Two-Dimensional Alternating Turing Machines

Katsushi Inoue, Itsuo Takanami, and Hiroshi Taniguchi

Department of Electronics
Faculty of Engineering
Yamaguchi University
Ube, 755 Japan

ABSTRACT This paper introduces a two-dimensional alternating Turing machine (2-ATM) which is an extension of an alternating Turing machine to two-dimensions. This paper also introduces a three-way two-dimensional alternating Turing machine (TR2-ATM) which is an alternating version of a three-way two-dimensional Turing machine. We first investigate a relationship between the accepting powers of space-bounded 2-ATM's (or TR2-ATM's) and ordinary space-bounded two-dimensional Turing machines (or three-way two-dimensional Turing machines). We then introduce a simple, natural complexity measure for 2-ATM's (or TR2-ATM's), called "leaf-size", and provides a spectrum of complexity classes based on leaf-size bounded computations. We finally investigate the recognizability of connected patterns by 2-ATM's (or TR2-ATM's).

1. Introduction

During the past ten years, many automata on a two-dimensional tape have been introduced, and several properties of them have been given [1-9]. Recently, (one-dimensional) alternating Turing machines were introduced in [10] as a generalization of nondeterministic Turing machines and as a mechanism to model parallel computation. In papers [11-15], several investigations of alternating machines have been continued. It seems to us, however, that there are many problems about alternating machines to be solved in the future.

This paper introduces a two-dimensional alternating Turing machine (2-ATM) which is an alternating version of a two-dimensional Turing machine (TM) [3,6,7]. That is, a 2-ATM is a TM whose states are partitioned into "existential" and "universal" states, like one-dimensional alternating Turing machines. This paper also introduces a three-way two-dimensional alternating Turing machine (TR2-ATM) which is an alternating version of three-way two-dimensional Turing machine (TRTM) [7]. The main purpose of this paper is to get the deeper understanding of two-dimensional Turing machines through the investigations about these new machines. Section 2 gives terminology and notation necessary for this paper. It is well-known [10,11] that (one dimensional) alternating finite automata are equivalent to ordinary finite automata. It is unknown [10,11], however, whether or not (one-dimensional) space-bounded alternating Turing machines are more powerful than non-alternating versions corresponding to those machines. Section 3 investigates a relationship between the accepting powers of space-bounded 2-ATM's (TR2-ATM's) and space-bounded TM's (TRTM's), and shows that for some space-bounded classes, 2-ATM's (TR2-ATM's) are more powerful than TM's (TRTM's). Section 4 introduces a simple, natural complexity measure for 2-ATM's (or TR2-ATM's), called "leaf-size". The "leaf-size" used by a 2-ATM (or TR2-ATM) on a given input is the number of leaves of its accepting computation tree with fewest leaves. Leaf-size is a useful abstraction which provides a spectrum of complexity classes intermediate between nondeterminism and full alternation. The same section first provides a spectrum of complexity classes of TR2-ATM's, and then provides a relationship between the accepting powers of leaf-size bounded 2-ATM's and TR2-ATM's. In Section 5, we investigate recognizability of connected patterns by a 2-ATM (or TR2-ATM).

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 $1 \le i \le \ell_1(x)$ and $1 \le j \le \ell_2(x)$, we let $x(i,j)$ denote the symbol in x with coordinates (i,j). Furthermore, we define $x[(i,j),(i'.j')]$, only when $1 \le i \le i' \le \ell_1(x)$ and $1 \le j \le j' \le \ell_2(x)$, 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 $(1 \le k \le \ell_1(z), 1 \le r \le \ell_2(z))$,
$z(k,r)=x(k+i-1,r+j-1)$.

This paper assumes that the reader is familiar with fundamental knowledges about two-dimensional Turing machines [6,7]. We now introduce a two-dimensional alternating Turing machine, which can be considered as a natural extension of an alternating Turing machine [10,11,12] to two-dimensions. (We also assume that the reader is familiar with fundamental knowledges about one-dimensional alternating automata.)

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 one semi-infinite storage tape, initially blank. Of course, M has a finite control, an input tape 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

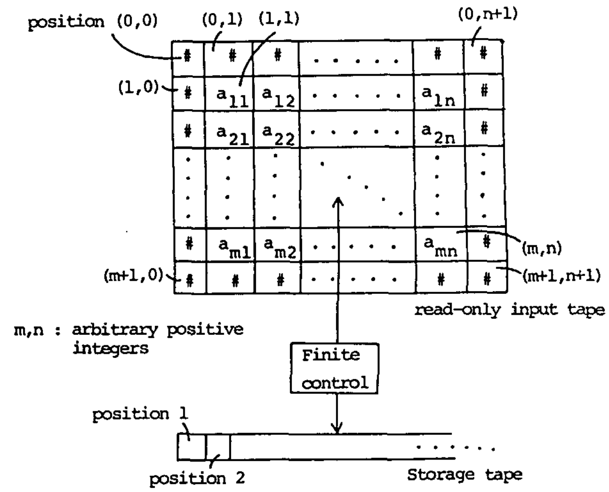

Fig.1. Two-dimensional alternating Turing machine.

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 more move.

Definition 2.3. A configuration of a 2-ATM $M=(Q,q_0,U,F,\Sigma,\Gamma,\delta)$ is a pair of an element of $\Sigma^{(2)}$ and an element of

$$C_M = Q \times (\Gamma - \{B\})^* \times (N \cup \{0\})^2 \times N,$$

where N denotes the set of all positive integers. The first component of a configuration $c = (x,(q,\alpha,(i,j),k))$ represents the input to M. The second component $(q,\alpha,(i,j),k)$‡ ($\in C_M$) of c represents the state of the finite control, the nonblank contents of the storage tape, the input head position, and the storage head position. An element of C_M is called a "semi-configuration 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 $I_M(x) = (x,(q_0,\lambda,(1,1),1))$, where λ is the null string. (A configuration represents an instantaneous description of M at some point in a computation.)

‡ We note that $0 \le i \le \ell_1(x)+1$, $0 \le j \le \ell_2(x)+1$, and $1 \le k \le |\alpha|+1$, where for any string w, $|w|$ denotes the length of w (with $|\lambda| = 0$).

