Space Bounds for Interactive Proof Systems with Public Coins and Bounded Number of Rounds

Maciej Liśkiewicz*
Rüdiger Reischuk†

May 1996

Abstract

This paper studies interactive proofs, resp. Arthur-Merlin games with a sublogarithmic space-bounded verifier. We provide examples of specific languages and show that such systems are unable to accept these languages. As a consequence, new separation is obtained for the complexity classes on the second level of the round/alternation hierarchy.

1 Introduction

A stochastic Turing machine model (STM) as introduced by Papadimitriou [Pa85] is a nondeterministic machine extended with the ability to perform random moves. Alternative characterizations can be given by interactive proof systems of Goldwasser, Micali and Rackoff [GMR89] and Arthur-Merlin-games of Babai see also [GoSi86] and [Co89]. In a probabilistic state a stochastic machine M chooses among the successor configurations with equal probability.

A computation of M on an input X can be described by a computation tree. To define acceptance of X, for each existential configuration one chooses a successor that maximizes the probability of reaching an accepting leaf. The acceptance probability of X is then given by the acceptance probability of the starting configuration in this truncated tree. M accepts a language L in space S if

- for all $X \in L$, the probability that M accepts X is more than $3/4$,
- every $X \notin L$ is accepted with probability less than $1/4$, and
- M never uses more than $S(|X|)$ space.

Definition 1 *Let $MA_k Space(S)$ (resp. $AM_k Space(S)$) denote the set of languages that can be accepted by such a machine in space S making at most $k-1$ alternations between existential and probabilistic configurations and starting in existential (resp. probabilistic) mode. For such a machine we also say that it works in space S and k rounds.*

This paper studies space complexity classes defined by stochastic machines that use less than logarithmic space. We provide examples of specific languages and show that such machines are unable to accept these languages. As a consequence, new separation results for considered classes are obtained.

*International Computer Science Institute, Berkeley, email: liskiewi@icsi.berkeley.edu. The research of this author was supported by DFG Research Grant Re 672/2 and KBN Grant 2 P301 034 07.
†Med. Universität zu Lübeck, email: reischuk@informatik.mu-luebeck.de

Dwork and Stockmeyer were the first who investigated interactive proof systems with small space-bounded verifiers [DwSt92]. The separation results presented in [DwSt92] are stated for constant space, but they can be extended to any sublogarithmic bound (see e.g. [Co93]).

It has been shown that for sublogarithmic space bounds it makes a difference whether the random moves are known, as it is the case for a STM, resp. for Arthur-Merlin games, or whether they are hidden as in interactive proof systems. The language PALINDROME of all strings that are palindromes, is an example that can easily be recognized by an interactive proof system with a constant space-bounded verifier and hidden random moves, but requires logarithmic space otherwise (see [DwSt92] and [Co93]).

This property does not hold for the corresponding polynomial time classes $AM_k Time(\mathrm{POL})$. Here, POL denotes the set of all polynomials. Furthermore, it has been shown [Ba85] that any number of rounds can be reduced to two rounds, that is

$$MA_2 Time(\mathrm{POL}) \subseteq AM_2 Time(\mathrm{POL}) = AM_{\mathrm{CON}} Time(\mathrm{POL}) .$$

One might expect that for at least logarithmic space bounds similar to alternating Turing machines (ATMs) the alternating stochastic hierarchy collapses to some level. On the other hand we have obtained some evidence that for space bounds $S \in \mathtt{SUBLOG} := \Omega(\mathrm{llog}) \cap o(\log)$ the situation is different.

Extending an impossibility result for 2-way probabilistic finite automata one can show that the language

$$\mathtt{CENTER} := \{w0x \mid w, x \in \{0,1\}^* \quad \text{and} \quad |w| = |x|\}$$

cannot be recognized by a sublogarithmic space-bounded probabilistic Turing machine with any error probability $\epsilon < \frac{1}{2}$ [FrKa94]. However, there exists a constant space interactive proof system for CENTER . Hence, for any $S \in o(\log)$, languages CENTER and PALINDROME yield the following separations:

$$BPSpace(S) = AM_1 Space(S) \subset AMSpace(S) \subset AMSpace(\log) = \mathcal{P} .$$

The last equivalence is due to Condon [Co89].

A sublogarithmic STM may require exponential expected time. It is shown in [DwSt92] that some languages cannot be recognized by such machines faster – CENTER is one of such example. On the other hand, the power of sublogarithmic space bounded STMs restricted to polynomial expected time is still an open problem. In this case we do not even know if stochastic finite automata can recognize nonregular languages (see [DwSt92] and [CHPW94]).

