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

Maciej Liśkiewicz*
Rüdiger Reischuk†

TR-96-025

July 1996

Abstract

This paper studies interactive proof systems using public coin tosses, respectively Arthur-Merlin games, with a sublogarithmic space-bounded verifier. We provide examples of specific languages and show that such systems working with bounded number of rounds of interaction are unable to accept these languages. As a consequence, a separation of the second and the third level of the round/alternation hierarchy is obtained. It is well known that such a property does not hold for the corresponding polynomial time classes: in ["Proceedings of the 17th ACM Symposium on Theory of Computing", ACM Press, 1985, 421-429] Babai showed that the hierarchy of complexity classes $AM_kTime(\mathcal{POL})$ collapses to the second level.

*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

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 nondeterministic 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 $MA_kSpace(S)$ (resp. $AM_kSpace(S)$) denote the set of languages that can be accepted by such a machine in space S making at most $k-1$ alternations between nondeterministic and probabilistic configurations and starting in nondeterministic (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.

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 $AM_kTime(\mathcal{POL})$. Here, $\mathcal{POL}$ denotes the set of all polynomials. Furthermore, it has been shown [Ba85] that for polynomial time any number of rounds can be reduced to two rounds, that is

$$MA_2Time(\mathcal{POL}) \subseteq AM_2Time(\mathcal{POL}) = AM_{\mathcal{CON}}Time(\mathcal{POL}) .$$

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 $S \in \mathcal{SUBLOG} := \Omega(\mathrm{llog}) \cap o(\log)$ the situation is different. Here llog denotes the logarithmic function log iterated twice.

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

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

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:

$$BPSpace(S) = AM_1Space(S) \subset AMSpace(S) \subset AMSpace(\log) = \mathrm{P} .$$

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

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

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*

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

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*

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

Moreover, M works in polynomial time.

Therefore, we obtain the following separation

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

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

$$\overline{\texttt{PATTERN}} \notin MA_3Space(o(\log)) .$$

Obviously, this result implies that at least for sublogarithmic space bounds S the classes $MA_2Space(S)$ and $MA_3Space(S)$ are not closed under complement. An interesting open problem is whether the same holds for remaining $MA_kSpace(S)$ and $AM_kSpace(S)$ classes.

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 bound 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). A *configuration* of M on an input X is a pair (α, j) consisting of a memory state α and a position j with $0 \leq j \leq |X| + 1$ of the input head. $j = 0$ or $j = |X| + 1$ means that this head scans the left, resp. the right end-marker. We say that a configuration (α, j), with $\alpha = (q, u, i)$, is nondeterministic, probabilistic (or random), accepting or rejecting, according to q.

We call a phase of computation of M a *probabilistic round* if M starts the phase in a probabilistic configuration and makes only probabilistic steps during the phase. Analogously we call a phase of computation a *nondeterministic round* if M starts in a nondeterministic configuration and performs only nondeterministic steps during the phase. Let for a probabilistic configuration C and an arbitrary configuration D

$$Rmove_{X,C}(D)$$

denote the event that M with X on its input tape and starting in C reaches the configuration D in a probabilistic round. Let for a nondeterministic configuration C

$$Nmove_{X,C}(D)$$

denote the predicate that is true if and only if M starting in C on the input X reaches the configuration D in a nondeterministic round.

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 $h \in \{1, |Z|\}$ what means that M starts according to the value of h on the first or on the last symbol 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 in a probabilistic round the input head falls off the h-th symbol of Z with M in memory state α,
2. "R-Loop" meaning that the probabilistic computation of M loops forever within Z, or
3. "Alter$_{\alpha,h}$" meaning that within Z M alternates from a probabilistic to a nondeterministic round 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 condition 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 $p(Z, \langle \alpha, h \rangle, \text{Alter}_{\alpha',h'}) = 0$ if $\langle \alpha, h \rangle \neq \langle \alpha', h' \rangle$.