Definition 2.4. Given $M=(Q,q_0,U,F,\Sigma,\Gamma,\delta)$, we write $c \vdash c'$ and say c' is a successor of c if configuration c' follows from configuration c in one step, according to the transition rules δ. The relation $\vdash$ is not necessarily single valued, since δ is not . A computation path of M on x is a sequence $c_0 \vdash c_1 \vdash \ldots \vdash c_n$ $(n \geq 0)$, 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 $\{c \mid \ell(\pi) \vdash c\}=\{c_1,\ldots,c_k\}$, then π has exactly k children $\rho_1,\ldots,\rho_k$ such that $\ell(\rho_i) = c_i$,

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

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

We next introduce a three-way two-dimensional alternating Turing machine which can be considered as an alternating version of a three-way two-dimensional Turing machine [7].

Definition 2.5. A three-way two-dimensional alternating Turing machine (TR2-ATM) is a 2-ATM $M = (Q,q_0,U,F,\Sigma,\Gamma,\delta)$ such that $\delta \subseteq (Q \times (\Sigma \cup \{\#\}) \times \Gamma) \times (Q \times (\Gamma - \{B\}) \times \{\text{left,right,down,no move}\} \times \{\text{left,right,no move}\})$. (That is, a TR2-ATM is a 2-ATM whose input head cannot move up.)

In this paper, we shall concentrate on investigating the properties of 2-ATM's and TR2-ATM's whose input tapes are restricted to square ones and whose storage tapes are bounded (in length) to use. Let $L(m): N \to R$ be a function with one variable m, where R denotes the set of all non-negative real numbers. With each 2-ATM (or TR2-ATM) M we associate a space complexity function SPACE which takes configurations to natural numbers. That is, for each configuration $c = (x,(q,\alpha,(i,j),k))$, let $\mathrm{SPACE}(c) = |\alpha|$. 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, $\mathrm{SPACE}(\ell(\pi)) \leq L(m)$‡. By $\text{2-ATM}^S(L(m))$ $(\text{TR2-ATM}^S(L(m)))$, we denote an L(m) space-bounded 2-ATM (TR2-ATM) whose input tapes are restricted to square ones. Define $\mathcal{L}[\text{2-ATM}^S(L(m))] = \{T \mid T=T(M)$ for some $\text{2-ATM}^S(L(m))$ $M\}$, $\mathcal{L}[\text{TR2-ATM}^S(L(m))] = \{T \mid T=T(M)$ for some $\text{TR2-ATM}^S(L(m))$ $M\}$. By using the well-known technique, it is easily proved that for any constant $k \geq 0$, $\mathcal{L}[\text{2-ATM}^S(k)] = \mathcal{L}[\text{2-ATM}^S(0)]$ and $\mathcal{L}[\text{TR2-ATM}^S(k)] = \mathcal{L}[\text{TR2-ATM}^S(0)]$. We especially denote a $\text{2-ATM}^S(0)$ $\text{TR2-ATM}^S(0))$ by 2-AFA^S (TR2-AFA^S). A 2-AFA^S (TR2-AFA^S) can be considered as an alternating version of a two-dimensional finite automaton [1,2,5] (three-way two-dimensional finite automaton [1,7]) whose input tapes are restricted to square ones. Deterministic and nondeterministic two-dimensional Turing machines (three-way two-dimensional Turing machines) [6,7] are special cases of 2-ATM's (TR2-ATM's). For example, a nondeterministic two-dimensional Turing machine is a 2-ATM which has no universal state, and a deterministic two-dimensional Turing machine is a 2-ATM whose configurations each have at most one successor. As in [7], by $\text{TM}^S(L(m))$ $(\text{DTM}^S(L(m))$, $\text{TRTM}^S(L(m))$, $\text{DTRTM}^S(L(m)))$ we denote an L(m) space-bounded# nondeterministic two-dimensional Turing machine (deterministic two-dimensional Turing machine, nondeterministic three-way two-dimensional Turing machine, deterministic three-way two-dimensional Turing machine) with square input tapes. (See [7] for definitions of these machines.) Furthermore, by 2-NA^S (TR2-NA^S) we denote a nondeterministic two-dimensional finite automaton (nondeterministic three-way two-dimensional finite automaton) with square input tapes. (See [2,7] for definitions of these machines.) Let $\mathcal{L}[\text{TM}^S(L(m))] = \{T \mid T=T(M)$ for some $\text{TM}^S(L(m))$ $M\}$. $\mathcal{L}[\text{DTM}^S(L(m))]$, $\mathcal{L}[\text{2-NA}^S]$, etc. are defined similarly.

3. A Relationship between Alternating and Non-alternating Machines

It is shown [10,11] that (one-dimensional) alternating finite automata are equivalent to ordinary

‡ Rigorously, "$\leq L(m)$" should be replaced with "$\leq \lceil L(m) \rceil$", where $\lceil r \rceil$ means the smallest integer greater than or equal to r. Below we omit $\lceil\ \rceil$, if no confusion occurs.

In [7], the term "L(m) tape-bounded" is used instead of the term "L(m) space-bounded".

finite automata. It is unknown [10,11], however, whether or not (one-dimensional) alternating space bounded Turing machines are more powerful than non-alternating versions.

We first show that for some space-bounded classes, 2-ATM's (TR2-ATM's) are more powerful than TM's (TRTM's). We give several preliminaries to get the desired result. For each $m \geq 2$ and each $1 \leq n \leq m-1$, a <u>(m,n)-chunk</u> is a pattern (over $\{0,1\}$) as shown in Fig.2, where $x_1 \in \{0,1\}^{(2)}$, $x_2 \in \{0\}^{(2)}$, $\ell_1(x_1)=m-1$, $\ell_2(x_1)=n$, $\ell_1(x_2)=m$, and $\ell_2(x_2)=m-n$. Let M be a $TM^S(\ell)$. 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 $s\ell t^{\ell}$. Let $\{0,1\}$ be the input alphabet of M, and # be the boundary symbol of M. For any (m,n)-chunk x, we denote by x(#) the pattern (obtained from x by surrounding x by #'s) as shown in Fig.3. Below, we assume without loss of generality that M enters or exits the pattern x(#) only at the face designated by the bold line in Fig.3. Thus, 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 exit points). Let $C = \{q_1,q_2,\ldots,q_u\}$ be the set of possible 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):

(1) $(j,p) \in M_{(i,q)}(x(\#))$
$\Leftrightarrow$ when M enters the pattern x(#) in storage state q and at point i, it may eventually exit 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, it may not exit x(#) at all.

