Lower Time Bounds for Randomized Computation

Rūsiņš Freivalds*and Marek Karpinski†

Abstract

It is a fundamental problem in the randomized computation how to separate different randomized time or randomized space classes (c.f., e.g., [KV87, KV88]). We have separated randomized space classes below $\log n$ in [FK94]. Now we have succeeded to separate small randomized time classes for multi-tape 2-way Turing machines. Surprisingly, these "small" bounds are of type $n + f(n)$ with $f(n)$ not exceeding linear functions. This new approach to "sublinear" time complexity is a natural counterpart to sublinear space complexity. The latter was introduced by considering the input tape and the work tape as separate devices and distinguishing between the space used for processing information and the space used merely to read the input word from. Likewise, we distinguish between the time used for processing information and the time used merely to read the input word.

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

*Institute of Mathematics and Computer Science, University of Latvia, Raina bulv. 29, Riga, Latvia, e-mail: rusins@mii.lu.lv. Research partially supported by Grant No.93-599 from the Latvian Council of Science

†Department of Computer Science, University of Bonn, 53117, Bonn, Germany. e-mail: karpinski@cs.bonn.edu. Research partially supported by the International Computer Science Institute, Berkeley, California, by the DFG grant KA 673/4-1, and by the ESPRIT BR Grants 7079 and ECUS030

these algorithms are more efficient than the purely deterministic in terms of running time, hardware size, circuits depth, etc. The adventages of randomized Turing machines over determinnistic machines have been studied early starting with [Fr75] where the sets of palindromes were proved to be computable by Monte Carlo off-line Turing machines much faster than by deterministic machines of the same type. Later similar results were obtained for space and reversal complexity for various types of machines [Fr83, Fr85, KF90]. On the other hand, it is universally conjectured that randomness do not always help. However, these conjectures usually cannot be supported by proofs since proving lower bounds in always hard, and proving lower bounds for complexity of randomized machines has turned out to be much harder than proving lower bounds for complexity of deterministic and nondeterministic machines.

In [FK94] we proved the first nontrivial small lower space bounds for various types of ranomized machines. In this paper we have proved the first nontrivial small lower time bounds for randomized multitape 2-way Turing machines.

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

We say that a Monte Carlo machine M recognizes language L in time $T(n)$ if there is a positive constant δ such that:

1. For arbitrary $x \in L$, the probability of event "M accepts x in time not exceeding $T(|x|)$" exceeds $1/2 + \delta$,

2. For arbitrary $x \notin L$, the probability of event "M rejects x in time not exceeding $T(|x|)$" exceeds $1/2 + \delta$.

We say that a probabilistic machine M recognizes language L in time $T(n)$ if:

1. For arbitrary $x \in L$, the probability of event "M accepts x in time not exceeding $T(|x|)$" exceeds $1/2$,

2. For arbitrary $x \notin L$, the probability of event "M rejects x in time not exceeding $T(|x|)$" exceeds $1/2$.

In a similar way one defines space complexity of Monte Carlo and probabilistic machines. Probabilistic machines are interesting theoretical devices

but they are rather remote from practical needs. Hence much more effort has been spent to study Monte Carlo machines.

There have already been lower time bounds ($const \cdot n^2$ for Monte Carlo off-line Turing machines to recognize palindromes [Fr75, Fr77]). On the other hand, these lower bounds employ the specific restrictions of the machine model. For multitape 2-way Turing machines, lower time bounds are weak even in the case of deterministic machines. In the case of Monte Carlo machines the situation is much worse. We did not even know whether $MCTIME(n) = MCTIME(n^{\log n})$. Nevertheless we prove in Section 2 the first nontrivial lower time bounds for recognition of specific languages by muultitape 2-way Turing machines. This allows us to prove separation theorem for small time complexity classes of Monte Carlo multitape 2-way Turing machines. The method used in this proof generalizes the methods used in [FI77, JKS84].

2 Separation Theorem

We consider a language

$$A = \{x2y | (\exists m)(x \in \{0,1\}^{2^m} \& y \in \{0,1\}^m \& (\exists i)(y \text{ is the binary notation}$$

$$\text{for the integer } i \ \& \ \text{the } i\text{-th digit of the word } x \text{ equals } 1))\}$$

Theorem 2.1 *The language A cannot be recognized by a multitape 2-way Monte Carlo Turing machine in time $n + o(n)$.*

Proof. Assume from the contrary that A is recognized by multitape 2-way Monte Carlo Turing machine M in time $n + f(n)$ where $f(n) = o(n)$.