We model computations of a probabilistic round 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 $p^*[k, R]$ 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 $p^*[k, R]$ and $p^*[k, R']$ are β^z-close where $z = 2s$.

We characterize a word Z according to a nondeterministic round of M on Z by *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 $h \in \{1, |Z|\}$. A stopping condition for the word transition is either:

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

Let us assume that M be an arbitrary STM, of space complexity $S \in o(\log)$. 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.

Proof of Lemma 1. We will show that M making first a probabilistic and than a nondeterministic round can be fooled when it works on the inputs of the form

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

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

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

Note that $Vol(N) \leq 2^{O(S(N))}$. Moreover, since $S(N) \in o(\log N)$ note also that $S(N) \in o(\log n)$.

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

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

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 $Z \in W_n$ the number of word probabilities

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

and the number of word transitions

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

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 p(Z) be the vector of the d_p probabilities and let 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 $|E_n|/2^{d_t}$. To guarantee the last condition we want to have

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

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. 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 phase of 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 $\texttt{BIN}(n)$ \$. $h = 0$ 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 "$\text{Alter}_{\alpha,h}$", with α and h as above. Its meaning is: M alternates from a probabilistic to a nondeterministic round scanning for the last time before this alternation a position in H in configuration (α, h). The last four states are the following: $\text{R-Loop}_1, \text{R-Loop}_2, \text{R-Loop}_3, \text{R-Loop}_4$. For chain R_{XY} they mean that the probabilistic computation of M loops forever on the left end-marker, within X, Y, x $\texttt{BIN}(n)$ \$, respectively. For remaining chains the meaning is analogous. Let $C_0 = (\alpha_0, 1)$ 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 $\texttt{BIN}(n)$ \$, and assuming the following definition of the set of trapping states:

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

More precisely, the transitions r_{ij} such that states i, j are applied both to the same substring: \$, X, Y, or x $\texttt{BIN}(n)$ \$ 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 $r_{ij} = p(Y, \langle\alpha, \bar{N}\rangle, \langle\beta, 1\rangle)$ since the position $2\bar{N}$ and $\bar{N}+1$ means the last resp., the first symbol of the substring Y in Z_{XY}. The transitions $r_{kk} = 1$ for any trapping state 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 $p(X)$ and $p(Y)$ are componentwise 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 $\text{Alter}_{\alpha,h}$ the probabilities $p^*[\text{Alter}_{\alpha,h}, R]$ and $p^*[\text{Alter}_{\alpha,h}, R']$ 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 $\texttt{ACCEPT}(Z)$ be the set of nondeterministic configurations D such that M starting in D with Z on the input tape accepts in a nondeterministic round. Then

$$\Pr[\ M \text{ accepts } Z\ ] \ = \sum_{D \in \text{ACCEPT}(Z)} \Pr[\ Rmove_{Z,C_0}(D)\ ] \ .$$

Let $Z_{j,\ell}$, with $0 \le j \le \ell \le |Z|$ denote the substring of the word Z consisting of j-th, $j+1$-st, ..., ℓ-th symbol of the word Z. Moreover, let for a probabilistic configuration (α, h), where $h \in \{j, \ell\}$, and for a nondeterministic configuration (β, i), with $j \le i \le \ell$,

$$b(Z_{j,\ell}, \alpha, h, \beta, i)$$

be the conditional probability that M started in configuration (α, h) and making a probabilistic round with the input head within $Z_{j,\ell}$ reaches configuration (β, i) under the condition that M alternates from a probabilistic to a nondeterministic round working within $Z_{j,\ell}$ when started in (α, h).