Let $\mathtt{BIN}(m) := \mathrm{bin}(0)\#\,\mathrm{bin}(1)\#\,\mathrm{bin}(2)\#\ldots\#\,\mathrm{bin}(m)$, and define

$$\begin{aligned} \mathtt{PATTERN} := \{ & w_1 * w_2 * \ldots * w_t * u * \mathtt{BIN}(2^k) \mid w_1, \ldots, w_t \in \{0,1\}^k \\ & \text{for some } t, k \in \mathbb{N} \text{ and } u \in \{w_1, \ldots, w_t\} \} . \end{aligned}$$

One of the main results of this paper says the following:

Lemma 1 *For $S \in o(\log)$, an S-space-bounded STM cannot recognize* PATTERN *in 2 rounds starting in probabilistic mode, that is*

$$\mathtt{PATTERN} \notin AM_2 Space(o(\log)) .$$

On the other hand, it is not hard to see that switching the quantifiers the problem can be solved within small space.

Lemma 2 *For arbitrary small* $\epsilon > 0$ *, there exists an* llog *–space-bounded STM* M *for* `PATTERN` *with error probability* ϵ *that works in 2 rounds starting in nondeterministic mode, that is*

$$\texttt{PATTERN} \in MA_2Space(\text{llog}) .$$

Moreover, M *works in polynomial time.*

Therefore, we obtain the following separation

Theorem 1 $MA_2Space(\text{llog}) \not\subseteq AM_2Space(o(\log))$.

We conjecture that the alternation hierarchy for sublogarithmic space-bounded STMs is infinite, similar as for standard ATMs, that means

$$AM_1Space(S) \subset AM_2Space(S) \subset AM_3Space(S) \subset \ldots$$

It is interesting to notice that the $AM_kSpace(S)$ and $MA_kSpace(S)$ classes do not seem to be complementary, since

Lemma 3

$$\overline{\texttt{PATTERN}} \notin AM_2Space(o(\log)) .$$

The remainder of this paper is organized as follows. In Section 2 some definitions and notions are introduced. Section 3 contains the proofs of our lower bounds results (Lemma 1 and 3). In Section 4 a stochastic Turing machine for `PATTERN` is described what proves Lemma 2.

2 Preliminaries

The Turing machine model we consider is equipped with a two-way read-only input tape and a single read-write work tape. The input word is stored on the input tape between end-markers \$.

A *memory state* of an STM M is an ordered triple $\alpha = (q, u, i)$, where q is a state of M , u a string over the work tape alphabet, and i a position in u (the location of the work tape head). We say that a memory state $\alpha = (q, u, i)$ is nondeterministic, probabilistic, accepting or rejecting, according to q . A *configuration* of M on an input X is a pair (α, j) consisting of a memory state α and a position j with $0 \le j \le |X| + 1$ of the input head. $j = 0$ or $j = |X| + 1$ means that this head scans the left, resp. the right end-marker.

Let

$$(\alpha, i) \models_X (\beta, j)$$

denote the property that M with X on its input tape has a computation path $C_1 = (\alpha, i), C_2, \ldots, C_t = (\beta, j)$ such that $t \ge 2$, the mode of the configurations $C_1, C_2, \ldots, C_{t-1}$ is probabilistic, and C_t is nondeterministic. Let $H = \{h_1, h_2, \ldots, h_t\}$ be a subset of input head positions on X with $0 \le h_1 < h_2 < \ldots < h_t \le |X| + 1$.

$$(\alpha, h_i) \models_{X,H} (\beta, j)$$

denotes the same fact as above, but with the following restriction: M starts with the input head on the h_i -th symbol but then never reaches the borders h_{i-1} and h_i if $j < h_i$ or h_i and h_{i+1} if $j > h_i$.

The *word probabilities* of M on a word Z over the input alphabet of M is defined as follows. A starting condition for the word probability is a pair $\langle \alpha, h \rangle$ where α is a probabilistic memory state of M and $h \in \{Left, Right\}$ what means that M starts on the h end of Z in memory state α . A stopping condition for the word probability is either:

1. a pair $\langle \alpha, h \rangle$ as above meaning that the input head falls off the h end of Z with M in memory state α ,
2. $\langle$R-Loop$\rangle$ meaning that the probabilistic computation of M loops forever within Z , or
3. a triple $\langle \alpha, h, \text{Alter} \rangle$ meaning that within Z M alternates from probabilistic to nondeterministic configuration visiting for the last time an end of Z in configuration (α, h) .