Let x, y be any two different (m,n)-chunks. We say that x and y are <u>M-equivalent</u> if for any $(i,q) \in P \times C$, $M_{(i,q)}(x(\#)) = M_{(i,q)}(y(\#))$. Thus, M cannot distinguish between two (m,n)-chunks which are M-equivalent. Clearly, M-equivalence is an equivalence relation on (m,n)-chunks, and we get the following lemma.

‡For any two-dimensional Turing machine M, we define the storage state of M to be a combination of the (1) state of the finite control, (2) contents of the storage tape, and (3) position of the storage tape head within the nonblank portion of storage tape.

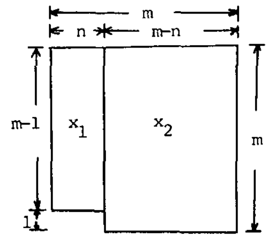

Fig.2.

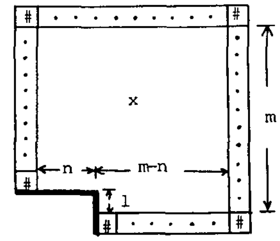

Fig.3.

<u>Lemma 3.1</u>. Let M be a $TM^S(\ell)$. There are at most
$$(2^{(n+3)u+1})^{(n+3)u}$$
M-equivalence classes of (m,n)-chunks, 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.

<u>Proof</u>. The proof is similar to that of Lemma 2.1 in [6]. Q.E.D.

We are now ready to prove the following lemma.

<u>Lemma 3.2</u>. Let $T_1=\{x \in \{0,1\}^{(2)} \mid \exists m \geq 2[\ell_1(x)=\ell_2(x)=m \;\&\; \exists i(1 \leq i \leq m-1)[x[(i,1),(i,m)]=x[(m,1),(m,m)]]]\}$. Then

(1) $T_1 \in \mathcal{L}[\text{TR2-AFA}^S]$ (thus, $\in \mathcal{L}[\text{2-AFA}^S]$);

(2) $T_1 \notin \mathcal{L}[TM^S(L(m))]$ for any $L(m):N \to R$ such that $\lim_{m\to\infty}[L(m)/\log m]=0$.‡

<u>Proof.</u> The set T_1 is accepted by a TR2-AFA^S M which acts as follows. Given an input x ($\ell_1(x)=\ell_2(x)=m \geq 2$), M existentially (i.e., in existential states) chooses some row, say the i-th row, of x. Then M universally (i.e., in universal states) tries to check that for each j ($1 \leq j \leq m$) $x(i,j)=x(m,j)$. That is, on the i-th row and the j-th column of x ($1 \leq j \leq m$), M enters a universal state to choose one of two further actions. One action is to pick up the symbol x(i,j), to move down with the symbol stored in the finite control, to compare the stored symbol with the symbol x(m,j), and to enter an accepting state if both symbols are identical. (It will be needless to say how M can pick up the symbol x(m,j).) The other action is to continue moving right one tape cell (in order to pick up the next symbol x(i,j+1) and compare it with the symbol x(m,j+1)). It will be obvious that $T(M)=T_1$. This completes the proof of part (1) of the lemma.

‡Below, let the base of logarithms be 2.

Suppose that there is a $TM^S(L(m))$ M accepting T_1, where $\lim_{m\to\infty}[L(m)/\log m]=0$. Let s and t be the numbers of states (of the finite control) and storage tape symbols of M, respectively. We assume without loss of generality that M starts on the lower left-hand corner of the input, and that when M accepts an input x in T_1, it halts on the lower left-hand corner of x (these assumptions are concerned with the shape of chunks described above), and that M never falls off an input out of the boundary symbol #. For each $n\geq 1$, let

$$V(n)=\{x\in\{0,1\}^{(2)} \mid \ell_1(x)=\ell_2(x)=2^n+1 \ \&\ x[(1,1),(2^n+1,n)]\in\{0,1\}^{(2)} \ \&\ x[(1,n+1),(2^n+1,2^n+1)]\in\{0\}^{(2)}\}$$

and

$$Y(n)=\{y\in\{0,1\}^{(2)} \mid \ell_1(y)=1 \ \&\ \ell_2(y)=n\}.$$

Clearly, $|Y(n)|=2^n$ (where for any set A, $|A|$ denotes the number of elements of A), and so we let $Y(n)=\{y_1,y_2,\ldots,y_{2^n}\}$. For each $n\geq 1$, let $R(n)=\{row(x) \mid x\in V(n)\}$, where for each x in V(n), $row(x)=\{y_j\in Y(n) \mid x[(i,1),(i,n)]$ is y_j for some i $(1\leq i\leq \ell_1(x)-1=2^n)\}$. Clearly,

$$|R(n)|=\binom{2^n}{1}+\binom{2^n}{2}+\binom{2^n}{2^n}=2^{2^n}-1.$$

Note that B={p| for some x in V(n), p is the pattern obtained from x by cutting the part $x[(2^n+1,1),(2^n+1,n)]$ off} is the set of all $(2^n+1,n)$-chunks. Since M can use at most $L(2^n+1)$ cells of the storage tape when M reads a tape in V(n), from Lemma 3.1, there are at most

$$E(n)=(2^{(n+3)u[n]}+1)^{(n+3)u[n]}$$

M-equivalence classes of $(2^n+1,n)$-chunks, where $u[n]=sL(2^n+1)t^{L(2^n+1)}$. We denote these M-equivalence classes by $C_1, C_2,\ldots, C_{E(n)}$. Since $\lim_{m\to\infty}[L(m)/\log m]=0$ (by assumption), $\lim_{n\to\infty}[L(2^n+1)/\log (2^n+1)]=0$, and so $\lim_{n\to\infty}[L(2^n+1)/n]=0$. By using this fact, it follows that for large n, $|R(n)|>E(n)$. For such n, there must be some Q, Q' ($Q\neq Q'$) in R(n) and some C_i $(1\leq i\leq E(n))$ such that the following statement holds.

"There exist two tapes x, y in V(n) such that

(i) for some row ρ in Q but not in Q',

$x[(2^n+1,1),(2^n+1,n)]=y[(2^n+1,1),(2^n+1,n)]=\rho$,

(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 $(2^n+1,n)$-chunk obtained from x (from y) by cutting the part $x[(2^n+1,1),(2^n+1,n)]$ (the part $y[(2^n+1,1),(2^n+1,n)]$) off ".

