INFORMATION AND CONTROL 54, 217–227 (1982)

On Reversal-Bounded Counter Machines and on Pushdown Automata with a Bound on the Size of the Pushdown Store

PAVOL DURIS

Computer Center, Slovak Academy of Science,
84235 Bratislava, Czechoslovakia

AND

ZVI GALIL*

Computer Science Department, Columbia University,
New York, New York 10027 and Tel-Aviv University, Tel-Aviv, Israel

The two main results of the paper are: (1) proving a fine hierarchy of reversal-bounded counter machine languages; and (2) showing that a tape is better than a pushdown store for two-way machines, in the case where their size is sublinear.

INTRODUCTION

If M is a two-way (multi) counter machine, we denote by $L(M)$ the language accepted by M. For a function $f(n)$, a two-way counter machine M is $f(n)$ *reversal bounded* if for every string $w \in L(M)$, there is an accepting computation of M on w using at most $O(f(|w|))$ reversals, where $|w|$ is the length of w, and a reversal is a change from pushing to popping or vice versa by one of the counters.

In [1] Chan proved the following theorem (Theorem 7.2): "The following bounds define strictly increasing reversal complexity classes for two-way deterministic counter machines: 0, 1, $\log n$, and n."

Our first main result is refining Chan's hierarchy: We say that a function $f(n)$ is *reversal constructible* if there is a deterministic two-way counter machine which, on input of length n, can create a counter of length $f(n)$, with all counters making at most $O(f(n))$ reversals in the process.

* Research supported by National Science Foundation Grant MCS-8303139.

0019-9958/82 $2.00

THEOREM 1. *Let* $f_1(n)$, $f_2(n)$ *be two integer-valued functions such that* $\lim_{n\to\infty} \inf(f_1(n)/f_2(n)) = 0$, $f_2(n) \leqslant (n-1)/2$ *for all* n, *and* $f_2(n)$ *is reversal-constructible. Then the language*

$$L = \{xy \# yx^R \mid |xy \# yx^R| = n, n > 0, x \in \{0, 1\}^*, |x| \leqslant f_2(n), y \in \{2\}^*\}$$

is recognized by an $f_2(n)$ *reversal-bounded two-way deterministic counter machine, but it cannot be recognized by any* $f_1(n)$ *reversal-bounded two-way deterministic counter machine.*

COROLLARY 1. *For every pair of integers* $0 \leqslant k_1 < k_2$ *(resp. for every pair of real numbers* $0 \leqslant r_1 < r_2 \leqslant 1$*), there is a language which is recognized by a* $(\log n)^{k_2}$ *(resp.* n^{r_2}*) reversal-bounded two-way deterministic counter machine, but it cannot be recognized by any* $(\log n)^{k_1}$ *(resp.* n^{r_1}*) reversal-bounded two-way deterministic counter machine.*

COROLLARY 2. *For every function* f *with* $1 \leqslant f(n)$ *and* $\lim_{n\to\infty} \inf(f(n)/n) = 0$, *there is a language accepted by an* $f(n)$ *reversal-bounded two-way* nondeterministic *counter machine and not be any* $f(n)$ *reversal-bounded two-way* deterministic *counter machine.*

We define 2DPDA($f(n)$) to be the class of languages accepted by two-way deterministic pushdown automata (2dpda's) whose pushdown stores are never longer than $f(n)$ on inputs of size n. We denote by DSPACE($f(n)$) the class of languages accepted by deterministic $f(n)$ space-bounded Turing machines. It is well known that for every f, $\text{2DPDA}(f(n)) \subseteq \text{2DPDA}(n) =$ 2DPDA. (The latter is the class of languages accepted by unrestricted 2dpda's.) A well-known open problem is whether $\text{2DPDA} \subsetneq \text{DSPACE}(n)$ (see Galil, 1977), or in our notation whether $\text{2DPDA}(n) \subsetneq \text{DSPACE}(n)$. Stated differently, this problem is actually whether a linear tape is better than a linear pushdown store for two-way machines. We still cannot solve the problem, but we can solve an easier version of it.

THEOREM 2. *For* f *that satisfies* $f(n) = o(n)$ *and*

$$\lim \sup(f(n)/\log \log n) > 0, \ \text{2DPDA}(f(n)) \subsetneq \text{DSPACE}(f(n)).$$

Remark. $\text{2DPDA}(f(n)) = \text{DSPACE}(f(n)) =$ regular languages, for $f(n) = o(\log \log n)$. Theorem 2 follows as a corollary from Theorem 3.

