Lower Space Bounds for Randomized Computation

Rūsiņš Freivalds
Dept. of Computer Science
University of Latvia
LV-1459 Riga

Marek Karpinski[*]
Dept. of Computer Science
University of Bonn
53117 Bonn

Abstract

It is a fundamental open problem in the randomized computation how to separate different randomized time or randomized small space classes (cf., e.g., [KV 87], [KV 88]). In this paper we study lower space bounds for randomized computation, and prove lower space bounds up to $\log n$ for the specific sets computed by the Monte Carlo Turing machines. This enables us for the first time, to separate randomized space classes below $\log n$ (cf. [KV 87], [KV 88]), allowing us to separate, say, the randomized space $\mathcal{O}(1)$ from the randomized space $\mathcal{O}(\log^* n)$. We prove also lower space bounds up to $\log \log n$ and $\log n$, respectively, for specific languages computed by probabilistic Turing machines, and one–way probabilistic Turing machines.

[*]Research partially supported by the International Computer Science Institute, Berkeley, California, by the DFG Grant KA 673/4–1, ESPRIT BR Grant 7097, and by the Volkswagen Stiftung.

1 Introduction

The advantages of using randomization in the design of algorithms have become increasingly evident in the last couple of years. It appears now that these algorithms are more efficient than the purely deterministic ones in terms of running time, hardware size, circuits depth, etc. The advantages of randomized Turing machines over deterministic machines have been studied early starting with [Fr 75] where the sets of palindromes were proved to be computed by Monte Carlo off-line Turing machines much faster than by the deterministic machines of the same type. Later similar results were obtained for space and reversal complexity for various types of machines [Fr 83, Fr 85, KF 90]. On the other hand, it is universally conjectured that randomness do not always help. However, these conjectures were not supported by proofs since proving lower bounds for randomized machines had turned out to be much harder than proving lower bounds for deterministic and nondeterministic machines.

In this paper we prove nontrivial small lower space bounds for various types of randomized Turing machines.

We distinguish between two types of randomized machines: Monte Carlo machines and probabilistic machines.

We say that a Monte Carlo machine $\mathcal{M}$ recognizes language L in space $S(n)$ if there is a positive constant δ such that:

1. for arbitrary $x \in L$, the probability of the event " $\mathcal{M}$ accepts x in space not exceeding $S(|x|)$ " exceeds $1/2 + \delta$,

2. for arbitrary $x \notin L$, the probability of the event " $\mathcal{M}$ rejects x in space not exceeding $S(|x|)$ " exceeds $1/2 + \delta$.

We say that a probabilistic machine $\mathcal{M}$ recognizes language L in space $S(n)$ if:

1. for arbitrary $x \in L$, the probability of the event " $\mathcal{M}$ accepts x in space not exceeding $S(|x|)$ " exceeds $1/2$,

2. for arbitrary $x \notin L$, the probability of the event " $\mathcal{M}$ rejects x in space not exceeding $S(|x|)$ " exceeds $1/2$.

Probabilistic machines are interesting theoretical devices but they are rather remote from practical needs. Hence much more effort has been spent to construct efficient Monte Carlo algorithms. On the other hand, nondeterministic Turing machines with space bound $S(n) \geq \log n$ can be simulated by probabilistic Turing machines in space $const \cdot S(n)$ [Gi 74, Tr 74] but it is conjectured that this may be not true for Monte Carlo Turing machines. Thus no wonder that it had been difficult to prove nontrivial lower space bounds for specific non-diagonal languages recognized by Monte Carlo Turing machines. However we have failed to find in the literature any lower space bounds for specific languages recognized by probabilistic Turing machines as well. The only exception is rather many proofs of languages being nonstochastic, i. e. for rather many languages L it is proved that arbitrary 1-way probabilistic Turing machine recognizing L cannot use constant space only (see monographs [Pa 71, Bu 77]).

It deserves to be mentioned that there have already been lower time bounds ($const \cdot n^2$ for Monte Carlo off-line Turing machines to recognize palindromes [Fr 75, Fr 77]). The essential ideas for lower space bounds for 1-way Monte Carlo Turing machines have been published in [Fr 83, Fr 85] but they have not materialized in any completed lower space bound for specific languages.

2 Randomized Turing Machines

The results in this section are based on a simple idea firstly used by M. Rabin [Ra 63], and then adapted in different contexts by A. Greenberg and A. Weiss [GW 86], R. Freivalds [Fr 79], C. Dwork and L. Stockmeyer [DS 88, DS 92]. Let $\mathcal{M}$ be a randomized Turing machine. Configuration (instanteous description) of the machine at a definite moment of the work of the machine shows:

(i) the internal state,

(ii) the positions of the heads on the work tapes (but not the position of the head on the input tape),

(iii) the content of the work-tapes at this moment.