We fix an arbitrarily large integer m and consider the work of M on all the words

$$x2y \tag{1}$$

where $y \in \{0,1\}^m$ and $x \in \{0,1\}^{2^m}$. We call $x2$ the head of the word (1), and y the tail of the word. The length n of the word (1) equals $2^m + m + 1$.

We consider the first moment when the machine M reads the symbol 2 from the input. Let the heads on the work tapes observe the squares $b_1, \ldots, b_r$ at this moment. There remains no more time than $m + f(n)$ till the moment of output. The contents of a square of the worktape can influence the result

only if this square is reachable from the squares $b_1, \ldots, b_r$ in no more than $m + f(n)$ steps. The absolute addresses $b_1, \ldots, b_r$ also do not influence the result. Hence all the needed information about the head of the word (1) is encoded in the configuration of the reachable part of the worktapes. We denote the set of all the a priori possible configurations of this part of the worktapes by

$$\{z_1, z_2, \ldots, z_u\} \tag{2}$$

It is easy to see that there is a constant $c > 0$ such that

$$u \leq c^{m+f(n)} \tag{3}$$

Hence all the needed information about the head of the word (1) is encoded by the probability distribution in the set (2). The contradiction obtained below shows the cardinality u of the set (2) is too small for this task.

We will use methods of Shannon's Information Theory for this proof. All the notions and notation not defined in this paper see in [Gal68].

Let the words

$$w_1, \ldots, w_{2^m}$$

denote the lexicographical ordering of all the words in $\{0,1\}^m$.

We consider a random variable $X = (X_1, \ldots, X_{2^m})$ each component of which takes value

$$\begin{cases} 1 & \text{with probability } 1/2 \\ 0 & \text{with probability } 1/2 \end{cases}$$

statistically independently from the other components. (In alternative way to explain the same thing, we consider the head of the word (1) as taken randomly with X_i being the i-th digit of the head of the word.)

Then all the 2^{2^m} possible values of the random varibale X are equiprobable, and each one of these 2^{2^m} values corresponds to a certain head of the word (1). Hence the entropy

$$H(X) = 2^m \tag{4}$$

Now we define a random variable Z taking values $z_1, \ldots, z_u$ out of (2). The probabilities of these values are defined by taking the random varibale X, considering the head of the word (1) corresponding to the particular value of X, processing it by the machine M and observing the configurations of the reachable part of the worktapes.

We estimate the amounbt of information about X in Z:

$$I(X|Z) = I(Z|X) \leq H(Z) \leq \log_2 u$$

Taking (3) into consideration, we get

$$I(X|Z) \leq (m + f(n)) \log_2 c \tag{5}$$

We introduce new random variables $Y_i'(i = 1, 2, \ldots, 2^m)$ taking the values 1, 0 and "no result". The probabilities of these values are defined as the probabilities for the machine M to output 1, 0 or no definite result in $n+f(n)$ steps, correspondingly, provided the tail of the word (1) is w_i. Finally, we introduce random variables $Y_i(i = 1, 2, \ldots, 2^m)$ taking the values 1 and 0 only. The probabilities of these values are defined as follows.

$$p(Y_i = 0) = p(Y_i' = 0) + \frac{1}{2} p(Y_i' = \text{"no result"})$$

$$p(Y_i = 1) = p(Y_i' = 1) + \frac{1}{2} p(Y_i' = \text{"no result"})$$

It is easy to see that

$$p(Y_i' = 0) + p(Y_i' = 1) + p(Y_i' = \text{"no result"}) = 1$$

and

$$p(Y_i = 0) + p(Y_i = 1) = 1$$

These probabilities can be calculated from the probabilities of the various values of z_j of the random variable Z and the conditional probabilities of the results of the machine M starting from a configuration z_j and reading w_i from the tail of the input word. This calculation shows that the random variables Y_i do not depend from X immediately but only through the random variable Z.

