TAPE REVERSAL BOUNDED
TURING MACHINE COMPUTATIONS

J. Hartmanis

TECHNICAL REPORT
No. 68-7
February 1968

Department of Computer Science
Cornell University
Ithaca, N.Y. 14850
TAPE REVERSAL BOUNDED TURING MACHINE COMPUTATIONS*

J. Hartmanis
Department of Computer Science
Cornell University
Ithaca, N. Y.

ABSTRACT

This paper studies the classification of recursive sets by the
number of tape reversals required for their recognition on a two-
tape Turing machine with a one-way input tape.

This measure yields a rich hierarchy of tape reversal limited
complexity classes and their properties and ordering are inves-
tigated. The most striking difference between this and the
previously studied complexity measures lies in the fact that the
"speed-up" theorem does not hold for slowly growing tape reversal
complexity classes. These differences are discussed, and several
relations between the different complexity measures and languages
are established.

* This research has been supported in part by National Science
  Foundation Grant GP-6426.
I. INTRODUCTION

Recent work in automata theory has discussed several computational complexity measures and shown that the quantitative aspects of computation can be submitted to a rigorous mathematical analysis [1-7].

Furthermore, this initial work on computational complexity has stimulated a more quantitative approach to other parts of automata theory and has given several measures against which to compare the computational power of automata. For example, recent work has established bounds on the memory and time required for the recognition of context-free languages and has characterized the computational power of certain stack automata in terms of memory bounded Turing machine computations [8,9,10,11]. The same approach has been applied to the study of the complexity of decision problems, and has raised several interesting problems about the recognition of the set of primes by memory bounded automata [12,13,14].

In this paper we continue the study of computational complexity by investigating a complexity measure based on the number of tape reversals which are required to perform the computation on an on-line Turing machine.
This measure yields a rich hierarchy of tape-reversal complexity classes and their properties and ordering are investigated. The most striking difference between this and the previously studied complexity measures lies in that the "speed-up" theorem [2,3,4] does not hold for slowly growing reversal complexity classes. These differences are discussed and several relations between the different measures are established.

II. PRELIMINARIES

In this paper we consider the classification of Turing machine computations by the number of tape reversals performed during the computation.

The computing devices which we consider are two-tape Turing machines with a one-way — read-only input tape. They have also been referred to as one-tape — on-line Turing machines [4].

To facilitate the discussion of these computing devices, we give a more formal description.

Definition. A Turing machine $\mathcal{T}$ is a sixtuple

$$\mathcal{T} = (S, \Sigma, \Gamma, \delta, s_0, F),$$

where
$S$ is the non-empty, finite set of states;

$\Sigma$ is the non-empty, finite set of input symbols and 
   $a$ in $\Sigma$ is referred to as the end symbol;

$\Gamma$ is the non-empty, finite set of working tape symbols 
   and the symbol $-$ in $\Gamma$ is referred to as the blank;

$\delta$ is the function describing machine operations,
   $\delta : S \times \Sigma \times \Gamma \rightarrow S \times \Gamma \times \{0,1\} \times \{-1,0,1\}$;

$e_0$ in $S$ is the starting state;

$F$ is the set of accepting states, $F \subseteq S$.

The formalism

$\delta(s, x, y) = (s', y', m_1, m_2)$

means the following: if $\mathcal{F}$ is in state $s$ scanning the symbol $x$ on the input tape and the symbol $y$ on the working tape, then

a) $\mathcal{F}$ enters the new state $s'$.

b) moves the reading head one square to the right if $m_1 = -1$, or makes no move if $m_1 = 0$.

c) overprints the symbol $y'$ on the square scanned on the working tape and moves the head on this tape one square left, no move, or one square right if $m_2 = -1, 0, 1$, respectively.
An input is an element of

\((\Sigma^* \cup \epsilon)^*\epsilon\),

where \(\Sigma^*\) denotes all finite strings over the alphabet \(\Sigma\).

To give a precise meaning to the operations of \(\mathcal{T}\) we define an instantaneous description of \(\mathcal{T}\) which gives the complete information about the state of \(\mathcal{T}\), the input tape, the working tape and the head positions on these tapes.

Definition. An instantaneous description of \(\mathcal{T}\) is an element of

\[ S \times (\Sigma^* \cup \Sigma^* \epsilon \cup \epsilon^*) \times (-\Gamma^* \cup \Gamma^* \cup \epsilon^*) \times \{\star\} \times \{\star\} \times \{\star\} \times (-\Gamma^* \cup \Gamma^* \cup \epsilon^*) \times \{\star\} \times \{\star\} \times \{\star\}, \]

where \(\star\) is a new symbol not contained in \(\Sigma\) or \(\Gamma\).

The instantaneous description

\[ I = s, a_1 a_2 \ldots a_i \epsilon a_{i+1} \ldots a_n \epsilon, b_1 b_2 \ldots b_j \epsilon b_{j+1} \ldots b_m \]

denotes the fact that \(\mathcal{T}\) is in state \(s\) with the input \(a_1 a_2 \ldots a_n \epsilon\), and the reading head is scanning the \(i\)-th symbol \(a_i\); the pattern \(-b_1 b_2 \ldots b_j b_{j+1} \ldots b_m\) is written on the working tape and the read-write head is scanning the symbol \(b_j\). The blank symbols are always added to the ends of the
working tape pattern. We can assume without any loss of
generality that $\mathcal{T}$ never prints the blank symbol.

