PAPER

Alternating Multihead Finite Automata with Constant Leaf-Sizes

Hiroshi MATSUNO†, Katsushi INOUE††, Itsuo TAKANAMI††
and Hiroshi TANIGUCHI†††, *Members*

SUMMARY We introduce alternating multihead finite automata with constant leaf-sizes (AMHFACLs). and investigate several properties of these automata. The main results of this paper are as follows: (1) two-way sensing AMHFACLs can be simulated by two-way nondeterministic simple multihead finite automata, (2) for one-way AMHFACLs, $k+1$ heads are better than k, and (3) for one-way alternating simple multihead finite automata with constant leaf-sizes, sensing versions are more powerful than non-sensing versions.

1. Introduction

Many investigations about multihead finite automata (MHFAs) have been made[(1)-(7)]. Recently, King[(6)] introduced an alternating multihead finite automaton (AMHFA) which is the same as an AMHFA except that its state set is divided into two disjoint sets, a set of universal states and a set of existential states. On the other hand, Matsuno et al.[(7)] introduced an alternating simple multihead finite automaton (ASPMHFA) which can be considered as a restricted version of AMHFA, and provided a relationship between the accepting powers of AMHFAs and ASPMHFAs. In this paper, we introduce AMHFAs with constant leaf-sizes (AMHFACLs), and give several properties of these automata. Leaf-size, in a sense, reflects the minimal number of processors which run in parallel in accepting a given input. AMHFAs with constant leaf-sizes are more realistic parallel computation models than ordinary AMHFAs because of the restriction of the number of processors which run in parallel to constant.

Section 2 gives terminologies and notations necessary for this paper. Section 3 investigates a relationship between the accepting powers of AMHFAs and ASPMHFAs. The main result of this section is that two-way sensing AMHFACLs can be simulated by two-way nondeterministic simple multihead finite automata. In Sect. 4, we show the following two results. For ony-way AMHFACLs, $k+1$ heads are better than k. For one-way SPMHFAs, sensing versions are more powerful than non-sensing versions.

Manuscript received May 2, 1988.
† The author is with Oshima National College of Maritime Technology, Yamaguchi-ken, 742-21 Japan.
†† The authors are with the Faculty of Engineering, Yamaguchi University, Ube-shi, 755 Japan.
††† The author is with the Faculty of Engineering, Tottori University, Tottori-shi, 680 Japan.

2. Preliminaries

The reader is referred to[(1)-(3)] for formal definitions of a multihead finite automaton (MHFA). A simple multihead finite automaton (SPMHFA) is an MHFA with the restriction that one head (called the 'reading head') can sense input symbols, while the others (called the 'counting heads') can only detect the left endmarker "¢" and right endmarker "\$". When the heads of MHFA (SPMHFA) are allowed to sense the presence of other heads on the same input position, we call such MHFA (SPMHFA) a 'sensing' MHFA (SPMHFA).

A two-way MHFA and a one-way MHFA are defined as usual. A two-way SPMHFA is an SPMHFA whose reading and counting head can move in two directions. A semi-one-way SPMHFA is an SPMHFA whose reading head can move only in one direction, but whose counting heads can move in two directions. A one-way SPMHFA is an SPMHFA whose reading and counting heads can move in one direction.

When an input string x is presented to an MHFA (SPMHFA) M, M starts in its initial state with each head on the left endmarker "¢". M accepts x if and only if it enters an accepting state during the course of computation.

An alternating MHFA (AMHFA)[(6)] and an alternating SPMHFA[(7)] are alternating versions of an MHFA and an SPMHFA, respectively. That is, an AMHFA (ASPMHFA) is the same as an MHFA (SPMHFA) except that the state set is divided into two disjoint sets, the set of universal states and the set of existential states. Of course, each alternating automaton has a specified set of accepting states, which is a subset of the state set.

A step of an AMHFA (ASPMHFA) M consists of reading a symbol from the input string by each head, moving the heads in specified directions (note that any of the heads can remain stationary during a move), and entering a new state, in accordance with the transition function. If one of the heads falls off the input string, then M can make no further move.

[Definition 2. 1] A configuration of a (sensing) alternating k-head finite automaton M is an element of

$$\Sigma^* \times C_M$$

where $\Sigma(¢, \$ \in \Sigma)$ is the input alphabet of M, and $C_M = Q \times (N \cup \{0\})^k$ (where Q is the set of states of the finite control of M and N denotes the set of all positive integers). The first component x of a configuration $c = (x, (q, (i_1, i_2, \cdots, i_k)))$ represents the input string. The second component $(q, (i_1, i_2, \cdots, i_k))$ $(\in C_M)$ of c represents the state of finite control and the positions of k input heads. 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 universal (existential, accepting) configuration if q is a universal (existential, accepting) state. The initial configuration of M on input x is

$$I_M(x) = (x, (q_0, \underbrace{(0, 0, \cdots, 0)}_{k}))$$

where q_0 is the initial state of the finite control of M.

