Comparison of the Power between
Reversal-Bounded ATMs and
Reversal-Bounded NTMs

HIROAKI YAMAMOTO* AND SHOICHI NOGUCHI†

* Department of Information Engineering, Faculty of Engineering,
Yamagata University, Yonezawa-shi, 992 Japan; and
† Research Institute of Electrical Communication, Tohoku University,
Sendai-shi, 980 Japan

We study the power of reversal-bounded ATMs (alternating Turing machines). The results obtained are as follows: (1) Every recursively enumerable set can be accepted by a 1-tape-1-counter ATM which runs in constant reversals (which are the number of times a head changes direction) but not by any 1-tape-1-counter NTM (nondeterministic TM) which runs in constant reversals, where a 1-tape-1-counter ATM (NTM, respectively) is a 1-tape ATM (NTM, respectively) with one counter tape. (2) For functions \( B(n) \) and \( R(n) \) satisfying \( B(n) \leq 2^{O(R(n))} \) and \( B(n) R(n) \geq n \), a class of languages accepted by 1-tape ATMs which run in \( O(R(n)) \) reversal and \( O(B(n)) \) leaf simultaneously is equivalent to a class of languages accepted by NTMs which run in \( O(B(n) R(n)) \) space.

1. INTRODUCTION

There have been several works on the reversal complexity of DTM (deterministic Turing machines) and NTM (nondeterministic TMs) (Baker and Book, 1974; Chan, 1981; Fischer, 1968; Greibach, 1978; Hartmanis, 1968; Kameda and Vollmar, 1970), but few works on the reversal complexity of alternating machines.

Recently, Hromkovic (1985) studied the reversal complexity of alternating \( k \)-counter machines and obtained \( P = 2ACMPRC = 1ACMPRC \), where \( 2ACMPRC = \bigcup_{k \in \mathbb{N}} 2ACMP(k, 1) \) and \( 1ACMPRC = \bigcup_{k \in \mathbb{N}} 1ACMP(k, 1) \). 2ACMP(k, 1) (respectively, 1ACMP(k, 1)) denotes the class of languages accepted by a two-way (respectively, one-way) alternating \( k \)-counter machine running in both time at most \( n^k \) and one counter reversal. In 1986 he also considered the recognition problem of specific languages for alternating machines and proved lower bounds for the complexity measures \( \text{REVERSALS, SPACE, PARALLELISM} \) and \( \text{TIME, SPACE, PARALLELISM} \). PARALLELISM complexity corresponds to the leaf complexity in this paper. In the proof of lower bounds, he used a notion similar to the ECS defined in Section 4.
We study here the power of reversal-bounded ATMs (alternating Turing machines) and then compare it with the power of reversal-bounded NTMs. The organization of the paper is as follows. In Section 2 the fundamental definitions are given. In Section 3 we discuss the power of reversal-bounded ATMs and NTMs in terms of recursively enumerable sets. The difference between the power of reversal-bounded NTMs and DTMs is discussed in Baker and Book (1974). In Section 4 we show the equivalence between the class of languages accepted by reversal- and leaf-bounded 1-tape ATMs and the class of the languages accepted by space-bounded NTMs. Section 5 contains some open problems.

2. Preliminaries

In this section we define ATMs informally and then we define the complexities of ATMs (see Chandra et al. (1981) for formal definitions). ATMs are a generalization of NTMs, described as follows. The states are partitioned into "existential" and "universal" states. As with NTMs, we can view a computation of an ATM as a tree of ID (instantaneous description, which consists of the state of the finite control, tape-head positions and contents of tapes). A tree is a computation tree of an ATM \( M \) on an input \( w \) if its nodes are labeled by IDs of \( M \) on \( w \), such that the descendants of any non-leaf labeled by a universal (existential) ID include all (resp. one) successors of that ID. A computation tree is accepting if the root is labeled by the initial ID and all the leaves are accepting IDs. Without loss of generality we may assume that each ID \( C \) of \( M \) has at most two IDs, each of which is reachable from \( C \) in one step. In this paper, "function" means a non-decreasing function from natural numbers to real numbers.

Definition 1. Let \( R(n) \), \( B(n) \), and \( S(n) \) be functions. It is said that an ATM runs in reversal \( R(n) \) (resp., leaf \( B(n) \), space \( S(n) \)) if, for every accepted input of length \( n \), there is an accepting computation tree such that, for each path from the root to a leaf, the number of times a head changes direction is at most \( R(n) \) (resp., the number of leaves is at most \( B(n) \); each of its nodes is labeled by an ID using at most space \( S(n) \)). Also such ATMs are called \( R(n) \) reversal (\( B(n) \) leaf, \( S(n) \) space, respectively)-bounded ATMs. If an accepting computation tree satisfies two or three complexity conditions, we say that an ATM runs in these complexities simultaneously. For example, it is said that an ATM runs in reversal \( R(n) \) and leaf \( B(n) \) simultaneously if, for every accepted input of length \( n \), there is an accepting computation tree such that the number of times a head changes direction on each path from the root to a leaf is at most \( R(n) \) and the number of
leaves is at most $B(n)$. Such an ATM is also called a $R(n)$ reversal- and $B(n)$ leaf-bounded ATM. The reversal and space complexities of NTMs are defined similarly.

**Definition 2.** A reversal, leaf $(R(n), B(n))$ denotes the class of languages accepted by $k$-tape ATMs running in reversal $O(R(n))$ and leaf $O(B(n))$ simultaneously, where $k$-tape ATMs are ATMs having $k$ storage-tapes (see Fig. 1 for $k$-tape TM).

