

TWO TOPICS CONCERNING TWO-DIMENSIONAL AUTOMATA OPERATING IN PARALLEL

KATSUSHI INOUE

*Department of Computer Science and Systems Engineering
Faculty of Engineering, Yamaguchi University, Ube 755 Japan*

ITSUO SAKURAMOTO

*Department of Mechanical and Electrical Engineering
Tokuyama National College of Technology, Tokuyama 745 Japan*

MAKOTO SAKAMOTO

*Department of Information Engineering
Oshima National College of Maritime Technology
Oshima-Gun 742-21 Japan*

and

ITSUO TAKANAMI

*Department of Computer Science and Systems Engineering
Faculty of Engineering, Yamaguchi University, Ube 755 Japan*

This paper deals with two topics concerning two-dimensional automata operating in parallel. We first investigate a relationship between the accepting powers of two-dimensional alternating finite automata (2-AFAs) and nondeterministic bottom-up pyramid cellular acceptors (NUPCAs), and show that Ω (diameter \times log diameter) time is necessary for NUPCAs to simulate 2-AFAs. We then investigate space complexity of two-dimensional alternating Turing machines (2-ATMs) operating in small space, and show that if $L(n)$ is a two-dimensionally space-constructible function such that $\lim_{n \rightarrow \infty} L(n)/\log \log n > 1$ and $L(n) \leq \log n$, and $L'(n)$ is a function satisfying $L'(n) = o(L(n))$, then there exists a set accepted by some strongly $L(n)$ space-bounded two-dimensional deterministic Turing machine, but not accepted by any weakly $L'(n)$ space-bounded 2-ATM, and thus there exists a rich space hierarchy for weakly $S(n)$ space-bounded 2-ATMs with $\log \log n \leq S(n) \leq \log n$.

Keywords: Two-dimensional automata, alternating Turing machine, pyramid cellular acceptor, complexity.

1. INTRODUCTION

In Ref. 1, it was left as an open problem whether two-dimensional alternating finite automata (2-AFAs) can be simulated by two-dimensional nondeterministic on-line tessellation acceptors. Recently, in Ref. 2, this problem was solved negatively. On the other hand, it was shown in Ref. 3 that two-dimensional nondeterministic on-line tessellation acceptors can be simulated by nondeterministic bottom-up pyramid cellular acceptors (NUPCAs) which operate in $O(\text{diameter})$ time. It is natural to ask whether NUPCAs which operate in $O(\text{diameter})$ time can simulate 2-AFAs. In Sec. 3 of this paper, we consider this problem, and show that Ω (diameter \times log diameter) time is necessary for NUPCAs to simulate 2-AFAs.

Although conventional Turing machines (TMs) assume one-dimensional input tapes, TMs with two-dimensional input tapes (called 2-TMs) have also been studied extensively in the literature. (See, e.g. Refs. 2, 4–13.) For a detailed survey on 2-TMs, the reader is referred to Refs. 1 and 14. Deterministic, nondeterministic, and alternating versions of 2-TMs are defined in the same way as their one-dimensional counterparts and are denoted by 2-DTM, 2-NTM, and 2-ATM, respectively. In this paper, we will restrict our attention to 2-TMs with square input tapes.

For any function $L(n)$ and each $X \in \{D, N, A\}$, let *weak* (or *strong*) $XSPACE(L(n))$ denote the class of languages accepted by *weakly* (or *strongly*, respectively) $L(n)$ space-bounded XTMs, and *weak* (or *strong*) $2-XSPACE(L(n))$ denote the class of languages (i.e. sets of square patterns over some finite alphabet) accepted by *weakly* (or *strongly*, respectively) $L(n)$ space-bounded 2-XTMs. In the following, we omit the prefix “*weak*” (or “*strong*”) if a statement holds in either case.

Hopcroft and Ullman¹⁵ showed that if $L(n)$ is a space-constructible function such that $L(n) \leq \log n$ and $L'(n)$ is a function satisfying $L'(n) = o(L(n))$, then *strong* $DSPACE(L(n)) - \textit{strong} NSPACE(L'(n))$ is not empty. But it is obvious that the same proof still works if the prefix “*strong*” is dropped. Thus we have

Theorem 1.1 (Hopcroft and Ullman). If $L(n)$ is a space-constructible function such that $L(n) \leq \log n$ and $L'(n)$ is a function satisfying $L'(n) = o(L(n))$, then $DSPACE(L(n)) - NSPACE(L'(n))$ is nonempty.

By using a similar technique, it is easy to generalize Theorem 1.1 so that the statement will still hold if $NSPACE(L'(n))$ is replaced by $ASPACE(L(n))$. That is, it can be shown that if $L(n)$ is a space-constructible function such that $L(n) \leq \log n$ and $L'(n)$ is a function satisfying $L'(n) = o(L(n))$, then $DSPACE(L(n)) - ASPACE(L'(n))$ is nonempty.

The two-dimensional analogue of Theorem 1.1 was shown in Ref. 7.

Theorem 1.2 (Ito *et al.*). If $L(n)$ is a two-dimensionally space-constructible function such that $L(n) \leq \log n$ and $L'(n)$ is a function satisfying $L'(n) = o(L(n))$, then $2-DSPACE(L(n)) - 2-NSPACE(L'(n))$ is nonempty.

Remark. The original theorem in Ref. 7 actually states that if $L(n)$ is a two-dimensionally space-constructible function such that $L(n) \leq \log n$ and $L'(n)$ is a function satisfying $L'(n) = o(L(n))$, then *weak* $2-DSPACE(L(n)) - \textit{weak} 2-NSPACE(L'(n))$ is nonempty. As pointed out in Ref. 11, it is easy to show that the same statement still holds if the prefix “*weak*” is dropped.

Since functions such as $\log^{(1)}n, \log^{(2)}n, \log^{(3)}n, \dots$, are all two-dimensionally space-constructible,¹² it follows from Theorem 1.2 that $2-XSPACE(\log^{(i+1)}n) \not\subseteq 2-XSPACE(\log^{(i)}n)$ for each $X \in \{D, N\}$ and $i \geq 1$. (Here, $\log^{(1)}n = \log n$ and $\log^{(i+1)}n = \log(\log^{(i)}n)$ for each $i \geq 1$.)

It was an open problem in Ref. 1 whether a similar space hierarchy result holds for 2-ATMs. More precisely, it was asked whether *weak* $2-ASPACE(L(n)) - \textit{weak} 2-ASPACE(L'(n))$ is nonempty for each two-dimensionally space-constructible function $L(n) \leq \log n$ and function $L'(n)$ satisfying $L'(n) = o(L(n))$.