We define the word probabilities of $\mathcal{M}$ on w as follows. A starting condition is a pair $(conf, \xi)$ where *conf* is a configuration of $\mathcal{M}$ and $\xi \in \{Left, Right\}$; its intuitive meaning is "start $\mathcal{M}$ on the ξ end of w in configuration *conf*". A stopping condition is either:

1. a pair $(conf, \xi)$ as above, meaning "the input head of $\mathcal{M}$ falls off the ξ end of w with $\mathcal{M}$ in configuration *conf*",

2. "*Loop*" meaning "the computation of $\mathcal{M}$ loops forever within w",

3. "*Accept*" meaning "$\mathcal{M}$ halts in state q_a before the input head falls off either end of w", or

4. "*Reject*" meaning "$\mathcal{M}$ halts in state q_r before the input head falls off either end of w".

For each starting condition σ and each stopping condition τ, let $p(w, \sigma, \tau)$ be the probability that stopping condition occurs given that $\mathcal{M}$ is started in starting condition σ on w.

Since we model computations of Turing machines by Markov chains, we first give some definitions and results about Markov chains. Basic facts about Markov chains with finite state space can be found, for example, in [KS 60]. We consider Markov chains with finite state space, say $1, 2, \ldots, s$ for same s. A particular Markov chain is completely specified by its matrix $R = \{r_{ij}\}_{i,j=1}^{s}$ of transition probabilities. If the Markov chain is in state i, then it next moves to state j with probability r_{ij}. The chains we consider have a designated starting state, say, state 1, and some set T of trapping states, so $r_{kk} = 1$ for all $k \in T$. For $k \in T$, let $a(k, R)$ denote the probability that the Markov chain R is trapped in state k when started in state 1.

We start with a lemma which bounds the effect of small changes in the transition probabilities of a Markov chain. This lemma has been taken from [DS 92] with a reference to Lemma 1 from [GW 86] which was however slightly different.

Let $\beta \geq 1$. Say that two numbers r and r' are β-close if either (i) $r = r' = 0$ or (ii) $r > 0$, $r' > 0$, and $\beta^{-1} \leq r/r' \leq \beta$. Two Markov chains $R = \{r_{ij}\}_{i,j=1}^{s}$ and $R' = \{r'_{ij}\}_{i,j=1}^{s}$ are β-close if r_{ij} and r'_{ij} are β-close for all pairs i, j.

Lemma 2.1 ([DS 92]) *Let R and R' be two s-state Markov chains which are β-close, and let k be a trapping state of both R and R'. Then $a(k, R)$ and $a(k, R')$ are β^{z}-close where $z = 2s$.*

Theorem 2.2 *Let $A, B \subseteq \Sigma^*$ with $A \cap B = \emptyset$. Suppose there is an infinite set I of positive integers and functions $G(m), H(m)$ such that $G(m)$ is a fixed polynomial in m, and for each $m \in I$ there is a set W_m of words in Σ^* such that:*

1. *$|w| \leq G(m)$ for all $w \in W_m$,*
2. *there is a constant $c > 1$ such that $|W_m| \geq c^m$ for all $m \in I$,*
3. *for every $m \in I$ and every $w, w' \in W_m$ with $w \neq w'$, there are words $u, v \in \Sigma^*$ such that:*

   (a) *$|uwv| \leq H(m), |uw'v| \leq H(m)$, and*

   (b) *either* $\begin{cases} uwv \in A \\ uw'v \in B \end{cases}$

   or $\begin{cases} uwv \in B \\ uw'v \in A \end{cases}$

Then, if a Monte Carlo 2-way Turing machine with space bound $S(n)$ separates A and B, then $S(H(m))$ cannot be $o(\log m)$.

Proof. Suppose that the Monte Carlo 2-way Turing machine separates A and B with error probability $\epsilon < \frac{1}{2}$. Let $S(n)$ be the space function for $\mathcal{M}$. By $Vol(n)$ we denote the number of the possible configurations of the machine $\mathcal{M}$ on words of length not exceeding n. It is obvious that $Vol(n) \leq O(\exp(S(n)))$.

Suppose to the contrary that $S(H(m)) = o(\log m)$ and $Vol(H(m)) = = 2^{o(\log m)}$. Consider the word probabilities $p(v, \sigma, \tau)$ defined above. We restrict ourselves to words v of length not exceeding $G(m)$ only. Formally, the length of v and the length of the input word (which is essential to compute the value of the functions $S(n)$ and $Vol(n)$) are not related. However for our considerations it suffices to consider the total length of words no more that $H(m)$. Hence for arbitrary word v we consider $d = 4(Vol(H(m)))^2 + 6Vol(H(m))$ word probabilities.

