensional alternating Turing machines. We show that for three-dimensional alternating Turing machines with only universal states, the hierarchy collapses to the deterministic class, as with the

*King independently introduced the same complexity measure as 'leaf-size' [9]. In [9], the term 'branching' is adopted instead of the term 'leaf-size'.

case of large space bound. In contrast, for normal three-dimensional alternating Turing machines using small space bound, a strict hierarchy emerges again. More precisely, it is shown that there exists a set of cubic tapes accepted by a $k+1$ leaf-bounded three-dimensional alternating finite automata, but not accepted by any k leaf- and $o(\log m)$ space bounded three-dimensional alternating Turing machines.

2. Preliminaries

Definition 2.1. Let Σ be a finite set of symbols. A *three-dimensional tape* over Σ is a three-dimensional rectangular array of elements of Σ. The set of all three-dimensional tapes over Σ is denoted by $\Sigma^{(3)}$.

Given a tape $x \in \Sigma^{(3)}$, for each integer $j(1 \leq j \leq 3)$, we let $l_j(x)$ be the length of x along the j-th axis. If $1 \leq i_j \leq l_j(x)$ for each $j(1 \leq j \leq 3)$, let $x(i_1, i_2, i_3)$ denote the symbol in x with coordinates (i_1, i_2, i_3). Furthermore, we define

$$x[(i_1, i_2, i_3), (i'_1, i'_2, i'_3)],$$

when $1 \leq i_j \leq i'_j \leq l_j(x)$ for each integer $j(1 \leq j \leq 3)$, as the three-dimensional tape y satisfying the following (i) and (ii) :

(i) for each $j(1 \leq j \leq 3)$, $l_j(y) = i'_j - i_j + 1$;

(ii) for each r_1, r_2, r_3 $(1 \leq r_1 \leq l_1(y), 1 \leq r_2 \leq l_2(y), 1 \leq r_3 \leq l_3(y))$, $y(r_1, r_2, r_3) = x(r_1 + i_1 - 1, r_2 + i_2 - 1, r_3 + i_3 - 1)$. (We call $x[(i_1, i_2, i_3), (i'_1, i'_2, i'_3)]$ the $[(i_1, i_2, i_3), (i'_1, i'_2, i'_3)]$-*segment* of x.)

As usual, an input three-dimensional tape x over Σ is surrounded by the boundary symbol $\sharp$ ($\sharp \notin \Sigma$). Coordinates are naturally assigned to boundary symbols. That is, if there is an integer i_j such that $i_j = 0$ or $i_j = l_j(x) + 1$ for some j $(1 \leq j \leq 3)$, then we let $x(i_1, i_2, i_3) = \sharp$. Furthermore, for each i $(1 \leq i \leq l_1(x))$, we call $x[(i, 1, 1), (i, l_2(x), l_3(x))]$ the *i-th* (2-3) *plane* of x, and denote it by $x(2\text{-}3)_i$. Similarly, for each $j(1 \leq j \leq l_2(x))$ and $k(1 \leq k \leq l_3(x))$, we call $x[(1, j, 1), (l_1(x), j, l_3(x))]$ and $x[(1, 1, k), (l_1(x), l_2(x), k)]$ the *j-th* (1-3) *plane* and *k-th* (1-2) *plane* of x, and denote them by $x(1\text{-}3)_j$ and $x(1\text{-}2)_k$, respectively.

We now introduce a three-dimensional alternating Turing machine (3-ATM), which can be considered as an alternating version of a three-dimensional Turing machine [11, 14].

Definition 2.2. A *three-dimensional alternating Turing machine* (3-ATM) M is defined by the septuple

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

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

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

and

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

A state q in $Q - U$ is said to be *existential*. The machine M has a read-only three-dimensional input tape with boundary symbols $\sharp$'s and one semi-infinite *storage tape*, initially blank. Of course, M has a *finite control*, an *input head*, 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. A *step* of M consists of reading one symbol from each tape, writing a symbol on the storage tape, moving the input and storage heads in specified directions, and entering a new state, in accordance with the next move relation δ. Note that the machine cannot write the blank symbol. If the input head falls off the input tape, or if the storage head falls off the storage tape (by moving left), then the machine M can make no further move.

