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

Makoto SAKAMOTO*, Katsushi INOUE** and Akira ITO**

(Received July 15, 1996)

Abstract

Alternating Turing machines were introduced as a generalization of nondeterministic Turing machines and as a mechanism to model parallel computation. "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. In this paper, we investigate a hierarchy of complexity classes based on leaf-size bounded computations for three-dimensional alternating Turing machines, and show that for any positive integer $k \geq 1$ and for any two functions $L : \mathbf{N} \rightarrow \mathbf{N}$ and $L' : \mathbf{N} \rightarrow \mathbf{N}$ such that (1) L is a three-dimensionally space -constructible function such that $L(m)^{k+1} \leq m$ $(m \geq 1)$, (2) $\lim_{m\rightarrow\infty} L(m)L'(m)^k/\log m = 0$ and (3) $\lim_{m\rightarrow\infty} L'(m)/L(m) = 0$, $L(m)$ space bounded and $L(m)^k$ leaf-size bounded three -dimensional alternating Turing machines are more powerful than $L(m)$ space bounded and $L'(m)^k$ leaf-size bounded three- dimensional alternating Turing machines. We let the input tapes, throughout this paper, be restricted to cubic ones.

1 Introduction

Alternating Turing machines were introduced in Chandra et al. [3] as a generalization of nondeterministic Turing machines and as a mechanism to model parallel computation. In related papers [3,4,6,7,11,15-18,22-24,30], several investigations of these automata have been continued.

After that, 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 computing abilities [1].
Morita et al. proposed an $L(m,n)$ space-bounded two-dimensional Turing machine and its variants to formalize memory limited computations in the two-dimensional information processing [19-21]. Inoue et al. [12] introduced two-dimensional alternating Turing machines (2-*ATM's*) as a generalization of two-dimensional nondeterministic Turing machines (2-*NTM's*). Moreover three-way two-dimensional alternating Turing machines (*TR*2-*ATM's*), which are restricted versions of the 2-*ATM's*, were investigated [13].

On the other hand, recently, due to the advances in computer vision, robotics and so forth, it has become increasingly apparent that the study of three- dimensional pattern

*Department of Shipping Technology, Oshima National College of Maritime Technology
**Department of Computer Sience and Systems Engineering, Faculty of Engineering, Yamaguchi University

processing should be very important. Thus, the research of three-dimensional automata as the computational model of three-dimensional pattern processing has also been meaningful [2,25-29,31,32]. In [27], we introduced a six-way three-dimensional alternating Turing machine (3-*ATM*) and a five-way three-dimensional alternating Turing machine (*FV*3-*ATM*), which are natural extentions of a 2-*ATM* and a *TR*2-*ATM*, respectively, to three dimensions. The motivation of introducing these three-dimensional machines is mainly from theoretical interest. We believe, however, that these three-dimensional machines are useful parallel models for analyzing three -dimensional images.

In this paper, we continue the investigations about three-dimensional alternating Turing machines described above, and mainly investigate a simple, natural complexity measure for space bounded three-dimensional alternating Turing machines, called " leaf-size", and provide a hierarchy of complexity classes based on leaf-size bounded computations. Specifically, we show that for any positive integer $k \geq 1$ and for any two functions $L : \mathbf{N} \rightarrow \mathbf{N}$ and $L' : \mathbf{N} \rightarrow \mathbf{N}$ such that (1) L is a three-dimensionally 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) $\lim_{m\to\infty} L'(m)/L(m) = 0$, $L(m)$ space bounded and $L(m)^k$ leaf-size bounded three-dimensional alternating Turing machines are more powerful than $L(m)$ space bounded and $L'(m)^k$ leaf-size bounded three- dimensional alternating Turing machines. We let the input tapes, throughout this paper, be restricted to cubic ones.

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 # ($\# \notin \Sigma$). 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 $x(i_1,i_2,i_3) = \#$. 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 (3-TM) [27,31].

In this paper, we assume that the reader is familiar with the concept of alternation. If necessary, see [3].

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) $U \subsetneq Q$ is the set of *universal states*,

(4) $F \subsetneq Q$ is the set of *accepting states*,

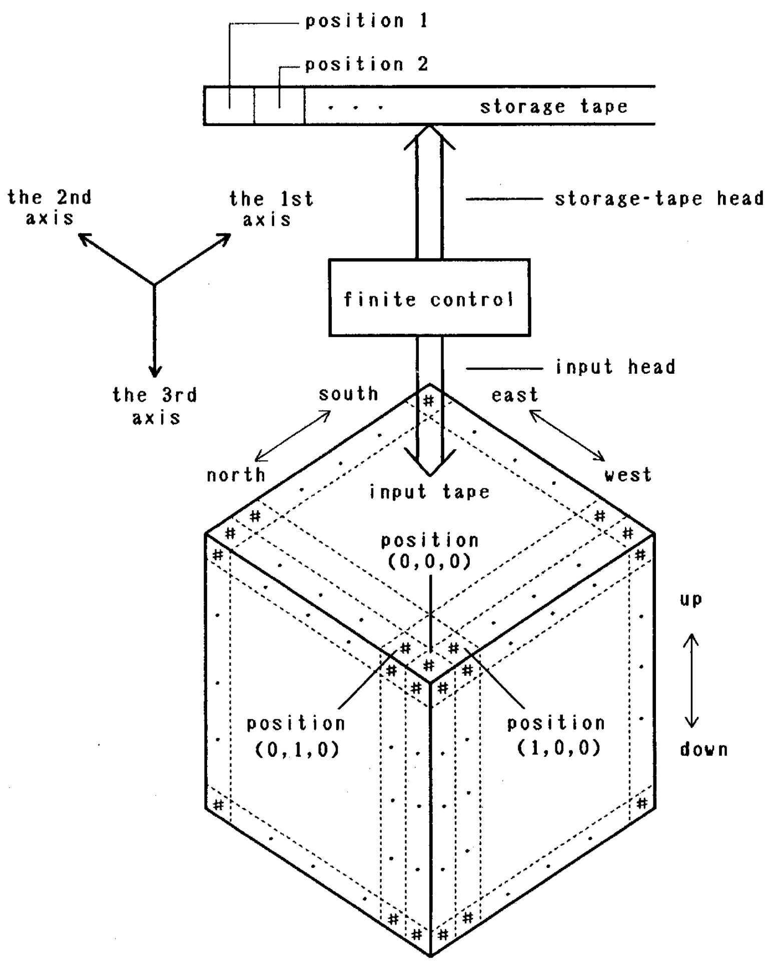

Fig. 1. Three-dimensional alternating Turing machine.