Fix some ordering of the pairs (σ, τ) of starting and stopping conditions involving the conditions with space not exceeding $S(H(m))$. Let $p(v)$ be the vector of these d probabilities according to this ordering.

We first show that if $|v| \leq m$ and if p is a nonzero element of $p(v)$, then $p \geq 2^{-Vol(H(m))G(m)}$. Form a Markov chain $K(v)$ with states of the form (conf, l) where *conf* is a configuration of $\mathcal{M}$ using no more space than $S(H(m))$, and $0 \leq l \leq |v| + 1$. The chain state (conf, l) with $1 \leq l \leq |v|$ corresponds to $\mathcal{M}$ being in configuration *conf* with the input tape head scanning the lth symbol of v. Transition probabilities from such states are obtained from the transition probabilities of $\mathcal{M}$ in the obvious way. For example, if the lth symbol of v is 0, and if $\mathcal{M}$ in configuration *conf* reading a 0 can move the input head left and enter configuration *conf′* with probability $1/2$, then the transition probability from state (conf, l) to state $(\mathit{conf}', l-1)$ is $1/2$. Chain states of the form $(\mathit{conf}, 0)$ and $(\mathit{conf}, |v| + 1)$ are trap sates of $K(v)$ and correspond to the input head of $\mathcal{M}$ falling off the left or right end, respectively, of v. Now consider, for example, $p = p(v, \sigma, \tau)$ where $\sigma = (\mathit{conf}_i, \mathit{Left})$ and $\tau = (\mathit{conf}_j, \mathit{Left})$. If $p > 0$, then there must be some path on nonzero probability in $K(v)$ from state $(\mathit{conf}_i, 1)$ to $(\mathit{conf}_j, 0)$ and since $K(v)$ has at most $Vol(H(m)) \cdot |v| \leq Vol(H(m)) \cdot G(m)$ nontrapping states, there is such a path of length at most $Vol(H(m)) \cdot G(m)$. Since $1/2$ is the smallest nonzero transition probability of $\mathcal{M}$, it follows that $p \geq 2^{-Vol(H(m)) \cdot G(m)}$. The other three cases $p(v, \sigma, \tau)$ where τ has the form (conf, ξ) are similar. If $\sigma = (\mathit{conf}, \mathit{Left})$ and $\tau = \mathit{Loop}$, there must be a path of nonzero probability in $K(v)$ from state $(\mathit{conf}, 1)$ to some state (conf', l) such that there is no path of nonzero probability from (conf', l) to any trap state of the form $(\mathit{conf}'', 0)$ or $(\mathit{conf}'', |v| + 1)$. Again, if there is such a path, there is one of lenght at most $Vol(H(m)) \cdot G(m)$. The remaining cases are similar.

Fix an arbitrary $m \in I$. Divide W_m into equivalence classes by making w and w' equivalent if $p(w)$ and $p(w')$ are zero in exactly the same coordinates. Let E_m be a largest equivalence class, so $|E_m| \geq |W_m|/2^d$.

Let d' be the number of nonzero coordinates of $p(w)$ for $w \in E_m$. Let $\hat{p}(w)$ be the d'-dimensional vector of nonzero coordinates of $p(w)$. Note that $\hat{p}(w) \in [2^{-Vol(H(m)) \cdot G(m)}, 1]^{d'}$ for all $w \in E_m$. Let $\log \hat{p}(w)$ be the componentwise log of $\hat{p}(w)$, so $\log \hat{p}(w) \in [-Vol(H(m)) \cdot G(m), 0]^{d'}$.

By dividing each coordinate interval $[Vol(H(m)) \cdot G(m), 0]$ into subintervals of length μ, we divide space $[Vol(H(m)) \cdot G(m), 0]^{d'}$ into at most $(Vol(H(m)) \cdot G(m)/\mu)^d$ cells, each of size $\mu \times \mu \times \cdots \times \mu$. We want to choose

μ large enough that the number of cells is smaller than the size of E_m, that is

$$\left(\frac{Vol(H(m))\cdot G(m)}{\mu}\right)^d < \frac{|W_m|}{2^d},$$

or, equivalently,

$$2^{4(Vol(H(m)))^2+6Vol(H(m))}\times$$

$$\times\left(\frac{Vol(H(m))\cdot G(m)}{\mu}\right)^{4(Vol(H(m)))^2+6Vol(H(m))} < |W_m| \qquad (1)$$

Since $|W_m|$ is assumed to grow faster than any polynomial in m, $G(m)$ is a polynomial in m, and since, by assumption from the contrary, $Vol(H(m)) = 2^{o(\log m)}$, for arbitrary $\mu > 0$ there is an m_μ such that (1) holds for all $m \in I$ with $m \geq m_\mu$.

