LETTER

A Note on Synchronized Alternating Turing Machines with Small Space Bounds

Katsushi INOUE†, Itsuo TAKANAMI†, Akira ITO†† *and* Hiroshi MATSUNO†††, *Members*

SUMMARY We solve open problems about synchronized alternating Turing machines. For example, we show that for any $L(n)=o(\log n)$, there is a set accepted by a loglog n space bounded two-way synchronized alternating Turing machine with only universal states, but not accepted by any $L(n)$ space bounded one-way synchronized Turing machine with only universal states.

1. Introduction and Preliminaries

In Ref. (1), synchronized alternating devices were introduced as a generalization of alternating devices. The synchronization enables the communication among parallel processes in alternating computations. In Refs. (1) and (2), many properties of synchronized alternating devices were given. In this paper, we solve several problems left open in Ref. (2).

We refer to Refs. (3)-(5) for a more formal introduction of an 'alternating Turing machine' (ATM). An ATM M has a read-only input tape with the left and right endmarkers ¢ and \$, and one semi-infinite storage tape, initially blank. A step of M consists of reading one symbol from each tape, writing a symbol on the storage tape, moving the input and storage tape heads in specified directions, and entering a new state, in accordance with the next move relation. The state set is partitioned into accepting, rejecting, existential and universal states.

A 'synchronized alternating Turing machine' (SATM) M is an ATM some states of which have a sync element from some given finite set. These states and the instantaneous descriptions (see below) associated with them are called sync states and sync instantaneous descriptions, respectively. When a process P enters a sync state, it stops and waits until all parallel processes either enter the states with the same sync element or stop in accepting states.

For each word w, let $|w|$ denote the length of w. An instantaneous description of an SATM M is of the form $(x, i, (q, \alpha, j))$, where x is the input (excluding the left and right endmarkers ¢ and \$), i is the input head position $(0 \leq i \leq |x|+1)$, and (q, α, j) is the storage state $(1 \leq j \leq |\alpha|+1)$, where a storage state of M is a combination of the state of the finite control, the contents of the storage tape, and the storage head position. The initial ID of M on input x is $I_M(x)=(x, 0, (q_0, \varepsilon, 1))$, where q_0 is the initial state of M, and ε denotes the empty word. An ID is called existential, universal, accepting, and rejecting, respectively, if the corresponding state is existential, universal, accepting, and rejecting, respectively.

Given an SATM M, we write $I \vdash_M I'$ and say I' is a successor of I if the ID I' follows from the ID I in one step of M according to the transition rules. The reflexive transitive closure of $\vdash_M$ is denoted by $\vdash_M^*$. A sequence of ID's of M, $I_0, I_1, \cdots, I_m$ $(m \geq 0)$, is called a sequential computation of M if $I_0 \vdash_M I_1 \vdash_M \cdots \vdash_M I_m$. If $I_0 = I_M(x)$ for some x, we call this sequence a computation path of M on x. Let $\bar{I}$ be a sequential computation of M and $I_1, I_2, \cdots, I_r$ be a subsequence of $\bar{I}$ which consists of all sync ID's of $\bar{I}$. For each $j (1 \leq j \leq r)$, let S_j be the sync element of the sync state in I_j. Then the sequence $S_1, S_2, \cdots, S_r$ is called the sync sequence of $\bar{I}$. A computation tree of M is a finite, nonempty labelled tree with the following properties:

(1) Each node v of the tree is labeled with an ID $l(v)$.

(2) If v is an internal node (a non-leaf) of the tree, $l(v)$ is universal and $\{I | l(v) \vdash_M I\} = \{I_1, \cdots, I_k\}$, then v has exactly k children $v_1, \cdots, v_k$ such that $l(v_i) = I_i$ $(1 \leq i \leq k)$.

(3) If v is an internal node of the tree and $l(v)$ is existential, then v has exactly one child u such that $l(v) \vdash_M l(u)$.

(4) For any two sync sequences $S = S_1, \cdots, S_p$ and $T = T_1, \cdots, T_r$ corresponding to two paths of the tree beginning at the root, it must be satisfied that $S_i = T_i$ for each $i \in \{1, 2, \cdots, \min\{p, r)\}$.

A computation tree of M on input x is a computation tree of M whose root is labeled with $I_M(x)$. An accepting computation tree of M on x is a computation tree of M on x whose leaves are all labeled with accepting ID's. We say that M accepts x if there is an accepting computation tree of M on x. Let $T(M) = \{x | M \text{ accepts } x\}$.

Manuscript received September 20, 1989.