[Definition 2.2] Given a (sensing) alternating multihead automaton M, we write $c \vdash c'$ and say that c' is a successor of c if configuration c' follows from configuration c in one step, according to the transition function of M. A computation path of M on input x is a sequence $c_0 \vdash c_1 \vdash \cdots \vdash c_n$ $(n \geq 0)$, where $c_0 = I_M(x)$. A computation tree of M is a finite, nonempty labeled tree with the following properties:

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

(2) If π is an internal node of the tree, $l(\pi)$ is universal and $\{c \mid l(\pi) \vdash c\} = \{c_1, c_2, \cdots, c_r\}$, then π has exactly r children $\rho_1, \rho_2, \cdots, \rho_r$ such that $l(\rho_i) = c_i$,

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

A computation tree of M on 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.

For any MHFA (SPMHFA) M, let $T(M)$ be the set of strings accepted by M.

Deterministic and nondeterministic MHFAs (SPMHFAs) are special cases of alternating versions. That is, a nondeterministic MHFA (SPMHFA) is an AMHFA (ASPMHFA) which has no universal states, and a deterministic MHFA (SPMHFA) is an AMHFA (ASPMHFA) whose configurations have at most one successor.

In this paper, to represent the different kinds of SPMHFAs (resp. MHFAs, sensing MHFAs) systematically, we use the notation $XYk\text{-}HZ$ (resp. $Xk\text{-}HZ$, $XSNk\text{-}HZ$), $k > 1$, where

(1) $X \in \{D, N, A, U\}$,
- D : deterministic,
- N : nondeterministic,
- A : alternating,
- U : alternating automaton with only universal states;

(2) $Y \in \{SP, SNSP\}$
- SP : simple,
- $SNSP$: sensing simple;

(3) $k\text{-}H$: k-head (the number of heads is k);

(4) $Z \in \{FA, SFA, TWFA\}$
- FA : one-way,
- SFA : semi-one-way,
- $TWFA$: two-way,
- (Of course, 'SFA' is used only for SPMHFA.).

For example,
- $DSPk\text{-}HFA$: deterministic simple k-head one-way finite automaton
- $USNk\text{-}HTWFA$: alternating sensing k-head two-way finite automaton with only universal states.

Furthermore, for each $X \in \{D, N, A, U\}$, $Y \in \{SP, SNSP\}$, $k \geq 1$, $Z \in \{TWFA, SFA, FA\}$, and $Z' \in \{TWFA, FA\}$,

$\mathcal{L}[XYk\text{-}HZ] = \{T \mid T = T(M)$ for some $XYk\text{-}HZ$ $M\}$,

$\mathcal{L}[XSNk\text{-}HZ'] = \{T \mid T = T(M)$ for some $XSNk\text{-}HZ'$ $M\}$, and

$\mathcal{L}[Xk\text{-}HZ'] = \{T \mid T = T(M)$ for some $Xk\text{-}HZ'$ $M\}$.

In this paper we shall introduce a simple, natural complexity measure for AMHFAs and ASPMHFAs, called leaf-size[7]. Basically, the 'leaf-size' used by AMHFA (ASPMHFA) on a given input is the number of leaves of an accepting computation tree with the fewest leaves.

[Definition 2.3] Let $L : N \rightarrow R$ be a function, where R denotes the set of all nonnegative real numbers. For each tree t, let LEAF (t) denote the leaf-size of t (i. e., the number of leaves of t). We say that an $XYk\text{-}HZ$ (resp. $Xk\text{-}HZ$, $XSNk\text{-}HZ$) $(X \in \{A, U\}$, $Y \in \{SP, SNSP\}$, $k \geq 1$, $Z \in \{TWFA, SFA, FA\})$ M is $L(n)$ leaf-size bounded if, for each n and for each input x of length n, if x is accepted by M, then there is an accepting computation tree of M on x such that LEAF $(t) \leq \lceil L(n) \rceil$†.

For each $X \in \{A, U\}$, $Y \in \{SP, SNSP\}$, $k \geq 1$, and $Z \in \{TWFA, SFA, FA\}$, we let $XYk\text{-}HZ(L(n))$ (resp. $Xk\text{-}HZ(L(n))$, $XSNk\text{-}Z(L(n))$) denote $L(n)$ leaf-size bounded $XYk\text{-}HZ$ (resp. $Xk\text{-}HZ$, $XSNk\text{-}HZ$). Define,

$\mathcal{L}[XYk\text{-}HZ(L(n))] = \{T \mid T = T(M)$ for some $XYk\text{-}HZ(L(n))M\}$,

$\mathcal{L}[Xk\text{-}HZ(L(n))] = \{T \mid T = T(M)$ for some $Xk\text{-}HZ(L(n))M\}$, and

$\mathcal{L}[XSNk\text{-}HZ(L(n))] = \{T \mid T = T(M)$ for some

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

$XSNk\text{-}HZ(L(n))M\}$.

3. Simple Versus Non-simple Alternating Automata

We first show that two-way AMHFACLs can be simulated by two-way nondeterministic SPMHFAs.

