A Time Complexity Gap For
2-Way Probabilistic Finite State Automata

Cynthia Dwork
Larry Stockmeyer

IBM Research Division K53/802
650 Harry Road
San Jose, California 95120

Abstract. It is shown that if a 2-way probabilistic finite state automaton (2pfa) M recognizes a nonregular language L with error probability bounded below 1/2, then there is a positive constant b (depending on M) such that, for infinitely many inputs x, the expected running time of M on input x must exceed 2^{n^b} where n is the length of x. This complements a result of Freivalds showing that 2pfa's can recognize certain nonregular languages in exponential expected time. It also establishes a time complexity gap for 2pfa's, since any regular language can be recognized by some 2pfa in linear time. Other results give roughly exponential upper and lower bounds on the worst-case increase in the number of states when converting a polynomial-time 2pfa to an equivalent 2-way nondeterministic finite-state automaton or to an equivalent 1-way deterministic finite-state automaton.

Keywords. Finite state automata, probabilisitic automata, complexity theory.

AMS(MOS) subject classifications. 68Q15, 68Q75

This paper appeared in the SIAM Journal on Computing, Vol. 19, No. 6, 1990, pp. 1011–1023.
Copyright © (1990) by The Society for Industrial and Applied Mathematics. All rights reserved.

1 Introduction

The power of randomization is a recurring theme in the theory of computation. One of the fundamental open questions of complexity theory asks whether the class BPP of languages recognizable in polynomial time by probabilistic Turing machines is larger than the class P of languages recognizable in polynomial time by deterministic Turing machines. To gain insight into the power of randomization, some research has focused on more restricted models of computation. For example, Freivalds [2] and Gill [4] have shown that probabilistic one-tape Turing machines can recognize certain languages more efficiently than deterministic one-tape Turing machines.

Another example of the power of randomization, which provided the motivation for this paper, concerns 2-way probabilistic finite-state automata (2pfa's). A 2pfa consists of a probabilistic finite-state control and an input tape which is scanned by a single 2-way head, that is, the head can move both left and right (a complete definition appears in Section 2). Freivalds [3] has shown that, for any $\varepsilon > 0$, there is a 2pfa which recognizes the nonregular language $L_0 = \{ 0^m 1^m \mid m \geq 1 \}$ with error probability at most ε. In contrast, it is known that deterministic (and even nondeterministic and alternating) 2-way finite-state automata can recognize only regular languages [8, 13, 15]. A property of Freivald's result is that the 2pfa constructed to recognize L_0 uses exponential expected time. In the original construction in [3], the expected time is proportional to $n2^n$, where n is the length of the input. The construction can be easily modified to give, for any constant $c > 0$, a 2pfa recognizing L_0 in expected time $O(2^{cn})$. Greenberg and Weiss [5] show that this expected time bound cannot be improved further. They show that no 2pfa running in expected time $2^{o(n)}$ can recognize L_0 with error probability bounded below 1/2. This raises the question of whether there is some other nonregular language which a 2pfa can recognize efficiently, for example, in polynomial expected time. The main result of this paper answers this question negatively: if a 2pfa M recognizes a nonregular language with error probability bounded below 1/2, then there is a constant $b > 0$ such that, for infinitely many n, the expected running time of M must exceed 2^{n^b}. Thus, although randomization adds power to 2-way finite-state automata, this added power comes with an added cost in expected running time. The result can also be viewed as a gap theorem for 2pfa's: for any language L, either L can be recognized in time $O(n)$ (if L is regular) or L requires time 2^{n^b} infinitely often (if L is not regular).

A tool in the proof of the main result is a measure $N_L(n)$ of the nonregularity of a language L. Intuitively, when L is restricted to words of length at most n, $N_L(n)$ measures the number of words which must be "distinguished" by any recognizer of L, where words w and w' must be distinguished on the restricted L if there exists a word v such that the lengths of wv and $w'v$ are both at most n, and $wv \in L$ iff $w'v \notin L$. In Section 3 we show that if L is not regular then $N_L(n) \geq \sqrt{n} - 1$ for infinitely many n. In Section 4 we show that, if a 2pfa M recognizes L in expected time $T(n)$, then a lower bound on $T(n)$ can be given in terms of $N_L(n)$. Combining this with the lower bound on $N_L(n)$ for any nonregular L, the main result follows easily.

In the final three sections we mention some related results. In Section 5 the main result is extended to show that a probabilistic Turing machine with space bound $o(\log\log n)$ which runs in polynomial expected time cannot recognize any nonregular language with error probability bounded below 1/2. In Section 6 we consider conciseness issues, that is, the relative number of states required by probabilistic versus deterministic or nondeterministic finite-state machines to recognize the same regular language. When 2pfa's are restricted to halt in polynomial expected time, 2pfa's are at most exponentially more concise than 1-way deterministic finite-state automata, and there are examples showing that 2pfa's can be exponentially more concise than 2-way nondeterministic finite state automata. In Section 7 we note that the main result is not true if the error probability is not bounded below 1/2.

2 Definitions

A particular 2pfa M is specified by a finite set Q of states, a finite input alphabet Σ, and a transition function δ. The set Q contains designated states q_0 (the initial state), q_a (the accepting state), and q_r (the rejecting state). Let $\cent$ be a symbol not in Σ. The transition function has the form

$$\delta : (Q - \{q_a, q_r\}) \times (\Sigma \cup \{\cent\}) \times Q \times \{\text{left, right, stationary}\} \to [0, 1],$$

where, for each fixed q and σ, the sum of $\delta(q, \sigma, q', d)$ over all q' and d equals 1. The meaning of δ is that, if M is in state q with the head scanning the symbol σ, then with probability $\delta(q, \sigma, q', d)$ the machine enters state q' and either moves the head one symbol in direction d if $d \in \{\text{left, right}\}$ or does not move the head if $d = \text{stationary}$. The computation of M on input $x \in \Sigma^*$ begins with the word $\cent x \cent$ written on the input tape; the head is positioned on the left endmarker $\cent$, and the state is q_0. The computation is then governed (probabilistically) by the transition function δ until M either accepts by entering state q_a or rejects by entering state q_r. We assume that δ is defined so that the head never moves outside the word $\cent x \cent$. M *halts* when it enters state q_a or q_r.