For each starting condition σ and each stopping condition τ , let $p(Z, \sigma, \tau)$ be the probability that stopping conditions occurs given that M started in starting condition σ on Z . Note that for any starting condition $\langle \alpha, h \rangle$ and any stopping condition of type (3) it holds that $p(Z, \langle \alpha, h \rangle, \langle \alpha', h', \text{Alter} \rangle) = 0$ if $\langle \alpha, h \rangle \neq \langle \alpha', h' \rangle$.

We model computations of a probabilistic phase of M by Markov chains with finite state space, say $1, 2, \ldots, s$ for some s . A particular Markov chain is completely defined by its matrix $R = \{r_{ij}\}_{1 \leq i,j \leq s}$ 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 the designated starting state, say, state 1, and some set T_R of trapping states, so $r_{kk} = 1$ for all $k \in T_R$. For $k \in T_R$, let $a[k, R]$ denote the probability that Markov chain R is trapped in state k when started in state 1.

Let $\beta \leq 1$. 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 $R = \{r_{ij}\}_{1 \leq i,j \leq s}$ and $R' = \{r'_{ij}\}_{1 \leq i,j \leq s}$ are β -close if r_{ij} and r'_{ij} are β -close for all pairs i, j .

Lemma 4 (Dwork, Stockmeyer) 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 $z = 2s$.

We characterize a word Z according to a nondeterministic phase of computation of M on Z by a *word transitions*. As previously, a starting condition for the word transition is a pair $\langle \alpha, h \rangle$ where α is a nondeterministic memory state of M and $h \in \{Left, Right\}$. A stopping condition for the word transition is either:

1. a pair $\langle \alpha, h \rangle$ as above,
2. $\langle$N-Loop$\rangle$ meaning that the nondeterministic computation of M loops forever within Z ,
3. $\langle$Accept$\rangle$ meaning that M halts in the accepting state before the input head falls off either end of Z , or
4. $\langle$Reject$\rangle$ meaning that M halts in the rejecting state before the input head falls off either end of Z .

For each starting condition σ and each stopping condition τ , the word transition $t(Z, \sigma, \tau)$ equals to one if M starting in σ can reach τ on Z and otherwise it is zero.

3 Lower bounds

In this section proofs of Lemma 1 and 3 will be given. To show this results, we extend the methods from [DwSt92] and some of [LiRe93].

Through the rest of this section we will assume that M is an arbitrary 2-round STM, of space complexity $S \in o(\log)$ that starts in a random state. Let by $Vol(N)$ we denote the number of possible memory states of the machine M on input words of length N. Let n be a sufficiently large integer of the form 2^k. Below we will show that M is fooled when it works on the inputs of the form

$$w_1 * w_2 * \ldots * w_n * w_{n+1} * \ldots * w_{2n} * u * \mathtt{BIN}(n) ,$$

with $|w_i| = |u| = k$. Let N denote the length of such input, i.e. let

$$N := 2n \cdot (k+1) + k + 1 + |\mathtt{BIN}(n)| .$$

Note that since $S(N) \in o(\log N)$ hence $S(N) \in o(\log n)$ and that $Vol(N) \leq 2^{O(S(N))}$.

We will consider the word probabilities and the word transitions as defined in the previous section. We restrict ourselves to words Z from a set

$$W_n \subseteq \{w_1 * w_2 * \ldots * w_n * \mid w_1, \ldots, w_n \in \{0,1\}^k\}$$

such that

1. for any word $w_1 * w_2 * \ldots * w_n *$ and $x_1 * x_2 * \ldots * x_n *$ in W_n it holds:

$$\{w_1, w_2, \ldots, w_n\} \setminus \{x_1, x_2, \ldots, x_n\} \neq \emptyset$$

2. $|W_n| = 2^n$.

Note that sets as assumed above exist. Let W_n be fixed. Denote by $\bar{N} = n \cdot (k+1)$ the length of words in W_n. Hence for arbitrary $Z \in W_n$ we consider

$$d_p = 8 \cdot (Vol(N))^2 + 2 \cdot Vol(N)$$

word probabilities and

$$d_t = 4 \cdot (Vol(N))^2 + 6 \cdot Vol(N)$$

word transitions.

Fix some order of the pairs (σ, τ) of starting and stopping conditions for word probabilities as well as some order of the pairs (σ, τ) for word transitions. Let $\mathrm{p}(Z)$ be the vector of the d_p probabilities and let $\mathrm{t}(Z)$ be the vector of the d_t transitions according to these orderings.

