JOURNAL OF COMPUTER AND SYSTEM SCIENCES 11, 56–67 (1975)

Hierarchies of Turing Machines with Restricted Tape Alphabet Size*

OSCAR H. IBARRA AND SARTAJ K. SAHNI

Department of Computer Science, University of Minnesota, Minneapolis, Minnesota 55455

Received November 3, 1973; revised March 29, 1974

It is shown that for any real constants $b > a \geqslant 0$, multitape Turing machines operating in space $L_1(n) = \lceil bn^r \rceil$ can accept more sets than those operating in space $L_2(n) = \lceil an^r \rceil$ provided the number of work tapes and tape alphabet size are held fixed. It is also shown that Turing machines with $k + 1$ work tapes are more powerful than those with k work tapes if the tape alphabet size and the amount of work space are held constant.

INTRODUCTION

Results on tape and time complexity of algorithms involving Turing machine models are usually derived with the assumption that the devices have finite but arbitrary work tape alphabets. A consequence of this is that constant factors on tape-bounds and most time-bounds do not affect the complexity classes they define [1, 5], nor does addition of more work tapes to tape-bounded Turing machines change their computing power [1, 2]. It seems natural to study tape and time complexity measures with the restriction that the Turing machines operate with the same work tape alphabet. A recent paper of Seiferas, Fischer, and Meyer [4] includes such a study. In particular, they prove that for tape bounds $L(n)$ satisfying certain properties, the class of sets accepted by single-tape Turing machines with m work tape symbols operating in $L(n)$ space is properly contained in the class of sets accepted by single-tape Turing machines with M_m work tape symbols operating in $L(n)$ space for some $M_m > m$. In [3] it is shown that for tape bounds of the form $L(n) = n^r$, M_m is, in fact, equal to $m + 1$. This paper continues the study started in [3].

Section 1 introduces the notation we shall be using. A theorem in [3] is also restated here as the proofs in Sections 2 and 3 will rely heavily on it. In Section 2, we study the effect of increasing the work space by a constant factor while holding the number of symbols and work tapes constant. We are able to prove the existence of a refined

* This work was supported in part by the National Science Foundation under Grant GJ-35614.

hierarchy for all $L(n)$ for which Theorem 1.1 is true. The main result of this section is Theorem 2.2, which states that for polynomial tape, $\lceil an^r \rceil$ tape complexity class is properly contained in $\lceil bn^r \rceil$ tape complexity class for any real constants $b > a \geqslant 0$.

The effect of increasing the number of work tapes while keeping the number of symbols and the work space fixed is studied in Section 3. We show that Turing machines with $k + 1$ work tapes are better than those with k work tapes. Theorem 3.1 also generalizes Theorem 1.1 to the case of $k > 1$ work tapes.

Finally, in Section 4 we present a simple proof of a generally known, but, to our knowledge, unpublished, result that two-way nondeterministic (deterministic) finite automata with $n + 1$ states are more powerful than those with n states.

1. Preliminaries

In this paper, we restrict our study to multitape Turing machines with a read-only input tape (which is delimited on both ends by markers) and $k \geqslant 1$ one-way infinite read–write work tapes. For a detailed description of these machines, see Hopcroft and Ullman [2].

A multitape Turing machine (TM) which has at most m symbols in its work tape alphabet and has the property that every input that is accepted is accepted by some sequence of moves which uses no more than $L(n)$ tape squares in each work tape (where n is the length of the input) is referred to as an m-symbol $L(n)$ tape-bounded TM. (The special symbol, B, which stands for a blank, is always assumed to be in the work tape alphabet.)

If A is a TM, then $L(A)$ will denote the set of input tapes accepted by A. By NSPACE($L(n)$, m, k) (DSPACE($L(n)$, m, k)) we shall mean the class of sets accepted by all nondeterministic (deterministic) m-symbol $L(n)$-tape bounded TM's with k work tapes. Often, when $k = 1$, we shall omit the third script, i.e. NSPACE($L(n)$, m) $=$ NSPACE($L(n)$, m, 1).

The following theorem is adopted from [3].

Theorem 1.1. *For any integers $r \geqslant 1$ and $m \geqslant 1$,*

(i) $\text{NSPACE}(n^r, m) \not\subseteq \text{NSPACE}(n^r, m + 1)$,

(ii) $\text{DSPACE}(n^r, m) \not\subseteq \text{DSPACE}(n^r, m + 1)$.

