数理解析研究所講究録
494巻 1983年 172-184

A Space Hierarchy Result of Two-Dimensional Alternating Turing Machines with Only Universal States

Katsushi INOUE (井上 克司)   Akira ITO (伊藤 暁)   Itsuo TAKANAMI (高浪 五男)   Hiroshi TANIGUCHI (谷口 弘)

(Faculty of Engineering, Yamaguchi University)

1. Introduction

It is shown [1-5] that there exists a hierarchy of the classes of languages accepted by deterministic or nondeterministic one-dimensional space-bounded Turing machines for ranges above $\log\log n$.

It is well-known [1,3] that the deterministic or nondeterministic one-dimensional Turing machines with space below $\log\log n$ accept only regular sets. On the other hand, for the two-dimensional case, as shown in [6], there exists an infinite hierarchy of the classes accepted by deterministic space-bounded Turing machines even below $\log\log n$.

This paper investigates a space hierarchy of the classes of sets accepted by "alternating" space-bounded two-dimensional Turing machines which have only universal states, and whose input tapes are restricted to square ones, and shows that there exists a dense hierarchy for the classes of sets accepted by these Turing machines with spaces less than or equal to $\log m$.

2. Preliminaries

Definition 2.1. Let Σ be a finite set of symbols. A two-dimensional tape over Σ is a two-dimensional rectangular array of elements of Σ.

The set of all two-dimensional tapes over Σ is denoted by $\Sigma^{(2)}$. Given a tape $x \in \Sigma^{(2)}$, we let $\ell_1(x)$ be the number of rows of x and $\ell_2(x)$ be the number of columns of x. If $1 \leq i \leq \ell_1(x)$ and $1 \leq j \leq \ell_2(x)$, we let x(i,j) denote the symbol in x with coordinates (i,j). Furthermore, we define

$$x[(i,j),(i',j')],$$

only when $1 \leq i \leq i' \leq \ell_1(x)$ and $1 \leq j \leq j' \leq \ell_2(x)$, as the two-dimensional tape z satisfying the following:

(i) $\ell_1(z)=i'-i+1$ and $\ell_2(z)=j'-j+1$;

(ii) for each k, r ($1 \leq k \leq \ell_1(z)$, $1 \leq r \leq \ell_2(z)$), $z(k,r)=x(k+i-1,r+j-1)$.

We now recall a two-dimensional alternating Turing machines introduced in [9].

Definition 2.2. A two-dimensional alternating Turing machine (2-ATM) is a seven-tuple

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

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 ($\# \notin \Sigma$ is the boundary symbol),

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

(7) $\delta \subseteq (Q \times (\Sigma \cup \{\#\}) \times \Gamma) \times (Q \times (\Gamma - \{B\}) \times \{\text{left,right,up,down,no move}\} \times \{\text{left,right,no move}\})$ 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 (rectangular) input tape with boundary symbols "#" and