Fact 1 If p is nonzero element of $\mathrm{p}(Z)$, then $p \geq 2^{-Vol(N) \cdot \bar{N}}$.

This claim follows easily from the fact that the shorter computation path of M on Z that starts and ends in specific configurations is not longer than $Vol(N) \cdot \bar{N}$.

Divide first W_n into equivalence classes defining that two words Z and Z' are equivalent if and only if $\mathrm{p}(Z)$ and $\mathrm{p}(Z')$ are zero in exactly the same coordinates. Let E_n be the largest equivalence class. Hence $|E_n| \geq |W_n|/2^{d_p}$. Let $Z \in E_n$. Note that by Fact 1 the interval in which each nonzero coordinate of $\mathrm{p}(Z)$ lies is $[2^{-Vol(N) \cdot \bar{N}}\,, 1]$.

Let $\log \mathrm{p}(Z)$ be the vector with the i-th coordinate equals to $\log \mathrm{p}(Z)(i)$ if $\mathrm{p}(Z)(i) > 0$ and zero otherwise. Hence each coordinate of $\log \mathrm{p}(Z)$ lies in the interval $[-Vol(N) \cdot \bar{N}\,, 0]$. We divide each interval $[-Vol(N) \cdot \bar{N}\,, 0]$ into subintervals of length μ. In this way we divide the space $[-Vol(N) \cdot \bar{N}\,, 0]^{d_p}$ into $(\frac{Vol(N) \cdot \bar{N}}{\mu})^{d_p}$ cells, each of the size $\mu \times \mu \times \ldots \times \mu$.

We want to choose μ very small but large enough that the number of cells is smaller than $|E_n|/2^{d_t}$ that is

$$\left(\frac{Vol(N) \cdot \bar{N}}{\mu}\right)^{d_p} < \frac{|W_n|}{2^{d_p} 2^{d_t}} . \tag{i}$$

Let us assign

$$\mu := 2^{-\sqrt{n}} .$$

For this value inequality (i) holds, for sufficiently large n, since

$$2^{12\cdot(Vol(N))^2+8\cdot Vol(N)} \cdot \left(\frac{Vol(N)\cdot\bar{N}}{\mu}\right)^{8\cdot(Vol(N))^2+2\cdot Vol(N)} < 2^n .$$

Assuming (i) there must be more than 2^{d_t} words Z in E_n such that $\log \mathrm{p}(Z)$ belong to the the same cell. Let us denote such a subset of E_n by F_n. Therefore for any pair of different words Z and Z' from F_n if p and p' are two nonzero probabilities in the same coordinate of $\mathrm{p}(Z)$ and $\mathrm{p}(Z')$, respectively, then

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

and it follows that p and p' are 2^μ-close. Therefore $\mathrm{p}(Z)$ and $\mathrm{p}(Z')$ are componentwise 2^μ-close. Moreover, since there are at most 2^{d_t} different transition vectors, hence there must be two different words $Z, Z' \in F_n$ with the same transition vector.

So, to sum up there are two different words X and Y in W_n such that $\mathrm{p}(X)$ and $\mathrm{p}(Y)$ are componentwise 2^μ-close and $\mathrm{t}(X) = \mathrm{t}(Y)$.

Let us fix such two different words X and Y. Let $X = x_1 * x_2 * \ldots * x_n *$ and let $Y = y_1 * y_2 * \ldots * y_n *$. ¿From the definition of W_n there exists a word $x \in \{x_1, x_2, \ldots, x_n\} \setminus \{y_1, y_2, \ldots, y_n\}$. We describe Markov chains R_{XY}, R_{YX}, R_{XX}, and R_{YY} which model the probabilistic computation of the machine M on inputs

$$\begin{aligned}
Z_{XY} &= X\ Y\ x\ \mathtt{BIN}(n)\ , \\
Z_{YX} &= Y\ X\ x\ \mathtt{BIN}(n)\ , \\
Z_{XX} &= X\ X\ x\ \mathtt{BIN}(n)\ , \\
Z_{YY} &= Y\ Y\ x\ \mathtt{BIN}(n)\ ,
\end{aligned}$$

respectively. Let us denote the set of these four inputs by $\tilde{Z}$. Each chain has

$$s = 14 \cdot Vol(N) + 4$$

states. The first $7 \cdot Vol(N)$ states have the form $\langle \alpha, h \rangle$, where α is a memory state, and