THEOREM 3. *If a language* L *over a one-symbol alphabet is in* 2DPDA($f(n)$) *and* $f(n) = o(n)$, *then* L *is regular.*

The proof of Theorem 2 is immediate (given Theorem 3) using the known result that there exist nonregular languages over a one-symbol alphabet in DSPACE($\log \log n$) (Freedman and Ladner, 1975). Theorem 3 does not hold for languages over a *two*-symbol alphabet. We define a nonregular language L_1 and prove

THEOREM 4. *L_1 is in* 2DPDA($\log\log n$).

THE PROOFS

The proof of Theorem 1 is similar to the proof of our main result in Duris and Galil (1982). Figures 1–5 in that article can be used to understand the proof here. The y axis in these figures should be understood as representing the contents of one of the counters. We define an internal computation of a counter machine A on a triple (x, y, z) as a computation on input xyz that starts at one of the end symbols of y, ends at a symbol out of y, during which A scans y and each counter is either always empty or always nonempty (Fig. 2 in Duris and Galil, 1982). We define functions f_y (Fig. 3 in Duris and Galil, 1982) that describe completely the internal computations on (x, y, z). This is possible because the length of internal computations is bounded. (Figure 4 in Duris and Galil, 1982 shows the three possible contradictions one gets if one assumes that an internal computation can be longer than a certain bound.) Using a counting argument we derive two strings u and v with $f_u = f_v$, and consequently show that A hardly distinguishes between u and v. For every x and z, there is an internal computation between two configurations of A on (x, u, z) if and only if there is an internal computation between the same configurations on (x, v, z). The latter fact follows from the fact that $f_u = f_v$ by the ability to "copy" the two computations implied by the definitions of f_u and f_v (Fig. 5 in Duris and Galil). Finally, we will be able to fool the machine by replacing an occurrence of u by v.

Proof of Theorem 1. Let M be a two-way deterministic counter machine with k counters and let Q be the set of internal states of M. A configuration of M is a $(k+2)$-tuple $(q, h, s_1,..., s_k)$, where $q \in Q$, h is the position of the input head of M and s_i is the length of the ith counter of M. (Note that there are $(n+2)$ positions of the input head of M on input of size n, where position 0 (resp. $(n+1)$) is the position of the left (resp. right) endmarker.) If x is an input of M and C and C' are configurations of M on x, we denote by $C \vdash_x C'$ the fact that M goes in one step from C to C'. If $C = (q, h, s_1,..., s_k)$ is a configuration of M, we define $pr_0(C) := q$, $pr_1(C) := h$, and $pr_j(C) := s_{j-1}$ for $j = 2, 3,..., k+1$. For a set S we denote by $|S|$ the size of S, and for a string x we denote by $|x|$ the length of x.

DEFINITION 1. Let $C_0, C_1,..., C_r$ be a sequence of configurations of M;

let x, y, z be strings, where $y \in \{0, 1\}^*$ and $|y| \geqslant 1$. We say that the sequence $C_0, C_1, \ldots, C_r$ is an internal computation of M from C_0 to C_r on the triple (x, y, z) if (i)–(iv) hold.

(i) $C_0 \vdash_{xyz} C_1 \vdash_{xyz} \cdots \vdash_{xyz} C_r$,

(ii) $|x| + 1 \leqslant pr_1(C_i) \leqslant |xy|$ for $i = 0, 1, \ldots, r-1$,

(iii) $pr_1(C_0) \in \{|x| + 1, |xy|\}$ and $pr_1(C_r) \in \{|x|, |xy| + 1\}$,

(iv) for $j = 2, 3, \ldots, k+1$, either $pr_j(C_i) > 0$ for $i = 0, 1, \ldots, r$ or $pr_j(C_i) = 0$ for $i = 0, 1, \ldots, r$.

Let $C_0, C_1, \ldots, C_r$ be a sequence of configurations of M. By $\min_j(C_0, C_1, \ldots, C_r)$ (resp. $\max_j(C_0, C_1, \ldots, C_r)$) we denote the minimum (resp. maximum) number of the sequence

$$0, pr_j(C_1) - pr_j(C_0), pr_j(C_2) - pr_j(C_0), \ldots, pr_j(C_r) - pr_j(C_0)$$

for $j = 2, 3, \ldots, k+1$.