A reversal, space $(R(n), B(n), S(n))$ denotes the class of languages accepted by $k$-tape ATMs running in reversal $O(R(n))$, leaf $O(B(n))$, and space $O(S(n))$ simultaneously. A reversal, space $(R(n), S(n))$ and Nspace $(S(n))$ are defined similarly for NTMs. If there are no suffixes for the number of tapes, it means the union over the number of tapes. For example,

$$A_{\text{reversal}, \text{leaf}}(R(n), B(n)) = U_{k \geq 1} A_{\text{reversal}, \text{leaf}}(R(n), B(n)).$$

3. **The Power of Reversal-Bounded ATMs**

In this section 1-tape-1-counter TMs which are 1-tape TMs with one counter-tape are introduced. For simplicity, we will call counter-tape and storage-tape, counter and tape, respectively. Further $+1$ ($-1$, respectively) will denote the operation to increase (decrease, respectively) counter by one. We assume that any 1-tape-1-counter TM $M$ cannot operate the counter more than $t$ times (i.e., the number of $+1$ and $-1$ is at most $t$ times, where $t$ is a constant depending on $M$) while the tape head is stationary, and the accepting ID of $M$ is that where the state of finite control is accepting and the content of counter is zero.

**Remark 1.** The restriction that a 1-tape-1-counter TM should only operate the counter a constant number of times while the tape head is stationary is essential in the proof of Lemma 2. We do not know whether Lemma 2 holds without this restriction or not. A further remark related to the relaxation of this restriction is given after Theorem 2.

![Fig. 1. A k-tape Turing machine.](image)
We will show that every recursively enumerable set can be accepted by a 1-tape-1-counter ATM running in constant reversals but not by any 1-tape-1-counter NTM running in constant reversals.

**Lemma 1.** $L = \{wcw | w \in \Sigma^*, \Sigma \text{ is an alphabet, } c \notin \Sigma \}$ can be accepted by a 1-tape 1-counter ATM running in both zero tape reversal and one counter reversal.

**Proof.** We construct a 1-tape-1-counter ATM which accepts in both zero tape reversal and one counter reversal.

Now let input $x$ be $a_1 \ldots a_n$.

**Step 1.** $j \leftarrow 1$ (j is the content of counter).

**Step 2.** If $a_j = c$ then go to Step 5, otherwise go to Step 3.

**Step 3.** Mark $a_j$.

**Step 4.** Universally do both of the following.

   (4-a) $j \leftarrow j + 1$, then go to Step 2.

   (4-b) Let $b$ be the $j$th symbol to the right of $c$. If $a_j = b$ then enter the accepting state, otherwise enter the rejecting state.

**Step 5.** If the $j$th tape square to the right of $c$ is a blank, then enter the accepting state, otherwise enter the rejecting state.

The proof of complexities is straightforward. Q.E.D.

**Theorem 1.** Every recursively enumerable set can be accepted by a 1-tape-1-counter ATM which runs in both four tape reversals and one counter reversal.

**Proof.** The class of recursively enumerable sets is equal to the class of languages accepted by 1-tape DTMs. Hence, for any 1-tape DTM, it suffices to construct a 1-tape-1-counter ATM which simulates it in both four tape reversals and one counter reversal. Let $D$ be a 1-tape DTM. Without loss of generality, $D$ keeps the accepting state when $D$ enters it. First, we define a language $L$ depending on $D$ as follows.

$L = \{z_0 \# z_1 \# \ldots \# z_{2k} \$ z_1 \# \ldots \# z_{2k+1} | \text{ each } z_i \text{ is the string encoding an ID of } D \text{ and identified with an ID of } D. \text{ } z_0 \text{ is an initial ID of } D \text{ and } z_{2k+1} \text{ is an accepting ID of } D. \text{ For } i = 0, \ldots, 2k, \text{ } z_{i+1} \text{ is reachable from } z_i \text{ by one transition of } D. \} \}

By definition of $L$, a string $z_0 \# \ldots \# z_{2k} \$ z_1 \# \ldots \# z_{2k+1}$ is in $L$ if and only if the sequence $z_0, z_1, \ldots, z_{2k+1}$ represents an accepting computation of $D$ on the input denoted in $z_0$. Hence a 1-tape-1-counter ATM which simulates $D$ can be constructed as follows. Now let $w$ be an input.

**Step 1.** Guess a string $z_0 \# \ldots \# z_{2k} \$ z_1 \# \ldots \# z_{2k+1}$ such that the sequence $z_0, \ldots, z_{2k+1}$ represents an accepting computation of $D$ on $w$, where $|z_0| = \ldots = |z_{2k+1}|$. 


Step 2. Comment: In this step, the presence of $z_0 \# \cdots \# z_{2k} \# z_1 \# \cdots \# z_{2k+1}$ in $L$ is checked. Universally do both of the following.

(2-a) Change $z_0 \# \cdots \# z_{2k} \# z_1 \# \cdots \# z_{2k+1}$ into $z_0 \# \cdots \# z_1 \# z_1 \# \cdots \# z_{2k+1}$ according to transitions of $D$, where $z_i'$ ($i = 0, 2, \ldots, 2k$) is the next ID of $z_i$. If $z_0 \# \cdots \# z_{2k}$ is equal to $z_1 \# \cdots \# z_{2k+1}$, then enter the accepting state, otherwise enter the rejecting state.

(2-b) Change $z_2 \# \cdots \# z_{2k} \# z_1 \# \cdots \# z_{2k-1}$ into $z_2 \# \cdots \# z_{2k} \# z_1 \# \cdots \# z_{2k-1}$ according to transitions of $D$, where $z_i'$ ($i = 1, \ldots, 2k - 1$) is the next ID of $z_i$. If $z_1 \# \cdots \# z_{2k-1}$ is equal to $z_2 \# \cdots \# z_{2k}$, then enter the accepting state, otherwise enter the rejecting state.

In the following, we discuss the complexity. The number of tape reversals is at most one in Step 1 for writing $z_0 \# \cdots \# z_{2k} \# z_1 \# \cdots \# z_{2k+1}$ and two in Step 2 for changing the string. By Lemma 1, the reversal required by checking the equality between two strings is at most one for both the tape and the counter. Therefore it is four reversals for tape and one reversal for counter in total.