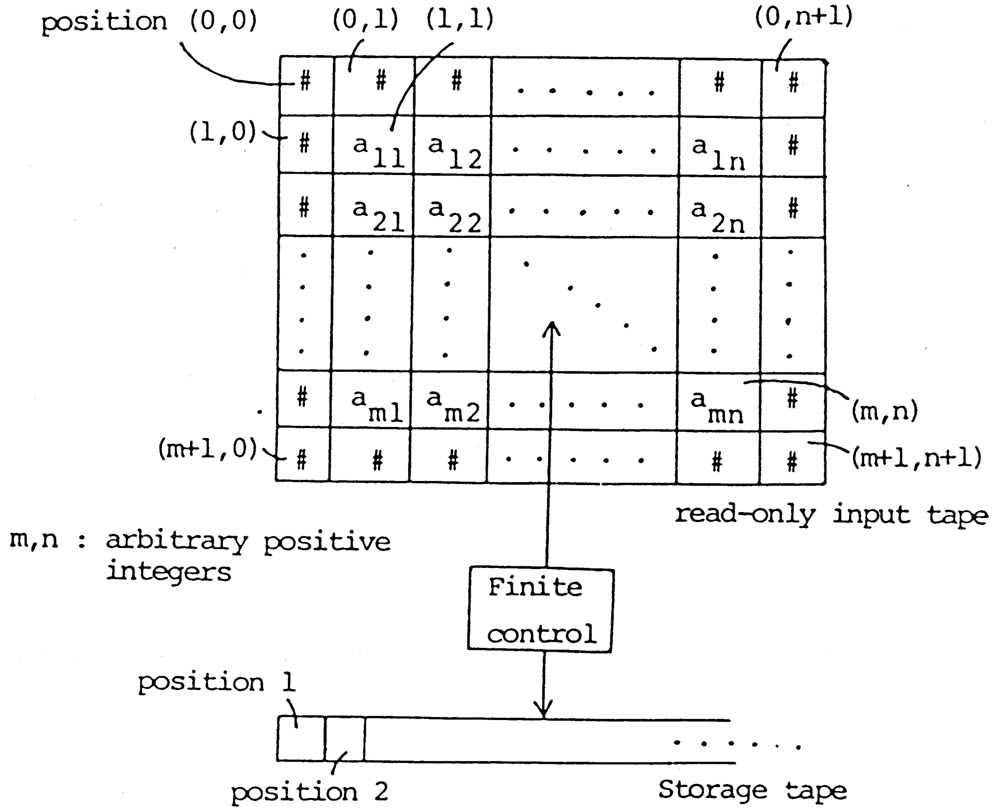

Fig.1. Two-dimensional alternating Turing machine.

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. A configuration of a 2-ATM $M=(Q,q_0,U,F,\Sigma,\Gamma,\delta)$ is an element of

$$\Sigma^{(2)} \times (N \cup \{0\})^2 \times S_M,$$

where $S_M = Q \times (\Gamma - \{B\})^* \times N$, and N denotes the set of all positive integers. The first component of a configuration $c=(x,(i,j),(q,\alpha,k))$‡ represents the input to M. The second component (i,j) of c represents the input head position. The third component (q,α,k) of c 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 configuration c, then c is said to be universal (existential, accepting) configuration if q is a universal (existential, accepting) state. The initial configuration of M on input x is

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

A configuration represents an instantaneous description of M at some point in a computation.

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 $\vdash_M$ is not necessarily single valued, since δ is not. The reflexive transitive closure of $\vdash_M$ is donoted $\overset{*}{\vdash}_M$. A computation path of M on x is a sequence $c_0 \vdash_M c_1 \vdash_M \cdots \vdash_M c_n$ $(n \geq 0)$, where $c_0 = I_M(x)$. A computation tree of M is a finite, nonempty labeled tree with the properties

(1) each node π of the tree is labeled with a configuration $\ell(\pi)$,

(2) if π is an internal node (a non-leaf) of the tree, $\ell(\pi)$ is universal and $\{c \mid \ell(\pi) \vdash_M c\} = \{c_1,\ldots,c_k\}$, then π has exactly k children $\rho_1,\ldots,\rho_k$ such that $\ell(\rho_i)=c_i$.

‡ We note that $0 \leq i \leq \ell_1(x)+1$, $0 \leq j \leq \ell_2(x)+1$, 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).

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

An accepting computation tree of M on an input x is a computation tree 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

$$T(M) = \{x \in \Sigma^{(2)} \mid M \text{ accepts } x\}.$$

In this paper, we mainly concerned with a 2-ATM which has only universal states, and whose input tapes are restricted to square ones.

We denote such a 2-ATM by 2-UTM^s. By 2-ATM^s we denote a 2-ATM whose input tapes are restricted to square ones.