We choose an integer m such that

$$[2\,|Q|(2\,|Q|\,m + 2)^k(|Q|\,m + 1)^k + 1]^{2^{k+1} \cdot |Q|} < 2^m. \tag{1}$$

DEFINITION 2. Let $\bar{x}$ and $\bar{z}$ be two arbitrary but fixed strings. Let $S_1 = \mathscr{C}$ and $S_2 = \mathscr{C} \times \mathbb{Z} \underbrace{\times \cdots \times}_{k \text{ times}} \mathbb{Z}$, where $\mathscr{C}$ is the set of all configurations of M and $\mathbb{Z}$ is the set of all integers. For each string y in $\{0, 1\}^m$ we define a partial function $f_y: S_1 \to S_2$ as follows. Let C_0 be an arbitrary configuration of M. If the sequence $C_0, C_1, \ldots, C_r$ of configurations of M is an internal computation of M from C_0 to C_r on the triple $(\bar{x}, y, \bar{z})$, and moreover, $pr_j(C_0) \in \{0, |Q|\,m + 1\}$ for $j = 2, 3, \ldots, k+1$, then $f_y(C_0) = (C_r, -\min_2(C_0, \ldots, C_r), -\min_3(C_0, \ldots, C_r), \ldots, -\min_{k+1}(C_0, \ldots, C_r))$ and if there is no such computation, then $f_y(C_0)$ is undefined.

Note that since M is deterministic, f_y is indeed a partial function.

LEMMA 1. *Let x, x', y, z, z' be five strings, where y is in $\{0, 1\}^m$. Let $C_0, C_1, \ldots, C_r$ be an internal computation of M from C_0 to C_r on (x, y, z) and let C'_0 be a configuration of M such that $pr_0(C'_0) = pr_0(C_0)$, $pr_1(C'_0) = pr_1(C_0) - |x| + |x'|$, and $pr_j(C'_0) = 0$ if $pr_j(C_0) = 0$ and $pr_j(C'_0) > -\min_j(C_0, \ldots, C_r)$ if $pr_j(C_0) > 0$ for $j = 2, 3, \ldots, k+1$. Then the sequence of configurations of M $C'_0, C'_1, \ldots, C'_r$, where $C'_0 \vdash_{x'yz'} C'_1 \vdash_{x'yz'} \cdots \vdash_{x'yz'} C'_r$, is an internal computation of M from C'_0 to C'_r on (x', y, z'), and moreover,*

$$pr_0(C'_i) = pr_0(C_i), pr_1(C'_i) = pr_1(C_i) - |x| + |x'|$$

and

$$pr_j(C'_i) = pr_j(C_i) - pr_j(C_0) + pr_j(C'_0) \text{ for } i = 0, 1, \ldots, r, j = 2, 3, \ldots, k+1.$$

The proof follows by induction from the fact that M moves the input head and decreases (resp. increases) the counters during the computation $C_0', C_1',..., C_r$ exactly as it does during the computation $C_0, C_1,..., C_r$, because the input head of M scans only the string y during the computation $C_0, C_1,..., C_{r-1}$ (see (ii) of Definition 1) and the inequality $pr_j(C_0') > -\min_j(C_0,..., C_r)$ guarantees that the jth counter is never empty during the computation $C_0',..., C_r'$.

LEMMA 2. *There are two different strings u, v in $\{0, 1\}^m$ such that for every pair of strings x, z and every pair of configurations of M C_0, C_r, there is an internal computation of M from C_0 to C_r on (x, u, z) if and only if there is an internal computation of M from C_0 to C_r on (x, v, z).*

Proof. Let $C_0, C_1,..., C_r$ be an internal computation of M from C_0 to C_r on (x, y, z), where y is a string in $\{0, 1\}^m$. We first show that for every j, $2 \leqslant j \leqslant k+1$, if $pr_j(C_0) > 0$, then

$$0 \leqslant -\min_j(C_0,..., C_{r-1}) \leqslant |Q|\, m - 1$$

and (2)

$$0 \leqslant \max_j(C_0,..., C_{r-1}) \leqslant |Q|\, m - 1.$$

We show only the first half of (2). The other half is similar. We assume to the contrary that for some j, $2 \leqslant j \leqslant k+1$, $pr_j(C_0) > 0$ and $-\min_j(C_0,..., C_{r-1}) \geqslant |Q|\, m$. We consider the sequence of pairs $(pr_0(C_0), pr_1(C_0))$, $(pr_0(C_1), pr_1(C_1)),..., (pr_0(C_{|Q|m}), pr_1(C_{|Q|m}))$. Note that $r-1 \geqslant |Q|\, m$, because the jth counter of M must decrease from $pr_j(C_0)$ by at least $|Q|\, m$ during $(r-1)$ computation steps. The number of all different pairs of the form $(pr_0(C_l), pr_1(C_l))$ is at most $|Q|\, m$ (since $|y| = m$). Therefore, there are two indices s and t, $s < t$, such that

$$pr_0(C_s) = pr_0(C_t) \qquad \text{and} \qquad pr_1(C_s) = pr_1(C_t). \tag{3}$$