Q.E.D.

Lemma 2. Let $L = \{ w w^r | w \in \{0, 1\}^* \}$. Let $N$ be a 1-tape-1-counter NTM which accepts $L$ in both $r$ tape reversals and one counter reversal. Then $N$ requires $r = \Omega(n)$.

Proof. Suppose that $L$ is accepted by a 1-tape-1-counter NTM $N$ running in both $r$ tape reversals and one counter reversal. Then we will evaluate the lower bound of $r$. The crossing sequences of $N$ are defined as the sequence of two-tuples "internal state, counter state," where counter state takes three states, that is, "zero state," "nonzero and increasing state," and "nonzero and decreasing state," as in Fig. 2. For simplicity, we will call them Z, I, and D, respectively. First we prove the following sub-lemma. (Note that the transition of the counter state takes the form of $Z \rightarrow I \rightarrow D \rightarrow Z$ because the number of counter reversals is at most one.)

![Fig. 2. Counter states.](image-url)
**Sub-lemma.** Let $x = w_1cw_1$ and $y = w_2cw_2$ be in $L$. If there exists a pair $(P_1, P_2)$ satisfying both the following conditions (i) and (ii), where $P_1$ (resp. $P_2$) is an accepting computation of $N$ on $x$ (resp. $y$), then $w_1cw_2$ and $w_2cw_1$ are also in $L$.

(i) The crossing sequence at the left boundary of $c$ in $P_1$ is the same as that in $P_2$.

(ii) Let $k_1$ be the number of $+1$ and $k_2$ be the number of $-1$ in $P_1$ under the condition that a tape head is on the left $w_1$ of $x$, and $k'_1$ and $k'_2$ are defined similarly for $P_2$ and the left $w_2$ of $y$. Then $k_1 - k_2 = k'_1 - k'_2$.

**Proof of Sub-lemma.** For $w_1cw_1$ and $w_2cw_2$, let $(P_1, P_2)$ be an accepting computation pair satisfying the conditions of the sub-lemma and let the crossing sequence of (i) be $(q_0, X_0) \cdots (q_k, X_k)$. We will show $w_1cw_2 \in L$. ($w_2cw_1 \in L$ is shown similarly.) We make the following claim to prove the sub-lemma.

**Claim.** There is a computation of $N$ on $w_1cw_2$ such that it is equal to $P_1$ on $w_1$ and $P_2$ on $cw_2$, respectively, and for any $j \leq k$, if it has $j$ crossings of the tape head at the left boundary of $c$, then the crossing sequence is $(q_0, X_0) \cdots (q_j, X_j)$.

**Proof of Claim.** It is proved by induction on $j$.

(I) $j = 1$. Obvious.

(II) For $j \leq 1$, assume that the claim holds.

Then we show that $N$ can cross in $(q_{j+1}, X_{j+1})$ if $N$ moves according to $P_1$ or $P_2$ after crossing in $(q_j, X_j)$. The proof is classified into three cases by the value of $X_j$, i.e., $Z$, $I$, $D$.

Case 1. $X_j = Z$.

By the definition of $P_1$ and $P_2$, the proof is straightforward.

Case 2. $X_j = I$.

(2-a) If $X_{j+1} = I$, then the proof is straightforward.

(2-b) $X_{j+1} = D$ and the crossing in $(q_j, X_j)$ is from the left to the right.

If the counter never becomes zero for the time interval from the crossing in $(q_j, x_j)$ to the next crossing, then $N$ can cross in $(q_{j+1}, X_{j+1})$. Hence we show it below.

Let $m_1, m_2, m_3, n_1, n_2$, and $n_3$ be defined as follows (see Fig. 3).

$m_1$: In $P_1$, the number of $+1$ operations during the time interval from the start to the crossing in $(q_j, X_j)$ under the condition that the tape head is on the left $w_1$. 

---

643/75/2-5
Fig. 3. The head movement related to the definition of $m_1$, $m_2$, $m_3$, $n_1$, $n_2$, and $n_3$.

$m_2$: In $P_2$, the number of $+1$ operations during the time interval from the start to the crossing in $(q_j, X_j)$ under the condition that the tape head is on $cw_2$.

$m_3$: In $P_2$, the number of $+1$ operations during the time interval from the crossing in $(q_j, X_j)$ to the crossing in $(q_{j+1}, X_{j+1})$ under the condition that the tape head is on $cw_2$.

$n_1$: In $P_2$, the number of $-1$ operations under the same condition as $m_3$.

$n_2$: In $P_1$, the number of $-1$ operations during the time interval from the crossing in $(q_{j+1}, X_{j+1})$ to a halt under the condition that the tape head is on the left $w_1$.

$n_3$: In $P_2$, the number of $-1$ operations during the time interval from the crossing in $(q_{j+1}, X_{j+1})$ to a halt under the condition that the tape head is on $cw_2$.

Then, by the condition (ii) of the sub-lemma,

$$m_1 - n_2 = -(m_2 + m_3 - n_1 - n_3).$$
(Note that \( k'_1 - k'_2 \) must be \(-(m_2 + m_3 - n_1 - n_3)\) as the counter is zero in accepting IDs.) Hence,

\[
m_1 + m_2 + m_3 = n_1 + n_2 + n_3. \tag{1}
\]

If \( P_2 \) makes the counter become zero during the time interval from the crossing in \((q_j, X_j)\) to the next crossing, then \( m_1 + m_2 + m_3 \leq n_1 \).

If \( m_1 + m_2 + m_3 < n_1 \), Eq. (1) is contradicted. If \( m_1 + m_2 + m_3 = n_1 \), then \( n_2 = n_3 = 0 \) by Eq. (1). Therefore \( X_{j+1} \) must be \( Z \). But this contradicts \( X_{j+1} = D \). Thus, the counter never becomes zero. Hence, the next crossing is in \((q_j + 1, X_{j+1})\).