Let $L:N \rightarrow R$ be a function with one variable m, where R denotes the set of all non-negative read numbers. With each 2-UTM^s (or 2-ATM^s) M we associate a space complexity function SPACE which takes configurations to natural numbers. That is, for each configuration $c=(x,(i,j),(q,\alpha,k))$, let SPACE(c)=$|\alpha|$. We say that M is L(m) space-bounded if for all m and for all x with $\ell_1(x)=\ell_2(x)=m$, 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($\ell(\pi)$) $\leq \lceil L(m) \rceil$[‡]. By 2-UTM^s(L(m)) (2-ATM^s(L(m)) we denote an L(m) space-bounded 2-UTM^s (2-ATM^s).

A two-dimensional deterministic Turing machine [7] is a 2-ATM whose configurations each have at most one successor. By 2-DTM^s(L(m)) we denote an L(m) space-bounded two-dimensional deterministic Turing machine whose input tapes are restricted to square ones. For each $X \in \{A,U,D\}$, define

$$\mathcal{L}[2\text{-}XTM^s(L(m))] = \{ T \mid T=T(M) \text{ for some } 2\text{-}XTM^s(L(m))\ M\}.$$

We need the following concepts in the next section.

---

‡ $\lceil r \rceil$ means the smallest integer greater than or equal to r.

Definition 2.5. A function $L:N\to R$ is two-dimensionally space constructable if there is a two-dimensional deterministic Turing machine M such that

(i) for each $m\geq 1$ and for each input tape x with $\ell_1(x)=\ell_2(x)=m$, M uses at most $\lceil L(m)\rceil$ cells of the storage tape,

(ii) for each $m\geq 1$, there exists some input tape x with $\ell_1(x)=\ell_2(x)=m$ on which M halts after its read-write head has marked off exactly $\lceil L(m)\rceil$ cells of the storage tape, and

(iii) for each $m\geq 1$, when given any input tape x with $\ell_1(x)=\ell_2(x)=m$, M never halts without marking off exactly $\lceil L(m)\rceil$ cells of the storage tape.

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

Definition 2.6. Let Σ_1, Σ_2 be finite sets of symbols. A projection is a mapping $\overline{\tau}:\Sigma_1^{(2)}\to\Sigma_2^{(2)}$ which is obtained by extending a mapping $\tau:\Sigma_1\to\Sigma_2$ as follows: $\overline{\tau}(x)=x' \Longleftrightarrow$ (i) $\ell_k(x)=\ell_k(x')$ for each k=1,2, and (ii) $\tau(x(i,j))=x'(i,j)$ for each $(i,j)\in\{(i,j)\mid 1\leq i\leq\ell_1(x) \text{ and } 1\leq j\leq\ell_2(x)\}$.

3. Results

It is well-known [6] that there is a dense hierarchy for the classes of sets of square tapes accepted by two-dimensional deterministic Turing machines with non-constant spaces. The main purpose of this section is to show that an analogous result also holds for 2-UTM^s's with spaces less than or equal to $\log m$.

We first give several preliminaries to get 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 pattern x over Σ as shown in Fig.2, where $x_1\in\Sigma^{(2)}$, $x_2\in\Sigma^{(2)}$, $\ell_1(x_1)=m-1$, $\ell_2(x_1)=n$, $\ell_1(x_2)=m$, and $\ell_2(x_2)=m-n$. Let M be a 2-UTM$^s(\ell)$. Note that if the numbers of states and storage tape symbols of M are s and t, respectively,

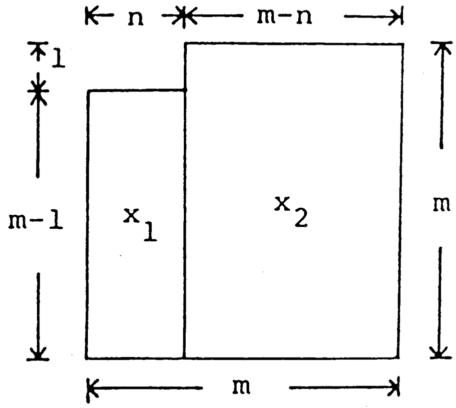

Fig.2. (m,n)-chunk.