Remark. The proof technique of [3] and Lemma 2.1 can be used to prove the following more general form of Theorem 1.1. For any rational number $c > 0$ and integers $r, m \geqslant 1$, $\text{NSPACE}(\lceil cn^r \rceil, m) \not\subseteq \text{NSPACE}(\lceil cn^r \rceil, m + 1)$ and $\text{DSPACE}(\lceil cn^r \rceil, m) \not\subseteq \text{DSPACE}(\lceil cn^r \rceil, m + 1)$.

2. Single Tape TM Space Hierarchies

In this section we study the effect on single tape TM's of increasing space while keeping the number of symbols constant. We begin by showing that increasing the tape bound by a constant number of tape squares does not increase the power (in terms of being able to accept more sets) of the TM. Then in Theorems 2.1 and 2.2 we show that increases by a constant factor result in the acceptance of new sets.

Lemma 2.1. *For any positive integer c*, NSPACE($L(n), m$) = NSPACE($L(n) + c, m$) *and* DSPACE($L(n), m$) = DSPACE($L(n) + c, m$).

Proof. Let L be in NSPACE($L(n) + c, m$) and A be an m-symbol $L(n) + c$ tape-bounded nondeterministic TM. We shall construct an m-symbol $L(n)$ tape-bounded nondeterministic TM B accepting L.

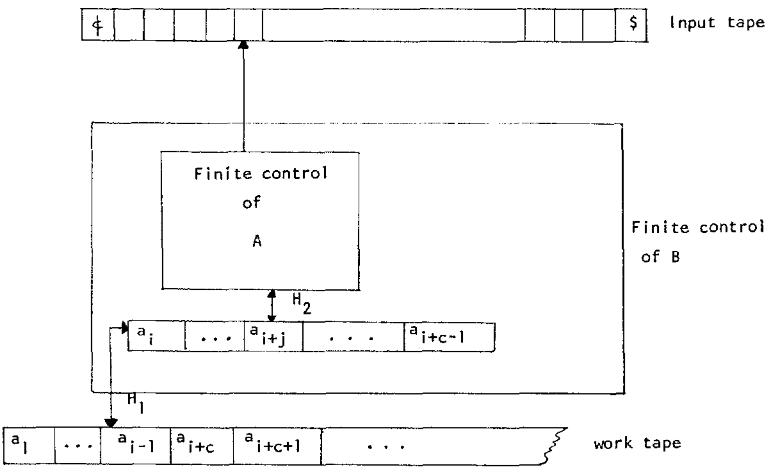

Fig. 1. TM B.

Figure 1 illustrates the construction of B. The finite control of B consists of the finite control of A and a buffer to hold any string of length c in m symbols. B partitions its work space into three segments: $a_1 \cdots a_{i-1}$, $a_i \cdots a_{i+j} \cdots a_{i+c-1}$, and $a_{i+c}a_{i+c+1} \cdots$, with only the segment $a_i \cdots a_{i+j} \cdots a_{i+c-1}$ (which is in the buffer) accessible to the finite control of A. For convenience, we can think of the finite control of A as having a read–write head, H_2, which can communicate with the buffer.

We now describe the operation of B.

(1) B starts by simulating A using the buffer as A's work space. Thus, B need not move its work tape head, H_1. (Initially, all squares of the buffer contain blanks.)

B continues the simulation of A using only the buffer as long as H_2 remains within the confines of the buffer. This situation is depicted in Fig. 1. The work tape and buffer configurations shown in Fig. 1 represent the work tape configuration of A as having the string $a_1 \cdots a_{i-1}a_ia_{i+1} \cdots a_{i+j} \cdots a_{i+c-1}a_{i+c}a_{i+c+1} \cdots$ with its work tape head on symbol a_{i+j}.

(2) If during the simulation, H_2 attempts to leave the right end of the buffer, B must perform the following steps before it can continue simulating A (refer to Fig. 1):

(a) B stores a_i in its finite control and shifts all symbols in the buffer one square left.

(b) B moves its work tape head to the square occupied by a_{i+c}, stores a_{i+c} in its finite control, and rewrites a_{i+c} by a_i.

(c) B writes a_{i+c} on the right end of the buffer and positions H_2 on this symbol.

The result of steps (a)–(c) is shown in Fig. 2.