Let $L \subseteq \Sigma^*$, let M be a 2pfa with input alphabet Σ, and let $0 \leq \varepsilon < 1/2$. Then M *recognizes* L *within error probability* ε if

1. for all $x \in L$, $\Pr[\, M \text{ accepts } x\,] \geq 1 - \varepsilon$, and
2. for all $x \notin L$, $\Pr[\, M \text{ rejects } x\,] \geq 1 - \varepsilon$.

M *recognizes* L if M recognizes L within error probability ε for some $\varepsilon < 1/2$. One reason to bound the error probability below 1/2 is so that the error probability can be significantly decreased by running the machine several times on the same input and taking the majority answer. Specifically, if a 2pfa M recognizes L and halts with probability 1 on all inputs, then for any $\varepsilon > 0$ there is a 2pfa M' which recognizes L within error probability ε.

Although our lower bound result allows δ to take arbitrary real values in $[0, 1]$, other results construct particular 2pfa's which need only a fair coin to make random choices. A 2pfa is a *coin-tossing 2pfa* if the range of its transition function δ is $\{0, 1/2, 1\}$. For example, the result of Freivalds [3] mentioned above is true for a coin-tossing 2pfa. (It is not hard to see that 2pfa's with rational transition probabilities are no more powerful than coin-tossing 2pfa's, although we do not know whether 2pfa's with real transition probabilities are more powerful than coin-tossing 2pfa's.)

Let $|x|$ denote the length of the word x. The *expected running time* of the 2pfa M on the input x is the expected number of steps in the computation of M on input x, where the expectation is taken over the random choices made by M. The 2pfa M *runs within expected time* $T(n)$ if, for all n and all inputs x with $|x| \leq n$, the expected running time of M on x is at most $T(n)$.

Logarithms in this paper are to the base 2.

3 A Measure of Nonregularity

Let $L \subseteq \Sigma^*$ and let n be a positive integer. We first define a relation $\sim_{L,n}$ on words. For $w, w' \in \Sigma^*$ with $|w| \leq n$ and $|w'| \leq n$, the words w and w' are *n-similar*, written $w \sim_{L,n} w'$, if, for all $v \in \Sigma^*$ such that $|wv| \leq n$ and $|w'v| \leq n$, we have $wv \in L$ iff $w'v \in L$. The words w and w' are *n-dissimilar*, written $w \not\sim_{L,n} w'$, if $|w| \leq n$, $|w'| \leq n$, and it is not the case that $w \sim_{L,n} w'$, that is, there exists a v with $|wv| \leq n$, $|w'v| \leq n$, and $wv \in L$ iff $w'v \notin L$. Note that if $w \not\sim_{L,n} w'$, then w and w' must be "distinguished" as described in the Introduction. The relations $\sim_{L,n}$ and $\not\sim_{L,n}$ are not defined for words of length greater than n. We remark that $\sim_{L,n}$ is reflexive and symmetric, although it is not transitive in general.

Let $N_L(n)$ be the maximum k such that there exist k distinct words $w_1, \ldots, w_k$ which are pairwise $\not\sim_{L,n}$, that is, $w_i \not\sim_{L,n} w_j$ for all $1 \leq i < j \leq k$. It is clear that $N_L(n)$ is nondecreasing in n.

Lemma 3.1 *$N_L(n)$ is bounded above by a constant iff L is regular.*

Proof. For words w and w', define wR_Lw' if, for all words v, we have $wv \in L$ iff $w'v \in L$. R_L is an equivalence relation. By the Myhill-Nerode Theorem (see [7]), R_L has finite index iff L is regular.

If L is regular, it is obvious from definitions that $N_L(n)$ is bounded above by the (finite) index of R_L. Conversely, if L is not regular, then, for any $k \geq 2$ there exist words $w_1, \ldots, w_k$ and words v_{ij} such that $w_iv_{ij} \in L$ iff $w_jv_{ij} \notin L$ for all $1 \leq i < j \leq k$. Choosing n_0 to be the maximum length of the words w_iv_{ij}, w_jv_{ij} over all such i and j, we have $N_L(n_0) \geq k$. Since k was arbitrary, $N_L(n)$ is unbounded. □

For the remaining four lemmas in this section, fix some language L, and abbreviate $\sim_{L,n}$ and $\not\sim_{L,n}$ by $\sim_n$ and $\not\sim_n$, respectively. The next two lemmas establish two useful properties of $\sim_n$ and $\not\sim_n$.

Lemma 3.2 *Let w, w', v be words such that $w \sim_n w'$, $|wv| \leq n$, and $|w'v| \leq n$. Then $wv \sim_n w'v$.*

Proof. Suppose to the contrary that $wv \not\sim_n w'v$. Therefore, there exists a word z such that $|wvz| \leq n$, $|w'vz| \leq n$, and $wvz \in L$ iff $w'vz \notin L$. But now vz is a witness to $w \not\sim_n w'$, contradicting the assumption that $w \sim_n w'$. □

Lemma 3.3 *Let w, w', s be words such that $|s| \leq |w|$, $w \sim_n s$, and $w \not\sim_n w'$. Then $s \not\sim_n w'$.*

Proof. Let v be such that $|wv| \leq n$, $|w'v| \leq n$, and $wv \in L$ iff $w'v \notin L$. We have $|sv| \leq |wv| \leq n$. Since $w \sim_n s$, we have $sv \in L$ iff $wv \in L$. Therefore, $sv \in L$ iff $w'v \notin L$, so $s \not\sim_n w'$. □

The next lemma shows that we can always find $N_L(n)$ "short" words which are pairwise $\not\sim_n$.

