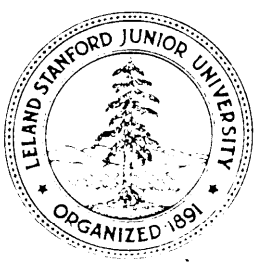
A LOWER BOUND TO PALINDROME RECOGNITION BY PROBABILSTIC TURING MACHINES

by

Andrew C. Yao

STAN-CS-77-647
DECEMBER 1977

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD UNIVERSITY



A Lower Bound to Palindrome Recognition by Probabilistic Turing Machines

Andrew Chi-Chih Yao

Computer Science Department
Stanford University
Stanford, California 94305

Abstract.

We call attention to the problem of proving lower bounds on probabilistic Turing machine computations. It is shown that any probabilistic Turing machine recognizing the language $L = \{w \not\in w \mid w \in \{0,1\}^*\}$ with error $\lambda < 1/2$ must take $\Omega(n \log n)$ time.

Keywords: computational complexity, crossing sequences, lower bound, palindrome, probabilistic Turing machine.

This research was supported in part by National Science Foundation grants MCS-72-03752 A03 and MCS-77-05313.

1. Introduction.

The idea of including controlled stochastic moves in algorithms has received considerable attention recently [4,9,10,12]. The demonstration by Rabin [9] and Solovay and Strassen [10], that fast tests for prime numbers can be done probabilistically with a small error, raised the hope that many more problems may allow similar fast algorithms. As in deterministic computations, a challenging problem is to prove lower bounds to the computational complexity of specific problems for such probabilistic algorithms. An investigation for decision tree type models was initiated in Yao [12]. The techniques used there [12], however, are not applicable to Turing machine computations, for which a number of lower bound results are known for the deterministic computations (see, e.g. [5]). In this paper, we call attention to proving lower bounds in probabilistic Turing machines, by proving a non-linear bound to a palindrome-like language.

It is well known that it takes any deterministic one-tape Turing machine $\Omega(n^2)$ steps to recognize the language $L = \{w \not\in w \mid w \in \{0,1\}^*\}$ (see, e.g. [5]). A very interesting result of Freivald [3], cited in Gill's paper [4], states that L can be recognized with a small error by a one-tape probabilistic Turing machine in time $O(n(\log n)^2)$. Recently, this bound was improved to $O(n \log n)$ by Nick Pippenger [7]. This seems to be the only example known in a Turing machine model where a provable speed-up is achieved, in an order-of-magnitude sense, by allowing stochastic decisions. The purpose of the present paper is to show that $\Omega(n \log n)$ -time is also a lower bound to any one-tape probabilistic Turing machine recognizing L with a small error (Theorem 4.1).

2. Definitions and Notations.

We first give an informal description of probabilistic Turing machines, the readers are referred to Gill [4] (and references therein) for more detailed discussions. A probabilistic one-tape Turing machine (1-PTM) M consists of a finite control, a read-write head on an infinite 1-dimensional tape, and a random symbol generator (RSG) capable of generating integers i between 1 and i_0 with fixed probabilities p_i. Before each move, a random symbol i is generated by the RSG, and the action of Turing machine M depends on the current state of M, the symbol on the tape being read, and the random symbol i. There are three distinguished states q_0, q_1, and q_2. The machine starts in q_0 (initial state), and if it halts, it must halt in either q_1 (the accepting state) or q_2 (the rejecting state). For a given input v, it is possible that M will not halt for some infinite sequences of random symbols that are generated by RSG. We shall restrict ourselves to M which, for any given input v, halts with probability 1 (it may still not halt for sane sequences). For each input v, let $\beta_i(v)$ be the probability that M halts in state q_i $(i = 1,2)$. A language L is recognized by M with error λ $(0 < \lambda < 1/2)$, if $\beta_1(v) > 1-\lambda$ for each $v \in L$, and $\beta_2(v) \geq 1-\lambda$ for each $v \notin L$. Intuitively, given an input v on M, if we accept or reject v when M halts depending on whether the state is q_1 or q_2, then we would be wrong at most with probability λ.