$$h \in H := \{0,\ 1,\ \bar{N},\ \bar{N}+1,\ 2\bar{N},\ 2\bar{N}+1,\ N+1\} .$$

An intuitive meaning of a state $\langle \alpha, h \rangle$ of the chain R_{XY} is: start M in configuration (α, h). Since $|X| = |Y| = \bar{N}$ and $|Z_{XY}| = N$ position $h = 1$ or $h = \bar{N}$ means that M scans the left end of X, resp., the right one; $h = \bar{N}+1$ or $h = 2\bar{N}$ means that M reads the left or the right end of the string Y and the position $h = 2\bar{N}+1$ means the left end of $x\ \mathtt{BIN}(n)\ \$$. $h = 0$ or $h = N+1$ means that M's head scans the left, resp. the right end-marker. The meaning of state $\langle \alpha, h \rangle$ for the remaining chains is analogous. The next $7 \cdot Vol(N)$ states of each chain have the form $\langle \alpha, h, \mathrm{Alter} \rangle$, with α and h as above. Its meaning is: M alternates from probabilistic to nondeterministic state scanning for the last time before this alternation a position in H in configuration (α, h). The last four states are the following: $\langle \text{R-Loop}_1 \rangle, \langle \text{R-Loop}_2 \rangle, \langle \text{R-Loop}_3 \rangle, \langle \text{R-Loop}_4 \rangle$. For chain R_{XY} they mean that the probabilistic computation of M loops forever on the left end-marker, within X, Y, $x\ \mathtt{BIN}(n)\ \$$, respectively. For remaining chains the meaning is analogous. Let $(\alpha_0, 1)$ be an initial configuration of machine M. The initial state of each chain is $\langle \alpha_0, 1 \rangle$.

The transition probabilities r_{ij} of R_{XY} are obtained from word probabilities of M on the substrings: $\$$, X, Y, $x\ \mathtt{BIN}(n)\ \$$, and from the following definition of the set of trapping states:

$$\{\langle \alpha, h, \mathrm{Alter} \rangle \mid \alpha \text{ is a memory state, } h \in H\} \cup \{\langle \text{R-Loop}_i \rangle | i = 1,2,3,4\} .$$

More precisely the transitions r_{ij} such that states i, j are applied both to the same substring: \$, X, Y, or x $\mathtt{BIN}(n)$ \$ are equal to an appropriate word probabilities of M on this substring. E.g. if $i = \langle \alpha, 2\bar{N} \rangle$ and $j = \langle \beta, \bar{N}+1 \rangle$ then $r_{ij} = p(Y, \langle \alpha, \mathit{Right} \rangle, \langle \beta, \mathit{Left} \rangle)$ since the position $2\bar{N}$ and $\bar{N}+1$ means the right resp., the left end of the substring Y in Z_{XY}. The transitions $r_{kk} = 1$ for any trapping statet k. Remaining values r_{ij}, i.e. transitions for states i, j connecting with two different substrings are defined as follows. If $i = \langle \alpha_1, h_1 \rangle$ and $j = \langle \alpha_2, h_2 \rangle$ then the transition equals to the probability that M reaches in one step the configuration (α_2, h_2) starting in (α_1, h_1). Otherwise $r_{ij} = 0$.

The transition probabilities of R_{YX}, R_{XX}, and R_{YY} are obtained analogously.

W.l.o.g. we assume that the left symbol of X is the same as the left one of Y. Note that by definition, the right symbols of X and Y are equal. Hence a transition for any pair of states i, j connecting with two different substrings, is the same in all considered chains. ¿From this and from the fact that X and Y are 2^{μ}-close we have that for any pair of chains $R, R' \in \{R_{XY}, R_{YX}, R_{XX}, R_{YY}\}$, they are 2^{μ}-close. Hence using Lemma 4 we obtain

Fact 2 For each state $\langle \alpha, h, \mathrm{Alter} \rangle$ the probabilities $a[\langle \alpha, h, \mathrm{Alter} \rangle, R]$ and $a[\langle \alpha, h, \mathrm{Alter} \rangle, R']$ are $2^{2s\mu}$-close.