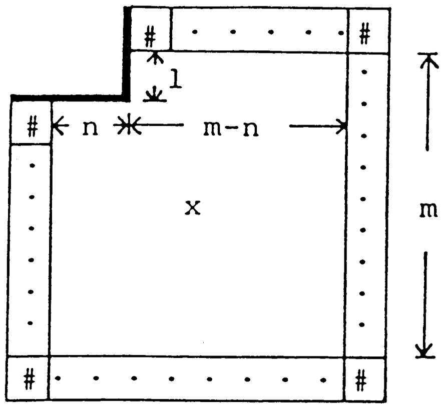

Fig.3.

then the number of possible storage states of M is $s\ell t^{\ell}$. Let Σ be the input alphabet of M, and let # be the boundary symbol of M. For any (m,n)-chunk x over Σ, we denote by x(#) the pattern (obtained from x by surrounding x by #s) as shown in Fig3. Below, we assume without loss of generality that for any (m,n)-chunk over Σ ($m \geq 2$, $1 \leq n \leq m-1$), M has the property (A)‡:

(A) M enters or exists the pattern x(#) only at the face designated by the bold line in Fig.3, and M never enters an accepting state in x(#).

Then the number of the entrance points to x(#) (or the exit points from x(#)) for M is n+3. We suppose that these entrance points (or exit points) are numbered 1,2,...,n+3 in an appropriate way. Let $P=\{1,2,\ldots,n+3\}$ be the set of these entrance points (or exists points). Let $C=\{q_1,q_2,\ldots,q_u\}$ be the set of possible storage storage states of M, where $u=s\ell t^{\ell}$. For each $i \in P$ and each $q \in C$, let $M_{(i,q)}(x(\#))$ be a subset of $P \times C \cup \{L\}$ which is defined as follows (L is a new symbol):

‡ Note that for any 2-UTM$^s(\ell)$ M', we can construct a 2-UTM$^s(\ell)$ M with the property (A) such that T(M)=T(M').

(1) $(j,p) \in M_{(i,q)}(x(\#))$

$\Leftrightarrow$ when M enters the pattern x(#) in storage state q at point i, there exists a sequence of steps of M in which M eventually exits x(#) in storage state p and at point j.

(2) $L \in M_{(i,q)}(x(\#))$

$\Leftrightarrow$ when M enters the pattern x(#) in storage state q and at point i, there exists a sequence of steps of M in which M never exists x(#). (Note the assumption that M never enters an accepting state in x(#).)

Let x, y be two (m,n)-chunks over Σ. We say that x and y are M-equivalent if for each $(i,q) \in P \times C$, $M_{(i,q)}(x(\#))=M_{(i,q)}(y(\#))$. For any (m,n)-chunk x over Σ and for any tape $v \in \Sigma^{(2)}$ with $\ell_1(v)=1$ and $\ell_2(v)=n$, let x[v] be the tape in $\Sigma^{(2)}$ consisting of v and x as shown in Fig.4.

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

Lemma 3.1. Let M be a 2-UTM(ℓ) with the property (A) described above, 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 $v \in \Sigma^{(2)}$ with $\ell_1(v)=1$ and $\ell_2(v)=n$, x[v] is accepted by M if and only if y[v] is accepted by M.

Proof. The lemma follows from the observation that there exists an accepting computation tree of M on x[v] if and only if there exists an accepting computation tree of M on y[v], since x and y are M-equivalent. Q.E.D.

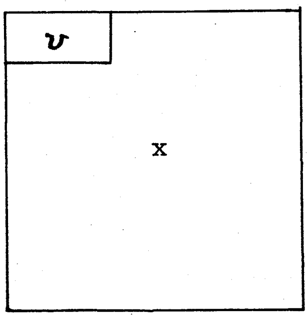

Fig.4. x[v]

Clearly, M-equivalence is an equivalence ralation on (m,n)-chunks, and we get the following lemma.