We shall now introduce some notations, and give a formal definition of terms involving probability described in the last paragraph. Let v be an input word, and σ a finite sequence of integers between 1 and i_0 generated by RSG that leads to the halting of M, i.e., M halts immediately after the last element in σ is generated. We say that σ is a decision sequence for v on M. Let $A(v)$ denote the set of all decision sequences for v, and $A_i(v) \subset A(v)$ $(i = 1,2)$ be the subsets consisting of σ leading to state q_i when M halts. We use $\sigma[j]$ for the j-th element in the sequence σ, and $|\sigma|$ for the length of σ. Define $P(\sigma) = p_{\sigma[1]}\, p_{\sigma[2]} \cdots p_{\sigma[|\sigma|]}$, the probability that the first $|\sigma|$ random symbols generated by RSG form the sequence σ. We now restate some terms defined earlier in these notations. The condition that M halts for any v with probability 1 means $\sum_{\sigma \in A(v)} P(\sigma) = 1$; the quantities $\beta_i(v)$ are $\sum_{\sigma \in A_i(v)} P(\sigma)$ for $i = 1,2$. Also, the expected number of steps M will make for input v is equal to

$$\bar{T}_M(v) = \sum_{\sigma \in A(v)} P(\sigma) \cdot |\sigma| \quad , \tag{1}$$

since $|\sigma|$ is the number of steps made when σ is the decision sequence generated. The running time of M for inputs of length n is defined to be the expected number of steps M will make for the worst input of length n, i.e.,

$$\bar{T}(M,n) = \max\{\bar{T}_M(v) \mid |v| = n\} \quad . \tag{2}$$

The quantities $\bar{T}_M(v)$, $\bar{T}(M,n)$ may be ∞.

We assume that the tape cells are numbered consecutively from $-\infty$ to ∞. Initially, an input v occupies cells 1 to $|v|$, and the head points to cell 0.

3. Crossing Sequences, Signatures, Patterns.

Let M be any probabilistic one-tape Turing machine. We develop some concepts concerning the behavior of M . For convenience, we assume that M satisfies the following conditions:

Standard-Form 1-PTM: Before it halts, the head always moves to the rightmost non-blank cell, where M enters either q_1 or q_2 . The machine then stays in the same state while making a full sweep to the left, and halts at the leftmost non-blank cell. Furthermore, it is assumed that M cannot enter either q_1 or q_2 until this last sweep,

A routine argument shows that any 1-PTM M can be transformed into an M' of standard form with $\bar{T}(M',n) = O(\bar{T}(M,n)+n)$.

Crossing Sequences.

We extend the notion of crossing sequences used in deterministic one-tape Turing machines (e.g. [5]). Consider the behavior of M for input v and some decision sequence $\sigma \in A(v)$. At the boundary between the j-th and the j+1 -st cells, let $\rho(v,\sigma,j)$ denote the sequence of states in which M passes through this position. The length of $\rho(v,\sigma,j)$ is denoted by $|\rho(v,\sigma,j)|$, which is 0 if $\rho(v,\sigma,j)$ is the empty sequence. The expected length of crossing sequence at j is defined as

$$\bar{\ell}(v, j) = \sum_{\sigma \in A(v)} P(\sigma) \cdot |\rho(v,\sigma,j)| ,$$

which may be ∞ .

A basic connection between running time and crossing sequences is given by the following lemma.

Lemma 3.1. For any input v , $\bar{T}_M(v) \geq \sum_j \bar{\ell}(v,j)$.

Proof. The lemma is obviously true if $\bar{T}_M(v) = \infty$; we therefore assume that $\bar{T}_M(v)$ is finite. For each $\sigma \in A(v)$, the number of steps taken is at least as large as the sum of the lengths of all crossing sequences. Thus,

$$|\sigma| \geq \sum_j |\rho(v, \sigma, j)| .$$

By definition,

$$\bar{T}_M(v) = \sum_{\sigma \in A(v)} P(\sigma) \cdot |\sigma| \geq \sum_{\sigma \in A(v)} \sum_j P(\sigma) |\rho(v, \sigma, j)| = \sum_j \bar{\ell}(v,j) ,$$

where in the last step we have changed the order of summation of an absolutely convergent double series (see, e.g. [11, p.28, Example 1]). □

Signatures and Patterns.