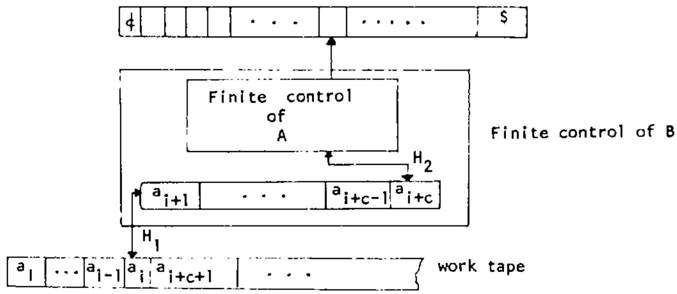

FIG. 2. TM B.

(3) A process similar to (2) is performed by B if H_2 attempts to leave the left end of the buffer.

It should be clear from the description above that B accepts L and that B is an m-symbol $L(n)$-tape bounded nondeterministic TM. Thus, L is in NSPACE($L(n)$, m, k).

The proof that DSPACE($L(n)$, m) = DSPACE($L(n) + c$, m) is identical.

The next two theorems give a relationship between hierarchies based on the tape alphabet size and hierarchies based on the amount of work space. A related result for single-tape Turing machines appears in [4].

THEOREM 2.1. *For any integer* $m \geq 2$ *and* $\epsilon \geq \log_m(m+1) - 1$,

(i) $\text{NSPACE}(L(n), m) \not\subseteq \text{NSPACE}(L(n), m+1)$
$\Rightarrow \text{NSPACE}(L(n), m) \not\subseteq \text{NSPACE}(\lceil(1+\epsilon)L(n)\rceil, m)$,

(ii) $\text{DSPACE}(L(n), m) \not\subseteq \text{DSPACE}(L(n), m + 1)$
$\Rightarrow \text{DSPACE}(L(n), m) \not\subseteq \text{DSPACE}(\lceil(1 + \epsilon) L(n)\rceil, m)$.

Proof. We show that for $m \geqslant 2$ and $\epsilon \geqslant \log_m(m + 1) - 1$,

$$\text{DSPACE}(L(n), m + 1) \subseteq \text{DSPACE}(\lceil(1 + \epsilon) L(n)\rceil, m).$$

Let L_1 be in $\text{DSPACE}(L(n), m + 1)$. Then there is an $(m + 1)$-symbol $L(n)$ tape-bounded deterministic TM T_1 such that $L(T_1) = L_1$. We construct an m-symbol deterministic TM, T_2, such that $L(T_2) = L(T_1) = L_1$ and T_2 is $\lceil(1 + \epsilon) L(n)\rceil$ tape-bounded. T_2 uses p squares of its work tape to encode q squares of T_1's work tape. In order to do this without loss of information we require $m^p \geqslant (m + 1)^q$ or $p/q \geqslant \log_m(m + 1)$.

T_2 has a buffer of size q in its finite control. Each element of this buffer can hold any of the $m + 1$ symbols of T_1. T_2 begins with a blank buffer and blank work tape. It simulates T_1 with the help of its buffer. If T_1 wants to access a tape square not currently represented in T_2's buffer, T_2 encodes the q-square buffer into a string of length p in m symbols and writes it on its work tape. It then reads from its work tape the next p symbols, decodes them into a string of length q and $m + 1$ symbols and so refills its buffer. In this way T_2 is able to simulate the behavior of T_1. The amount of space T_2 needs is then bounded by $\lceil (p/q) L(n)\rceil$. But $p/q \geqslant \log_m(m + 1)$. So, L_1 is in $\text{DSPACE}(\lceil(1 + \epsilon) L(n)\rceil, m)$ for $\epsilon \geqslant \log_m(m + 1) - 1$. Hence,

$$\text{DSPACE}(L(n), m + 1) \subseteq \text{DSPACE}(\lceil(1 + \epsilon) L(n)\rceil, m).$$

(ii) immediately follows from this.

We note that if T_1 had been nondeterministic, then the construction outlined above would result in a nondeterministic T_2 such that $L(T_2) = L(T_1)$ and T_2 would have a tape bound $\lceil(1 + \epsilon) L(n)\rceil$. This leads to (i).

