JAIST Repository
https://dspace.jaist.ac.jp/

Title	Inkdot versus Pebble over Two-Dimensional Languages
Author(s)	INOUE, Atsuyuki; ITO, Akira; HIRAISHI, Kunihiko; INOUE, Katsushi
Citation	IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences, E88-A(5): 1173-1180
Issue Date	2005-05-01
Type	Journal Article
Text version	publisher
URL	http://hdl.handle.net/10119/4692
Rights	Copyright (C)2005 IEICE. A. Inoue, A. Ito, K. Hiraishi, K. Inoue, IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences, E88-A(5), 2005, 1173-1180. http://www.ieice.org/jpn/trans_online/
Description	

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

PAPER Special Section on Discrete Mathematics and Its Applications

Inkdot versus Pebble over Two-Dimensional Languages

Atsuyuki INOUE[†], ***Nonmember***, **Akira ITO**[†a)], **Kunihiko HIRAISHI**[††], ***and*** **Katsushi INOUE**[†], ***Members***

SUMMARY This paper investigates a relationship between inkdot and one-pebble for two-dimensional finite automata (2-fa's). Especially we show that (1) alternating inkdot 2-fa's are more powerful than nondeterministic one-pebble 2-fa's, and (2) there is a set accepted by an alternating inkdot 2-fa, but not accepted by any alternating one-pebble 2-fa with only universal states.

key words: *inkdot, pebble, two-dimensional automata, nondeterminism, alternation*

1. Introduction

Related to the historical open problem of whether deterministic and nondeterministic space (especially lower-level) complexity classes are separated, Ranjan et al. [6] introduced inkdot Turing machines. An inkdot Turing machine is a conventional Turing machine capable of dropping an inkdot on a given input tape for a landmark, but unable to further pick it up. It was shown in [6] that nondeterministic inkdot Turing machines are more powerful than nondeterministic ordinary Turing machines for spaces between $\log\log n$ and $o(\log n)$, and deterministic inkdot Turing machines have the same accepting power as deterministic ordinary Turing machines for any space bound. Ito et al. [5], on the other hand, introduced inkdot two-dimensional finite automata, which are conventional two-dimensional finite automata (2-fa's) [4] capable of dropping an inkdot on a given two-dimensional input tape, and showed, for example, that (i) alternating (resp., nondeterministic) inkdot 2-fa's are more powerful than ordinary alternating (resp., nondeterministic) 2-fa's, (ii) deterministic inkdot 2-fa's have the same accepting power as ordinary deterministic 2-fa's, (iii) nondeterministic (resp., deterministic) inkdot 2-fa's are less powerful than nondeterministic (resp., deterministic) one-pebble 2-fa's, which were introduced by Blum and Hewitt [1]. See Sect. 3 below for more detailed known and related results concerning inkdot and one-pebble 2-fa's. (See [1], [2], [6]–[8] for another results concerning inkdot and pebble machines.)

Manuscript received August 13, 2004.
Manuscript revised October 17, 2004.
Final manuscript received January 12, 2005.

[†]The authors are with the Department of Computer Science and Systems Engineering, Faculty of Engineering, Yamaguchi University, Ube-shi, 755-8611 Japan.
[††]The author is with School of Information Science, Japan Advanced Institute of Science and Technology, Ishikawa-ken, 923-1292 Japan.
a) E-mail: ito@csse.yamaguchi-u.ac.jp

DOI: 10.1093/ietfec/e88–a.5.1173

Let $AIFA$ denote the class of sets of two-dimensional tapes accepted by alternating inkdot 2-fa's, and let $APFA$ (resp., $NPFA$, $UPFA$) denote the class of sets of two-dimensional tapes accepted by alternating one-pebble 2-fa's (resp., nondeterministic one-pebble 2-fa's, alternating one-pebble 2-fa's with only universal states). It is shown in [5] that $NPFA$ is a subset of $AIFA$, but it is unknown whether $NPFA$ is properly contained in $AIFA$. One main purpose of this paper is to solve this problem, and show, in Sect. 4, that $NPFA$ is a proper subclass of $AIFA$. Another purpose of this paper is to investigate a relationship between $UPFA$ and $AIFA$, and show, in Sect. 5, that there is a set in $AIFA$, but not in $UPFA$. As a corollary of this result, we have a new result that $APFA$ propely contains $UPFA$. Section 6 concludes this paper by giving open problems.

2. Definitions and Notations

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 $l_1(x)$ be the number of rows of x, and $l_2(x)$ be the number of columns of x. For each $m,\ n \geq 1$, let $\Sigma^{(m,n)} = \{x \in \Sigma^{(2)} \mid l_1(x) = m \ \wedge \ l_2(x) = n\}$. If $1 \leq i \leq l_1(x)$ and $1 \leq j \leq l_2(x)$, we let $x(i, j)$ denote the symbol in x with coordinates (i, j). Furthermore, we define $x[(i, j), (i', j')]$, only when $1 \leq i \leq i' \leq l_1(x)$ and $1 \leq j \leq j' \leq l_2(x)$, as the two-dimensional tape z satisfying the following:

(1) $l_1(z) = i' - i + 1$ and $l_2(z) = j' - j + 1$;
(2) for each $k, r (1 \leq k \leq l_1(z), 1 \leq r \leq l_2(z))$, $z(k, r) = x(k + i - 1, r + j - 1)$.

For any two two-dimensional tapes x and y with $l_1(x) = l_1(y)$, we denote by xy the two-dimensional tape obtained by concatenating y to the right of x. Below, We denote a two-dimensional finite automaton by 2-fa.

