Sublinear Space-Bounded One-Way Self-Verifying Nondeterministic Turing Machines

Tsunehiro YOSHINAGA[*1], Jianliang XU[*2] and Katushi INOUE[*3]

Abstract

We investigate the accepting power of one-way self-verifying nondeterministic Turing machines with sublinear space. A self-verifying nondeterministic Turing machine has four types of states: working, accepting, rejecting, and neutral ("I don't know") ones. There is no possible move from any rejecting, accepting and neutral states. The machine is not allowed to make mistakes. We show that there exists a language accepted by a log n space-bounded one-way self-verifying nondeterministic Turing machine, but not accepted by any $o(n)$ space-bounded deterministic Turing machine, and also show that there exists a language accepted by a log n space-bounded one-way nondeterministic Turing machine, but not accepted by any $o(n)$ space-bounded self-verifying nondeterministic Turing machine.

Key Words: self-verifying nondeterminism, nondeterminism, determinism, Turing machines, one-way computation, sublinear space complexity

1. Introduction and Preliminaries

The comparative study of the computational powers of nondeterministic and deterministic computations is one of the central tasks of complexity theory. Some of recent researches have focused on self-verifying nondeterminism for restricted computational models. For example, several properties of self-verifying nondeterministic multi-head and multi-counter finite automata have been explicated in Refs (1 — (4. But there are few investigations about self-verifying nondeterminism for more general computational models, such as Turing machines and pushdown automata, as far as we know.

On the other hand, Inoue et al.[5] and Xu et al.[6] have investigated the accepting powers of one-way alternating Turing machines and pushdown automata with sublinear space, respectively.

In this paper, from our theoretical interests, we will investigate a few fundamental properties of one-way self-verifying nondeterministic Turing machines with sublinear space. We show that (1) there exists a language accepted by a log n space-bounded self-verifying nondeterministic Turing machine, but not accepted by any $o(n)$ space-bounded deterministic one, and (2) there exists a language accepted by a log n space-bounded nondeterministic Turing machine, but not accepted by any $o(n)$ space-bounded self-verifying nondeterministic one. Throughout this paper, we assume that all logarithms are base 2.

We assume that our Turing machine M has a read-only input tape delimited by the left end-marker "¢" and the right end-marker "$", and a semi-infinite read-write work tape. We also assume, without loss of generality, that M can enter an accepting state only when falling off $.

An *instantaneous description* (ID) of M is of the form $(w, i, (q, \alpha, j))$. The first and second components w and i represent the input string and the input head position, respectively. From now on, we note that $0 \leqq i \leqq |w|+2$, where for any string v, $|v|$ denotes the length of v. "0", "1", "$|w|+1$" and "$|w|+2$" represent the positions of the left end-marker ¢, the leftmost symbol of w, the right end-marker $, and the immediate right to $, respectively. The third component (q, α, j) is a *storage state* which represents a combination of the

[*1] Department of Computer Science and Electrics Engineering

[*2] Ocean University of China

[*3] Yamaguchi University

state of the finite control, non-blank contents of the work-tape, and the work-tape head position.

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 of M, according to its transition rules. The reflexive transitive closure of $\vdash$ is denoted by $\vdash^*$.

A *computation path* of M on input w is a sequence $I_0 \vdash_M I_1 \vdash_M \cdots \vdash_M I_n$ $(n > 0)$, where $I_0 = (w, 0, (q_0, \lambda, 1))$ is the *initial* ID, where q_0 is the initial state and λ denotes the empty string.

An *accepting computation path of M on input w* is a computation path of M on w which ends with an ID I_n, $n \geqq 0$, relating to an accepting state. We say that M accepts w if there is an accepting computation path of M on input w. We denote by $L(M)$ the set of all inputs accepted by M.

