There Is No Polynomial Deterministic Space Simulation of Probabilistic Space with a Two-Way Random-Tape Generator *

Marek Karpinski †
Dept. of Computer Science
University of Pittsburgh
Pittsburgh, Pennsylvania 15213

Rutger Verbeek
Department of Computer Science
University of Bonn
5300 Bonn 1

Abstract

We prove there is no polynomial deterministic space simulation for two-way random-tape probabilistic space (Pr₂SPACE) (as defined in [BCP 83]) for all functions \( f : \text{IN} \rightarrow \text{IN} \) and all \( \alpha \in \text{IN} \), \( \text{Pr}_2 \text{SPACE}(f(n)) \not\subseteq \text{DSPACE}(f(n)^\alpha) \). This is answer to the problem formulated in op cit., whether the deterministic squared-space simulation (for recognizers and transducers) generalizes to the two-way random-tape machine model. We prove, in fact, a stronger result saying that even space-bounded Las Vegas two-way random-tape algorithms (yielding always the correct answer and terminating with probability 1) are exponentially more efficient than the deterministic ones.

†Supported by the Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA 15213
1 Introduction

Jung (1981) and Borodin, Cook, and Pippenger (1983) prove that both the probabilistic acceptors and transducers working in space \( f(n) \geq \log n \) can be simulated in deterministic \( f(n)^2 \) space. The definition of probabilistic Turing machines uses a one-way read-only random tape. The model of probabilistic machine [Gi 77] may be reviewed as a deterministic machine with a one-way only access to the random bits sequence. A two-way random tape proposed in [BCP 83] allows multiple access to the random bits sequence which is stored on the two-way read-only tape. The problem posed in [BCP 83] whether the \( f(n)^2 \) deterministic space simulation holds also for the two-way random-tape \((\Pr_2 \text{SPACE}(f(n)))\).

Let \( \Psi \subseteq \Sigma^* \times \{0,1\}^\omega \) be a binary predicate, where \( \Psi(x, y) \) is computed by a deterministic machine \( M \) with two two-way read-only input tapes. If \( M \) stops on an initial segment of \( Y \), then \( \Psi(x, y) \) is defined. \( x \in \Sigma^* \) is recognized by \( M \) if and only if \( \Pr\{\Psi(x, y) = \text{true}\} > \frac{1}{2} \). We call \( M \) a probabilistic machine (over the alphabet \( \Sigma \)) with two-way random tape. Let \( L_M \subseteq \Sigma^* \) denote the set recognized by \( M \). If \( M \) is \( S(|x|) \) space bounded, then \( L_M \) belongs to the two-way random-tape probabilistic space \( S(n), L_M \in \Pr_2 \text{SPACE}(S(n)) \). If in addition \( M \) is \( T(|x|) \) time bounded, then \( L_M \in \Pr_2 \text{TISP}(T(n), S(n)) \). We say that \( L_M \) belongs to the two-way Las Vegas [BGM 82] space \( S(n), L_M \in \Delta_2 \text{SPACE}(S(n)) \), if for all \( x \in \Sigma^* \) either \( \Pr\{\Psi_M(x, y) = \text{true}\} = 1 \) or \( \Pr\{\Psi_M(x, y) = \text{false}\} = 1 \).

We prove that the class of \( \log F(n) \) space bounded Las Vegas algorithms with two-way random-tape (terminating with probability 1 and yielding always the correct result) denoted by \( \Delta_2 \text{SPACE}(\log f(n)) \) (time bounded Las Vegas algorithms are defined in [AM 77]; [BGM 82] are as powerful as \( \text{DSPACE}(f(n)) \)). Therefore there is no polynomial simulation for this class, which answers the problem of [BCP 83].

2 Remarks

1. This result is related to the recent result of Savitch and Dymond ([SD 84]) that “consistent” NSPACE is exponentially more powerful than DSPACE. The similarity becomes clear, if the reset mechanism in the original definition of consistent NSPACE is replaced by a two-way tape, of which the initial nondeterministic choices are stored. The proof of our Theorem 2 can be applied to this case.

2. The model of a probabilistic machine with two-way random tape may be viewed
as a deterministic machine with a random oracle stored on a two-way tape. The oracle tape records the outcome of an infinite sequence of independent unbiased coin tosses. The classical model of Gill ([Gi 77]) may be viewed as a deterministic machine with a random oracle stored on a one-way tape. The classical oracle machine ([BG 81]) is a deterministic machine with oracle stored on a derive resembling random-access store rather than tape (i.e., the question must be written on a query tape within the space bound). Denote by \( \text{DSpace}^{(A)}(f(n)) \) the class of sets recognized by \( f(n) \) space bounded deterministic Turing machines with oracle \( A \) stored on a two-way tape. Then, with probability 1 (i.e., for almost all oracles), \( \text{DSpace}^{(A)}(f(n)) \nsubseteq \Delta_2 \text{SPACE}(f(n)) \) (the inequivalence results from the fact that, with probability 1, \( A \nsubseteq \Delta_2 \text{SPACE}(f(n)) \)).

