

# Sublinear Space-Bounded Multi-Inkdot Alternating Multi-Counter Automata with Only Universal States

Tsunehiro YOSHINAGA<sup>\*1</sup>, Jianliang XU<sup>\*2</sup> and Katushi INOUE<sup>\*3</sup>

## Abstract

This paper investigates a hierarchical property based on the number of inkdots in the accepting powers of sublinear space-bounded multi-inkdot two-way alternating multi-counter automata with only universal states. For each  $l \geq 1$ , each  $m \geq 0$ , and any function  $L(n)$ , let  $weak\text{-}2UCA^m(l, L(n))$  and  $strong\text{-}2UCA^m(l, L(n))$  denote the classes of sets accepted by weakly and strongly  $L(n)$  space-bounded  $m$ -inkdot two-way alternating  $l$ -counter automata with only universal states, respectively. We show that for any function  $L(n)$  such  $\log L(n) = o(\log n)$ ,  $strong\text{-}2UCA^{m+1}(1, \log n) \cup \bigcup_{1 \leq l < \infty} weak\text{-}2UCA^m(l, L(n)) \neq \emptyset$ . So, we have  $x\text{-}2UCA^m(l, L(n)) \subsetneq x\text{-}2UCA^{m+1}(l, L(n))$  for each  $l \geq 1$ , each  $x \in \{strong, weak\}$  and any function  $L(n) \geq \log n$  such that  $\log L(n) = o(\log n)$ .

**Key Words:** alternating multi-counter automata, multi-inkdot, universal states, sublinear space, computational complexity,

## 1. Introduction

A *multi-counter automaton* is a multi-pushdown automaton whose pushdown stores operate as counters, i.e., each storage tape is a pushdown tape of the form  $Z^i$  ( $Z$  is a fixed symbol). It is shown in Ref. 1) that 2-counter automata without time or space limitations have the same power as Turing machines; however, when time or space restrictions are applied, a different situation occurs (See, for example, Refs. 2), 3)).

In order to show a strong separation of deterministic and nondeterministic complexity classes, in Ref. 4), Ranjan et al. introduced a slightly modified Turing machine model, called an inkdot Turing machine. The inkdot Turing machine is a 2-way Turing machine with the additional power of marking at most 1 tape-cell in the input tape (with an inkdot) once. The action of the machine depends on the current states, the input and the work tape symbols scanned currently, and the presence of the inkdot on the currently scanned tape-cell. It is shown in Ref. 4) that for sublogarithmic space-bounded inkdot Turing machines, deterministic

and nondeterministic space complexity classes are not equal. Inoue et al. showed in Ref. 5), that there exists a set accepted by a strongly  $\log \log n$  space-bounded inkdot 2-way nondeterministic Turing machine, but not accepted by any weakly  $o(\log n)$  space-bounded 2-way nondeterministic Turing machines. From now on, logarithms are base 2.

After that, the multi-inkdot Turing machine was introduced in Ref. 6) as an extension of the inkdot Turing machine. An  $m$ -inkdot Turing machine,  $m \geq 1$ , is a 2-way Turing machine with  $m$  dots of ink. Thus, it can mark at  $m$  tape-cells on the input, once on each cell. Its action is similar to that of the inkdot Turing machine. In Ref. 6), it is shown that for nondeterministic sublogarithmic space complexity class,  $m+1$  inkdots are better than  $m$ .

Furthermore, in Ref. 7), Inoue et al. generalized the inkdot nondeterministic Turing machine to the inkdot alternating Turing machine, and showed that there is a set accepted by a strongly  $\log \log n$  space-bounded 2-way alternating inkdot Turing machine, but not accepted by any weakly  $o(\log n)$  space-bounded 2-way

---

<sup>\*1</sup> Department of Computer Science and Electrics Engineering

<sup>\*2</sup> Ocean University of China

<sup>\*3</sup> Yamaguchi University

alternating Turing machine. In Ref. 8), inkdot alternating multi-counter automata were also introduced, and it is proven that the class of sets accepted by  $L(n)$  space-bounded inkdot 2-way alternating multi-counter automata with only existential (universal) states is not closed under complementation, where  $L(n)$  is any function such that  $L(n) \geq \log n$  and  $\log L(n) = o(\log n)$ .