Definition 2.3. A *configuration* of a 3-ATM $M = (Q, q_0, U, F, \Sigma, \Gamma, \delta)$ is a pair of an element $\Sigma^{(3)}$ and an element of

$$C_M = (\mathbf{N} \bigcup \{0\})^3 \times S_M,$$

where $S_M = Q \times (\Gamma - \{B\})^* \times \mathbf{N}$, and $\mathbf{N}$ denotes the set of all positive integers. The first component of a configuration $c = (x, ((i_1, i_2, i_3), (q, \alpha, k)))$ represents the input to M. The first component (i_1, i_2, i_3) of the second component of c represents the input head position. The second component (q, α, k) of the second component of c represents the state of the finite control, nonblank contents of the storage state, and the storage-head position. An element of C_M is called a *semi-configuration* of M and an element of S_M is called a *storage state* of M. If q is the state associated with configuration c, then c is said to be a *universal* (*existential*, *accepting*) configuration if q is a universal (existential, accepting) state. The *initial configuration* of M on input x is

$$I_M(x) = (x, ((1,1,1), (q_0, \lambda, 1))),$$

where λ is the null string.

Definition 2.4. Given $M = (Q, q_0, U, F, \Sigma, \Gamma, \delta)$, we write

$$c \vdash_M c'$$

and say c' is a *successor* of c if configuration c' follows from configuration c in one step of M, according to the transition rules δ. The relation $\vdash_M^*$ denotes the reflexive transitive closure of $\vdash$. A *computation path* of M on x is a sequence

$$c_0 \vdash_M c_1 \vdash_M \cdots \vdash_M c_n \ (n \geq 1).$$

A *computation tree* of M is a nonempty labeled tree with the following properties :
(1) each node π of the tree is labeled with a configuration $l(\pi)$,
(2) if π is an internal node (a nonleaf) of the tree, $l(\pi)$ is universal and

$$\{c \mid l(\pi) \vdash_M c\} = \{c_1, \ldots, c_k\},$$

then π has exactly k children $\rho_1, \ldots, \rho_k$ such that $l(\rho_i) = c_i$ $(1 \leq i \leq k)$;
(3) if π is an internal node of the tree and $l(\pi)$ is existential, then π has exactly one child ρ such that

$$l(\pi) \vdash_M l(\rho).$$

An *accepting computation tree* of M on an input x is a finite computation tree of M whose root is labeled with $I_M(x)$ and whose leaves are all labeled with accepting configurations. We say that M *accepts* x if there is an accepting computation tree of M on input x. Define

$$T(M) = \{x \in \Sigma^{(3)} \mid M \text{ accepts } x\}.$$

A three-dimensional deterministic Turing machine and a three-dimensional alternating Turing machine with only universal states are special cases of a 3-ATM. That is, the former is a 3-ATM whose configurations each have at most one successor and the latter is a 3-ATM which has no existential states [12]. By '3-DTM' ('3-UTM') we denote a *three-dimensional deterministic Turing machine* (a *three-dimensional alternating Turing machine with only universal states*).

Definition 2.5. Let $L(m) : \mathbf{N^2} \mapsto \mathbf{N}$ be a function with two variables m and n. For any $3 - ATM\ M$, we associate a complexity function $SPACE$ which takes configuration $c = (x, ((i_1, i_2, i_3), (q, \alpha, k)))$ to natural numbers. Let $SPACE(c) = \mid \alpha \mid$. We say that M is $L(m)$ *space-bounded* if for all $m \geq 1$ and for each x with $l_1(x) = l_2(x) = l_3(x) = m$, 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, $SPACE(l(\pi)) \leq \lceil L(m) \rceil^{\dagger}$. By '3-$ATM(L(m))$', we denote an $L(m)$ space-bounded 3-ATM. '3-$DTM(L(m))$', '3-$UTM(L(m))$' are similarly defined.

Especially, 3-$ATM(0)$ is denoted by '3-AFA' and called a *three-dimensional alternating finite automaton.* '3-DFA', and '3-UFA' are similarly defined.

We next present a simple, natural complexity measure for 3-ATM's, called *leaf-size* [5, 9]. Basically, the leaf-size used by a 3-ATM on a given input is the number of leaves of an accepting computation tree with the fewest leaves. Leaf-size, in a sense, reflects the minimum number of processors that run in parallel in accepting a given input.