Fact 3 Let (β, i) be a nondeterministic configuration, and let h_1, h_2 be two consecutive numbers in H such that $h_1 \leq i \leq h_2$. Then it holds that

$$\Pr[\ Rmove_{Z,C_0}(\beta, i)\ ] \ = \sum_{\substack{\alpha-\text{random} \\ h \in \{h_1, h_2\}}} p^*[\text{Alter}_{\alpha,h}, R] \cdot b(Z_{h_1,h_2}, \alpha, h, \beta, i)\ .$$

Proof. Denote by $A(\alpha, h)$, with $h \in \{h_1, h_2\}$, the event that M with Z on its input tape has a computation path $C_0, \ldots, C_t$ such that

- $C_0, \ldots, C_{t-1}$ are probabilistic and the last configuration C_t is nondeterministic;
- the input head position of C_t lies between h_1 and h_2;
- in the sequence $C_0, \ldots, C_t$ (α, h) is the last configuration with the input head position equals to h_1 or to h_2.

Let $B(\alpha, h)$ denote the same event as $A(\alpha, h)$ but with the restriction that the last configuration $C_t = (\beta, i)$. Moreover, let $A_{h_1,h_2}(\alpha, h)$, be the event that M alternates from a probabilistic to a nondeterministic round working within Z_{h_1,h_2} when started in (α, h) and let $B_{h_1,h_2}(\alpha, h)$ denote the event that M started in configuration (α, h) and working in a probabilistic phase with the input head within Z_{h_1,h_2} reaches configuration (β, i). Obviously,

$$\Pr[A(\alpha, h)] = \sum_{\mathcal{C} \in \text{RPATH}(\alpha,h)} \Pr[M \text{ performs } \mathcal{C}] \cdot \Pr[A_{h_1,h_2}(\alpha, h)]\ ,$$

where $\texttt{RPATH}(\alpha, h)$ denotes the set of all probabilistic paths of computation that start in the initial configuration C_0 and end in (α, h). Similarly

$$\Pr[B(\alpha, h)] = \sum_{\mathcal{C} \in \text{RPATH}(\alpha,h)} \Pr[M \text{ performs } \mathcal{C}] \cdot \Pr[B_{h_1,h_2}(\alpha, h)]\ .$$

If $\Pr[A(\alpha, h)] \neq 0$ then it holds that

$$\frac{\Pr[B(\alpha, h)]}{\Pr[A(\alpha, h)]} \ = \ \frac{\Pr[B_{h_1,h_2}(\alpha, h)]}{\Pr[A_{h_1,h_2}(\alpha, h)]} \ = \ b(Z_{h_1,h_2}, \alpha, h, \beta, i)$$

since $B_{h_1,h_2}(\alpha, h) \subseteq A_{h_1,h_2}(\alpha, h)$. Hence

$$\Pr[B(\alpha, h)] = \Pr[A(\alpha, h)] \cdot b(Z_{h_1,h_2}, \alpha, h, \beta, i)\ .$$

Clearly, this equality holds in case $\Pr[A(\alpha, h)] = 0$, too. Therefore we have

$$\Pr[\ Rmove_{Z,C_0}(\beta, i)\ ] \ = \sum_{\substack{\alpha-\text{random} \\ h \in \{h_1, h_2\}}} \Pr[B(\alpha, h)] \ = \sum_{\substack{\alpha-\text{random} \\ h \in \{h_1, h_2\}}} \Pr[A(\alpha, h)] \cdot b(Z_{h_1,h_2}, \alpha, h, \beta, i)\ ,$$

what proves the fact since $\Pr[A(\alpha, h)] = p^*[\text{Alter}_{\alpha,h}, R]$. ∎