An *alternating* 2-fa [4] is a sixtuple $M = (Q, q_0, U, F, \Sigma, \delta)$, 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* ($\sharp \notin \Sigma$ is the *boundary symbol*), and (6) $\delta \subseteq (Q \times (\Sigma \cup \{\sharp\})) \times (Q \times \{\text{left,right,up,down,no move}\})$ 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 $\sharp$, and a finite control. A position is assigned to each cell of the input

Copyright © 2005 The Institute of Electronics, Information and Communication Engineers

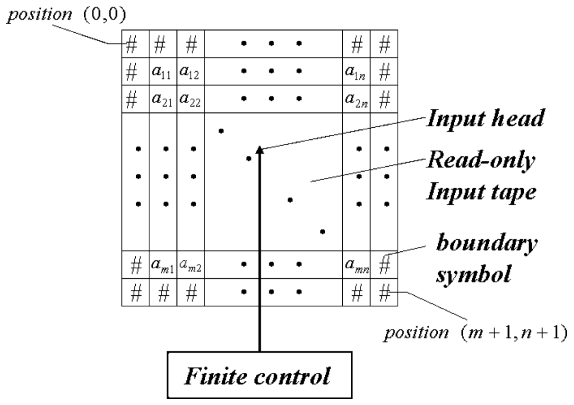


Fig. 1 Two-dimensional finite automaton.

tape as shown in Fig. 1.

At each moment, the machine M is in one of the states. A *step* of M consists of reading the symbol currently under the input head, changing its state, and moving the input head in specified direction (left, right, up, down, or no move) which is determined by the next-move relation δ. If the input head falls off the input tape, then M can make no further move.

A *configuration* of an alternating 2-fa M on an input $x \in \Sigma^{(2)}$ is of the form $((i, j), q)$, where (i, j), $0 \le i \le l_1(x)+1$ and $0 \le j \le l_2(x) + 1$, is a position of the input head, and q is a state of the finite control. 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 $I_M(x) = ((1, 1), q_0)$. For each input tape x, we write $c \vdash_{M,x} c'$, and say that c' is an *immediate successor* of c (of M on x), if configuration c' is derived from configuration c in one step of M on x according to the next-move relation. A configuration with no immediate successor is called a *halting* configuration. Below, we assume that every accepting configuration is a halting configuration.

We can view the computation of M as a tree whose nodes are labelled by configurations. A *computation tree* of M on an input tape x is a tree whose nodes are labelled by configurations of M on x. The root of the tree is labelled by the initial configuration $I_M(x)$; the children of any node labelled by a universal configuration are all the immediate successors of that configuration on x; and any node labelled by an existential configuration has one child, which is labelled by one of the immediate successors of that configuration on x (provided there are any). An *accepting computation tree* of M on x is a computation tree of M on x whose leaves are all labelled by accepting configurations. We say that M *accepts* x if there is an acceptint computation tree of M on input x. Define $T(M) = \{x \in \Sigma^{(2)} | M \text{ accepts } x\}$.

An *alternating one-pebble* 2-fa [5] is an alternating 2-fa with the capability of using one-pebble which the finite control can use as a marker on the input tape. During the computation, the device can deposit (retrieve) a pebble on (from) any cell of the tape. The action of the machine depends on the current state of the finite control, the currently scanned input tape symbol, and on the presence of the pebble on the current input tape cell. The action consists of moving the input head, changing the state of the finite control, and picking up or placing the pebble on the currently scanned cell of the input tape. A *configuration* of an alternating one-pebble 2-fa M on an input tape x is of the form $((i, j), \textit{pebble-position}, q)$, where (i, j) is the input head position, *pebble-position* is the position of the pebble on x (let *pebble-position* be "*no*" if the pebble is not placed on the input tape x), and component q represents a state of the finite control. The *initial configuration* of M on x is $((1, 1), no, q_0)$, where q_0 is the initial state of M. That is, the machine M starts with the pebble in the finite control and with the input head on the upper-leftmost corner of the input tape. An *accepting computation tree* of M on an input tape is defined as in the case of an alternating 2-fa. We say that M *accepts* an input tape x if there is an accepting computation tree of M on x. By $T(M)$, we denote the set of all the two-dimensional tapes accepted by M.

An *alternating inkdot* 2-fa [5] is an alternating 2-fa capable of dropping an inkdot on a given input tape for a landmark, but unable to further pick it up. That is, an alternating inkdot 2-fa is an alternating one-pebble 2-fa which cannot pick up the pebble again, once it has put down the pebble on a given input tape. See [5] for the formal definitions of alternating one-pebble and inkdot 2-fa's.

A *nondeterministic* 2-fa is an alternating 2-fa which has no universal states, and a *deterministic* 2-fa is an alternating 2-fa whose configurations each have at most one immediate successor. Nondeterminism and determinism for one-pebble and inkdot 2-fa's are defined similarly.

By AFA (resp., NFA, DFA, $APFA$, $NPFA$, $DPFA$, $AIFA$, $NIFA$, $DIFA$), we denote the class of sets of two-dimensional tapes accepted by alternating (resp., nondeterministic, deterministic, alternating one-pebble, nondeterministic one-pebble, deterministic one-pebble, alternating inkdot, nondeterministic inkdot, deterministic inkdot) 2-fa's. Furthermore, by UFA (resp., $UPFA$, $UIFA$), we denote the class of sets of two-dimensional tapes accepted by alternating (resp. alternating one-pebble, alternating inkdot) 2-fa's with only universal states.

Let M be an alternating one-pebble (inkdot) 2-fa, and x be an input tape. A sequence of configurations $c_1 c_2 \dots c_m$ $(m \ge 1)$ is called a *computation path of M on x* if $c_1 \vdash_{M,x} c_2 \vdash_{M,x} \dots \vdash_{M,x} c_m$. For simplicity, we below call a computation path a *computation*. For any set S, $|S|$ denotes the cardinality of S.

3. Known Results and Related Results

This section surveys known results and related results in [3], [5] concerning inkdot and pebble 2-fa's.

The following result in [5] shows a relationship among the accepting powers of 2-fa's, inkdot 2-fa's, and one-pebble 2-fa's.

Theorem 3.1. [5].

(1) $DFA = DIFA \subsetneqq DPFA$,
(2) $NFA \subsetneqq NIFA \subsetneqq NPFA$,
(3) $UFA \subsetneqq UIFA \subsetneqq UPFA$, and
(4) $AFA \subsetneqq AIFA \subseteq APFA$.

It is unknown whether $AIFA \subsetneqq APFA$. What are the relationships between $NIFA$ and $DPFA$, and between $UIFA$ and $DPFA$? The following theorem answers this question:

Theorem 3.2.
(1) $NIFA$ is incomparable with $DPFA$, and
(2) $UIFA$ is incomparable with $DPFA$.