Next we relate two instantaneous descriptions through an
operation of $\mathcal{T}$. The relation $\Rightarrow$ holds between two instan-
taneous descriptions $I_1$ and $I_2$, if and only if one operation
$\mathcal{T}$ transforms $I_1$ into $I_2$. For example, if

$$\delta(s, a_i, b_j) = (s', b', 1, -1)$$

then

$$s, a_1 a_2 \ldots a_i \dagger a_{i+1} \ldots a_n \varepsilon,$$

$-b_1 b_2 \ldots b_j \dagger b_{j+1} \ldots b_m \Rightarrow$

$$s', a_1 a_2 \ldots a_{i+1} \dagger a_{i+2} \ldots a_n \varepsilon,$$

$-b_1 b_2 \ldots b_{j-1} \dagger b' b_{j+1} \ldots b_m \varepsilon.$

We write

$$I_1 \Rightarrow I_j$$

if, and only if, there exists a finite sequence of instantaneous
descriptions

$$I_1, I_2, \ldots, I_k,$$
such that

\[ I_1 = I_1 \Rightarrow I_2 \Rightarrow \cdots \Rightarrow I_k = I_j. \]

**Definition.** The input

\[ w = a_1 a_2 \cdots a_n \]

is accepted by \( \mathcal{T} \) if, and only if

\[ s_0, a_1 a_2 \cdots a_n e, - t - \Rightarrow \]

\[ s, a_1 a_2 \cdots a_n e +, - b_1 b_2 \cdots b_j + b_{j+1} \cdots b_m - \]

for some \( s \) in \( F \) and

\[ \delta(s, e, b_j) = (s, b_j, 0, 0). \]

The set of sequences accepted by \( \mathcal{T} \) is denoted by \( T(\mathcal{T}) \).

If for \( \mathcal{T} \)

\[ I_1 \Rightarrow I_2 \Rightarrow \cdots \Rightarrow I_n, \]

then the *motion sequence* for this computation is given by the corresponding working tape motions:

\[ m_1, m_2, m_3, \ldots, m_{n-1}. \]
where \( m_1 = -1, 0, 1 \) if going from \( I_i \) to \( I_{i+1} \) \( \mathcal{T} \) moved its head on the working tape one square left, no motion, or one square right, respectively.

Definition. The number of tape reversals in the computation of \( \mathcal{T} \) described by

\[
I_1 \Rightarrow I_2 \Rightarrow \cdots \Rightarrow I_n
\]

is given by the number of sign changes in the corresponding motion sequence

\[
m_1, m_2, \ldots, m_{n-1}.
\]

For example, the motion sequence

\[
0 1 0 0 1 1 -1 1 0 -1 0 1
\]

has four sign changes and thus the corresponding computation required four tape reversals.

Definition. The machine \( \mathcal{T} \) accepts the set \( A \) with \( R(n) \) tape reversals if, and only if

\[
A = T(\mathcal{T})
\]

and \( \mathcal{T} \) makes no more than \( R(n) \) tape reversals for any input \( w \) of length \( n \), \( l(w) = n \).
The class of all sets of sequences accepted with $R(n)$ tape reversals is denoted by $C_{R(n)}$ and is referred to as a complexity class. The set containing all Turing machines which do not exceed $R(n)$ tape reversals for inputs of length $n$ is denoted by $M_{R(n)}$. Thus $A$ is in $C_{R(n)}$ if, and only if, there exists a $\mathcal{F}$ in $M_{R(n)}$ such that

$$T(\mathcal{F}) = A.$$ 

III. REVERSAL COMPLEXITY CLASSES

In this section we investigate some properties of tape reversal complexity classes. The first set of results establishes a relation between the number of tape reversals and the computation time and shows that for a computable $R(n)$,

$$T(\mathcal{F}) \text{ in } C_{R(n)}$$ 

implies that $T(\mathcal{F})$ is a recursive set.

Lemma 1. For any instantaneous description

$$I = a_1 a_2 \ldots a_i + a_{i+1} \ldots a_n,$$

$$-b_1 b_2 \ldots b_j + b_{j+1} \ldots b_m.$$
it is recursively decidable whether $T$ will reverse its working tape when started in $I$.

**Proof.** Let $|S|$ and $|Γ|$ denote the size of the sets $S$ and $Γ$, respectively. Then we note that, if $T$ has not moved one of its heads in

$$t = |S| \cdot |Γ| + 1$$

operations, then the configuration of state of $T$, input symbol and working tape symbol must repeat itself and $T$ must cycle without moving its heads. Therefore, if $T$ does not cycle and does not reverse its working tape in $t(m+n)$ operations $T$ must have moved its working tape head onto a blank. By the same reasoning we now conclude that after $T$ has entered the blank part of the working tape $T$ must either reverse its tape in $tn$ operations or it will never reverse its tape because it is cycling or has stopped. Thus by observing $T$ for

$$t \cdot (m+2n)$$

operations we can determine whether $T$ will reverse its tape when started in $I$.

**Lemma 2.** For every $T$ there exists a positive constant $C$ such that if $T$ stops for an input $w$ of length $n$, then
where \( R \) denotes the number of tape reversals, \( L \) denotes the number of working tape squares used and \( T \) denotes the number of operations performed in this computation.