Let $Q = \{q_0, q_1, q_2, \ldots, q_r\}$ be the set of states, and Γ be the set of tape symbols used by M . Denote $Q - \{q_1, q_2\}$ by Q' .

Suppose during the computation process of M , the following configuration is encountered. A word $u \in \Gamma^*$ is on the tape from the $j+1$ -st cell to $(j+|u|)$ -th cell, and all cells to the right of u are blank (see Figure 1 top). The machine M is in state s , and its head is just crossing from the j-th to the $j+1$ -st cell.

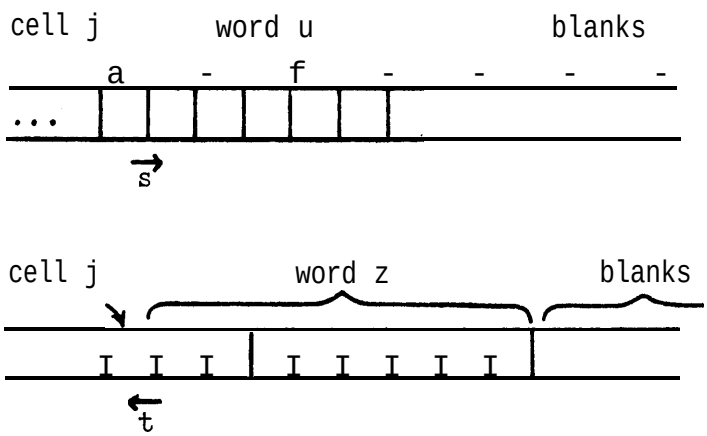


Figure 1. Illustration for $g(s,u;t,z)$.

We are interested in the situation when M comes back crossing from the $j+1$ -st cell to the j-th cell for the first time. As M is not deterministic, the state M is in and the contents of cells at this time may not be unique. We shall use $g(s,u;t,z)$, where $t\epsilon Q$, $z\epsilon\Gamma^*$, to denote the probability that M is in state t and the contents of cells from the $j+1$ -st cells on are the word z followed by blanks (see Figure 1, bottom). The function g is independent of the contents in cells $i \leq j$, and the explicit value of j . Clearly $\sum_{t,z} g(s,u;t,z) \leq 1$.

Definition 3.2. Let $u\epsilon\Gamma^*$ and $k \geq 1$. The k-th-order left signature of u is the following $2\cdot(r-1)^{2k-1}$ -tuple of numbers,

$$G^{(k)}(u; s_1,t_1,s_2,t_2,\ldots s_k,t_k)$$

$$= \sum_{z_1,z_2,\ldots,z_k \epsilon \Gamma^*} g(s_1,u;t_1,z_1) \times g(s_2,z_1;t_2,z_2) \times \cdots \times g(s_k,z_{k-1};t_k,z_k) , \tag{3}$$

for each $s_1,t_1,s_2,t_2,\ldots,s_k \epsilon Q'$ and $t_k \epsilon \{q_1,q_2\}$.

We shall show that $G^{(k)}$ are well defined by (3) and in fact, satisfy

$$0 \leq G^{(k)}(u;s_1,t_1,\ldots,s_k,t_k) \leq 1 . \tag{4}$$

As all terms in the summation (3) are non-negative, it is sufficient to prove that, for every finite subset $V \subseteq \Gamma^*$, and every $u\epsilon\Gamma^*$, $s_1,t_1,\ldots,s_k \epsilon Q'$, $t_k \epsilon \{q_1,q_2\}$, the following is true:

$$\sum_{z_1, z_2, \ldots, z_k \in V} g(s_1, u; t_1, z_1) \times g(s_2, z_1; t_2, z_2) \times \cdots \times g(s_k, z_{k-1}; t_k, z_k) \leq 1 \ .$$

This can be proved by induction on k ; we have

$$\sum_{z_1, z_2, \ldots, z_k \in V} g(s_1, u; t_1, z_1) \times g(s_2, z_1; t_2, z_2) \times \cdots \times g(s_k, z_{k-1}; t_k, z_k)$$

$$\leq \sum_{z_1 \in V} g(s_1, u; t_1, z_1) \left[\max_{z \in V} \sum_{z_2, z_3, \ldots, z_k \in V} g(s_2, z; t_2, z_2) \times \cdots \times g(s_k, z_{k-1}; t_k, z_k) \right] 1$$

$$< \sum_{z_1 \in V} g(s_1, u; t_1, z_1) \times 1 < 1 \ ,$$

when the induction hypothesis is used to bound the expression $\left[\max_{z \in V} \sum \ldots \right]$ by 1 .