Proof. Let $T_1 = \{x \in \{0,1\}^{(2)} \mid \exists n \geq 1\ [l_1(x) = l_2(x) = 2n \wedge x[(1,1),(n,2n)] = x[(n+1,1),(2n,2n)]$(i.e., the top half of x is the same as the bottom half of x)$]\}$, and $T_2 = \{x \in \{0,1\}^{(2)} \mid \exists n \geq 2\ [l_1(x) = n \wedge \exists i\ (2 \leq i \leq n)\ [x[(1,1),(1,l_2(x))] = x[(i,1),(i,l_2(x))]$(i.e., the first row of x s the same as the i-th row of x]$]\}$. It is shown in [5] that the complement of T_1 is in $NIFA$, $T_1 \in UIFA$, and $T_2 \notin NIFA \cup UIFA$. Furthermore, it is shown in [3] that $T_2 \in DPFA$. By using the same idea as in the proof of Theorem 4.1 in [3], we can show that the complement of T_1 is not in $DPFA$. From these observations, the theorem follows. ∎

In Sect. 4 (resp., Sect. 5), we investigate a relationship between $NPFA$ and $AIFA$ (resp., $UPFA$ and $AIFA$). It is unknown what are the relationships between $NIFA$ and $UPFA$, and between $UIFA$ and $NPFA$.

The following result in [5] shows a relationship among the accepting powers of determinism, nondeterminism, alternation, and alternation with only universal states for inkdot 2-fa's.

Theorem 3.3. [5].
(1) $DIFA \subsetneqq NIFA \subsetneqq AIFA$, and
(2) $DIFA \subsetneqq UIFA \subsetneqq AIFA$.

A relationship between $NIFA$ and $UIFA$ is shown in the following theorem:

Theorem 3.4. $NIFA$ is incomparable with $UIFA$.

Proof. Let T_1 be the set described in the proof of Theorem 3.2. It is implicitely shown in [5] that $T_1 \in UIFA - NIFA$, and the complement of T_1 is in $NIFA$, but not in $UIFA$. From this fact, the theorem follows. ∎

As shown in the following result in [3], also for one-pebble 2-fa's, alternation is better than nondeterminism, which is better than determinism.

Theorem 3.5. [3]. $DPFA \subsetneqq NPFA \subsetneqq APFA$.

As a corollary of the main result in Sect. 5, we have "$DPFA \subsetneqq UPFA \subsetneqq APFA$." We conjecture that $NPFA$ is incomparable with $UPFA$, but we have no proof of this conjecture.

4. Alternating Inkdot Versus Nondeterministic One-Pebble

This section shows that alternating inkdot is better than nondeterministic one-pebble for 2-fa's.

Theorem 4.1. $NPFA \subsetneqq AIFA$.

Proof. It is shown in [5] that $NPFA \subseteq AIFA$. To prove the theorem, we below show that $AIFA - NPFA \neq \phi$. For each $m \geq 1$, let $V(m) = \{x_1c_1x_2c_2 \ldots x_mc_m | \forall i(1 \leq i \leq m)[x_i \in \{0,1\}^{(m,m)} \wedge c_i \in \{2\}^{(m,1)}]\}$, and $T_3 = \{xx | \exists m \geq 1[x \in V(m)]\}$. We prove that

(1) $T_3 \in AIFA$, and
(2) $T_3 \notin NPFA$.

Proof of (1): T_3 is accepted by an alternating inkdot 2-fa M which acts as follows. Suppose that an input tape

$$w = x_1c_1x_2c_2 \ldots x_kc_kx_{k+1}c_{k+1} \ldots x_{2k}c_{2k}$$

is presented to M, where $k \geq 1$, $x_i \in \{0,1\}^{(m,m)}$ and $c_i \in \{2\}^{(m,1)}$, $m \geq 1$, for each $1 \leq i \leq 2k$. (Input tapes of the form different from the above can easily be rejected by M.) M first checks that $k = m$. To do so, M first moves the input head H on the symbol $w(1,1)$. After that, moving H one row down each time H reads symbol "2" twice, M moves H in the right-lower direction. Acting like this, if H reaches the symbol "2" situated on the right-lower corner of w, then M can know that $k = m$.

After knowing that $k = m$, M tries to check that the left and right halves of the input tape w are the same, i.e., that $x_i(r,s) = x_{m+i}(r,s)$ for every i, r, s $(1 \leq i, r, s \leq m)$. To do so, M first moves H to the right-lower corner of x_m. Then, scanning the left segment $x_1c_1x_2c_2 \ldots x_m$ of w from right to left, M universally makes process $P_i(r,s)$ on the symbol $x_i(r,s)$ for every $1 \leq i, r, s \leq m$. Process $P_i(r,s)$ is used to check that $x_i(r,s) = x_{m+i}(r,s)$. In order to check that $x_i(r,s) = x_{m+i}(r,s)$, $P_i(r,s)$ drops the inkdot on the symbol $x_i(r,s)$. After that, $P_i(r,s)$ tries to move H to the s-th column of x_{m+i}. For the purpose, $P_i(r,s)$ first moves H on the symbol $x_{m+i}(m,1)$ as follows:

1-1. Move H to the first row and to the right on the first row of x_i until H meets a symbol 2. After that, go to step 1-2.
1-2. Move H one cell down, and go to step 1-3.
1-3. If H reads a symbol 2, then go to step 1-4. If H reads a bottom boundary symbol #, then go to step 1-5.
1-4. Move H to the right on the current row until H meets a symbol 2. After that, go to step 1-2.
1-5. Move H one cell up, and then one cell right.

After executing step 1-5, process $P_i(r,s)$ has moved H on the symbol $x_{m+i}(m,1)$. Then, in order to guess the s-th column of x_{m+i}, $P_i(r,s)$ existentially chooses one of the following two actions on each symbol of the m-th row of x_{m+i}:

(i) (Guessing that the current column is not the s-th column of x_{m+i}) move H one cell right.
(ii) Guess that the current column is the s-th column of x_{m+i}.