Assuming (1), there must be two different words $w, w' \in E_m$ such that $\log \hat{p}(w)$ and $\log \hat{p}(w')$ belong to the same cell. Therefore, if p and p' are two nonzero probabilities in the same coordinate of $p(w)$ and $p(w')$, respectively, then

$$|\log p - \log p'| \leq \mu$$

It follows that p and p' are 2^μ-close. Therefore, $p(w)$ and $p(w')$ are componentwise 2^μ-close.

For this pair (w, w'), let u and v be the words in Assumption 3 in the statement of the theorem. We describe two Markov chains, R and R', which model the computation of $\mathcal{M}$ on uwv and $uw'v$, respectively. Both chains have $4 \cdot Vol(H(m)) \cdot G(m) + 4$ states. $4 \cdot Vol(H(m)) \cdot G(m)$ of these states have the form (conf, l) where *conf* is a configuration of $\mathcal{M}$ and $l \in \{1, 2, 3, 4\}$. The other states are *Initial, Accept, Reject* and *Loop*. The state (conf, l) of R corresponds to $\mathcal{M}$ being in cofiguration *conf* reading the right end of $¢u$ if $l = 1$, the left end of w if $l = 2$, the right end of w if $l = 3$, or the left end of $v¢$ if $l = 4$. The state *Initial* corresponds to $\mathcal{M}$ being in its initial state q_0 reading the leftmost endmarker $¢$, the states *Accept* and *Reject* correspond to the computation halting in the accepting state or the rejecting state, respectively, and *Loop* means that $\mathcal{M}$ has entered an infinite loop. The transition probabilities of R are obtained from the word probabilities of $\mathcal{M}$ on $¢u$,w and $v¢$. For example, the transition probability from $(\mathit{conf}_i, 3)$ to $(\mathit{conf}_j, 1)$ is just $p(w, (\mathit{conf}_i, \mathit{Right}), (\mathit{conf}_j, \mathit{Left}))$, the transition probability from *Initial* to $(\mathit{conf}_j, 2)$ is $p(¢u, (\mathit{conf}_{\mathit{Initial}}, \mathit{Left}), (\mathit{conf}_j, \mathit{Right}))$ and the transition probability from $(\mathit{conf}_i, 4)$ to *Accept* is $p(v¢, (\mathit{conf}_i, \mathit{Left}), \mathit{Accept})$. The states *Accept, Reject* and *Loop* are trap states of R. The chain R' is defined similarly, but using w' in place of w.

Suppose that $uwv \in A$ and $uw'v \in B$, the other case being symmetric. Let $z = 2(4 \cdot (Vol(H(m)) \cdot G(m) + 4)$. Let a (resp., a') be the probability that $\mathcal{M}$ accepts input uwv (resp., $uw'v$). Then a (resp., a') is exactly the probability that the Markov process R (resp., R') is trapped in state *Accept* when started in state *Initial*. Now $uwv \in A$ implies $a \geq 1 - \epsilon$. Since R and R' are 2^μ-close, Lemma 2.1 implies that

$$\frac{a'}{a} \geq 2^{-\mu z}$$

which implies

$$a' \geq (1-\epsilon) \cdot 2^{-\mu z}$$

Now we are ready to put our arguments together. Take μ so small that

$$(1-\epsilon) \cdot 2^{-\mu(8 \cdot Vol(H(m)) \cdot G(m)+8)} > 1/2 \tag{2}$$

Then take sufficiently large μ to ensure that (1) holds. Choose two different but 2^μ-close words in E_m. Then R and R' are 2^μ-close and (2) holds. But since $uw'v \in B$, this contradicts the assumption that μ separates A and B. □

Example 2.3. Consider the language $PAL \subset \{0,1\}^*$ consisting of all the palindromes.

Corollary of Theorem 2.2. *If a Monte Carlo 2-way Turing machine with space bound $S(n)$ recognizes PAL, then $S(n)$ cannot be $o(\log n)$.*

Notice that there exists a deterministic 2-way Turing machine recognizing PAL in space $\log n$.

Example 2.4. Consider the language D_2 containing strings of balanced parentheses of 2 types. This language is generated by the context-free grammar with productions: $S \to ()$, $S \to []$, $S \to SS$, $S \to (S)$, $S \to [S]$.