By (iv) of Definition 1, the sequence of pairs $(pr_0(C_i), pr_1(C_i))$, $i = 0,..., r$ is periodic, and by (3) the size of the period is at most r. But this implies that $|x| + 1 \leqslant pr_1(C_r) \leqslant |xy|$—a contradiction to (iii) of Definition 1.

Since $pr_j(C_{r-1}) - 1 \leqslant pr_j(C_r) \leqslant pr_j(C_{r-1}) + 1$ for $j = 2, 3,..., k+1$, then by (2) we have that for every j, $2 \leqslant j \leqslant k+1$, if $pr_j(C_0) > 0$, then

$$0 \leqslant -\min_j(C_0,..., C_r) \leqslant |Q|\, m$$

and (4)

$$0 \leqslant \max_j(C_0,..., C_r) \leqslant |Q|\, m.$$

If $C_0, C_1, \ldots, C_r$ are the configurations from Definition 2, then $pr_1(C_0) \in \{0, |Q|\,m+1\}$, and by (4) and by (iv) of Definition 1,

$$0 \leqslant pr_j(C_r) \leqslant 2\,|Q|\,m+1 \qquad \text{for } every \quad j = 2, 3, \ldots, k+1. \tag{5}$$

By Definitions 1, 2, by (4) and (5), each f_y is a partial function from S_1' into S_2', where $S_1' = Q \times \{|\bar{x}|+1, |\bar{x}|+m\} \times \{0, |Q|\,m+1\}^k$ and $S_2' = (Q \times \{|\bar{x}|, |\bar{x}|+m+1\} \times \{0, 1, \ldots, 2\,|Q|\,m+1\}^k) \times \{0, 1, \ldots, |Q|\,m\}^k$. The cardinality of the set of all partial functions from S_1' into S_2' is $[2\,|Q|(2\,|Q|\,m+2)^k(|Q|\,m+1)^k+1]^{2^{k+1}\cdot|Q|}$. On the other hand, there are 2^m strings in $\{0, 1\}^m$. By (1) there are two different strings u and v in $\{0, 1\}^m$ with $f_u = f_v$.

Now, let $C_0, C_1, \ldots, C_r$ be an internal computation of M from C_0 to C_r on (x, u, z). By (iv) of Definition 1, for $2 \leqslant j \leqslant k+1$, if $pr_j(C_0) > 0$, then

$$pr_j(C_0) > -\min_j(C_0, \ldots, C_r), \tag{6}$$

because the jth counter does not become empty during the computation $C_0, C_1, \ldots, C_r$. We consider the sequence of configurations of M $\bar{C}_0, \bar{C}_1, \ldots, \bar{C}_r$, where $\bar{C}_0 \vdash_{\bar{x}u\bar{z}} \bar{C}_1 \vdash_{\bar{x}u\bar{z}} \cdots \vdash_{\bar{x}u\bar{z}} \bar{C}_r$, ($\bar{x}, \bar{z}$ are the strings from Definition 2), and

$$pr_0(\bar{C}_0) = pr_0(C_0), \qquad pr_1(\bar{C}_0) = pr_1(C_0) - |x| + |\bar{x}|,$$

and for $j = 2, 3, \ldots, k+1$,

$$pr_j(\bar{C}_0) = \begin{cases} |Q|\,m+1 & \text{if } \; pr_j(C_0) > 0, \\ 0 & \text{if } \; pr_j(C_0) = 0. \end{cases} \tag{7}$$