In a similar way, we shall define the "right-signatures". Denote by $h(t, x; s, y)$ the probability that, given M entering the word x from the right end in state t , the head first comes back across the right end, having changed the word x to y (Figure 2). We further use the notation $h_0(x; s, y)$ for the probability that, given that M entered the region containing x from the left end in state q_0 , it will first cross the right end of x in state s and have changed x to y (Figure 3).

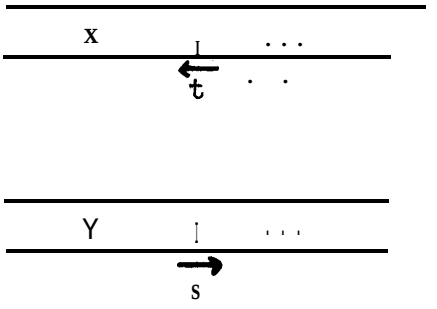


Figure 2. Illustration for $h(t,x;s,y)$.

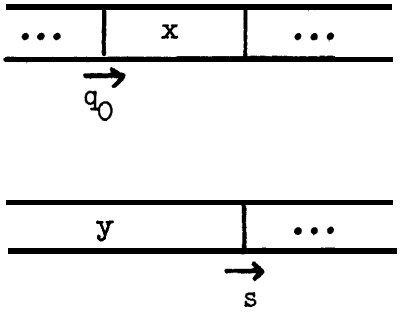


Figure 3. Illustration for $h_0(x;s,y)$.

Definition 3.3. Let $x \in \Gamma^*$ and $k > 1$. The k-th-order right signature of x is the $(r-1)^{2k-1}$-tuple,

$$H^{(k)}(x; s_1, t_1, s_2, t_2, \ldots, s_{k-1}, t_{k-1}, s_k)$$

$$= \sum_{y_1, y_2, \ldots \in \Gamma^*} h_0(x; s_1, y_1) \times h(t_1, y_1; s_2, y_2) \times h(t_2, y_2; s_3, y_3)$$

$$\times \cdots \times h(t_{k-1}, y_{k-1}; s_k, y_k) \ ,$$

where all $s_i, t_i \in Q'$.

As in the case of $G^{(k)}$, the numbers $H^{(k)}$ are well defined and satisfy

$$0 < H^{(k)}(x; s_1, t_1, \ldots, s_k) < 1 \ .$$

Definition 3.4. For each $u \in \Gamma^*$ and $k \geq 1$, the k-th-order pattern of u is a b_k-tuple of integers defined below, where $b_k = \sum_{1 \leq i \leq k} 2 \cdot (r-1)^{2i-1} < 2r^{2k}$. Let $I_k = r^{k^3}$, the b_k-tuple is given by

$$\left(\lceil I_k \cdot G^{(i)}(u; s_1, t_1, \ldots, s_i, t_i) \rceil \mid 1 \leq i \leq k, \ s_1, t_1, s_2, t_2, \ldots, s_i \in \ldots, t_i \in \{q_1, q_2\} \right) .$$

Some Facts.

Lemma 3.5. The number of distinct k-th -order patterns is at most $\exp(4k^3 r^{2k} \ln r)$.

Proof. Because of (4), the value of each component in a k-th -order pattern is an integer between 0 and Ik' There are thus at most $(I_k+1)^{b_k} \leq (r^{k^3}+1)^{2r^{2k}} \leq \exp(2k^3 \ln r \times 2r^{2k})$ distinct patterns. □

Definition 3.6. A sequence of states $(s_1, t_1, s_2, t_2, \ldots, s_k, t_k)$ is said to be $s_1, t_1, s_2, t_2, \ldots, s_k \in$ Q' and $t_k \in \{q_1, q_2\}$.