Of course, an alternating  $m$ -inkdot 2-way Turing machine is an alternating version of the  $m$ -inkdot Turing machine stated above, in the same sense as in Ref. 9). Xu et al. in Ref 10) introduced sublogarithmic space-bounded multi-inkdot 2-way alternating Turing machines and pushdown automata with constant leaf-size and showed that for the classes of sets accepted by these,  $m+1$  inkdots are better than  $m$ . In Ref 11), Miyamoto et al. showed the corresponding result to the above for multi-inkdot 2-way alternating multi-counter automata with constant leaf-size and sublinear space. In Ref. 12), Yoshinaga et al. got the pair to the result in Ref. 6). That is, they showed that sublogarithmic space-bounded alternating Turing machines with only universal states which have  $m+1$  inkdots are more powerful than those which have  $m$ .

From a theoretical point of view, in this paper, we are interested in knowing fundamental properties of multi-inkdot alternating multi-counter automata, and especially investigate a hierarchy in the accepting powers of the automata which have only universal states and sublinear space, in correspondence to the result in Ref. 12).

Section 2 gives some definitions and notations necessary for this paper. Section 3 investigates, for multi-inkdot 2-way alternating multi-counter automata with sublinear space and only universal states, how the number of inkdots affects the accepting powers of these automata. For each  $l \geq 1$ , each  $m \geq 0$ , and any function  $L(n)$ , let  $weak\text{-}2UCA^m(l, L(n))$  ( $strong\text{-}2UCA^m(l, L(n))$ ) denote the class of sets accepted by weakly (strongly)  $L(n)$  space-bounded  $m$ -inkdot 2-way alternating  $l$ -counter automata with only universal states. We show that for any function  $L(n)$  such that  $\log L(n) = o(\log n)$ ,  $strong\text{-}2UCA^{m+1}(1, \log n) \cup_{l \leq l \infty} weak\text{-}2UCA^m(l, L(n)) \neq \emptyset$ . Section 4 concludes this paper by giving a few open problems.

## 2. Preliminaries

A 2-way alternating multi-counter automaton (2amca)  $M$  is a generalization of a two-way nondeterministic multi-counter automaton. The state set of  $M$  is

partitioned into *universal* and *existential* states. Intuitively, in a universal state  $M$  splits into some submachines which act in parallel, and in an existential state  $M$  nondeterministically chooses one of possible subsequent actions.  $M$  has the left endmarker “ $\$$ ” and the right endmarker “ $\$$ ” on the input tape, reads the input tape right or left, and can enter an accepting state only when falling off  $\$$ . In one step  $M$  can also increment or decrement the contents (i.e., the length) of each counter by at most one.

For each  $l \geq 1$ , we denote a two-way alternating  $l$ -counter automaton by  $2aca(l)$ . An *instantaneous description* (ID) of  $2aca(l)$   $M$  is an element of

$$\Sigma^* \times \mathbb{N} \times S_M,$$

where  $\Sigma$  ( $\$, \epsilon \in \Sigma$ ) is the input alphabet of  $M$ ,  $\mathbb{N}$  denotes the set of all non-negative integers, and

$$S_M = Q \times (\{Z\}^*)^l$$

where  $Q$  is the set of states. The first and second components,  $w$  and  $i$ , of an ID

$$I = (w, i, (q, (\alpha_1, \alpha_2, \dots, \alpha_l)))$$

represent the input string and the input head position, respectively. The third component  $(q, (\alpha_1, \alpha_2, \dots, \alpha_l))$  of  $I$  represents the state of the finite control and the contents of the  $l$  counters.  $I$  is said to be a *universal* (*existential*, *accepting*) ID if  $q$  is a universal (an existential, an accepting) state. An element of  $S_M$  is called a *storage state* of  $M$ . The *initial* ID of  $M$  on  $w \in \Sigma^*$  is

$$I_M(w) = (w, 0, (q_0, \underbrace{(\lambda, \lambda, \dots, \lambda)}_l)),$$

where  $q_0$  is the initial state of  $M$  and  $\lambda$  denotes the empty string.

We write  $I \vdash_M I'$  and say  $I'$  is a *successor* of  $I$  if an ID  $I'$  follows from an ID  $I$  in one step, according to the transition function of  $M$ .