COROLLARY 2.1. *Let $L(n) = \lceil cn^r\rceil$, where $c > 0$ is a rational number and $r \geqslant 1$, $m \geqslant 2$ integers. Then* $\text{NSPACE}(L(n), m) \not\subseteq \text{NSPACE}(\lceil(1 + \epsilon) L(n)\rceil, m)$ *and*

$$\text{DSPACE}(L(n), m) \not\subseteq \text{DSPACE}(\lceil(1 + \epsilon) L(n)\rceil, m) \qquad \textit{for all } \epsilon \geqslant \log_m(m + 1) - 1.$$

Proof. Immediate consequence of Theorem 2.1 and remark following Theorem 1.1.

We remark that for $m = 1$, the work tape is of no use since this means that only the symbol B (i.e. the blank) can occur on the tape and so only regular sets are recognizable. So, there is no hierarchy for $m = 1$.

While Corollary 2.1 establishes a refined hierarchy for polynomial tape, we can prove the existence of a much sharper hierarchy. To do this, we need the following lemma.

LEMMA 2.2. *For rational numbers $c, d > 0$ and integers $m \geqslant 2$ and k such that $kc \geqslant 1$,*

(i) $\mathrm{DSPACE}(\lceil cn^r \rceil, m) \subseteq \mathrm{DSPACE}(\lceil dn^r \rceil, m)$
$\Rightarrow \mathrm{DSPACE}(\lceil ck^r n^r \rceil, m) \subseteq \mathrm{DSPACE}(\lceil dk^r n^r \rceil, m)$,

(ii) $\mathrm{NSPACE}(\lceil cn^r \rceil, m) \subseteq \mathrm{NSPACE}(\lceil dn^r \rceil, m)$
$\Rightarrow \mathrm{NSPACE}(\lceil ck^r n^r \rceil, m) \not\subseteq \mathrm{NSPACE}(\lceil dk^r n^r \rceil, m)$.

Proof. We prove (i). The proof of (ii) is similar.

Let L be in $\mathrm{DSPACE}(\lceil ck^r n^r \rceil, m)$. Then there exists a deterministic TM (DTM), A, which is $\lceil ck^r n^r \rceil$ tape-bounded, and $L(A) = L$. The proof proceeds by constructing a DTM, B, such that $L(B) = L'$ (to be defined below) and B is $\lceil cn^r \rceil$ tape-bounded. This implies a DTM, C, such that $L(C) = L'$ and C is $\lceil dn^r \rceil$ tape-bounded. Using C, we show how to construct a DTM, D, which accepts L and which is $\lceil dk^r n^r \rceil$ tape-bounded, thus proving (i).

(a) *Construction of* DTM, *B*, *from A*

Define $L' = \{xd^i \mid x \text{ in } L, \mid x \mid = n, \mid xd^i \mid = kn\}$, where d is a new symbol not occurring in any string of L, and $\mid x \mid =$ length of x. The DTM B operates as follows.

1. B checks that the input is of the right form. B can do this using only two symbols within space $\lceil ck^r n^r \rceil$ since $\lceil ck^r n^r \rceil \geqslant n$.

2. B simulates the actions of A on x.

It is clear that B accepts L'. Now $c \mid xd^i \mid^r = ck^r n^r$. Since the input is of length $N = kn$, B is $\lceil cN^r \rceil$ tape-bounded.

By hypothesis, there is a DTM, C, such that $L(C) = L'$ and C is $\lceil dN^r \rceil$ tape-bounded with m symbols.

(b) *Construction of* DTM, *D*, *from C*

The DTM, D, will simulate C. Simulating C requires $\lceil dN^r \rceil = \lceil dk^r n^r \rceil$ tape squares. Thus, D has enough space on its work tape. (Note that D's input head can easily simulate the behavior of C's input head on segment d^i without making use of its work tape.)

While Corollary 2.1 establishes a refined hierarchy for polynomial tape, the next theorem proves the existence of a much sharper hierarchy.

THEOREM 2.2. *For all integers $r \geqslant 1$ and $m \geqslant 2$ and real constants a and b, $b > a \geqslant 0$,*

(i) $\mathrm{DSPACE}(\lceil an^r \rceil, m) \not\subset \mathrm{DSPACE}(\lceil bn^r \rceil, m)$,

(ii) $\mathrm{NSPACE}(\lceil an^r \rceil, m) \not\subset \mathrm{NSPACE}(\lceil bn^r \rceil, m)$.