Recently, Jiang *et al.*¹¹ gave a partial solution to this problem, and showed that if $L(n)$ is a two-dimensionally space-constructible function such that $L(n) \leq \log n$ and $L'(n)$ is a function satisfying $L'(n) = o(L(n))$, then $strong\ 2-DSPACE(L(n)) - strong\ 2-ASPACE(L'(n))$ is nonempty. In Sec. 4 of this paper, we give another partial solution to this problem. That is, we show that if $L(n)$ is a two-dimensionally space-constructible function such that $L(n) \leq \log n$ and $\lim_{n \rightarrow \infty} L(n)/\log \log n > 1$, and $L'(n)$ is a function satisfying $L'(n) = o(L(n))$, then $strong\ 2-DSPACE(L(n)) - weak\ 2-ASPACE(L'(n))$ is nonempty.

Since functions $(\log \log n)^k$, $k \geq 1$, are two-dimensionally space-constructible (in fact one-dimensionally space-constructible), it follows from this result that $weak\ 2-ASPACE((\log \log n)^k) \not\subseteq weak\ 2-ASPACE((\log \log n)^{k+1})$ for each $k \geq 1$.

2. PRELIMINARIES

Let Σ be a finite set of symbols. A *two-dimensional tape* (or *pattern*) over Σ is a two-dimensional rectangular array of elements of Σ . The set of all patterns over Σ is denoted by $\Sigma^{(2)}$. Given a pattern $x \in \Sigma^{(2)}$, 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) . Further, 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 pattern 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 call $x[(i, j), (i', j')]$ the “ $[(i, j), (i', j')]$ -segment of x ”.)

We next review some basic concepts about a *nondeterministic bottom-up pyramid cellular acceptor* (NUPCA). An NUPCA is a pyramid stack of arrays of cells in which the bottom layer has size $2^n \times 2^n$ ($n \geq 0$), the next lowest $2^{n-1} \times 2^{n-1}$, and so forth, and the $(n + 1)$ st layer consists of a single cell, called the *root*. Each cell is defined as an identical finite state machine, $M = (Q_N, Q_T, \delta, A)$, where Q_N is a nonempty, finite set of states, $Q_T \subseteq Q_N$ is a finite set of input states, $A \subseteq Q_N$ is the set of accepting states, and $\delta: Q_N^5 \rightarrow 2^{Q_N}$ is the state transition function, mapping the current states of M and its four son cells in a 2×2 block on the layer below into M 's next state. As shown in Fig. 1, let c be some cell on the $(i + 1)$ st layer, and let $c(\text{NW})$, $c(\text{SW})$, $c(\text{SE})$ and $c(\text{NE})$ be four son cells (on the i th layer) of c , where $c(\text{NW})$ is c 's northwest son, etc. Then

$$q_c(t + 1) \in \delta(q_c(t), q_{c(\text{NW})}(t), q_{c(\text{SW})}(t), q_{c(\text{SE})}(t), q_{c(\text{NE})}(t)) ,$$

where for example $q_c(t)$ means the state of c at time t . At time $t = 0$, the input tape

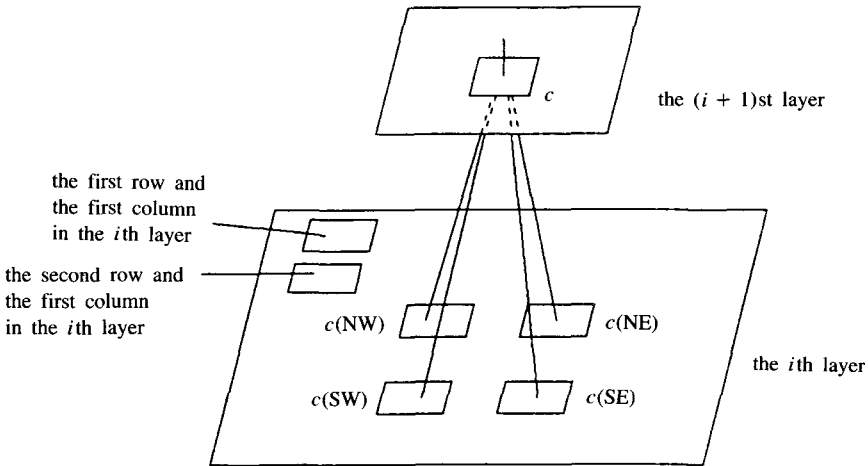


Fig. 1. Bottom-up pyramid cellular acceptor.

$x \in Q_T^{(2)}$ ($\ell_1(x) = \ell_2(x) = 2^n$, $n \geq 0$) is stored as the initial states of the bottom layer in such a way that $x(i, j)$ is stored at the cell of the i th row and the j th column, and the other cells are initialized to a *quiescent state* $q_s \in Q_N - Q_T - A$. As usual, we let $\delta(q_s, q_s, q_s, q_s, q_s) = \{q_s\}$. A computation of the NUPCA M on an input x is called *accepting* if the root cell ever enters an accepting state during the computation. An input x is *accepted* by M if there is an accepting computation of M on x .

Let $T: \mathbb{N} \rightarrow \mathbb{N}$ be a function, where \mathbb{N} denotes the set of all the non-negative integers. We say that an NUPCA M *operates in time* $T(n)$ if for every tape of size $2^n \times 2^n$ ($n \geq 0$) which is accepted by M , there is an accepting computation of M on x which uses no more than time $T(n)$. By $\text{NUPCA}(T(n))$, we denote an NUPCA which operates in time $T(n)$.

We next recall a *two-dimensional alternating Turing machine (2-ATM)*.^{1,10} A 2-ATM M has a read-only two-dimensional input tape with boundary symbols $\#$, and one semi-infinite storage tape, initially blank. A position is assigned to each cell of the read-only input tape, as shown in Fig. 2. A *step* of M consists of reading one symbol from each tape, writing a symbol on the storage tape, moving the input tape head in direction d ($d \in \{\text{left, right, up, down, no move}\}$), moving the storage tape head in the specified direction, and entering a new state, in accordance with the next move relation. The state set is partitioned into accepting, rejecting, existential and universal states.