A *computation path* of  $M$  on input  $w$  is a sequence

$$I_0 \vdash_M I_1 \vdash_M \dots \vdash_M I_n \quad (n \geq 0),$$

where  $I_0 = I_M(w)$ .

A *computation tree* of  $M$  on input  $w$  is a finite, nonempty tree such that the root is labeled by the initial ID  $I_0$ , and the children of any non-leaf node  $\pi$  labeled by a universal (an existential) ID,  $\ell(\pi)$ , include all (one) of the immediate successors of  $\ell(\pi)$ .

A computation tree of  $M$  on input  $w$  is *accepting* if all the leaves are labeled by accepting ID's. We say that  $M$  *accepts*  $w$  if there is an accepting computation tree of  $M$  on  $w$ .

For any function  $L(n)$ ,  $M$  is *weakly* (*strongly*)  $L(n)$  *space-bounded* if for any  $n \geq 1$  and any input  $w$  of length  $n$  accepted by  $M$ , there is an accepting computation tree  $\tau$  of  $M$  on  $w$  such that for each node  $\pi$  of

$\tau$  (if for any  $n \geq 1$  and any input  $w$  of length  $n$  (accepted or not), and each node  $\tau$  of any computation tree of  $M$  on  $w$ ), the length of each counter in  $\ell(\tau)$  is bounded by  $L(n)$ .

We denote by  $2uca(l)$  ( $2nca(l)$ ) a  $2aca(l)$  with only universal states (existential states, i.e., a 2-way nondeterministic  $l$ -counter automaton). Further, we denote by  $2dca(l)$  a 2-way deterministic  $l$ -counter automaton.

For each  $y \in \{a, u, n, d\}$ , An  $m$ -inkdot  $2yca(l)$  ( $2yca^m(l)$ ),  $m \geq 0$ , is a  $2yca(l)$  with  $m$  dots of ink (Note that ‘ $2yca^0(l)$ ’ stands for the ‘ $2yca(l)$ ’). It can mark at most  $m$  tape-cells on the input (with its  $m$  inkdots), each of which is marked once with one inkdot and the inkdot is never erased. Its action depends on the current state, the currently scanned input symbol, the contents of the counters and the presence of inkdot on the currently scanned tape-cell. The action consists of entering a new state, moving the input head in specified directions, and making appropriate changes on the counters, in accordance with the transition relation. In addition, one of the unused inkdots may be used to mark the currently scanned cell on the input tape.

For each  $m \geq 0$ , each  $l \geq 1$ , and any function  $L(n)$ , let denote by *weak* (*strong*)- $2ACA^m(l, L(n))$ , *weak* (*strong*)- $2UCA^m(l, L(n))$ , *weak* (*strong*)- $2NCA^m(l, L(n))$ , and *weak* (*strong*)- $2DCA^m(l, L(n))$  the classes of sets accepted by strongly (weakly)  $L(n)$  space-bounded  $2aca^m(l)$ ,  $2uca^m(l)$ ,  $2nca^m(l)$ , and  $2dca^m(l)$ , respectively.

A 2-way alternating Turing machine (2ATm) we consider in this paper has a read-only input tape and a separate storage tape. We denote a 2ATm with only universal and existential states by  $2uTm$  and  $2nTm$ , respectively, and also denote a 2-way deterministic Turing machine by  $2dTm$ . For each  $m \geq 0$  and any function  $L(n)$ , let denote the classes of sets accepted by strongly (weakly)  $L(n)$  space-bounded  $m$ -inkdot  $2aTm$ ,  $2uTm$ ,  $2nTm$  and  $2dTm$  by *weak* (*strong*)- $2ATM^m(L(n))$ , *weak* (*strong*)- $2UTM^m(L(n))$ , *weak* (*strong*)- $2NTM^m(L(n))$ , and *weak* (*strong*)- $2DTM^m(L(n))$ , respectively.

### 3. Result

It is shown in Refs. (6, (10 that for each  $m \geq 1$ ,  
*strong*- $NTM^{m+1}(\log \log n)$   
 $-weak-2NTM^m(o(\log n)) \neq \phi$ .