If $P_i(r,s)$ continues to choose the action (i) above and H meets a symbol 2, then $P_i(r,s)$ halts without entering an accepting state. If $P_i(r,s)$ chooses the action (ii) above, then $P_i(r,s)$ universally branches into two sub-processes $P_i(r,s)_1$ and $P_i(r,s)_2$. Sub-process $P_i(r,s)_1$ tries to check whether the current column is the s-th column of x_{m+i}, and if so, then enters an accepting state. Otherwise, $P_i(r,s)_1$ halts without entering an accepting state. That is, $P_i(r,s)_1$ acts as follows:

2-1. Move H diagonally in the left-upper direction. Then,

(i) if H meets a top boundary symbol #, then move H one cell left (note that at this time, H has been moved $m+1$ cells left compared with the position just before executing step 2-1), and go to step 2-2, and
(ii) if H meets a left boundary symbol #, then halt without entering an accepting state.

2-2. Move H downwards. In this course, if H meets the inkdot, then enter an accepting state. If H reaches a bottom boundary symbol # without meeting the inkdot, then move H one cell up, and go to step 2-1.

Sub-pross $P_i(r,s)_2$ existentially chooses one of the following two actions on each symbol of the current column (hopefully, the s-th column) of x_{m+i}:

(i) Move H one cell up.
(ii) Guess that the current row is the r-th row of x_{m+i}.

If $P_i(r,s)_2$ continues to choose the action (i) above and H meets a top boundary symbol #, then $P_i(r,s)_2$ halts without entering an accepting state. If $P_i(r,s)_2$ chooses the action (ii) above, then it memoririzes the symbol under H in the finite control, and continues to move H to the left. If H reaches a left boundary symbol # without meeting the inkdot, then $P_i(r,s)_2$ halts without entering an accepting state. If H meets the inkdot somewhere, then $P_i(r,s)_2$ compares the symbol memorized in the finite control with the symbol $x_i(r,s)$ under the inkdot. $P_i(r,s)_2$ enters an accepting state only if these symbols are the same. It is obvious that M accepts T_3.

Proof of (2). The proof borrows an idea in the proof of Theorem 4.1 in [3]. We suppose to the contrary that a nondeterministic one-pebble 2-fa M accepts T_3. Let Q be the set of states of the finite control of M. We divide Q into two disjoint subsets Q^+ and Q^- which corresponds to the sets of states when M holds and does not hold the pebble in the finite control, respectively. M starts from the initial state in Q^+ with the input head on the upper-leftmost symbol of an input tape. We assume without loss of generality that M satisfies the following condition (A):

(A) M does not go out of the boundary symbols #.(Of course, M does not go into the input tape from the outside of the boundary symbols #.) Furthermore, when M accepts an input tape in T_3, M enters an accepting state in Q^+ on a right boundary symbol #, and halts.

For each $m \geq 1$, let $W(m) = \{xy \mid x, y \in V(m)\}$. Below we shall consider the computations of M on tapes in $W(m)$ for large m. Let x be any tape in $V(m)$ that is supposed to be a left or right half of an input tape (in $W(m)$) to M, and let $x(\#)$ be the tape obtained from x by attaching the boundary symbols # to the upper and lower sides of x. That is, $x(\#)$ is a two-dimensional tape over $\{0,1,2,\#\}$ such that $l_1(x(\#)) = m+2, l_2(x(\#)) = l_2(x) = m(m+1), x(\#)[(2,1),(m+1,m(m+1))] = x$, and $x(\#)(1,k) = x(\#)(m+2,k) = \#$ for each $k(1 \leq k \leq m(m+1))$. Note that, from the above condition (A), the entrance points to $x(\#)$ (or the exit points from $x(\#)$) for M are the left and right sides of $x(\#)$. Let $PT(m)$ be the set of these entrance points (or exit points). Clearly, $|PT(m)| = 2(m+2)$. Suppose that the pebble of M is not placed on this $x(\#)$. Then, we define a mapping M_x, which depends on M and x, from $Q^- \times PT(m)$ to the power set of $Q^- \times PT(m)$ as follows:

- $(s', p') \in M_x(s,p) \Leftrightarrow$ when M enters $x(\#)$ in state s from entrance point p, there exists a sequence of steps of M in which M eventually exits $x(\#)$ in state s' from exit point p'.

Let $x, y \in V(m)$. We say that x and y are M^--*equivalent* if $M_x(s,p) = M_y(s,p)$ for each $(s,p) \in Q^- \times PT(m)$ (i.e., if two mappings M_x and M_y are the same). Thus, M cannot distinguish between two tapes in $V(m)$ which are M^--equivalent, without the pebble. M^--equivalence is an equivalence relation on $V(m)$, and $|V(m)| = 2^{m^3}$. Clearly, there are at most

$$e(m) = (2^{|Q^-|\times 2(m+2)})^{|Q^-|\times 2(m+2)}$$

M^--equivalence classes of tapes in $V(m)$. Let $P(m)$ be a largest M^--equivalence class of tapes in $V(m)$. Then we have

$$|P(m)| \geq \frac{2^{m^3}}{e(m)}. \tag{1}$$

Note that $|P(m)| \gg 1$ for large m. Let x_1 and x_2 be in $P(m)$. For any computation $comp(x_1x_2)$ of M on x_1x_2, let

- $cross(comp(x_1x_2)) \stackrel{\triangle}{=}$ the sequence of pairs of (i) states and (ii) cross-points of the input head when M crosses the boundary between $x_1(\#)$ and $x_2(\#)$ from left to right or from right to left in $comp(x_1x_2)$, and
- *pebble-cross*$(comp(x_1x_2)) \stackrel{\triangle}{=}$ the sequence of pairs of (i) states (in Q^+) and (ii) cross-points of the input head when M crosses the boundary between $x_1(\#)$ and $x_2(\#)$ with the pebble in the finite control from left to right or from right to left in $comp(x_1x_2)$.

Of course, *pebble-cross*$(comp(x_1x_2))$ is a subsequence of $cross(comp(x_1x_2))$.