Lemma 3.4 *Let $k = N_L(n)$. There exist words $s_1, \ldots, s_k$ which are pairwise $\not\sim_n$ and $|s_i| \leq k$ for all i.*

Proof. By definition, there are words $w_1, \ldots, w_k$ which are pairwise $\not\sim_n$. If $|w_i| \leq k$ for all i, we are done. Suppose that $|w_i| > k$ for some i. Consider the $|w_i|$ nonempty prefixes of w_i. Since $N_L(n) = k$, these prefixes cannot be pairwise $\not\sim_n$. Therefore, we can write $w_i = xyz$ where $|y| \geq 1$ and $x \sim_n xy$. Since $|xz| < |xyz| \leq n$, it follows from Lemma 3.2 that $xz \sim_n xyz = w_i$. For an arbitrary j with $1 \leq j \leq k$ and $j \neq i$, we have $w_i \not\sim_n w_j$ by choice of the w's. Taking $s = xz$ in Lemma 3.3, we see that $xz \not\sim_n w_j$. Replacing w_i by xz in the list $w_1, \ldots, w_k$, we obtain a new set of pairwise $\not\sim_n$ words where the length of w_i has been reduced. Continuing in this way, k pairwise $\not\sim_n$ words of length at most k can be constructed. □

Lemma 3.5 *Let n and k be such that $N_L(n-1) < N_L(n) = k$. Then $n \leq 2k + k^2$.*

Proof. Assume to the contrary that $n > 2k + k^2$. By Lemma 3.4, there are pairwise $\not\sim_n$ words $s_1, \ldots, s_k$ with $|s_i| \leq k$ for all i. Our goal is to show that, for all i and j with $1 \leq i < j \leq k$, there exists a word u such that $|s_i u| \leq n-1$, $|s_j u| \leq n-1$, and $s_i u \in L$ iff $s_j u \notin L$. This implies that $s_1, \ldots, s_k$ are pairwise $\not\sim_{n-1}$, contradicting the assumption that $N_L(n-1) < k$.

Since $s_i \not\sim_n s_j$, there is a word v with $|s_i v| \leq n$, $|s_j v| \leq n$, and $s_i v \in L$ iff $s_j v \notin L$. If $|s_i v|$ and $|s_j v|$ are both less than n, we are done. So assume (without loss of generality) that $|s_i v| = n$. Let $v_1, \ldots, v_m$ be the prefixes of v having length at least k. Since $|s_i| \leq k$ and $n > 2k + k^2$, it follows that $m > k^2$. For each word $s_i v_l$ with $1 \leq l \leq m$ there exists an a such that $s_i v_l \sim_n s_a$, for otherwise there would be $k+1$ pairwise $\not\sim_n$ words, contradicting the assumption that $k = N_L(n)$. Similarly, for each l there is a b such that $s_j v_l \sim_n s_b$. Since there are more than k^2 choices for l, and exactly k^2 choices of pairs (a, b), it follows that there exist numbers a, b and words x, y, z such that $v = xyz$, $|x| \geq k$, $|y| \geq 1$, and

$$
\begin{aligned}
s_i x &\sim_n s_a && (1)\\
s_i x y &\sim_n s_a && (2)\\
s_j x &\sim_n s_b && (3)\\
s_j x y &\sim_n s_b. && (4)
\end{aligned}
$$

Since $|v| \leq n$ and $|x| \geq k$, we have $|z| \leq n - k$, so $|s_a z| \leq k + (n-k) = n$. Similarly, $|s_b z| \leq n$. Finally,

$$
\begin{array}{rlll}
s_i xz \in L & \text{iff} & s_a z \in L & \text{by (1)}\\
& \text{iff} & s_i xyz \in L & \text{by (2)}\\
& \text{iff} & s_i v \in L & \text{since } v = xyz\\
& \text{iff} & s_j v \notin L & \text{by choice of } v\\
& \text{iff} & s_j xyz \notin L & \text{since } v = xyz\\
& \text{iff} & s_b z \notin L & \text{by (4)}\\
& \text{iff} & s_j xz \notin L & \text{by (3).}
\end{array}
$$

Therefore, $u = xz$ is a word such that $|s_i u| \leq n-1$, $|s_j u| \leq n-1$, and $s_i u \in L$ iff $s_j u \notin L$. □

We can now obtain our main result about $N_L(n)$.

Theorem 3.6 *If L is not regular, then $N_L(n) \geq \sqrt{n} - 1$ for infinitely many n.*

Proof. By Lemma 3.1, there are infinitely many n such that $N_L(n-1) < N_L(n)$. For each such n, we have $N_L(n) \geq \sqrt{n} - 1$ by Lemma 3.5. □

Two remarks about strengthening Theorem 3.6 can be made. First, the lower bound $\sqrt{n} - 1$ could be improved to at most $n+1$, since if $L \subseteq \{0\}^*$ then $N_L(n) \leq n+1$ for all n. We have not pursued such an improvement, since it would not improve the lower bound on expected time obtained in the next section. Second, regarding that the lower bound has been shown to hold only for infinitely many n, no interesting lower bound on $N_L(n)$ can be proved for almost all n, given only that L is not regular. If $g(n)$ is any unbounded function, there is a nonregular L such that $N_L(n) \leq g(n)$ for infinitely many n. For integers $0 = n_1 < n_2 < n_3 < \cdots$, define

$$L = \{\, 0^m \mid (\exists i)[\, n_{i-1} < m \leq n_i \;\text{ and }\; m \equiv 0 \bmod i \,]\,\}.$$

Then $N_L(n_i) \leq n_{i-1} + 1 + i$ for all $i \geq 2$. So by choosing the n_i to be far enough apart, we can satisfy $N_L(n_i) \leq g(n_i)$ for all $i \geq 2$.

4 A Lower Bound on Expected Time

In this section we give a lower bound, in terms of $N_L(n)$, on the expected time required by a 2pfa to recognize L. The proof method is one which we used in [1]; it is similar to methods used previously by Rabin [12] and Greenberg and Weiss [5].

Since we model computations of 2pfa's by Markov chains, we first give some definitions and results about Markov chains. Basic facts about Markov chains can be found, for example, in [14]. We consider Markov chains with finite state space $\{1, 2, \ldots, m\}$ for some m. A particular Markov chain is completely specified by its matrix $P = \{p_{ij}\}_{i,j=1}^m$ of transition probabilities. If the Markov chain is in state i, then it next moves to state j with probability p_{ij}. The chains we consider have a designated starting state, state 1, and two absorbing states, states $m-1$ and m (so $p_{m-1,m-1} = p_{m,m} = 1$). States other than $m-1$ and m are either transient or inaccessible from state 1. Let $a(P)$ denote the probability that the Markov chain P is absorbed in state m when started in state 1. Let $t(P)$ denote the expected time to absorption, meaning absorption into one of the states $m-1$ or m.