(5) Σ is a finite *input alphabet* ($\# \not\in \Sigma$ is the *boundary symbol*),
(6) Σ is a finite *storage-tape alphabet* ($B \in \Gamma$ is the *blank symbol*), and
(7) $\delta \subsetneq (Q \times (\Sigma \cup \{\#\}) \times \Gamma) \times (Q \times (\Gamma - \{B\}) \times$ {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*. As shown in Fig. 1, the machine M has a read-only three-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. 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. An *instantaneous description* (*ID*) of a 3-*ATM* $M = (Q, q_0, U, F, \Sigma, \Gamma, \delta)$ is an element of

$$\Sigma^{(3)} \times (N \cup \{0\})^3 \times S_M,$$

where $S_M = Q \times (\Gamma - \{B\})^* \times \mathbf{N}$, and $\mathbf{N}$ denotes the set of all positive integers.

The first component of an *ID* $I = (x, (i_1, i_2, i_3), (q, \alpha, k))$[1] represents the input to M. The second component (i_1, i_2, i_3) of I represents the input head position. The third component (q, α, k) of I represents the state of the finite control, nonblank contents of the storage tape, and the storage-head position. An element of S_M is called a *storage state* of M. If q is the state associated with an *ID* I, then I is said to be a *universal* (*existential, accepting*) *ID* if q is a universal (existential, accepting) state. The *initial ID* of M on x is

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

where λ is the null string.

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

$$I \vdash_M I'$$

and say I' is a successor of I if an *ID* I' follows from an *IDI* in one step of M, according to the transition rules δ. The relation $\vdash_M$ is not necessarily single-valued, because δ is not. A *computation path* of M on x is a sequence $I_0 \vdash_M I_1 \vdash_M \cdots \vdash_M I_n$ ($n \geq 0$), where $I_0 = I_M(x)$. A *computation tree* of M is a finite, nonempty labeled tree with the following properties :

(1) each node π of the tree is labeled with an *ID* $l(\pi)$;

---

[1]We note that $0 \leq i_1 \leq l_1(x)+1$, $0 \leq i_2 \leq l_2(x)+1$, $0 \leq i_3 \leq l_3(x)+1$, and $1 \leq k \leq |\alpha|+1$, where for any string w, $|w|$ denotes the length of w (with $|\lambda| = 0$, where λ is the null string).

(2) if π is an internal node (a nonleaf) of the tree, $l(\pi)$ is universal and

$$\{I \mid l(\pi) \vdash_M I\} = \{I_1, \cdots, I_k\},$$

then π has exactly k children $\rho_1 \cdots, \rho_k$ such that $l(\rho_i) = I_i$ $(1 \leq i \leq k)$;

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

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

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

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

In this paper, we are mainly concerned with a 3-ATM whose input tapes are restricted to cubic ones. We denote such a 3-ATM by 3-ATM^c.

Definition 2.4. Let $L(m): \mathbf{N} \rightarrow \mathbf{N}$ be a function with one variable m. With each 3-ATM^c M we associate a *space complexity function SPACE* which takes *ID's* to natural numbers. That is, for each *ID* $I = (x, (i_1, i_2, i_3), (q, \alpha, k))$, let $SPACE(I) = |\alpha|$. M is said to be $L(m)$ *space-bounded* if for each $m \geq 1$ and for each x with $l_1(x) = l_2(x) = l_3(x) = m$, 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, $SPACE(l(\pi)) \leq L(m)$. By 3-$ATM^c(L(m))$, we denote an $L(m)$ space-bounded 3-ATM^c [5,8,9].

Especially, 3-$ATM^c(0)$ is denoted by 3-AFA^c and called a *three-dimensional alternating finite automaton.*

We next present a simple, natural complexity measure for 3-ATM^c's, called *leaf-size* [10,12,14,17]. Basically, the leaf-size used by a 3-ATM^c 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 2.5. Let $Z(m): \mathbf{N} \rightarrow \mathbf{N}$ be a function with one variable m. For each finite 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^c M is $Z(m)$ *leaf-size bounded* if, for each m and for each input x with $l_1(x) = l_2(x) = l_3(x) = m$, each computation tree t of M on x is such that $LEAF(t) \leq Z(m)$.

By 3-$ATM^c(L(m), Z(m))$, we denote a simultaneously $L(m)$ space-bounded and $Z(m)$ leaf-size bounded 3-ATM^c. Especially, 3-$ATM^c(0, Z(m))$ is denoted by 3-$AFA^c(Z(m))$. Define £[3-$ATM^c(L(m), Z(m))$] $= \{T \mid T = T(M)$ for some 3-$ATM^c(L(m), Z(m))$ $M\}$, and £[3-$AFA^c(Z(m))$] $= \{T \mid T = T(M)$ for some 3-$AFA^c(Z(m))$ $M\}$.

We need the following concepts in the next section.