Definition 2.6. Let $Z(m) : \mathbf{N^2} \mapsto \mathbf{N}$ 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 3-$ATM\ M$ is $Z(m)$ *leaf-size bounded* if for all input x with $l_1(x) = l_2(x) = l_3(x) = m$, if x is accepted by M then there is an accepting tree t of M such that $LEAF(t) \leq \lceil Z(m) \rceil$. By '3-$ATM(L(m), Z(m))$', we denote a simultaneously $L(m)$ space-bounded and $Z(m)$ leaf-size bounded 3-ATM. '3-$UTM(L(m), Z(m))$', '3-$AFA(Z(m))$', and '3-$UFA(Z(m))$' are similarly defined.

In some part of this paper, we concentrate on the properties of 3-ATM's whose input tapes are restricted to cubic ones. In this case, complexity function L or Z has only one variable, conventionally m. By '3-$ATM^c(L(m))$' we denote an $L(m)$ space-bounded 3-ATM whose input tapes are restricted to cubic ones. '3-$DTM^c(L(m))$', etc. are defined similarly. The class of sets accepted by 3-ATM^c's is defined as follows.

$\mathcal{L}$[3-$ATM^c(L(m))$] $= \{T \mid T = T(M)$ for some

$^{\dagger}\lceil r \rceil$ is the smallest integer greater than or equal to r.

3-$ATM^c(L(m))M\}$. $\mathcal{L}$[3-$DTM^c(L(m))$], etc. are similarly defined.

Definition 2.7. Let $g : \mathbf{N} \to \mathbf{N}$ be a function and x be a three-dimensional tape with $l_1(x) = l_2(x) = n$. For each k $(1 \leq k \leq l_3(x)/g(n))$, we call

$$x[(1, 1, (k-1)g(n) + 1), (n, n, kg(n))]$$

the *k-th* $g(n)$-*block of* x, when $l_3(x)$ is divided by $g(n)$. We simply denote it by $x[block_{g(n)}(k)]$.

3. Results

We mainly investigate a constant leaf-size hierarchy : Are $k+1$ leaves better than k ?

Now 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* $L(m)$,

$$\mathcal{L}[3\text{-}UTM^c(L(m), k)] = \mathcal{L}[3\text{-}DTM^c(L(m))].$$

Proof : Given a k leaf-size bounded 3-*UTM* M and an input tape x, a 3-*DTM* M' performs a *depth-first-search* (see [1]) 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 to the memories of the finite control. When the whole travel have been done and if M is surely k leaf-size bounded, M' enters an accepting state. Note that M' accepts exactly $T(M)$ and that M' is L space-bounded iff is L space-bounded. □

Corollary 3.1. *For any* $k \in \mathbf{N}$, $\mathcal{L}$[3-$UFA^c(k)$] $= \mathcal{L}$[3-DFA^c].

In contrast to six-way universal machines, we can show that there exists an infinite hierarchy of $o(\log m)$ space-bounded three-dimensional alternating Turing machines based on leaf-size. To this end, we have to give several preliminaries at first.

Let M be a 3-$ATM^c(l, z)$. Note that if the numbers of states and storage-tape symbols of M are s and t, respectively, then the number of possible storage states of M is slt^l. Let Σ be the input alphabet of M, and let $\sharp$ be the boundary symbol of M. For each k, m, n $(k \geq 1, m \geq k+1, 1 \leq n \leq m-1)$, we now consider an (m, n, k)-*chunk* over Σ. For any (m, n, k)-chunk x over Σ, we denote by $x(\sharp)$ the pattern (obtained from x by surrounding x with $\sharp$'s). Below we assume without loss of generality that for any (m, n, k)-chunk, M has the following property :

(A) M enters or exits the object $x(\sharp)$ only at the $[(m, 1, 1), (m-k+1, n, 1)]$-segment of x, and never enters an accepting state in $x(\sharp)$.

Then the number of entrance points to $x(\sharp)$ [or the exit points from $x(\sharp)$] for M is $(n+3)k + 3n + 5$. We suppose that these entrance points (or exit points) are numbered $1, 2, \ldots, (n+3)k + 3n + 5$. For each (m, n, k)-chunk x, a configuration of M on $x(\sharp)$ is of the form

$$(x(\sharp), (p, (q, \alpha, j))),$$

where p represents the position of the head of M on $x(\sharp)$, and (q, α, j) represents a storage state of M. The second component $(p, (q, \alpha, j))$ of a configuration $I = (x(\sharp), (p, (q, \alpha, j)))$ is called the *semi-configuration component* of I. For convenience sake, for each i $(1 \leq i \leq (n+3)k + 3n + 5)$, let the position of the cell confronted with entrance point i of $x(\sharp)$ be 'i'. Further, we consider $(n+2)k + 2n + 2$ virtual cells (confronted with $x(\sharp)$) by using the same idea in [6], and we assign position $1', 2', \ldots, ((n+2)k + 2n + 2)'$ to these virtual cells. We include these positions in the set of positions of the head of M on $x(\sharp)$.