**Proof.** Recall that at the start of the computation the working tape is blank and that

\[
t = |S| \cdot |T| + 1.
\]

If no tape reversals are performed by \( \mathcal{T} \) then the computation time before stopping cannot exceed \( nt \), since \( \mathcal{T} \) does not cycle and it can spend no more than \( t \) operations scanning a square on the input tape. Similarly, we conclude that each time \( \mathcal{T} \) enters the blank part of the working tape it cannot write on more than \( t \) tape squares without moving its input head or reversing the working tape. Since there are \( n \) input symbols and \( R \) tape reversals, we conclude that \( \mathcal{T} \) can write on at most \( t(R + n) \) working tape squares before stopping. Thus

\[
t(R + n) \geq L,
\]

which establishes the first inequality.

We now use this conclusion to compute a bound for \( T \).

The machine \( \mathcal{T} \) can perform no more than \( t \) operations without
moving one of its heads. Since the working tape length is bounded by \( t(R + n) \) and the input tape length is \( n \), we conclude that without reversing the working tape \( \mathcal{T} \) cannot perform more than

\[
t[(tR + tn) + n] \leq t^2R + 2t^2n
\]

consecutive operations (this accounts for scanning both tapes). Since the number of tape reverses is \( R \), we see that no more than

\[
(R + 1)(t^2R + 2t^2n)
\]

operations can be performed before \( \mathcal{T} \) stops. Thus

\[
T = t^2R^2 + 2t^2nR + t^2R + 2t^2n
\]

and, for \( C \leq 3t^2 \), we have

\[
R \leq T \leq C(R^2 + nR + n),
\]

as was to be shown.

We say that the Turing machine \( \mathcal{T} \) defines \( R(n) \) if \( \mathcal{T} \) stops for all inputs, operates with \( R(n) \) reversals and for some input of length \( n \) performs exactly \( R(n) \) reversals.
Corollary. Let $T$ stop for all inputs and define $R(n)$ such that $R(n) \geq n$. Then for some positive $C$

$$T(n) \leq R(n) \leq C \sqrt{T(n)}.$$ 

Proof. Immediate consequence of Lemma 2.

Theorem 1. If $R(n)$ is a computable function, then $A$ in $C_R$ implies that $A$ is a recursive set.

Proof. If $A$ is in $C_R$, then there exists a $T$ in $M_R(n)$ (that is, $T$ operates within $R(n)$ tape reversals) such that

$$A = T(T).$$

From the previous lemma we know that if $T$ performs no more than $R(n)$ tape reversals, then the computation time $T(n)$ is bounded by

$$T(n) \leq C(R^2(n) + nR(n) + n).$$

Thus to determine whether the string $w$, $l(w) = n$, is in $A$ we determine whether $T$ accepts $w$ in $C(R^2(n) + nR(n) + n)$ operations. Hence we have an algorithm to test whether $w$ in in $A$ and therefore $A$ is a recursive set, as was to be shown.
IV. FINITE REVERSAL COMPUTATIONS

We now turn to the problem of determining the minimal $R(n)$ in which specific computations can be performed. Later, we will discuss the general problem for which $R_1(n)$ and $R_2(n)$ can we show that

$$C_{R_1(n)}^c \subseteq C_{R_2(n)}^c.$$  

The main result of this section shows that for every

$$R(n) = k, \quad k = 1, 2, \ldots,$$

there exists a set $A_k$ which is in $C_k$ and not in $C_{k-1}$.

In the next section we will show that there also exist unbounded (slowly growing) $R(n)$ and $A$ in $C_{R(n)}$ such that $A$ is not in $C_{R(n)-1}$. Thus even if $R(n)$ goes to infinity, there are computations which can be performed with $R(n)$ reversals but cannot be performed with $R(n)-1$ reversals.

These results show that (off-line) bounded tape reversal computations behave radically differently from the time and memory bounded computations for which we have general speed up theorems [2,3]: if $A$ is accepted in time $T(n)$ or with tape length $L(n)$ then $A$ is also accepted in time $\frac{1}{2}T(n)$.
and on $\frac{1}{2}L(n)$ tape. Furthermore it has been shown recently [15] that for fast growing $R(n)$ there also exists a general speed up theorem and another result [16] shows that for one-tape off-line reversal bounded Turing machines there exists a general speed up theorem provided $R(n) > 1$.

If a set $A$ is accepted by $\mathcal{F}$ without any tape reversals, that is $R(n) = 0$, then $\mathcal{F}$ cannot utilize its working tape in the computation and therefore the same set can be accepted by a finite state machine.

**Lemma 3.** The set $A$ is in $C_0$ if, and only if, $A$ is a regular set.

We now show that for every $k$, $k = 1, 2, \ldots$, there is a set of sequences recognizable with $k$ tape reversals and not with $k-1$ tape reversals.

**Theorem 2.** The set

$$A_k = \{ o^1 \theta o^1 \theta o^2 \theta o^2 \theta \ldots \theta o^j \theta o^j \theta | 1 \leq j \leq k \}$$

is in $C_k$ and not in $C_r$, for $r < k$.