Definition 2.6. A three-dimensional *deterministic* Turing machine [8] is a 3-ATM^c

whose *ID's* each have at most one successor. A function $L : \mathbf{N} \rightarrow \mathbf{N}$ is three-dimensionally space constructible if there is a three-dimensional deterministic Turing machine M such that

(1) for each $m \geq 1$ and for each input tape x with $l_1(x) = l_2(x) = l_3(x) = m$, M uses at most $L(m)$ cells of the storage tape,

(2) for each $m \geq 1$, there exists some input tape x with $l_1(x) = l_2(x) = l_3 1(x) = m$ on which M halts after its read-write head has marked off exactly $L(m)$ cells of the storage tape, and

(3) for each $m \geq 1$, when given any input tape x with $l_1(x) = l_2(x) = l_3(x) = m$, M never halts without marking off exactly $L(m)$ cells of the storage tape.

(In this case, we say that M constructs the function L.)

Definition 2.7. Let Σ_1, Σ_2 be finite sets of symbols. A *projection* is a mapping $\tilde{\tau} : \Sigma_1^{(3)} \rightarrow \Sigma_2^{(3)}$ *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 $i (1 \leq i \leq 3)$, and (ii) $\tau(x(i_1,i_2,i_3)) = x'(i_1,i_2,i_3)$ for each (i_1,i_2,i_3) $(1 \leq i_1 \leq l_1(x),\ 1 \leq i_2 \leq l_2(x),\ 1 \leq i_3 \leq l_3(x))$. If $T \subseteq \Sigma_1^{(3)}$, we let $\tilde{\tau}(T) = \{\tilde{\tau}(x) \mid x \in T\}$.

Definition 2.8. Let $g : \mathbf{N} \rightarrow \mathbf{N}$ 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 $x[block_{g(n)}(k)]$.

Here, we give some mathematical notations. For any set A, "$P(A)$" denotes the power set of A and "m-$P_i(A)$" denotes the set of multisets consisting of i elements from A. We assume that any function is a mapping from $\mathbf{N}$ to $\mathbf{N}$.

3 Results

This section investigates a hierarchical property of the powers of space bounded 3-ATM^c's based on leaf-size bounded computations.

We first give several preliminaries to obtain the desired result. Let Σ be a finite alphabet. For each $m \geq 2$ and each $1 \leq n \leq m-1$, an (m,n)*- chunk* over Σ is a three-dimensional object over $\{0,1\}$ as shown in Fig. 2. (Below, "(m,n)-chunk" means an (m,n)-chunk over Σ.)

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 # be the boundary symbol of M. For any (m,n)-chunk x, we denote by $x(\#)$ the pattern (obtained from x by surrounding x with #'s) as shown in Fig. 3. Below we assume without loss of generality that for any (m,n)-chunk $(m \geq 2,\ 1 \leq n \leq m-1)$, M has the following property[2]:

(A) M enters or exits the pattern $x(\#)$ only at the shaded face in Fig. 3, and M never

[2]Note that for any 3-$ATM^c(l,z)$ M', we can construct a 3-$ATM^c(l,z)$ M with property (A) such that $T(M) = T(M')$.

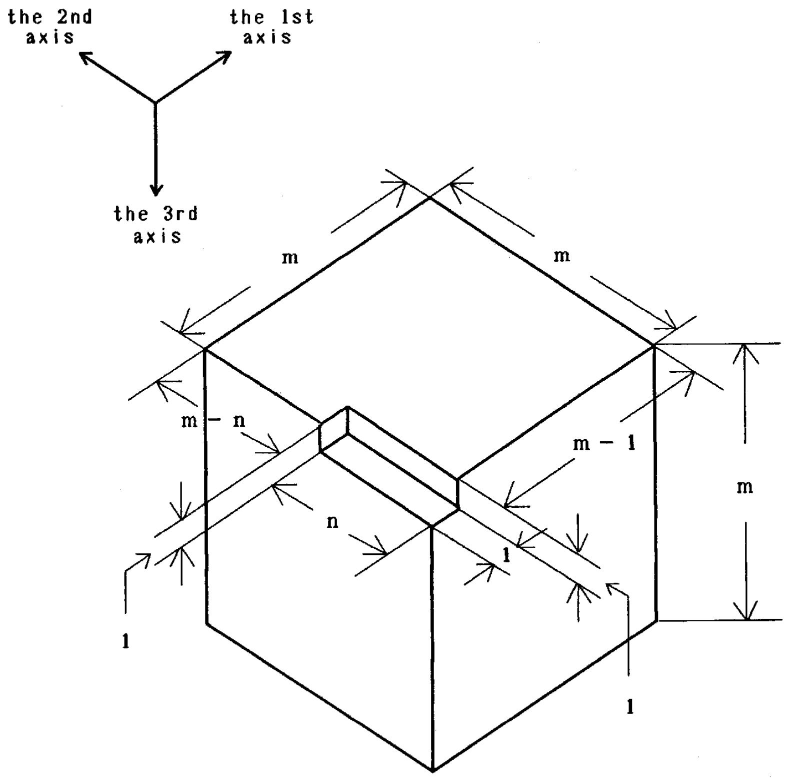

Fig. 2. An (m,n)-chunk.