A *configuration* of a 2-ATM M is of the form $(x, (i, j), (q, \alpha, k))$, where x is the two-dimensional input tape, (i, j) is the input head position ($0 \leq i \leq \ell_1(x) + 1$, $0 \leq j \leq \ell_2(x) + 1$), (q, α, k) is the storage state, where a *storage state* of M is a

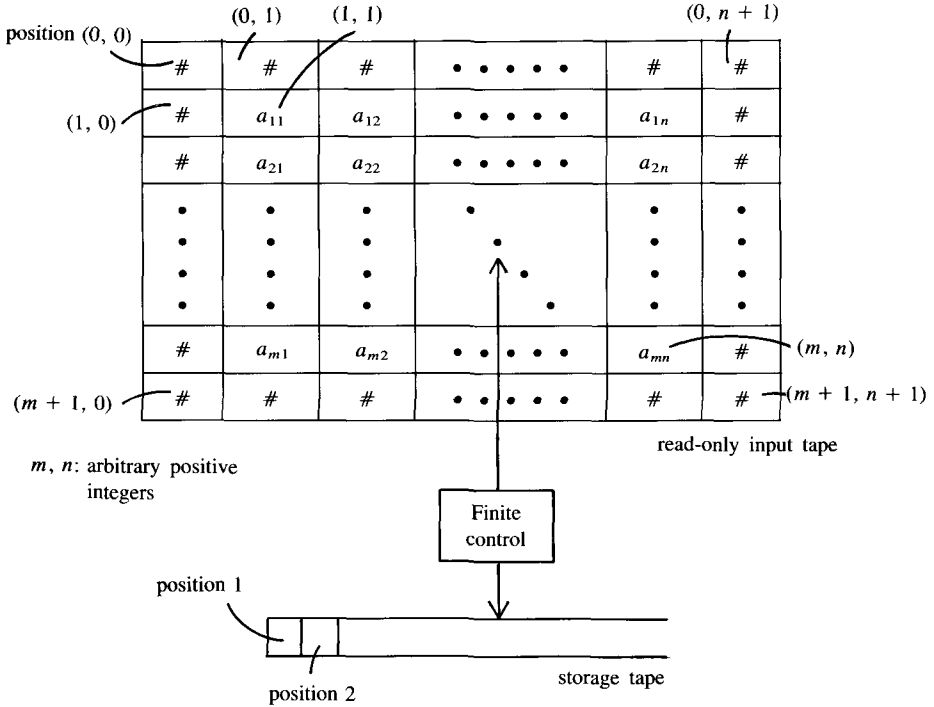


Fig. 2. Two-dimensional alternating Turing machine.

combination of the state of the finite control, the non-blank contents of the storage tape, and the storage head position. The *initial configuration* of M on input x is $I_M(x) = (x, (1, 1), (q_0, \varepsilon, 1))$, where q_0 is the initial state of M and ε denotes the empty string. A configuration is called *existential*, *universal*, *accepting*, and *rejecting*, respectively, if the corresponding state is existential, universal, accepting, and rejecting, respectively.

Given a 2-ATM M , we write $c \vdash_M c'$ and say c' is a *successor* of c if the configuration c' follows from the configuration c in one step of M according to the transition rules.

A *computation tree* of M is a finite, nonempty labeled tree with the following properties:

- (1) Each node v of the tree is labeled with a configuration $\ell(v)$.
- (2) If v is an internal node (a non-leaf) of the tree, $\ell(v)$ is universal and $\{c \mid \ell(v) \vdash_M c\} = \{c_1, \dots, c_k\}$, then v has exactly k children v_1, \dots, v_k such that $\ell(v_i) = c_i (1 \leq i \leq k)$.
- (3) If v is an internal node of the tree and $\ell(v)$ is existential, then v has exactly one child u such that $\ell(v) \vdash_M \ell(u)$.

A *computation tree of M on 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 configurations. We say that M *accepts x* if there is an accepting computation tree of M on x . Let $T(M) = \{x \mid M \text{ accepts } x\}$.

In this paper, we only consider 2-ATMs whose input tapes are restricted to square ones. Let $L: \mathbb{N} \rightarrow \mathbb{N}$ be a function and M be a 2-ATM. A computation tree of M (on some input tape) is $L(n)$ space-bounded if all nodes of the tree are labeled with configurations using at most $L(n)$ cells of the storage tape. We say that M is *weakly $L(n)$ space-bounded* if for each $n \geq 1$ and for each input tape x with $\ell_1(x) = \ell_2(x) = n$ that is accepted by M , there exists an $L(n)$ space-bounded accepting computation tree of M on x . We say that M is *strongly $L(n)$ space-bounded* if for each $n \geq 1$ and for each input tape x with $\ell_1(x) = \ell_2(x) = n$ (accepted by M or not), M never uses more than $L(n)$ cells of the storage tape.

A *two-dimensional nondeterministic Turing machine* (2-NTM) is a 2-ATM which has no universal states. A *two-dimensional deterministic Turing machine* (2-DTM) is a 2-ATM whose configurations each have at most one successor. A *two-dimensional alternating finite automaton* (2-AFA) is a 2-ATM which never uses the storage tape. For each $X \in \{D, N, A\}$, let $W2\text{-XTM}(L(n))$ (or $S2\text{-XTM}(L(n))$) denote a *weakly* (or *strongly*, respectively) $L(n)$ space-bounded 2-XTM, and let $W2\text{-XSPACE}(L(n))$ (or $S2\text{-XSPACE}(L(n))$) denote the class of sets (of square tapes) accepted by $W2\text{-XTM}(L(n))$ s (or $S2\text{-XTM}(L(n))$ s, respectively).

3. A RELATIONSHIP BETWEEN 2-AFAS and TIME-BOUNDED NUPCAS

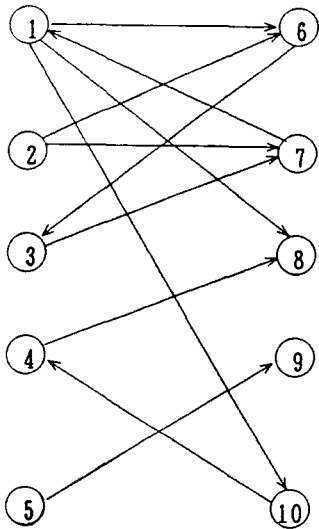
In this section, we investigate a relationship between the accepting powers of 2-AFAs and time-bounded NUPCAs, and show that Ω (diameter \times log diameter) time is necessary for NUPCAs to simulate 2-AFAs.

Theorem 3.1. Let $T(n)$ be any time function satisfying $T(n) = o(n \times 2^n)$. Then there exists a set accepted by some 2-AFA, but not accepted by any NUPCA($T(n)$).