Let $S(n)$ be a function. A computation path of M (on some input) is $S(n)$ *space-bounded* if all ID's of the path use at most $S(n)$ work tape-cells. M is $S(n)$ *space-bounded* if for any input w of length n, $n \geqq 1$, any computation path of M on w is $S(n)$ space-bounded.

We denote by 1nTm's and 1dTm's one-way nondeterministic and deterministic Turing machines, respectively. The states of these Turing machines are considered to be divided into three disjoint sets of working, accepting and rejecting states. No action is impossible from any rejecting or accepting states.

On the other hand, a one-way *self-verifying nondeterministic* Turing machine, denoted by 1svnTm, M has four types of states: working, accepting, rejecting, and neutral ("I don't know") ones. There is no possible move from rejecting, accepting and neutral states. M is not allowed to make mistakes. If there is a computation of M on an input w finishing in an accepting (resp., a rejecting) state, then w must be (resp., not be) in $L(M)$. For every input w, there is at least one computation of M that finishes either in an accepting state (if $w \in L(M)$) or in a rejecting state (if $w \notin L(M)$).

For each $x \in \{\text{n, d, svn}\}$, we denote by 1xTm($S(n)$) a $S(n)$ space-bounded 1xTm. Furthermore, for each $X \in \{\text{N, D, SVN}\}$, 1XTM($S(n)$) denotes the class of sets accepted by the corresponding Turing machines.

2. Results

We first show the following result:

Theorem 2.1:

$$1\text{SVNTM}(\log n) - 1\text{DTM}(o(n)) = \phi.$$

Proof: Let

$$L_1 = \{w \in \{0, 1\}^+ \mid (|w| \text{ is odd}) \ \& \ (\text{the center symbol in } w \text{ is '1'})\}.$$

Then, it is sufficient to show that
(1) $L_1 \in$ 1SVNTM($\log n$) and
(2) $L_1 \notin$ 1DTM($o(n)$).

Proof of (1): The language L_1 is accepted by a 1svnTm($\log n$) M operating as follows. Let H be the input head of M. M has two tracks t_1, t_2 on its work tape.

Suppose that an input $w \in \{0, 1\}^+$ of odd length is presented to M (Input strings of even length can be easily rejected by M).

First, M moves its input head H right while maintaining in one track t_1 on the work-tape the distance dl of H from the left end-marker ¢ in binary, until M nondeterministically guesses that H reaches the center symbol s_c of w.

Then, M stores s_c read by H in its finite control.(Note that at this time, dl is the distance of s_c from ¢).

After that, M moves H right while measuring and storing in another track t_2 on the work-tape the distance dr of H from the position of s_c, until H reaches the right end-marker $.

Finally, M enters
(i) an accepting state if $dl = dr$ and s_c is '1',
(ii) a rejecting state if $dl = dr$ and s_c is '0', and
(iii) a neutral state if $dl \neq dr$.

It is obvious that M can do the actions above operating in one-way and using at most $\log n$ work-tape cells.

Proof of (2): Suppose that there is a 1dTm($S(n)$) M which accepts L_1, where $S(n) = o(n)$. For each $n \geqq 1$, let

$$V(n) = \{0^n w 0^p \mid w \in \{0, 1\}^n \ \& \ 1 \leqq p \leqq 2n-1\}.$$

We consider the computation of M on the string in $V(n)$. For each $x = 0^n w 0^p$ in $V(n)$, let $s(x)$ be the storage state of M just after the input head H of M reads through the initial segment $0^n w$ of length $2n$. Then, the following proposition must hold:

Proposition 2.2: For any two different strings x and x' in $V(n)$ whose initial segments of length $2n$ are different, $s(x) \neq s(x')$.