Proof. We prove (i). (ii) follows from a similar argument. We make use of Lemma 2.2 which establishes that for rational numbers c, $d > 0$ and integer k such that $kc \geqslant 1$:

$$\text{DSPACE}(\lceil cn^r \rceil, m) \subsetneq \text{DSPACE}(\lceil dn^r \rceil, m) \Rightarrow \text{DSPACE}(\lceil ck^r n^r \rceil, m) \subsetneq \text{DSPACE}(\lceil dk^r n^r \rceil, m).$$

Suppose that $\text{DSPACE}(\lceil bn^r \rceil, m) \subsetneq \text{DSPACE}(\lceil an^r \rceil, m)$ for some real constants $b > a \geqslant 0$. Then we can find two rational numbers c and d such that $b \geqslant d > c > a \geqslant 0$. This implies that

(1) $\text{DSPACE}(\lceil dn^r \rceil, m) \subsetneq \text{DSPACE}(\lceil cn^r \rceil, m)$.

Let $c = i/j$, $d = p/q$, where i, j, p, and q are positive integers. Then (1) becomes

(2) $\text{DSPACE}(\lceil pn^r/q \rceil, m) \subsetneq \text{DSPACE}(\lceil in^r/j \rceil, m)$.

By Lemma 2.2 with $k = (qj)^r$, we get

(3) $\text{DSPACE}(\lceil pj(qj)^{r-1}n^r \rceil, m) \subsetneq \text{DSPACE}(\lceil iq(qj)^{r-1}n^r \rceil, m)$.

From (3) we conclude that

(4) $\text{DSPACE}((M+1)n^r, m) \subsetneq \text{DSPACE}(Mn^r, m)$, where $M = iq(qj)^{r-1} > 0$.

Again, from Lemma 2.2, we get

(5) $\text{DSPACE}((M+1)k^r n^r, m) \subsetneq \text{DSPACE}(Mk^r n^r, m)$ for all $k \geqslant 1$.

(6) Clearly, there exists $k_0 \geqslant 0$ such that $(M+1)k^r \geqslant M(k+1)^r$ for all $k \geqslant k_0$. From (5) we have

$$\begin{aligned} \text{DSPACE}((M+1)k_0^r n^r, m) &\subsetneq \text{DSPACE}(Mk_0^r n^r, m), \\ \text{DSPACE}((M+1)(k_0+1)^r n^r, m) &\subsetneq \text{DSPACE}(M(k_0+1)^r n^r, m), \\ &\vdots \\ \text{DSPACE}((M+1)k^r n^r, m) &\subsetneq \text{DSPACE}(Mk^r n^r, m). \end{aligned}$$

From (6) and the above sequence of inclusions, it follows that

$$\text{DSPACE}((M+1)k^r n^r, m) \subsetneq \text{DSPACE}(Mk_0^r n^r, m) \qquad \text{for all } k \geqslant k_0.$$

However, from Corollary 2.1 we conclude that there is some integer K such that $\text{DSPACE}(Mk_0^r n^r, m) \not\subseteq \text{DSPACE}((M+1)K^r n^r, m)$, contradicting (6).

3. Multitape Turing Machines

In this section the effect of varying the parameters m and k in the classes $\text{NSPACE}(L(n), m, k)$ and $\text{DSPACE}(L(n), m, k)$ is investigated. Some of the results are stated only for NSPACE($L(n)$, m, k), although it should be clear that the proofs used apply just as well to DSPACE($L(n)$, m, k).

To prove the main result of this section, we need the following two lemmas.

LEMMA 3.1. *Let k, l, and m be positive integers such that $m^k \geqslant l$. Then,*

$$\text{NSPACE}(L(n), l, 1) \subseteq \text{NSPACE}(L(n), m, k).$$

Proof. Obvious.

LEMMA 3.2. *For all positive integers m and k,*

$$\text{NSPACE}(L(n), m, k) \subseteq \text{NSPACE}(L(n), m^k + 1, 1).$$

Proof. By Lemma 2.1, it is sufficient to prove that

$$\text{NSPACE}(L(n), m, k) \subseteq \text{NSPACE}(L(n) + k, m^k + 1, 1).$$

So let L be in NSPACE($L(n)$, m, k) and A be an m-symbol $L(n)$ tape-bounded nondeterministic TM (NTM) with k work tapes accepting L. We shall first construct an $(m^k + k)$-symbol $L(n)$ tape-bounded NTM B with a single work tape accepting L. B will have k tracks on its work tape, one track for each work tape of A. A track symbol is in the form of a k-tuple $(a_1, a_2, \ldots, a_k)$, where each a_i is a symbol in the work tape alphabet of A. Since the work tape alphabet of A has at most m symbols, B needs no more than m^k symbols to represent all k-tuples. To simulate A properly, B must be able to encode the positions of the k work tape heads of A. For this purpose, B uses k extra symbols, $H_1, H_2, \ldots, H_k$. B places each of these symbols on its work