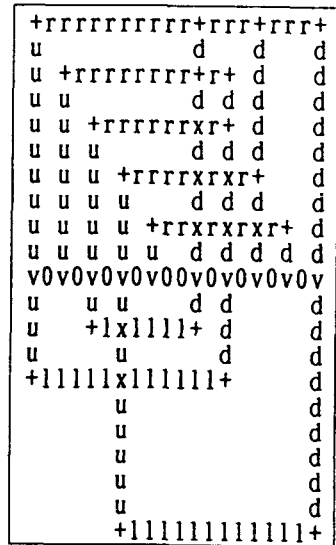
Proof. We consider the planar embedding of directed bipartite graphs with equal number of vertices on both sides, which Jiang *et al.*¹¹ introduced. Let $\Sigma = \{v, l, r, u, d, +, x, 0\}$ be the alphabet used for the embedding. We use the following embedding rule. The symbol v represents a vertex, symbol $+$ means an intersection of two edges (i.e. where they join or split), symbol x is for a cross-over of two edges, symbol 0 represents a blank space, and symbol u, d, l and r are the symbols needed to form upward, downward, leftward and rightward edges, respectively. Let pattern P be a planar embedding of a directed bipartite graph with m vertices on both sides. P will have $4m + 1$ rows and $4m$ columns. Row $2m + 1$ of P defines $2m$ vertices of a bipartite graph, where the left m vertices form one group and the right m vertices form the other. The $2m$ v 's are placed such that there are two blanks separating the first m vertices from the second m vertices, and there is a blank between consecutive v 's in both the left and right groups. The upper $2m$ rows of P specify the

set of (directed) edges from the left group of vertices to the right group of vertices, and the lower $2m$ rows specify the set of edges from the right group of vertices to the left group of vertices. An example of such an embedding is given in Fig. 3. It is easy to see that every directed bipartite graph with an equal number of vertices on both sides can be embedded in the plane following the above rule. Let $L_1 = \{x \in (\Sigma \cup \{2\})^{(2)} \mid \exists n \geq 1[\ell_1(x) = \ell_2(x) = 2^n \ \& \ \exists m \geq 1[2^n \geq 4m + 1 \ \& \ (x[(2^{n-1} - 2m, 1), (2^{n-1} + 2m, 4m)] \text{ is a planar embedding of some directed acyclic bipartite graph with } m \text{ vertices on both sides}) \ \& \ (\text{each symbol not in } x[(2^{n-1} - 2m, 1), (2^{n-1} + 2m, 4m)] \text{ is } 2)]]\}$. It is shown in Ref. 16 that a two-dimensional deterministic finite automaton can decide whether an input has 2^n rows and 2^n columns for some $n \geq 1$. Further, it is shown in Ref. 2 that a 2-AFA can recognize patterns that are planar embeddings of some directed acyclic bipartite graphs with an equal number of vertices on both sides. From these observations, we can easily see that L_1 can be accepted by a 2-AFA. Below, by using a technique similar to that in the proof of Lemma 3.1 in Ref. 2, we show that L_1 cannot be accepted by any NUPCA($T(n)$) with $T(n) = o(n \times 2^n)$.

Suppose that there exists an NUPCA($T(n)$) M accepting L_1 , where $T(n) = o(n \times 2^n)$. For each pattern P that is the embedding of a directed acyclic bipartite graph with m vertices on both sides, let P_U (P_L) be the upper (lower) half (i.e. $2m$ rows) of P . Note that the $(2m + 1)$ st row of P is for vertices. We fix an ordering on the edges from the right group of vertices to the left group of vertices as follows: for



(a) A directed bipartite graph G .



(b) The planar embedding of G .

Fig. 3. An example of embedding of G .

edges (a, b) and (c, d) in a graph, $(a, b) < (c, d)$ if a is to the left of c or $a = c$ and b is to the left of d in the embedding. We also fix an ordering on the P_L 's of the patterns of the same size as follows. Let $P_L^1 = \{e_1^1, e_2^1, \dots, e_s^1\}$ and $P_L^2 = \{e_1^2, e_2^2, \dots, e_t^2\}$, where $e_1^1 > e_2^1 > \dots > e_s^1$ and $e_1^2 > e_2^2 > \dots > e_t^2$. Then $P_L^1 < P_L^2$ if there exists some j such that $j < t$, $e_1^1 = e_1^2, \dots, e_j^1 = e_j^2$, and either $j = s$ or $j < s$ and $e_{j+1}^1 < e_{j+1}^2$. For each P_U , let $L(P_U) = \max\{P_L \mid \text{pattern } P \text{ formed by } P_U, (2m+1)\text{st row and } P_L \text{ is acyclic}\}$, and let $\text{MAXLP}(m) = \{L(P_U) \mid \text{for some } P_U \text{ with } 2m \text{ rows and } 4m \text{ columns}\}$. It is shown in Ref. 2 that the following proposition holds.

Proposition 3.1. $|\text{MAXLP}(m)| \geq m!$

Below we shall consider the case $m = 2^n$ for some large n . For each $n \geq 3$, let $V(n) = \{x \in (\Sigma \cup \{2\})^{(2)} \mid \ell_1(x) = \ell_2(x) = 2^{n+3} \ \& \ (x[(2^{n+2} - 2 \times 2^n, 1), (2^{n+2} + 2 \times 2^n, 4 \times 2^n)] \text{ is a planar embedding of a directed acyclic bipartite graph with } 2^n \text{ vertices on both sides) \ \& \ (each symbol not in } x[(2^{n+2} - 2 \times 2^n, 1), (2^{n+2} + 2 \times 2^n, 4 \times 2^n)] \text{ is } 2)\}$. Note that $V(n) \subseteq L_1$, and thus each tape in $V(n)$ is accepted by M . Below, we consider the case when the tapes with 2^{n+3} rows and 2^{n+3} columns are presented to M . Let r be the root cell of M , and let $\text{cell}(\text{NW})$ and $\text{cell}(\text{NE})$ be r 's northwest and northeast sons, respectively. For each pattern P that is the embedding of a directed acyclic bipartite graph with 2^n vertices on both sides (note that such a pattern P has $4 \times 2^n + 1$ rows and 4×2^n columns), let P' be the tape in $V(n)$ whose $[(2^{n+2} - 2 \times 2^n, 1), (2^{n+2} + 2 \times 2^n, 4 \times 2^n)]$ -segment is P . Note that P' is uniquely determined from P . We fix an accepting computation of M on P' , and denote it by $\text{comp}(P)$. Further, let $S_{\text{NW}}(P)$ ($S_{\text{NE}}(P)$) be the sequence of states which $\text{cell}(\text{NW})$ ($\text{cell}(\text{NE})$) enters in $\text{comp}(P)$ until time $T(n+3)$, and let $c(P) = (S_{\text{NW}}(P), S_{\text{NE}}(P))$. Clearly, there are $k^{2T(n+3)}$ possible $c(P)$'s where k is the number of states of each cell of M . From Proposition 3.1, it follows that $|\text{MAXLP}(2^n)| \geq (2^n)!$. From Stirling's approximation, we have $\log(2^n)! = \Omega(n2^n)$. Since $T(n) = o(n \times 2^n)$, it follows by a simple calculation that $|\text{MAXLP}(2^n)| > k^{2T(n+3)}$ for large n . Let n be sufficiently large such that $|\text{MAXLP}(2^n)| > k^{2T(n+3)}$. For each P_U , define $P(P_U)$ as the pattern formed by P_U , the $(2 \times 2^n + 1)$ st row and $L(P_U)$. Then, there exist P_U^1 and P_U^2 such that $L(P_U^1) < L(P_U^2)$ and $c(P(P_U^1)) = c(P(P_U^2))$. We denote $P(P_U^1)$ by P_1 , and $P(P_U^2)$ by P_2 . Both P_1 and P_2 are in L_1 , and so accepted by M . Now let P be the pattern formed by P_U^1 , the $(2 \times 2^n + 1)$ st row and $L(P_U^2)$. Then it follows that P' must also be accepted by M . Since M accepts L_1 , P is acyclic. But this contradicts the definition of $L(P_U^1)$. This completes the proof of the latter part of the theorem. \square