[Proof: Suppose to the contrary that
(i) $x = 0^n w 0^p = 0^n w_1 1 w_2 0^p$ and

$x' = 0^n w' 0^p = 0^n w_1' 0 w_2' 0^p$
in $V(n)$, where $|w_1| = |w_1'| = l$ $(0 \leqq l \leqq n-1)$ and $p = n + l - |w_2|$, and
(ii) $s(x) = s(x')$.
Clearly, $x \in L$ (note that $|0^n w_1| = |w_2 0^p| = n + l$), and so x is accepted by M. It follows that x' must be also accepted by M, since $s(x) = s(x')$ and the suffix 0^p of x' is the same as the one of x. But x' is not in L_1. The proposition follows from this contradiction. □]

Proof (continued)**:** Let $l(n)$ be the length of each element in $V(n)$. Then, $l(n) = O(n)$. Again, let $C(n)$ denote the set of all possible storage states of M when M in the computation uses at most $S(l(n))$ work-tape cells., and let $e(n)$ be the number of elements of $C(n)$. Then, $e(n) \leqq tS(l(n))k^{S(l(n))}$, where t and k are the numbers of states of the finite control and work-tape symbols of M. On the other hand, let $u(n)$ be the number of strings in $V(n)$ whose initial segments of length $2n$ are different. Clearly, $u(n) = 2^n$. Since $S(n) = o(n)$ and $l(n) = O(n)$, $u(n) > e(n)$ for large n, and so it follows that for such n, there must be two different strings x and x' in $V(n)$ such that
(i) the initial segments of length $2n$ of x and x' are different, and
(ii) $s(x) = s(x')$.
This contradicts Proposition 2.2. This completes the proof of (2). □

We also show the following theorem:

Theorem 2.3:

$$1\mathrm{NTM}(\log n) - 1\mathrm{SVNTM}(o(n)) = \phi .$$

Proof: Let

$$L_2 = \{ w2w' \mid w, w' \in \{0, 1\}^+ \& w \neq w' \}.$$

Then, we will show that
(1) $L_2 \in 1\mathrm{NTM}(\log n)$ and
(2) $L_2 \notin 1\mathrm{SVNTM}(o(n))$.
Proof of (1): The language L_2 is accepted by a 1nTm(log n) M which acts as follows. Suppose that an input $w2w'$, where $w, w' \in \{0, 1\}^+$, is presented to M (Inputs of the form different from the above can be easily rejected by M). M nondeterministically checks that $w(j) \neq w'(j)$ for some j, where for any strung v, $v(i)$ denotes the i-th symbol (from the left) of v. That is, M guesses some j $(1 \leqq j \leqq |w|)$, stores j in its work tape in binary, when it picks up the symbol $w(j)$, and compares the symbol $w(j)$ with the symbol $w'(j)$ by using j stored (in binary) in the work tape. M enters an accepting state only if $w(j) \neq w'(j)$. It is easily observed that log n space is sufficient.
Proof of (2): Suppose to the contrary that there is a 1svnTm($S(n)$) M which accepts L_1, where $S(n) = o(n)$. For each $n \geqq 1$, let

$$V(n) = \{w2w \mid w \in \{0, 1\}^+ \& |w| = n \}.$$

For each $x = w2w$ in $V(n)$, there is at least one computation path of M on x in which M enters a rejecting state, because x is not in L_2. Fix such a computation path of M on x, and denote it by $c(x)$. Let $s(x)$ be the storage state of M just after the point where in $c(x)$ the input head has left the symbol "2" of x. Then, the following proposition must hold:

Proposition 2.4: For any two different strings x, y in $V(n)$, $s(x) \neq s(y)$.