Lemma 3.2. Let M be a 2-UTM(ℓ) with the property (A) above, and Σ be the input alphabet of M. Then there are at most

$$(2^{(n+3)u+1})^{(n+3)u}$$

M-equivalence classes of (m,n)-chunks over Σ, where $u=s\ell t^{\ell}$, s is the number of states of the finite control of M, and t is the number of storage tape symbols of M.

Proof. The proof is similar to that of Lemma 2.1 in [8]. Q.E.D.

We are now ready to prove the following key lemma.

Lemma 3.3. Let $L:N\to R$ be a two-dimensionally space constructible function such that $L(m)\leq \log m$ $(m\geq 1)$, and M be a two-dimensional deterministic Turing machine which constructs the function L. Let T[L,M] be the following set, which depends on L and M:

$T[L,M]=\{x\in(\Sigma\times\{0,1\})^{(2)} \mid \exists m\geq 2[\ell_1(x)=\ell_2(x)=m$ & (when the tape $\bar{h}_1(x)$ is presented to M, its read-write head marks off exactly $\lceil L(m)\rceil$ cells of the storage tape and then halts) & $\exists i(2\leq i\leq m)[\bar{h}_2(x[(1,1),(1,\lceil L(m)\rceil)])=\bar{h}_2(x[(i,1),(i,\lceil L(m)\rceil)])]]\}$,

where Σ is the input alphabet of M, and $\bar{h}_1$ ($\bar{h}_2$) is the projection which is obtained by extending the mapping $h_1:\Sigma\times\{0,1\}\to\Sigma$ ($h_2:\Sigma\times\{0,1\}\to\{0,1\}$) such that for any $c=(a,b)\in\Sigma\times\{0,1\}$, $h_1(c)=a$ ($h_2(c)=b$). Then

(1) $T[L,M]\in\mathcal{L}[2\text{-DTM}^s(L(m))]$, and

(2) $T[L,M]\notin\mathcal{L}[2\text{-UTM}^s(L'(m))]$ for any function $L':N\to R$ such that $\lim_{m\to\infty}[L'(m)/L(m)]=0$.

Proof. (1): The set T[L,M] is accepted by a $2\text{-DTM}^s(L(m))$ M_1 which acts as follows. Suppose that an input x with $\ell_1(x)=\ell_2(x)=m$ $(m\geq 2)$ is presented to

M_1. First, M_1 directly simulates the action of M on $\bar{h}_1(x)$. (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 has marked off exactly $\lceil L(m)\rceil$ cells of the storage tape because M constructs the function L), then M_1 stores the segment $\bar{h}_2(x[(1,1),(1,\lceil L(m)\rceil)])$ on the storage tape. (Of course, M_1 uses exactly $\lceil L(m)\rceil$ cells marked off.) After that, M_1 simply checks that for some $i(2\le i\le m)$, $\bar{h}_2(x([(i,1),(i,\lceil L(m)\rceil)])$ is identical with $\bar{h}_2(x[(1,1),(1,\lceil L(m)\rceil)])$ stored on the storage tape, and M_1 accepts the input x if this check is successful. It will be obvious that $T(M_1)=T[L,M]$.