We are concerned with how the probability $a(P)$ is affected by small changes to the transition probabilities. Let $\beta \geq 1$. Say that two numbers p and p' are *β-close* if either (i) $p = p' = 0$ or (ii) $p > 0$, $p' > 0$, and $\beta^{-1} \leq p/p' \leq \beta$. Two Markov chains $P = \{p_{ij}\}_{i,j=1}^m$ and $P' = \{p'_{ij}\}_{i,j=1}^m$ are *β-close* if, for all i and j, p_{ij} and p'_{ij} are β-close.

As noted by Greenberg and Weiss [5] (see also [1, Lemma 3.2]), the following lemma can be proved easily from the Markov chain tree theorem [9, 10].

Lemma 4.1 *Let P and P' be two m-state Markov chains which are β-close. Then $a(P)$ and $a(P')$ are β^{2m}-close.*

We need a variation of this lemma, where certain corresponding pairs of probabilities p_{ij} and p'_{ij} are not known to be β-close, but only in the case that these probabilities are both much smaller than the reciprocal of the expected time to absorption. For $\beta \geq 1$ and $\lambda \geq 0$, we say that P and P' are *β-close mod λ* if, for each pair i, j, either

1. $p_{ij} \leq \lambda$ and $p'_{ij} \leq \lambda$, or

2. $p_{ij} > \lambda$, $p'_{ij} > \lambda$, and p_{ij} and p'_{ij} are β-close.

The basic idea is that, if λ is much smaller than both $t(P)^{-1}$ and $t(P')^{-1}$, then a transition from state i to state j is unlikely to occur (before absorption) in case 1, so such transitions can be essentially ignored. The reader satisfied with this intuition might want to skip the proof of the next lemma.

Lemma 4.2 *Let P and P' be two m-state Markov chains which are β-close mod λ. Let $t = \max\{t(P), t(P')\}$. Then*

$$a(P') \geq (1 - 2\lambda m^3)\beta^{-2m} a(P) - 4\sqrt{\lambda m t}.$$

Proof. Assume $\lambda \leq m^{-3}$, since the lemma clearly holds otherwise. Say that a number p is *small* if $p \leq \lambda$. We transform P and P' to Markov chains R and R', respectively, by changing all small transition probabilities to zero and altering the non-small probabilities only slightly. We can then bound $a(P)$ in terms of $a(R)$, similarly bound $a(P')$ in terms of $a(R')$, and apply Lemma 4.1 to R and R'.

In order to define certain events, we first describe a particular way of "running" the Markov chains P and P'. Since the description is essentially identical for P and P', we focus on P. For each state i, order the transitions from state i so that all small transition probabilities precede all non-small transition probabilities and so that the first non-small transition probability is at least $1/m$. More precisely, for each state i, choose a permutation π_i of $\{1, 2, \ldots, m\}$ and an integer l_i with $1 \leq l_i \leq m$ such that $p_{i,\pi_i(k)}$ is small iff $k < l_i$, and $p_{i,\pi_i(l_i)} \geq 1/m$. We say that $\pi_i(l_i)$ is *special* for i. Let S be a random variable which takes a real value uniformly distributed in $[0, 1]$. We use S to run P as follows. If P is in state i, call S to obtain a random value v. Then P next enters state $\pi_i(z)$ where

$$\sum_{k=1}^{z-1} p_{i,\pi_i(k)} < v \leq \sum_{k=1}^{z} p_{i,\pi_i(k)}.$$

So for each z, P next enters state $\pi_i(z)$ with probability $p_{i,\pi_i(z)}$.

Let $\mathcal{E}$ be the event that P is absorbed into state m, and let $\mathcal{F}$ be the event that P is absorbed (into state m or $m-1$) before any call of S produces a value v with $v \leq \lambda m$. Note that if $\mathcal{F}$ holds, then no transition $i \to j$ with small p_{ij} is taken before absorption. Let $\mathcal{E}'$ and $\mathcal{F}'$ be the analogous events for P'. Let A be the random variable giving the number of steps of P before absorption. Let $b = (\lambda m t)^{-1/2}$. Let $\overline{\mathcal{F}}$ be the complement of $\mathcal{F}$. Since $E(A) = t(P) \leq t$,

$$\Pr[\, A > bt \,] \leq b^{-1}.$$

By definition of $\mathcal{F}$, and using the inequality $(1-x)^y \geq 1 - xy$ which is valid for all real x and y with $0 \leq x \leq 1$ and $y \geq 1$ [6, Thm. 42],

$$\Pr[\, \overline{\mathcal{F}} \mid A \leq bt \,] \leq 1 - (1 - \lambda m)^{bt} \leq \lambda m b t.$$

Therefore,

$$\Pr[\, \overline{\mathcal{F}} \,] \leq b^{-1} + \lambda m b t = 2\sqrt{\lambda m t}. \tag{5}$$

Note that $a(P) = \Pr[\mathcal{E}]$ by definition. We will soon define the Markov chains R and R' so that $a(R) = \Pr[\,\mathcal{E} \mid \mathcal{F}\,]$ and $a(R') = \Pr[\,\mathcal{E}' \mid \mathcal{F}'\,]$. The law of conditional probability,

$$\Pr[\mathcal{E}] = \Pr[\,\mathcal{E} \mid \mathcal{F}\,](1 - \Pr[\overline{\mathcal{F}}]) + \Pr[\,\mathcal{E} \mid \overline{\mathcal{F}}\,]\Pr[\overline{\mathcal{F}}],$$

yields

$$\Pr[\,\mathcal{E} \mid \mathcal{F}\,] - \Pr[\overline{\mathcal{F}}] \le \Pr[\mathcal{E}] \le \Pr[\,\mathcal{E} \mid \mathcal{F}\,] + \Pr[\overline{\mathcal{F}}].$$

It follows from this and (5) (and from the analogous inequalities for the primed case) that

$$|a(P) - a(R)| \le 2\sqrt{\lambda m t} \quad \text{and} \quad |a(P') - a(R')| \le 2\sqrt{\lambda m t}. \tag{6}$$

We now define R to model P in the case that S produces a value uniformly in $(\lambda m, 1]$. The definition of R' in terms of P' is analogous. For each state i, let σ_i be the sum of p_{ij} over all j such that p_{ij} is small. For each i and j,

1. if p_{ij} is small, then $r_{ij} = 0$;
2. if p_{ij} is not small and j is not special for i, then $r_{ij} = p_{ij}/(1 - \lambda m)$;
3. if j is special for i, then $r_{ij} = (p_{ij} + \sigma_i - \lambda m)/(1 - \lambda m)$.