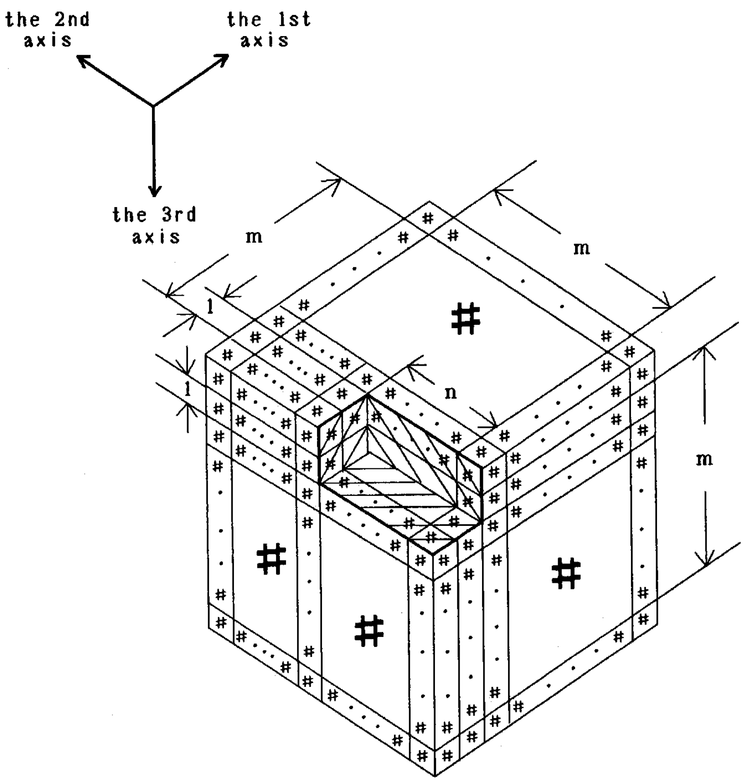

Fig. 3. An (m,n)-chunk surrounded by #'s.

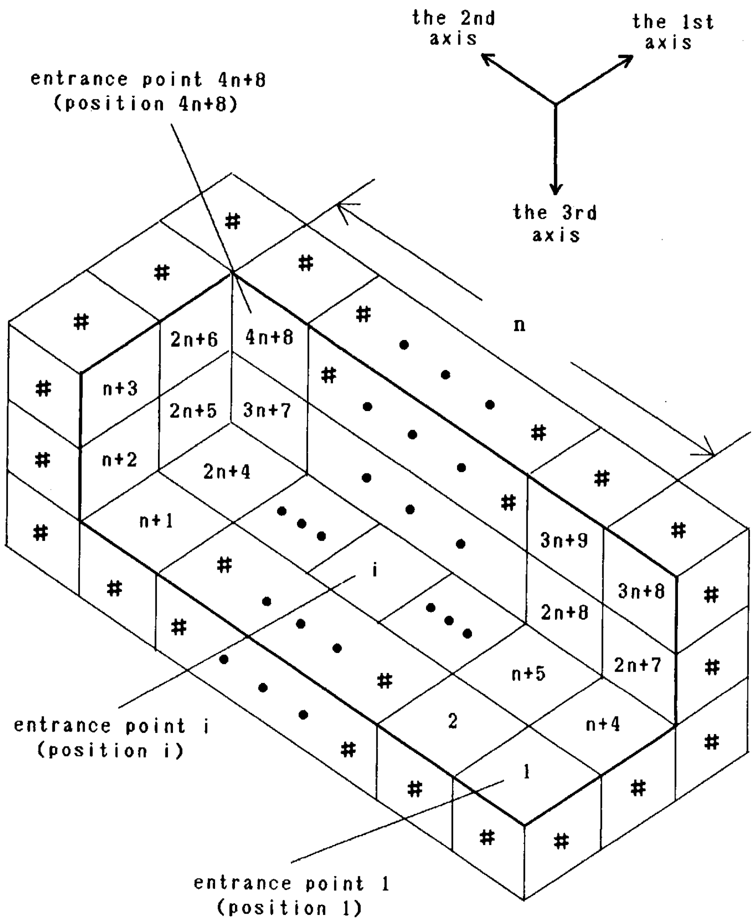

Fig. 4. Entrance points to x(#) and positioning of the cells of x(#).

enters an accepting state in $x(\#)$.

Then the number of entrance points to $x(\#)$ [or the exit points from $x(\#)$] for M is $4n+8$. We suppose that these entrance points (or exit points) are numbered $1,2,\cdots,4n+8$ as shown in Fig. 4. For each (m,n)-chunk x, an *ID* of M on $x(\#)$ is of the form

$$(x(\#),(p,(q,\alpha,k))),$$

where p represents the position of the head of M on $x(\#)$, and (q,α,k) represents a storage state of M. The second component $(p,(q,\alpha,k))$ of an *ID* $I=(x(\#),(p,(q,\alpha,k)))$ is called the *configuration component* of I. For convenience sake, for each i $(1 \leq i \leq 4n+8)$, let the position of the cell confronted with entrance point i of $x(\#)$ be "i" (see Fig. 4.) Further, as shown in Fig. 5, we consider $3n+4$ virtual cells (confronted with $x(\#)$) designated by dotted line cubes, and we assign position $1',2',\cdots,(3n+4)'$ to these virtual cells. We include these positions in the set of positions of the head of M on $x(\#)$.

An *ID* $I=(x(\#),(p,(q,\alpha,k)))$ is said to be *universal* (*existential*) if q is a universal (existential) state. For any *ID's* I and I' of M on $x(\#)$, 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 $x(\#)$. Note that for any *ID*