(2): Suppose that there is a 2-$UTM^s(L'(m))$ M_2 accepting T[L,M], where $\lim_{m\to\infty}[L'(m)/L(m)]=0$ (note that $L(m)\le \log m$ $(m\ge 1)$). 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[L,M], it enters an accepting state only on the upper left-hand corner of x, and that M_2 never falls off an input tape out of the boundary symbol #. (Thus, M_2 satisfies the property (A) above.) For each $m\ge 2$, let $z(m)\in\Sigma^{(2)}$ be a fixed tape such that (i) $\ell_1(z(m))=\ell_2(z(m))=m$ and (ii) when z(m) is presented to M, it marks off exactly $\lceil L(m)\rceil$ cells of the storage tape and halts. (Note that for each $m\ge 2$, there exists such a tape z(m) because M constructs the function L.) For each $m\ge 2$, let

$$V(m) = \{x \in (\Sigma\times\{0,1\})^{(2)} \mid \ell_1(x)=\ell_2(x)=m \;\&\; \bar{h}_2(x[(1,1),(m,\lceil L(m)\rceil)])\in \{0,1\}^{(2)} \;\&\; \bar{h}_2(x[(1,\lceil L(m)\rceil+1),(m,m)])\in\{0\}^{(2)} \;\&\; \bar{h}_1(x)=z(m)\},$$

$$Y(m) = \{y\in\{0,1\}^{(2)} \mid \ell_1(y)=1 \;\&\; \ell_2(y)=\lceil L(m)\rceil\}, \text{ and}$$

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

where for each x in V(m), $row(x)=\{y\in V(m) \mid y=\bar{h}_2(x[(i,1),(i,\lceil L(m)\rceil)])$ for some $i(2\le i\le m)\}$. Since $|Y(m)|$‡$=2^{\lceil L(m)\rceil}$, it follows that

‡ For any set S, |S| denotes the number of elements of S.

$$|R(m)| = \begin{cases} = \binom{2^{\lceil L(m)\rceil}}{1} + \binom{2^{\lceil L(m)\rceil}}{2} + \dots + \binom{2^{\lceil L(m)\rceil}}{m-1}, \text{ if } 2^{\lceil L(m)\rceil} \geq m-1; \\ = \binom{2^{\lceil L(m)\rceil}}{1} + \dots + \binom{2^{\lceil L(m)\rceil}}{2^{\lceil L(m)\rceil}} = 2^{2^{\lceil L(m)\rceil}} -1, \text{ otherwise.} \end{cases}$$

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

$$E(m)=(2^{(\lceil L(m)\rceil+3)u[m]}+1)^{(\lceil L(m)\rceil+3)u[m]}$$

M_2-equivalence classes of $(m,\lceil L(m)\rceil)$-chunks (over $\Sigma\times\{0,1\}$) in B, where u[m]= $sL'(m)t^{L'(m)}$. We denote these M_2-equivalence classes by $C_1, C_2, \dots, C_{E(m)}$. Since $L(m) \leq \log m$ 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) $x[(1,1),(1,\lceil L(m)\rceil)]=y[(1,1),(1,\lceil L(m)\rceil)]$, and $\overline{h}_2(x[(1,1),(1,\lceil L(m)\rceil)])=$ $\overline{h}_2(y[(1,1),(1,\lceil L(m)\rceil)])=\rho$ 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 $(m,\lceil L(m)\rceil)$-chunk over $\Sigma\times\{0,1\}$ obtained from x (from y) by cutting the part $x[(1,1),(1,\lceil L(m)\rceil)]$ (the part $y[(1,1),(1,\lceil L(m)\rceil)]$) off."

As is easily seen, x is in T[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[L,M].) This completes the proof of (2).

Q.E.D.

From Lemma 3.3, we can get the following main theorem.

Theorem 3.1. For any $L_1:N\to R$ and $L_2:N\to R$ such that (i) L_2 is a two-dimensionally space constructible function, (ii) $L_2(m)\le \log m$, and (iii) $\lim_{m\to\infty}[L_1(m)/L_2(m)]=0$, there is a set in $\mathcal{L}[2\text{-DTM}^s(L_2(m))]$, but not in $\mathcal{L}[2\text{-UTM}^s(L_1(m))]$.