It is an open question whether 2-AFAs can be simulated by NUPCAs which operate in $O(\text{diameter} \times \log \text{diameter})$ time.

4. SPACE HIERARCHY OF 2-ATMs

We need the following concepts.

Definition 4.1. A function $L: \mathbb{N} \rightarrow \mathbb{N}$ is *two-dimensionally space constructible* if

there is a S2-DTM($L(n)$) M such that for each $n \geq 1$, there exists some input tape x with $\ell_1(x) = \ell_2(x) = n$ on which M halts after its storage tape head has marked off exactly $L(n)$ cells of the storage tape. (In this case, we say that M constructs the function L .)

Definition 4.2. Let Σ_1, Σ_2 be finite sets of symbols. A *projection* is a mapping $\bar{\tau}: \Sigma_1^{(2)} \rightarrow \Sigma_2^{(2)}$ which is obtained by extending a mapping $\tau: \Sigma_1 \rightarrow \Sigma_2$ as follows:

$$\begin{aligned} \bar{\tau}(x) = x' \Leftrightarrow & \text{(i) } \ell_k(x) = \ell_k(x') \text{ for each } k = 1, 2, \text{ and} \\ & \text{(ii) } \tau(x(i, j)) = x'(i, j) \text{ for each } (i, j) \in \\ & \quad \{(i, j) \mid 1 \leq i \leq \ell_1(x) \text{ and } 1 \leq j \leq \ell_2(x)\} . \end{aligned}$$

Theorem 4.1. Let $L: \mathbb{N} \rightarrow \mathbb{N}$ be a two-dimensionally space-constructible function such that $\lim_{n \rightarrow \infty} L(n)/\log \log n > 1$ and $L(n) \leq \log n$, and let $L': \mathbb{N} \rightarrow \mathbb{N}$ be any function satisfying $L'(n) = o(L(n))$. Then there exists a set in S2-DSPACE($L(n)$), but not in W2-ASPACE($L'(n)$).

Proof. Let M be a S2-DTM($L(n)$) which constructs the function L . Let $T[L, M]$ be the following set, which depends on L and M : $T[L, M] = \{x \in (\Sigma \times \{0, 1, 2\})^{(2)} \mid \exists n \geq 1 [\ell_1(x) = \ell_2(x) = 2n \ \& \ \exists r \leq L(2n) \text{ (when the tape } \bar{h}_1(x) \text{ is presented to } M, \text{ its storage tape head uses } r \text{ cells of the storage tape and then halts) \ \& \ } (\bar{h}_2(x[(1, 1), (2^r/2, 2n)]) = \bar{h}_2(x[(n+1, 1), (n+(2^r/2), 2n)]) \in \{0, 1\}^{(2)}) \ \& \ } (\bar{h}_2(x[(1+(2^r/2), 1), (n, 2n)]) = \bar{h}_2(x[(n+1+(2^r/2), 1), (2n, 2n)]) \in \{2\}^{(2)})]\}$, 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 $h_1: \Sigma \times \{0, 1, 2\} \rightarrow \Sigma$ ($h_2: \Sigma \times \{0, 1, 2\} \rightarrow \{0, 1, 2\}$) such that for any $c = (a, b) \in \Sigma \times \{0, 1, 2\}$, $h_1(c) = a$ ($h_2(c) = b$). (Note that for each tape x in $T[L, M]$, the top and bottom halves of $\bar{h}_2(x)$ are identical.) Below, we shall show that $T[L, M] \in \text{S2-DSPACE}(L(n))$ and $T[L, M] \notin \text{W2-ASPACE}(L'(n))$.

The set $T[L, M]$ is accepted by a S2-DTM($L(n)$) M_1 which acts as follows. Suppose that an input $x \in \{0, 1, 2\}^{(2)}$ with $\ell_1(x) = \ell_2(x) = 2n$, $n \geq 1$, is presented to M_1 . 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_1 has used at most $L(2n)$ cells of the storage tape because M is an S2-DTM($L(n)$), then M_1 checks by using the non-blank part of the storage tape that $\bar{h}_2(x)$ is a desired form. M_1 enters an accepting state only if this check is successful.

We next show that $T[L, M] \notin \text{W2-ASPACE}(L'(n))$. Suppose that there is a W2-ATM($L'(n)$) M_2 accepting $T[L, M]$. For each $n \geq 1$, let $w(n) \in \Sigma^{(2)}$ be a fixed tape such that (i) $\ell_1(w(n)) = \ell_2(w(n)) = 2n$ and (ii) when $w(n)$ is presented to M , M marks off exactly $L(2n)$ cells of the storage tape and halts. (Note that for each $n \geq 1$, there exists such a tape $w(n)$ because M constructs the function L .)

For each $n \geq 1$, let

$$V(n) = \{x \in T[L, M] \mid \ell_1(x) = \ell_2(x) = 2n \ \& \ \bar{h}_1(x) = w(n)\} .$$

For each x in $V(n)$, let $t(x)$ be a fixed accepting computation tree of M_2 on x such that all nodes of the tree are labeled with configurations using at most $L'(2n)$ cells of the storage tape. For each node v of $t(x)$, let $\ell(v)$ denote the label of v . We assume without loss of generality that for any $t(x)$,

- (i) any two different nodes on any path of $t(x)$ are labeled by different configurations, and
- (ii) if any two different nodes of $t(x)$ have the same label, then the subtrees with these nodes as the roots are identical.