**Proof.** It is easily seen that $A_k$ is acceptable with $R(n) = k$ tape reversals. Thus $A_k$ is in $C_k$.

To prove that $A_k$ is not in $C_{k-1}$, we first show that if
then $\mathcal{F}$ cannot perform more than
\[
t = |S| \cdot |\Gamma| + 1
\]
operations without moving its working tape while the input head is scanning one of the segments \(i^r\) of the input. Otherwise during the computation $\mathcal{F}$ enters an instantaneous description
\[
I_1 = s, o_1^1 \# o_1^1 \# \ldots \# o_s^i \# o_{q-1}^i \# \ldots \# o_j^i \# o_{j+1}^i \#
\]
\[
- b_1 b_2 \ldots b_d + b_{d+1} \ldots b_m -
\]
and, since for $|S| \cdot |\Gamma| + 1$ operations the head on the working tape is not moved, the state of $\mathcal{F}$ and the symbol printed on the working tape square must repeat itself, that is
\[
I_1 \Rightarrow s, o_1^1 \# o_1^1 \# \ldots \# o_{s+p}^i \# o_{q-p}^i \# \ldots \# o_j^i \#
\]
\[
- b_1 b_2 \ldots b_d + b_{d+1} \ldots b_m -
\]
But then $\mathcal{F}$ repeats itself everytime it has scanned $p$ zeros of this segment and a change of the length of this segment of zeros by $p$ additional zeros cannot be detected by $\mathcal{F}$. Thus
we conclude that if $T$ accepts
\[ o_1 \# o_1 \# o_2 \# o_2 \ldots \# o_r \# o_r \ldots \# o_j \# o_j. \]

it will also accept
\[ o_1 \# o_1 \# o_2 \# o_2 \ldots \# o_{r+p} \# o_r \ldots \# o_j \# o_j. \]

Since the second sequence is not in $A_k$, we conclude that scanning any segment of zeros $T$ must move its working tape head at least once every $t$ operations.

Let
\[ T(S) = A_k \]

and assume that $T$ performs no more than $k-1$ tape reversals. Consider now a sequence
\[ w = o_1^{n_1} \# o_1^{n_1} \# o_2^{n_2} \# o_2^{n_2} \ldots \# o_k^{n_k} \# o_k^{n_k} \in A_k. \]

Since there are $k$ segments of the form
\[ o_1^{n_1} \# o_1^{n_1} \]
in $w$ and since $T$ can perform only $k-1$ tape reversals, we conclude that while $T$ is scanning one of these segments the
working tape is not reversed. Our previous observation, furthermore, assures us that by picking a sequence of rapidly increasing \( n_1, n_2, \ldots, n_k \) we can force \( \mathcal{T} \) to enter the blank part of the working tape while it is scanning the first part of each of the segments

\[
o_i \not\circ n_i, \quad i = 1, 2, \ldots, k,
\]

and stay on the blank part for at least \( |S| \cdot |\Gamma| + 1 \) operations. Assume now that while \( \mathcal{T} \) is scanning the segment

\[
o_i \circ n_i
\]

the working tape is not reversed. Then while \( \mathcal{T} \) is scanning the first segment \( o_i \) it enters the blank part of the working tape and, since it stays on the blank part for at least \( |S| \cdot |\Gamma| + 1 \) operations, we conclude that it must repeat the state of \( \mathcal{T} \) and the symbol under the working tape head. More formally, \( \mathcal{T} \) enters \( I \),

\[
I = s, o_1 \circ n_1 \not\circ \ldots \not\circ o_{i-1} \circ o_i \circ o_{i+1} \not\circ \ldots \not\circ o_{k} \not\circ o_{k+1} = b_1 b_2 \ldots b_{\nu} t \not\circ -,
\]
\[ I \pi_{s}, o^{n_1} \pi o^{n_1} \ldots o^{n_{i-1}} \pi o^{u-p} \pi o^{n_1} \ldots o^{n_k} \pi o^{n_k_e} \]
\[ - b_1 b_2 \ldots b_v b_{v+1} \ldots b_{v+r} b_v \pi - \]

This implies that if we replace in \( w \) the segment
\[ o^{n_i} \pi o^{n_i} \] by \[ o^{n_i+pq} \pi o^z \], \( q, z = 0, 1, 2, \ldots \),
then \( \mathcal{T} \) will also scan this new segment without reversing its working tape.

Consider now the actions of \( \mathcal{T} \) on the strings
\[ w_{q,s} = o^{n_1} \pi o^{n_1} \pi o^{n_2} \pi o^{n_2} \ldots \pi o^{n_i+pq} \pi o^z \], \( q, s = 1, 2, \ldots \).

Because of the previous conclusion we know that \( \mathcal{T} \) scans the segment
\[ o^{n_i+pq} \pi o^z \]
without reversing its working tape and that \( \mathcal{T} \) enters the blank part of the working tape while scanning the first \( n_i \) zeros of this segment. Thus for an input \( w_{q,s} \) \( \mathcal{T} \) records a
fixed pattern on its working tape and then enters the blank part of its tape. After the end of $w_{q,z}$ is reached the machine can only use the working tape and in no more than $k-1$ tape reversals it must determine whether $w_{q,z}$ is in $A_k$.

that is whether

$$n_i + pq = z.$$ 

Since we know [4 or 5] that a one-tape machine can perform only regular computations with a bounded number of tape reversals, we conclude that $\mathcal{F}$ cannot determine whether $w_{q,z}$ is in $A_k$.