By (7) and (4), for $2 \leqslant j \leqslant k+1$, if $pr_j(C_0) > 0$, then $pr_j(\bar{C}_0) = |Q|\,m+1 > -\min_j(C_0, \ldots, C_r)$, and thus by (7) and by Lemma 1, the sequence $\bar{C}_0, \bar{C}_1, \ldots, \bar{C}_r$ is an internal computation of M from $\bar{C}_0$ to $\bar{C}_r$ on $(\bar{x}, u, \bar{z})$; and moreover,

$$pr_0(\bar{C}_i) = pr_0(C_i), \qquad pr_1(\bar{C}_i) = pr_1(C_i) - |x| + |\bar{x}|$$

and (8)

$$pr_j(\bar{C}_i) = pr_j(C_i) - pr_j(C_0) + pr_j(\bar{C}_0) \qquad \text{for} \quad i = 0, 1, \ldots, r, j = 2, 3, \ldots, k+1.$$

Since $f_u = f_v$, there is an internal computation $\tilde{C}_0, \tilde{C}_1, \ldots, \tilde{C}_s$ of M from $\tilde{C}_0 = \bar{C}_0$ to $\tilde{C}_s = \bar{C}_r$ on $(\bar{x}, v, \bar{z})$; and moreover,

$$\min_j(\tilde{C}_0, \ldots, \tilde{C}_s) = \min_j(\bar{C}_0, \ldots, \bar{C}_r) \qquad \text{for} \quad j = 2, 3, \ldots, k+1. \tag{9}$$

By (8),

$$\min_j(C_0, \ldots, C_r) = \min_j(\bar{C}_0, \ldots, \bar{C}_r) \qquad \text{for} \quad j = 2, 3, \ldots, k+1. \tag{10}$$

We consider the configurations $C'_0 = C_0, C'_1, \ldots, C'_s$, where $C'_0 \vdash_{xvz} C'_1 \vdash_{xvz} \cdots \vdash_{xvz} C'_s$. Since $\tilde{C}_0 = \bar{C}_0$, by (7), (6), (10), and (9) we have

$$pr_0(C_0) = pr_0(\tilde{C}_0), \qquad pr_1(C_0) = pr_1(\tilde{C}_0) - |\bar{x}| + |x|,$$

and

$$pr_j(C_0) = 0 \quad \text{if} \quad pr_j(\tilde{C}_0) = 0 \quad \text{and} \quad pr_j(C_0) > -\min_j(\tilde{C}_0, \ldots, \tilde{C}_s) \tag{11}$$

$$\text{if} \quad pr_j(\tilde{C}_0) > 0 \quad \text{for} \quad j = 2, 3, \ldots, k+1.$$

Hence, by Lemma 1, the sequence $C'_0, C'_1, \ldots, C'_s$ is an internal computation of M from $C'_0 = C_0$ to C'_s on (x, v, z); and moreover,

$$pr_0(C'_i) = pr_0(\tilde{C}_i), \qquad pr_1(C'_i) = pr_1(\tilde{C}_i) - |\bar{x}| + |x|$$

and (12)

$$pr_j(C'_i) = pr_j(\tilde{C}_i) - pr_j(\tilde{C}_0) + pr_j(C'_0) \qquad \text{for} \quad i = 0, 1, \ldots, s, j = 2, 3, \ldots, k+1.$$

But $\tilde{C}_s = \bar{C}_r$, $\tilde{C}_0 = \bar{C}_0$ and $C'_0 = C_0$, and by (8) and (12), we have $pr_j(C'_s) = pr_j(C_r)$ for $j = 0, 1, \ldots, k+1$, i.e., $C'_s = C_r$, and therefore, $C_0 = C'_0, C'_1, \ldots, C'_s = C_r$ is the internal computation of M from C_0 to C_r on (x, v, z). ∎

