数理解析研究所講究録
第458巻 1982年 84-93

84

On Reliability and Efficiency of Probabilistic Algorithms

by

Osamu Watanabe

Dept. of Information Sciences

Tokyo Institute of Technology

1. Introduction

Probabilistic algorithms are algorithms that make use of random choices and may make mistakes with some error probability. Recently several examples of efficient probabilistic algorithms have been proposed [7, 9]. Time and space efficiency of these algorithms have been extensively studied using probabilistic Turing machines (PTMs) as their formal models [4, 8].

But it is also important to study their reliability. Even if some problem is solvable by a fast PTM, it is not so useful if it is unreliable. We are interested in the following question: Can we get desirable reliability without losing efficiency?

To find counter examples to this question is one of the time-precision tradeoff problems and it seems to be very difficult like ordinary tradeoff problems ([1]). In this paper we show this sort of time-precision tradeoff results on some kinds of Turing macheines (random accessible TMs and on-line

TMs). Moreover these examples are also shown to be examples of the difference between nondeterministic executions and probabilistic ones.

2. Turing machine models

In this paper we consider some special types of Turing machines such as random accessible Turing machines and on-line Turing machines. They are obtained from standard Turing machines by slight modifications. As a standard Turing machine we use a multitape Turing machine that consists of a finite control uint equiped with a read-only input tape, a write-only output tape, and a finite number of read-write work tapes ([2]) Here we use Turing machines only as acceptors, so the output of a Turing machine is always "accept" or "reject". We use the words "deterministic" and "nondeterministic" in the conventional way ([2]).

A random accessible Turing machine is a model of the computation which uses a random access input device such as a disk, and is defined as follows.

Definition 2.1

A random accessible Turing machine (RTM) is a Turing machine which has the special work tape called address counter which contains only an integer by binary representation.

The move of a RTM is almost the same as that of a usual Turing machine except that its input tape head moves to the ith cell on the input tape in one move where i is the number in the address counter. ■

On-line Turing machines have often been used to find good lower bounds on the computation time ([3, 5, 6]). They are defined as follows.

Definition 2.2

An on-line Turing machine (ONTM) is a Turing machine with following restrictions:

(1) at any time the input tape head can only move to the right or stay on the input tape, and

(2) it must halt at the time when the input tape head marches off an input sequence.

Remark

From the restriction of on-line execution, before an ONTM moves its input tape head to the right, it must decide whether or not it accepts the sequence that is to the left of the head. ■

A random accessible probabilistic Turing machine (RPTM) and a on-line probabilistic Turing machine (ONPTM) are obtained respectively from a RTM and ONTM as follows.

Definition 2.3

A RPTM (ONPTM) is a RTM (ONTM) which may have coin-tossing states. The computation of a RPTM (ONPTM) is deterministic except when the machine enters a coin-tossing state where an unbiased coin is tossed to decide the next movement from two possible ones. ■

Let M be a RPTM (ONPTM). Define M(x) to be the output of M with input x. Because of the probabilistic execution, M(x) is a random variable for each x. We define the language accepted (recognized) by M, the error probability, and the maximum run time of M as follows (the motivation of these definitions is discussed in [10] but is omitted here).

Definition 2.4

The language accepted by M is

$$L(M) = \{x \mid P(\, M(x) = \text{accepts}\,) > \tfrac{1}{2}\},$$

The error probability of M is

$$e_M(x) = \begin{cases} 1 - P(\, M(x) = \text{accept}\,) & \text{if } x \in L(M), \\ P(\, M(x) = \text{accept}\,) & \text{if } x \notin L(M), \text{ and} \end{cases}$$

The maximum run time of M is

$\Phi_M(x)$ = least m such that every possible computation of M on input x halts in m steps. ■

Because of the above definition, the error probability of a RPTM (ONPTM) is less than $\frac{1}{2}$, and we say the error probability of a RPTM (ONPTM) is bounded if there is a constant $0 < \varepsilon < \frac{1}{2}$, such that $e_M(x) < \varepsilon$ for every possible input x.

3. The time-precision tradeoff results

Here we describe the time-precision tradeoff results on RPTMs and ONPTMs formally. In this paper we omit their proofs, which can be found in [10].

First we show a language such that time-precision tradeoff actually occurs on its recognition by RPTMs.

Definition 3.1

The set $D \subset \{0, 1, 2, 3\}^*$ is defined as follows:

$D = \{ku \mid$ (1) k is the binary representation of an integer $k > 0$,

(2) $u \in \{2, 3\}^{2N}$, and

(3) $u[i] = u[N + i]$ for all i, $1 \leq i \leq N$,

where N is 2^k and $u[i]$ denotes the ith symbol of u $\}$. ■

The time-precision tradeoff on the recognition of the set D is described formally as follows.

Theorem 3.2

(a) There is a RPTM M such that

(1) $L(M) = D$,

(2) M is $O(\log n)$ time bounded, and

(3) the error probability is bounded by $\frac{1}{2} - \frac{1}{2n}$

(in this paper the base of logarithm is 2).

(b) If there is a RPTM M such that

(1) $L(M) = D$,

(2) M is $T(n)$ time bounded, and

(3) the error probability of M is bounded,

then there is a constant $c > 0$ such that

$T(n) > cn$ for infinitely many $n > 0$,

where n is the length of an input. ■

Next we show the time-precision tradeoff result on ONPTMs.

Let Σ denote the set $\{0, 1, 2\}$. Hennie showed that a subset A of Σ^* needs more than $O((\frac{n}{\log n})^2)$ steps to recognize by ONTMs ([5]). Here we show the time-precision tradeoff on the recognition of the set A by ONPTMs. So we define the set A again.

Definition 3.3

The set $A \subset \Sigma^*$ is defined as follows.