[Theorem 3.1] For each $k\geq 1$ and each $s\geq 1$,

(1) $\mathcal{L}[ASNk\text{-}HTWFA(s)]\subseteq\mathcal{L}[NSP(ks+1)-HTWFA]$, and

(2) $\mathcal{L}[USNk\text{-}HTWFA(s)]\subseteq\mathcal{L}[DSP(ks+1)-HTWFA]$.

(Proof) (1) Let M_1 be an $ASNk-HTWFA(s)$. We construct an $NSP(ks+1)-HTWFA$ M_2 accepting $T(M_1)$. Let R be the reading head of M_2, and $H_{12}, H_{13},\cdots, H_{1k}, H_{21}, H_{22},\cdots, H_{2k},\cdots, H_{s1}, H_{s2},\cdots, H_{sk}, C$ be the counting heads of M_2. Suppose that an input string x is presented to M_2. M_2 starts to simulate the action of M_1 on x by using $R, H_{12},\cdots, H_{1k}$. Other counting heads are left on the left endmarker "¢".

During the simulation, if M_2 need to check whether any two heads of M_2 are on the same position, M_2 continues to move, for every one right move of C, these two heads simultaneously one cell to the left until at least one of these two heads reach "¢" (Note that initially, C is on "¢"). Moving each head in this way, M_2 can detect whether these two heads were on the same input position by checking whether these two heads reach "¢" at the same time. After this check, M_2 replaces these two heads to the previous positions by using C.

On the other hand, by using C, M_2 can read the symbol under H_{1j} $(2\leq j\leq k)$ as follows. First, M_2 moves C one cell to the right, for every one left move of R until it reaches "¢". Secondly, M_2 moves R one cell to the right, for every one left move of H_{1j} until it reaches "¢". Then, M_2 can read the symbol where H_{1j} was positioned. After this, by using C, M_2 replaces R and H_{1j} to the previous positions. That is, M_2 moves H_{1j} one cell to the right, for every one left move of R until it reaches "¢", and thereafter M_2 moves R one cell to the right, for every one left move of C until it reaches "¢".

Suppose that, during the simulation, M_1 enters a universal state. Let $q_1, q_2,\cdots, q_t (t\leq s)$ be the states of successors of the configuration with that universal state. When M_2 notices that M_1 enters the universal state, M_2 stores $q_2,\cdots, q_t$ in its finite control and makes $t-1$ groups of counting heads coincide with $R, H_{12},\cdots, H_{1k}$ (One group of counting heads consists of k counting heads). That is, M_2 makes each H_{i1} $(1\leq i\leq t-1)$ coincide with R and for each j $(2\leq j\leq k)$, M_2 makes each H_{ij} $(1\leq i\leq t-1)$ coincide with H_{1j}. M_2 continues to simulate the action of M_1 with the state q_1 by using $R, H_{12},\cdots, H_{1k}$. During the simulation, if M_2 notices that M_1 enters a universal state again, M_2 do the same actions mentioned above. Repeating the action above, if M_2 notices that M_1 enters a non-accepting state and halts, then M_2 enters a non-accepting state and halts. Conversely, suppose that M_2 notices that M_1 enters an accepting state and halts, and suppose that r $(1\leq r\leq s)$ states of M_1 are stored in the finite control of M_2. Then, M_2 arbitrarily selects one state (say q_u $(2\leq u\leq s)$) from these r states, makes $R, H_{12},\cdots, H_{1k}$ coincide with $H_{u1}, H_{u2},\cdots, H_{uk}$, respectively, and continues to simulate the action of M_1 with the state q_u by using heads $R, H_{12},\cdots, H_{1k}$.

Repeating the actions above, M_2 enters an accepting state if and only if the state (of M_1) stored in the finite control of M_2 are all successefully consumed. It will be obvious that M_2 accepts $T(M_1)$.

(2) The proof is similar to the proof of (1) except that during the simulation, M_1 and M_2 do not make a nondeterministic action. (Q. E. D.)

As a corollary of Theorem 3.1, we can get

[Corollary 3.1]

(1) $\bigcup_{1\leq k<\infty}\bigcup_{1\leq s<\infty}\mathcal{L}[ASNk-HTWFA(s)] = \bigcup_{1\leq k<\infty}\mathcal{L}[NSPk-HTWFA]$, and

(2) $\bigcup_{1\leq k<\infty}\bigcup_{1\leq s<\infty}\mathcal{L}[USNk-HTWFA(s)] = \bigcup_{1\leq k<\infty}\mathcal{L}[DSPk-HTWFA]$.

It is known that $\mathcal{L}[ASPk-HFA]$ is equal to $\mathcal{L}[Ak-HFA]$ for each $k\geq 1$[7]. We next show that, when leaf-sizes are bounded, different situation occurs for one-way AMHFAs.