In correspondence to the result above, it is implicitly shown in Ref. (11 that for each  $m \geq 1$  and any function  $L(n)$  such that  $\log L(n) = o(\log n)$ ,

$$strong-2NCA^{m+1}(1, \log n)$$

$$- \cup_{1 \leq l < \infty} weak-2NCA^m(l, L(n)) \neq \phi.$$

It is shown in Ref. (12 that for each  $m \geq 1$ ,

$$strong-UXTM^{m+1}(\log \log n)$$

$$- weak-2UTM^m(o(\log n)) \neq \phi.$$

Our corresponding result is:

**Theorem 3.1:** For each  $m \geq 0$  and any function  $L(n)$  such that  $\log L(n) = o(\log n)$ ,

$$strong-2UCA^{m+1}(1, \log n)$$

$$- \cup_{1 \leq l < \infty} weak-2UCA^m(l, L(n)) \neq \phi.$$

**Proof:** For each  $m \geq 1$ , let

$$T(m) = \{B(1)\#B(2)\#\dots\#B(n)cw_{11}cw_{12}c\dots$$

$$cw_{1r_1}ccw_{21}cw_{22}c\dots cw_{2r_2}cc\dots$$

$$ccw_{m1}cw_{m2}\dots c.. cw_{mr_m}cu \in \{0, 1, c, \#\}^+ |$$

$$n \geq 2 \& u \in \{0, 1\}^{\lceil \log n \rceil} \&$$

$$\forall s (1 \leq s \leq m) [r_s \geq 1 \& \forall t (1 \leq t \leq r_s)$$

$$[w_{st} \in \{0, 1\}^{\lceil \log n \rceil}]] \& \exists i (1 \leq i \leq m)$$

$$[\forall j (1 \leq j \leq r_i) [u \neq w_{ij}]]\},$$

where for each integer  $k \geq 1$ ,  $B(k)$  denotes the string in  $\{0, 1\}^+$  that represents the integer  $k$  in binary notation (with no leading zeros).

[1] We first show that

$$"T(m+1) \in strong-2UCA^{m+1}(1, \log n)".$$

One can construct a strongly  $\log n$  space-bounded  $2uca^{m+1}(1)M$  which acts as follows.

Suppose that an input string:

$$\phi y_1 \# y_2 \# \dots \# y_n cw_{11} cw_{12} c.. cw_{1r_1} ccw_{21} cw_{22} c\dots$$

$$cw_{2r_2} cc\dots ccw_{m+1,1} cw_{m+1,2} c\dots cw_{m+1,r_{m+1}} cu\$$$

(where  $n \geq 2$ ,  $r_i \geq 1$ , and  $y_k$ 's,  $w_{ij}$ 's and  $u$  are all in  $\{0, 1\}^+$ ) is presented to  $M$  (Input strings in the form different from the above can easily be rejected by  $M$ ). It is shown in Ref. (3) that the set  $\{B(1)\#B(2)\#\dots\#B(n) | n \geq 2\}$  can be accepted by a strongly  $\log n$  space-bounded  $2dca(1)$ . So,  $M$  can store  $\lceil \log n \rceil$  stack symbols in the counter using the initial segment  $B(1)\#B(2)\#\dots\#B(n)$  of the input (Of course,  $M$  never enters an accepting state if  $y_k \neq B(k)$  for some  $1 \leq k \leq n$ ).

If  $M$  successfully complete this, then it checks, by using  $\lceil \log n \rceil$  stack symbols stored in the counter, whether  $|w_{ij}| = |u| = \lceil \log n \rceil$ .

After that, in the first block:

$$cw_{11} cw_{12} c.. cw_{1r_1}$$

$M$  universally branches and marks the symbol ‘ $c$ ’ just before  $w_{1j}$  by the (first) inkdot in order to check whether  $u \neq w_{1j}$  for each  $1 \leq j \leq r_1$ . That is,  $M$  can check by using  $\lceil \log n \rceil$  stack symbols stored in the counter and the inkdot as a pilot if  $u \neq w_{1j}$  while moving its input head back and forth. If  $M$  verifies that  $u \neq w_{1j}$  then  $M$  goes to the right endmarker  $\$$  and en-

ters an accepting state. Otherwise,  $M$  moves to the second block and universally checks whether  $u \neq w_{2j}$  for each  $1 \leq j \leq r_2$  in the same manner as the first block above, using the (second) inkdot. This action is continued until for some  $1 \leq i \leq m+1$ , in the  $i$ -th block,  $u \neq w_{ij}$  is verified for each  $1 \leq j \leq r_i$  (that is, this input is accepted).