Lemma 3.7. Let B be a set of legal sequences and $v = xu$ an input word to M. Then $\alpha(v, |x|, B)$, the probability that, with input word v, the crossing sequence at position $|x|$ is in B, is given by

$$\alpha(v, |x|, B) = \sum_{k \geq 1} \sum_{(s_1, t_1, \ldots, s_k, t_k) \in B} H^{(k)}(x; s_1, t_1, \ldots, s_k) \times G^{(k)}(u; s_1, t_1, \ldots, s_k, t_k) .$$

Proof. The probability that the crossing sequence at $|x|$ is $(s_1, t_1, \ldots, s_k, t_k)$ is equal to

$$\sum_{y_1, z_1, y_2, z_2, \ldots, y_k, z_k \in \Gamma^*} h_0(x; s_1, y_1) \times g(s_1, u; t_1, z_1)$$

$$\times h(t_1, y_1; s_2, y_2) \times g(s_2, z_1; t_2, z_2) \times h(t_2, y_2; s_3, y_3) \ldots \times g(s_k, z_{k-1}; t_k, z_k) ,$$

which is $H^{(k)}(x; s_1, t_1, \ldots, s_k) \times G^{(k)}(u; s_1, t_1, \ldots, s_k, t_k)$. The lemma follows. □

Lemma 3.8. Let $x, u, w \in \Gamma^*$ and d, m positive integers. If $\bar{l}(xu, |x|) \leq d$ and w, u have the same (md)-th -order pattern, then

$$\beta_1(xw) \geq (1 - r^{-d^2})(\beta_1(xu) - m^{-1}) - 2r^{-(m^3 d^3 - d^2 - md)}$$

Corollary. Let $x,u,w \in \Gamma^*$ and $d > 10$. If $\bar{\ell}(xu, |x|) < d$, $\beta_1(xu) \geq 9/10$, and w, u have the same $(2d)$-th -order pattern, then $\beta_1(xw) \geq 1/5$.

Proof. Since $\bar{\ell}(xu; |x|) \leq d$, the probability that the crossing sequence at $|x|$ has length exceeding md is at most $1/m$. Let B_i $(i = 1,2)$ be the set of legal crossing sequences ending in q_i and of length at most md. Then

$$\alpha(xu, |x|, B_1) + \alpha(xu, |x|, B_2) \geq 1 - \frac{1}{m}.$$

Since $\alpha(xu, |x|, B_2) \leq \beta_2(xu) = 1 - \beta_1(xu)$, we have

$$\alpha(xu, |x|, B_1) \geq 1 - \frac{1}{m} - (1 - \beta_1(xu)) = \beta_1(xu) - \frac{1}{m}. \qquad (5)$$

Now, by Lemma 3.7,

$$\begin{aligned}
&\alpha(xw, |x|, B_1) - \alpha(xu, |x|, B_1) \\
&= \sum_{\sigma \in B_1} H(x;\sigma)(G(w;\sigma) - G(u;\sigma)) \\
&= \sum_{\substack{\sigma \in B_1 \\ G(u;\sigma) < r^{d^2}/I_{md}}} H(x;\sigma)(G(w;\sigma) - G(u;\sigma)) \\
&\qquad + \sum_{\substack{\sigma \in B_1 \\ G(u;\sigma) \geq r^{d^2}/I_{md}}} H(x;\sigma)(G(w;\sigma) - G(u;\sigma)) \quad . \qquad (6)
\end{aligned}$$

We have used here abbreviations $G(v;\sigma)$ and $H(v;\sigma)$ for $G^{(k)}(v;s_1,t_1,\ldots,s_k,t_k)$ and $H^{(k)}(v;s_1,t_1,\ldots,s_k)$ respectively, where $\sigma = (s_1,t_1,\ldots,s_k,t_k)$. The absolute value of the first term in (6) is bounded by $|B_1| \times \frac{r^{d^2}}{I_{md}} \leq 2r^{md} \times \frac{r^{d^2}}{I_{md}} = 2 \cdot r^{-(m^3d^3-d^2-md)}$, and that of the second term by $\sum_{\sigma \in B_1} H(x;\sigma) \frac{1}{I_{md}} \leq$