Corollary of Theorem 2.2. *If a Monte Carlo 2-way Turing machine with space bound $S(n)$ recognizes D_2, then $S(n)$ cannot be $o(\log n)$.*

3 Separation Theorem

Theorem 3.1 *Let $g(n)$ be arbitrary self-constructible space function for Monte Carlo 2-way Turing machines, $g(n) \leq \log n$. Then there is a language L_g such that:*

1. *L_g can be recognized by a $g(n)$-space bounded Monte Carlo 2-way Turing machine,*

2. *L_g cannot be recognized by a $h(n)$-space bounded Monte Carlo 2-way Turing machine, where $h(n) = o(g(n))$.*

Proof. L_g consists of words $w \in \{0,1,2,3,4\}^*$ in the form $w = v22\ldots233\ldots344\ldots4$, where:

(i) v is a palindrome in $\{0,1\}^*$,

(ii) the number of 2's equals the length of v,

(iii) if k denotes the number of 3's then the number of 2's equals 2^k,

(iv) the number of 3's equals $g(|w|)$.

The Monte Carlo Turing machine asserted in 1) first constructs the function $g(n)$, compares it with the number of 3's. Then the machine deterministically recognizes whether or not the assertion (iii) holds. This can be done in space k. Finally, the machine deterministically recognizes whether the assertions (i)–(ii) hold. No more than logarithmic (in $|v|$) space is needed for this.

The Assertion 2) is a corollary from Theorem 2.2. □

M. Karpinski and R. Verbeek [KV 87] have shown that there are many small functions which are self–constructible for space complexity of 2-way Monte Carlo Turing machines. Among these functions one should mention $\log\log\ldots\log n$ (repeated arbitrarily many times), $\log^* n$, the inverse Ackermann function. It follows from our Theorem that the corresponding complexity classes are pairwise different. For instance, there is a language recognizable by 2-way Monte Carlo Turing machines in space $\log^* n$ but not in space equal to the inverse Ackerman function (For the related problems of randomized time bounded computation cf. [KV 93]).

4 1-way Monte Carlo machines

Consider a language $L \subseteq \Sigma^*$. We say that the words u and v are equivalent with respect to L if and only if $(\forall w)(uw \in L \Leftrightarrow vw \in L)$. Rank of the language L is the function $rank_L(n)$ expressing the number of non-equivalent words among all the words in $\Sigma^{\leq n}$.

Theorem 4.1 *If a Monte Carlo 1-way Turing machine with space bound $S(n)$ recognizes L, then $S(n)$ cannot be $o(\log\log rank_L(n)$.*

Proof. Fix some ordering of the configurations of the machine $\mathcal{M}$ such that the lengths of used work-tapes do not decrease. Let $Vol(n)$ be the number of possible configurations of $\mathcal{M}$ with the length of work-tape not exceeding $S(n)$. It is ovbious that $Vol(n) \leq O(\exp(S(n)))$.

Let p_x be the $Vol(n)$-dimensional vector of the probabilities of the corresponding configurations reached by $\mathcal{M}$ after processing the input word x, The total of these probabilities may be less than 1 since with small probability longer configurations nay be obtained. The vector p_v may be interpreted as a point in a $Vol(n)$-dimensional unit cube. We introduce metrics

$$\rho(p_x, p_y) = |p_x(conf_1) - p_y(conf_1)| + \cdots + |p_x(conf_{Vol(n)}) - p_y(conf_{Vol(n)})|.$$

Lemma 2. 3 in [Fr 85] asserts that there is a positive constant c such that if x and y are not Myhill–Nerode equivalent with respect to L, then $\rho(p_x, p_y) \geq c$. Let $x_1, x_2, \ldots, x_r$ be all possible words in $\Sigma^{\leq n}$ pairwise non-equivalent with respect to L ($r = rank_L(n)$). Consider the bodies defined by the equations $\rho(p_x - p_{x_i}) < \frac{c}{2}$. These bodies do not intersect. Their volumes equal

$$\frac{2^{Vol(n)} \cdot (\frac{c}{2})^{Vol(n)}}{(Vol(n))!} = \frac{c^{Vol(n)}}{(Vol(n))!}$$

These bodies are situated in a cube with the length of edge $1 + 2c$. Hence the number of the bodies cannot exceed

$$\frac{(1+2c)^{Vol(n)} \cdot (Vol(n))!}{c^{Vol(n)}} = 2^{O(Vol(n) \cdot \log Vol(n))}$$