For the sake of clarity we give a more detailed description of the last conclusion: since in a bounded number of tape reversals a one-tape Turing machine can recognize only regular sets, we conclude that whatever $\mathcal{F}$ does in $k-1$ sweeps over the working tape can be done in one sweep after the input has been scanned. On the other hand the pattern recorded on the working tape, before $\mathcal{F}$ enters the blank part while scanning the segment

$$n_i + pq \rightarrow.$$
is the same for all inputs \( w_{q,z} \). Thus the one sweep of the working tape can be restricted just to scanning once the symbols written down after \( T \) enters the blank part of the working tape while scanning

\[ o_{i+pq}^{n_1+pq}, o_{q}^{n_2} \]

But then the writing and scanning could be combined, which implies that the writing can be omitted. This implies that the set

\[ \{ w_{q,z} \} \]

can be recognized with a finite amount of memory. This is a contradiction since \( \{ w_{q,z} \} \) is not a regular set. Thus \( A_{k} \) cannot be recognized with \( k-1 \) or fewer tape reversals, as was to be shown.

V. RELATION TO LANGUAGES

The preceding result shows that for every \( k, k = 1, 2, \ldots \) there exist sets of sequences which can be recognized with \( k \) tape reversals and cannot be recognized with a smaller number of reversals. We now relate this result to context-free and context-sensitive languages.
As stated before, \( R(n) = 0 \) yields only regular sets and thus we know that

\[ C_0 = \{ A \mid A \text{ is regular} \} \]

Since every set in \( C_1 \) (that is \( R(n) = 1 \)) can be recognized by a push-down-automaton, we conclude that \( C_1 \) contains only context-free languages [17]. As a matter of fact, every \( C \) with \( R(n) = 1 \) can be converted to a push-down automaton if it is forced after the tape reversal to overprint every scanned symbol of the working tape with a blank.

With \( R(n) = 2 \), a machine can recognize the set

\[ \{1^n 0^n 1^n \mid n = 1, 2, \ldots \} \]

and, since this set is not a context-free language, we see that \( C_2 \) contains languages which are not context-free.

On the other hand, for every \( k \) there is a context-free language which is in \( C_k \) and not in \( C_{k-1} \). To see this we just observe that the previously defined set \( A_k \) is a context-free language and then use the preceding theorem to show that it is not in \( C_{k-1} \).
Finally, we observe that the context-free language

\[ L = \{1^n \# 1^n \# \mid n = 1, 2, \ldots \}^* \]

is not contained in any \( C_k \).

Next we show that every \( A \) in \( C_n \) is a context-sensitive language.

**Corollary.** If \( \mathcal{F} \) operates with \( R(n) \leq n \) then \( T(\mathcal{F}) \) can be recognized by a linearly-bounded automaton and thus it is a context-sensitive language.

**Proof.** It can easily be shown that if

\[ T(\mathcal{F}) = A \]

and \( \mathcal{F} \) operates with \( R(n) \leq n \), then there exists a \( \mathcal{F}' \) in \( \mathcal{K}_{R(n)} \) which stops for all inputs and such that

\[ T(\mathcal{F}') = A. \]

From Lemma 2 we know that

\[ L(n) \leq C(R + n) \leq 2Cn. \]

Thus the memory of \( \mathcal{F}' \) is bounded by a linear function and therefore
is accepted by a linearly bounded automaton, as was to be shown.

VI. UNBOUNDED \( R(n) \) COMPUTATIONS.

In this section we study the complexity classes for unbounded \( R(n) \).

**Lemma 4.** If \( R(n) \) is a computable function then there exists a recursive set \( A \) which is not in \( C_{R(n)} \).

**Proof.** Elementary diagonal process.

Next we show that even for unbounded reversal functions \( R(n) \) there are sets of sequences which can be recognized with \( R(n) \) reversals but cannot be recognized with \( R(n) - 1 \) reversals.

**Theorem 3.** There exists an unbounded \( R(n) \) and a set \( A \) such that

\[
A \text{ is in } C_{R(n)} \text{ and } A \text{ is not in } C_{R(n)-1}.
\]

**Proof.** (This proof relies heavily on the use of crossing sequences and therefore familiarity with [4] or [5] is helpful in following the reasoning.)

Consider the set
\[ A = (0^{2^N} \cdot v_1^T \cdot v_2^T \cdot \ldots \cdot v_{N}^T) \mid N = 1, 2, \ldots \] 
\[ v_k \in \{0, 1\}^n \quad \text{and} \quad 1(v_k) = 2^N \] 

where

\[(x_1 \ x_2 \ \ldots \ x_p)^T = x_p \ x_{p-1} \ \ldots \ x_1 .\]

Observe that the length of \( w \) in \( A \) is given by

\[ 1(w) = 2^N + 1 + N2^N + 2N = (2N + 1)(2^N + 1) .\]

We now show that the set \( A \) is in \( \mathcal{C}_R(n) \) with

\[ R(n) = \min \{ N \mid n \in (2N + 1)(2^N + 1) \} .\]

To do this we first observe [4, 5] that with \( N \) tape reversals a one-tape machine can decide whether a segment on its tape has length \( 2^N \). This can be done as follows: the machine scans the segment and marks off the first, third, fifth, etc., unmarked tape squares, then reverses the tape and repeats the process going the other way. If on each sweep the last unmarked square (at the other end) of the tape segment is not marked off, then the number of unmarked tape squares at the start of this sweep.
was even. Thus, if on each sweep the last unmarked square is not crossed off and after \( N \) sweeps only one square is unmarked, we conclude that the length of the segment is \( 2^N \). This shows that with \( N \) tape reversals a machine can check whether the length of the segment is \( 2^N \).