$$G(u;\sigma) \geq r^{d^2}/I_{md}$$

$\sum_{\sigma \in B_1} H(x;\sigma) \frac{1}{r^{d^2}} G(u;\sigma) \leq \alpha(xu, |x|, B_1) \frac{1}{r^{d^2}}$. Thus, we have $G(u;\sigma) \geq r^{d^2}/I_{md}$

from (6),

$$\alpha(xw, |x|, B_1) - \alpha(xu, |x|, B_1) \geq -2r^{-(m^3d^3-d^2-md)} - r^{-d^2}\alpha(xu, |x|, B_1) .$$

$$\alpha(xw, |x|, B_1) \geq (1-r^{-d^2})\alpha(xu, |x|, B_1) - 2r^{-(m^3d^3-d^2-md)} \qquad (7)$$

Using (5), (7) we obtain

$$\beta_1(xw) \geq \alpha(xw, |x|, B_1) \geq (1-r^{-d^2}) \left(\beta_1(xu) - \frac{1}{m}\right) - 2r^{-(m^3d^3-d^2-md)} .$$

This proves the lemma. The corollary follows by setting $m = 2$. □

4. Palindrome Recognition.

In this section we prove the following main result of this paper, using lemmas developed in Section 3.

Theorem 4.1. Let M be a probabilistic one-tape Turing machine that recognizes the language $L = \{w \not\in w \mid w \in \{0,1\}^*\}$ with error λ, where $0 < \lambda < 1/2$. Then there exists a constant $c > 0$ such that $\bar{T}(M,n) \geq cn \log n$ for infinitely many n.

Corollary. If M is a 1-PTM recognizing the language $\{1^n \not\in 1^n \mid n > 1\}$ with error $\lambda < 1/2$, then for infinitely many n, $\bar{T}(M,n) \geq cn \log \log n$ for some constant $c > 0$.

Proof. We shall assume that $\lambda = 1/10$. The general case follows because, from any 1-PTM M that recognizes L with error $\lambda = \frac{1}{2} - \Delta < \frac{1}{2}$, one can construct an M' recognizing L with error 1/10 and with running time at most a constant multiple of M. In fact, one can run M $2t-1$ times, where t satisfies $t(1-4\Delta^2)^t < 1/10$, and pick the majority answer as the output. This new M' has an error bounded by

$$\sum_{k>t} \binom{2t-1}{k} (1-\lambda)^{2t-1-k} \lambda^k \leq t\binom{2t-1}{t}(1-\lambda)^{t-1}\lambda^t < \frac{1}{2(1-\lambda)}\, t(1-4\Delta^2)^t < 1/10 ,$$

where we have used $\binom{2t-1}{t} < 2^{2t-1}$. Without loss of generality, we can further assume that M is of the standard form.

Let $L_n = \{1^n w \not\in 1^n w \mid w \in \{0,1\}^n\} \subseteq L$. Roughly, the idea is to show that, at each of the $n+1$ positions after the $\not\in$ mark (between the j-th and $j+1$-st cells for $2n+1 \leq j \leq 3n+1$), most of the words in L_n have an expected length of crossing sequence greater than $\Omega(\log n)$. This

leads to the existence of a $v \in L_n$ with an $\Omega(n \log n)$ expected total length of crossing sequences. From Lemma. 3.1, we would then have $\bar{T}(M, 4n+1) \geq \bar{T}_M(v) \geq \Omega(n \log n)$, proving the theorem.

Definition 4.2. $F_{n,j}(d) = \{v \mid v \in L_n , \bar{\ell}(v,j) \leq d\}$.

Claim 4.3. Let $d \geq 10$ be an integer. For each $2n+1 \leq j \leq 3n+1$, $|F_{n,j}(d)| \leq \exp(32d^3 r^{4d} \ln r)$.

Proof of Claim 4.3. For any $v \in L_n$, write $v = v'v''$ with $|v'| = j$. It is easy to--verify that, for any $v \neq w \in L_n$, the word $v'w'' \notin L$.