\((2-c)\) \( X_{j+1} = D \) and the crossing in \((q_j, X_j)\) is from the right to the left.

The proof is similar to that for \((2-b)\).

Case 3. \( X_j = D \).

The proof is similar to that of Case 2.

Thus the claim has been proved.

By the claim, there is a computation of \( N \) on \( w_1cw_2 \) which is equal to \( P_1 \) on \( w_1 \) and \( P_2 \) on \( cw_2 \) respectively and it has the crossing sequence, \((q_0, X_0) \cdots (q_k, X_k)\), at the left boundary of \( c \). For this computation, if \( N \) moves according to \( P_1 \) or \( P_2 \) after crossing in \((q_k, X_k)\), \( N \) can enter the accepting state because the content of the counter in \( P_1 \) is the same as that in \( P_2 \) when \( N \) crosses in \((q_k, X_k)\). Thus \( w_1cw_2 \in L \), and the sub-lemma follows.

Let \( n \) be any odd number. Then there are \( 2^{(n-1)/2} \) words of length \( n \) in \( L \). These words are classified by the conditions (i) and (ii) of the sub-lemma.

I. Classification by the condition (i). If the number of reversals is \( r \), the length of the crossing sequence is at most \( r + 1 \). Consequently, since the number of distinct crossing sequences is \((3b)^{r+1}\), words of length \( n \) in \( L \) are classified into \((3b)^{r+1}\) classes, where \( b \) is the number of states of \( N \).

II. Classification by the condition (ii). The number of operations of +1 and -1 is a constant (\( t \) times) while the tape head is stationary. Hence the total number of counter operations (i.e., +1 or -1) concerning the left part of an input is at most \( t(n-1)(r+1)/2 \). Let \( h \) be the difference between the number of +1 operations and -1 operations. Then

\[-t(n-1)(r+1)/2 \leq h \leq t(n-1)(r+1)/2.
\]

Thus the number of classes is at most \( t(n-1)(r+1) + 1 \). By I and II above, the total number of classes is

\[
(3b)^{r+1}(t(n-1)(r+1) + 1). \tag{2}
\]
If Eq. (2) \( < 2^{(n-1)/2} \), then there is a pair of words \( (w_1 c w_2, w_2 c w_1) \) satisfying the sub-lemma. This means that \( w_1 c w_2 \) and \( w_2 c w_1 \) are in \( L \). This is a contradiction.

Therefore Eq. (2) \( \geq 2^{(n-1)/2} \); we get

\[
\begin{align*}
\begin{array}{c}
r \geq (n - 1 - 2(\log(n - 1) + \log(t) + \log(3b)))/2(1 + \log(3b)),
\end{array}
\end{align*}
\]

and Lemma 2 follows. Q.E.D.

The following theorem is obtained from Lemma 2 directly.

**Theorem 2.** "ATM" cannot be changed into "NTM" in Theorem 1.

**Remark 2.** Lemma 2 is powerful for proving Theorem 2. This fact depends on the restriction stated in Remark 1. Actually, for proving Theorem 2, it suffices to show that \( r \) is not bounded by a constant from above in Lemma 2. Note that the evaluation Eq. (3) of \( r \) in the proof of Lemma 2 shows that \( r \) is not bounded by a constant from above if \( t = 2^{g(n)} \) for \( g(n) = o(n) \), where \( t \) is the number of counter operations while the tape head is stationary.

4. **Relationships between Reversal of ATMs and Space of NTMs**

In the previous section, although we showed that a 1-tape-1-counter ATM running with a constant reversal bound can accept every recursively enumerable set, we do not know whether a 1-tape ATM running with a constant reversal bound can accept it or not.

In this section we show the equivalence between the class of languages accepted by reversal- and leaf-bounded ATMs and that accepted by space-bounded NTMs. Then, from this result and the space hierarchy theorem for NTMs, \( N_{\text{reversal,}}(R(n)) \supseteq A_{\text{reversal,}}(R(n)) \) is obtained. Throughout this paper, without loss of generality, every 1-tape ATM \( M \) moves as follows: The input of length \( n \) stands originally on the tape squares 1, ..., \( n \), and the head of \( M \) stands originally on square 0. \( M \) starts by going in initial state to square 1 and visits square 0 again only if it is in the accepting state, where tape squares are numbered from 0 from the left-end.

4.1. **Extended Crossing Sequences**

In this sub-section, extended crossing sequences (ECSs) are defined. A notion similar to the ECS can be also seen in Hromkovic (1986). He used it to obtain lower bounds for alternating machines.

Let \( M \) be a 1-tape ATM and \( w \) be an input. Let \( T_w \) be a computation tree of \( M \) on \( w \). Then each node of \( T_w \) is labeled as follows: Let \( V \) be a set
of nodes of $T_w$. Then, for any $v \in V$, $v$ is labeled by a pair $(q, i)$, where $q$ is the state of $M$ and $i$ is the boundary number crossed in state $q$ when $M$ moves from ID $v$ to ID $u$; $u$ is the parent of $v$ in $T_w$. The boundary between square $i$ and $i + 1$ has number $i$. The root is labeled by $(q_0, *)$, where $q_0$ is the state of the root ID of $T_w$.

For each $i (i \geq 0)$, we define the ECS at the boundary $i$ in the following manner.

First all of the nodes with boundary-number $i$ in their labels are picked up, and $m$ trees, $T_1, ..., T_m$, satisfying the following condition are constructed from these nodes (see Fig. 4).

(Condition) Let $v$ be any node in $T_j$ ($1 \leq j \leq m$). Then $v_1, ..., v_x$ are sons of $v$ in $T_j$ iff $v_1, ..., v_x$ are successors of $v$ and there is no node with boundary-number $i$ in its label on the path from $v$ to $v_k$ ($1 \leq k \leq s$) in $T_w$, where $v$ and $v_k$ are ignored.

Next, for each node of $T_j$, the EECSs (element of ECS) are defined recursively by the following four rules. Let the label of $v$ be $(q, i)$.