3 Results

**Theorem 1.** For every function \( f : \text{IN} \rightarrow \text{IN} \),

\[
\bigcup_{k \in \text{IN}} \Delta_2 \text{TISP}(2^{2^k \log f(n)}, \log f(n)) \supseteq \text{DSpace}(f(n)).
\]

**Corollary.** For every function \( f \),

\[
\text{Pr}_2 \text{SPACE}(\log f(n)) \supseteq \Delta_2 \text{SPACE}(\log f(n)) \supseteq \text{DSpace}(f(n)).
\]

**Corollary (Problem of [BCP 83]).**

\[
\text{Pr}_2 \text{SPACE}(f(n)) \nsubseteq \text{DSpace}(f(n)^2).
\]

**Proof of Theorem 1.** Suppose \( \mathcal{T} \) is a \( f(n) \) space bounded deterministic Turing machine with one work tape. Suppose that \( \mathcal{T} \) stops on every input (see [Si 80]).

For \( x \in \Sigma^* \), \( \text{comp}_\mathcal{T}(x) \in \Sigma^* \) will denote the computation of \( \mathcal{T} \) over \( x \) (not recording the input or input position). The probability that the random tape will contain as a subsequence \( \text{comp}_\mathcal{T}(x) \) (encoded as a binary sequence), is equal to 1. On the other hand, the set \( \{ (x, u \mid \text{comp}_\mathcal{T}(v) \mid x \in \Sigma^* \text{, } u, v \in \Sigma^* \} \) is recognized by a \( \log f(n) \) bounded deterministic Turing machine \( \mathcal{M} \) with two input tapes (only the position in the current storage-configuration of \( \mathcal{T} \) must be stored).
Take now this machine $M$, put it on the random tape and let it search for $\hat{1}$ \text{comp}_T(x) \hat{2}$. This string will appear on the random tape with probability 1. Thus $M$ stops with probability 1 and gives the correct result (according to the halting configuration in \text{comp}_T(x)). The expected time for the simulation lies in
\[
\bigcup_k (2^k |\text{comp}_T(x)|) \leq \bigcup_k (2^j (|x|) \cdot 2^k f(|x|)) \leq \bigcup_k (2^{2^k \log f(|x|)}) .
\]

Theorem 1 is valid also for transducers; in this case $M$ begins outputing after it has found and verified \text{comp}_T(x).

**Theorem 2.** For every function $f$,
\[
\Delta_2 \text{SPACE}(f(n)) \subseteq \bigcup_k \text{SPACE}(n^4 \cdot 2^k f(n)) .
\]

**Corollary.** If $f(n) \geq \log n$, then
\[
\Delta_2 \text{SPACE}(f(n)) = \bigcup_k \text{DSPACE}(2^k f(n)) .
\]

In particular,
\[
\Delta_2 \text{SPACE}(\log n) = \text{PSPACE}.
\]

**Proof of Theorem 2.** Let $M$ be an $f(n)$ bounded $\Delta_2$ machine. A configuration of $M$ contains the position on the input and the content of the work tape (but not the position on the random tape). The number of configurations accessible on input $x$ is bounded by $|x| \cdot 2^{|x|} |x|$. $M$ is simulated by a $\Delta_1$-machine $T$ (i.e. with one-way random-tape) in the same way as a two-way finite automaton is simulated by a one-way FA (see [HU 79]). It holds a table which says for each pair of configurations: if $M$ is in configuration $c$ and goes left (on the random tape) then it can (or cannot) come back in configuration $c'$. In addition it is stored whether or not $M$ starting in configuration $c$ can go left and never come back (in this case it is stored whether $M$ accepts or rejects).

It is easy to see that $T$ uses $(|x| \cdot 2^k |x|)^2$ space for two such tables and that these tables are sufficient to determine whether $M$ stops, and if it stops, to determine the decision. Since $M$ never gives a wrong result, $T$ accepts the same sets as $M$. Since $\Delta_1 \text{SPACE}(f(n)) \subseteq \text{PrSPACE}(f(n)) \subseteq \text{DSPACE}(f(n)^2)$ [BCP 83] $T$ can be simulated by a deterministic machine in $O(|x|^4 \cdot 2^k |x|)$ space. \hfill \Box
We were not able to extend the upper bound of Theorem 2 to the case of probabilistic machines with non-zero error probability. It is even not known whether or not $\Pr_2 \text{SPACE}$ is Blum complexity measure [Bl 67].

4 Open Problem

Is there a recursive function $h$, such that for every $f$

$$\Pr_2 \text{SPACE}(f(n)) \subseteq \text{DSPACE}(h(f(n)))?$$

Is every set recognized by a probabilistic finite automaton with two-way random-tape recursive, i.e., $\Pr_2 \text{SPACE}(O(1)) \subseteq \text{DSPACE}(h(n))$ for some recursive $h$?

(By [KV 84] the set of computations can be recognized by probabilistic finite two-way automata with one-way random-type and bounded error probability).\(^1\)

References


\(^1\)Note in proof. Meanwhile the authors were able to solve this problem. The first function $h$ mentioned above is in fact recursive and $2^{O(n)}$ and the second is $n^2 \log^2 n$. Therefore $\Pr_2 \text{SPACE}$ is a Blum complexity measure.