[Lemma 3.1] For each $r\geq 1$

$$B(r)=\{a^u0^{m_1}10^{m_2}1\cdots10^{m_r}20^{m_1}10^{m_2}1\cdots10^{m_r}b^t\}$$
$$\forall i(1\leq i\leq r)[m_i\geq 1]\ \&\ (a, b\in\{0, 1, 2\})$$
$$\&\ (u, t\geq 0)\}.$$

Then,

(1) for each $r\geq 1$, $B(r)\in\mathcal{L}[D2-HFA]$, and

(2) for each $k\geq 1$ and each $s\geq 1$, $B(ks)\notin\mathcal{L}[ASNSPk-HFA(s)]$.

(Proof) The proof is omitted, since part (1) is easy to prove and part (2) is shown in Lemma 5.10 in Ref. (7). (Q. E. D.)

From Lemma 3.1, we can get the following theorem.

[Theorem 3.2] For each $X\in\{A, U\}$, $k\geq 1$, and $s\geq 1$,

(1) $\mathcal{L}[XSPk-HFA(s)]\subsetneqq\mathcal{L}[Xk-HFA(s)]$, and

(2) $\mathcal{L}[XSNSPk-HFA(s)]\subsetneqq\mathcal{L}[XSNk-HFA(s)]$.

[Remark 3.1] Let $T=\{w2w|w\in\{0, 1\}^+\}$. It is shown in Remark 5.8 in Ref. (7) that $T\notin\bigcup_{1\leq r<\infty}\mathcal{L}[ASNSPr-HSFA(L(n))]$ for any function L such that $\lim_{n\to\infty}[L(n)\log_2 n/n]=0$. From this fact and the fact that $T\in\mathcal{L}[D2-HFA]$, we can get for each $X\in\{A, U\}$, $k\geq 1$, and for any function L such that $\lim_{n\to\infty}[L(n)\log_2 n/n]=0$,

(1) $\mathcal{L}[XSPk-HFA(L(n))]\subsetneqq\mathcal{L}[Xk-HFA(L(n))]$,

(2) $\bigcup_{1\leq r<\infty}\mathcal{L}[XSPr-HFA(L(n))]\subsetneqq\bigcup_{1\leq r<\infty}\mathcal{L}[Xr-HFA(L(n))]$,

(3) $\mathcal{L}[XSNSPk-HFA(L(n))]\subsetneqq\mathcal{L}[XSNk-HFA(L(n))]$, and

(4) $\bigcup_{1\le r<\infty}\mathcal{L}[XSNSPr-HFA(L(n))]\subsetneq\bigcup_{1\le r<\infty}\mathcal{L}[XSNr-HFA(L(n))]$.

4. Some Properties of One-Way AMHFAs with Constant Leaf-Sizes

In Ref. (6), the problem whether or not an additional input head increase the power of one-way AMHFAs is posed as an open problem. In this section, we first give a partial solution for this problem and show that $\mathcal{L}[Ak-HFA(s)]\subsetneq\mathcal{L}[A(k+1)-FA(s)]$ for each $k\ge1$ and each $s\ge1$. We need the following lemma.

[Lemma 4.1] For each $r\ge1$, let

$$C(r)=\{w_1 * w_2 *\cdots* w_{2r}\mid\forall i(1\le i\le 2r)[w_i\in\{0,1\}^*]\&\ \forall j(1\le j\le r)[w_j=w_{2r+1-j}]\}$$

and for each string $x=w_1 * w_2 *\cdots* w_{2r}$ in $C(r)$ and for each $i(1\le i\le r)$, let the pair of w_i and w_{2r+1-i} be called twins of x. Then, $Ak-HFA(s)$ can compare all twins of a string in $C(r)$ if and only if $r\le k(k-1)s/2$.

(Proof) By using the same idea as in the formar part of the proof of Theorem 1 in Ref. (4), we can easily seen that 'if' direction holds. To prove 'only if' direction, we assume that $r>k(k-1)s/2$. On the other hand, it is easily seen that if a pair of heads of $Ak-HFA(s)$ is comparing some twins of string x in $C(r)$ at the same time during the computation on x then, at any other time during the computation, that pair of heads could not read any other twins of x. From this fact and the fact that the leaf-size of $Ak-HFA(s)$ is s, we can see that $Ak-HFA(s)$ can compare at most $k(k-1)s/2$ twins of x in $C(r)$. It follows that there must be a twins of a string in $C(r)$ such that $Ak-HFA(s)$ can not compare. This is a contradiction. (Q. E. D.)

[Lemma 4.2] Let $C(r)$ be the set given in Lemma 4.1. Then, for each $k\ge1$ and each $s\ge1$,

(1) $C(k(k-1)s/2+1)\in\mathcal{L}[U(k+1)-HFA(s)]$, and

(2) $C(k(k-1)s/2+1)\notin\mathcal{L}[ASNk-HFA(s)]$.

(Proof) (1) It is clear that

$$(k+1)ks/2-(k(k-1)s/2+1)=ks-1$$

From this and Lemma 4.1, we can get the part (1) of this lemma.