1. If $v$ is a leaf, then the EECS of $v$ is $[q]$.
2. If $v$ has the only son $v_1$ and the EECS of $v_1$ is $[q_1 P]$, then the EECS of $v$ is $[qq_1 P]$.

Next, for each node of $T_j$, the EECSs (element of ECS) are defined recursively by the following four rules. Let the label of $v$ be $(q, i)$.

1. If $v$ is a leaf, then the EECS of $v$ is $[q]$.
2. If $v$ has the only son $v_1$ and the EECS of $v_1$ is $[q_1 P]$, then the EECS of $v$ is $[qq_1 P]$.

Fig. 4. The reduction of computation trees.
(3) If \( v \) has \( k \) sons \( v_1, \ldots, v_k \) and, for each \( s \) \((1 \leq s \leq k)\), the EECS of \( v_s \) is \( Q_s \), then the EECS of \( v \) is \([qQ_1 \cdots Q_k]\).

(4) The EECS of every node is defined by (1), (2), and (3).

Let the EECS of the root node of \( T_j \) be \( Q_j \). Then the ECS at boundary \( i \) is defined as \( Q_1 \cdots Q_m \).

4.2. Reversal Complexity of 1-Tape ATMs

The following is the main theorem in this section.

**Theorem 3.** Let \( B(n) \) and \( R(n) \) be functions satisfying \( B(n) \leq 2^{O(R(n))} \) and \( B(n) R(n) \geq n \). Then,

\[
\text{Areversal, leaf}_1(R(n), B(n)) = \text{Nspace}(R(n) B(n)).
\]

Before proving the above theorem, we define the consistency of ECSs. This is a generalization of the consistency of crossing sequences introduced by Hopcroft and Ullman (1968).

Let \( M \) be a 1-tape ATM, and let \( Q_1 \) and \( Q_2 \) be strings taking the same format as the ECSs of \( M \). (In the rest of the paper, these strings are also called ECS.) Then the consistency of \([M, \alpha, Q_1, Q_2]\) is defined by the following CHECK routine, where \( \alpha \) is a string of tape symbols of \( M \).

**Procedure** CHECK \((\alpha, D, Q_1, Q_2)\)

**COMMENT:** \( D \) takes \( L, R, \) or \((q, x)\) as its value, where \( L \) and \( R \) mean the left-end of \( \alpha \) and the right-end of \( \alpha \), respectively; \( q \) is the internal state of \( M \); and \( x \) is the relative address in \( \alpha \). This procedure is recursive and has eight cases in terms of the value of each parameter. In the CHECK routine, it is checked whether or not the state of \( M \) is consistent with \( Q_1 \) (\( Q_2 \), resp.) each time the head of \( M \) crosses the left-end (the right-end, resp.) of \( \alpha \). If so, then true is returned, otherwise false is returned.

In the following let \( q \) and \( q' \) denote the state of \( M \), and \( P \) and \( P' \) denote the EECSs.

(1) \( D = L \), \( Q_1 = P_1 \cdots P_k \) \((k \geq 2)\). If there exists a permutation \( \sigma \) over \( \{1, \ldots, k\} \) such that all of CHECK \((\alpha, L, P_1, \bar{P}_{\sigma(1)}), \ldots, \) and CHECK \((\alpha, L, P_k, \bar{P}_{\sigma(k)}) \) return true, then return true, otherwise return false, where if \( Q_2 = P'_1 \cdots P'_k \) \((k' \geq 0)\), each \( \bar{P}_j \) is a concatenation of the EECSs from \( \{P'_1, \ldots, P'_k\} \) or an empty string, and \( \bar{P}_1 \cdots \bar{P}_k = P'_1 \cdots P'_k \).

(2) \( D = R \) and \( Q_2 = P'_1 \cdots P'_{k'} \) \((k' \geq 2)\). If there exists a permutation \( \sigma \) over \( \{1, \ldots, k'\} \) such that all of CHECK \((\alpha, R, \bar{P}_{\sigma(1)}, P'_1), \ldots, \) and CHECK \((\alpha, R, \bar{P}_{\sigma(k')}, P'_{k'}) \) return true, then return true, otherwise return false, where the definition of \( \bar{P}_1, \ldots, \bar{P}_{k'} \) is similar to (1) in terms of \( Q_1 = P_1 \cdots P_k \).
(3) \(D = L\) and \(Q_1 = [q_1 \cdots q_m, P_1 \cdots P_k]\) \((m, k \geq 1)\), or \(D = L, Q_1 = [q_1 \cdots q_m]\) and \(Q_2 = [q_1' \cdots q_{m'}', P_1' \cdots P_{k'}']\) \((m', k' \geq 1, m' \geq 0)\). With a head on the left-end of \(x\) and state \(q_1\) initially, simulate \(M\) until any one of (3-1), (3-2), and (3-3) is encountered.

(3-1) \(M\) enters the universal state \(q\). Let \(x'\) be the content of the tape at this point. Then, do \(\text{CHECK}(x', D, Q_1', Q_2')\) for one transition from \(q\) and \(\text{CHECK}(x', D, Q_1'', Q_2'')\) for the other transition, where \(D\) takes \(q\) and the relative address of the head in \(x\), and \(Q_1', Q_1'', Q_2',\) and \(Q_2''\) satisfy (a), (b), and (c) below. If there exist \(Q_1', Q_2', Q_1'',\) and \(Q_2''\) such that both of them return true, then return true, otherwise return false.

(a) If the unchecked part of \(Q_1\) is \([q_i \cdots q_m, P_1 \cdots P_k]\) (or \([q_i \cdots q_m]\)), then, for \(Q_1'\) and \(Q_1''\), one is \([q_i \cdots q_m, P_1 \cdots P_k]\) (or \([q_i \cdots q_m]\), resp.), and the other is an empty string.