$A = \{u_1 2 u_2 2 \ldots u_N 2 u'_1 2 u'_2 2 \ldots u'_{N'} 2 \mid$

(1) there is $k > 0$ such that

for all i, $1 \leq i \leq N$, $u_i \in \{0, 1\}^k$ and

for all j, $1 \leq j \leq N'$, $u'_j \in \{0, 1\}^k$,

(2) $N = 2^k$ and $N' > 0$, and

(3) there is i, $1 \leq i \leq N$, such that $u'_{N'} = u_i\}$. ■

The time-precision tradeoff result on the recognition of the set A is stated as follows.

Theorem 3.4

(a) There is an ONPTM M such that

(1) $L(M) = A$,

(2) M is $O(n)$ time bounded, and

(3) the error probability is bounded by $\frac{1}{2} - \frac{\log n}{8n}$.

(b) If there is an ONPTM M such that

(1) $L(M) = A$,

(2) M is $T(n)$ time bounded, and

(3) the error probability of M is bounded,

then there is a constant $c > 0$ such that

$$T(n) > c\left(\frac{n}{\log n}\right)^2 \quad \text{for almost all } n,$$

where n is the length of an input. ■

4. Nondetermistic executions vs. probabilistic ones

In this section we show that time-precision tradeoff examples showed in the previous section are also examples of the essential difference between nondeterministic executions and probabilistic ones.

The basic idea used in the proof of the first part of Theorem 3.2, 3.4 was proposed by Gill in the proof of NP PP ([4]). Here we analyze this technique little more precisely.

Let M be a nondeterministic Turing machine (NDTM). We define the number of nondeterministic configurations of M as follows.

Definition 4.1

The number of nondeterministic configurations of M on input x is the function $c_M(x)$ defined by

$c_M(x)$ = the maximum number of nondeterministic configurations in the execution of M for every possible computation sequence on input x. ■

In general every language accepted by a T(n) time bounded NDTM can be accepted by some O(T(n)) time bounded PTM by Gill's technique, which is stated little more precisely in the following proposition.

Proposition 4.2

Let L be a set of finite strings. If there is a NDTM M such that

(1) $L(M) = L$,

(2) M is $T(n)$ time bounded, and

(3) there is a function η from $\mathbb{N}$ to $\mathbb{N}$ such that is computed by some $T(n)$ time bounded determinictic TM and for all $x \in L$, $c_M(x) < \eta(n)$, where n and $\mathbb{N}$ is the length of an input x and the set of all positive integers respectively,

then there is a PTM M' such that

(1) $L(M') = L$,

(2) M' is $O(T(n))$ time bounded, and

(3) for all $x \in \Sigma^*$, $e_M(x) < \frac{1}{2} - \frac{1}{4 \cdot 2^{\eta(n)}}$,

where n is the length of an input x. ■

This proposition shows that NDTM acceptors are simulated by PTMs in the same order of time, but it also shows that their error probabilities are large (close to). So it is not an essential comparison.

What we really want to know is whether the same thing can be said with the restriction of bounded error probability. Although we intuitively expect that there is some difference, it seems very difficult to prove it in general. But the time-precision tradeoff results showed in the previous section are also examples that nondeterministic executions are more powerful than probabilistic ones with restriction of bounded error probability. That is, we have, from Theorem 3.2 and 3.4,

the following theorems.

Theorem 4.3

(a) There is a random accessible NDTM M such that

(1) $L(M) = D^c$ (where D^c denotes the compliment of D),

(2) M is O(log n) time bounded.

(b) If there is a RPTM M such that

(1) $L(M) = D^c$,

(2) M is T(n) time bounded, and

(3) the error probability of M is bounded,

then there is a constant $c > 0$ such that

$T(n) > cn$ for infinitely many $n > 0$. ■

Theorem 4.4

(a) There is an on-line NDTM M such that

(1) $L(M) = A$,

(2) M is O(n) time bounded.

(b) If there is an ONPTM M such that

(1) $L(M) = A$,

(2) M is T(n) time bounded, and

(3) the error probability of M is bounded,

then there is a constant $c > 0$ such that

$T(n) > c(\frac{n}{\log n})^2$ for almost all n. ■

Acknowledgement

The author wishes great appreciation to Prof. Kojiro Kobayashi for his careful reading of the first draft of this paper.

References

[1] A. Adachi, T. Kasai, The time-space trade-off problem on relativized Turing machine, The Trans. of the IECE of Japan E-64 (6) (1981).

[2] A. V. Aho, J. E. Hopcroft, J. D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading (1974).

[3] H. Gallaire, Recognition time of context-free languages by on-line Turing machines, Inform. Control, 15 (1969) 288-295

[4] J. Gill, Computaional complexity of probabilistic Turing machines, SIAM J. Comput. 6 (4) (1977) 675-695.

[5] F. C. Hennie, On-line Turing machine computations, IEEE Trans. Elec. Comput. EC-15 (1) (1966) 35-44.

[6] T. Kasami, A note on computing time for recognition of languages, Inform. Control, 10 (1967) 209-214.

[7] M. O. Rabin, Probabilistic algorithms, Algorithms and Complexity: New Directions and Recent Results, J. F. Traub, ed., Academic Press, New York (1976) 21-39.

[8] J. Simon, On tape-bounded probabilistic Turing machine acceptors, Theoret. Comp. Sci. 16 (1981) 75-91.

[9] R. Solovay, V. Strassen, A fast Monte-Carlo test for primality, SIAM J. Comput. 6 (4) (1977) 84-85.

[10] O. Watanabe, On reliability and efficiency of probabilistic algorithms, MS. Thesis, Dept. of Information Sciences, Tokyo Institute of Technology, Tokyo (1982).