We consider a lower bound for $I(X|Z)$. We have

$$I(X|Z) = H(X) - H(X|Z)$$

$$H(X|Z) = \sum_{j=1}^{u} p(z_j) H(X|Z = z_j)$$

$$H(X|Z=z_j) = H(X_1, X_2, \ldots, X_{2^m}|z=z_j) \leq \sum_{i=1}^{2^m} H(X_i|Z=z_j) \qquad (6)$$

Hence

$$H(X|Z) \leq \sum_{j=1}^{u} p(z_j) \sum_{j=1}^{2^m} H(X_i|Z=z_j) =$$

$$= \sum_{i=1}^{2^m} \sum_{j=1}^{u} p(z_j H(X_i|Z=z_j) = \sum_{i=1}^{2^m} H(X_i|Z) \qquad (7)$$

We proceed to estimate $H(X_i|Z)$. We start by trying to prove the intuitively valid inequivalence

$$H(X_i|Z) \leq H(X_i|Y_i) \qquad (8)$$

Indeed,

$$H(Y_i|Z) = H(Y_i|X_i, Z)$$

because the distribution of probabilities for Y_i is determined by the random variable Z and do not depend on the value of X. Hence we get

$$H(Z, Y_i) - H(Z) = H(X_i, Z, Y_i) - H(X_i, Z)$$

$$H(X_i, Z) - H(Z) = H(X_i, Z, Y_i) - H(Z, Y_i)$$

$$H(X_i|Z) = H(X_i|Z, Y_i) \leq H(X_i|Y_i)$$

This completes the proof of (8).

It follows from (7) and (8) that

$$H(X|Z) \leq \sum_{i=1}^{2^m} H(X_i|Y_i) \qquad (9)$$

We fix an arbitrary $i (1 \leq i \leq 2^m)$ and the values for the random variables $X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_{2^m}$. It follows from the definition of X that

$$p(X_i = 0) = p(X_i = 1) = 1/2 \qquad (10)$$

The Monte-Carlo machine M recognizes the language A with probability exceeding $1/2 + \delta$. Hence there are positive real numbers ϵ_0 and ϵ_1 such that

$$p(Y_i = 0|X_i = 0) = 1/2 + \delta + \epsilon_0 \qquad (11)$$

$$p(Y_i = 1|X_i = 1) = 1/2 + \delta + \epsilon_1$$

It follows from (10) and (11)

$$p(X_i = 0|Y_i = 0) =$$

$$\frac{p(X_i = 0)p(Y_i = 0|X_i = 0)}{p(X_i = 0)p(Y_i = 0|X_i = 0) + p(X_i = 1)p(Y_i = 0|X_i = 1)} =$$

$$= \frac{1/2(1/2 + \delta + \epsilon_0)}{1/2(1/2 + \delta + \epsilon_0) + 1/2(1/2 - \delta - \epsilon_1)} \geq \frac{1/2 + \delta + \epsilon_0}{1 + \epsilon_0} \geq 1/2 + \delta$$

Hence

$$H(X_i|Y_i = 0) \leq H(1/2 + \delta, 1/2 - \delta)$$

Similarly one can prove

$$H(X_i|Y_i = 1) \leq H(1/2 + \delta, 1/2 - \delta)$$

It follows that

$$H(X_i|Y_i) = p(Y_i = 0)H(X_i|Y_i = 0) + p(Y_i = 1)H(X_i|Y_i = 1) \leq$$

$$\leq H(1/2 + \delta, 1/2 - \delta)$$

Combining this inequality and (9), we get

$$H(X|Z) \leq 2^m H(1/2 + \delta, 1/2 - \delta)$$

Taking into account (4) and (6), we get

$$I(X|Z) \geq D \cdot 2^m \tag{12}$$

where $D = 1 - H(1/2 + \delta, 1/2 - \delta) > 0$.

Comparing (5) and (12), we get

$$D \cdot 2^m \leq (m + f(n)) \cdot \log_2 c$$

This implies

$$f(n) \geq 2^m \cdot const$$

$$f(n) \geq n \cdot const$$

Contradiction. $\square$

Binary notation of integers is natural, and it has nice properties. However, from our point of view, it has a serious deficiency. Namely, two consecutive integers can have binary notations differing in very many symbols.

We call a notation $sbin(n)$ of non-negative integers n *superbinary* if it has two properties:

1. The length $|sbin(n)|$ for arbitrary non-negative integer n is no more and no less than $const \cdot \log n$;

2. There is a deterministic real time "clock", i.e. a multitape deterministic Turing machine maintaining $sbin(t)$ on one of its worktapes at every moment t (it is allowed to have several symbols of $sbin(t)$ in the same square of the worktape).