For each x in $V(h)$, let $t'(x)$, which we call the *reduced accepting computation tree* of M_2 on x , be a tree obtained from $t(x)$ by the following procedure (for each node v of $t(x)$, we denote by $d(v)$ the length of the path from the root of $t(x)$ to v (i.e. the number of edges from the root of $t(x)$ to v)):

begin

1. $T_r = t(x)$
2. $i = 1$
3. Let $N(i) = \{v \mid v \text{ is a node of } T_r \text{ and } d(v) \leq i\}$. Divide $N(i)$ as follows: $N(i) = P(1) \cup P(2) \cup \dots \cup P(j_i)$, where ① if $i_1 \neq i_2$ ($1 \leq i_1, i_2 \leq j_i$), $P(i_1) \cap P(i_2) = \phi$, and ② for each i_1 ($1 \leq i_1 \leq j_i$) and for each $v_1, v_2 \in P(i_1)$, $\ell(v_1) = \ell(v_2)$ (i.e. the labels of v_1 and v_2 are identical). For each i_1 ($1 \leq i_1 \leq j_i$), let $\text{dis}(i_1) = \min\{d(v) \mid v \in P(i_1)\}$ and let $n(i_1)$ be the leftmost node among those nodes v 's in $P(i_1)$ such that $d(v) = \text{dis}(i_1)$. Further, let $N'(i) = N(i) - \{n(1), n(2), \dots, n(j_i)\}$. By removing from T_r all the subtrees whose roots are included in $N'(i)$, we make the new T_r .
4. If the height of T_r (i.e. the length of the longest path of T_r) is less than or equal to i , then we let $t'(x) = T_r$. Otherwise, we let $i = i + 1$ and go to step 3.

end

Example. Let $x \in V(n)$, and let $t(x)$ be a tree as shown in Fig. 4. Here, suppose that nodes A and D have the same label, nodes B and C have the same label, and other nodes have different labels from one another. (From the above assumption (ii) concerning $t(x)$, the subtrees with nodes A and D as the roots are identical, and the subtrees with nodes B and C as the roots are also identical.) Then, $t'(x)$ is a tree as shown in Fig. 5. That is, $t'(x)$ is obtained from $t(x)$ by removing the subtrees with nodes D and C as the roots from $t(x)$.

It is easily seen that for each x in $V(n)$, all the nodes of $t'(x)$ have different labels from one another, and the set of all the paths from the root of $t'(x)$ to the leaves of $t'(x)$ represents necessary and sufficient accepting computations of M_2 on x . From $t'(x)$, we now define an *extended crossing sequence* (ECS) at the boundary between the top and bottom halves of x . The concept of ECS was first introduced in Ref. 17. We relabel each node of $t'(x)$ as follows. (We denote this new labeling by ℓ' .) For each node v of $t'(x)$, let $f(v)$ denote the father node of v . Then, for each node v of $t'(x)$, where $x \in V(n)$, let $\ell'(v) = (j, (q, \alpha, k))$ if, for some storage states (q, α, k) and (q', α', k') ,

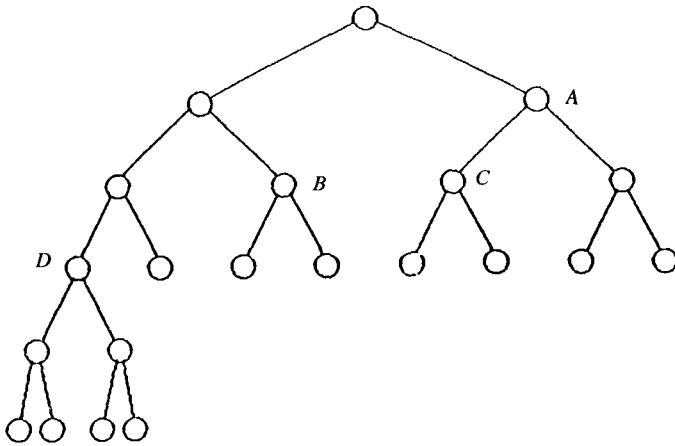


Fig. 4. $t(x)$.

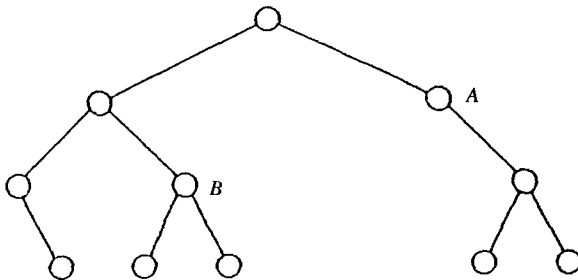


Fig. 5. $t'(x)$.

- (i) $\ell(f(v)) = (x, (n, j), (q', \alpha', k'))$ and $\ell(v) = (x, (n + 1, j), (q, \alpha, k))$, or
- (ii) $\ell(f(v)) = (x, (n + 1, j), (q', \alpha', k'))$ and $\ell(v) = (x, (n, j), (q, \alpha, k))$,

$\ell'(v) = *$ otherwise.

That is, if the action of M_2 when M_2 moves from $f(v)$ to v represents the action of crossing the boundary between the top and bottom halves of x along the j th column of x , then v is newly labeled by $(j, (q, \alpha, k))$, where (q, α, k) is the storage state component of $\ell(v)$. Otherwise, v is newly labeled by $*$. From the newly labeled $t'(x)$, we extract those nodes v 's such that $\ell'(v) \neq *$, and by using these nodes, we construct a tree $t''(x)$ satisfying the following condition (A):

(A): For any node v of $t''(x)$, nodes v_1, v_2, \dots, v_s are children of v if and only if v_1, v_2, \dots, v_s are descendants of v in $t'(x)$ and $\ell'(u) = *$ for each node u on the path from v to each v_i .

In general, there can be two or more such trees $t''(x)$. Let these trees be $t_1(x), \dots, t_m(x)$. (See Fig. 6.) For each node v of each $t_i(x) (1 \leq i \leq m)$, we now define an *element of ECS (EECS)* recursively as follows: let $\ell'(v) = (j, (q, \alpha, k))$;

- (1) If v is a leaf, then $[(j, (q, \alpha, k))]$ is an EECS of v .
- (2) If v has only one child v_1 and $Q_1 = [(j_1, (q_1, \alpha_1, k_1))P]$ is an EECS of v_1 , then $[(j, (q, \alpha, k))(j_1, (q_1, \alpha_1, k_1))P]$ is an EECS of v .