For each $x \in P(m)$, xx is in T_3, and so it must be accepted by M. Therefore, there exists an accepting computation of M on xx. Let "$accomp(xx)$" be such a fixed loop-free accepting computation of M on xx. Since *pebble-cross*$(accomp(xx))$ is loop-free, it follows that *pebble-cross*$(accomp(xx))$ is such that the same pair of (i) a state (in Q^+) and (ii) a cross-point of the input head appears at most twice (one is with M crossing the boundary between the left $x(\#)$ and the right $x(\#)$ from left to right (or from right to left), and the other is with M crossing the boundary from right to left (or from left to right) in *pebble-cross*$(accomp(xx))$. Therefore, the length of *pebble-cross*$(accomp(xx))$ is bounded by $2|Q^+|(m+2)$. For each $m \gg 1$, let $PEBBLE\text{-}CROSS(m) =$

$\{pebble\text{-}cross(accomp(xx))|x \in P(m)\}$. From the observation above, it follows that

$$|PEBBLE\text{-}CROSS(m)| \leq (|Q^+|(m+2))^{2|Q^+|(m+2)}. \quad (2)$$

By a simple calculation, it follows from inequalities (1) and (2) that for large m, we have $|P(m)| \gg |PEBBLE\text{-}CROSS(m)|$. Thus, there must be two different tapes x and y in $P(m)$ such that $pebble\text{-}cross(accomp(xx)) = pebble\text{-}cross(accomp(yy))$. From condition (A) mentioned above, we can assume without loss of generality that for some *odd* number $k \geq 1$,

(i) $pebble\text{-}cross(accomp(xx)) = pebble\text{-}cross(accomp(yy)) = p_1 p_2 \dots p_k$ (each $p_i \in Q^+ \times PT(m)$),
(ii) $cross(accomp(xx)) = p^x_{01} p^x_{02} \cdots p^x_{0i_0} p_1 p^x_{11} p^x_{12} \cdots p^x_{1i_1} p_2 p^x_{21} p^x_{22} \cdots p^x_{2i_2} p_3 \dots p_k p^x_{k1} p^x_{k2} \cdots p^x_{ki_k}$ ($i_0, i_1, \dots, i_k \geq 0$, and each $p^x_{ij} \in Q^- \times PT(m)$),
(iii) $cross(accomp(yy)) = p^y_{01} p^y_{02} \cdots p^y_{0j_0} p_1 p^y_{11} p^y_{12} \cdots p^y_{1j_1} p_2 p^y_{21} p^y_{22} \cdots p^y_{2j_2} p_3 \dots p_k p^y_{k1} p^y_{k2} \cdots p^y_{kj_k}$ ($j_0, j_1, \dots, j_k \geq 0$, and each $p^y_{ij} \in Q^- \times PT(m)$), and
(iv) $accomp(yy)$ ends with an accepting state $q_a \in Q^+$ on a right boundary symbol #.

Since x and y are M^--equivalent, by combining $accomp(xx)$ and $accomp(yy)$, we can construct an accepting computation $comp(xy)$ of M on xy such that:

(i) $cross(comp(xy)) = p^x_{01} p^x_{02} \cdots p^x_{0i_0} p_1 p^y_{11} p^y_{12} \cdots p^y_{1j_1} p_2 p^x_{21} p^x_{22} \cdots p^x_{2i_2} p_3 p^y_{31} p^y_{32} \cdots p^y_{3j_3} p_4 \cdots p_k \ p^y_{k1} p^y_{k2} \cdots p^y_{kj_k}$, and
(ii) $pebble\text{-}cross(comp(xy)) = p_1 p_2 \dots p_k$, and
(iii) $comp(xy)$ ends with M entering the accepting state $q_a \in Q^+$ on the right boundary symbol #.

(A similar idea is used in the proof of Lemma 4.1 in [3].) Therefore, xy would be also accepted by M. This contradicts the fact that xy is not in T_3. This completes the proof of (2). ∎

5. *UPFA* Versus *AIFA*

This section investigates a relationship between $UPFA$ and $AIFA$.

Here is some preliminary. Let M be an alternating one-pebble 2-fa with only universal states, and let $c_1 c_2 \dots c_m$ ($m \geq 1$) be a computation of M on an input tape x. Then, this computation is called:

- a *halting computation* of M on x if c_m is a halting configuration other than any accepting confuguration,
- a *double-looping computation* of M on x if there exist some i ($1 \leq i \leq m-2$) and some (possibly empty) sequence of confugurations s such that (i) $c_j \neq c_k$ for each $1 \leq j < k \leq i$, (ii) $c_1 c_2 \dots c_m = c_1 c_2 \dots c_{i-1} c_i s c_i s c_i$ (we call $c_i s c_i s c_i$ the *double-looping segment* of the computation), and (iii) each configuration in $c_i s$ is different from each other, and different from each c_r ($1 \leq r < i$), and
- a *rejecting computation* of M on x if the sequence $c_1 c_2 \dots c_m$ is a halting, or double-looping computation.

We note the following: Since M has only universal states, M does not accept an input if and only if there is a halting computation (defined here) or a nonterminating computation. There is a nonterminating computation if and only if there is a double-looping computation. Hence, M does not accept an input if and only if there is a rejecting computation (defined here).

Theorem 5.1. $AIFA - UPFA \neq \phi$.

Proof. Let $V(m)$, $m \geq 1$, be the set described in the proof of Theorem 4.1, and $T_4 = \{xy \mid \exists m \geq 1\ [x,\ y \in V(m)] \ \wedge \ x \neq y\}$. To prove the theorem, we below show that

(1) $T_4 \in AIFA$, and
(2) $T_4 \notin UPFA$.

Proof of (1): The proof is almost the same as the proof of (1) in the proof of Theorem 4.1. The only difference is to check that $x_i(r,s) \neq x_{m+i}(r,s)$ for some i, r, s ($1 \leq i, r, s \leq m$). This check can be done by using a process similar to process $P_i(r,s)$ in the proof of (1) in the proof of Theorem 4.1. The details are omitted here.

Proof of (2): The proof is essentially similar to the proof of (2) in the proof of Theorem 4.1, where we derived a contradiction by constructing an accepting computation on $xy \notin T_3$ by combining accepting computations on xx and yy. We suppose to the contrary that there is an alternating one-pebble 2-fa with only universal states M which accepts T_4. Here, we derive a contradiction by constructing a rejecting computation (defined above) of M on $xy \in T_4$ by combining rejecting computations of M on xx and yy (where $x, y \in V(m)$ for large m). Let Q be the set of states of the finite control of M, and Q^+ and Q^- be defined as in the proof of Theorem 4.1. M starts from the initial state in Q^+ with the input head on the upper-leftmost symbol of an input tape. We assume without loss of generality that M satisfies the following condition (B):

(B) M does not go out of the boundary symbols #.(Of course, M does not go into the input tape from the outside of the boundary symbols #.)