For instance, the following notation of non-negative integers in the alphabet $\Sigma_1 = \{0, 1, \boxed{1}, *, \underline{*}, \overrightarrow{0}, \overleftarrow{0}, \overrightarrow{1}, \overleftarrow{1}\}$ is superbinary:

n	s	bi	$n(n)$
0			$\underline{*}$
1		$\boxed{1}$	$*$
2		$\overrightarrow{1}$	$*$
3		1	$\underline{*}$
4		$\overleftarrow{0}$	$*$
5	$\boxed{1}$	0	$*$
6	$\overrightarrow{1}$	0	$*$
7	1	$\overrightarrow{0}$	$*$
8	1	0	$\underline{*}$
9	1	$\boxed{1}$	$*$
10	1	$\overrightarrow{1}$	$*$
11	1	1	$\underline{*}$
12	1	$\overleftarrow{0}$	$*$
13	$\overleftarrow{0}$	0	$*$
14	$\boxed{1}$	0	0	$*$
15	$\overrightarrow{1}$	0	0	$*$
16	1	$\overrightarrow{0}$	0	$*$
17	1	0	$\overrightarrow{0}$	$*$
18	1	0	0	$\underline{*}$
19	1	0	$\boxed{1}$	$*$
20	1	0	$\overrightarrow{1}$	$*$
21	1	0	1	$\underline{*}$
22	1	0	$\overleftarrow{0}$	$*$
23	1	$\boxed{1}$	0	$*$
24	1	$\overrightarrow{1}$	0	$*$

...

We consider language

$$A' = \{x2y|(\exists i)(\exists j)(x \in \{0,1\}^j \& y \in \Sigma_1^* \& y = sbin(i) \& j \geq i \&$$

$$\&\text{the } i\text{-th digit of the word } x \text{ equals } 1\}$$

Theorem 2.2 *The language A' cannot be recognized by a multitape 2-way Monte Carlo Turing machine in time $n + o(n)$.*

Proof. Essentially the same as for Theorem 2.1. □

We go on to consider more complicated languages. Let M be a deterministic multitape 2-way Turing machine, Σ be the input alphabet of M, $t_M(x)$ be the running time of M on input x, and $g(n)$ be the maximum of $(t_M(x) - |x|)$ over all the words $x \in \Sigma^*$ of length not exceeding n. Let $\#$ be a symbol not in $\Sigma \cup \{0, 1, 2\}$.

$$B_M = \{x\#y | x \in A' \text{ and } |y| = t_M(|x\#y|)\}$$

Theorem 2.3 *For arbitrary deterministic multitape 2-way Turing machine, the language B_M cannot be recognized by a multitape 2-way Monte Carlo Turing machine in time $n + o(g(n))$.*

Proof. Essentially the same as for Theorem 2.1. □

Theorem 2.4 *For arbitrary deterministic multitape 2-way Turing machine, the language B_M can be recognized by a multitape 2-way Monte Carlo Turing machine in time $n + 2g(n)$.*

Proof. The TM recognizing B_g performs several actions in parallel:

1. Simulates M on $x\#y$ and stores the value of $t_M(|x\#y|)$ on an additional worktape (in unary notation). (This is done in time $n + g(n)$.)

2. Finds out whether $|y| = t_M(|x\#y|)$. (This is done after the action 1 in no more time than $g(n)$.)

3. Finds out whether $y \in A$. (This is done after the action 1 in no more time than $g(n)$; can be done in parallel with the action 2.)

□

Using Theorem 3.1 (below in Section 3) we can improve this result to get

Theorem 2.5 *For arbitrary deterministic multitape 2-way Turing machine, and for arbitrary $\epsilon > 0$ the language B_M can be recognized by a multitape 2-way deterministic Turing machine in time $n + \epsilon g(n)$.*

This way, $n + const \cdot g(n)$ is optimal time for recognition of the language B_M, and randomization does not help.

3 Reduced Time Complexity

Theorems 2.3 and 2.4 show that the complexity measure $t(n) = n + g(n)$ is a natural complexity measure, any way, no less natural than sublinear space complexity introduced in [SHL65] by separating input tape from the worktapes. Before [SHL65] it was considered that the lowest nontrivial space complexity class was LINSPACE since reading all the input demands at least linear space. Separating input tape allowed to consider LOGSPACE and other important small space complexity classes. For instance, it was proved in [SHL65] that DLOGLOGSPACE contains nonregular languages but every language recognizable in $o(\log\log n)$ space even by nondeterministic Turing machines is regular. However there are regular languages recognizable by Monte Carlo 2-way Turing machines in constant space[Fr81], and there are nontrivial space classes defined by $\log\log\log n$, $\log\cdots\log n$, $\log^* n$ for Monte Carlo 2-way Turing machines[KV87].