(2) Suppose that there is an $ASNk-HFA(s)$ M accepting $C(k(k-1)s/2+1)$ for each $k\ge1$ and each $s\ge1$. For each $n\ge1$, let

$$V(n)=\{w_1 * w_2 *\cdots* w_{2p}\mid\forall i(1\le i\le 2p)[w_i\in\{0,1\}^*\ \&\ |w_i|^{\dagger}=n]\ \&\ \forall j(1\le j\le p)[w_j=w_{2p+1-j}]\},$$

where $p=k(k-1)s/2+1$.

Note that for each string x in $V(n)$, $|x|=2p(n+1)-1$.

† For any string w, $|w|$ denotes the length of w.

Clearly, each string x in $V(n)$ is in $C(p)$, and so x is accepted by M. Let x be a string in $V(n)$.

The type of a semi-configuration $c=(q,(i_1,i_2,\cdots,i_k))$, denoted by Type(c), is a k-tuple $(\lceil i_1/(n+1)\rceil,\cdots,\lceil i_k/(n+1)\rceil)$. Note that the i-th element h_i of the type specifies that i-th head of M is on $w_{h_i}*$ ($w_{2p}\$$ if $h_i=2p$). For each $j(1\le j\le s)$, let $C_1^j(x), C_2^j(x),\cdots, C_{l_{x,j}}^j(x)$ be the sequence of semi-configuration on the j-th path of an (arbitrarily selected) accepting computation tree of M on input x in $V(n)$. Here $l_{x,j}$ is the length of this path. Let $d_1^j(x), d_2^j(x),\cdots, d_{l_{x,j}}^j(x)$ be the sequence obtained by selecting $C_1^j(x)$ and all subsequent $C_i^j(x)$s such that Type $C_i(x))\ne$ Type $(C_{i+1}(x))$. We call $d_1^j(x), d_2^j(x),\cdots, d_{l_{x,j}}^j(x)$, denoted by $P_j(x)$, the j-pattern of x. For each x in $V(n)$, we let $(P_1(x), P_2(x),\cdots, P_s(x))$ denote the pattern of x. Let $P(n)$ be the number of possible patterns of M on strings in $V(n)$. Since $l_{x,j}\le k(2p-1)+1$ for each x in $V(n)$ and each $j(1\le j\le s)$, we can get the following inequality,

$$P(n)\le\{(u(2p(n+1))^k)^{k(2p-1)+1}\}^s,$$

where u is the number of states of M. Then we classify the strings in $V(n)$ according to their patterns. Clearly, there is a set $S(n)$ ($\subseteq V(n)$) such that $|S(n)|>2^{np}/P(n)$, where each element has the pattern $(\hat{P}_1,\hat{P}_2,\cdots,\hat{P}_s)$.

From Lemma 4.1, we can see that for each string in $V(n)$, there must be an i such that M cannot read w_i* and $w_{2p+1-i}*$ ($w_{2p}\$$ if $i=1$) simultaneously. The possible values for i are determined entirely by the pattern of the computation. Let i_0 be such a value of i for the pattern $(\hat{P}_1,\hat{P}_2,\cdots,\hat{P}_s)$. We now define a binary relation E on string in $S(n)$ as follows. Let

$$u=u_1 * u_2 *\cdots* u_{i_0} *\cdots* u_{2p+1-i_0} *\cdots* u_{2p}\ \text{and}$$

$$v=v_1 * v_2 *\cdots* v_{i_0} *\cdots* v_{2p+1-i_0} *\cdots* v_{2p}.$$

Then, $uEv\Longleftrightarrow\forall i(\notin\{i_0,2p+1-i_0\})[u_i=v_i]$.

Obviously the relation E is an equivalence relation and there are at most $q(n)=2^{n(p-1)}$ E-equivalence classes. It is easily seen that $|S(n)|>q(n)$ for large n. Therefore, there exist two different string which belongs the same equivalence class. Let

$$x=x_1 * x_2 *\cdots* x_{i_0} *\cdots* x_{2p+1-i_0} *\cdots* x_{2p}\ \text{and}$$

$$y=y_1 * y_2 *\cdots* y_{i_0} *\cdots* y_{2p+1-i_0} *\cdots* y_{2p}$$

be such string in $S(n)$. Note that for each $i\notin\{i_0,2p+1-i_0\}$, $x_i=y_i$. Let

$$z=z_1 * z_2 *\cdots* z_{2p}=x_1 * x_2 *\cdots* x_{i_0} *\cdots* x_{2p-i_0} * y_{2p+1-i_0} * x_{2p+2-i_0} *\cdots* x_{2p}$$

obtained by replacing y_{2p+1-i_0} for x_{2p+1-i_0} in x. By an argument similar to that in the proof of Theorem 1 in Ref. (4), it can be shown that there is a accepting computation tree of M on z. Consequently, z must be accepted by M. This contradicts the fact that z is not in $C(p)$. (Q. E. D.)

From Lemma 4.2, we can get the following theorem.

[Theorem 4.1] For each $X \in \{A, U\}$, $k \geq 1$, and $s \geq 1$,

(1) $\mathcal{L}[Xk-HFA(s)] \subsetneq \mathcal{L}[X(k+1)-HFA(s)]$, and

(2) $\mathcal{L}[XSNk-HFA(s)] \subsetneq \mathcal{L}[XSN(k+1)-HFA(s)]$.