If the lemma is false, then $|F_{n,j}(d)| > \exp(32d^3 r^{4d} \ln r)$, and by Lemma 3.5, there exist $v \neq w \in F_{n,j}(d)$ such that v'' , w'' have the same (2d) -th order pattern. Now $\beta_1(v'v'') > 9/10$. By the corollary to Lemma 3.8, we have $\beta_1(v'w'') > 1/5$, contradicting the fact that $v'w'' \notin L_n$ (thus $\beta_1(v'w') = 1 - \beta_2(v'w'') \leq 1/10$). □

Let $d = \left\lceil \frac{1}{10} \log_r n \right\rceil$. Then, for all sufficiently large n , Claim 4.3 leads to

$$|L_n - F_{n,j}(d)| \geq 2^n - \exp(32d^3 r^{4d} \ln r) \geq \frac{1}{2} \times 2^n = \frac{1}{2} |L_n| .$$

Thus, for each $2n+1 \leq j \leq 3n+1$, we have

$$\sum_{v \in L_n} \bar{\ell}(v,j) \geq d \cdot |L_n - F_{n,j}(d)| \geq \frac{1}{2} d \cdot |L_n| . \qquad (8)$$

Now Lemma 3.1 implies

$$\max_{v \in L_n} \bar{T}_M(v) \geq \frac{1}{|L_n|} \sum_{v \in L_n} \bar{T}_M(v) \geq \frac{1}{|L_n|} \sum_{v \in L_n} \sum_{2n+1 \leq j \leq 3n+1} \bar{\ell}(v,j)$$

$$= \frac{1}{r^{L_n} r} \sum_{2n+1 \leq j \leq 3n+1} \sum_{v \in L_n} \bar{\ell}(v,j) \quad .$$

Because of (8), this gives

$$\max_{v \in L_n} \bar{T}_M(v) \geq \frac{1}{l^{L_n} r} \sum_{2n+1 \leq j \leq 3n+1} \frac{1}{2} d \cdot |L_n| = \frac{1}{2} d(n+1)$$

$$= \Omega(n \log n) \quad .$$

As explained earlier, this proves Theorem 4.1.

The corollary can be proved using the same idea. Denote the words $1^i \not\phi 1^i$ 'by v_i $(i \geq 1)$ and define $L'_n = \{v_i \mid 2n \leq i \leq 3n\}$. One shows that, at each of the n+1 positions after the $\not\phi$ mark, say the j-th position $(1 \leq j \leq n+1)$, at least half of the words v_i in L'_n have an expected length of crossing sequence $\bar{\ell}(v_i, i+j)$ greater than $\Omega(\log \log n)$. As before, this implies the existence of a $v_i \in L'_n$ whose expected total length of crossing sequence is $\Omega(n \log \log n)$. The corollary follows. We omit details of the derivation, as they are very-similar to the proof of the theorem, □

5. Some Remarks.

One curious fact is that, while the recognition of $\{w \not\in w\}$ requires only $O(n \log n)$ steps on an 1-PTM allowing a small error, a closely related "copying" problem -- changing an input w to $w \not\in w$ -- needs $\Omega(n^2)$ steps on any 1-PTM allowing a small error [13]. It seems in general easier to speed up the computation probabilistically if only the "checking" of an answer is involved.

In this connection the following interesting phenomenon concerning integer multiplication is worth noting -- one can check the answer of multiplying two n-bit integers x and y probabilistically with a small error faster than calculating the answer exactly. In fact, on a random access machine, if two n-bit numbers can be multiplied without error (probabilistically or deterministically) in $M(n)$ bitwise operations, then one can check the validity of $x \times y = z$, for a $2n$-bit number z, probabilistically with a small error in $O\left(\frac{n}{m} M(m)\right)$ bitwise operations, where $m = 2\lceil \lg(2n) \rceil$. For example, the Schönhage-Strassen algorithm (see, e.g. [1]) gives $M(n) = O(n(\log n)(\log \log n))$, which implies that the checking of $x \times y = z$ can be done probabilistically in only $O(n(\log \log n)(\log\log \log n))$ operations.