To recognize the set \( A \) the machine \( \mathcal{T} \) combines two processes:

a) \( \mathcal{T} \) checks, by the above-described method, whether the length of the first segment is \( 2^N \);

b) \( \mathcal{T} \) checks whether the following pairs of sequences are mirror images and whether they have the same length as the first segment of zeros.

To visualize the details of this process assume that the working tape of \( \mathcal{T} \) is divided in an upper and lower track. At the start of the computation \( \mathcal{T} \) copies the segment of zeros and the segment \( w_{i1} \) on the upper track of the working tape and marks off on these segments the odd numbered squares. Then \( \mathcal{T} \) reverses its tape, checks the second part of \( w_{i1} \) for identity with the first and after that copies \( w_{i2} \) on the lower track of the working tape under the segment of zeros; at the same time \( \mathcal{T} \) also marks off the odd numbered unmarked tape squares of the
two segments on the upper track of the working tape. Now the process is repeated: \( v_{12} \) is checked for identity with the following segment, \( v_{13} \) is copied on the lower track under \( v_{11} \) and the odd-numbered unmarked squares on the upper track are marked off, etc.

If \( i \leq N \) sweeps on both segments of the upper track all but one tape square is unmarked we conclude that they have length \( 2^N \); and if all \( N \) segments \( v_{1j} \) written below them (had the same length and) are followed by their mirror images then the sequence is in \( A \). If at any time the above described process cannot be carried out the sequence is not in \( A \). Thus we see that the set \( A \) is accepted by a \( \mathcal{T} \) within \( R(n) \) tape reversal.

Next we show that \( A \) is not \( R(n)-1 \) recognizable. To see this assume that

\[
T(\mathcal{T}) = A
\]

and that \( \mathcal{T} \) operates with \( R(n)-1 \) tape reversal. Consider now for a large \( N \) the set of sequences

\[
A_N = \{ 0^2 \uparrow v_{11} \uparrow v_{11}^T \uparrow \ldots \uparrow v_{1N} \uparrow v_{1N}^T \uparrow e \mid 1(v_{1j}) = 2^N \}.
\]
Since \( \mathcal{F} \) performs only \( N-1 \) tape reversals for every sequence there is a segment \( v_{i_j} / v_{i_j}^T \) which \( \mathcal{F} \) scans without reversing its working tape. More than that, among the sequences in \( A_N \) is a sequence

\[
0^N \, w_{i_1} \, w_{i_1}^T \, \ldots \, w_{i_j} \, w_{i_j}^T \, \ldots \, w_{i_N} \, w_{i_N}^T \, e
\]
such that, no matter what binary sequence \( v \) of length \( 2^N \) is substituted for \( v_{i_j} \), the machine will scan the \( i_j \)-th segment, \( v \, \# \, v^T \), without reversing the working tape. To see that such a sequence exists we just have to check whether \( \mathcal{F} \) reverses itself on any of the possible first segments, \( w_{i_1} \, w_{i_1}^T \). If \( \mathcal{F} \) does not reverse its tape for any sequence we have found the desired segment. If it reverses its tape for

\[
v_{i_j} = v_1
\]

then consider what \( \mathcal{F} \) does on all possible \( w_{i_2} \, w_{i_2}^T \) in

\[
0^N \, w_1 \, w_1^T \, w_{i_2} \, w_{i_2}^T \, \ldots \, w_{i_N} \, w_{i_N}^T \, e
\]
Either we find the desired segment on which \( \mathcal{T} \) does not reverse the working tape or we find a value \( v_2 \neq v_2^T \) on which \( \mathcal{T} \) reverses its tape. Proceeding this way we either find among the first \( N-1 \) segments one on which \( \mathcal{T} \) does not reverse its tape or the \( N-1 \) tape reversals available to \( \mathcal{T} \) are used up before the last segment \( v_{1N} \neq v_{1N}^T \) is scanned and therefore the last segment is scanned without reversing the tape. Thus the desired sequence exists.