(b) If the unchecked part of \(Q_1\) is \(P_1 \cdots P_k\), then \(P_1, \ldots, P_k\) is divided into two parts, and for \(Q_1'\) and \(Q_1''\), one is \(Q_1'\), and the other is \(Q_1''\).

(c) \(Q_2'\) and \(Q_2''\) are defined similarly.

(3-2) The head crosses the left-end boundary of \(x\) and goes out of \(x\). If \(\text{CHECK}(x', L, Q_1', Q_2')\) returns true, then return true, where \(Q_1'\) and \(Q_2'\) are the unchecked parts of \(Q_1\) and \(Q_2\), respectively, and \(x'\) is the content of the tape at this point.

(3-3) The head crosses the right-end boundary of \(x\) and goes out of \(x\). If \(\text{CHECK}(x', R, Q_1', Q_2')\) returns true, then return true, where \(x'\), \(Q_1'\) and \(Q_2'\) are defined as in (3-2).

(4) \(D = R\) and \(Q_2 = [q_1' \cdots q_{m'}, P_1' \cdots P_{k'}']\) \((m', k' \geq 1)\), or \(D = R, Q_1 = [q_1 \cdots q_m, P_1 \cdots P_k]\) and \(Q_2 = [q_1' \cdots q_{m'}]\) \((m', k \geq 1, m \geq 0)\). This case is similar to (3) except that simulation of \(M\) is started with a head on the right-end of \(x\) and state \(q_1'\).

(5) \(D = L, Q_1 = [q_1 \cdots q_m]\) and \(Q_2 = [q_1' \cdots q_{m'}]\) \((m, m' \geq 0)\). Place the head on the left-end of \(x\), and start to simulate \(M\) in state \(q_1\). Then, check whether or not there exists a computation of \(M\) for which \(Q_1\) and \(Q_2\) are consistent with the ECSs of both ends of \(x\). If there exists such a computation, then return true, otherwise return false.

(6) \(D = R, Q_1 = [q_1 \cdots q_m]\) and \(Q_2 = [q_1' \cdots q_{m'}]\). This case is similar to (5) except that simulation of \(M\) is started with a head on the right-end of \(x\) and state \(q_1'\).

(7) \(D = (q, x), Q_1 = [q_1 \cdots q_m]\) and \(Q_2 = [q_1' \cdots q_{m'}]\). This case is similar to (5) except that simulation of \(M\) is started with a head on the position \(x\) and state \(q\), where \(q\) is not regarded as the universal state even if \(q\) is so.

(8) \(D = (q, x)\) but not (7). This case is similar to (3) except that simulation of \(M\) is started with a head on the position \(x\) and state \(q\).

End of procedure.
DEFINITION 3. Let $M$ be a 1-tape ATM. Let $Q_1$ and $Q_2$ be ECSs and $\alpha$ be a string of tape symbols of $M$. We say that $[M, \alpha, Q_1, Q_2]$ is consistent if \text{CHECK}(\alpha, L, Q_1, Q_2) returns true.

Lemma 3, below, is obtained from Definition 3. Its proof is a generalization of the proof for the crossing sequence (Hopcroft and Ullman, 1968) and is not difficult. Hence, it is omitted here.

**Lemma 3.** Let $M$ be a 1-tape ATM. Let $Q_1$, $Q_2$, and $Q_3$ be ECSs, and $\alpha_1$ and $\alpha_2$ be strings of tape symbols of $M$. If $[M, \alpha_1, Q_1, Q_2]$ and $[M, \alpha_2, Q_2, Q_3]$ are consistent, then $[M, \alpha_1 \alpha_2, Q_1, Q_3]$ is consistent.

**Lemma 4.** Let $M$ be a 1-tape ATM, and $w = a_1 \cdots a_n$ be an input. Then $w \in L(M)$ if and only if there exists a positive integer $m$ and a sequence of ECSs, $Q_0, \ldots, Q_m$, satisfying the following conditions.

1. $Q_0 = [q_0[q_1] \cdots [q_n]]$ and $Q_m = \varepsilon$ (empty string), where $q_0$ is the initial state and $q_n$ is the accepting state of $M$.

2. For every $i$ satisfying $1 \leq i \leq m$, $[M, a_i, Q_{i-1}, Q_i]$ is consistent, where if $i > n$, $a_i = B$ (blank symbol).

**Proof.** This lemma is obtained from Lemma 3. Q.E.D.

**Lemma 5.** Let $M$ be a 1-tape ATM running in reversal $O(R(n))$ and leaf $O(B(n))$ simultaneously. Then, for any $w \in L$, there exists an accepting computation tree such that the length of the ECS at each boundary of $M$'s tape is $O(R(n)B(n))$.

**Proof.** Since $M$ is a 1-tape ATM running in reversal $O(R(n))$ and leaf $O(B(n))$ simultaneously, for any input of length $n$ in $L(M)$, there exists an accepting computation tree such that the number of its leaves is $O(B(n))$ and the number of reversals is $O(R(n))$ on any computation path from the root to the leaf.

We consider the ECSs defined by the above accepting computation tree. The number of states contributed to each ECS by one computation path is at most $O(R(n))$. The number of such computation paths is at most $O(B(n))$. Therefore the length of each ECS is $O(R(n)B(n))$. Q.E.D.

**Lemma 6.** Let $R(n)$ and $B(n)$ be functions satisfying $R(n)B(n) \geq n$. Then, $\text{Arvls, leaf} _1(R(n), B(n)) \subseteq Nspace(R(n)B(n))$.

**Proof.** Let $L \in \text{Arvls, leaf} _1(R(n), B(n))$, and $M$ be an $O(R(n))$ reversal- and $O(B(n))$ leaf-bounded 1-tape ATM accepting $L$. We construct a NTM $N$ accepting $L$ in $O(R(n)B(n))$, space below. Now let the input be $w = a_1 \cdots a_n$. 