We now complete the proof of Theorem 1. We assume to the contrary that M is $f_1(n)$ reversal bounded and accepts L. This implies that M accepts every string $w \in L$ using at most $df_1(|w|)$ reversals for some constant $d > 0$. Let u, v be the strings from Lemma 2. Note that $|u| = |v| = m$. Since $\lim_{n\to\infty} \inf(f_1(n)/f_2(n)) = 0$, there is an integer n_0 such that $m(df_1(n_0) + k + 1) \leqslant f_2(n_0)$. Let $g = df_1(n_0) + k + 1$ and let $C_0, C_1, \ldots, C_f$ be the accepting computation of M on the string $w = x_1 x_2 \cdots x_g y \# y x_g^R x_{g-1}^R \cdots x_1^R$ in L, where $|w| = n_0$, $y \in \{2\}^*$ and each $x_i \in \{u, v\}$. Without loss of generality we assume that M scans the left endmarker of the input tape at C_0 and at C_f. For $j = 1, 2, \ldots, k$, let p_j be the number of the configurations C_i, $0 \leqslant i \leqslant f$, at which the jth counter of M is increased from zero or decreased to zero. Clearly, $\sum_{j=1}^k p_j \leqslant$ number of reversals $+ k \leqslant df_1(n_0) + k$ and therefore, $\sum_{j=1}^k p_j < g$. This implies that there is an index h, $1 \leqslant h \leqslant g$, such that if x_h is scanned by M at step i, $0 \leqslant i \leqslant f-1$, then no counter is increased from zero or decreased to zero at step $i+1$. Let $C_{i_1}, C_{i_2}, \ldots, C_{i_t}$ be all the configurations at which the input head of M leaves or enters the substring x_h. Consider the string $w' = x_1 x_2 \cdots x_{h-1} x'_h x_{h+1} \cdots x_g y \# y x_g^R \cdots x_h^R \cdots x_1^R$, where x'_h is u (resp. v) if x_h is v (resp. u). We derive a contradiction by showing that M accepts also w' ($w' \notin L$): Let $C_{i_0} = C_0$ and $C_{i_{t+1}} = C_f$. It suffices to show that there is a computation of M from C_{i_l} to $C_{i_{l+1}}$ on w' for $l = 0, 1, \ldots, t$. If l is even, then the computation from C_{i_l} to $C_{i_{l+1}}$ on w' is identical to the computation from C_{i_l} to $C_{i_{l+1}}$ on w, because the input head does not scan the substring x_h during the latter. If l is odd, then there is an internal computation of M from C_{i_l} to $C_{i_{l+1}}$ on $(x_1 \cdots x_{h-1}, x_h, x_{h+1} \cdots x_g y \# y x_g^R \cdots x_1^R)$, by the choice of x_h. By Lemma 2, there is also a computation of M from C_{i_l} to $C_{i_{l+1}}$ on w'. ∎

Proof of Corollary 1. Chan, 1981, showed that the functions $\lceil \log n \rceil^k$ and $\lceil n^{1/p} \rceil^q$ (for integers $k, p > q \geqslant 1$) are reversal constructible. ∎

Proof of Corollary 2. The language $L' = \{x \# x' \mid x' \neq x^R, x, x' \in \{0, 1\}^*\}$ is recognized by a one reversal-bounded one-way nondeterministic counter machine. If there were an $f(n)$ reversal-bounded two-way deterministic counter machine M_1 recognizing L', then there would be such a machine M_2 recognizing $\{x \# x^R \mid x \in \{0, 1\}^*\}$, because these deterministic machines (with reversal-constructible $f(n)$), are closed under complement. But M_2 cannot exist by Theorem 1. (In this case $f_1(n) = f(n)$ and $f_2(n) = (n-1)/2$.) ∎

Proof of Theorem 3. Let A be a 2dpda with a set Q of internal states and with a set Γ of stack symbols. By $ls(a^n)$ we denote the maximum length of stack used by A on the input a^n. We define two constants

$$p = |Q|\,|\Gamma|^{|Q|\,|\Gamma| + 2}, \qquad k = 1/(3p), \tag{13}$$

and prove

LEMMA 3. *There is an $n_0 = n_0(p)$, such that for $n > n_0$, if A accepts a^n with $ls(a^n) < kn$, then A must accept $a^{n'}$ with $ls(a^{n'}) = ls(a^n)$, where $n' = n - p!$*

Now, assume $L \subseteq \{a\}^*$ is accepted by a 2dpda A whose pushdown store is never longer than $f(n) = o(n)$ on a^n. Choose $n_1 \geqslant n_0$ such that for all $n > n_1$ $f(n) < kn$. If $a^n \in L$ and $n > n_1$, then by Lemma 3 there is $n' < n$ such that $a^{n'} \in L$ and $ls(a^{n'}) = ls(a^n)$. Consequently, $\max_{a^n \in L} ls(a^n) = \max_{a^n \in L, n \leqslant n_1} ls(a^n) = \text{constant}$. Hence, L is regular because its pushdown store can be simulated by the finite state control.

Proof of Lemma 3. A configuration of 2dpda A is a triple (q, z, i), where $q \in Q$, $z \in \Gamma^*$ is the string in the stack and i is the position of the head on the input tape. If $C = (q, z, i)$ is a configuration of A, then we define $pr_0(C) := q$, $pr_1(C) := z$ and $pr_2(C) := i$. We denote by $[z]_l$, the suffix of z of size l. $[z]_1$ is the symbol at the top of the stack. Without loss of generality we assume that A accepts only when its input head scans the left endmarker. As before we use the notation $C \vdash_x C'$ if A goes in one step from C to C' on input x.