A configuration $I = (x(\sharp), (p, (q, \alpha, j)))$ is said to be *universal* (*existential*) if q is a universal (existential) state. For any configurations I and I' of M on $x(\sharp)$, we write $I \vdash_M I'$ and say I' is a *successor* of I if I' follows from I in one step of M on $x(\sharp)$. Note that for any configuration $I = (x(\sharp), (p, (q, \alpha, j)))$, where x is an (m, n, k)-chunk, such that $p \in \{1', 2', \ldots, ((n+2)k + 2n + 2)'\}$ (i.e., p is a virtual position), I has no successor.

A *computation tree* of M on $x(\sharp)$ is a finite, nonempty labeled tree with the properties :

(1) each node π of the tree is labeled with a configuration, $l(\pi)$, of M on $x(\sharp)$;

(2) if π is an internal node (a nonleaf) of the tree and $l(\pi)$ is universal and $\{I \mid l(\pi) \vdash_M I\} = \{I_1, \dots, I_r\}$, then π has exactly r children $\rho_1, \dots, \rho_r$ such that $l(\rho_i) = I_i$;

(3) if π is an internal node of the tree and $l(\pi)$ is existential, then π has exactly one child ρ such that $l(\pi) \vdash_M l(\rho)$.

A *prominent computation tree* of M on an (m, n, k)-chunk x is a computation tree of M on $x(\sharp)$ with the properties :

(1) the root node is labeled with a configuration of the form $(x(\sharp), (i, (q, \alpha, j)))$, where $1 \leq i \leq (n+3)k + 3n + 5$ (i.e., the root node is labeled with a configuration of M just after M entered the pattern $x(\sharp)$ from some entrance point i);

(2) each leaf node is labeled either

(a) with a configuration of the form $(x(\sharp), (p, (q, \alpha, j)))$, where $p \in \{1', 2', \dots, ((n+2)k + 2n + 2)'\}$ (i.e., a configuration of M just after M exited the pattern $x(\sharp)$), or

(b) with a configuration I such that starting from the configuration I, M never reaches a universal configuration which has two or more successors,and M never exists from $x(\sharp)$.

(A leaf node labeled with a configuration of type (b) above is called a *looping leaf node*. A leaf node labeled with a configuration of type (a) above is called a *normal leaf node*.)

Let $C = \{c_1, c_2, \dots, c_u\}$ be the set of possible storage states of M, where $u = slt^l$. For each prominent computation tree t of M on an (m, n, k)-chunk, let the *leaf semi-configuration set* of t (denoted by $LSCS(t)$) be a 'multiset' of elements of $\{1', 2', \dots, ((n+2)k+2n+2)'\} \times C \bigcup \{L\}$ (where L is a new symbol) defined as follows :

(1) for each normal leaf node π of t, $LSCS(t)$ contains the semi-configuration component of $l(\pi)$;

(2) for each looping leaf node of t, $LSCS(t)$ contains the symbol L ;

(3) $LSCS(t)$ does not contain any element other than elements described in (1) and (2) above.

(Note that any prominent computation tree t of M, $\mid LSCS(t) \mid \leq z$, since M is z leaf-size bounded.)

For each (m, n, k)-chunk x and for each $(i, c) \in \{1, 2, \dots, (n+3)k + 3n + 5\} \times C$, let

$M_{(i,c)}(x) = \{LSCS(t) \mid t$ is a prominent computation tree of M on x whose root is labeled with the configuration $(x(\sharp), (i, c))\}$.

Let x, y be two (m, n, k)-chunks. We say that x and y are *M-equivalent* if for each $(i, c) \in \{1, 2, \dots, (n+3)k + 3n + 5\} \times C$, $M_{(i,c)}(x) = M_{(i,c)}(y)$.

For any (m, n, k)-chunk x and for any tape $v \in \Sigma^{(3)}$ with $l_1(v) = k$, $l_2(v) = n$, and $l_3(v) = 1$, let $x[v]$ be the tape in $\Sigma^{(3)}$ consisting of v and x.