Step 1 (guess crossing sequences at both boundaries of square 1). \( j \leftarrow 1, Q_1 \leftarrow [q_0[q_a] \cdots [q_a]] \). Then guess the ECS at the right boundary of square 1 and assign it to \( Q_2 \), where \( q_0 \) and \( q_a \) are the initial and the accepting states of \( M \), respectively.

Step 2. Do CHECK(\( a_j, L, Q_1, Q_2 \)), where if \( j > n \) then \( a_j = B \) (blank symbol). If it returns true, go to Step 3, otherwise reject \( w \).

Step 3. If \( Q_2 = \varepsilon \) (empty string) and \( j > n \), then accept \( w \), otherwise go to Step 4.

Step 4. \( j \leftarrow j + 1 \) and \( Q_1 \leftarrow Q_2 \). Then guess the ECS at the right boundary of square \( j \), and assign it to \( Q_2 \). Go to Step 2.

The fact that \( N \) accepts \( L \) is obtained from Lemma 4.

The space used by \( N \) depends on the CHECK routine. We can show that the space required by the CHECK routine is at most \( 2(|Q_1| + |Q_2|) \) by induction on the depth of recursion in the CHECK routine. Then, from Lemma 5, we obtain \( 2(|Q_1| + |Q_2|) = O(R(n) B(n)) \). Q.E.D.

**Lemma 7.** Let \( R(n) \) and \( B(n) \) be functions satisfying \( R(n) B(n) \geq n \), and \( R_1(n) \geq n \) be any function. Then,

\[
N_{\text{reversal, space}}(R_1(n), R(n) B(n)) \leq A_{\text{reversal, leaf, space}}(R(n) + \log B(n), B(n), R_1(n)).
\]

**Proof.** Let \( L \in N_{\text{reversal, space}}(R_1(n), R(n) B(n)) \) and \( N \) be an \( O(R_1(n)) \) reversal- and \( O(R(n) B(n)) \) space-bounded 1-tape NTM accepting \( L \).

In the following, we regard the space of length \( O(R(n) B(n)) \) as the collection of \( O(B(n)) \) blocks each of which consists of \( R(n) \) squares, and "boundary" means the boundary between the blocks. Then a 1-tape ATM \( M \) accepting \( L \) is constructed as follows.

The tape of \( M \) has four tracks. The input is given to the first track and tracks from the second to the fourth are used to store crossing sequences. They are stored from square \( n + 1 \). The squares from 1 to \( n \) are used to indicate the position of each crossing sequence if it is in the squares from 1 to \( n \) (see Fig. 5).

![Fig. 5. The tape of \( M \).](image-url)
Step 1 (initial setting). In the following the crossing sequences, $s_1$, $s_2$, $s_3$, are stored in the second, the third, and the fourth tracks, respectively. First $s_1 \leftrightarrow q_0q_3$, $s_2 \leftrightarrow c$, and print the symbol * on the second track of the square 1 (because $s_1$ is the crossing sequence at the left boundary of square 1). Then go to Step 2.

Step 2 (guess the crossing sequences at both boundaries of each block).

**Procedure** CROSS($s_1$, $s_2$).

Guess whether or not $s_1$ and $s_2$ are the crossing sequences at both boundaries of a block. If so, go to (1–1), else go to (1–2).

(1–1) End the CROSS routine and go to Step 3.

(1–2) Guess the crossing sequence at the middle boundary between two boundaries corresponding to $s_1$ and $s_2$, and assign it to $s_3$. If the boundary corresponding to $s_3$ is in from square 1 to square $n$, then mark that position (i.e., print * on the right square of the boundary). Then, do both of the following, (1–2–a) and (1–2–b), universally.

(1–2–a) $s_2 \leftrightarrow s_3$ and CROSS($s_1$, $s_2$).

(1–2–b) $s_1 \leftrightarrow s_3$ and CROSS($s_1$, $s_2$).

We specify how to store the crossing sequences before describing Step 3. The time interval from a reversal to the next reversal is called a phase. $M$ provides two squares, $n + 2t - 1$ and $n + 2t$, in each track for the $t$th phase of a computation of $N$. Now, let a crossing sequence of $N$ be $q_1 \cdots q_s$ and assume that $q_j$ ($1 \leq j \leq s$) is generated by the $t$th phase. Then, each $q_j$ is stored according to the following rules.

(a) In the case where this crossing sequence corresponds to $s_1$, if $t_j$ is odd then $q_j$ is stored in square $n + 2t_j - 1$, otherwise in square $n + 2t_j$.

(b) In the case where this crossing sequence corresponds to $s_2$ or $s_3$, if $t_j$ is even then $q_j$ is stored in the square $n + 2t_j - 1$, otherwise in square $n + 2t_j$. As an exception, when $s_1 \leftrightarrow s_3$, the crossing sequence is rewritten under the rule (a).

The above rules save reversals in Step 3.

Step 3 (check the consistency of $s_1$ and $s_2$). This is done by guessing the crossing sequences and checking their consistency sequentially from the left-end in the block. In this step, $\tilde{s}_1$ and $\tilde{s}_2$ are stored in the second and the fourth tracks in the same way as $s_1$ and $s_2$, respectively.

(3–1) $\tilde{s}_2 \leftarrow s_1$.

(3–2) Guess whether or not the square checked just now is the right-end in the block. If so, go to (3–4), else $\tilde{s}_1 \leftarrow \tilde{s}_2$ and the crossing sequence at the right boundary of the square is guessed. Then assign it to $\tilde{s}_2$ and go to (3–3).
(3–3) Do CHECK(a, L, \(\tilde{s}_1, \tilde{s}_2\)). If it returns true then go to (3–2), else enter the rejecting state, where \(a\) is the corresponding input symbol if the block includes a part of input, otherwise \(a\) is a blank symbol of \(N\). (\(M\) can find the corresponding symbol by * in the second and third tracks.)