Assume that \( \mathcal{T} \) does not reverse the working tape on the \( i_j \)-th segment of the set of sequences.

\[
\{ 0^2^N \ 0 \ 0^T \ \ 0 \ 0^T \ \ ... \ \ 0 \ v_{1j} \ 0^T \ \ ... \ \ 0 \ v_N \ 0^T \ \ e \ |
\]

\[
l(v_{1j}) = 2^N \cdot v_{1j} \in (0+1)^* .
\]

When \( \mathcal{T} \) is started on any one of these sequences it writes a fixed pattern on the working tape before the input head starts scanning the segment \( v_{1j} \neq v_{1j}^T \), as shown by the instantaneous description which \( \mathcal{T} \) enters,

\[
s, 0^2^N \ 0 \ 0^T \ \ ... \ \ 0 \ v_{1j} \ 0^T \ \ ... \ \ 0 \ v_N \ 0^T \ e ,
\]

\[
- b_1 \ b_2 \ ... \ b_v \ + \ ... \ b_m - .
\]
The input has length

\[ n = (2N + 1) (2^N + 1) \quad \text{and} \quad R(n) < M. \]

Since \( \overline{f} \) stops we conclude by Lemma 2 that the length of the pattern, \( b_1 b_2 \ldots b_m \), is such that

\[ m \leq C(R^2 + nR + n) < C(N^2 + (2N+1) (2^N + 1) N + (2N+1) (2^N+1)) \]

and, therefore, for some \( C_1 > 0 \),

\[ m < C_1 N^2 2^{N+1}. \]

Recall now that \( \overline{f} \) does not reverse the working tape while it scans the segment

\[ w_{ij} \uparrow w_j^T \]

and, without loss of generality we can assume that the head on the working tape is moving to the right. Let us now consider where the head can be on the working tape while the input head scans the marker \( \theta \) of the segment \( w_{ij} \uparrow w_j^T \). There are

\[ 2^{2^N} \]

different segments \( w_{ij} \uparrow w_j^T \).
and the length of the working tape pattern is bounded by
\[ c_1 N^2 2^{N+1} \]. Then there exists at least one working tape square
\[ b_g \] which is scanned simultaneously with the marker \( \theta \) in
\[ w^j_i \theta w^T_i \] for at least
\[ 2^N / c_1 N^2 2^{N+1} \]
different segments \( w^j_i \theta w^T_i \). For large \( N \)
\[ 2^N / c_1 N^2 2^{N+1} > 2^{N/2} \]
and, therefore, the square \( b_g \) is scanned simultaneously with
the marker at least
\[ 2^{N/2} \] times.

This situation is illustrated by the two instantaneous
descriptions entered by \( \mathcal{F} \) when started on the two inputs \( \ldots \ldots \)
which we obtain by setting
\[ w^j_i = x \] and \[ w^j_i = y \].
\[ \begin{align*}
\text{a, } O \cdot 2^N & \quad \theta \vphantom{w_1'} \vphantom{w_1} w_1 \vphantom{w_1} w_1^T \vphantom{w_1} \ldots \vphantom{w_N} \theta \times \theta + x^T \vphantom{w_N} \ldots \vphantom{w_N} \theta w_N \vphantom{w_N} w_N^T \vphantom{w_N}. \\
\text{b, } b_1 \quad b_2 \ldots \quad b_{v-1} \quad b_v \ldots \quad b_{g-1} \quad b_g \ldots \quad b_m \quad 1 \\
\text{c, } O \cdot 2^N & \quad \theta \vphantom{w_1'} \vphantom{w_1} w_1 \vphantom{w_1} w_1^T \vphantom{w_1} \ldots \vphantom{w_N} \theta y \vphantom{w_N} \theta + y^T \vphantom{w_N} \ldots \vphantom{w_N} \theta w_N \vphantom{w_N} w_N^T \vphantom{w_N}. \\
\text{d, } b_1 \quad b_2 \ldots \quad b_{v-1} \quad b_v' \ldots \quad b_{g-1} \quad b_g \ldots \quad b_m \quad .
\end{align*} \]

After \( \mathcal{F} \) has scanned the segment \( w_{i,j} \neq w_{i,j}^T \), without reversing the working tape, it can perform no more than \( N - 1 \) tape reversals before stopping. Let us now count how many different input sequences \( \mathcal{F} \) can distinguish with this number of reversals. Consider the boundary between the working tape squares \( b_g \) and \( b_{g+1} \). Every time \( \mathcal{F} \) crosses this boundary the information carried across this boundary is completely specified by

a) the state of \( \mathcal{F} \),
b) the tape square \( \mathcal{F} \) is scanning on its input tape.

There are no more than

\[ |S| \cdot (2N+1) (2^N+1) \]

such possibilities. Since only \( N-1 \) reversals can be performed there can be no more than
\[ |S| \cdot (2N + 1) \cdot (2^N + 1) \cdot 1 \cdot N-1 \]

different "crossing sequences" on this boundary (see [4] or [5]). Since for large \( N \)

\[ |S| \cdot (2N + 1) \cdot (2^N + 1) \cdot N-1 < 2^{N/2} \]

we conclude that there are at least two different segments

\[ x \neq x^T \text{ and } y \neq y^T \]

such that:

a) \( \mathcal{F} \) reaches the instantaneous descriptions

\[
\begin{align*}
0^N & \uparrow v_1 \uparrow v_1^T \uparrow \ldots \uparrow x \uparrow x^T \uparrow \ldots \uparrow v_N \uparrow v_N^T \downarrow, \\
- b_1 & b_2 \ldots b_v \ldots b_g \uparrow b_{g+1} \ldots b_m \downarrow \\
0^N & \uparrow v_1 \uparrow v_1^T \uparrow \ldots \uparrow y \uparrow y^T \uparrow \ldots \uparrow v_N \uparrow v_N^T \downarrow, \\
- b_1 & b_2 \ldots b_v \ldots b_g \uparrow b_{g+1} \ldots b_m \downarrow \\
\end{align*}
\]

b) after that \( \mathcal{F} \) generates identical crossing sequences on the \( b_g, b_{g+1} \) boundary of both working tapes. But then all the information carried across this boundary is identical. Thus,
since both input sequences are accepted we conclude that the input sequence

$$\nu^1 = 0^2 N \# \nu_1 \# \nu_1^T \# \ldots \# x \# y^T \# \ldots \# \nu_N \# \nu_N^T$$

must also be accepted, since on the $b_g$, $b_{g+1}$ boundary $\mathcal{G}$ generates the same behavior for tapes with the $x$ and the $y$ segment.