The following lemma means that M cannot distinguish between two (m, n, k)-chunks which are M-equivalent.

Lemma 3.1. *Let M be a 3-$ATM^c(l, z)$, with the property* (A) *and Σ be the input alphabet of M. Let x, y be M-equivalent (m, n, k)-chunks over Σ. Then, for any tape $v \in \Sigma^{(3)+}$ with $l_1(v) = k$, $l_2(v) = m$, and $l_3(v) = 1$, is accepted by M if and only if $y[v]$ is accepted by M.*

Proof : (If part). We assume that $y[v]$ is accepted by M. Then there exists an accepting computation tree t of M on $y[v]$ such that $LEAF(t)$ (i.e., the number of leaves of t) $\leq z$. Since x and y are M-equivalent, we can construct from t an accepting computation tree t' of M on $x[v]$ such that $LEAF(t') = LEAF(t) \leq z$. Therefore, $x[v]$ is accepted by M.

(Only-if part). Analogous to 'if part'. □

Clearly, M-equivalent is an equivalence relation on (m, n, k)-chunks, and we obtain the following lemma.

Lemma 3.2. *Let M be a 3-$ATM^c(l, z)$, and Σ be the input alphabet of M. Further, let s and t be the numbers of states and storage tape symbols of M, respectively, and let $u = slt^l$. Then there are at most $(2^{b^{z+1}})^d$ M-equivalence classes of (m, n, k)-chunks over Σ, where $b = ((n+2)k + 2n + 2)u + 1$ and $d = ((n+3)k + 3n + 5)u$.*

Proof : The lemma follows from the observation that

(1) $\mid \{1, 2, \dots, (3+n)k + 3n + 5\} \times C \mid = ((n+3)k + 3n + 5)u = d$ (where C is the set of possible storage states of M), and

(2) the number of possible leaf configuration sets of prominent computation trees of M on (m, n, k)-chunks is bounded by

$$b + b^2 + \cdots + b^z \leq b^{z+1}$$
$$\text{(where } b = ((n+2)k + 2n + 2)u + 1)$$

since M is z leaf-size bounded. □

We are now ready to prove the key lemma.

Lemma 3.3. *For each* $k \in \mathbf{N}$, *define*

$T < k >= \{ x \in \{0,1\}^{(3)+} \mid \exists m \geq 2\ [l_1(x) = l_2(x) = l_3(x) = km\ \& \exists i(1 \leq i \leq m-1)[x[block_k(i)] = x[block_k(m)]]\&\forall i_1 (1 \leq i_1 \leq l_1(x))$ [(*each row of the top* (1-2) *plane of* x *has exactly one* '1')]].

Then

(1) $T < k+1 > \in \mathcal{L}[3\text{-}AFA^c(k)]$, *and*

(2) *if* $L(m) = o(\log m)$, $T < k+2 > \notin \mathcal{L}[3\text{-}ATM^c(L(m), k)]$.

Proof : (1) We construct a 3-$AFA^c(k)$ M which accepts $T < k+1 >$ as follows. Given an input tape $x \in \{0,1\}^{(3)+}$, M checks that x has m $(k+1)$-blocks for some $m \geq 2$ and that each row of the top (1-2) plane of x has just one '1'. (In order to locate itself within a $(k+1)$-block of x, M uses a $mod(k+1)$ counter and increases or decreases the counter at each step along the 1st axis.) If this check succeeds, M moves to the position of the symbol '1' on the first row of the last $(k+1)$-block of x. From this position, M begins to move north looking for '1'. Each time M meets the symbol '1' on the first row of some $(k+1)$-block, it guesses whether or not the current $(k+1)$-block is equal to the last $(k+1)$-block. If so, it moves south from the first row to the last row of this block. On the l-th row in the block $(2 \leq l \leq k)$, it universally branches into two machines, one to continue descending along the 1st axis and the other to move east or west along the 2nd axis looking for '1' on the l-th row. On the $k+1$st row in the block, M only moves east or west along the 2nd axis looking for '1'. Each machine, say M_l $(2 \leq l \leq k+1)$, which has reached the symbol '1' on the l-th row begins to move south along the 1st axis for row-by-row check of two $(k+1)$-blocks equality. In the last $(k+1)$-block of x, machine M_l $(2 \leq l \leq k+1)$ enters an accepting state if and only if the symbol of the l-th row in the last block is '1'. It is clear that $T(M) = T < k+1 >$ and M is k leaf-bounded.