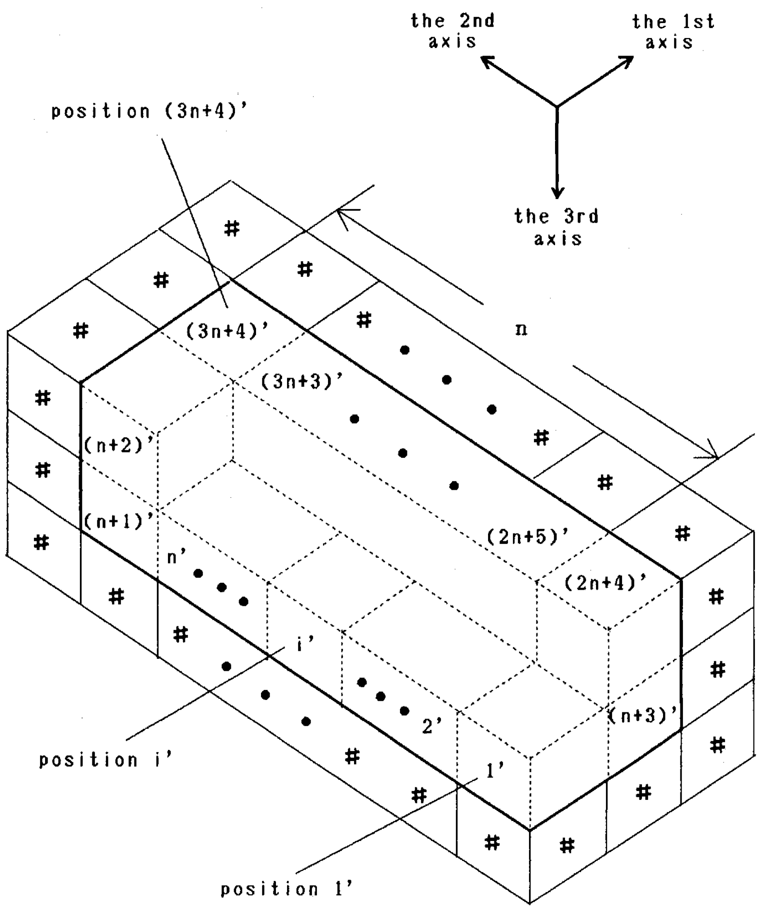

Fig. 5. Virtual cells of x(#) and positioning of virtual cells.

$I=(x(\#),(p,(q,\alpha,k)))$, where x is an (m,n)-chunk, such that $p \in \{1',2',\cdots,(3n+4)'\}$ (*i.e.*, p is a virtual position), I has no successor.

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

(1) each node π of the tree is labeled with an *ID*, $l(\pi)$, of M on $x(\#)$;

(2) if π is an internal node (a nonleaf) of the tree and $l(\pi)$ is universal and $\{I \mid l(\pi)\vdash_M I\}=\{I_1,\cdots,I_k\}$, then π has exactly k children $\rho_1\cdots\rho_k$ such tha $l(\rho_i)=I_i$;

(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)-chunk x is a computation tree of M on $x(\#)$ with the properties :

(1) the root node is labeled with an *ID* of the form $(x(\#),(i,(q,\alpha,k)))$, where $1 \leq i \leq 4n+8$ (i.e., the root node is labeled with an *ID* of M just after M entered the pattern $x(\#)$ from some entrance point i);

(2) each leaf node is labeled either

(a) with an *ID* of the form $(x(\#),(j,(q,\alpha,k)))$, where $j \in \{1',2',\cdots,(3n+4)'\}$ (i.e., an *ID* of M just after M exited the pattern $x(\#)$), or

(b) with an *ID I* such that starting from the *ID I, M* never reaches a universal

ID which has two or more successors and *M* never exists $x(\#)$.

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

Let $C = \{c_1, c_2, \cdots, c_u\}$ 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)-chunk, let the *leaf configuration set* of t (denoted by $LCS(t)$) be a "multiset" of elements of $\{1', 2', \cdots, (3n+4)'\} \times C \cup \{L\}$ (where L is a new symbol) defined as follows :

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

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

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

(Note that any prominent computation tree t of M, $|LCS(t)| \leq z$, since M is z leaf-size bounded.)[3]

For each (m,n)-chunk x and for each $(i,c) \in \{1,2,\cdots,4n+8\} \times C$, let

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

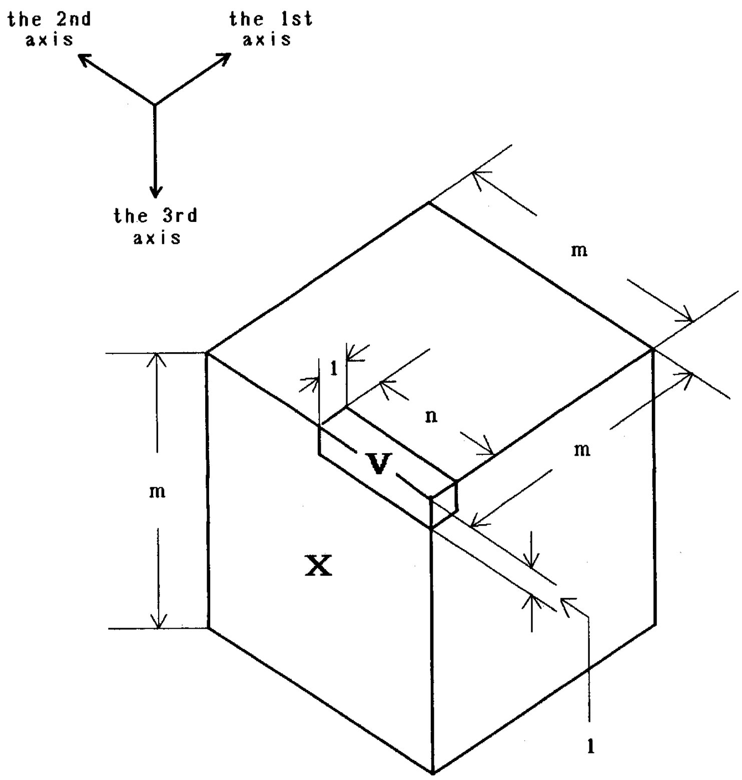

Fig. 6. x [V].

[3]For any set S, $|S|$ denotes the number of elements of S.

Let x,y be two (m,n)-chunks. We say that x and y are *M-equivalent* if for each $(i,c) \in \{1,2,\cdots,4n+8\}\times C$, $M_{(i,c)}(x) = M_{(i,c)}(y)$.

For any (m,n)-chunk x and for any tape $v \in \Sigma^{(3)}$ with $l_1(v)=1$, $l_2(v)=n$ and $l_3(v)=1$, let $x[v]$ be the tape in $\Sigma^{(3)}$ consisting of v and x as shown in Fig. 6.