[proof: Suppose to the contrary that $x = w2w$, $y = w'2w'$, $w \neq w'$, and $s(x) = s(y)$. Let $z = w2w'$. Then, there is a computation path $I_0 \vdash_M I_1 \vdash_M \cdots \vdash_M (z, |w2|, s(x))$ of M on z. When, starting with ID $(z, |w2|, s(x))$, M proceeds to read the segment w' of z, there exists a sequence of steps of M in which M enters a rejecting state, since $s(x) = s(y)$. This means that z is rejected by M. This contradicts the fact that z is in $L_2 = L(M)$. □]

Proof (continued)**:** Let $C(n) = \{s(x) \mid x \in V(n)\}$. Clearly, $|V(n)| = 2^n$, where for any set T, $|T|$ denotes the number of elements of T. $|C(n)| = O(p^{S(2n+1)})$, where p is some constant. Since $S(n) = o(n)$, we have $|V(n)| > |C(n)|$ for large n, and hence it follows that for such n, there must be two different strings x, y in $V(n)$ such that $s(x) = s(y)$. This contradicts Proposition 2.4. This completes the proof of (2). □

From Theorems 2.1 and 2.3, we can show:

Corollary 2.5: For any function $S(n)$ such that log $n \leqq S(n) = o(n)$,

$$1\mathrm{DTM}(S(n)) \subsetneq 1\mathrm{SVNTM}(S(n)) \subsetneq 1\mathrm{NTM}(S(n)).$$

3. Conclusion

We have investigated the accepting power of sublinear space-bounded 1svnTm's. Our main result is that

$$1NTM(\log n) - 1SVNTM(o(n)) = \phi, \text{ and}$$
$$1SVNTM(\log n) - 1DTM(o(n)) = \phi.$$

Let $1ATM(S(n))$ be the class of sets accepted by $S(n)$ space-bounded one-way alternating Turing machines. Inoue et al.5) implicitly showed that

$$1ATM(\log n) - 1NTM(o(n)) = \phi.$$

From this result and Corollary 2.5, it follows that for any function $S(n)$ such that $\log n \leqq S(n) = o(n)$,

$$1DTM(S(n)) \subsetneq 1SVNTM(S(n))$$
$$\subsetneq 1NTM(S(n)) \subsetneq 1ATM(S(n)).$$

It is obvious, from the definition of self-verifying nondeterminism, that the class of sets accepted by any self-verifying nondeterministic machines is closed under complementation. But the following problem remains open:

- is $1SVNTM(S(n))$ is closed under intersection, union, length-preserving homomorphism, concatenation with regular set, and Kleene closure for any function $S(n)$ such that $\log n \leqq S(n) = o(n)$?

References

1) Inoue, K., Tanaka, Y., Ito, A., and Wang, Y.: Self-verifying nondeterministic and Las Vegas multihead finite automata, IEICE Trans. Fundamentals, Vol.E84-A , No.5, PP.1094—1101 (2001).

2) Inoue, S., Inoue, K., Ito, A., and Wang, Y.: Self-verifying nondeterministic and Las Vegas multihead two dimensional finite automata, Information, Vol.5, No.1, PP.103—115 (2002).

3) Yoshinaga, T., and Inoue, K.: Las Vegas, self-verifying nondeterministic and deterministic one-way multi-counter automata with bounded time, IEICE Trans. Fundamentals, Vol.E86-A, No.5, PP. 1207—1212 (2003).

4) Yoshinaga, T., and Inoue, K.: Realtime one-way self-verifying nondeterministic multi-counter automata, Proc. of the 53rd Chugoku-shibu Joint Conference of Institutions Associated with Electricity and Information, PP.352—353 (2002).

5) Inoue, K., Takanami, I., Taniguchi, H., and Ito, A.: A note on alternating on-line Turing machines with only universal states, IEICE Trans., Vol.E66, No.6, PP.395—396 (1983).

6) Xu, J., Yoshinaga, T., and Inoue, K.: Some observations on one-way alternating pushdown automata with sublinear space, IEICE Trans. Fundamentals, Vol.E87-A, No.5, PP. 1012—1019 (2004).

7) Chandra, A.K., Kozen, D.C., and Stockmeyer, L.J.: Alternation, J. ACM, Vol.28, No.1, PP.114—133 (1981).

8) Szepietowski, A.: Turing machines with sublogarithmic space, LNCS 843 (Springer-Verlag, 1994).

(Received September 2, 2005)