We consider now a probability that M accepts an input $Z \in \tilde{Z}$. Let R be the Markov chain for Z and let $\mathtt{ACCEPT}(Z)$ be the set of nondeterministic configurations (β, i) such that M starting in (β, i) with Z on the input tape accepts. Then

$$\Pr[\ M \text{ accepts } Z\ ] \quad = \sum_{(\beta,i) \in \mathtt{ACCEPT}(Z)} \Pr[\ (\alpha_0, 1)\ \models_Z\ (\beta, i)\ ]\ .$$

Fact 3 For any nondeterministic configuration (β, i) it holds

$$\Pr[\ (\alpha_0, 1)\ \models_Z\ (\beta, i)\ ] \quad = \sum_{\substack{\alpha-\text{random} \\ h \in \{h_1, h_2\}}} a[\ \langle \alpha, h, \mathrm{Alter} \rangle, R\ ] \cdot \Pr[\ (\alpha, h)\ \models_{Z,H}\ (\beta, i)\ ]\ ,$$

where h_1, h_2 are two consecutive numbers in H such that $h_1 \le i \le h_2$.

Proof. Let $A(\alpha, h)$ denote an event that M with Z on its input tape has a computation path $C_1 = (\alpha_0, 1), \ldots, C_t$ such that

- $C_1, \ldots, C_{t-1}$ are probabilistic and the last configuration C_t is nondeterministic;
- the input head position of C_t lies between h_1 and h_2;
- (α, h) is the last configuration with the input head position equals to h_1 or to h_2 in sequence $C_1, \ldots, C_t$.

Let $B(\alpha, h)$ denote the same event as $A(\alpha, h)$ but with the restriction that the last configuration $C_t = (\beta, i)$. Note that

$$\Pr[\ (\alpha_0, 1)\ \models_Z\ (\beta, i)\ ] \quad = \sum_{\substack{\alpha-\text{random} \\ h \in \{h_1, h_2\}}} \Pr[B(\alpha, h)]\ ,$$

since for any pair of configurations $(\alpha, h) \ne (\alpha', h')$ events $B(\alpha, h)$ and $B(\alpha', h')$ are independent. Note also that a probability that event $A(\alpha, h)$ holds is equal to $a[\ \langle \alpha, h, \mathrm{Alter} \rangle, R\ ]$.

Let (α, h) be an arbitrary probabilistic configuration. To prove the fact we show that

$$\Pr[B(\alpha,h)] = \Pr[A(\alpha,h)] \cdot \Pr[\ (\alpha,h) \models_{Z,H} (\beta,i)\ ]\ . \tag{ii}$$

Assume first that $\Pr[A(\alpha,h)] > 0$. Hence

$$\Pr[B(\alpha,h) \mid A(\alpha,h)] = \Pr[\ (\alpha,h) \models_{Z,H} (\beta,i)\ ]\ .$$

On the other hand

$$\Pr[B(\alpha,h) \mid A(\alpha,h)] = \frac{\Pr[B(\alpha,h)]}{\Pr[A(\alpha,h)]}$$

and hence equation (ii) holds.

It is obvious that for $\Pr[A(\alpha,h)] = 0$ the equation holds, too. ∎

Now define

$$P_1(Z) := \sum_{\substack{(\beta,i)\in \mathtt{ACCEPT}(Z)\\ 1<i<\bar{N}}} \ \sum_{\substack{\alpha-\text{random}\\ h\in\{1,\bar{N}\}}} a[\ \langle\alpha,h,\text{Alter}\rangle, R\ ] \cdot \Pr[\ (\alpha,h) \models_{Z,H} (\beta,i)\ ]$$

$$P_2(Z) := \sum_{\substack{(\beta,i)\in \mathtt{ACCEPT}(Z)\\ \bar{N}+1<i<2\bar{N}}} \ \sum_{\substack{\alpha-\text{random}\\ h\in\{\bar{N}+1,\ 2\bar{N}\}}} a[\ \langle\alpha,h,\text{Alter}\rangle, R\ ] \cdot \Pr[\ (\alpha,h) \models_{Z,H} (\beta,i)\ ]$$

$$P_3(Z) := \sum_{\substack{(\beta,i)\in \mathtt{ACCEPT}(Z)\\ 2\bar{N}+1<i<N+1}} \ \sum_{\substack{\alpha-\text{random}\\ h\in\{2\bar{N}+1,\ N+1\}}} a[\ \langle\alpha,h,\text{Alter}\rangle, R\ ] \cdot \Pr[\ (\alpha,h) \models_{Z,H} (\beta,i)\ ]$$