(2) Suppose to the contrary that there exists a 3-$ATM^c(L(m), k)$ M accepting $T < k+2 >$, where $L(m) = o(\log m)$. Without loss of generality, we assume that when M accepts a given input tape x, it enters an accepting state at the westmost position of the last row of the top (1-2) plane of x. For each $n \geq 1$, let

$$V(n) = \{ x \in T < k+2 > \mid l_1(x) = l_2(x) = l_3(x) = (k+2)(n^{k+2}+1)\&x[(1, n+1, 1), (l_1(x), l_2(x), 1)] \in \{0\}^{(3)+}\}, \text{ and}$$

$$Y(n) = \{x \in \{0,1\}^{(3)+} \mid l_1(x) = k+2\&l_2(x) = n\&l_3(x) = 1\}.$$

Clearly, $A(n) = \{p \mid$ for some x in $V(n)$, p is the pattern obtained from x by cutting the segment $x[(l_1(x) - (k+2) + 1, 1, 1), (l_1(x), n, 1)]$ off $\}$ is a set of $((k+2)(n^{k+2}+1), n, k+2)$-chunks over $\{0,1\}$. For each chunk x in $A(n)$, let

$$BLOCK(x) = \{y \in Y(n) \mid \exists i(1 \leq i \leq n^{k+2}) [x[v_0][(i-1)(k+2)+1, 1, 1), (i(k+2), n, 1)] = y\},$$

where v_0 is an arbitrary fixed element in $Y(n)$. Furthermore, let $eqv_M(x)$ denote the M-equivalent class of a $((k+2)(n^{k+2}+1), n, k+2)$-chunk x over $\{0,1\}$. Then, the following proposition holds.

Proposition 3.1. *For any chunks* x *and* y *in* $A(n)$,

$$\text{if } BLOCK(x) \neq BLOCK(y), \text{ then } eqv_M(x) \neq eqv_M(y).$$

[**Proof :** Suppose to the contrary that $BLOCK(x) \neq BLOCK(y)$ but $eqv_M(x) = eqv_M(y)$. Without loss of generality, we assume $\beta \in BLOCK(x)$ and $\beta \notin BLOCK(y)$ for some $\beta \in Y(n)$. Consider M on two tapes $x[\beta]$ and $y[\beta]$. Since $x[\beta] \in V(n) \subseteq T < k+2 >$, M accepts $x[\beta]$. Then, from Lemma 3.1, it follows that M also accepts $y[\beta]$, which is a contradiction. (Note that $y[\beta] \notin T < k+2 >$.) □
]

Proof of Lemma 3.3 (*continued*) **:** Since M can use at most $L((k+2)(n^{k+2}+1))$ cells of the storage tape and M is k leaf-size bounded when M reads a tape in $V(n)$, from Lemma 3.2, there are at most

$$E(n) = (2^{b[n]^{k+1}})^{d[n]}$$

M-equivalence classes of $((k+2)(n^{k+2}+1), n, k+2)$-chunks (over $\{0,1\}$) in $A(n)$, where $b[n] = ((n+2)k + 2n + 2)u[n] + 1$, $d[n] = ((n+3)k + 3n + 5)u[n]$ and $u[n] = sL((k+2)(n^{k+2}+1))t^{L((k+2)(n^{k+2}+1))}$.

We denote these M-equivalence classes by $C_1, C_2, \ldots, C_{E(n)}$. On the other hand, defining $B(n) = \{BLOCK(x) \mid x \in A(n)\}$, we have

$$\begin{aligned} \mid B(n) \mid &= \binom{n^{k+2}}{1} + \binom{n^{k+2}}{2} + \cdots + \binom{n^{k+2}}{n^{k+2}} \\ &= 2^{n^{k+2}} - 1. \end{aligned}$$

Since $L(m) = o(\log m)$, it follows that $L((k+2)(n^{k+2}+1)) = o(\log n)$. Thus, it follows that for large n,

$$\mid B(n) \mid > E(n).$$

For such an n, it follows that there exist two M-equivalent $((k+2)(n^{k+2}+1), n, k+2)$-chunks x and y such that $BLOCK(x) \neq BLOCK(y)$, which contradicts Proposition 3.1. We have finished the proof of Lemma 3.3. □

From Lemma 3.3, we got the desired result.