We must bound the closeness of r_{ij} to p_{ij} in the case 3 that j is special. Using $\sigma_i \le \lambda m$,

$$r_{ij} \le p_{ij}/(1 - \lambda m).$$

Using $p_{ij} \ge 1/m$,

$$r_{ij} \ge (p_{ij} - \lambda m)/(1 - \lambda m) \ge p_{ij}(1 - \lambda m^2)/(1 - \lambda m).$$

Since P and P' are β-close mod λ, it follows that R and R' are $(\beta(1 - \lambda m^2)^{-1})$-close. Using Lemma 4.1 and the bounds (6),

$$\begin{aligned}
a(P') &\ge a(R') - 2\sqrt{\lambda m t} \\
&\ge \left(\frac{\beta}{1 - \lambda m^2}\right)^{-2m} a(R) - 2\sqrt{\lambda m t} \\
&\ge (1 - 2\lambda m^3)\beta^{-2m}(a(P) - 2\sqrt{\lambda m t}) - 2\sqrt{\lambda m t} \\
&\ge (1 - 2\lambda m^3)\beta^{-2m} a(P) - 4\sqrt{\lambda m t}.
\end{aligned}$$

□

We can now prove the main lemma of this section.

Lemma 4.3 *For every $\varepsilon < 1/2$ there are positive constants α_ε and η_ε such that, if a 2pfa M having c states recognizes the language L within error probability ε and within expected time $T(n)$, then*

$$(\alpha_\varepsilon c(\log T(n) + \log cn))^{c^2} \ge N_L(n) \quad \textit{for all } n \ge \eta_\varepsilon. \tag{7}$$

Proof. Let M be a 2pfa with c states which recognizes L within error probability $\varepsilon < 1/2$ and within expected time $T(n)$. It is convenient to adopt a new starting convention by giving M a new initial state q_1. M is started on input $\cent x \cent$ in state q_1 with the head scanning the right $\cent$; the machine then moves left across the input, remaining in state q_1, until the head reads the left $\cent$, at which point M enters its original initial state. Let $q_1, q_2, \ldots, q_{c+1}$ be the states of the modified M, where q_1 is the new initial state, q_c is the rejecting state, and q_{c+1} is the accepting state.

Note that $\sqrt{(1-\varepsilon)/2} > 1/2$, since $\varepsilon < 1/2$. Therefore, we can choose the constant η_ε so that

$$(1-2/n)\sqrt{(1-\varepsilon)/2} - 4\sqrt{1/n} > 1/2 \quad \text{for all } n \geq \eta_\varepsilon.$$

Fix some $n \geq \eta_\varepsilon$. Let W be a set of pairwise $\not\sim_n$ words with $|W| = N_L(n)$. For each $w \in W$, we define certain probabilities, called *word probabilities*, which capture the behavior of M on $\cent w$. A *starting condition* is an integer i with $1 \leq i \leq c-1$. A *stopping condition* is an integer j with $1 \leq j \leq c+1$. The starting condition i means "start M in state q_i on the word $\cent w$ with the head on the rightmost symbol of $\cent w$." We say that M *halts on $\cent w$ in state q_j* if either (i) $1 \leq j \leq c-1$ and the head moves off the right end of $\cent w$ at the same step as M enters state q_j, or (ii) $c \leq j \leq c+1$ and M enters state q_j before or at the step when the head moves off the right end of $\cent w$. If $j \leq c-1$ or $j = c+1$, the stopping condition j means "M halts on $\cent w$ in state q_j." The stopping condition c means "either M halts on $\cent w$ in state q_c or M does not halt on $\cent w$." For each starting condition i and stopping condition j, let $q_{ij}(w)$ be the probability of the associated event. Let $Q(w)$ be the $(c-1)\times(c+1)$ matrix $Q(w) = \{q_{ij}(w)\}$.

Let

$$d = (c-1)(c+1),$$
$$m = 2c,$$
$$\lambda = (nm^3T(n))^{-1}.$$

Define an equivalence relation $\equiv$ on words in W as follows: $w \equiv w'$ iff, for all i and j, $q_{ij}(w) \leq \lambda$ iff $q_{ij}(w') \leq \lambda$. Let E be a largest equivalence class. Since there are at most 2^d equivalence classes, $|E| \geq N_L(n)/2^d$.

For $w \in E$, let d' be the number of entries of $Q(w)$ which are strictly larger than λ. (The number and positions of these entries does not depend on the particular choice of w.) By projecting on these entries, map $Q(w)$ to an element $\mathbf{q}(w)$ of the space $[\lambda, 1]^{d'}$. Let $\log \mathbf{q}(w)$ be the componentwise log of $\mathbf{q}(w)$, so $\log \mathbf{q}(w) \in [\log \lambda, 0]^{d'}$. Let

$$\mu = \frac{\log(1-\varepsilon)+1}{4m}.$$

By dividing each coordinate interval $[\log \lambda, 0]$ into subintervals of length μ (with possibly one subinterval of length less than μ), we partition the space $[\log \lambda, 0]^{d'}$ into at most $\lceil(-\log \lambda)/\mu\rceil^d$ *cells*, each of size at most $\mu \times \mu \times \cdots \times \mu$.

We now show that

$$\lceil(-\log \lambda)/\mu\rceil^d \geq N_L(n)/2^d. \tag{8}$$

This suffices to prove the lemma, since by substituting the values of λ, μ and d into (8),

$$\left(\frac{16c(\log T(n) + \log(8nc^3))}{\log(1-\varepsilon)+1} + 2\right)^{(c-1)(c+1)} \geq N_L(n).$$

So it is easy to see that the required α_ε exists.