W.l.o.g. assume that M does not alternate with the input head positions in H. Hence and by Fact 3 we have

$$\Pr[\ M \text{ accepts } Z\ ] = P_1(Z) + P_2(Z) + P_3(Z)\ .$$

By the fact that X and Y have the same word transitions (i.e. $\mathrm{t}(X) = \mathrm{t}(Y)$), we conclude that for any i with $1 < i < \bar{N}$ and for a pair of input strings $Z = Z_{XY}, Z' = Z_{XX}$ or $Z = Z_{YX}, Z' = Z_{YY}$ i.e. for a pair of inputs from $\tilde{Z}$ with the same substring on positions $1, 2, \ldots, \bar{N}$ holds:

$$(\beta,i) \in \mathtt{ACCEPT}(Z) \qquad \Longleftrightarrow \qquad (\beta,i) \in \mathtt{ACCEPT}(Z')\ .$$

This equivalence also holds for i with $\bar{N}+1 < i < 2\bar{N}$ and a pair of inputs: $Z = Z_{XY}, Z' = Z_{YY}$ or $Z = Z_{YX}, Z' = Z_{XX}$ as well as for i with $2\bar{N}+1 < i < N+1$ and any pair of substrings $Z, Z' \in \tilde{Z}$. Therefore, by Fact 2 we obtain the following property of machine M working with inputs from $\tilde{Z}$.

Fact 4 The vectors

$$\begin{array}{ll}(\ P_1(Z_{YY}),\quad P_2(Z_{YY}) + P_3(Z_{YY}),\quad P_1(Z_{XY}),\quad P_2(Z_{YX}) + P_3(Z_{YX})\ ) & \text{and}\\ (\ P_1(Z_{YX}),\quad P_2(Z_{XY}) + P_3(Z_{XY}),\quad P_1(Z_{XX}),\quad P_2(Z_{XX}) + P_3(Z_{XX})\ ) & \end{array}$$

are $2^{2s\mu}$ -close.

Note that because $2^{2s\mu}$ is close to one, the above fact says that the appropriate values are very close to each other.

Now we can easily show that the machine M is unable to accept neither `PATTERN` nor its complement.

Proof of Lemma 1 Let us assume to the contrary that M recognizes `PATTERN` with probability $1-\epsilon$, where $\epsilon < 1/4$. Since $Z_{XY}, Z_{YX}, Z_{XX} \in \mathtt{PATTERN}$ we have that for these inputs Z, Pr[M accepts Z] $\geq 3/4$. Hence, and by Fact 4 the probability that M accepts the input Z_{YY} can be estimated as follows:

$$\begin{aligned}
\text{Pr}[\ M \text{ accepts } Z_{YY}\ ] &= P_1(Z_{YY}) + P_2(Z_{YY}) + P_3(Z_{YY}) \\
&\geq 2^{-2s\mu} \cdot (P_1(Z_{YX}) + P_2(Z_{XY}) + P_3(Z_{XY})) \\
&\geq 2^{-2s\mu} \cdot (6/4 - (P_2(Z_{YX} + P_3(Z_{YX}) + P_1(Z_{XY}))) \\
&\geq 2^{-2s\mu} \cdot (6/4 - 2^{2s\mu} \cdot (P_1(Z_{XX}) + P_2(Z_{XX}) + P_3(Z_{XX}))) \\
&\geq 2^{-2s\mu} \cdot 6/4 - 1\ .
\end{aligned}$$

But for the chosen value μ and sufficiently large n we have

$$2^{-2s\mu} \cdot 6/4 - 1 \ = \ 2^{-\mu\cdot(28Vol(N)+8)} \cdot 6/4 - 1 \ > \ 1/3\ .$$

what yields a contradiction since $Z_{YY} \notin$ `PATTERN`. ∎

Proof of Lemma 3 Let assume now that M recognizes $\overline{\texttt{PATTERN}}$. Hence we have that for $Z \in \{Z_{XY}, Z_{YX}, Z_{XX}\}$ Pr[M accepts Z] $\leq 1/4$. On the other hand, by Fact 4 we have

$$\begin{aligned}
\text{Pr}[\ M \text{ accepts } Z_{YY}\ ] &= P_1(Z_{YY}) + P_2(Z_{YY}) + P_3(Z_{YY}) \\
&\leq 2^{2s\mu} \cdot (P_1(Z_{YX}) + P_2(Z_{XY}) + P_3(Z_{XY})) \\
&\leq 2^{2s\mu} \cdot (1/2 - (P_2(Z_{YX} + P_3(Z_{YX}) + P_1(Z_{XY}))) \\
&\leq 2^{2s\mu} \cdot (1/2 - 2^{-2s\mu} \cdot (P_1(Z_{XX}) + P_2(Z_{XX}) + P_3(Z_{XX}))) \\
&\leq 2^{2s\mu} \cdot 1/2 \\
&< 2/3\ .
\end{aligned}$$

This yields a contradiction since $Z_{YY} \in \overline{\texttt{PATTERN}}$. ∎

4 Space efficient algorithms for STMs

In this section we show a 2-round, llog -space bounded STM M that starting in nondeterministic mode recognizes `PATTERN` . This proves Lemma 2. An algorithm for M is based on [Fr79].