In this section we prove some results motivating the naturalness of our approach to sublinear time complexity.

Theorem 3.1 *If $n + g(n)$ is the running time of a deterministic (nondeterministic, Monte Carlo, alternating) multitape 2-way Turing machine recognizing a language L, then the language L can be recognized by a deterministic (nondeterministic, Monte Carlo, alternating) multitape 2-way Turing machine in time $n + \frac{1}{2}g(n)$.*

Proof. Every worktape of the given machine is simulated by a worktape with a larger alphabet. Content of four squares from the old tape is stored in a single square of the new tape. An additional worktape is used to make a copy of the input tape (again with 4 symbols stored in a single square of the worktape) thus allowing more sppedy retrieval. If the time interval between two consequtive moments of reading a new input symbol by the given machine is between 2 and 4, the simulating machine reads the corresponding input symbols with time interval 1. If the time interval exceeds 4, the simulating machine speeds up. □

Corollary 3.1 *If a language L is recognized by a deterministic (nondeterministic, Monte Carlo, alternating) multitape 2-way Turing machine in time $n + const$, then the language L can be recognized by a deterministic (nondeterministic, Monte Carlo, alternating) multitape 2-way Turing machine in real time.*

It is well-known that even deterministic multitape 2-way Turing machines recognize some non-regular languages in real time, e.g. the language

$$\{x2x | x \in \{0,1\}^*\}$$

For the sake of brevity we introduce a new term for the complexity measure "running time minus the length of the input word". We will call it *reduced running time*.

So far we have shown that the reduced running time can be effectively decreased arbitrary constant number of times (Theorem 3.1). Nonregular languages can be recognized in reduced time 0. Now we go on to show that the reduced time complexity can be very slowly growing: $\log n$, $\log\log n$, $\log\log\log n$, ..., $\log^* n$, etc.

Theorem 3.2 *For arbitrary integer $k \geq 1$, there is a language L_k such that:*

1. *For arbitrary $\epsilon > 0$, the language L_k can be recognized in time $n + \epsilon \cdot \underbrace{\log\log\ldots\log}_{k \text{ times}} n$ by a deterministic 2-way Turing machine;*
2. *No Monte Carlo 2-way Turing machine can recognize L_k in time $n + o(\underbrace{\log\log\ldots\log}_{k \text{ times}} n)$.*

Proof. It suffices to prove the existence of a deterministic Turing machine with the reduced running time $\underbrace{\log\log\cdot\log}_{k \text{ times}} n$, and our theorem will be implied by Theorems 2.3, 2.4 and 2.5.

For $k = 1$ we consider a deterministic 2-way Turing machine which recognizes whether the input word is of type

$$sbin(0) * sbin(1) * sbin(2) * sbin(3) * \ldots * sbin(i)$$

After the real time recognition whether the input word is of the needed type the machine additionally reads the last fragment $sbin(i)$ once more. The running time exceeds the length of the input word by $|sbin(i)| = const \cdot \log i = const' \cdot \log n$.

For $k > 1$ the same idea is used iteratively. Let

$$m_1 = |sbin(m_2)|$$

$$m_2 = |sbin(m_3)|$$
$$m_3 = |sbin(m_4)|$$
$$\ldots$$
$$m_k = |sbin(m_{k+1})|$$

The input word is supposed to be of type

$$sbin(0) * sbin(1) * \ldots * sbin(m_{k+1}) * *sbin(0) * sbin(1) * \ldots * sbin(m_k * * \ldots$$

$$\ldots * *sbin(0) * sbin(1) * \ldots * sbin(m_1)$$

□

Theorem 3.3 *There is a language L_* such that:*

1. *For arbitrary $\epsilon > 0$, the language L_k can be recognized in time $n + \epsilon \cdot \log^* n$ by a deterministic 2-way Turing machine;*
2. *No Monte Carlo 2-way Turing machine can recognize L_k in time $n + o(\log^* n)$.*

Like other complexity measures, the reduced time complexity turns out to be sensitive in choosing the level of determinism (deterministic, nondeterministic, probabilistic, Monte Carlo, alternating) of the machine.