Assuming that (8) does not hold, there must be two n-dissimilar words $w, w' \in E$ such that $\log \mathbf{q}(w)$ and $\log \mathbf{q}(w')$ belong to the same cell. Therefore, if q_{ij} and q'_{ij} are corresponding entries of $Q(w)$ and $Q(w')$, respectively, then either

1. $q_{ij} \leq \lambda$ and $q'_{ij} \leq \lambda$, or
2. $q_{ij} > \lambda$, $q'_{ij} > \lambda$, and $|\log q_{ij} - \log q'_{ij}| \leq \mu$.

In the second case, q_{ij} and q'_{ij} are 2^μ-close.

Since $w \not\sim_n w'$, let v be such that $|wv| \leq n$, $|w'v| \leq n$, and $wv \in L$ iff $w'v \notin L$. By symmetry, assume that $wv \in L$ and $w'v \notin L$. We describe Markov chains P and P' which model the computation of M on wv and $w'v$, respectively. First define word probabilities for $v\not{c}$ similarly to the definition above for $\not{c}w$. The only difference is that starting conditions mean to start M in some state on the leftmost symbol of $v\not{c}$, and halting on $v\not{c}$ occurs when the head moves off the left end of $v\not{c}$ or when q_c or q_{c+1} is entered. For starting condition i with $1 \leq i \leq c-1$ and stopping condition j with $1 \leq j \leq c+1$, let $r_{ij}(v)$ be the probability of the associated event, and let $R(v)$ be the $(c-1)\times(c+1)$ matrix $\{r_{ij}(v)\}$.

The partial computations represented by $Q(w)$ and $R(v)$ are glued together by P in the obvious way. Formally, write $Q(w) = (Q_1(w)\; Q_2(w))$, $Q(w') = (Q_1(w')\; Q_2(w'))$, and $R(v) = (R_1(v)\; R_2(v))$, where $Q_1(w), Q_1(w')$ and $R_1(v)$ are $(c-1)\times(c-1)$, and $Q_2(w), Q_2(w')$ and $R_2(v)$ are $(c-1)\times 2$. Let I_2 be the 2×2 identity matrix. Then the $m\times m$ transition matrices P and P' are given by

$$P = \begin{pmatrix} 0 & Q_1(w) & Q_2(w) \\ R_1(v) & 0 & R_2(v) \\ 0 & 0 & I_2 \end{pmatrix} \qquad P' = \begin{pmatrix} 0 & Q_1(w') & Q_2(w') \\ R_1(v) & 0 & R_2(v) \\ 0 & 0 & I_2 \end{pmatrix}.$$

Note that state 1 corresponds to starting M on the rightmost symbol of w (or w') in the new initial state q_1. State m corresponds to M accepting. Therefore, $a(P)$ (resp. $a(P')$) is the probability that M accepts wv (resp., $w'v$). In particular, $a(P) \geq 1-\varepsilon$ since $wv \in L$. If $t = \max\{t(P), t(P')\}$, then $t \leq T(n)$ since M runs in expected time at most $T(n)$ on input wv or $w'v$. By choice of w and w', the Markov chains P and P' are 2^μ-close mod λ. Therefore, we can apply Lemma 4.2 to conclude

$$\begin{aligned} \Pr[\,M \text{ accepts } w'v\,] &\geq (1-2\lambda m^3)2^{-2\mu m}(1-\varepsilon) - 4\sqrt{\lambda m t} \\ &\geq (1-2/n)2^{-(\log(1-\varepsilon)+1)/2}(1-\varepsilon) - 4\sqrt{1/n} \\ &= (1-2/n)\sqrt{(1-\varepsilon)/2} - 4\sqrt{1/n} \\ &> 1/2. \end{aligned}$$

The final inequality follows by the choice of η_ε and the assumption $n \geq \eta_\varepsilon$. This is a contradiction since M recognizes L and $w'v \notin L$. $\square$

The main result now follows immediately from Theorem 3.6 and Lemma 4.3.

Theorem 4.4 *Let M be a 2pfa which recognizes a nonregular language within expected time $T(n)$. Then there is a constant $b > 0$ such that $T(n) \geq 2^{n^b}$ for infinitely many n.*

5 Turing Machines With Small Space

Recall that if a deterministic or nondeterministic Turing machine has space bound $o(\log\log n)$, then the machine recognizes a regular language [7, Thm. 10.8]. The proof of the main result can easily be extended to prove the same result for probabilistic Turing machines with space bound $o(\log\log n)$, provided that the expected running time is not too large. The definition of a probabilistic Turing machine (ptm) is similar to the definition of a 2pfa, except that the machine has a fixed number of read/write worktapes. The ptm M *runs within space* $S(n)$ if, on any input x with $|x| \leq n$, at most $S(n)$ cells are used on any worktape. A ptm M can be viewed as a 2pfa with a growing number of states: On inputs of length at most n, the 2pfa has at most $c(n) = 2^{dS(n)}$ states, where d is a constant which depends on M. If $S(n) = o(\log\log n)$ then, for any constant $a > 0$, we have $c(n) \leq (\log n)^a$ for all sufficiently large n. The proof of Lemma 4.3 remains valid for 2pfa's with a growing number $c(n)$ of states. Combining this observation with Theorem 3.6, the next result follows by a simple calculation which is left to the reader.

Theorem 5.1 *Let M be a ptm which recognizes a nonregular language within space $o(\log\log n)$ and within expected time $T(n)$. Then for every $b < 1$,*

$$\log\log T(n) \geq (\log n)^b \quad \textit{for infinitely many } n.$$

In particular, $T(n)$ is not bounded above by any polynomial in n.

This result does not hold for larger space bounds. There is a deterministic Turing machine which recognizes a nonregular language within space $O(\log\log n)$ and time $O(n \log n)$ (see [7, Problem 10.2]).

6 Number of States

A measure of the complexity of finite-state automata is the number of states. Some types of automata can be much more concise, that is, use many fewer states, than other types. Previous work has studied conciseness relationships among 1-way and 2-way deterministic and nondeterministic finite-state automata. A machine is *1-way* if the head can move only from left to right. For $j \in \{1,2\}$, a $j$dfa (resp., $j$nfa) is a j-way deterministic (resp., nondeterministic) finite state automaton. For example, Shepherdson [15] shows that any c-state 2dfa can be converted to an equivalent 1dfa having at most $(c+2)^{(c+1)}$ states. Conversely, Meyer and Fischer [11] describe a sequence of regular languages L_c, $c = 1, 2, \ldots$, such that L_c can be recognized by a 2dfa with $5c + 5$ states, and any 1dfa recognizing L_c requires at least c^c states. In this section we make some observations about the conciseness of 2pfa's relative to other types of automata.