If all checks throughout  $m+1$  blocks are unsuccessful, then  $M$  which has no its inkdots any more never enters an accepting state. It will be obvious that  $\lceil \log n \rceil$  space and  $m+1$  inkdots are sufficient, and  $M$  accepts the set  $T(m+1)$ .

[2] We then show that

$$"T(m+1) \in \cup_{1 \leq l < \infty} \text{weak-2UCA}^m(l, L(n)),"$$

where  $L(n)$  is a function such that  $\log L(n) = o(\log n)$ . Suppose to the contrary that there exists a weakly  $L(n)$  space-bounded  $2\text{uca}^m(l)$   $M$  accepting  $T(m+1)$ , where  $l \geq 1$  is some constant. For each  $n \geq 2$ , let

$$\begin{aligned} V(n) &= \{B(1)\#B(2)\#\dots\#B(n)y_1y_2\dots y_{m+1}u \\ &\in \{0,1,c,\#\}^+ \mid \forall i(1 \leq i \leq m+1) \\ &[y_i \in W(n)] \& u \in \{0,1\}^{\lceil \log n \rceil}\}, \end{aligned}$$

where

$$\begin{aligned} W(n) &= \{cw_1cw_2c\dots cw_nc \in \{0,1,c\}^+ \\ &\forall j(1 \leq j \leq n)[w_j \in \{0,1\}^{\lceil \log n \rceil}]\}. \end{aligned}$$

We consider the computations of  $M$  on the strings in  $V(n)$ . Let  $l(n)$  be the length of each element in  $V(n)$ . Then,  $l(n) = O(n \log n)$ . Let  $C(n)$  denote the set of all possible storage states of  $M$  when  $M$  in the computation uses at most  $L(l(n))$  stack symbols in each counter, and let  $u(n)$  be the number of elements of  $C(n)$ . Then,  $u(n) = O(L(l(n)))$ .

For any two strings  $x$  and  $y$  in  $W(n)$ , we say that  $x$  and  $y$  are  $M$ -equivalent, if for each pair of storage states  $q, q' \in C(n)$ , each integer  $1 \leq i \leq m$ , and  $d, d' \in \{\text{right, left}\}$ ,

- (i) there exists an  $L(l(n))$  space-bounded computation in which  $M$  enters  $x$  in  $q$  with  $i$  inkdots (resp., without inkdots) from the  $d$  end, and afterwards exits  $x$  in  $q'$  from the  $d'$  end without consuming the inkdots on the way,  
 $\Leftrightarrow$   
 there exists an  $L(l(n))$  space-bounded computation in which  $M$  enters  $y$  in  $q$  with  $i$  inkdots (resp., without inkdots) from the  $d$  end, and afterwards exits  $y$  in  $q'$  from the  $d'$  end without consuming the inkdots on the way,
- (ii) there exists a computation in which  $M$  enters  $x$  in  $q$  with  $i$  inkdots (resp., without inkdots) from the  $d$  end, and afterwards exits  $x$  from the  $d'$  end using some counters of length larger than  $L(l(n))$  with-

out consuming the inkdots on the way,

$\Leftrightarrow$

there exists a computation in which  $M$  enters  $y$  in  $q$  with  $i$  inkdots (resp., without inkdots) from the  $d$  end, and afterwards exits  $y$  from the  $d'$  end using some counters of length larger than  $L(l(n))$  without consuming the inkdots on the way, or

- (iii) there exists a computation in which  $M$  enters  $x$  in  $q$  with  $i$  inkdots from the  $d$  end, and never exits  $x$ ,  
 $\Leftrightarrow$

there exists a computation in which  $M$  enters  $y$  in  $q$  with  $i$  inkdots from the  $d$  end, and never exits  $y$ .

Clearly  $M$ -equivalence is an equivalence relation on  $W(n)$ , and there are at most  $e(n) = O(t^{u(n) \times u(n)})$   $M$ -equivalent classes, where  $t$  is a constant. For each

$$y = cw_1cw_2c\dots cw_n$$

in  $W(n)$ , let  $b(y) = \{w \mid \exists i(1 \leq i \leq n)[w = w_i]\}$ .