Theorem 3.4 *There is a language A' such that:*

1. *A' can be recognized in real time by a nondeterministic 1-way Turing machine,*
2. *$\bar{A}'$ can be recognized in real time by a nondeterministic 1-way Turing machine,*
3. *No deterministic or Monte Carlo 2-way Turing machine can recognize A' in $n + o(n)$ time.*

Proof. The language A' from Theorem 2.2 has the needed properties. □

4 Advantages of Randomization

In spite of the huge literature on complexity advantages of randomized machines over deterministic ones there had not been proved any running time advantages of Monte Carlo multitape 2-way Turing machines over deterministic machines of the same type.

Theorem 4.1 *There is a language E such that:*

1. *For arbitrary $\epsilon > 0$, the language E can be recognized in real time by a Monte Carlo multitape 2-way machine with probability $1 - \epsilon$,*
2. *E cannot be recognized in time $n + o(n)$ by a deterministic multitape 2-way Turing machine.*

Sketch of proof. The language E consists of all the words in the form

$$x_1 2 x_2 2 y_1 2 y_2 3 u(1) 2 u(2) 2 \ldots 2 u(i) 3 v(1) 2 v(2) 2 \ldots 2 v(i)$$

such that

$$(\exists m)(\exists j)(x_1 = 1^m \& x_2 = 1^{2^m} \& y_2 = 1^j \& 2^m \leq i \leq j \leq 8i \&$$

$$\&(\forall k)(1 \leq k \leq i \rightarrow u(k) \in \{0,1\}^m \cdot 2 \cdot 1^{m^3}) \& \text{the string}$$

$$v(1), v(2), \ldots, v(i) \text{ is permutation of } u(1), u(2), \ldots, u(i)$$

1. The prefix $x_1 2 x_2 2 y_1 2 y_2$ is designed to allow the randomized machine time to prepare the random parameters and organize the worktapes.

   Let c denote $\lceil \frac{2}{\epsilon} \rceil$. A random integer r is chosen ($1 \leq r \leq c \cdot 2^m$). A random prime number p exceeding $c \cdot 2^m$ is chosen among the first $c \cdot j$ prime numbers (i.e. among the random integers of size not exceeding $\log_2(c \cdot j \cdot \ln j)$ being prime numbers). The random number theorem implies that on average the length of the prefix suffices for the randomized machine to generate and test such a random prime.

   When the blocks $u(1), u(2), \ldots, u(i)$ are read from the input, and $u(k) = bin(z) \cdot 2 \cdot 1^{m^3}$, the number $r^z \pmod p$ is added to a counter. When the blocks $v(1), v(2), \ldots, v(i)$ are read from the input, and $v(k) =$

$bin(z) \cdot 2 \cdot 1^{m^3}$, the number $r^z \pmod p$ is subtracted from the counter. If the input word is in the language E, at the end the counter is empty.

If the input word is not in E, the properties of Vandermonde determinant show that no more than the fraction $\frac{1}{c}$ of all the random numbers r would make the totals of r^z (not $r^z \pmod p$) for $u(1), u(2), \ldots, u(i)$ and $v(1), v(2), \ldots, v(i)$ equal.

If

$$\sum_u r^z \neq \sum_v r^z$$

but the totals are congruent modulo p, then the difference is a multiple of p. If inequal totals are congruent modules $p_1, p_2, \ldots, p_s$ then the difference is a multiple of the product $p_1 \cdot p_2 \cdot \ldots \cdot p_s$. Since the difference of totals cannot be larger than the maximum value of the total, the fraction of "defective" prime modulos is small.

Computing of $r^z \pmod p$ is not real-time but we have added blocks 1^{m^3} when constructing the blocks $u(1), \ldots, u(i), v(1), \ldots, v(i)$. This way, the values

$$\begin{aligned} & r^0 \pmod p \\ & r^1 \pmod p \\ & r^2 \pmod p \\ & r^4 \pmod p \\ & \ldots \\ & r^{2^m} \pmod p \end{aligned}$$

are preprocessed, and added to the counter when the block $u(k)$ (containing binary notation of z) is read from the input.

2. Notion of Kolmogorov complexity is used for this part of the proof.

   Assume from the contrary that there is such a deterministic machine. Take a large integer m and a binary string α such that $|\alpha| = 2^m$ and α has nearly maximal Kolmogorov complexity.