† The authors are with the Faculty of Engineering, Yamaguchi University, Ube-shi, 755 Japan.
†† The author is with the Technical College, Yamaguchi University, Ube-shi, 755 Japan.
††† The author is with Oshima National College of Maritime Technology, Yamaguchi-ken, 742-21 Japan.

For each ID $I=(x, i, (q, \alpha, j))$ of an SATM, let SPACE(I) be the length of α. Let $L: N \rightarrow R$ be a function. We say that an SATM M is $L(n)$ space bounded if for each $n \geqq 1$ and for each input x (accepted by M) of length n, there is an accepting computation tree of M on x such that for each node v of the tree, SPACE $(l(v)) \leqq \lceil L(n)m/2 \rceil$.

An SATM is called one-way if the input head cannot move left, and two-way otherwise. We denote a one-way (two-way) SATM by 1SATM (2SATM), and an $L(n)$ space bounded 1SATM (2SATM) by 1SATM $(L(n))$ (2SATM $(L(n))$). Further we denote a 1SATM $(L(n))$ (2SATM $(L(n))$) with only universal states by 1SUTM $(L(n))$ (2SUTM $(L(n))$). Let 1ATM $(L(n))$ (2ATM $(L(n))$) denote an $L(n)$ space bounded one-way (two-way) ATM, and let 1UTM $(L(n))$ (2UTM $(L(n))$) denote a 1ATM $(L(n))$ (2ATM $(L(n))$) with only universal states. For each $X \in \{$ 1SA, 1SU, 2SA, 2SU, 1A, 1U, 2A, 2U $\}$, Let $\mathcal{L}$[XTM$(L(n))$] denote the class of sets accepted by XTM$(L(n))$'s.

In Ref. (2), the following problems are left open: for any function $L(n)$ such that $L(n) \geqq \log\log n$ and $\lim_{n\to\infty} L(n)/\log n = 0$,

(1) $\mathcal{L}$ [1SUTM$(L(n))$] $\subsetneqq \mathcal{L}$ [2SUTM$(L(n))$] ?
(2) $\mathcal{L}$ [1ATM$(L(n))$] $\subsetneqq$ [1SATM$(L(n))$] ?
(3) $\mathcal{L}$ [2ATM$(L(n))$] $\subsetneqq \mathcal{L}$ [2SATM$(L(n))$] ?
(4) $\mathcal{L}$ [1UTM$(L(n))$] $\subsetneqq \mathcal{L}$ [1SUTM$(L(n))$] ?

In this letter we solve these problems positively. Further we show that there exists an infinite hierarchy among $\mathcal{L}$ [1SUTM$(L(n))$]'s with $\log\log n \leqq L(n) \leqq \log n$.

2. Results

In this section we assume that $L(n)$ be any function such that $L(n) \geqq \log\log n$ and $\lim_{n\to\infty} L(n)/\log n = 0$.

[Theorem 1] There exists a set in $\mathcal{L}$ [2SUTM$(\log\log n)$], but not in $\mathcal{L}$ [1SUTM$(L(n))$].

(Proof) Let $T_1 = \{B(1) \# B(2) \# \cdots \# B(n) 2wcw' \in \{0, 1, 2, c, \#\}^+ | n \geqq 2$ & $(w, w' \in \{0, 1\}^+)$ & $|w| = |w'| = \lceil \log n \rceil$ & $w \neq w'\}$, where for each positive integer $i \geqq 1$, $B(i)$ denotes the string in $\{0, 1\}^+$ that represents the integer i in binary notation (with no leading zeros). It is shown in Ref. (4) that T_1 is in $\mathcal{L}$ [2UTM$(\log\log n)$]. Thus $T_1 \in \mathcal{L}$ [2SUTM$(\log\log n)$]. Below we show that T_1 is not in $\mathcal{L}$ [1SUTM$(L(n))$].

Suppose that there exists a 1SUTM$(L(n))$ M accepting T_1. Let s and t be the numbers of states (of the finite control) and storage tape symbols of M, respectively. For each $n \geqq 2$, let

$$V(n) = \{B(1) \# B(2) \# \cdots \# B(n) 2wcw | w \in \{0, 1\}^+ \ \& \ |w| = \lceil \log n \rceil\}.$$

For each $x = B(1) \# B(2) \# \cdots \# B(n) 2wcw$ in $V(n)$, let $S(x)$ be the set of storage states of M defined as follows:

$S(x) = \{(q, \alpha, j) |$ there exists a computation path $I_M(x) \vdash_M^* (x, r(n), (q', \alpha', j')) \vdash_M (x, r(n)+1, (q, \alpha, j))$ of M on x (that is, $(x, r(n)+1, (q, \alpha, j))$ is an ID of M just after the point where the input head left the symbol "c" of x)$\}$,

where $r(n) = |B(1) \# B(2) \# \cdots \# B(n) 2| + \lceil \log n \rceil + 1$, and let

$C(x) = \{\{\sigma_1, \sigma_2\} | \sigma_1$ and σ_2 are storage states in $S(x)$ such that
(i) in case of $\sigma_1 = \sigma_2$, there exists a sequential computation of M which starts with the ID $(x, r(n)+1, \sigma_1)$ and either terminates in a rejecting ID, or enters an infinite loop, and
(ii) in case of $\sigma_1 \neq \sigma_2$, there exist two sequential computations of M which start with the ID's $(x, r(n)+1, \sigma_1)$ and $(x, r(n)+1, \sigma_2)$, respectively, and terminate in sync ID's with different sync elements$\}$.

(Note that, for each x in $V(n)$, $C(x)$ is not empty, since x is not in T_1, and so not accepted by M.) Then the following proposition must hold.

[Proposition 1] For any two different strings x, y in $V(n)$, $C(x) \cap C(y) = \phi$.

[For otherwise, suppose that $X = B(1) \# B(2) \# \cdots \# B(n) 2wcw$, $y = B(1) \# B(2) \# \cdots \# B(n) 2w'cw'$, $w \neq w'$, $C(x) \cap C(y) \neq \phi$, and $\{\sigma_1, \sigma_2\} \in C(x) \cap C(y)$. Let $z = B(1) \# B(2) \# \cdots \# B(n) 2wcw'$. Since $\{\sigma_1, \sigma_2\} \in C(x)$, there exist computation paths $I_M(z) \vdash_M^* (z, r(n)+1, \sigma_1)$ and $I_M(z) \vdash_M^* (z, r(n)+1, \sigma_2)$. Since $\{\sigma_1, \sigma_2\} \in C(y)$, in case of $\sigma_1 = \sigma_2$, tnere exists a sequential computation of M which starts with the ID $(z, r(n)+1, \sigma_1)$ and either terminates in a rejecting ID, or enters an infinite loop, and in case of $\sigma_1 \neq \sigma_2$, there exist two sequential computations of M which start with the ID's $(z, r(n)+1, \sigma_1)$ and $(z, r(n)+1, \sigma_2)$, respectively, and terminate in sync ID's with different sync elements. This means that z is not accepted by M. This contradicts the fact that z is in $T_1 = T(M)$.]

Let $p(n)$ denote the number of pairs of possible storage states of M just after the point where the input head left the symbol "c" of strings in $V(n)$. Then

$$p(n) = \binom{K}{2} + K,$$

where $K = sL(r(n) + \lceil \log n \rceil) t^{L(r(n) + \lceil \log n \rceil)}$. On the other hand, $|V(n)| = 2^{\lceil \log n \rceil}$, where $|V(n)|$ denotes the number of elements of $V(n)$. Since $\lim_{n\to\infty} L(n)/\log n = 0$, it follows that

$$\lim_{n\to\infty} L(r(n) + \lceil \log n \rceil)/\log(r(n) + \lceil \log n \rceil) = 0. \quad (1)$$

It is easily seen that, for some constant $c \geqq 0$, $r(n) + \lceil \log n \rceil \leqq cn \log n$, and thus $\log(r(n) + \lceil \log n \rceil) \leqq \log n + \log\log n + \log c$. From this and Eq. (1), we have

$$\lim_{n\to\infty} L(r(n)+\lceil \log n\rceil)/(\log n+\log\log n+\log c)=0.$$

From this, it follows that $\lim_{n\to\infty} L(r(n)+\lceil \log n\rceil)/\log n=0$. Therefore, we have $|V(n)|>p(n)$ for large n, and so it follows that for large n there must be two different strings x, y in $V(n)$ such that $C(x)\cap C(y)\neq\phi$. This contradicts Proposition 1, and completes the proof of "$T_1\notin \mathcal{L}$ [1SUTM($L(n)$)]". (Q. E. D.)

[Corollary 1] $\mathcal{L}$ [1SUTM($L(n)$)] $\subsetneq \mathcal{L}$ [2SUTM ($L(n)$)].

[Theorem 2] There exists a set in $\mathcal{L}$ [1SUTM(0)], but not in $\mathcal{L}$ [2ATM($L(n)$)].