Theorem 3.2. *For each* $k \in \mathbf{N}$, *if* $L(m) = o(\log m)$, *then*

$$\mathcal{L}[3\text{-}ATM^c(L(m), k)] \subsetneq \mathcal{L}[3\text{-}ATM^c(L(m), k+1)].$$

Corollary 3.2. *For each* $k \in \mathbf{N}$,

$$\mathcal{L}[3\text{-}AFA^c(k)] \subsetneq \mathcal{L}[3\text{-}AFA^c(k+1)].$$

4. Concluding Remarks

In this paper, we introduced restricted types of three-dimensional alternating Turing machines, called 'leaf-size bounded three-dimensional alternating Turing machines'. We mainly investigated the constant leaf-size hierarchy of three-dimensional alternating Turing machines. We first showed that for three-dimensional alternating Turing machines with only universal states, the hierarchy collapses to the deterministic class, as with the case of large space bound. In contrast, we next showed that for three-dimensional alternating Turing machines using small space bound, a strict hierarchy emerges again. More precisely, it was shown that there exists a set of cubic tapes accepted by a $k+1$ leaf-size bounded three-dimensional alternating finite automata, but not accepted by any $o(\log m)$ space-bounded and k leaf-size bounded three-dimensional alternating Turing machines. Thus, even the three-dimensional alternating finite automata of two leaves are more powerful than three-dimensional nondeterministic finite automata.

It will also be interesting to investigate leaf-size hierarchy properties of the classes of sets accepted by 3-ATM^c's with spaces of size greater than $\log m$.

References

[1] Aho,A., J.E.Hopcroft, and J.D.Ullman, *The Design and analysis of computer algorithms*, Addison-Wesley, Reading, Mass., 1974.

[2] Blum,M. and C.Hewitt, Automata on a two-dimensional tape, *in IEEE Symposium on Switching and Automata Theory*, pages 155-160, 1967.

[3] Blum,M. and W.J.Sakoda, On the capability of finite automata in 2 and 3 dimensional space, *in Proc. 18th Annual Symposium on Fundations of Computer Sciences*, pages 147-161, 1977.

[4] Hromkovic,J., K.Inoue, and I.Takanami, Lower bounds for language recognition on two-dimensional alternating multihead machines, *to appear in Journal of Computer and System Sciences.*

[5] Inoue,K., I.Takanami, and H.Taniguchi, Two-dimensional alternating Turing machines, *Theoretical Computer Science*, 27:61-83, 1983.

[6] Inoue,K., I.Takanami, and J.Hromkovič, A leaf-size hierarchy of two-dimensional alternating Turing machines, *Theoretical Computer Science*, 67:99-110, 1989.

[7] Ito, A., K. Inoue, I. Takanami, and H. Taniguchi, Two-dimensional alternating Turing machines with only universal states, *Information and Control*, 55(1-3):193-221, 1982.

[8] Ito,A.,K.Inoue,I.Takanami,and Y.Inagaki, Constant leaf-size hierarchy of two-dimensional alternating Turing machines, *International Journal of Pattern Recognition and Artificial Intelligence*, 8(2):509-524, 1994.

[9] King,K.N., Measures of parallelism in alternating computation trees, *in Proc. 13th ACM Symposium on Theory of Computing*, pages 189-201,1981.

[10] Matsuno,H.,K.Inoue, I.Takanami, and H.Taniguchi, Alternating simple multi-head automata, *Theoretical Computer Science*, 36(2/3):291-308, 1985.

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

[12] Sakamoto, M., and K. Inoue, Three-dimensional alternating Turing machines with only universal states, *Information Sciences — Information and Computer Science*, 95:155-190, 1996.

[13] Sakamoto,M., K.Inoue, and A.Ito, A leaf-size hierarchy of three-dimensional alternating Turing machines, *Technical Report of the Yamaguchi University*, 5(5) :367-382, 1996.

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

[15] Yamamoto,H., Leaf reduction theorem on time- and leaf-bounded alternating Turing machines, *IEICE Transactions on Information and Systems*, E75-D(1):133-140, 1992.

[16] Yamamoto,Y., K.Morita, and K.Sugata, Space complexity for recognizing connected patterns in a three-dimensional tape (in Japanese), *Technical Report of the IECE*, AL79-104:91-96, 1979.

(Received August 2, 1999)