Corollary 3.1. Let $L_1:N\to R$ and $L_2:N\to R$ be any functions satisfying the condition that $L_1(m)\le L_2(m)$ $(m\ge 1)$ and satisfying conditions (i) (ii) and (iii) described in Theorem 3.1. Then

(1) $\mathcal{L}[2\text{-DTM}^s(L_1(m))]\subsetneq \mathcal{L}[2\text{-DTM}^s(L_2(m))]$, and

(2) $\mathcal{L}[2\text{-UTM}^s(L_1(m))]\subsetneq \mathcal{L}[2\text{-UTM}^s(L_2(m))]$.

For each $k\in N$, let $\log^{(k)}m$ be the function defined as follows:

i) $\log^{(1)}m \begin{cases} =0 & (m=0) \\ =\lceil \log m\rceil & (m\ge 1) \end{cases}$

ii) $\log^{(k+1)}m=\log^{(1)}(\log^{(k)}m)$.

as shown in Theorem 3 in [6], the function $\log^{(k)}m$ $(k\ge 1)$ is two-dimensionally space constructible. It is easy to see that $\log^{(k+1)}m\le \log^{(k)}m$ $(m\ge 1)$ and $\lim_{m\to\infty}[\log^{(k+1)}m/\log^{(k)}m]=0$. From these facts and Corollary 3.1, we have

Corollary 3.2. For any $k\in N$,

(1) $\mathcal{L}[2\text{-DTM}^s(\log^{(k+1)}m)]\subsetneq \mathcal{L}[2\text{-DTM}^s(\log^{(k)}m)]$, and

(2) $\mathcal{L}[2\text{-UTM}^s(\log^{(k+1)}m)]\subsetneq \mathcal{L}[2\text{-UTM}^s(\log^{(k)}m)]$.

Remarks. It is shown [10] that $\mathcal{L}[2\text{-DTM}^s(L(m))]\subsetneq \mathcal{L}[2\text{-UTM}^s(L(m))]\subsetneq \mathcal{L}[2\text{-ATM}^s(L(m))]$ for any L such that $\lim_{m\to\infty}[L(m)/\log m]=0$. It is unknown whether a result analogous to Theorem 3.1 also holds for 2-ATM^s's. It will also be interensting to investigate a space hierarchy property of the classes of sets accepted by 2-ATM^s's (or 2-UTM^s's) with spaces greater than $\log m$.

REFERENCES

[1] J.Hartmanis, P.M.Lewis II, and R.E.Stearns, Hierarchies of memory limited computations, IEEE Conference Record of Switching Circuit Theory and Logical Design, p.179 (1965).

[2] J.E.Hopcroft and J.D.Ullman, Introduction to automata theory, languages and computation, Addison-Wesly, Reading, Mass., 1979.

[3] J.E.Hopcroft and J.D.Ullman, Some results on tape-bounded Turing machines, J.ACM, 16, p.168 (1967).

[4] J.I.Seiferas, Techniques for separating space complexity classes, J.Comp.Syst.Sci., 14, 73-99 (1977).

[5] J.I.Seiferas, Relating refined space complexity classes, J.Comp. Syst. Sci., 14, 100-129 (1977).

[6] K.Morita, H.Umeo, H.Ebi, and K.Sugata, Lower bounds on tape complexity of two-dimensional tape Turing machine. IECE of Japan Trans. (D), Jun. 1978, p.381.

[7] K.Inoue and I.Takanami, Three-way tape-bounded two-dimensional Turing machines, Information Sci., 17. 195-220 (1979).

[8] K.Inoue and I.Takanami, A note on closure properties of the classes of sets accepted by tape-bounded two-dimensional Turing machines, Information Sci., 15, 143-158 (1978).

[9] K.Inoue, I.Takanami and H.Taniguchi, Two-dimensional alternating Turing machines, Proc. 14th Ann. ACM Symp. on Theory of Computing, (May 1982) 37-46.

[10] K.Inoue, A.Ito, I.Takanami and H.Taniguchi, A note on Two-dimensional alternating Turing machines with only universal states, Technical Report No.AL82-45, IECE of Japan, 1982.