Methods of Freivalds [3] can be used to show that there is no recursive function f such that, for any c, if L is a regular language which can be recognized by a c-state 2pfa, then L can be recognized by some 1dfa with at most $f(c)$ states. In [3] it is stated that the emptiness problem for 2pfa's is undecidable. The proof (which is not given in [3] but which is easy to derive from techinques described there) actually shows the following: Any deterministic Turing machine Z can be effectively transformed to a 2pfa M such that (i) if Z halts on blank tape then M recognizes a finite nonempty language L, and (ii) if Z does not halt on blank tape then M recognizes the language $L = \emptyset$. In either case, L is regular. It is decidable, given a 2pfa M and an input word x, whether M accepts x with probability greater than $1/2$. Therefore, if there exists a recursive bound f as above, then the halting problem would be decidable.

With a sufficiently small bound on the expected running time of the 2pfa, a recursive bound f does exist. For simplicity, we focus on 2pfa's which run in polynomial expected time since this is a natural class. Generalizations to somewhat larger time bounds are possible and are left to the interested reader.

Theorem 6.1 *For every $\varepsilon < 1/2$, $a > 0$ and $d > 0$, there exists a constant $b > 0$ such that, for any c, if L is recognized by a c-state 2pfa within error probability ε and within expected time an^d, then L is recognized by some 1dfa with at most c^{bc^2} states.*

Proof. Let L be as in the statement of the theorem. By Theorem 4.4, L is regular. Let k be the maximum of $N_L(n)$ over all n. Then some 1dfa with k states recognizes L (cf. the proof of Lemma 3.1). Let n' be such that $N_L(n'-1) < N_L(n') = k$. By Lemma 3.5, $n' \leq 2k + k^2$. Let $n = 2k + k^2$, so $N_L(n) = k$ since N_L is nondecreasing. If $n < \eta_\varepsilon$ where η_ε is the constant of Lemma 4.3, then $k < \eta_\varepsilon$, so we can choose b large enough to make the theorem true in this case. Assuming $n \geq \eta_\varepsilon$ and noting that $n \leq 3k^2$, the inequality (7) becomes

$$(\alpha_\varepsilon c(\log(a(3k^2)^d) + \log(3k^2c)))^{c^2} \geq k. \tag{9}$$

A routine calculation shows that (9) implies $k \leq c^{bc^2}$ for some b depending only on α_ε, a, and d. $\square$

A large part of the exponential blow-up in Theorem 6.1 is necessary in the worst case since, as noted above, 2dfa's can be exponentially more concise than 1dfa's [11]. To study the effect of randomization on the conciseness of 2-way automata, we must compare 2pfa's with 2dfa's or 2nfa's. The next result gives a lower bound on the improvement possible to the state bound c^{bc^2} in Theorem 6.1. Specifically, c^2 in the exponent cannot be replaced by a function of c which grows more slowly than $\sqrt{c/\log c}$, even if 1dfa is replaced by 2nfa.

The proof of this result uses a random walk technique of Greenberg and Weiss [5] which is also used in the next section. Let w be a nonempty word. A 2pfa M executes the procedure *RW*(w) as follows. M begins with its head on the leftmost symbol of w. It then performs a random walk, moving the head right with probability $1/2$ or left with probability $1/2$ at each step. The procedure terminates when either (i) the head moves off the left end of w (outcome 0), or (ii) the head scans the rightmost symbol of w (outcome 1). It is known (see, for example, [14, Section 4.2]) that

1. the probability of outcome 1 is $1/|w|$, and

2. if *RW*(w) is run repeatedly until it gives outcome 1, then the expected total number of steps is $O(|w|^2)$.

Theorem 6.2 *1. For every ε with $0 < \varepsilon < 1/2$, there are constants $a, b > 0$ such that, for every integer $m \geq 4$ which is a power of 2, there is a coin-tossing 2pfa M_m having at most $b\log^2 m/\log\log m$ states such that M_m recognizes $\{0^m\}$ within error probability ε and within expected time an^2.*

2. For every m, if the 2nfa M accepts the language $\{0^m\}$, then M has at least m states.

Proof. 1. The proof combines the random walk technique with a method of Freivalds [3].

Let $m = 2^k$. The definition of M_m depends on two positive integer parameters B and D which are chosen below to make the error probability small. Although B and D depend on ε, they are independent of m. Let p_i denote the i^{th} prime. Recall that $p_i = \Theta(i \log i)$. Let N be the smallest number such that $p_1 p_2 \cdots p_N \geq Dm$, and note that $N = O(\log m / \log\log m)$.

The computation of M_m on input 0^n has three parts. The first two parts are deterministic.

First, check that $n \geq \log m$, and reject if not.

Second, check that $n \equiv m \bmod p_i$ for all $1 \leq i \leq N$, and reject if not.

If 0^n passes the first two tests, then either $n = m$ or $n \geq Dm$. The first test needs $O(\log m)$ states and takes time $O(n)$. The second test needs $O(\sum_{i=1}^{N} p_i)$ states. Since $p_i = O(\log m)$ for $i \leq N$, the number of states is $O(\log^2 m / \log\log m)$. The second part takes time $O(Nn)$. Since $n \geq \log m$ at this point, the time is $O(n^2)$.

The third part involves two probabilistic procedures $CF(k)$ and $RW(0^n)$. The random walk procedure $RW(0^n)$ is described above. To run $CF(k)$, M_m flips a fair coin k times; if all flips produce "heads," then the outcome is 1, otherwise it is 0. Note that

$$\Pr[\, CF(k) = 1\,] = 2^{-k} = m^{-1},$$

$$\Pr[\, RW(0^n) = 1\,] = n^{-1}.$$

Let C_1 and C_2 be two counters which are initially 0. We can now describe the third test:

do until $C_1 + C_2 = B$
  begin
  if $CF(k) = 1$ then $C_1 \leftarrow C_1 + 1$;
  if $RW(0^n) = 1$ then $C_2 \leftarrow C_2 + 1$;
  end
if $C_1 > 0$ and $C_2 > 0$ then accept, else reject.