Now define

$$P_1(Z) \ := \sum_{\substack{(\beta,i) \in \text{ACCEPT}(Z) \\ 1 < i < \bar{N}}} \ \sum_{\substack{\alpha-\text{random} \\ h \in \{1, \bar{N}\}}} p^*[\text{Alter}_{\alpha,h}, R] \cdot b(Z_{1,\bar{N}}, \alpha, h, \beta, i)$$

$$P_2(Z) \quad := \sum_{\substack{(\beta,i)\in \mathrm{ACCEPT}(Z)\\ \bar{N}+1<i<2\bar{N}}} \sum_{\substack{\alpha-\mathrm{random}\\ h\in\{\bar{N}+1,\ 2\bar{N}\}}} p^*[\mathrm{Alter}_{\alpha,h}, R]\cdot b(Z_{\bar{N}+1,2\bar{N}},\alpha,h,\beta,i) +$$

$$\sum_{\substack{(\beta,i)\in \mathrm{ACCEPT}(Z)\\ 2\bar{N}+1<i<N+1}} \sum_{\substack{\alpha-\mathrm{random}\\ h\in\{2\bar{N}+1,\ N+1\}}} p^*[\mathrm{Alter}_{\alpha,h}, R]\cdot b(Z_{2\bar{N}+1,N+1},\alpha,h,\beta,i)$$

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

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

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:

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

This equivalence also holds for i with $\bar{N}+1 < i < N+1$ and a pair of inputs: $Z = Z_{XY}, Z' = Z_{YY}$ or $Z = Z_{YX}, Z' = Z_{XX}$.

Fact 4 The vectors

$$\begin{array}{llll} (\ P_1(Z_{YY}), & P_2(Z_{YY}), & P_1(Z_{XY}), & P_2(Z_{YX})\ ) \\ (\ P_1(Z_{YX}), & P_2(Z_{XY}), & P_1(Z_{XX}), & P_2(Z_{XX})\ ) \end{array} \qquad \text{and}$$

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

Proof. We show that the numbers $P_1(Z_{YY})$, and $P_1(Z_{YX})$ are $2^{2s\mu}$-close. The proof that the remaining pairs are $2^{2s\mu}$-close is similar and we omit it here.

From the definition we have

$$P_1(Z_{YY}) \ = \sum_{\substack{(\beta,i)\in \mathrm{ACCEPT}(Z_{YY})\\ 1<i<\bar{N}}} \sum_{\substack{\alpha-\mathrm{random}\\ h\in\{1,\bar{N}\}}} p^*[\mathrm{Alter}_{\alpha,h}, R_{YY}]\cdot b(Y,\alpha,h,\beta,i)\ .$$

Since for each configuration $C = (\beta, i)$, with $1 < i < \bar{N}$, $C \in \texttt{ACCEPT}(Z_{YX})$ if and only if $C \in \texttt{ACCEPT}(Z_{YY})$ (by the equality (ii)) hence

$$P_1(Z_{YX}) \ = \sum_{\substack{(\beta,i)\in \mathrm{ACCEPT}(Z_{YY})\\ 1<i<\bar{N}}} \sum_{\substack{\alpha-\mathrm{random}\\ h\in\{1,\bar{N}\}}} p^*[\mathrm{Alter}_{\alpha,h}, R_{YX}]\cdot b(Y,\alpha,h,\beta,i)\ .$$

The claim follows since by Fact 2, for each state $\mathrm{Alter}_{\alpha,h}$ the probabilities $p^*[\mathrm{Alter}_{\alpha,h}, R_{YY}]$ and $p^*[\mathrm{Alter}_{\alpha,h}, R_{YX}]$ 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 AM_2 machine M is unable to accept **PATTERN** . Let us assume to the contrary that M recognizes the language with probability $1-\epsilon$, where $\epsilon < 1/4$. Since $Z_{XY}, Z_{YX}, Z_{XX} \in$ **PATTERN** we have that for these inputs Z, $\Pr[\ M \text{ accepts } Z\ ] \geq 3/4$. Hence, and by Fact 4 the probability that M accepts the input Z_{YY} can be estimated as follows:

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

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 $Z_{YY} \notin \texttt{PATTERN}$. ∎

Proof of Lemma 3 (Sketch). Assume now that M is a MA_3 machine that recognizes $\overline{\texttt{PATTERN}}$. We show that M can be fooled when it works on the inputs of the form

$$w_1 * \ldots * w_n * w_{n+1} * \ldots * w_{2n} * w_{2n+1} * \ldots w_{3n} * u * \texttt{BIN}(n) \;,$$

with $|w_i| = |u| = k$. To show this we slightly modify the previous proof.