For each $m \geq 1$, let $W(m) = \{xy \mid x, y \in V(m)\}$. Below we shall again consider the computations of M on tapes in $W(m)$ for large $m \geq 1$. Let x be any tape in $V(m)$ that is supposed to be a left or right half of an input tape (in $W(m)$) to M, and let $\#x$ (resp., $x\#$) be the tape obtained from x by attaching the boundary symbols # to the left, upper and lower (resp., right, upper and lower) sides. That is, for example, $\#x$ is a two-dimensional tape over $\{0, 1, 2, \#\}$ such that $l_1(\#x) = m+2$, $l_2(\#x) = l_2(x) + 1 = m(m+1)+1$, $\#x[(2,2), (m+1, m(m+1)+1)] = x$, $\#x(1,k) = \#x(m+2,k) = \#$ for each k ($1 \leq k \leq m(m+1)+1$), and $\#x(r,1) = \#$ for each r ($1 \leq r \leq m+2$). Note that, from the above condition (B), both the entrance points to $\#x$ (resp., $x\#$) and the exit points from $\#x$ (resp., $x\#$) are the right (resp., left) side of $\#x$ (resp.,

$x\#$). Let $PT(m)$ be the set of these entrance (or exit) points. Clearly, $|PT(m)| = m + 2$. Suppose that the pebble of M is not placed on this $\#x$ (resp., $x\#$). Then, we define a mapping M_x^l (resp., M_x^r), which depends on M and x, from $Q \times PT(m)$ to the power set of $(Q \times PT(m)) \cup Q_{stop} \cup \{loop\}$ as follows (where Q_{stop} is the set of halting states other than accepting states, and $loop$ is a new symbol):

- for any (s, p), $(s', p') \in Q^- \times PT(m)$, $(s', p') \in M_x^l(s, p)$ (resp., $M_x^r(s, p)$) $\Leftrightarrow$ when M enters $\#x$ (resp., $x\#$) in state s from entrance point p of the right (resp., left) side of $\#x$ (resp., $x\#$), there exists a computation of M in which M eventually exits $\#x$ (resp., $x\#$) in state s' from exit point p' of the right (resp., left) side of $\#x$ (resp., $x\#$),
- for any $(s, p) \in Q \times PT(m)$ and for any $s' \in Q_{stop}$, $s' \in M_x^l(s, p)$ (resp., $M_x^r(s, p)$) $\Leftrightarrow$ when M enters $\#x$ (resp., $x\#$) in state s from entrance point p of the right (resp., left) side of $\#x$ (resp., $x\#$), there exists a computation of M in which M eventually enters state s' in $\#x$ (resp., $x\#$), and halts, and
- for any $(s, p) \in Q \times PT(m)$, $loop \in M_x^l(s, p)$ (resp., $M_x^r(s, p)$) $\Leftrightarrow$ when M enters $\#x$ (resp., $x\#$) in state s from entrance point p of the right (resp., left) side of $\#x$ (resp., $x\#$), there exists a computation in which M enters a loop in $\#x$ (resp., $x\#$).

Let $x_1, x_2 \in V(m)$. We say that x_1 and x_2 are

- *M-equivalent* if two mappings $M_{x_1}^l$ and $M_{x_2}^l$ are the same, and two mappings $M_{x_1}^r$ and $M_{x_2}^r$ are the same, and
- *M^--equivalent* if for any (s, p), $(s', p') \in Q^- \times PT(m)$ and for any $a \in \{l, r\}$, $(s', p') \in M_{x_1}^a(s, p)$ if and only if $(s', p') \in M_{x_2}^a(s, p)$.

(Note that if x_1 and x_2 are M-equivalent, then x_1 and x_2 are M^--equivalent.) Clearly, M-equivalence is an equivalence relation on $V(m)$. Clearly, there are at most $e(m) = ((2^{|Q|(m+2)+d+1})^{|Q|(m+2)})^2$, where $d = |Q_{stop}|$, M-equivalence classes of $V(m)$. Let $P(m)$ be a largest M-equivalence class of $V(m)$. Then we have $|P(m)| \geq \frac{|V(m)|}{e(m)} = \frac{2^{m^3}}{e(m)}$. Note that $|P(m)| \gg 1$ for large m. Let x_1 and x_2 be in $P(m)$. For any computation $comp(x_1x_2)$ of M on x_1x_2, let $cross(comp(x_1x_2))$ and $pebble\text{-}cross(comp(x_1x_2))$ be defined as in the proof of (2) in the proof of Theorem 4.1.

For each pair p_i in $cross(comp(x_1x_2)) = p_1p_2 \ldots p_i \ldots$, let

- $comp(x_1x_2)[-, p_i] \stackrel{\triangle}{=}$ the sub-computation of $comp(x_1x_2)$ from the beginning of $comp(x_1x_2)$ to the moment of M crossing the boundary between $\#x_1$ and $x_2\#$ in pair p_i, and
- $comp(x_1x_2)[p_i, -] \stackrel{\triangle}{=}$ the sub-computation of $comp(x_1x_2)$ after the moment of M crossing the boundary between $\#x_1$ and $x_2\#$ in pair p_i.

For any pairs p_i and p_j $(i < j)$ in $cross(comp(x_1x_2)) = p_1p_2 \ldots p_i \ldots p_j \ldots$, let

- $comp(x_1x_2)[p_i, p_j] \stackrel{\triangle}{=}$ the sub-computation of $comp(x_1x_2)$ from the moment of M crossing the boundary between $\#x_1$ and $x_2\#$ in pair p_i to the moment of M crossing again the boundary in pair p_j.