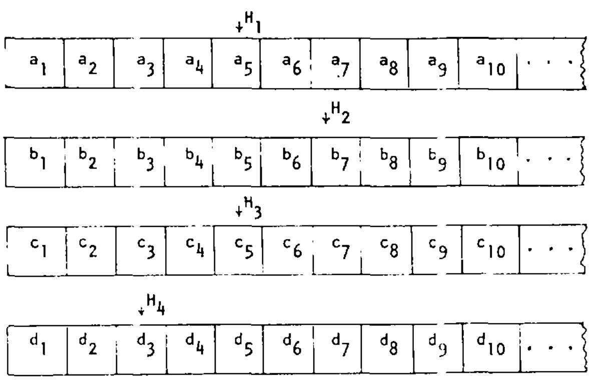

FIGURE 3

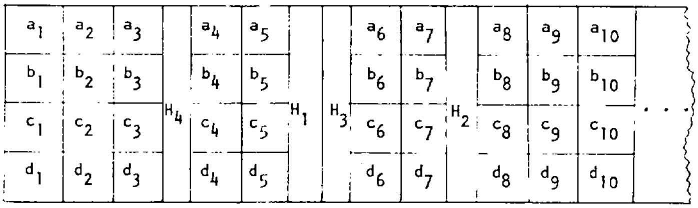

FIGURE 4

tape as follows. H_i is placed immediately to the right of the square containing the track symbol whose ith coordinate is the symbol scanned by the ith work tape head of A. (Of course, if the squares immediately to the right of the track symbol already contain symbols $H_{i_1}, H_{i_2}, ..., H_{i_l}$, then H_i is placed to the right of these symbols.) For example, if $k = 4$ and the configurations of the four work tapes of A are as shown in Fig. 3, then B's work tape would have the configuration shown in Fig. 4.

We may assume that at the start of every move simulation of A, the work tape head of B is scanning one of the H_i's. We may also assume that B keeps track of the relative positions of the symbols $H_1, H_2, ..., H_k$ in its finite control. With these assumptions, it is clear that B can retrieve the symbols scanned by all the work tape heads of A. Moreover, B can easily update the locations of the symbols $H_1, H_2, ..., H_k$ to correspond to the new positions of the work tape heads of A. It follows that B accepts L and that B is $L(n) + k$ tape-bounded with $m^k + k$ work tape symbols.

Now, since B keeps track of the relative positions of the symbols $H_1, H_2, ..., H_k$, they need not be distinct. In fact, one symbol is sufficient. Thus, we can modify B to an equivalent NTM B' which is $L(n) + k$ tape-bounded and has only $m^k + 1$ work tape symbols.

We are now ready to prove the main result of this section.

THEOREM 3.1. *Let* NSPACE($L(n), m, 1$) $\not\subseteq$ NSPACE($L(n), m + 1, 1$) *for each* $m \geqslant 1$. *Then we have*

(1) NSPACE($L(n), m, k$) $\not\subseteq$ NSPACE($L(n), m + 1, k$) *for all* $m \geqslant 1$ *and* $k \geqslant 1$.

(2) NSPACE($L(n), m, k$) $\not\subseteq$ NSPACE($L(n), m, k + 1$) *for all* $m \geqslant 2$ *and* $k \geqslant 1$.

(3) *For each* $m \geqslant 1$ *and* $i \geqslant 1$, *there exists* $k_0 \geqslant 1$ *such that*

$$\text{NSPACE}(L(n), m, k + i) \not\subseteq \text{NSPACE}(L(n), m + 1, k) \qquad \textit{for all } k \geqslant k_0.$$

(4) *For each* $k \geqslant 1$ *and* $i \geqslant 1$, *there exists* $m_0 \geqslant 1$ *such that*

$$\text{NSPACE}(L(n), mi, k) \not\subseteq \text{NSPACE}(L(n), m, k + 1) \qquad \textit{for all } m \geqslant m_0.$$

Proof.