As is easily seen, x is in T_1, and so x is accepted by M. It follows that y is also accepted by M, which is a contradiction. (Note that y is not in T_1.) This completes the proof of (2) of the lemma.

Q.E.D.

Furthermore, we need the following two lemmas.

<u>Lemma 3.3</u>. Let $T_2=\{x\in\{0,1\}^{(2)} \mid \exists m\geq 2[\ell_1(x)=\ell_2(x)=m \ \&\ x[(1,1),(1,m)]=x[(2,1),(2,m)]]\}$. Then

(1) $T_2\in\mathcal{L}[TR2\text{-}AFA^S]$;

(2) $T_2\notin\mathcal{L}[TRTM^S(L(m))]$ for any $L(m):N\to R$ such that $\lim_{m\to\infty}[L(m)/m]=0$.

<u>Proof</u>. (1): The proof is similar to that of (1) of Lemma 3.2. (The details are left to the reader.)

(2): The proof is given in the proof of (2) of Lemma 3.1 in [7]. Q.E.D.

<u>Lemma 3.4</u>. Let $T_3=\{x\in\{0,1\}^{(2)} \mid \exists m\geq 1[\ell_1(x)=\ell_2(x)=2m \ \&\ x[(1,1),(m,2m)]=x[(m+1,1),(2m,2m)]]\}$. Then

(1) $T_3\in\mathcal{L}[TR2\text{-}ATM^S(\log m)]$;

(2) $T_3\notin\mathcal{L}[TRTM^S(L(m))]$ for any $L(m):N\to R$ such that $\lim_{n\to\infty}[L(m)/m^2]=0$.

<u>Proof</u>. (1): The set T_3 is accepted by a $TR2\text{-}ATM^S(\log m)$ M which, given an input x ($\ell_1(x)=\ell_2(x)=2m, m\geq 1$), simply checks by using universal states that for each i, j $(1\leq i\leq m,\ 1\leq j\leq 2m)$ $x(i,j) = x(m+i,j)$. The details of the action of M are again left to the reader.

(2): The proof is given in the proof of (2) of Lemma 3.2 in [7]. Q.E.D.

From Lemmas 3.2 through 3.4, we can get the following theorem.

<u>Theorem 3.1</u>. (1) $\mathcal{L}[TM^S(L(m))]\subsetneq\mathcal{L}[2\text{-}ATM^S(L(m))]$ for any $L(m):N\to R$ such that $\lim_{m\to\infty}[L(m)/\log m]=0$. (2) $\mathcal{L}[TRTM^S(L(m))]\subsetneq\mathcal{L}[TR2\text{-}ATM^S(L(m))]$ for any $L(m):N\to R$ such that (i) $\lim_{m\to\infty}[L(m)/m]=0$ or (ii) $L(m)\geq\log m$ and $\lim_{m\to\infty}[L(m)/m^2]=0$.

<u>Corollary 3.1</u>. (1) $\mathcal{L}[TR2\text{-}NA^S]\subsetneq\mathcal{L}[TR2\text{-}AFA^S]$.
(2) $\mathcal{L}[2\text{-}NA^S]\subsetneq\mathcal{L}[2\text{-}AFA^S]$.

Below, we shall be concerned with the problem of how much space is necessary and sufficient for deterministic three-way two-dimensional Turing machines to simulate $TR2\text{-}AFA^S$'s and $2\text{-}AFA^S$'s.

<u>Lemma 3.5</u>. Let $T_4=\{x\in\{0,1,2\}^{(2)} \mid \exists m\geq 1[\ell_1(x)=\ell_2(x)=2m \ \&\ \exists i(1\leq i\leq m)[x(i+m,1)=2 \ \&\ \forall(r,s)(\neq(i+m,1))[x(r,s)\in\{0,1\}] \ \&\ x[(i,2),(i,2m)]\neq x[(i+m,2),(i+m,2m)]]]\}$.

Then

(1) $T_4 \in \mathcal{L}[\text{TR2-AFA}^S]$ (thus, $\in \mathcal{L}[\text{2-AFA}^S]$);

(2) $T_4 \notin \mathcal{L}[\text{DTRTM}^S(L(m))]$ for any $L(m):N \to R$ such that $\lim_{m\to\infty}[L(m)/m^2]=0$.

Proof. (1): The set T_4 is accepted by a TR2-AFA^S M which acts as follows. Given an input x ($\ell_1(x)=\ell_2(x)=2m$, $m\geq 1$), on the upper left-hand corner of x, M enters a universal state to choose one of two further actions:

① One action is to check that there exists exactly one "2" only on the leftmost column of x. (Clearly, this check can be done deterministically.) If this check is successful, M enters an accepting state.

②. The other action is to existentially choose some i,j ($1\leq i\leq 2m$, $2\leq j\leq 2m$), to pick up the symbol x(i,j), and to store it in the finite control. Then M enters a universal state to choose one of two further actions:

(a) One action is to move right until it reaches the right boundary symbol #. Then M continues to move its input head H one cell down for every two left moves of H. M then enters an accepting state if H meets the symbol "2" (on the leftmost column).

(b) The other action is to existentially choose one of the following two actions, each time H meets a symbol which differs from the symbol x(i, j) stored in the finite control:

(i) One action is to continue moving down along the j-th column, seeking for another symbol different from x(i,j). (In this case, M will not enter an accepting state on the way.)

(ii) The other action is to move H to the left, and to check whether H meets the symbol 2. If so, M enters an accepting state.

It will be obvious that M accepts the set T_4.

(2): The proof is given in the proof of (2) of Lemma 3.5 in [9]. Q.E.D.

Let M be a 2-AFA^S, and s be the number of states of M. Given an input x with $\ell_1(x)=\ell_2(x)=m$, the number of possible configurations of M is $s(m+2)^2$, which is bounded by cm^2 for some constant c. From this, it is easily seen that if the input x is accepted by M, then there is an accepting computation tree of M on x whose computation paths from root to leaves each are of length at most cm^2. From this observation, it is easily ascertained that we can constract, by using the same idea as in the proof of Theorem 3.2 in [10], a DTM$^S(m^2)$ M' which, given an input with $\ell_1(x)=\ell_2(x)=m$, generates every possible computation path (of M on x) of length at most cm^2 in a systematic way, and checks whether there is an accepting computation tree of M on x. This implies that $\mathcal{L}[\text{2-AFA}^S]\subseteq\mathcal{L}[\text{DTM}^S(m^2)]$. In [7], it is shown that $\mathcal{L}[\text{DTM}^S(m^2)]=\mathcal{L}[\text{DTRTM}^S(m^2)]$. Therefore, we can get the following lemma.