Since the input $\nu^1$ is not in $A$ and is accepted by $\mathcal{G}$ we conclude that $A$ is not acceptable with $R(n)-1$ tape reversals. This completes the proof.

The previous proof can easily be extended to other slowly growing tape reversal functions $R(n)$.

To make this generalization we define reversal functions which play in those considerations the same role as real-time functions, constructable functions and sweep functions in previous work on computational complexity [2,3,5].

Definition A monotone increasing function $F(n)$ from integers into integers is a reversal function if, and only if there exists $\mathcal{G}$ in $M_{F^{-1}(n)}$ which accepts the set

$$\{0^{F(n)} | n = 1, 2, \ldots \}$$
and performs exactly \( R(n) = [F^{-1}(n)] \) tape reversals on the input

\[ 0_{F(n)}^n. \]

For example,

\[ F(n) = 2^k \quad , \quad k = 1, 2, 3, \ldots \]

\[ F(n) = 2^{k[\log_2 n]} \quad , \quad k = 1, 2, 3, \ldots \]

are reversal functions. So far the reversal functions have not been systematically investigated.

**Theorem 4.** If \( F(n) \) is a reversal function such that

\[
\lim_{N \to \infty} \frac{(N F(N))^N}{2^F(N)} = 0
\]

then the set

\[ A = \{0_{F(N)}^N \theta w_1 \theta w_1^T \theta \ldots \theta w_N \theta w_N^T \mid N = 1, 2, \ldots \} \]

\[ 1(w_i) = F(N) \}

is recognizable with
\[ R(n) = \min \{ N \mid n \geq (2^N + 1) (F(N) + 1) \} \]

and not with \( R(n) = 1 \).

**Proof.** The same as the proof of Theorem 3 when we replace \( 2^N \) by \( F(N) \) and observe that the limit ensures that we have the right number of crossing sequences to carry through the same reasoning as in the proof of Theorem 3.

The above results showed that even for unbounded slowly growing \( R(n) \) the increase of \( R(n) \) to \( R(n) + 1 \) extended the computational capability of our machines. It is interesting to note that for fast growing \( R(n) \) the situation is quite different. It has been shown by M. Blum [15] that if

\[ \lim_{n \to \infty} \frac{n}{R(n)} = 0 \]

then

\[ C_{R(n)} = C \left[ \frac{R(n)}{2} \right] \]

Thus for these computations \( R(n) \) can be decreased by any constant factor.

For the sake of completeness we include the next result which can be used to construct infinitely many different tape reversal complexity classes.
Theorem 5. For any computable \( f(n) \) there exists a \( R(n) \),
\[
R(n) > f(n) \quad , \quad n = 1, 2, \ldots
\]
and a set \( A \) such that \( A \) is in \( C_{R(n)} \) and not in \( C_{R(n)} \) if
\[
\lim_{n \to \infty} \frac{R(n)}{R(n)^2} = 0.
\]

Proof. By a simple diagonal process. (For the basic ideas see the proof of Theorem 9 in [2]).

In conclusion we state some undecidability results.

Theorem 6. It is recursively undecidable whether

a) \( \mathcal{F} \) is in \( M_{R(n)} \) for some \( R(n) < = \);

b) \( T(\mathcal{F}) \) is in \( C_{R(n)} \) for some \( R(n) < = \);

d) \( T(\mathcal{F}) \) is in \( C_{R(n)} \).

there exists a computable \( R(n) \) such even if it is known that \( \mathcal{F} \) is in \( M_{R(n)} \) and \( T(\mathcal{F}) \) in \( C_{R(n)} \) it is recursively undecidable whether

\[
c) \quad \mathcal{F} \quad \text{is in} \quad M_{R(n)}
\]
\[
d) \quad T(\mathcal{F}) \quad \text{is in} \quad C_{R(n)}.
\]

Proof. By standard techniques it can be shown that the decidability of any of these problems implies that the halting
problem for Turing machines is decidable. Thus these problems
are undecidable. (To contrast these results with results for
finite-turn push-down automata see [18]).

Discussion

The results in this paper show that the tape reversal
bounded computations yield an infinite hierarchy of complexity
classes. Furthermore, these complexity classes differ from
time-limited and tape-limited complexity hierarchies in that
for finite and slowly growing tape reversal bounds just one
additional reversal increased the computational power of the
automaton. On the other hand, for fast growing tape reversal
bounds even the doubling of the reversals does not increase the
computational power of the automaton.
REFERENCES


5. J. Hartmanis, "Computational Complexity of One-Tape Turing Machine Computations." To be published in the J. of the ACM.


14. J. Hartmanis and H. Shank, "On the Recognition of Primes by Automata," To be published by the J of the ACM.

15. M. Blum, private communication.