and

$$rank_L(n) \leq 2^{O(Vol(n) \cdot \log Vol(n))}$$

$$\log rank_L(n) \leq O(Vol(n) \cdot \log Vol(n))$$

$$O\left(\frac{\log rank_L(n)}{\log\log rank_L(n)}\right) \leq Vol(n)$$

$$O(\log\log rank_L(n)) \leq S(n) \qquad \square$$

5 1-way probabilistic machines

Theorem 5.1 *Let $A, B \subseteq \Sigma^*$ with $A \cap B = \emptyset$. Suppose there is an infinite set I of positive integers and a function $H(m)$ such that for each $m \in I$ there is an ordered set of pairs of words $W_m = \{(u_1, v_1), (u_2, v_2), \ldots, (u_m, v_m)\}$ such that for every string $\alpha(1)\alpha(2)\ldots\alpha(m) \in \{0,1\}^m$, there is a word w such that*

$$\begin{cases} u_i w v_i \in A, & \textit{if } \alpha(i) = 1, \\ u_i w v_i \in B, & \textit{if } \alpha(i) = 0. \end{cases}$$

and $|u_i w v_i| \leq H(m)$ for all $i \in \{1, 2, \ldots, m\}$.

Then, if a 1-way probabilistic Turing machine with space bound $S(n)$ separates A and B, then $S(H(m))$ cannot be $o(\log m)$.

Proof. Assume the contrary. Let $\mathcal{M}$ be a probabilistic 1-way Turing machine with the acceptance probability $p(x) > \lambda$ if $x \in A$ and $p(x) < \lambda$ if $x \in B$, with $S(H(m)) = o(\log m)$ which implies $Vol(H(m)) = 2^{o(\log m)}$.

Enumerate all the configurations of $\mathcal{M}$ using no more space than y. Denote the number of possible configurations of $\mathcal{M}$ using no more than y space by Y. It is obvious that $(\exists c > 0)(Y \leq c^y)$.

Denote by a_{ij} the transition probability from configuration i to configuration j when $\mathcal{M}$ processes the input word u. Similarly, denote by b_{ij} and c_{ij} the transition probabilities when $\mathcal{M}$ processes w and v, respectively. If we neglect the configurations using space exceeding y, then there is only a finite number of configurations and the probability $p(x)$ for $x = uwv$ equals

$$(\delta_1, \ldots, \delta_{1Y}) \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1Y} \\ a_{21} & a_{22} & \ldots & a_{2Y} \\ - & - & - & - \\ a_{Y1} & a_{Y2} & \ldots & a_{YY} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & \ldots & b_{1Y} \\ b_{21} & b_{22} & \ldots & b_{2Y} \\ - & - & - & - \\ b_{Y1} & b_{Y2} & \ldots & b_{YY} \end{pmatrix} \times$$

$$\times \begin{pmatrix} c_{11} & c_{12} & \ldots & c_{1Y} \\ c_{21} & c_{22} & \ldots & c_{2Y} \\ - & - & - & - \\ c_{Y1} & c_{Y2} & \ldots & c_{YY} \end{pmatrix} \begin{pmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_Y \end{pmatrix}$$

Our proof is based heavily on the simple observation that $p(x)$ may be expressed as a linear form of the products $\delta_i a_{ij} b_{jk} c_{kl} \eta_l$. Hence for fixed words u, v the value $p(x)$ is expressed as a linear form of the values $b_{11}, \ldots, b_{YY}$. These linear forms may be considered as a linear Y^2-dimensional space. The linear dependence of any $(Y^2 + 1)$ vectors in an Y^2-dimensional linear space implies that there are numbers $c_1, \ldots, c_{Y^2+1}$ which are not all equal to 0, and there are $Y^2 + 1$ pairs $(u_1, v_1), (u_2, v_2), \ldots, (u_{Y^2+1}, v_{Y^2+1})$ such that, for arbitrary w,

$$c_1 \cdot p(u_1 w v_1) + c_2 p(u_2 w v_2) + \cdots + c_{Y^2+1} p(u_{Y^2+1} w v_{Y^2+1}) = 0 \qquad (3)$$

and

$$c_1 + c_2 + \cdots + c_{Y^2+1} = 0 \qquad (4)$$

Let $c_{i_1}, c_{i_2}, \ldots c_{i_l}$ be all positive numbers in this set. By Assumption (2) of the Theorem, for every string $\alpha(1)\alpha(2)\ldots\alpha(Y^2+1) \in \{0,1\}^{Y^2+1}$ there is a word w such that

$$\begin{cases} u_i w v_i \in A, & \text{if } \alpha(i) = 1, \text{and} \\ u_i w v_i \in B, & \text{if } \alpha(i) = 0. \end{cases}$$

Take $\alpha(i) = 1$ if and only if $c_i > 0$. Then $p(u_i w v_i) > \lambda$ if and only if $\alpha(i) = 1$. Hence from (4) it follows

$$c_1 \cdot p(u_1 w v_1) + \cdots + c_{Y^2+1} p(u_{Y^2+1} w v_{Y^2+1}) =$$

$$= c_1 (p(u_1 w v_1) - \lambda) + \cdots + c_{Y^2+1} \cdot (p(u_{Y^2+1} w v_{Y^2+1}) - \lambda) > 0 \qquad (5)$$