Lemma 3.6. (1) $\mathcal{L}[\text{TR2-AFA}^S]\subseteq\mathcal{L}[\text{DTRTM}^S(m^2)]$.
(2) $\mathcal{L}[\text{2-AFA}^S]\subseteq\mathcal{L}[\text{DTRTM}^S(m^2)]$.

From Lemmas 3.5 and 3.6, we can get the following theorem.

Theorem 3.2. m^2 space is necessary and sufficient for DTRTM^S's to simulate TR2-AFA^S's and 2-AFA^S's.

Remark 3.1. By using the same idea as in Remark 2.2 of [11], we can easily show that for any $L(m)\geq \log m$ ($m\geq 1$), $\mathcal{L}[\text{TR2-ATM}^S(L(m))]=\mathcal{L}[\text{2-ATM}^S(L(m))]$.

4. Leaf-Size Bounded Alternation

In this section, we shall present a simple, natural complexity measure for TR2-ATM's (or 2-ATM's), called "leaf-size". (Recently [15], K.N.King introduced the same complexity measure as "leaf-size" independently. In [15], the term "branching" is adopted instead of the term "leaf-size".) Basically, the "leaf-size" used by a TR2-ATM (or 2-ATM) on a given input is the number of leaves of its accepting computation tree with fewest leaves. Leaf-size, in a sense, reflects the minimal number of processors which run in parallel in accepting a given input. One motivation for introducing leaf-size bounded computations below is to provide a restriction of a TR2-ATM (or 2-ATM) which is intermediate in power between nondeterministic and (full) alternating computations. A model of intermediate power can prove very useful in classifying problems and sharpening our intuitions about the relationships between various complexity classes. (The "tree-size bounded" in [12], in a sense, takes into account both time and the number of processors.)

Definition 4.1. Let $Z(m):N\to R$ be a function with one variable m. For each tree t, let LEAF(t) denote the leaf-size (i.e., the number of leaves of t). We say that a TR2-ATM^S (2-ATM^S) M is Z(m) leaf-size 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 t of M on x such that LEAF(t)$\leq$ Z(m) .

By TR2-ATM^S(L(m),Z(m)) (2-ATM^S(L(m),Z(m))), we denote a Z(m) leaf-size bounded TR2-ATM^S(L(m)) (2-ATM^S(L(m))). That is, for example, a TR2-ATM^S(L(m),Z(m)) is a simultaneously L(m) space- and Z(m) leaf-size bounded TR2-ATM^S. Let TR2-AFA^S(Z(m)) (2-AFA^S(Z(m))) denote a Z(m) leaf-size bounded TR2-AFA^S (2-AFA^S). Define $\mathcal{L}$[TR2-ATM^S(L(m),Z(m))]={T | T=T(M) for some TR2-ATM^S(L(m),Z(m)) M}. $\mathcal{L}$[2-ATM^S(L(m),Z(m))], $\mathcal{L}$[TR2-AFA^S(Z(m))], etc. are defined similarly.

4.1. Leaf-Size Bounded TR2-ATM^S's

We first provide a spectrum of complexity classes of TR2-ATM^S's, based on simultaneously space- and leaf-size bounded computations.

Lemma 4.1. For each k≥1, let $T[k]=\{x \in \{0,1\}^{(2)} \mid \exists m \geq k[\ell_1(x)=\ell_2(x)=m$ & (there exist exactly k 1's on the first row of x) & $x[(1,1),(1,m)]=x[(2,1),(2,m)]$]}. Then

(1) T[k]$\in \mathcal{L}$[TR2-AFA^S(k)];

(2) T[k+1]$\notin \mathcal{L}$[TR2-ATM^S(L(m),k)] for any $L(m):N\to R$ such that $\lim_{m\to\infty}[L(m)/\log m]=0$.

Proof. The proof of (1) is omitted here. Suppose that there is a TR2-ATM^S(L(m),k) M (with $\lim_{m\to\infty}[L(m)/\log m]=0$) accepting T[k+1]. We assume without loss of generality that M enters an accepting state only on the bottom boundary symbol #. Let r and s be the numbers of states (of the finite control) and storage tape symbols of M, respectively. For each accepting computation tree t of M, let SC(t) be a "multi-set" of semi-configurations of M defined as follows (see Definition 2.3 for semi-configurations):

SC(t)={(q,α,(i,j),i') | c=(x,(q,α,(i,j),i')) is a node label of t, and c is a configuration of M just after the point where the input head left the first row of x},

where x is the input associated with t. For each input x, let ACT(x) be the set of all accepting computation trees of M on x whose leaf-sizes are at most k. Furthermore, for each m≥k+1, let

$V(m)=\{x\in T[k+1] \mid \ell_1(x)=\ell_2(x)=m$ & $x[(3,1),(m,m)] \in\{0\}^{(2)}\}$

and for each x in V(m) let C(x)={SC(t) | t∈ACT(x)}. (Clearly, each tape x in V(m) is accepted by M, and so it follows, since we assumed that M enters an accepting state only on the bottom boundary symbol #, that for each x in V(m) C(x) is not empty.) Then the following proposition must hold.

Proposition 4.1. For any two different tapes x, y ∈V(m), C(x)∩C(y) = ∅ (empty set).

[For otherwise, suppose that C(x)∩C(y) ≠ ∅. Then there exist accepting computation trees t and t' in ACT(x) and ACT(y), respectively, such that SC(t) = SC(t'). We consider the tape z (with $\ell_1(z)=\ell_2(z)=m$) satisfying the following two conditions:

(i) z[(1,1),(1,m)]=x[(1,1),(1,m)];

(ii) z[(2,1),(m,m)]=y[(2,1),(m,m)].

Recalling that for any accepting computation tree t, of M SC(t) is a "multi-set", it is easily seen that one can construct, from the trees t and t', an accepting computation tree of M on z whose leaf-size is at most k. Thus, it follows that z is in T(M). This contradicts the fact that z is not in T[k+1].]