   We denote

$$\alpha = \alpha_1 \alpha_2 \ldots \alpha_{2^m}$$

We construct $u(1)2u(2)2\ldots2u(i)$ taking the strings z in the lexicographical order (00000, 00001, 00010, 00011, ...). If the current α_b equals 0, we take one block $u(k)$ corresponding to this z. If the current α_b equals 1, we take two blocks $u(k)$ and $u(k+1)$ corresponding to this z. This way, $2^m \leq 2 \cdot 2^m$.

We consider a special graph representing the information transfer in the multitape Turing machine. The vertices of the graph correspond to the time moments and positions of the heads on the tapes. If there are d tapes (including the input tape) then d new vertices correspond to every time moment (one vertice per head). The vertices are connected if:

(a) either the time moments are the same or adjacent,

(b) or the same square of the tape is visited again at some different moment (but there have been no visits to this square of the tape between these moments).

Since the input word corresponds to α which has a high Kolmogorov complexity, the number of sqaures on the worktapes is to be at least linear with respect to $|\alpha|$. This implies the nearly-linear diameter of the graph. Hence there are fragments of $u(1)2u(2)2\ldots2u(i)$ and $v(1)2v(2)2\ldots2v(i)$ containing the same linear number of z's such that the distance in the graph is of linear size. Next, we consider cuts in this graph separating these fragments. Since the cuts are disjoint sets of vertices, and the diameter is nearly-linear, there is a cut of about-logarithmic size. However the cut is supposed to contain all the information about α, otherwise the machine can be fooled. But one cannot compress α into about-logarithmic size. Contradiction.

□

5 Acknowledgement

The idea to use prime modulos for the randomized algorithm in the proof of Theorem 4.1 was proposed to the authors by Andris Ambainis. We thank also Leonid Levin and Peter Gacs for interesting discussions.

References

[Fr75] Freivalds, R., *Fast computations by probabilistic Turing machines*, Proceedings of Latvian State University, 233(1975), pp. 201-205 (Russian)

[Fr77] Freivalds, R., *Probabilistic machines can use lsess runnning time*, Information Processing'77 (Proc. IFIP Congress'77), North Holland, 1977, pp. 839-842

[Fr79] Freivalds, R., *Speeding up recognition of some sets by usage of random number generators*, Problemi kibernetiki, 36(1979), pp. 209-224 (Russian)

[Fr81] Freivalds, R., *Probabilistic two-way machines*, LNCS, 118(1981), pp.33-45

[Fr83] Freivalds, R., *Space and reversal complexity of probabilistic one-way Turing machines*, LNCS, 158(1983), pp. 159-170

[Fr85] Freivalds, R., *Space and reversal complexity of probabilistic one-way Turing machines*, Annals of Discrete Mathematics, 24(1985), pp. 39-50

[FI77] Freivalds, R., Ikaunieks, E., *On advantages of nondeterministic machines over probabilistic ones*, Izvestiya VUZ. Matematika, No.2(177), 1977, pp.108-123 (Russian)

[FK94] ...

[Gal68] Gallager, R.G., *Information Theory and Reliable Communication.* John Wiley, NY, 1968

[JKS84] Ja'Ja', J., Prasanna Kumar, V.K., Simon, J., *Information transfer under different sets of protocols*, SIAM J. Computation, 13(1984), pp.840-849

[KF90] Kaneps, J. and Freivalds R., *Minimal nontrivial space complexity of probabilistic one-way Turing machines*, LNCS, 452(1990), pp. 355-361

[KV87] Karpinski, M. and Verbeek, R., *On the Monte Carlo space constructible functions and space separation results for probabilistic complexity classes*, Information and Computation, 75(1987), pp. 178-189

[KV88] Karpinski, M. and Verbeek, R., *Randomness, probability, and the separation of Monte Carlo time and space*, LNCS, 270(1988), pp. 189-207

[KV93] Karpinski, M. and Verbeek, R., *On randomized versus deterministic computation*, Proc. ICALP'93, LNCS, 700(1993), pp. 227-240

[LV93] Ming Li, Paul Vitanyi, *An Introduction to Kolmogorov Complexity and Its Applications*, Springer, 1993

[SHL65] Stearns, R.E., Hartmanis, J., Lewis, P.M., *Hiearchies of memory limited computations*, Proc. IEEE Conference on Switch., Circuit Theory and Logical Design, 1965, pp. 179-190