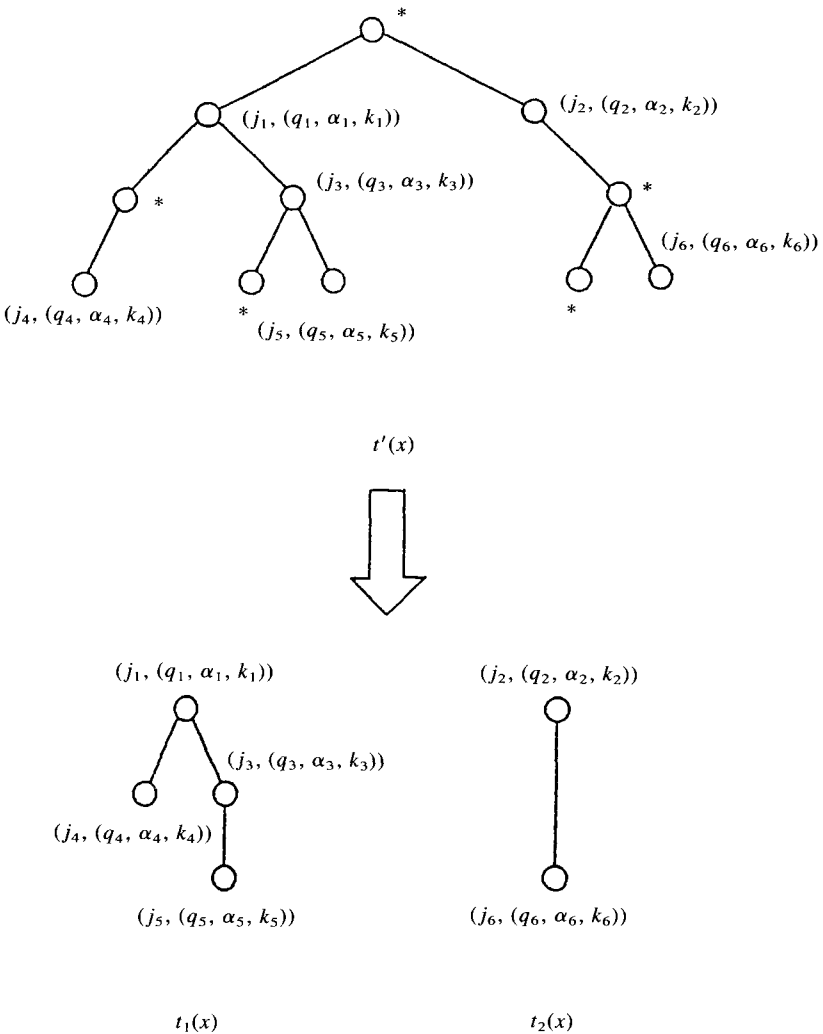


Fig. 6. Extraction of $t_1(x), \dots, t_m(x)$ from $t'(x)$.

- (3) If v has d children $v_1, v_2, \dots, v_d (d \geq 2)$ and Q_1, \dots, Q_d are EECSs of v_1, \dots, v_d , respectively, then $[(j, (q, \alpha, k))Q_{\sigma(1)} \cdots Q_{\sigma(d)}]$ is an EECS of v for any permutation $\sigma: \{1, \dots, d\} \rightarrow \{1, \dots, d\}$.
- (4) An EECS of v is defined only by (1), (2), or (3) above.

Let Q_1, \dots, Q_m be EECSs of the root nodes of $t_1(x), \dots, t_m(x)$, respectively. Then, for any permutation $\sigma: \{1, \dots, m\} \rightarrow \{1, \dots, m\}$, we call $Q_{\sigma(1)} \cdots Q_{\sigma(m)}$ an ECS of $t'(x)$ (or an ECS of x). As is easily seen from the definitions, there can be two or more EECSs of each node v of each $t_i(x)$, and there can be two or more ECSs of $t'(x)$.

Let Q_1 and Q_2 be any two EECSs. We say that Q_1 and Q_2 are *equivalent*, which we denote by $Q_1 \equiv Q_2$, if the following condition (B) is satisfied:

(B): Let $Q_1 = [(j_1, (q_1, \alpha_1, k_1)) \cdots (j_m, (q_m, \alpha_m, k_m))P_1 \cdots P_s]$ and $Q_2 = [(j'_1, (q'_1, \alpha'_1, k'_1)) \cdots (j'_{m'}, (q'_{m'}, \alpha'_{m'}, k'_{m'}))P'_1 \cdots P'_{s'}]$. Then, $m = m', s = s', (j_i, (q_i, \alpha_i, k_i)) = (j'_i, (q'_i, \alpha'_i, k'_i))$ for each $i(1 \leq i \leq m)$, and there exists a permutation $\sigma: \{1, \dots, s\} \rightarrow \{1, \dots, s\}$ such that $P_i \equiv P'_{\sigma(i)}$ for each $i(1 \leq i \leq s)$, where $m, s \geq 0$, and $(j, (q, \alpha, k))$'s, $(j', (q', \alpha', k'))$'s are pairs (column number, storage state)'s, and further P, P' are EECSs.

Let $Q = Q_1 \cdots Q_m$ and $Q' = Q'_1 \cdots Q'_{m'}$ be any two ECSs. We say that Q and Q' are *equivalent* if $m = m'$ and there exists a permutation $\sigma: \{1, \dots, m\} \rightarrow \{1, \dots, m\}$ such that $Q_i \equiv Q'_{\sigma(i)}$ for each $i(1 \leq i \leq m)$. (As is easily seen from the definition, any two ECSs of $t'(x)$ are equivalent for any x in $V(n)$.) For any ECS Q , let $|Q|$ denote the number of pairs (column number, storage state)'s in Q . For each $n \geq 1$, let

$$E(n) = \{Q \mid Q \text{ is an ECS of } t'(x) \text{ for some } x \text{ in } V(n)\} .$$

Then the following two propositions must hold:

Proposition 4.1. $|E(n)| \leq Z(n)^{cZ(n)}$, where $Z(n) = r(2n + 2)L'(2n)s^{L'(2n)}$, r and s are the numbers of states (of the finite control) and storage tape symbols of M_2 , respectively, and c is a positive constant.

Proof. The number of different (column number, storage state)'s when the input head of M_2 crosses the boundary between the top and bottom halves of tapes in $V(n)$ is $Z(n) = r(2n + 2)L'(2n)s^{L'(2n)}$. From the definitions, it is easily seen that for each Q in $E(n)$, the (column, storage state)'s in Q are different from one another. Thus we have $|Q| \leq Z(n)$. From this, we can show by using the same idea as in the proof of Lemma 5 of Ref. 17 that the proposition holds. \square

Proposition 4.2. Let x and y be any two different tapes in $V(n)$, and let Q_x and Q_y be any ECSs of x and y , respectively. Then Q_x and Q_y are not equivalent.