(1) By Lemma 3.2, NSPACE($L(n), m, k$) $\subseteq$ NSPACE($L(n), m^k + 1, 1$). By hypothesis, NSPACE($L(n), m^k + 1, 1$) $\not\subseteq$ NSPACE($L(n), m^k + 2, 1$). Since $(m + 1)^k \geqslant m^k + 2$ for all $m \geqslant 1$ and $k \geqslant 2$, it follows from Lemma 3.1 that

$$\text{NSPACE}(L(n), m, k) \not\subseteq \text{NSPACE}(L(n), m + 1, k).$$

The case $k = 1$ is solved in [3].

(2) The proof is similar to that of (1) by noting that $m^{k+1} \geqslant m^k + 2$ for all $m \geqslant 2$ and $k \geqslant 1$.

(3) NSPACE($L(n), m, k + i$) $\subseteq$ NSPACE($L(n), m^{(k+i)} + 1, 1$) by Lemma 3.2. Now, NSPACE($L(n), m^{(k+i)} + 1, 1$) $\not\subseteq$ NSPACE($L(n), m^{(k+i)} + 2, 1$) by hypothesis. Clearly, there exists $k_0 \geqslant 1$ such that $(m + 1)^k \geqslant m^{(k+i)} + 2$ for all $k \geqslant k_0$. It follows from Lemma 3.1 that NSPACE($L(n), m, k + i$) $\not\subseteq$ NSPACE($L(n), m + 1, k$) for all $k \geqslant k_0$.

(4) The proof is similar to that of (3) by noting that for each $k \geqslant 1$ and $i \geqslant 1$, there exists $m_0 \geqslant 1$ such that $m^{k+1} \geqslant (mi)^k + 2$ for all $m \geqslant m_0$.

Using Theorems 1.1 and 3.1, we have

COROLLARY 3.1. *For any positive integer r,*

(1) NSPACE(n^r, m, k) $\not\subseteq$ NSPACE($n^r, m + 1, k$) *for all* $m \geqslant 1$ *and* $k \geqslant 1$.

(2) NSPACE(n^r, m, k) $\not\subseteq$ NSPACE($n^r, m, k + 1$) *for all* $m \geqslant 2$ *and* $k \geqslant 1$.

(3) *For each* $m \geqslant 1$ *and* $i \geqslant 1$, *there exists* $k_0 \geqslant 1$ *such that*

$$\text{NSPACE}(n^r, m, k + i) \not\subseteq \text{NSPACE}(n^r, m + 1, k) \qquad \textit{for all } k \geqslant k_0.$$

(4) *For each* $k \geqslant 1$ *and* $i \geqslant 1$, *there exists* $m_0 \geqslant 1$, *such that*

$$\text{NSPACE}(n^r, mi, k) \not\subseteq \text{NSPACE}(n^r, m, k + 1) \qquad \textit{for all } m \geqslant m_0.$$

The following is an analog of Theorem 2.1 for multitape TM's.

THEOREM 3.2. *For the case where the* TM's *have* $k > 1$ *work tapes and* $m \geqslant 2$ *work tape symbols, the following hold.*

(i) DSPACE($L(n), m$) $\not\subseteq$ DSPACE($L(n), m + 1$)
$\Rightarrow$ DSPACE($L(n), m, k$) $\not\subseteq$ DSPACE($\lceil \log_{m^k}(m^k + 2) L(n) \rceil, m, k$),

(ii) NSPACE($L(n), m$) $\not\subseteq$ NSPACE($L(n), m + 1$)
$\Rightarrow$ NSPACE($L(n), m, k$) $\not\subseteq$ NSPACE($\lceil \log_{m^k}(m^k + 2) L(n) \rceil, m, k$).

Proof. The proof follows the same pattern as that of Theorem 2.1 with the aid of Theorem 3.1. Thus, for (i) the following strategy is used.

$$\mathrm{DSPACE}(L(n), m, k) \subseteq \mathrm{DSPACE}(L(n), m^k + 1) \not\subseteq \mathrm{DSPACE}(L(n), m^k + 2) \subseteq \mathrm{DSPACE}(\lceil \log_{m^k}(m^k + 2)\, L(n) \rceil, m, k).$$

COROLLARY 3.2. *Let* $L(n) = \lceil cn^r \rceil$, *where* $c > 0$ *is a rational number and* $r \geqslant 1$, $m \geqslant 2, k \geqslant 2$ *integers. Then* $\mathrm{DSPACE}(L(n), m, k) \not\subseteq \mathrm{DSPACE}(\lceil (1 + \epsilon) L(n) \rceil, m, k)$ *and* $\mathrm{NSPACE}(L(n), m, k) \not\subseteq \mathrm{NSPACE}(\lceil (1 + \epsilon) L(n) \rceil, m, k)$ *for all* $\epsilon \geqslant \log_{m^k}(m^k + 2) - 1$.