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

Lemma 3.1. Let M be a 3-$ATM^c(l,z)$ with the property (A) described before, and Σ be the input alphabet of M. Let x and y be M-equivalent (m,n)-chunks over Σ $(m \geq 2,\ 1 \leq n \leq m-1)$. Then, for any tape $v \in \Sigma^{(3)}$ with $l_1(v)=1$, $l_2(v)=n$ and $l_3(v)=1$, $x[v]$ 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)-chunks, and we obtain the following lemma.

Lemma 3.2. Let M be a 3-$ATM^c(l,z)$ with the property (A) described before, 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)-chunks over Σ, where $b=(3n+4)u+1$ *and* $d=(4n+8)u$.

Proof. The lemma follows from the observation that

(1) $|\{1,2,\cdots,4n+8\}\times C| = (4n+8)u = d$ (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)-chunks is bounded by

$$b + b^2 + \cdots + b^z \leq b^{z+1} \text{ (where } b=(3n+4)u+1)$$

since M is z leaf-size bounded. ∎

We are now ready to prove the main theorem.

Theorem 3.3. 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 three-dimensionally 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 $\pounds[3\text{-}ATM^c(L(m),L(m)^k)]$, but not in $\pounds[3\text{-}ATM^c(L(m),L'(m)^k)]$.

Proof. Let M be a three-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 :

$T_k[L,M] = \{x \in (\Sigma \times \{0,1\})^{(3)} \mid \exists\, m \geq 2\, [l_1(x) = l_2(x) = l_3(x) = m$ & (when the tape $\tilde{h}(x)$ is presented to M, its read-write head marks off exactly $L(m)$ cells of the storage tape and then halts) & $\exists\, i\ (2 \leq i \leq m)[\tilde{h}_2(x[(1,1,1),(1,L(m)^{k+1},1)]) = [\tilde{h}_2(x[(i,1,1),(i,L(m)^{k+1},1)])]\}$,

where Σ is the input alphabet of M, and $\tilde{h}_1$ ($\tilde{h}_2$) is the projection which is obtained by extending the mapping $h_1 : \Sigma\times\{0,1\} \rightarrow \Sigma (h_2 : \Sigma\times\{0,1\}\rightarrow\{0,1\})$ such that for any $c = (a,b) \in \Sigma\times\{0,1\}$, $h_1(c)=a$, $(h_2(c)=b)$. Below, we shall show that $T_k[L,M] \in$ £$[3\text{-}ATM^c(L(m),L(m)^k)]$ and $T_k[L,M] \notin$ £$[3\text{-}ATM^c(L(m),L'(m)^k)]$.

The set $T_k[L,M]$ is accepted by a $3\text{-}ATM^c(L(m),L(m)^k)$ M_1 which acts as follows. Suppose that an input x with $l_1(x)=l_2(x)=l_3(x)=m$ $(m \geq 2)$ is presented to M_1. M_1 directly simulates the action of M on $\tilde{h}_1(x)$. 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 exactly $L(m)$ cells of the storage tape because M constructs the function L), then M_1 existentially chooses some $i (2 \leq i \leq m)$ and moves its input tape head on $x(i,1,1)$. After that, M_1 universally tries to check that, for each $1 \leq j \leq L(m)^k$,

$$\tilde{h}_2(x[(i,\ (j-1)L(m)+1,1),(i,jL(m),1)]) = \tilde{h}_2(x[(1,(j-1)L(m)+1,1),(1,jL(m),1)]).$$

That is, on $x(i,(j\text{-}1)L(m)+1,1)$ $(1 \leq j \leq L(m)^k)$, M_1 enters a universal state to choose one of two further actions. One action is to pick up and store the segment

$$\tilde{h}_2(x[(i,(j-1)L(m)+1,1),(i,jL(m),1)])$$

on some track of the storage tape (of course, M_1 uses exactly $L(m)$ cells marked off), to compare the segment stored above with the segment

$$\tilde{h}_2(x[(1,(j-1)L(m)+1,1),(1,jL(m),1)]),$$

and to enter an accepting state only if both segments are identical. The other action is to continue moving to $x(i,jL(m)+1,1)$ (in order to pick up the next segment

$$\tilde{h}_2(x[(i,jL(m)+1,1),(i,(j+1)L(m),1)]).$$

and compare it with the corresponding segment

$$\tilde{h}_2(x[(1,jL(m)+1,1),(1,(j+1)L(m),1)]).$$

Note that the number of pairs of segments which should be compared with each other in the future can be easily seen by using $L(m)$ 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 tree of M_1 on x with $L(m)^k$ leaves. Thus $T_k[L,M] \in$ £$[3\text{-}ATM^c(L(m),L(m)^k)]$.

We next show that $T_k[L,M] \notin$ £$[3\text{-}ATM^c(L(m),L'(m)^k)]$. Suppose that there is a $3\text{-}ATM^c(L(m),L'(m)^k)$ M_2 accepting $T_k[L,M]$. Let s and t be the numbers of states (of the finite control) and storage tape symbols of M_2, respectively. We assume without loss of generality that when M_2 accepts a tape x in $T_k[L,M]$, it enters an accepting state only on x(1,1,1), and that M_2 never falls off an input tape out of the boundary symbol #. (Thus M_2 satisfies the property (A) described before.) For each $m \geq 2$, *let* $w(m) \in \Sigma^{(3)}$ be a fixed tape such that

(i) $l_1(w(m)) = l_2(w(m)) = l_3(w(m)) = m$ and

(ii) when w(m) is presented to M, it marks off exactly $L(m)$ cells of the storage tape and halts.