For each $x \in P(m)$, xx is not in T_4, and so it must be rejected by M. Therefore, there exists a rejecting computation of M on xx. Let "$recomp(xx)$" be such a fixed rejecting computation of M on xx. Since a rejecting computation contains at most three same configurations, in $recomp(xx)$, M crosses at most three times the boundary between $\#x$ and $x\#$ in the same direction and with the same pair of (i) a state (in Q^+) and (ii) a cross-point of the input head. This implies that the same pair appears at most six times in $pebble\text{-}cross(recomp(xx))$. Therefore, the length of $pebble\text{-}cross(recomp(xx))$ is bounded by $6|Q^+|(m + 2)$. For each $m \gg 1$, let $PEBBLE\text{-}CROSS(m) = \{pebble\text{-}cross(recomp(xx)) | x \in P(m)\}$. From the observation above, it follows that $|PEBBLE\text{-}CROSS(m)| \leq (|Q^+|(m + 2))^{6|Q^+|(m+2)}$. By a simple calculation, it follows that for large m, we have $|P(m)| \gg |PEBBLE\text{-}CROSS(m)|$. Thus, there must be two different words x and y in $P(m)$ such that $pebble\text{-}cross(recomp(xx)) = pebble\text{-}cross(recomp(yy))$. We below derive a contradiction by showing that a computation of M on xy which forces the tape xy to be rejected can be constructed by combining $recomp(xx)$ and $recomp(yy)$, and thus xy would be rejected by M. We only consider the case where for some *odd* number $k \geq 1$,

(i) $pebble\text{-}cross(recomp(xx)) = pebble\text{-}cross(recomp(yy)) = p_1p_2 \ldots p_k$ (each $p_i \in Q^+ \times PT(m)$),
(ii) $cross(recomp(xx)) = p_{01}^x p_{02}^x \cdots p_{0i_0}^x p_1 p_{11}^x p_{12}^x \cdots p_{1i_1}^x p_2 p_{21}^x p_{22}^x \cdots p_{2i_2}^x p_3 \ldots p_k p_{k1}^x p_{k2}^x \cdots p_{ki_k}^x$ $(i_0, i_1, \ldots, i_k \geq 0,$ and each $p_{ij}^x \in Q^- \times PT(m))$, and
(iii) $cross(recomp(yy)) = p_{01}^y p_{02}^y \cdots p_{0j_0}^y p_1 p_{11}^y p_{12}^y \cdots p_{1j_1}^y p_2 p_{21}^y p_{22}^y \cdots p_{2j_2}^y p_3 \cdots p_k p_{k1}^y p_{k2}^y \cdots p_{kj_k}^y$ $(j_0, j_1, \ldots, j_k \geq 0,$ and each $p_{ij}^y \in Q^- \times PT(m))$.

For other cases, a similar idea is used to derive a contradiction. Note that for each $w \in \{x, y\}$,

(i) in $recomp(ww)[-, p_1]$ and in $recomp(ww)[p_i, p_{i+1}]$ for each even number $i, 2 \leq i \leq k - 1$, the pebble is on the left half $\#w$ of the input $\#ww\#$, and
(ii) in $recomp(ww)[p_i, p_{i+1}]$ for each odd number $i, 1 \leq i \leq k - 2$, and in $recomp(ww)[p_k, -]$, the pebble is on the right half $w\#$ of the input $\#ww\#$.

Since x and y are M-equivalent, it follows that we can construct a computation $comp(xy)$, which never enters any accepting state, of M on xy such that:

(i) $cross(comp(xy)) = p_{01}^x p_{02}^x \cdots p_{0i_0}^x p_1 p_{11}^y p_{12}^y \cdots p_{1j_1}^y p_2 p_{21}^x p_{22}^x \cdots p_{2i_2}^x p_3 p_{31}^y p_{32}^y \cdots p_{3j_3}^y p_4 \cdots p_k \; p_{k1}^y p_{k2}^y \cdots p_{kj_k}^y$, and

(ii) $pebble\text{-}cross(comp(xy)) = p_1 p_2 \dots p_k$,
(iii) $comp(xy)[-, p^x_{01}] = recomp(xx)[-, p^x_{01}]$ ($comp(xy)[-, p_1] = recomp(xx)[-, p_1]$ if $i_0 = 0$), and
(iv) for each even number $l(0 \le l \le k-1)$,

- $comp(xy)[p_l, p^x_{l1}] = recomp(xx)[p_l, p^x_{l1}]$(where $l \ne 0$),
- for each even number $r(2 \le r \le i_l - 1)$, $comp(xy)[p^x_{lr}, p^x_{l,r+1}] = recomp(xx)[p^x_{lr}, p^x_{l,r+1}]$,
- $comp(xy)[p^x_{li_l}, p_{l+1}] = recomp(xx)[p^x_{li_l}, p_{l+1}]$, and

(v) for each odd number $l(1 \le l \le k)$,

- $comp(xy)[p_l, p^y_{l1}] = recomp(yy)[p_l, p^y_{l1}]$,
- for each even number $r(2 \le r \le j_l - 1)$, $comp(xy)[p^y_{lr}, p^y_{l,r+1}] = recomp(yy)[p^y_{lr}, p^y_{l,r+1}]$,
- $comp(xy)[p^y_{lj_l}, p_{l+1}] = recomp(yy)[p^y_{lj_l}, p_{l+1}]$ (where $l \ne k$).

Note that the M-equivalence of x and y implies the following:

- For each even number $l(0 \le l \le k-1)$ and each odd number $r(1 \le r \le i_l - 1)$, $comp(xy)[p^x_{lr}, p^x_{l,r+1}]$ can be constructed owing to the fact that x and y are M^--equivalent, and for each odd number $l(1 \le l \le k)$ and each odd number $r(1 \le r \le j_l - 1)$, $comp(xy)[p^y_{lr}, p^y_{l,r+1}]$ can also be constructed owing to the fact that x and y are M^--equivalent.
- If $recomp(yy)$ is a halting computation, then we can construct $comp(xy)[p^y_{kj_k}, -]$ ($comp(xy)[p_k, -]$ if $j_k = 0$) from $recomp(yy)[p^y_{kj_k}, -]$ ($recomp(yy)[p_k, -]$ if $j_k = 0$) so as for $comp(xy)$ to be a halting computation.
- Let $recomp(yy)$ be a double-looping computation, and $recomp(yy) = c_1 c_2 \dots c_m = c_1 c_2 \dots c_{i-1} c_i s c_i s c_i$, where $c_i s c_i s c_i$ is the double-looping segment of $recomp(yy)$.
  (i) If M crosses the boundary between the left half $\#y$ and the right half $y\#$ with a pair $p \in Q \times PT(m)$ from left to right (resp., from right to left) in the double-looping segment $c_i s c_i s c_i$, and enters a configuration c, then afterwards M again crosses the boundary with the same pair p from left to right (resp., from right to left) in $c_i s c_i s c_i$, and enters the same configuration c, because $c_i s c_i s c_i$ is the "double-looping" segment of $recomp(yy)$. Therefore, in this case, we can construct $comp(xy)[-, p^y_{kj_k}]$ ($comp(xy)[-, p_k]$ if $j_k = 0$) from $recomp(yy)[-, p^y_{kj_k}]$ ($recomp(yy)[-, p_k]$ if $j_k = 0$) and $recomp(xx)$ so as for it to have a loop.
  (ii) If M never crosses the boundary between $\#y$ and $y\#$ in the double-looping segment $c_i s c_i s c_i$ (that is, $recomp(yy)[p^y_{kj_k}, -]$ ($recomp(yy)[p_k, -]$ if $j_k = 0$) has a loop), then we can construct $comp(xy)[p^y_{kj_k}, -]$ ($comp(xy)[p_k, -]$ if $j_k = 0$) from $recomp(yy)$ $[p^y_{kj_k}, -]$ ($recomp(yy)[p_k, -]$ if $j_k = 0$) and $recomp(xx)$ so as for it to have a loop.

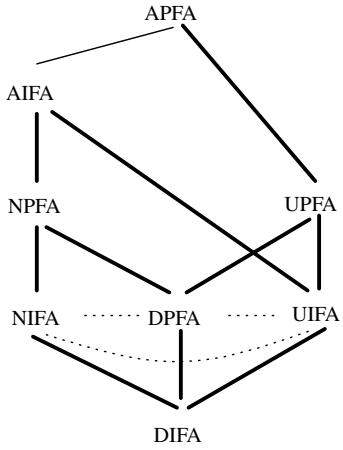


Fig. 2 Inclusion relationship among the classes of sets accepted by inkdot and one-pebble 2-fa's.

Clearly, this $comp(xy)$ forces the input xy to be rejected by M, which contradicts the fact that xy is in T_4. This completes the proof of (2). ∎

Unfortunately, it is unknown whether $UPFA \subsetneqq AIFA$. From Theorem 5.1, we get the following corollary:

Corollary 5.1. $UPFA \subsetneqq APFA$.

6. Conclusion

Figure 2 shows the inclusion relationship obtained in [3], [5] and in this paper. The bold lines indicate the proper inclusions, and the dotted lines indicate the incomparable relationships.

We conclude this paper by posing several open problems.

(1) $AIFA \subsetneqq APFA$?
(2) What are the relationships between $NIFA$ and $UPFA$ and between $UIFA$ and $NPFA$?
(3) Is $NPFA$ incomparable with $UPFA$?
(4) $UPFA \subsetneqq AIFA$?

Acknowledgement

We would like to thank the referees for carefully reading our paper, and for giving us useful comments.

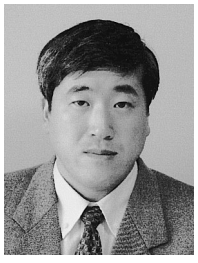
References

[1] M. Blum and C. Hewitt, "Automata on a 2-dimensional tape," IEEE Symp. on Switching and Automata Theory, pp.155–160, 1967.
[2] J.H. Chang, O.H. Ibarra, M.A. Palis, and B. Ravikumar, "On pebble automata," Theor. Comput. Sci., vol.44, pp.111–121, 1986.
[3] A. Inoue, K. Inoue, A. Ito, Y. Wang, and T. Okazaki, "A note on one-pebble two-dimensional turing machines," Inf. Sci., vol.162, pp.295–314, 2004.
[4] K. Inoue, I. Takanami, and H. Taniguchi, "Two-dimensional alternating Turing machines," Theor. Comput. Sci., vol.27, pp.61–83, 1983.
[5] A. Ito, K. Inoue, I. Takanami, and Y. Wang, "The effect of inkdots for two-dimensional automata," Int. J. Pattern Recognition and Artificial Intelligence, vol.9, no.5, pp.777–796, 1995.

[6] D. Ranjan, R. Chang, and J. Hartmanis, "Space bounded computations: Review and new separation results," Theor. Comput. Sci., vol.80, pp.289–302, 1991.
[7] A. Rosenfeld, Picture Languages, Academic Press, New York, 1979.
[8] A. Szepietowski, "Turing machines with sublogarithmic space," Lecture Notes in Computer Science 843, 1994.

Atsuyuki Inoue received his B.E. degree in Electronics and Computer Science from Science University of Tokyo in Yamaguchi in 1999, and M.E. degree in information Science from Japan Advanced Institute of Science and Technology in 2003. He is now a Ph.D. student of Yamaguchi University. His main interests are automata and computation theory.

Akira Ito received his B.E. and M.E. degrees in electronic engineering from Yamaguchi University in 1981 and 1983, respectively, and his Ph.D. degree in Engineering from Nagoya University in 1992. Since 1983, he has been with Yamaguchi University. He is presently an associate professor in the Faculty of Engineering. His current interests are automata theory and algorithm design especially for digital image processing.



Kunihiko Hiraishi received from the Tokyo Institute of Technology the B.E. degree in 1983, the M.E. degree in 1985, and D.E. degree in 1990. He is currently a professor at School of Information Science, Japan Advanced Institute of Science and Technology. His research interests include discrete event systems and formal modeling of concurrent systems. He is a member of the IEEE, IPSJ, and SICE.

Katsushi Inoue received his B.E. and M.E. degrees in electrical engineering from Hiroshima University in 1969 and 1971, respectively, and his Ph.D. degree in electrical engineering from Nagoya University in 1977. From 1973 to 1977, he was with Hiroshima University. Since 1977, he has been with Yamaguchi University. He is presently a Professor in the Faculty of Engineering. His main interests are automata theory and computation theory.