Furthermore, for each  $n \geq 2$ , let  $R(n) = \{b(y) \mid y \in W(n)\}$ . Then,

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

(Intuitively,  $|R(n)|$  is equal to the number of all the nonempty subsets of  $\{0, 1\}^{\lceil \log n \rceil}$ .)

From the assumption that  $L(n) = o(\log n)$  and from the fact that  $l(n) = O(n \log n)$ , it follows that  $|R(n)| > e(n)$  for  $n$  large enough, and so there must exist two  $M$ -equivalent elements  $y$  and  $y'$  in  $W(n)$  such that  $b(y) \neq b(y')$ . We can, without loss of generality, assume that there is a string

$$u \in \{0, 1\}^{\lceil \log n \rceil} \text{ such that } u \in b(y) - b(y').$$

Consider the following string  $z$ :

$$z = B(1)\#B(2)\#\dots\#B(n)y_1y_2\dots y_{m+1}u,$$

where  $y_1 = y_2 = \dots = y_{m+1} = y$ . Since  $y (= cw_1cw_2c\dots cw_n)$  contains some segment  $w_i$  ( $1 \leq i \leq n$ ) which equals  $u$ ,  $z$  is not in  $T(m+1)$ . So  $z$  is never accepted by  $M$ , that is, there is at least one computation path  $I_0 \vdash_M I_1 \vdash_M \dots \vdash_M I_q$  ( $q \geq 1$ ) of  $M$  on  $z$  with the following properties (P<sub>1</sub>), (P<sub>2</sub>) and (P<sub>3</sub>):

- (P<sub>1</sub>)  $I_i$  is not an accepting ID and the length of each counter in  $I_i$  is bounded by  $L(|z|)$  for each  $i$  ( $0 \leq i \leq m$ );
- (P<sub>2</sub>)  $I_i \neq I_j$  for each  $i, j$  ( $0 \leq i < j < q$ );
- (P<sub>3</sub>)  $I_q$  is such that
  - (i)  $I_q$  is a non-accepting halting ID,
  - (ii)  $I_q = I_i$  for some  $i$  ( $0 \leq i < q$ ), or
  - (iii) the length of some counter is larger than  $L(|z|)$ .

Fix such a computation path of  $M$  on  $z$  with the properties above, and denote it by  $\text{comp}(z)$ . Since  $M$  has at most  $m$  inkdots, it follows that there is some string  $y_i$  ( $1$

$\leq i \leq m+1$ ) such that  $M$  never consumes any of its  $m$  inkdots on  $y_i$  in  $\text{comp}(z)$ . Let  $z'$  be the string obtained from  $z$  by replacing  $y_i$  ( $=y$ ) by  $y'$ . From  $\text{comp}(z)$  and the fact that  $y$  and  $y'$  are  $M$ -equivalent, we can easily construct a computation path of  $M$  on  $z'$  in which  $M$  never reaches an accepting state or  $M$  enters an accepting state only after using more than  $L(l(n))$  stack symbols in some counters. Thus,  $z'$  is never accepted by  $M$ . This is a contradiction, because  $y'$  never contains the segment  $u$ , and so  $z'$  is in  $T(m+1)$ .  $\square$

From Theorem 3.1, we have:

**Corollary 3.2:** For each  $x \in \{\text{strong}, \text{weak}\}$ , each  $l \geq 1$ , each  $m \geq 0$ , and any function  $L(n)$  such that  $L(n) \geq \log n$  and  $\log L(n) = o(\log n)$ ,

$$x\text{-2UCA}^m(l, L(n)) \subsetneq x\text{-2UCA}^{m+1}(l, L(n)).$$

#### 4. Conclusion

We have investigated the accepting power of sublinear space-bounded multi-inkdot 2-way amca's with only universal states. Our main result is that for each  $m \geq 0$  and any function  $L(n)$  such that  $\log L(n) = o(\log n)$ ,

$$\text{strong-2UCA}^{m+1}(1, \log n)$$

$$- \cup_{1 \leq l < \infty} \text{weak-2UCA}^m(l, L(n)) \neq \phi.$$

It is shown that for each  $m \geq 1$ , each  $x \in \{\text{weak}, \text{strong}\}$ , and any function  $\log \log n \leq L(n) = o(\log n)$ ,

$$x\text{-2DTM}^m(L(n)) = x\text{-2DTM}^0(L(n)) \text{ and}$$

$$\text{strong-2ATM}^1(\log \log n)$$

$$- \text{weak-2ATM}^0(o(\log n)) \neq \phi,$$

in Refs. 4) and 7) respectively. But unfortunately, it is unknown whether for each  $m \geq 1$ ,

$$\text{strong-2ATM}^{m+1}(\log \log n)$$

$$- \text{weak-2ATM}^m(o(\log n)) \neq \phi.$$

Finally, we conclude this paper by giving two open problems relating this research:

(1) For each  $m \geq 1$ , each  $l \geq 1$ , each  $x \in \{\text{weak}, \text{strong}\}$ , and any function  $\log n \leq L(n)$  such that  $\log L(n) = o(\log n)$ ,

$$\cdot x\text{-2DCA}^m(l, L(n)) = x\text{-2DCA}^0(l, L(n))?$$

$$\cdot \cup_{1 \leq l < \infty} x\text{-2DCA}^m(l, L(n))$$

$$= \cup_{1 \leq l < \infty} x\text{-2DCA}^0(l, L(n))?$$

(2) For each  $m \geq 0$  and any function  $L(n)$  such that  $\log L(n) = o(\log n)$ ,

$$\text{strong-2ACA}^{m+1}(1, \log n)$$

$$- \cup_{1 \leq l < \infty} \text{weak-2ACA}^m(l, L(n)) \neq \phi?$$

#### References

- 1) Minsky, M.L.: Recursive unsolvability of Post's problem of 'Tag' and other topics in the theory of Turing Machines, *Annals of Math.*, Vol.74, No.3, pp.437-455 (1961).
- 2) Inoue, K., Ito, A. and Takanami, I.: A note on real-time one-way alternating multicounter machines, *Theoret. Comput. Sci.*, Vol.88, pp.287-296 (1991).
- 3) Yoshinaga, T. and Inoue, K.: A note on alternating multi-counter automata with small space, *Trans. of IPSJ.*, Vol.36, No.12, pp.2741-2753 (1995).
- 4) Ranjan, D., Chang, R. and Hartmans, J.: Space bounded computations: Review and new separation results, *Theoret. Comput. Sci.*, Vol.80, pp.289-302 (1991).
- 5) Inoue, K., Ito, A. and Takanami, I.: A relationship between nondeterministic Turing machines and 1-inkdot Turing machines with small space, *Inform. Process. Lett.*, Vol.43, pp.225-227 (1992).
- 6) Inoue, K., Ito, A., Takanami, I. and Yoshinaga, T.: A note on multi-inkdot nondeterministic Turing machines with small space, *Inform. Process. Lett.*, Vol.48, pp.285-288 (1993).
- 7) Inoue, K., Ito, A. and Takanami, I.: On 1-inkdot alternating Turing machines with small space, *Theoret. Comput. Sci.*, Vol.127, pp.171-179 (1994).
- 8) Yoshinaga, T. and Inoue, K.: Some observations on 1-inkdot alternating multi-counter automata with sublinear space, *IEICE Trans., INF. & SYST.*, Vol.E83-D, No.2, pp.285-290 (2000).
- 9) Chandra, A.K., Kozen D.C. and Stockmeyer, L.J.: Alternation, *J. ACM*, Vol.28, No.1, pp.114-133 (1981).
- 10) Xu, J., Inoue, K., Wang, Y. and Ito, A.: On multi-inkdot two-way alternating Turing machines and pushdown automata with sublogarithmic space and constant leaf-size, *IEICE Trans., INF. & SYST.*, Vol.E80-D, No.1, pp.86-90 (1997).
- 11) Miyamoto, S., Yoshinaga, T., and Inoue, K.: Multi-inkdot alternating multi-counter automata with sublinear space and constant leaf-size, *Research Reports of the Tokuyama College of Technology*, No.25, pp.19-24 (2001).
- 12) Yoshinaga, T. and Inoue, K.: Sublogarithmic space-bounded multi-inkdot two-way alternating Turing machines with only universal states, *IEICE Trans., INF. & SYST.*, Vol.E84-D, No.1, pp.61-64 (2001).

(Received September 11, 2006)