(Proof) Let $T_2=\{wcw\in\{0, 1, c\}^+|w\in\{0, 1\}^+\}$. It is shown in Ref. (2) that T_2 is in $\mathcal{L}$ [1SUTM(0)]. On the other hand, it can be shown, by using the same technique as in the proof of Theorem 4 in Ref. (5), that T_2 is not in $\mathcal{L}$ [2ATM($L(n)$)]. (Q. E. D.)

[Corollary 2] For each $X\in\{A, U\}$, $\mathcal{L}$ [1XTM($L(n)$)] $\subsetneq \mathcal{L}$ [1SXTM($L(n)$)] and $\mathcal{L}$ [2XTM($L(n)$)] $\subsetneq \mathcal{L}$ [2SXTM($L(n)$)].

[Corollary 3] $\mathcal{L}$ [1SUTM($L(n)$)] is incomparable with $\mathcal{L}$ [XATM($L(n)$)] for each $X\in\{1, 2\}$.

(Proof) Let T_1 be the set described above. It is shown in Ref. (4) that $T_1\in \mathcal{L}$ [1ATM(loglog n)]. On the other hand, it is shown in the proof of Theorem 1 that $T_1 \notin \mathcal{L}$ [1SUTM($L(n)$)]. From these facts and from Theorem 2, the corollary follows. (Q. E. D.)

We finally show that there exists an infinite hierarchy among the classes of sets accepted by 1SUTM's with space bounds between loglog n and log n.

[Theorem 3] Let $f: N\to R$ be a fully space constructible function[6] such that $1\leq \log\log n\leq f(\lceil \log\log n\rceil)\leq \log n$ for all $n\geq n_0$ (where n_0 is some constant), and $g: N\to R$ be a nondecreasing function such that $\lim_{n\to\infty} g(2n)/f(n)=0$. Further, for each function $h: N\to R$, let $L_h: N\to R$ be the function such that $L_h(n)=h(\lceil \log\log n\rceil)$, $n\geq 1$. Then there exists a set in $\mathcal{L}$ [1UTM($L_f(n)$)] (thus in $\mathcal{L}$ [1SUTM($L_f(n)$)]), but not in $\mathcal{L}$ [1SUTM($L_g(n)$)].

(Proof) Let $S(f)$ be the following set depending on the function f in the theorem: $S(f)=\{B(1)\#B(2)\#\cdots\#B(n)2wcw'\in\{0, 1, 2, c, \#\}^+|n\geq n_0\ \&\ (w, w'\in\{0, 1\}^+)\ \&\ |w|=|w'|=\lceil L_f(n)\rceil\ \&\ w\neq w'\}$ (where n_0 is the constant in the theorem). It is shown in Ref. (4) that $S(f)\in \mathcal{L}$ [1UTM($L_f(n)$)]. On the other hand, by replacing $\lceil \log n\rceil$ and $L(r(n)+\lceil \log n\rceil)$ in the latter part of the proof of Theorem 1 with $\lceil L_f(n)\rceil$ and $L_g(r(n)+\lceil L_f(n)\rceil)$, respectively, and using the same techniques as in this part, we can show that $S(f)\notin \mathcal{L}$ [1SUTM($L_g(n)$)]. (Q. E. D.)

3. Conclusions

This letter solves several open problems left in Ref. (2). It is unknown, however, whether or not $\mathcal{L}$ [1UTM($L(n)$)] $\subsetneq \mathcal{L}$ [1SUTM($L(n)$)] for any L such that $L(n)\geq \log n$ and $\lim_{n\to\infty} L(n)/n=0$.

References

(1) A. Slobodova: "On the power of communication in alternating machines", Proc. 13th MFCS '88, LNCS 324, Springer-Verlag pp. 518-528 (1988).

(2) A. Slovodova: "Some properties of space-bounded synchronized alternating Turing machines with only universal states", Proc. 5th IMYCS '88, Hungarian Academy of Sciences, Budapest (1988).

(3) A. K. Chandra, D. C. Kozen and L. J. Stockmeyer: "Alternation", J. ACM, **28**, pp. 114-133 (1981).

(4) K. Inoue, I. Takanami, R. Vollmar: "Alternating on-line Turing machines with only universal states and small space bounds", Theoretical Computer Science, **41**, pp. 331-339 (1985).

(5) A. Ito, K. Inoue and I. Takanami: "A note on alternating Turing machines using small space", Trans. IEICE, **E70**, 10, pp. 990-996 (Oct. 1987).

(6) J. E. Hopcroft and J. D. Ullman: "Introduction to Automata Theory, Languages, and Computation", Addison-Wesley, Reading, MA (1979).