(Note that for each $m \geq 2$, there exists such a tape $w(m)$ because M constructs the function L.) For each $m \geq 2$, let

$$V(m)=\{x \in (\Sigma \times \{0,1\})^{(3)} \mid l_1(x)=l_2(x)=l_3(x)=m \ \ \& \ \ \tilde{h}_2(x[(1,1,1),(m,L(m)^{k+1},1)]) \in \{0,1\}^{(3)} \& \text{ (the other part of } \tilde{h}_2(x) \text{ consists of 0's) } \& \ \tilde{h}_1(x)=w(m)\}^4,$$

$$Y(m)=\{y \in \{0,1\}^{(3)} \mid l_1(y)=1 \ \& \ l_2(y)=L(m)^{k+1} \& \ l_3(y)=1\},$$

and

$$R(m)=\{row(x) \mid x \in V(m)\},$$

where for each x in $V(m)$,

$$row(x)=\{y \in Y(m) \mid y=\tilde{h}_2(x[(i,1,1),(i,L(m)^{k+1},1)]) \text{ for some } i(2 \leq i \leq m)\}.$$

Since $|Y(m)|=2^{L(m)^{k+1}}$, it follows that

$$|R(m)| = \begin{cases} \binom{2^{L(m)^{k+1}}}{1}+\cdots+\binom{2^{L(m)^{k+1}}}{m-1}, & \text{if } 2^{L(m)^{k+1}}>m-1\,; \\ \binom{2^{L(m)^{k+1}}}{1}+\cdots+\binom{2^{L(m)^{k+1}}}{2^{L(m)^{k+1}}}=2^{2^{L(m)^{k+1}}}-1, & \text{otherwise.} \end{cases}$$

Note that $B = \{p \mid$ for some x in $V(m)$, p is the pattern obtained from x by cutting the segment $x[(1,1,1),(1,L(m)^{k+1},1)]$ off$\}$ is a set of $(m, L(m)^{k+1})$-chunks over $\Sigma \times \{0,1\}$. Since M_2 can use at most $L(m)$ cells of the storage tape and M_2 is $L'(m)^k$ leaf-size bounded when M_2 reads a tape in $V(m)$, from Lemma 3.2, there are at most

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

M_2-equivalence classes of $(m, L(m)^{k+1})$-chunks (over $\Sigma \times \{0,1\}$) in B, where $b[m]=(3L(m)^{k+1}+4)u[m]+1$, $d[m]=(4L(m)^{k+1}+8)u[m]$ and $u[m]=sL(m)t^{L(m)}$. We denote these M_2-equivalence classes by $C_1, C_2, \cdots C_{E(m)}$. Since $\lim_{m\to\infty} L(m)L'(m)^k/\log m = 0$ and $\lim_{m\to\infty} L'(m)/L(m) = 0$ (by assumption), it follows that for large m, $|R(m)|>E(m)$. For such m, there must be some $Q,Q'(Q \neq Q')$ in $R(m)$ and some C_i $(1 \leq i \leq E(m))$ such that the following statement holds : There exist two tapes x, y in $V(m)$ such that

(i) $x[(1,1,1),(1,L(m)^{k+1},1)] = y[(1,1,1),(1,L(m)^{k+1},1)]$ and $\tilde{h}_2(x[(1,1,1),(1,L(m)^{k+1},1)])=\tilde{h}_2(y[1,1,1),(1,L(m)^{k+1},1)])=\rho$ *for some* ρ *in* Q *but not in* Q',

(ii) $row(x)=Q$ *and* $row(y)=Q'$, and

(iii) both p_x and p_y are in C_i, where p_x (p_y) is the $(m,L(m)^{k+1})$-chunk over $\Sigma \times \{0,1\}$ obtained from x (from y) by cutting the segment $x[(1,1,1),(1,L(m)^{k+1},1)]$ (the segment $y[(1,1,1),(1,L(m)^{k+1},1)]$) off.

As is easily seen, x is in $T_k[L,M]$, and so x is accepted by M_2. Therefore, from Lemma 3.1, It follows that y is also accepted by M_2, which is a contradiction. (Note that y is not in $T_k[L,M]$.) Thus $T_k[L,M] \in \pounds[3\text{-}ATM^c(L(m),L'(m)^k)]$. This completes the proof of the theorem. ∎

---

[4]By the assumption that $L(m)^{k+1} \leq m(m \geq 1)$, $V(m)$ is well defined

Corollary 3.4. 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 satisfying the condition that $L'(m) \leq L(m)$ $(m \geq 1)$ and satisfying conditions (1),(2) and (3) described in Theorem 3.3. Then
$\pounds[3\text{-}ATM^c(L(m), L'(m)^k)] \subsetneq \pounds[3\text{-}ATM^c(L(m), L(m)^k)]$.
For each r in $\mathbf{N}$, let $\log^{(r)}m$ be the function defined as follows :

$$\log^{(1)}m = \begin{cases} 0 & (m=0) \\ \lceil \log m \rceil & (m \geq 1), \end{cases}$$
$$\text{lob}^{(r+1)}m = \log^{(1)}(\log^{(r)}m),$$

where $\lceil \log m \rceil$ denotes the smallest integer greater than or equal to $\log m$. As shown in Theorem 2.32 of [19], the function $\log^{(r)} m (r \geq 1)$ is two-dimensionally space -constructible, and thus three-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.
Corollary 3.5. For any $r \geq 2$ and any $k \geq 1$, $\pounds[3\text{-}ATM^c(log^{(r)}m, (log^{(r+1)}m)^k)] \subsetneq \pounds[3\text{-}ATM^c(log^{(r)}m, (log^{(r)}m)^k)]$.

4 Conclsion

In this paper, we have investigated a hierarchy of complexity classes based on leaf -size bounded computations for three-dimensional alternating Turing machines whose inputs are restricted to cubic ones ($3\text{-}ATM^c$'s). On the other hand, the accepting powers of leaf-size bounded computations for five-way three-dimensional alternating Turing machines whose inputs are restricted to cubic ones ($FV3\text{-}ATM^c$'s) are shown in [26]. $FV3\text{-}ATM^c$ is a $3\text{-}ATM^c$ wohse input head can move east, west, south, north, or down, but not up.