A computation segment of A on input x is a sequence of configurations $C_0, \ldots, C_m$ such that $C_0 \vdash_x C_1 \vdash_x \cdots \vdash_x C_m$ and A scans an endmarker in C_0

and in C_m but not in C_i for $0 < i < m$. The lemma follows from the claim below by an induction on the number of computation segments in the computation of A on a^n.

CLAIM. Assume $C_0, \ldots, C_m$ is a computation segment of A on a^n. Then there is a computation segment $C'_0, \ldots, C'_{m'}$ of A on $a^{n'}$ such that:

(i) $pr_0(C'_0) = pr_0(C_0)$, $pr_1(C'_0) = pr_1(C_0)$,

(ii) $pr_0(C'_{m'}) = pr_0(C_m)$, $pr_1(C'_{m'}) = pr_1(C_m)$,

(iii) *in* C_0 *and* C'_0, *A scans the same endmarker,*

(iv) *in* C_m *and* $C'_{m'}$, *A scans the same endmarker,*

(v) $\max_{0 \leqslant i \leqslant m'}\{|pr_1(C'_i)|\} = \max_{0 \leqslant i \leqslant m}\{|pr_1(C_i)|\}$.

Proof. Without loss of generality we assume that $pr_2(C_0) = 0$. First, assume that there is no index h, $1 \leqslant h \leqslant m-1$, such that $n/3 + p \leqslant pr_2(C_h)$. For n large enough $n/3 + p < n'$, and $C'_i = C_i$ for $i = 0, \ldots, m$ and $m' = m$ will do. So we can assume that there is such an index h and we choose a minimal such h. Note that $h \geqslant n/3 + p$. There must be an index t, $1 \leqslant t \leqslant h - p$, such that $|pr_1(C_{t+r})| \geqslant |pr_1(C_t)|$ for every $r = 1, \ldots, p$. Otherwise, for every p steps there must be a decrease in the size of the stack and the size of the stack decreases eventually by $h/p \geqslant n/(3p) = kn$—a contradiction $(ls(a^n) < kn)$. We choose a minimal such t.

There are two cases left:

Case 1. $|pr_1(C_{t+r})| - |pr_1(C_t)| \leqslant |Q|\,|\Gamma|$ for every $r = 1, 2, \ldots, p$. Then there are two indices i, j, $t \leqslant i < j \leqslant t + p$, such that $pr_0(C_i) = pr_0(C_j)$ and $[pr_1(C_i)]_{l_i} = [pr_1(C_j)]_{l_j}$, where $l_s = 1 + |pr_1(C_s)| - |pr_1(C_t)|$, because $l_s \leqslant l \equiv |Q|\,|\Gamma| + 1$, the number of all strings over Γ with length at most l is at most $|\Gamma|^{|Q|\,|\Gamma|+2}$ and $p + 1 > |Q| \cdot |\Gamma|^{|Q|\,|\Gamma|+2}$. If $pr_2(C_i) > pr_2(C_j)$ (resp. $pr_2(C_i) < pr_2(C_j)$), then A periodically approaches the left endmarker ¢ (resp. right endmarker \$) with a period of size at most p and simultaneously the stack is in a loop; see Fig. 1a (resp. 1b). Therefore, for sufficiently large n there are

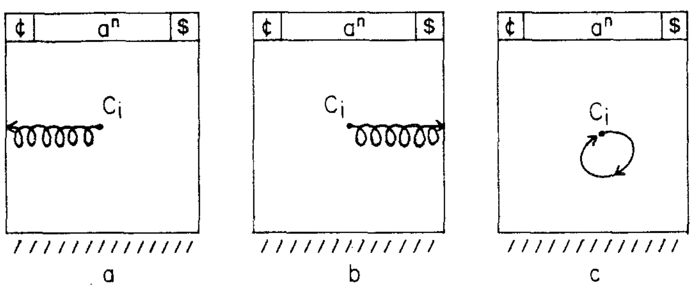

FIG. 1. The three subcases of Case 1.

configurations $C'_0, \ldots, C'_m$, with the desired properties. If $pr_2(C_i) = pr_2(C_j)$, then A is in a loop; see Fig. 1c. But then, it is impossible for A to scan an endmarker at C_m for the first time after C_0—a contradiction.