Let p(m) be the number of possible semi-configurations of M just after the input head left the first rows of tapes in V(m). Then

$$p(m) \leq r(m+2)L(m)s^{L(m)}.$$

Since for each x in V(m) and for each t in ACT(x) LEAF(t) is at most k, it follows that for each x in V(m) and for each t in ACT(x)

$$|SC(t)| \leq k.$$

Therefore, letting S(m)={SC(t) | t∈ACT(x) for some x in V(m)}, it follows that for some constants c and c',

$$|S(m)| \leq cp(m)^k \leq c'm^k L(m)^k s^{kL(m)}.$$

As is easily seen, $|V(m)|=\binom{m}{k+1}$. Since $\lim_{m\to\infty}[L(m)/\log m]=0$, we have |S(m)| < |V(m)| for large m. Therefore, it follows that for large m there must be different tapes x, y in V(m) such that C(x)∩C(y) ≠ ∅. This contradicts Proposition 4.1, and thus the part (2) of the lemma holds. Q.E.D.

From Lemma 4.1, we can get the following theorem.

Theorem 4.1. For any $L(m):N\to R$ such that $\lim_{m\to\infty}[L(m)/\log m]=0$ and for any integer k≥1,

$\mathcal{L}$[TR2-ATM^S(L(m),k)]$\subsetneq\mathcal{L}$[TR2-ATM^S(L(m),k+1)].

Corollary 4.1. For any integer k≥1,

$\mathcal{L}$[TR2-AFA^S(k)]$\subsetneq\mathcal{L}$[TR2-AFA^S(k+1)].

As shown in the next theorem, if L(m) ≥ log m, then a situation which differs from Theorem 4.1 emerges.

(The proof is omitted here.)

Theorem 4.2. For any $L(m) \geq \log m$ $(m \geq 1)$ and for any integer $k \geq 1$, $\mathcal{L}[\text{TR2-ATM}^S(L(m),k)] = \mathcal{L}[\text{TR2-ATM}^S(L(m),1)] = \mathcal{L}[\text{TRTM}^S(L(m))]$.

We need the following three definitions for the next theorem.

Definition 4.2. A function $L(m):N \to R$ is fully space constructible if there is a one-dimensional deterministic Turing machine M which, when given a string of length m, halts after its read-write head has visited exactly $\lceil L(m) \rceil$ tape cells of the sorage tape, where M has a read-only input tape with end markers and one semi-infinite storage tape [16].

Definition 4.3. A function $Z(m):N \to R$ is log-space countable if there is a one-dimensional deterministic Turing machine M which, when given a string of length m, halts after its read-write head has written down the k-adic notation of the number $\lceil Z(m) \rceil$, for some $k \geq 2$, by using at most $\lceil \log m + 1 \rceil$ cells of the storage tape, where M has again a read-only input tape with end markers and one semi-infinite storage tape.

Definition 4.4. Let x be a two-dimensional tape with $\ell_1(x)=\ell_2(x)=m$. As shown in Fig.4(a), let each tape cell of x be numbered $1,2,\ldots,m^2$ from top to bottom and from left to right on the same row. Then, for each $1 \leq i \leq j \leq m^2$, let $x<<i,j>>$ be the segment of x enclosed by the heavy solid line as shown in Fig. 4(b).

Theorem 4.3. Let $L(m):N \to R$, $Z_1(m):N \to R$, and $Z_2(m):N \to R$ be any functions such that

(i) $L(m) \geq \log m$ $(m \geq 1)$,

(ii) $L(m)$ is fully space-constructible,

(iii) $Z_2(m)$ is log-space countable,

(iv) $\lceil L(m) \rceil \lceil Z_2(m) \rceil \leq m^2/2$ $(m \geq 1)$,

(v) $Z_1(m) \leq Z_2(m)$ $(m \geq 1)$, and

(vi) $\lim_{m \to \infty} [Z_1(m)/Z_2(m)] = 0$. Then

$\mathcal{L}[\text{TR2-ATM}^S(L(m),Z_1(m))] \subsetneq \mathcal{L}[\text{TR2-ATM}^S(L(m),Z_2(m))]$.

Proof. Let $T[L,Z_2]$ be the following set depending on the functions $L(m)$ and $Z_2(m)$ in the theorem.

$$T[L,Z_2]=\{x \in \{0,1\}^{(2)} \mid \exists m \geq 1[\ell_1(x)=\ell_2(x)=2m \ \& \ x<<1, \lceil L(2m) \rceil \lceil Z_2(2m) \rceil>>=x<<2m^2+1, 2m^2+\lceil L(2m) \rceil \lceil Z_2(2m) \rceil>>]\}.$$

(Note that, from the condition (iv) in the theorem, this set can be well defined.) The set $T[L,Z_2]$ is accepted by a $\text{TR2-ATM}^S(L(m),Z_2(m))$ M which acts as follows. Suppose that an input x with $\ell_1(x)=\ell_2(x)=2m$ $(m \geq 1)$ is presented to M. While moving on the first row of x, M first marks off exactly $\lceil L(2m) \rceil$ cells of the storage tape by using the number 2m of columns. While again moving on the first row, M then writes down the k-adic notation (for some $k \geq 2$) of the number $\lceil Z_2(2m) \rceil$ on one track of the storage tape by using the number 2m of columns. (These actions are possible because of conditions (i), (ii), and (iii) in the theorem.) After that, M universally tries to check that, for each $1 \leq i \leq \lceil Z_2(2m) \rceil$, $x<<(i-1)\lceil L(2m) \rceil+1, i\lceil L(2m) \rceil>>=x<<2m^2+(i-1)\lceil L(2m) \rceil+1, 2m^2+i\lceil L(2m) \rceil>>$. That is, on the cell numbered $(i-1)\lceil L(2m) \rceil+1$ of x $(1 \leq i \leq \lceil Z_2(2m) \rceil)$, M enters a universal state to choose one of two further actions. One action is to pick up and store the segment $x<<(i-1)\lceil L(2m) \rceil+1, i\lceil L(2m) \rceil>>$ on some track of the storage tape (of course, M uses the cells marked off above of the storage tape), to move its input head to the cell numbered $2m^2+(i-1)\lceil L(2m) \rceil+1$ of x, to compare the segment stored above with the segment $x<<2m^2+(i-1)\lceil L(2m) \rceil+1, 2m^2+i\lceil L(2m) \rceil>>$, and to enter an accepting state if both segments are identical. The other action is to continue moving to the cell numbered $i\lceil L(2m) \rceil+1$ (in order to pick up the next segment $x<<i\lceil L(2m) \rceil+1, (i+1)\lceil L(2m) \rceil>>$ and compare it with the corresponding segment $x<<2m^2+i\lceil L(2m) \rceil+1, 2m^2+(i+1)\lceil L(2m) \rceil>>$). Note that the number of pairs of segments which should be compared each other in the future can be seen by updating the k-adic notation of $\lceil Z_2(2m) \rceil$. Note also that the position-information of the input head can be obtained