It is unknown whether a result analogous to Corollary 3.5 also holds for $r=1$ and $k \geq 1$. It will also be interesting to investigate leaf-size hierarchy properties of the classes of sets accepted by $3\text{-}ATM^c$'s with spaces of size greater than $\log m$.

Moreover it is possible to show the constant lea-size hierarchy of $3\text{-}ATM^c$'s by using the same technique as in the proof of Lemma 3.1 in [14], but this investigation will be dealt in a future paper.

References

[1] Blum, M. and Hewitt, C., Automata on a two-dimensional tape, IEEE Symp. on Switching and Automata Theory, pp.155-160, 1967.
[2] Blum, M. and W.J. Sakoda, On the capability of finite automata in 2 and 3 dimensional space, in "Proceedings, 18th Ann. Symp. on Fundations of Computer Sciences," pp.147-161, 1977.
[3] Chandra, A.K., D.C. Kozen and L.J. Stockmeyer, Alternation, J. ACM28 (1), pp.114-133, 1981.
[4] Gurari, E.M. and O.H. Ibarra, (Semi-) alternating stack automata, Math. Systems Theory 15, pp.211-224, 1982.
[5] Hopcroft, J.E. and J.D. Ullman, Introduction to automata theory, languages, and computation,

Addison-Wesley, Reading, Mass., 1979.

[6] Hromkovic, J. On the power of alternation in automata theory, J. Comput. Systems Sci. 31(1), pp.28-39, 1985.

[7] Inoue, K., A. Ito, I. Takanami and H. Taniguchi, A space-hierarchy result on two-dimensional alternating Turing machines with only universal states, Inform. Sci. 35, pp.79-90, 1985.

[8] Inoue, K. and I. Takanami, Three-way tape-bounded two-dimensional Turing machines, Inform. Sci. 17, pp.195-220, 1979.

[9] Inoue, K. and I. Takanami, A note on deterministic three-way tape-bounded two-dimensional Turing machines, Inform. Sci. 20, pp.41-55, 1980.

[10] Inoue, K., I. Takanami and J. Hromkovivc, A leaf-size hierarchy of two-dimensional alternating Turing machines, Theoret. Comput. Sci.67, pp.99-110, 1989.

[11] Inoue, K., I. Takanami and H. Taniguchi, A note on alternating on-line Turing machines, Inform. Process. Lett. 15(4), pp.164-168, 1982.

[12] Inoue, K., I. Takanami and H. Taniguchi, Two-dimensional alternating Turing machines, Theoret. Comput. Sci. 27, pp.61-83, 1983.

[13] Ito, A., K. Inoue and I. Takanami, A note on three-way two-dimensional alternating Turing machines, Inform. Sci. 45, pp.1-22, 1988.

[14] Ito, A., K. Inoue, I. Takanami and Y. Inagaki, Constant leaf-size hierarchy of two-dimensional alternating Turing machines, Inter. J. Pat. Recog. and Arti. Intelli. 8(2), pp.509-524, 1994.

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

[16] King, K.N., Alternating multihead finite automata, Lecture Notes in Computer Science 115 in " Automata, Languages, and Programming, Eight Colloquium," pp.506-520,1980.

[17] King, K.N., Measures of parallelism in alternating computation trees, in "Processings, 13th Ann. ACM, Symp. on Theory of Computing," pp.189-201,1981.

[18] Ladner, R.E., R.J. Lipton and L.J. Stockmeyer, Alternating pushdown automata, in " Proceedings, 19th IEEE Symp. on Foundations of Computer Science," 1978.

[19] Morita, K., Computational complexity in one- and two-dimensional tape automata, Ph.D. Thesis, Osaka Univ., 1978.

[20] Morita, K., H. Umeo, H. Ebi and K. Sugata, Lower bounds on tape complexity of two -dimensional tape Turing machines (in Japanese), IECE Japan Trans. D, pp.381-386, 1978.

[21] Morita, K., H. Umeo and K. Sugata, Computational complexity of $L(m,n)$ tape- bounded two -dimensional tape Turing machines (in Japanese), IECE of Japan Trans. (D), pp.982-989, 1977.

[22] Paul, W.J., E.J. Prauss and R. Reischuk, On alternation, Acta Inform. 14, pp. 243-255, 1980.

[23] Paul, W.J. and R. Reischuk, On alternation, Acta Inform. 14, pp. 391-403, 1980.

[24] Ruzzo, W.L., Tree-size bounded alternation, J. Comput. System Sci. 21, pp. 218-235, 1980.

[25] Sakamoto, M. and K. Inoue, Space Hierarchies of three-dimensional Turing machines, Tech. Rep. of the Yamaguchi Univ. 5(3), 1994.

[26] Sakamoto, M. and K. Inoue, Three-dimensional alternating Turing machines with only universal states, to appear in Information. Sciences.

[27] Sakamoto, M., K. Inoue and I. Takanami, A note on three-dimensional alternating Turing machines with space smaller than log m, Inform. Sci. 72, pp.225-249, 1993.

[28] Sakamoto, M., K. Inoue and I. Takanami, Three-dimensionally fully space constructible functions, IEICE Trans. Inf. & Syst. E77-D(6), 1994.

[29] Sakamoto, M., A. Ito, K. Inoue and I. Takanami, Simulation of three- dimensional one-marker automata by five-way Turing machines, Inform. Sci. 77, pp.77-99, 1994.

[30] Sudborough, I.H., Efficient algorithms for paths system programs and applications to alternating and time-space complexity classes, in "Proceedings, 21st Ann. Symp. on Foundations of Computer Science," pp.62-73, 1980.

[31] Taniguchi, H., K. Inoue and I. Takanami, A note on three-dimensional finite automata, Inform.

Sci. 26, pp.65-85,1982.
[32] Yamamoto, Y., K. Morita, and K. Sugata, Space complexity for recognition connected patterns in a three-dimensional tape (in Japanese), Tech. Rep. AL79-104, IECE of Japan, 1980.