We now show that the above result easily follows from some basic observations of Pippenger [7] in his $O(n \log n)$-time 1-PTM for recognizing $\{w \not\in w\}$ -- (a) A t-bit random prime p (i.e., a random prime between 1 and $2^t - 1$) can be generated probabilistically with a small error in time $O(t^6)$ for some constant δ, and (b) if w_1, w_2 are two distinct positive integers of at most n bits, then for a $2\lceil \lg n \rceil$-bit random prime p, we have $w_1 \pmod p \neq w_2 \pmod p$ with probability greater than some absolute constant $\epsilon > 0$; thus one can

decide if $w_1 = w_2$ with only a small chance of error by comparing $w_1 \pmod p$ with $w_2 \pmod p$ for a fixed number of such random primes p generated by, for example, the method used in (a). These ideas imply that we can check the equation $x \times y = z$ with only a small chance of error by generating a few m-bit random primes p, computing $x \pmod p$, $y \pmod p$ and $z \pmod p$, and checking equations $(x \pmod p \cdot y \pmod p) \pmod p = z \pmod p$. The running time is dominated by the computing of $x, y, z \pmod p$, which takes $0\left(\frac{n}{m} M(m)\right)$ time (cf. [1]).

To end this section, we remark that the bound in the corollary to Theorem 4.1 is the best possible. By a slight adaptation of Pippenger's 1-PTM for recognizing $\{w \notin w\}$[7], one can construct a 1-PTM recognizing $\{1^n \notin 1^n \mid n > 1\}$ with a small error in time $O(n \log \log n)$, thus achieving the lower bound stated in the corollary.

6. Conclusions.

The subject of proving lower bounds for probabilistic Turing machines offers many challenging problems, of which only one is solved in this paper. It seems to be most fruitful to consider problems where good bounds exist in the deterministic case. We believe such studies will provide insights to probabilistic computations beyond the framework of Turing machine models. We mention only two such problems for further research.

(i) With a read-only input tape and several working tapes, is the extra space requirement for recognizing $\{w \not\in w\}$ probabilistically (with error) $\Omega(\log n)$? (See [5, p. 154, Exercise 10.3] for the deterministic analogue.

(ii) Can Rabin's language defined in [8] be recognized in real time by a probabilistic Turing machine with one working tape? (Deterministically it cannot [8].)

Finally we like to mention that the overlap argument for on-line multiplication (Cook and Aanderaa [2], also Paterson, Fischer, and Meyer [6]) can be extended to the probabilistic case [13].

Acknowledgments. I wish to thank John Gill for a stimulating conversation and for communicating Pippenger's result [7] to me.

References

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, Mass,, 1974.

[2] S. A. Cook and S. 0. Aanderaa, "On the minimum computation time of functions," Trans. Amer. Math. Soc. 142 (1969), 291-314.

[3] R. V. Freivald, "Fast computation by probabilistic Turing machines," Theory of Algorithms and Programs, no. 2, Latvian State University, Riga, 1975, 201-205 (in Russian).

[4] J. T. Gill III, "Computational complexity of probabilistic Turing machines," SIAM J. on Computing 6 (1977), 675-695.

[5] J. E. Hopcroft and J. D. Ullman, Formal Languages and Their Relation to Automata, Addison-Wesley, Reading, Mass., 1969.

[6] M. S. Paterson, M. J. Fischer, and A. R. Meyer, "An improved overlap argument for on-line multiplication," SIAM-AMS Proc., vol. 7, Amer. Math. Soc., Providence, R.I., 1974, 97-111.

[7] N. Pippenger, private communication, November 1977.

[8] M. 0. Rabin, "Real-time computation," Israel J. Math. 1 (1963), 203-211.

[9] M. 0. Rabin, "Probabilistic algorithms," in Algorithms and Complexity: New Directions and Recent Results, J. F. Traub, ed., Academic Press, New York, 1976, 21-39.

[10] R. Solovay and V. Strassen, "A fast Monte-Carlo test for primality," SIAM J. on Computing 6 (1977), 84-85.

[11] E. T, Whittaker and G. N. Watson, A Course of Modern Analysis, 4th edition, Cambridge University Press, 1958.

[12] A. C. Yao, "Probabilistic computations -- toward a unified measure of complexity," Proc. 18th Annual Symp. on Foundations of Computer Science, 1977, 222-227.

[13] A. C. Yao, unpublished.