(3–4) \(\tilde{s}_1 \leftarrow \tilde{s}_2, \tilde{s}_2 \leftarrow s_2\), and do CHECK(a, L, \(\tilde{s}_1, \tilde{s}_2\)). If it returns true then enter the accepting state, else enter the rejecting state.

**Analysis of the complexities.** The depth of the recursion in Step 2 is \(O(\log B(n))\). The job of each level of the recursion is only to write the crossing sequences, and the reversal required by it is at most two. Therefore the reversal required by Steps 1 and 2 is \(O(\log B(n))\). In Step 3 it is possible to check the consistency of crossing sequences in six reversals if they are stored under the rules (a) and (b). Since the number of squares in the block is at most \(R(n)\), the reversal required by Step 3 is \(O(R(n))\). Hence the total reversal is \(O(R(n) + \log B(n))\). The leaf complexity of \(M\) is equal to the number of blocks, i.e., \(O(B(n))\). The space complexity is \(O(R_1(n))\) because it depends on the space to store the crossing sequences. Q.E.D.

**Proof of Theorem 3.** \(\exists\) reversal, leaf\(_1\)(\(R(n), B(n)\))

\[
\leq N\text{space}(R(n), B(n)) \quad \text{(by Lemma 6)}
\]

\[
\leq N\text{space}_1(R(n), B(n))
\]

\[
\leq N\text{reversal, space}_1(2^{O(R(n)B(n))}, R(n), B(n)) \quad \text{(by Lemma 7)}
\]

\[
\leq N\text{reversal, leaf}_1(R(n) + \log B(n), B(n))
\]

\[
\leq N\text{reversal, leaf}_1(R(n), B(n)) \quad \text{(by } B(n) \leq 2^{O(R(n))})
\]

Q.E.D.

We can get a reversal-space trade-off result for NTMs as follows.

**Corollary 1.** Let \(R(n) \geq n\) and \(S(n) \geq n\) be functions. Then,

\[
N\text{reversal, space}_1(R(n), S(n)) = N\text{reversal, space}_1(S(n), R(n))
\]

**Proof.** Consider the case of \(B(n) = 1\) in Lemma 7. Then the corollary follows directly. Q.E.D.

Next, we will show the difference in the power between reversal-bounded 1-tape ATMs and reversal-bounded 1-tape NTMs.
Theorem 4 (Seiferas, 1977). Let \( S_1(n) \geq n \) be a space constructable function, and \( S(n) \geq n \) be a function satisfying \( S(n + 1) = O(S_1(n)) \) and \( S(n + 1) = o(S_1(n + 1)) \). Then,

\[
N_{\text{space}}(S(n)) \subseteq N_{\text{space}}(S_1(n)).
\]

The following corollary is obtained from Theorems 3 and 4.

**Corollary 2.** For any function \( R(n) \geq n \), if there exists a function \( B(n) \) satisfying the following conditions, then

\[
N_{\text{reversal}}(R(n)) \subseteq N_{\text{reversal}}(R(n)).
\]

(Condition 1) \( B(n) \leq 2^{O(R(n))} \) and \( R(n) B(n) \) is a space constructable function.

(Condition 2) \( R(n + 1) = O(R(n) B(n)) \) and \( R(n + 1) = o(R(n + 1) B(n + 1)) \).

**Proof.**

\[
N_{\text{reversal}}(R(n)) = N_{\text{space}}(R(n)) \quad \text{(by Corollary 1)}
\]
\[
\subseteq N_{\text{space}}(R(n) B(n)) \quad \text{(by Theorem 4)}
\]
\[
= A_{\text{reversal, leaf}}(R(n), B(n)) \quad \text{(by Theorem 3)}
\]
\[
\subseteq A_{\text{reversal}}(R(n)).
\]

Q.E.D.

4.3. Reversal Complexity of Off-line 1-Tape ATMs

We discuss off-line 1-tape TMs which are TMs having one storage tape and a read-only two-way input tape. Similarly to 1-tape ATMs, we do not know whether an off-line 1-tape ATM running with a constant reversal bound can accept every recursively enumerable set or not.

We can extend the ECSs to off-line 1-tape ATMs by attaching the input head position to each state. Then the CHECK routine and the consistency of the ECSs are defined similarly, and the result, similar to that in Theorem 3, is obtained as follows. The class of languages for off-line machines is denoted by the prefix “off-,” for example, “off-Areversal, space,” and “off-Nspace.”

**Theorem 5.** Let \( R(n) \) and \( B(n) \) be functions satisfying \( B(n) \leq 2^{O(R(n))} \). Then,

\[
\text{off-Areversal, leaf}_1(R(n), B(n)) = \text{off-Nspace}(R(n) B(n) \log(n)).
\]
Sketch of Proof. \( \subseteq \): This inclusion is obtained by modifying Lemma 6 as the length of each ECS is at most \( O(R(n) B(n) \log(n)) \).

\( \supseteq \): Greibach (1978, Theorem 4.2) has shown \( \text{off-Nspace}(R(n) B(n) \log(n)) \subseteq \text{off-Nreversal},(R(n) B(n)) \). Since an off-line 1-tape ATM can simulate \( O(R(n) B(n)) \) reversal-bounded off-line 1-tape NTM in \( O(R(n) + \log B(n)) \) reversals and \( O(B(n)) \) leaves simultaneously by partitioning \( O(R(n) B(n)) \) reversals into \( O(B(n)) \) blocks each of which has \( R(n) \) reversals. Hence \( \text{off-Nspace}(R(n) B(n) \log(n)) \subseteq \text{off-Areversal}, \text{leaf},(R(n), B(n)) \). Q.E.D.

5. Conclusions

There are some open problems concerning this paper as follows.

1. In Section 3, we assume that 1-tape–1-counter TMs cannot operate the counter more than a constant while the tape head is stationary. Can this assumption be removed?

2. Can the results similar to those in Theorems 1 and 2 be obtained for 1-tape TMs or off-line 1-tape TMs?

Received August 27, 1986; accepted April 1987

References