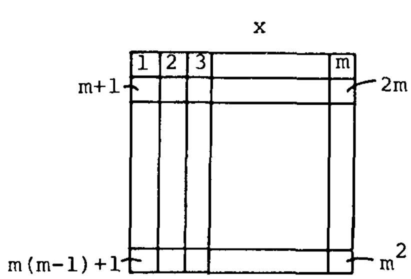

Fig.4(a). The numbering of tape cells of x.

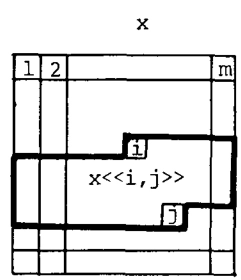

Fig.4(b). Illustration of $x<<i,j>>$.

by using one track of length 2m. It will be obvious that the input x is in $T[L,Z_2]$ if and only if there is an accepting computation tree of M on x with $\lceil Z_2(2m)\rceil$ leaves. Thus $T[L,Z_2] \in \mathcal{L}[\text{TR2-ATM}^S(L(m),Z_2(m))]$.

We next show that $T[L,Z_2]$ is not in $\mathcal{L}[\text{TR2-ATM}^S(L(m),Z_1(m))]$. Suppose that there is a $\text{TR2-ATM}^S(L(m),Z_1(m))$ M accepting $T[L,Z_2]$. We assume without loss of generality that M enters an accepting state only on the bottom boundary symbol #. Let r and s be the numbers of states (of the finite control) and storage tape symbols of M, respectively. For each accepting computation tree t of M, let SC(t) be a multi-set of semi-configurations of M defined as follows:

$SC(t)=\{(q,\alpha,(i,j),i') \mid c=(x,(q,\alpha,(i,j),i'))$ is a node label of t, and c is a configuration of M just after the point where the input head left the top half of x$\}$,

where x is the input associated with t. For each x with $\ell_1(x)=\ell_2(x)=2m$ $(m\geq 1)$, let ACT(x) be the set of all accepting computation trees of M on x whose leaf-sizes are at most $Z_1(2m)$. For each $m\geq 1$, let

$V(m)=\{x\in\{0,1\}^{(2)} \mid \ell_1(x)=\ell_2(x)=2m$ & $x<<1,\lceil L(2m)\rceil\times\lceil Z_2(2m)\rceil>>=x<<2m^2+1,2m^2+\lceil L(2m)\rceil\lceil Z_2(2m)\rceil>>$ & $x<<\lceil L(2m)\rceil\lceil Z_2(2m)\rceil+1,2m^2>>=x<<2m^2+\lceil L(2m)\rceil\lceil Z_2(2m)\rceil+1,4m^2>>\in\{0\}^{(2)}\}$

and for each x in V(m), let $C(x)=\{SC(t) \mid t\in ACT(x)\}$. (Clearly, each tape in V(m) is in $T[L,Z_2]$, and so it is accepted by M. Thus, it follows, since we assumed that M enters an accepting state only on the bottom boundary symbol #, that for each x in V(m) C(x) is not empty.) Then the following proposition must hold.

<u>Proposition 4.2.</u> For any two different tapes x, y in V(m), $C(x)\cap C(y)=\emptyset$.

[For otherwise, suppose that $C(x)\cap C(y)\neq\emptyset$. Then there exist accepting computation trees t and t' in ACT(x) and ACT(y), respectively, such that $SC(t)=SC(t')$. We consider the tape z (with $\ell_1(z)=\ell_2(z)=2m$) satisfying the following two conditions:

(i) $z[(1,1),(m,2m)]=x[(1,1),(m,2m)]$;

(ii) $z[(m+1,1),(2m,2m)]=y[(m+1,1),(2m,2m)]$.

It is easily seen that one can construct, from the trees t and t', an accepting computation tree of M on z whose leaf-size is at most $Z_1(2m)$. Thus, it follows that z is in T(M). This contradicts the fact that z is not in $T[L,Z_2]$.]

Let p(m) be the number of possible semi-configurations of M just after the input head left the top halves of tapes in V(m). Then

$$p(m)\leq r(2m+2)L(2m)s^{L(2m)}.$$

Since for each x in V(m) and for each t in ACT(x) LEAF(t) is at most $Z_1(2m)$, it follows that for each x in V(m) and for each t in ACT(x)

$$|SC(t)|\leq Z_1(2m).$$

Therefore, letting $S(m)=\{SC(t)\mid t\ ACT(x)$ for some x in $V(m)\}$, it follows that for some constants c and c',

$$|S(m)|\leq cp(m)^{Z_1(2m)} \leq c'm^{Z_1(2m)}L(2m)^{Z_1(2m)}s^{L(2m)Z_1(2m)}.$$

As is easily seen, $|V(m)|=2^{\lceil L(2m)\rceil\lceil Z_2(2m)\rceil}$. From the conditions (i) and (vi) in the theorem, we have $|S(m)|<|V(m)|$ for large m. Therefore, it follows that for large m there must be different tapes x, y in V(m) such that $C(x)\cap C(y)\neq\emptyset$. This contradicts Proposition 4.2, and thus it follows that $T[L,Z_2]\notin\mathcal{L}[\text{TR2-ATM}^S(L(m),Z_1(m))]$. From the condition (v) in the theorem, it directly follows that $\mathcal{L}[\text{TR2-ATM}^S(L(m),Z_1(m))]\subseteq\mathcal{L}[\text{TR2-ATM}^S(L(m),Z_2(m))]$. This completes the proof of the theorem. Q.E.D.