Case 2. There is an index r, $1 \leqslant r \leqslant p$, such that $|pr_1(C_{t+r})| - |pr_1(C_t)| \geqslant |Q||\Gamma| + 1$. Let r be such a minimal index. For $j = 0, 1, \ldots, |Q||\Gamma|$, let i_j, $t \leqslant i_j \leqslant t + r$ be the maximal index with $|pr_1(C_{i_j})| = |pr_1(C_t)| + j$. Obviously, there are two indices i_u and i_v, $t \leqslant i_u < i_v \leqslant t + r$, such that $pr_0(C_{i_u}) = pr_0(C_{i_v})$ and $[pr_1(C_{i_u})]_1 = [pr_1(C_{i_v})]_1$, because the number of the configurations C_{i_j} is $|Q||\Gamma| + 1$. This means that if $pr_2(C_{i_u}) > pr_2(C_{i_v})$ (resp. $pr_2(C_{i_u}) < pr_2(C_{i_v})$, resp. $pr_2(C_{i_u}) = pr_2(C_{i_v})$), then the stack periodically increases and simultaneously the input tape head periodically approaches the left endmarker (resp. the right endmarker, resp. the input tape head is in a loop); see Figs. 2a (resp. 2b, resp. 2c). So in all three cases the stack periodically increases (with a period of size at most $|Q||\Gamma| \leqslant p$) during at least $n/3$ steps (by the choice of h), and therefore A must use a stack of length at least $n/(3p) = kn$—a contradiction. ∎

We now define the language L_1 of Theorem 4. Let $a, b, 0, 1$ be four different symbols. We define a homomorphism h as follows: $h(a) = 0$, $h(b) = 1$, $h(0) = h(1) =$ empty string. Then

$$L_1 = \{w_1 \# w_2 \cdots w_{2^{2^n}} \mid n \geqslant 0,\ w_i = x_1 y_1 x_2 y_2 \cdots x_{2^n} y_{2^n}$$
$$\text{for every } i = 1, 2, \ldots, 2^n, \text{ where } y_1 < \cdots < y_{2^n}, \text{ every } y_j \in \{0, 1\}^n,$$
$$\text{every } x_j \in \{a, b\}, \text{ and } h(w_1) < h(w_2) < \cdots < h(w_{2^{2^n}})\}.$$

By $y_i < y_j$ we mean that the binary number represented by y_i is smaller than the one represented by y_j. Note that $y_1 = \underbrace{00 \cdots 0}_{n}$, $y_2 = \underbrace{00 \cdots 01}_{n}, \ldots$, $y_{2^n} = \underbrace{11 \cdots 111}_{n}$, and $w_1 = ay_1 ay_2 \cdots ay_{2^n}$, $w_2 = ay_1 ay_2 \cdots ay_{2^n - 1} by_{2^n}, \ldots$, $w_{2^{2^n}} = by_1 by_2 \cdots by_{2^n}$.

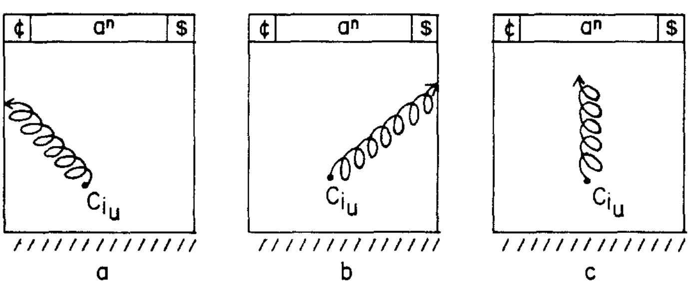

FIG. 2. The three subcases of Case 2.

We leave the details of the proof of Theorem 4 to the reader. We note only that the 2dpda has to be constructed with some care so that its stack is never longer than log log n also for strings not in L_1.

REFERENCES

CHAN, T.-H. (1981), Reversal complexity of counter machines, *in* "Proceedings, 13th Annual ACM STOC," Milwaukee, pp. 146–157.

DURIS, P., AND GALIL, Z. (1982), Fooling a two-way automaton or one pushdown store is better than one counter for two-way machines, *Theoret. Comput. Sci.* **21**, 39–53.

FREEDMAN, A. R., AND LADNER, R. E. (1975), Space bounds for processing contentless inputs, *J. Comput. System Sci.* **11**, 118–128.

GALIL, Z. (1977), Some open problems in the theory of computation as questions about two-way deterministic pushdown automaton languages, *Math. Systems Theory* **10**, 211–228.