It is known that $XYk-HFA(s+1)$s are more powerful than $XYk-HFA(s)$s for each $X \in \{A, U\}$, $Y \in \{SP, SNSP\}$, $k \geq 2$, and $s \geq 1$. We next show that similar result holds for nonsimple one-way AMHFACLs.

[Lemma 4.3] Let $C(r)$ be the set given in Lemma 4.1. Then for each $k \geq 1$ and each $s \geq 1$,

(1) $C(k(k-1)s/2+1) \in \mathcal{L}[Uk-HFA(s+1)]$, and

(2) $C(k(k-1)s/2+1) \notin \mathcal{L}[ASNk-HFA(s)]$.

(Proof) (1) It is clear that

$$(k(k-1)(s+1)/2)-(k(k-1)s/2+1)$$
$$=(k+1)(k-2)/2.$$

From this and Lemma 4.1, we can get the part (1) of this lemma.

(2) The proof is shown in the proof of Lemma 4.2(2). (Q. E. D.)

From Lemma 4.3, we can get the following theorem.

[Theorem 4.2] For each $X \in \{A, U\}$, $k \geq 2$, and each $s \geq 1$,

(1) $\mathcal{L}[Xk-HFA(s)] \subsetneq \mathcal{L}[Xk-HFA(s+1)]$, and

(2) $\mathcal{L}[XSNk-HFA(s)] \subsetneq \mathcal{L}[XSNk-HFA(s+1)]$.

It is unknown whether or not two-way AMHFAs are more powerful than one-way AMHFAs.[6] We next give a partial solution for this problem.

[Theorem 4.3] For each $X \in \{A, U\}$, $k \geq 2$, and $s \geq 1$,

(1) $\mathcal{L}[Xk-HFA(s)] \subsetneq \mathcal{L}[Xk-HTWFA(s)]$ and

(2) $\mathcal{L}[XSNk-HFA(s)] \subsetneq \mathcal{L}[XSNk-HTWFA(s)]$.

(Proof) For each $k \geq 1$ and $s \geq 1$, let $C(k(k-1)s/2+1)$ be the set given in Lemma 4.1. It is easily seen that $C(k(k-1)s/2+1) \in \mathcal{L}[D2-HTWFA]$. On the other hand, it is shown in Lemma 4.2(2) that $C(k(k-1)s/2+1) \notin \mathcal{L}[ASNk-HFA(s)]$. This completes the proof of this theorem. (Q. E. D.)

In Theorem 3.1, we showed that two-way AMHFACLs can be simulated by two-way nondeterministic SPMHFAs. The following theorem shows that stronger results hold for one-way sensing AMHFACLs.

[Theorem 4.4] For each $k \geq 2$ and each $s \geq 1$,

(1) $\mathcal{L}[ASNk-HFA(s)] \subsetneq \mathcal{L}[NSN(ks)-HFA]$ and

(2) $\mathcal{L}[USNk-HFA(s)] \subsetneq \mathcal{L}[DSN(ks)-HFA]$.

(Proof) We omit the proofs of $\mathcal{L}[ASNk-HFA(s)] \subseteq \mathcal{L}[NSN(ks)-HFA]$ and $\mathcal{L}[USNk-HFA(s)] \subseteq \mathcal{L}[DSN(ks)-HFA]$, since they can be proved by using the same technique as in the proof of Theorem 3.1. Let $C(k(k-1)s/2+1)$ be the set given in Lemma 4.1. It is easily seen that $DSN(ks)-HFA$ can compare $ks(ks-1)/2$ twins of a string in $C(k(k-1)s/2+1)$. On the other hand it is clear that

$$ks(ks-1)/2-(k(k-1)s/2+1)=ks(k-1)/2-1.$$

It follows that $C(k(k-1)s/2+1) \in \mathcal{L}[DSN(ks)-HFA]$. It is shown in Lemma 4.2 that $C(k(k-1)s/2+1) \notin \mathcal{L}[ASNk-HFA(s)]$. This completes the proof of the theorem. (Q. E. D.)

It is shown in Ref. (5) that for one-way SPMHFAs, sensing versions are more powerful than non-sensing versions. We show that a similar result also holds for ASPMHFAs with constant leaf-sizes.

[Lemma 4.4] Let $L=\{a^n b^n | n \geq 1\}^*$. Then,

(1) $L \in \mathcal{L}[DSNSP2-HFA]$ and

(2) $L \notin \bigcup_{1 \leq k < \infty} \bigcup_{1 \leq s < \infty} \mathcal{L}[ASPk-HFA(s)]$.

(Proof) (1) The proof of (1) is omitted here, since it is shown in the proof of Lemma 4 in Ref. (5).

(2) Suppose that there is an $ASPk-HFA(s)$ ($k \geq 1$, $s \geq 1$)M accepting L, and that M has t states. We assume without loss of generality that M does not use the left endmarker ¢, since for any $ASPk-HFA(s)B$ with the left and right endmarkers, we can construct an $ASPk-HFA(s)B'$ without the left endmarker such $T(B')=T(B)$.