<u>Remark 4.1.</u> The condition (iv) in Theorem 4.3 can be replaced with the following condition (iv)':

(iv)' For some constant $k>0$, $\lceil L(m)\rceil\lceil Z_2(m)\rceil\leq km^2$ $(m\geq 1)$.

<u>Remark 4.2.</u> We can show that a similar result to Theorem 4.2 holds for 2-ATM^S's. It is unknown, however, whether the similar results to Theorems 4.1 and 4.3 hold for 2-ATM^S's.

4.2. <u>Leaf-Size Bounded TR2-ATM^S's versus 2-ATM^S's</u>

We next investigate a relationship between the accepting powers of lesf-size bounded TR2-ATM^S's and 2-ATM^S's.

<u>Theorem 4.4.</u> For any $L(m):N\to R$ and $Z(m):N\to R$ such that $\lim_{m\to\infty}[L(m)Z(m)/m]=0$ and $\lim_{m\to\infty}[Z(m)\log m/m]=0$, $\mathcal{L}[\text{TR2-ATM}^S(L(m),Z(m))]\subsetneq\mathcal{L}[2\text{-ATM}^S(L(m),Z(m))]$.

<u>Proof.</u> We can prove the theorem by showing that the set T_2 described in Lemma 3.3 is in $\mathcal{L}[2\text{-ATM}^S(0,1)]$, but not in $\mathcal{L}[\text{TR2-ATM}^S(L(m),Z(m))]$. The details are omitted here. Q.E.D.

Theorem 4.5. For any $L(m):N\to R$ and $Z(m):N\to R$ such that $L(m)\geq \log m$ $(m\geq 1)$ and $\lim_{m\to\infty}[L(m)Z(m)/m^2]=0$,

$$\mathcal{L}[\text{TR2-ATM}^S(L(m),Z(m))]\subsetneq\mathcal{L}[\text{2-ATM}^S(L(m),Z(m))].$$

Proof. We can prove the theorem by showing that the set T_3 described in Lemma 3.4 is in $\mathcal{L}[\text{2-ATM}^S(\log m,1)]$, but not in $\mathcal{L}[\text{TR2-ATM}^S(L(m),Z(m))]$. The details are omitted here. Q.E.D.

Theorem 4.6. If $L(m)\geq m^2$ $(m\geq 1)$, then for any $Z(m):N\to R$,

$$\mathcal{L}[\text{TR2-ATM}^S(L(m),Z(m))] = \mathcal{L}[\text{2-ATM}^S(L(m),Z(m))].$$

Proof. The proof is omitted here. Q.E.D.

5. Recognizability of Connected Pictures

It is unknown [2] whether a two-dimensional finite automaton can accept the set of connected pictures. (See [17] for the formal definition of a connected picture.) In Theorem 5.1, we first show that there exists a two-dimensional alternating finite automaton accepting connected pictures. It is shown [18] that m space is necessary and sufficient for TRTM^S's to accept the set of all the square connected pictures. In Theorem 5.2, we provide a result which may be viewed as a strengthening of that result. Let T_c be the set of all the square connected pictures.

Theorem 5.1. $T_c\in\mathcal{L}[\text{2-AFA}^S(m^2)]$.

Proof. The set T_c is accepted by a $\text{2-AFA}^S(m^2)$ M which acts as follows. Given an input x with $\ell_1(x)=\ell_2(x)=m\geq 1$, M scans the input x from top to bottom and from left to right on the same row. In the course of this scanning, each time M meets the symbol "1", it enters a universal state to choose one of two further actions: One action is to continue moving right or to the next row until M meets the next "1". The other action is to existentially check whether there exists a (connected) path of 1's from the current "1" to the lowermost and rightmost "1" on x, and to enter an accepting state if there exists such a path. It is obvious that the input x is in T_c if and only if there exists an accepting computation tree of M on x with at most m^2 leaves. Q.E.D.

Theorem 5.2. For any $L(m):N\to R$ and $Z(m):N\to R$ such that $\lim_{m\to\infty}[L(m)Z(m)/m]=0$ and $\lim_{m\to\infty}[Z(m)\log m/m]=0$,

$$T_c\notin\mathcal{L}[\text{TR2-ATM}^S(L(m),Z(m))].$$

Proof. The proof is omitted here. Q.E.D.

It is unknown whether or not T_c is accepted by a TR2-AFA^S.

REFERENCES

[1] A.Rosenfeld, Picture languages (Formal models for picture recognition), Academic Press, New York, 1979.

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

[3] K.Morita, H.Umeo, and K.Sugata, Computational complexity of L(m,n) tape-bounded two-dimensional tape Turing machines, IECE of Japan Trans.(D), Nov. 1977, p.982.

[4] S.Seki, Real-time recognition of two-dimensional tapes by cellular automata, Information Sci., 19, 3, 179-198 (1979).

[5] K.Inoue and A.Nakamura, Some properties of two-dimensional on-line tessellation acceptors, Information Sci., 13, 95-121 (1977).

[6] 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).

[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, Closure properties of three-way and four-way tape-bounded two-dimensional Turing machines, Information Sci., 18, 247-265 (1979).

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

[10] A.K.Chandra, D.C.Kozen, and L.J.Stockmeyer, Alternation, J.ACM., vol.28, No.1,114-133 (1981).

[11] R.E.Ladner, R.J.Lipton, and L.J.Stockmeyer, Alternating pushdown automata, Proc.19th IEEE Symp. on Foundations of Computer Science, Ann Arbor, Mich., 1978.

[12] W.L.Ruzzo, Tree-size bounded alternation, J. Comput.Syst.Sci., vol.21, 218-235 (1980).

[13] W.Paul and R.Reischuk, On alternation, Acta Informat., vol.14, 243-255 (1980).

[14] W.Paul and R.Reischuk, On alternation II, Acta Informat., vol.14, 391-403 (1980).

[15] K.N.King, Measures of parallelism in alternating computation trees, Proc.13th Ann.ACM Symp. on Theory of Computing (1981), 189-201.

[16] J.D.Hopcroft and J.D.Ullman, Formal languages and their relation to automata, Addison-Wesley, Reading, Mass., 1969.

[17] S.M.Selkow, One-pass complexity of digital picture properties, J.ACM., vol.19, No.2, 283-295 (1972).

[18] Y.Yamamoto, K.Morita, and K.Sugata, Space complexity for recognizing connectedness in three-dimensional pictures, IECE of Japan Trans.Section E (English), 1981.