Now observe that the lengths of all words $u_i w v_i$ do not exceed $H(Y^2 + 1)$. Hence the space used by $\mathcal{M}$ on these words does not exceed $S(H(Y^2 + 1)) \leq y$ (because, by the contrary, $S(H(m)) = o(\log m)$). Contradiction between (5) and (3). □

Consider the language *NH* defined by M. Nasu and N.Honda [NH 71]. It is the set of words over an alphabet $\{a, b\}$ of the form $a^i b a^{j_1} b \ldots b a^{j_r} b$ $(r = 1, 2, \ldots)$ such that for some $1 \leq l \leq r, i = j_1 + \cdots + j_l$ holds, where $i, j_1, j_2, \ldots, j_r$ are nonnegative integers.

Corollary 5.2 *If a probabilistic 1-way Turing machine with space bound $S(n)$ recognizes the language* NH, *then $S(n)$ cannot be $o(\log n)$.*

Proof. For arbitrary m, the pairs of words (u_i, v_i) are as follows. $u_i = a^i$, v_i is empty. For the string $\alpha(1)\alpha(2)\ldots\alpha(m)$, let $0 < k_1 < k_2 < \cdots < k_l$ be all the values of i such that $\alpha(i) = 1$. Let $j_1, j_2, \ldots, j_l$ be positive integers such that, for every $1 \leq s \leq l, j_1 + \cdots + j_s = k_s$. Then the word w corresponding to the string $\alpha_1\alpha(2)\ldots\alpha(m)$ equals $ba^{j_1}ba^{j_2}b\ldots ba^{j_l}b$. It is easy to see that $u_i w v_i \in NH$ if and only if $\alpha(i) = 1$. For all m, $H(m) \leq 3m$. □

It deserves to be noticed that NH can be recognized by a deterministic 1-way Turing machine in log-space as well. Hence, randomness does not help to recognize NH even if we allow non-isolated cut-points.

6 Probabilistic machines

Theorem 6.1 *Let $A, B \subseteq \Sigma^*$ with $A \cap B = \emptyset$. Suppose there is an infinite set I of positive integers and a function $H(m)$ such that for each $m \in I$ there is an ordered set of pairs of words $W_m = \{(u_1, v_1), (u_2, v_2), \ldots (u_m, v_m)\}$ such that for every string $\alpha(1)\alpha(2)\ldots\alpha(m) \in \{0,1\}^m$, there is a word w such that*

$$\begin{cases} u_i w v_i \in A, & \text{if } \alpha(i) = 1, \\ u_i w v_i \in B, & \text{if } \alpha(i) = 0, \end{cases}$$

and $|u_i w v_i| \leq H(m)$ for all $i \in \{1, 2, \ldots, m\}$. Then, if a 2-way probabilistic Turing machine with space bound $S(n)$ separates A and B, then $S(H(m))$ cannot be $o(\log\log m)$.

Proof. It follows from Theorem 5.1 that arbitrary 1-way probabilistic Turing machine separating A from B cannot have space bound $o(\log m)$. J. Kaneps [Ka 89] proved that every language recognizable by a 2-way probabilistic finite automaton with k states can be recognized by a 1-way probabilistic finite automaton with $2^{O(k^2)}$ states as well. This proof can be modified to obtain our result. □

7 Discussion

It may seem that all the lower bounds proved in the paper are based on the same assumption about the given language. The assumptions are indeed related. For instance, if, for a language L, the assumptions of Theorem 5.1 hold, then the assumptions of Theorem 2.2 hold as well. However, our lower bounds show that space complexity features may be different for different sets.

(For the set PAL there are the following space optimal Turing machines:

1-way deterministic TM: linear

2-way deterministic TM: $\log n$

1-way Monte Carlo TM: $\log n$

2-way Monte Carlo TM: $\log n$

1-way probabilistic TM: $const$

2-way probabilistic TM: $const$

For the set NH the space bounds are:

1-way deterministic TM: $\log n$

2-way deterministic TM: $\log n$

1-way Monte Carlo TM: $\log n$

2-way Monte Carlo TM: $\log n$

1-way probabilistic TM: $\log n$

2-way probabilistic TM: ?