Consider the word $w=z_1z_2\cdots z_s$, where $z_i=z$ $(1 \leq i \leq s)$, $z=x_1x_2\cdots x_{(k-1)s+1}$, where $x_i=x(1 \leq i \leq (k-1)s+1)$, $x=y_1y_2\cdots y_n$ $(n>t^s)$, and $y_i=a_{i1}a_{i2}\cdots a_{in}b^n$ $(a_{ij}=a$, $1 \leq i, j \leq n)$. Clearly w is in L, so there is an accepting computation tree $T_M(w)$ of M on $w\$$.

Without loss of generality, we assume that each node of $T_M(w)$ which is labeled by a universal configuration has exactly two children. Then, because of the bounded leaf-size s, there are at most $s-1$ nodes labeled by universal configurations in $T_M(w)$. From this fact and the fact that the word w has s subwords z's, there is a subword z in the word w such that on each computation path of $T_M(w)$, there is a sequence of steps which implies that M never enters a universal state while reading the subword z. We let such subword z be z_f.

Let h $(1 \leq h \leq s)$ be the number of sequences of steps while M reads z_f in the accepting computation tree $T_M(w)$, and let $S(1), S(2), \cdots, S(h)$ be these h sequences of steps. For each m $(1 \leq m \leq h)$ and for each i $(1 \leq i \leq (k-1)s+1)$, let $N_m(i)$ be the number of counting heads that reach the right endmarker \$ while the reading head R reads the x_i in z_f, in the m-th sequence $S(m)$. Since M has only $(k-1)$ counting heads and leaf-size s, it follows that $N_1(r)=N_2(r)=\cdots=N_h(r)=0$ for some $r(1 \leq r \leq (k-1)s+1)$.

Consider the case when in the accepting computation tree $T_M(w)$ R reads the subword x_r such that $N_1(r)=N_2(r)=\cdots=N_h(r)=0$. Recall that

$x_r = y_1 y_2 \cdots y_n \quad (n > t^s)$ and

$y_i = a_{i1} a_{i2} \cdots a_{in} b^n \quad (a_{ij} = a, 1 \le i, j \le n).$

For each m $(1 \le m \le h)$ and for each symbol a_{ij}, let q_{ij}^m be the state in which M is when R moves onto a_{ij} on the m-th sequence $S(m)$. For each symbol a_{ij}, we consider the h-tuple of states as follows.

$(q_{ij}^1, q_{ij}^2, \cdots, q_{ij}^h) = Q_{ij}$

We call Q_{ij} above a multi-state of M.

Since $n > t^s$ $(\ge t^h)$, it follows that for each y_i, M must be in the same multi-state at least twice on the y_i i.e., for every $i (1 \le i \le n)$ there exist j, $l (1 \le j < l \le n)$ such that $Q_{ij} = Q_{il}$. For each y_i, let $\bar{Q}_i$ be one such multi-state being repeated and let a_{ij_i} and a_{id_i} be two a's in y_i where the entering multi-state is $\bar{Q}_i$ $(1 \le j_i < d_i \le n)$. Since there are $n (> t^s)$ segments of y's, $\bar{Q}_u = \bar{Q}_v$ for some u and v $(1 \le u < v \le n)$. Note that while R reads x_r, M acts like h independent ordinary one-way finite automata.

Let x_r' be the word obtained from x_r by moving $(d_u - j_u)$ a's from y_u to the segment of a's in $y_{v'}$ and let w' be the word obtained from w by replacing x_r with x_r. It follows from the above note that M will have an accepting computation tree with leaf-size s on w' \$. But $w' \notin L$. This completes the proof of the lemma. (Q. E. D.)

From Lemma 4.4, we can get the following theorem.

[Theorem 4.5] For each $X \in \{A, U\}$, $k \ge 2$, and $s \ge 1$,

(1) $\mathcal{L}[XSPk - HFA(s)] \subsetneq \mathcal{L}[XSNSPk - HFA(s)]$

and

(2) $\bigcup_{1 \le k < \infty} \bigcup_{1 \le s < \infty} \mathcal{L}[XSPk - HFA(s)] \subsetneq \bigcup_{1 \le k < \infty} \bigcup_{1 \le s < \infty} \mathcal{L}$ $[XSNSPk - HFA(s)]$.

It is easily seen that $L \in \mathcal{L}[DSP2 - HSFA]$. From this fact and part (2) of Lemma 4.4, we can get the following theorem.

[Theorem 4.6] For each $X \in \{A, U\}$, $k \ge 2$, and $s \ge 1$.

(1) $\mathcal{L}[XSPk - HFA(s)] \subsetneq \mathcal{L}[XSPk - HSFA(s)]$ and

(2) $\bigcup_{1 \le k < \infty} \bigcup_{1 \le s < \infty} \mathcal{L}[XSPk - HFA(s)] \subsetneq \bigcup_{1 \le k < \infty} \bigcup_{1 \le s < \infty} \mathcal{L}$ $[XSPk - HSFA(s)]$.