The proof of Theorem 2.2 can be extended to prove the following more general result.

COROLLARY 3.3. *For all integers* $r, k \geqslant 1$, *and* $m \geqslant 2$ *and real constants* a *and* b, $b > a \geqslant 0$,

(i) $\mathrm{DSPACE}(\lceil an^r \rceil, m, k) \not\subseteq \mathrm{DSPACE}(\lceil bn^r \rceil, m, k)$,

(ii) $\mathrm{NSPACE}(\lceil an^r \rceil, m, k) \not\subseteq \mathrm{NSPACE}(\lceil bn^r \rceil, m, k)$.

4. FINITE AUTOMATA

We conclude our results with a simple proof of a generally known, but, to our knowledge, unpublished, result concerning two-way nondeterministic (deterministic) finite automata. We refer the reader to [2] for the definitions of these devices.

THEOREM 4.1. *For each* $n \geqslant 1$, *the class of sets accepted by two-way nondeterministic (deterministic) finite automata with* $n + 1$ *states properly contains the class of sets accepted by two-way nondeterministic (deterministic) finite automata with* n *states.*

Proof. Containment is obvious. We prove proper containment for the deterministic case, the argument being similar for the nondeterministic case.

For each $n \geqslant 1$, let $L_n = \{a^n\}$. L_n is accepted by a two-way deterministic finite automaton A with $n + 1$ states, $q_1, q_2, \ldots, q_{n+1}$, where q_1 is the initial state, q_{n+1} is the only accepting state, and the state transition is defined as follows. For $1 \leqslant i \leqslant n$, A, in state q_i with input a, moves right and enters state q_{i+1}. A, in state q_{n+1} with input a, moves left and remains in state q_{n+1}. The only string that will cause A to enter state q_{n+1} upon leaving the right end of the string starting from the initial state (on the left end of the string) is a^n. Thus, A accepts L_n.

Suppose that L_n is accepted by some two-way deterministic finite automaton B. Consider the input a^n to B. For $1 \leqslant j \leqslant n$, let q_{i_j} be the state of B the last time it scans the jth symbol of a^n before moving right. Let $q_{i_{n+1}}$ be the state B enters upon leaving the right end of a^n. Clearly, $q_{i_{n+1}}$ is an accepting state. We claim that $q_{i_s} \neq q_{i_t}$ for all $1 \leqslant s, t \leqslant n + 1$ with $s \neq t$.

Suppose $q_{i_s} = q_{i_t}$ for some $s < t$. Then the string $a^n(a^{t-s})^k$ would also be accepted by B for all $k \geqslant 1$. This contradicts the fact that B accepts only the string a^n. It follows that $q_{i_1}, q_{i_2}, \ldots, q_{i_{n+1}}$ are distinct. Thus, B must have at least $n + 1$ states.

For the case of one-way finite automata one can easily show that deterministic (nondeterministic) $(n + 1)$-state finite automata are more powerful than n-state deterministic (nondeterministic) finite automata.

References

1. J. Hartmanis and R. E. Stearns, On the computational complexity of algorithms, *Trans. Amer. Math. Soc.* **117** (1965), 285–306.
2. J. E. Hopcroft and J. D. Ullman, "Formal Languages and Their Relation to Automata," Addison–Wesley, Reading, Mass., 1969.
3. O. H. Ibarra, A hierarchy theorem for polynomial-space recognition, *SIAM J. Comput.* **3** (1974), 184–187.
4. J. I. Seiferas, M. J. Fischer, and A. R. Meyer, Refinements of the hierarchies of time and tape complexities, presented at the 14th Annual Symposium on Switching and Automata Theory, Iowa City, Iowa, October, 1973.
5. R. E. Stearns, J. Hartmanis, and P. M. Lewis II, Hierarchies of memory limited computations, IEEE Conference Record of Sixth Annual Symposium on Switching Circuit Theory and Logical Design, pp. 179–190, Ann Arbor, Michigan, 1965.