The machine M checks first whether the input is of the form $w_1 * w_2 * \ldots * w_t * u\ \texttt{BIN}(2^k)$ for some integer $k \geq 0$ and words $w_1, w_2, \ldots, w_t, u \in \{0,1\}^k$. Then it verifies the condition $u \in \{w_1, \ldots, w_t\}$ as follows:

- existentially it guesses an index i with $1 \leq i \leq t$;
- randomly it chooses a prime q with $2 \leq q \leq k^2$ and then computes $r := n_{w_i} \bmod q$, where n_{w_i} denotes an integer with the binary representation w_i;
- it accepts if $r = n_u \bmod q$; otherwise it rejects.

The integers r and q are stored in binary on the work tape. It is obvious that M uses $O(\text{llog}\ n)$ space. The correctness of M follows from [Fr79].

5 Conclusions and Open Problems

In this paper a separation was obtained for sublogarithmic $AM_k Space$ complexity classes on the second level of the round/alternation hierarchy. An interesting open problem is if our separation can be generalized on any level. Is it true that similar to ATMs, this hierarchy is infinite?

How looks the alternating stochastic hierarchy for at least logarithmic space bounds? Using a simple simulation of space-bounded NTMs by one-sided-error probabilistic TMs (see e.g. [Gil77] or the survey paper [Ma95]) one can easily show that

$$AM_2 Space(\log) \;=\; AM_1 Space(\log) ,$$

that means the AM_2 -class is quite weak in case of space bounds – contrary to time bounded classes. Is it also true that

$$AM_2 Space(\texttt{SUBLOG}) \;=\; AM_1 Space(\texttt{SUBLOG}) \ ?$$

References

[Ba85] L. Babai, *Trading group theory for randomness,* in Proceedings of the 17th ACM Symposium on Theory of Computing, ACM Press, 1985, 421-429.

[Co89] A. Condon, *Computational model of games,* MIT Press, 1989.

[Co93] A. Condon, *The complexity of space bounded interactive proof systems,* in Complexity Theory: Current Research, S. Homer, U. Schöning, K. Ambos-Spies (Ed.), Cambridge Univ. Press, 1993, 147-190.

[CHPW94] A. Condon, L. Hellerstein, S. Pottle, and A. Wigderson, *On the power of finite automata with both nondeterministic and probabilistic states,* in Proceedings of the 26th ACM Symposium on Theory of Computing, ACM Press, 1994, 676-685.

[DwSt92] S. Dwork and L. Stockmeyer, *Finite state verifiers I: the power of interaction,* Journal of the ACM, 39, 1992, 800-828.

[Fr79] R. Freivalds, *Fast probabilistic algorithms,* in Proceedings of the 8th International Symposium on Mathematical Foundations of Computer Science, Springer-Verlag, Heidelberg, 1979, 57-69.

[FrKa94] R. Freivalds and M. Karpinski, *Lower space bounds for randomized computation,* in Proceedings of the 21st International Colloquium on Automata, Languages, and Programming, Springer-Verlag, Heidelberg, 1994, 580-592.

[Gil77] J. Gill, *Computational complexity of probabilistic Turing machines,* SIAM Journal on Computing, 7, 1977, 675-695.

[GMR89] S. Goldwasser, S. Micali, and C. Rackoff, *The knowledge complexity of interactive proof systems,* SIAM Journal on Computing, 18, 1989, 186-208.

[GoSi86] S. Goldwasser and M. Sipser, *Private coins versus public coins in interactive prove systems,* in Proceedings of the 18th ACM Symposium on Theory of Computing, ACM Press, 1986, 59-68.

[LiRe93] M. Liśkiewicz and R. Reischuk, *The sublogarithmic space world,* Technical Report 048-93 ICSI Berkeley, to appear in SIAM Journal on Computing.

[Ma95] I. Macarie, *Space-bounded probabilistic computation: old and new stories,* SIGACT News, vol. 26(3), 1995, 2-12.

[Pa85] C. Papadimitriou, *Games against nature,* Journal of Computer and System Sciences, 31, 1985, 288-301.