If $n = m$, then $\Pr[\, CF(k) = 1\,] = \Pr[\, RW(0^n) = 1\,]$. Therefore, for large B it is extremely unlikely (probability approaching 0 as $B \to \infty$) that one counter will be incremented B times before the other counter is incremented once. Fix B sufficiently large. If $n \neq m$, then $n \geq Dm$, so $\Pr[\, CF(k) = 1\,] \geq D \cdot \Pr[\, RW(0^n) = 1\,]$. Therefore, for large D, it is extremely likely (probability approaching 1 as $D \to \infty$) that C_1 will reach value B before C_2 is incremented once.

It is easy to see that the third part needs $O(\log m)$ states and takes expected time $O(n^2)$.

2. The proof is straightforward, and we only sketch the idea. Fix an m. Assume that the 2nfa M accepts the input 0^n iff $n = m$. Let the sequence $\{\, (s_t, h_t) \mid 1 \leq t \leq T\,\}$ be an accepting computation of M on input 0^m. The pair (s_t, h_t) gives the state s_t and head position h_t of M at time t. Head positions h_t satisfy $0 \leq h_t \leq m+1$; in particular, $h_t = 0$ (resp., $h_t = m+1$) corresponds to reading the left (resp., right) endmarker. A *right pass* (resp., *left pass*) is a time interval $[y, z]$ such that $h_y = 0$ and $h_z = m+1$ (resp., $h_y = m+1$ and $h_z = 0$) and $1 \leq h_t \leq m$ for all t with $y < t < z$.

Assume that M has fewer than m states. For each right pass $[y, z]$, we can find times i and j with $y < i < j < z$ such that $s_i = s_j$ and $h_i < h_j$. The situation for a left pass is the same, except that $h_i > h_j$. Call $|h_i - h_j|$ the *period* of the pass (if a pass has more than one period, then choose one arbitrarily). It is not hard to see that if $n > m$ and n is congruent to m modulo the period of every pass, then M accepts 0^n, contradicting the requirement that M should accept 0^n iff $n = m$. □

7 Unbounded Error Probability

The proof of Lemma 4.3 uses the assumption that the error probability ε is bounded below 1/2. Another model of probabilistic computation which has been considered in the literature (e.g., [4]) allows the error probability to approach 1/2 as n increases. For a 2pfa M, define

$$L(M) = \{\, x \mid \Pr[\, M \text{ accepts } x \,] > 1/2 \,\}.$$

In the next result we observe that the main result, Theorem 4.4, does not hold in the unbounded error model.

Theorem 7.1 *There is a coin-tossing 2pfa M which runs in expected time $O(n^2)$ such that $L(M)$ is not a regular language.*

Proof. We describe a 2pfa M such that

$$L(M) = \{\, 0^a 1^b \mid 1 \le a \le b \,\}.$$

On input x, M first checks that $x = 0^a 1^b$ for some positive a and b. M then runs the following program until it halts:

1. if $RW(0^a) = 1$, then accept;
2. if $RW(1^b) = 1$, then reject, else go to 1.

Let

$$\begin{aligned} p_a &= \Pr[\, RW(0^a) = 1 \,] = 1/a \\ p_b &= \Pr[\, RW(1^b) = 1 \,] = 1/b. \end{aligned}$$

For $i \in \{1, 2\}$, let q_i be the probability that the program accepts when it is started at statement i. Since

$$\begin{aligned} q_1 &= p_a + (1 - p_a) q_2, \\ q_2 &= (1 - p_b) q_1, \end{aligned}$$

we have

$$\Pr[\, M \text{ accepts } 0^a 1^b \,] = q_1 = \frac{p_a}{p_a + p_b - p_a p_b}.$$

Substituting $p_a = 1/a$ and $p_b = 1/b$, it is easy to check that this probability is greater than 1/2 iff $a \le b$. □

References

[1] C. Dwork and L. Stockmeyer, Interactive proof systems with finite state verifiers, Report RJ 6262, IBM Research Division, San Jose, CA, May 1988.

[2] R. Freivalds, Probabilistic machines can use less running time, Information Processing '77, IFIP, North-Holland, 1977, pp. 839–842.

[3] R. Freivalds, Probabilistic two-way machines, Proc. International Symposium on Mathematical Foundations of Computer Science, Lecture Notes in Computer Science, Vol. 118, Springer-Verlag, New York, 1981, pp. 33–45.

[4] J. Gill, Computational complexity of probabilistic Turing machines, SIAM J. Comput., 6 (1977), pp. 675–695.

[5] A. G. Greenberg and A. Weiss, A lower bound for probabilistic algorithms for finite state machines, J. Comput. Syst. Sci., 33 (1986), pp. 88–105.

[6] G. H. Hardy, J. E. Littlewood, and G. Pólya, Inequalities, 2nd ed., Cambridge University Press, Cambridge, England, 1959.

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

[8] R. E. Ladner, R. J. Lipton, and L. J. Stockmeyer, Alternating pushdown and stack automata, SIAM J. Comput., 13 (1984), pp. 135–155.

[9] F. T. Leighton and R. L. Rivest, The Markov chain tree theorem, Report MIT/LCS/TM-249, Laboratory for Computer Science, MIT, Cambridge, MA, 1983.

[10] F. T. Leighton and R. L. Rivest, Estimating a probability using finite memory, IEEE Trans. Information Theory, IT-32 (1986), pp. 733–742.

[11] A. R. Meyer and M. J. Fischer, Economy of description by automata, grammars, and formal systems, Conference Record 12th IEEE Symp. on Switching and Automata Theory, 1971, pp. 188–191.

[12] M. O. Rabin, Probabilistic automata, Information and Control, 6 (1963), pp. 230–245.

[13] M. O. Rabin and D. Scott, Finite automata and their decision problems, IBM J. Research and Development, 3 (1959), pp. 114–125.

[14] E. Seneta, Non-Negative Matrices and Markov Chains, 2nd Ed., Springer-Verlag, New York, 1981.

[15] J. C. Shepherdson, The reduction of two-way automata to one-way automata, IBM J. Research and Development, 3 (1959), pp. 198–200.