For the set

$$\{010^2 2 \ldots 2010^2 10^4 10^8 1 \ldots 10^{2^k} 2010^2 10^4 10^8 1 \ldots 10^{2^{2k}}\}$$

the space bounds are:

1-way deterministic TM: $\log n$

2-way deterministic TM: $\log n$

1-way Monte Carlo TM: $\begin{cases} \geq \log\log n \\ \leq (\log\log n)^2 \end{cases}$

2-way Monte Carlo TM: $const$

1-way probabilistic TM: $const$

2-way probabilistic TM: $const$

) □

Acknowledgements

We thank Eric Allender, Richard Beigel, Johan Håstad, Sasha Razborov, and Rutger Verbeek for the number of interesting discussions on the various issues of the randomized separation and the lower bounds.

References

[ABHH 92] Allender, E., Beigel, R., Hertrampf, U., and Homer, S., *Almost–Everywhere Complexity Hierarchies for Nondeterministic Time,* Manuscript, 1992, A preliminary version has appeared in Proc. STACS'90, LNCS 415, Springer–Verlag, 1990, pp. 1–11

[BCP 83] Borodin, A., Cook, S., and Pippenger, N., *Parallel computation for wellendowed rings and space-bounded probabilistic machines,* Information and Control 58, pp. 113–136.

[Bu 77] Bukharaev, R. E., *Probabilistic Automata,* Kazan University Press, 1977 (Russian).

[DS 88] Dwork, C., and Stockmeyer, L., *Interactive proof systems with finite state verifiers,* Res. Rep. RJ6262. IBM Research Division, San Jose, Calif., May 1988.

[DS 92] Dwork, C., and Stockmeyer, L., *Finite state verifiers I: The power of interaction,* Journal of ACM, 39, 4 (Oct. 1992), pp. 800–828.

[Fr 75] Freivalds, R., *Fast computation by probabilistic Turing machines,* Proceedings of Latvian State University, 233 (1975), pp. 201–205 (Russian).

[Fr 77] Freivalds, R., *Probabilistic machines can use less running time,* In: Information Processing'77 (Proc. IFIP Congress'77), North Holland, 1977, pp. 839–842.

[Fr 79] Freivalds, R., *Speeding up recognition of some sets by usage of random number generators,* Problemi kibernetiki, 36 (1979), pp. 209–224 (Russian).

[Fr 83] Freivalds. R., *Space and reversal complexity of probabilistic one-way Turing machines,* Lecture Notes in Computer Science, 158 (1983), pp. 159–170.

[Fr 85] Freivalds, R., *Space and reversal complexity of probabilistic one-way Turing machines,* Annals of Discrete Mathematics, 24 (1985), pp. 39–50.

[Gi 74] Gill, J. T., *Computational complexity of probabilistic Turing machines,* SIAM J. Comput. 6 (1977), pp. 675–694.

[GW 86] Greenberg, A. G. and Weiss A., *A lower bound for probabilistic algorithms for finite state machines,* Journal of Computer and System Science, 33 (1986), pp. 88–105.

[Ka 89] Kaneps, J., *Stochasticity of languages recognized by two-way finite probabilistic automata,* Diskretnaya matematika, 1 (1989), pp. 63–77 (Russian).

[KF 90] Kaneps, J. and Freivalds, R., *Minimal nontrivial space complexity of probabilistic one-way Turing machines,* Lecture Notes in Computer Science, Springer, 452 (1990), pp. 355–361.

[Kr 90] Karp, R. M., *An Introduction to Randomized Algorithms,* Technical Report TR–90–029, International Computer Science Institute, Berkeley, 1990.

[KV 87] Karpinski, M., and Verbeek, R., *On the Monte Carlo space constructible functions and separation results for probabilistic complexity classes,* Information and Computation, 75 (1987), pp. 178–189.

[KV 88] Karpinski, M., and Verbeek, R., *Randomness, Probability, and the Separartion of Monte Carlo Time and Space,* LNCS 270, Springer–Verlag, 1988, pp. 189–207.

[KV 93] Karpinski, M. , and Verbeek, R., *On Randomized versus Deterministic Computation,* Proc. ICALP '93, LNCS 700 (1993), Springer–Verlag, pp. 227–240.

[KS 60] Kemeny, J. G., and Snell, J. L., *Finite Markov Chains,* Van Nostrand, 1960.

[NH 71] Nasu, M. and Honda, N., *A context-free language which is not acceptable by a probabilistic automaton,* Information and Control, 18 (1971), pp. 233–236.

[Pa 71] Paz, A., *Introduction to Probabilistic Automata,* Academic Press, 1971.

[Ra 63] Rabin, M. O., *Probabilistic automata,* Information and Control, 6 (1963), pp. 230–245.

[Tr 74] Trakhtenbrot, B. A., *Notes on the complexity of computation by probabilistic machines,* In: Theory of Algorithms and Mathematical Logics, VC AN SSSR, 1974, pp. 159–176 (Russian).