5. Conclusions

In this paper, we mainly showed the following results.

(1) For each $k \ge 1$ and each $s \ge 1$

$\mathcal{L}[ASNk - HTWFA(s)] \subsetneq \mathcal{L}[NSP(ks+1) - HTWFA]$.

(2) For each $X \in \{A, U\}$, $k \ge 1$, and $s \ge 1$,

$\mathcal{L}[Xk - HFA(s)] \subsetneq \mathcal{L}[X(k+1) - HFA(s)]$.

(3) For each $X \in \{A, U\}$, $k \ge 2$, and $s \ge 1$,

$\mathcal{L}[XSPk - HFA(s)] \subsetneq \mathcal{L}[XSNSPk - HFA(s)]$.

We conclude this paper by giving several open problems.

(1) How much leaf-size is necessary and sufficient for AMHFAs to be more powerful than ASPMHFAs?

(2) How much leaf-size is necessary and sufficient for sensing ASPMHFAs to be more powerful than non-sensing ASPMHFAs?

(3) $\mathcal{L}[ASPk - HTWFA(s)] \subsetneq \mathcal{L}[ASP(k+1) - HTWFA(s)]$ and $\mathcal{L}[Ak - HTWFA(s)] \subsetneq \mathcal{L}[A(k+1) - HTWFA(s)]$, for each $k \ge 2$, $s \ge 1$?

References

(1) O. H. Ibarra: "On two-way multihead automata", J. Comput. System Sci., **7**, pp. 28-36 (1973).

(2) I. H. Sudborough: "On tape-bounded complexity classes and multihead finite automata", J. Comput. System Sci., **10**, pp. 62-76 (1975).

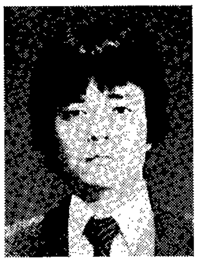
(3) I. H. Sudborough: "One-way multihead writing finite automata", Information and Control, **30**, pp. 1-20 (1976).

(4) A. C. Yao and R. L. Rivest: "$K+1$ heads are better than k", J. ACM, **25**, 2, pp. 337-340 (1978).

(5) K. Inoue, I. Takanami, A. Nakamura and T. Ae: "One-way simple multihead finite automata", Theoret. Comput. Sci. 9, pp. 311-328 (1979).

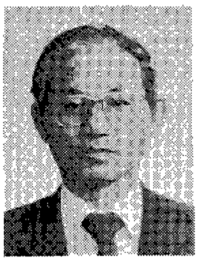
(6) K. N. King: "Alternating multihead finite automata", Lecture Notes in Computer Science, 115, Automata, Languages and Programming, Eighth Colloquium, pp. 506-520 (July 1981).

(7) H. Matsuno, K. Inoue, H. Taniguchi and I. Takanami: "Alternating simple multihead finite automata", Theoret. Comput. Sci., **36**, pp. 291-308 (1985).



Hiroshi Matsuno was born in Yamaguchi Prefecture, on February 17, 1960. He recieved the B.E. and M.E. degrees in electronics engineering from Yamaguchi University in 1982 and 1984, respectively. From 1984 to 1987, he joined Yamaguchi Junior College, Hofu. He has been a research associate of the Department of Electronic-Mechanical Engineering Oshima National College of Maritime Technology. His main interests are the theory of alternating automata.

Katsushi Inoue was born in Yamaguchi Prefecture, on February 4, 1946. He recieved the B. S. and M. S. degrees in electrical engineering from Hiroshima University, in 1969 and 1971, respectively, and the Ph. D. degrees in electrical engineering from Nagoya Univesity in 1977. From 1971 to 1973, he joined Musashino Electrical Communications Laboratories, NTT, Musashino. From 1973 to 1977, he was an Assistant Professor of Hiroshima University. Since 1977 to 1987, he was an Associate Professor in the Department of Electronics of Yamaguchi University, where from 1987 he has been a Professor. His main interests are automata and languages theory.



Itsuo Takanami was born in Fukuoka Prefecture, on March 14, 1937. He recieved the M. E. and D. E. degrees in electrical and communication engineering from Tohoku University, Sendai, in 1967 and 1970, respectively. On March 1970 he joined Tohoku University, where from 1971 to 1974 he was an Associate Professor. He is presently a Professor in the Faculty of Engineering, Yamaguchi University. His current interests are automata theory, parallel processing and microcomputer applications.

Hiroshi Taniguchi was born in Kumamoto Prefecture, on June 22, 1953. He received the B. E. and M. E. degrees in computer science from Kyushu Institute of Technology, Kita-Kushu, in 1976 and 1978, respectively, and the Ph. D. degrees from Nagoya University in 1987. He was a research associate of the Department of Electronics of Yamaguchi University, since 1978 to 1988. He is presently an Associate Professor in the Department of Social Systems Engineering, Tottori University. His main research interests are theory of two-dimensional automata. He is a member of the Information Processing Society of Japan.