Proof. Suppose that Q_x and Q_y are equivalent. We consider the tape z satisfying the following: (i) $\ell_1(z) = \ell_2(z) = 2n$, (ii) the top half of z is identical with the top half of x , and (iii) the bottom half of z is identical with the bottom half of y . By using the well-known technique, we can easily construct an accepting computation tree of M_2 on

z (such that all nodes of the tree are labeled with configurations using at most $L'(2n)$ cells of the storage tape) from $t(x)$ and $t(y)$. Thus z is also accepted by M_2 . This is a contradiction, because $z \notin T[L, M]$. \square

Clearly, $|V(n)| = 2^{2n(2^{\lfloor 2n \rfloor / 2})}$. From the assumptions concerning functions L and L' and from Proposition 4.1, it follows that $|V(n)| > |E(n)|$ for large n . For such a large n , there must be two different tapes x, y in $V(n)$ such that some ECS of x is equivalent to some ECS of y . This contradicts Proposition 4.2. This completes the proof of “ $T[L, M] \notin W2\text{-SPACE}(L'(n))$ ”. \square

Since functions $(\log \log n)^k$, $k \geq 1$, are two-dimensionally space-constructible (in fact one-dimensionally space-constructible), it follows from Theorem 4.1 that the following corollary holds.

Corollary 4.1. For any positive integer k ,

$$W2\text{-SPACE}((\log \log n)^k) \not\subseteq W2\text{-SPACE}((\log \log n)^{k+1}).$$

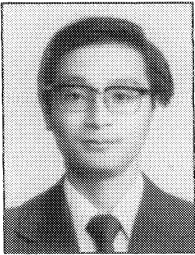
It is still unknown whether there exists an infinite space hierarchy for $W2\text{-ATM}(L(n))$'s with $L(n) \leq \log \log n$.

REFERENCES

1. K. Inoue and I. Takanami, “A survey of two-dimensional automata theory”, *Inf. Sci.* **55** (1991) 99–121.
2. T. Jiang, O. Ibarra and H. Wang, “Some results concerning 2-D on-line tessellation acceptors and 2-D alternating finite automata”, (1990), submitted to *J. Comput. Syst. Sci.*; also McMaster Univ. Dept. of Computer Science and Systems Technical Report No. 90–03.
3. K. Inoue, I. Takanami and A. Nakamura, “A note on time-bounded bottom-up pyramid cellular acceptors”, *Inf. Sci.* **51** (1990) 121–133.
4. M. Blum and C. Hewitt, “Automata on a 2-dimensional tape”, *Proc. IEEE Symp. on Switching and Automata Theory*, 1967, pp. 155–160.
5. J. Hromkovič, K. Inoue and I. Takanami, “Lower bounds for language recognition on two-dimensional alternating multihead machines”, *J. Comput. Syst. Sci.* **38** (1989) 431–451.
6. A. Ito, K. Inoue, I. Takanami and H. Taniguchi, “Two-dimensional alternating Turing machines with only universal states”, *Inf. Control* **55** (1982) 193–221.
7. A. Ito, K. Inoue, I. Takanami and H. Taniguchi, “A note on space complexity of nondeterministic two-dimensional Turing machines”, *Trans. IECE Japan* **E66**, 8 (1983) 508–509.
8. A. Ito, K. Inoue and I. Takanami, “A note on three-way two-dimensional alternating Turing machines”, *Inf. Sci.* **45** (1988) 1–22.
9. O. Ibarra and R. Melson, “Some results concerning automata on two-dimensional tapes”, *Int. J. Comput. Math.* **4**, A(1974) 269–279.
10. K. Inoue, I. Takanami and H. Taniguchi, “Two-dimensional alternating Turing machines”, *Theoret. Comput. Sci.* **27** (1983) 61–83.
11. T. Jiang, O. Ibarra, H. Wang and Q. Zheng, “A hierarchy result for 2-dimensional TM's operating in small space”, to appear in *Inf. Sci.*

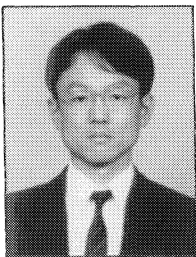
12. K. Morita, H. Umeo, H. Ebi and K. Sugata, "Lower bounds on tape complexity of two-dimensional tape Turing machines", *Trans. IECE Japan* **J61-D**, 6 (1978) 381-386.
13. A. Szepletowski, "On three-way two-dimensional Turing machines", *Inf. Sci.* **47** (1989) 135-147.
14. P. S. P. Wang (Ed.), *Array Grammars, Patterns and Recognizers*, World Scientific, 1989.
15. J. E. Hopcroft and J. D. Ullman, "Some results on tape-bounded Turing machines", *J. ACM* **16**, 1 (1967) 168-177.
16. H. Umeo, "Formal Studies on Pattern Recognition by Two-Dimensional Tape Automata", Ph. D. Thesis, Osaka University, 1978.
17. H. Yamamoto and S. Noguchi, "Time- and leaf-bounded 1-tape alternating Turing machines", *Trans. IECE Japan* **J68-D**, 10 (1985) 1719-1726.

Received 15 June 1991.

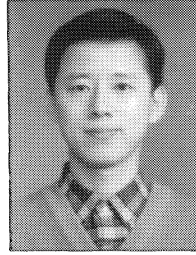


Katsushi Inoue received the B.E. and M.E. degrees in electrical engineering from Hiroshima University, in 1969 and 1971 respectively, and the Ph.D. degree in electrical engineering from Nagoya University in 1977. From

1971 to 1973, he joined Musashino Electrical Communication Laboratory, NTT, Musashino. From 1973 to 1977, he was an Assistant Professor of Hiroshima University and from 1977 to 1987, he was an Associate Professor of Yamaguchi University. He is presently a Professor in the Faculty of Engineering, Yamaguchi University. His main interests are automata theory, computer geometry and parallel processing.

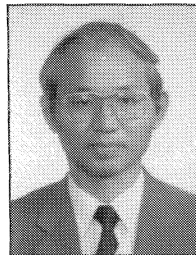


Itsuo Sakuramoto received the B.S. and M.S. degrees in electronics engineering from Yamaguchi University. Since 1990, he has been a lecturer of Tokuyama College of Technology. His current interests are digital image processing and automata theory.



Makoto Sakamoto received the B.M.M.S. and the M.M.M.S. degrees in nautical science from Tokyo University of Mercantile Marine, in 1984 and 1987, respectively. Since April 1987, he has been with Oshima National College of Maritime Tech-

nology. He is presently an Assistant Professor at the Department of Information Engineering. His first interests lay in hydrodynamics and time series analysis, especially the directional wave spectrum. His current main research interests are automata theory and digital image processing.



Itsuo Takanami received, the M.E. and D.E. degrees in electrical and communication engineering from Tohoku University, Sendai, in 1967 and 1970 respectively. In March 1970 he joined Tohoku University, where from 1971 to 1972 and from

1972 to 1974 he was an Assistant Professor and an Associate Professor respectively. He is presently a Professor in the Faculty of Engineering, Yamaguchi University. His current interests are automata theory, parallel processing and multi-valued logic circuits.