Let $N := 3n \cdot (k+1) + k + 1 + |\texttt{BIN}(n)|$ be the length of the considered input words and let X and Y be two words in W_n as described in the proof of Lemma 1. Now we consider a behaviour of M on the input

$$Z_{YY} \;=\; Y\ Y\ Y\ x\ \texttt{BIN}(n)$$

that obviously belongs to $\overline{\texttt{PATTERN}}$. Let p be the probability that M accepts the input, i.e. let

$$p := \max_{C-\text{universal}} \{\Pr[M \text{ accepts } Z_{YY} \text{ started in } C] \mid \; Nmove_{Z_{YY},C_0}(C)\} \;.$$

Denote by $C_p = (\alpha_p, i_p)$ a probabilistic configuration, with $Nmove_{Z_{YY},C_0}(C_p)$, and the probability that M accepts Z_{YY} started in C_p equal to p. Then three cases can occur:

- $0 \le i_p \le \bar{N}$,
- $\bar{N}+1 \le i_p \le 2\bar{N}$, or
- $2\bar{N}+1 \le i_p \le N+1$.

We will consider the first case only. The remaining cases are analogous and we omit them here.

Similarly as in the proof of Lemma 1 we will consider Markov chains R_{XY}, R_{YX}, R_{XX}, and R_{YY} which model the probabilistic round of the machine M on inputs

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

and on Z_{YY}, respectively. Each chain has $s \;=\; 20 \cdot Vol(N) + 5$ states. The first $10 \cdot Vol(N)$ states have the form $\langle \alpha, h \rangle$, where α is a memory state, and

$$h \;\in \{0,\ i_p,\ i_p+1,\ \bar{N},\ \bar{N}+1,\ 2\bar{N},\ 2\bar{N}+1,\ 3\bar{N},\ 3\bar{N}+1,\ N+1\} \;.$$

The next $10 \cdot Vol(N)$ states of each chain have the form "$\text{Alter}_{\alpha,h}$", with α and h as above. The last five states are the following: R-Loop$_j$ with $j = 1, \ldots, 5$. The meaning of the states and the definition of the transitions r_{ij} is the same as in the proof of Lemma 1. The only difference is that we assume now that the initial state of each chain is $\langle \alpha_p, i_p \rangle$. Obviously, for any pair of the Markov chains Fact 2 holds.

Let us define for each word $Z \in \{Z_{XY}, Z_{YX}, Z_{XX}, Z_{YY}\}$

$$P_1(Z) \;:=\; \sum_{\substack{(\beta,i) \in \text{ACCEPT}(Z) \\ 0 < i < i_p}} \; \sum_{\substack{\alpha-\text{random} \\ h \in \{0, i_p\}}} p^*[\text{Alter}_{\alpha,h}, R] \cdot b(Z_{0,i_p}, \alpha, h, \beta, i) \;+$$

$$\sum_{\substack{(\beta,i)\in \mathrm{ACCEPT}(Z)\\ i_p+1<i<\bar{N}}} \sum_{\substack{\alpha-\mathrm{random}\\ h\in\{i_p+1,\bar{N}\}}} p^*[\mathrm{Alter}_{\alpha,h},R]\cdot b(Z_{i_p+1,\bar{N}},\alpha,h,\beta,i) \;+$$

$$\sum_{\substack{(\beta,i)\in \mathrm{ACCEPT}(Z)\\ \bar{N}+1<i<2\bar{N}}} \sum_{\substack{\alpha-\mathrm{random}\\ h\in\{\bar{N}+1,2\bar{N}\}}} p^*[\mathrm{Alter}_{\alpha,h},R]\cdot b(Z_{\bar{N}+1,2\bar{N}},\alpha,h,\beta,i)$$

$$P_2(Z) \;:=\; \sum_{\substack{(\beta,i)\in \mathrm{ACCEPT}(Z)\\ 2\bar{N}+1<i<3\bar{N}}} \sum_{\substack{\alpha-\mathrm{random}\\ h\in\{2\bar{N}+1,\ 3\bar{N}\}}} p^*[\mathrm{Alter}_{\alpha,h},R]\cdot b(Z_{2\bar{N}+1,3\bar{N}},\alpha,h,\beta,i) \;+$$

$$\sum_{\substack{(\beta,i)\in \mathrm{ACCEPT}(Z)\\ 3\bar{N}+1<i<N+1}} \sum_{\substack{\alpha-\mathrm{random}\\ h\in\{3\bar{N}+1,\ N+1\}}} p^*[\mathrm{Alter}_{\alpha,h},R]\cdot b(Z_{3\bar{N}+1,N+1},\alpha,h,\beta,i)$$

It is easy to see that for the values as defined above Fact 4 holds.

Now we are ready to estimate the value p. Note that for any $Z \in \{Z_{XY}, Z_{YX}, Z_{XX}\}$ because word transitions $t(X)$ and $t(Y)$ are equal and because we have that $Nmove_{Z_{YY},C_0}(C_p)$, $Nmove_{Z,C_0}(C_p)$ holds, too. Therefore the probability that M accepts Z starting in C_p does not exceed $1/4$ since $Z \notin \overline{\texttt{PATTERN}}$. On the other hand, by Fact 4 we have

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

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 $k \ge 0$ 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:

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

If for some i, with $1 \le i \le t$, the strings w_i and u are equal then of course $n_{w_i} = n_u \bmod q$ for any value q and machine M accepts correctly in step 3. If $w_i \ne u$ than it could happen that $n_{w_i} = n_u \bmod q$ and M reaches in step 3 the accepting state that is wrong however. We show that this happens with small probability. Indeed. Since $|n_{w_i} - n_u| \le 2^k$ hence $n_{w_i} - n_u$ has at most 2^k

different prime divisors. On the other hand, M chooses from about $\frac{k^2}{2\log_e k}$ different primes at the beginning of step 2. So the probability that it chooses value q that divides both n_{w_i} and n_u is at most $\frac{2\log_e k}{k}$ what tends to 0.

Obviously, M uses $O(\text{llog } n)$ space.

5 Conclusions and Open Problems

In this paper a separation was obtained for sublogarithmic AM_kSpace 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 round/alternation 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

$$AM_2Space(\log) \;=\; AM_1Space(\log)\ ,$$

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

$$AM_2Space(\mathcal{SUBLOG}\ ) \;=\; AM_1Space(\mathcal{SUBLOG}\ )\ ?$$

What is the situation for space bounds S smaller than llog? The most interesting case seems to be space bounds restricted to constant functions. It is well known that $MA_1Space(\mathcal{CON}) = NSpace(\mathcal{CON})$ coincides with the class of regular languages. This result, however, does not extend to the probabilistic classes. Freivalds has shown the surprising result [Fr81] that

$$\texttt{COUNT} \;:=\; \{1^n01^m \mid n = m\},$$

can be accepted by a probabilistic TM in constant space with an arbitrarily small constant for the error probability. Is there a language that separate $AM_1Space(\mathcal{CON})$ and $AM_kSpace(\mathcal{CON})$ classes, for some $k > 1$? Dwork and Stockmeyer ([DwSt92]) showed that `CENTER` does not belong to $AM_1Space(\mathcal{CON})$ and that there exists a constant space interactive proof system for this language ([DwSt92]). Can the system be improved to make only constant number of rounds?

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.

[Fr81] R. Freivalds, *Probabilistic 2-way machines,* in Proceedings of the 10th International Symposium on Mathematical Foundations of Computer Science, Springer-Verlag, Heidelberg, 1981, 33-45.

[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.