stichting
mathematisch
centrum

MC

AFDELING INFORMATICA
(DEPARTMENT OF COMPUTER SCIENCE)

IW 245/83 DECEMBER

P.M.B. VITÁNYI

AN Ω1.618 LOWER BOUND ON THE TIME TO SIMULATE ONE QUEUE OR TWO PUSHDOWN STORES BY ONE TAPE

Preprint

kruislaan 413 1098 SJ amsterdam

Printed at the Mathematical Centre, Kruislaan 413, Amsterdam, The Netherlands.

The Mathematical Centre, founded 11 February 1946, is a non-profit institution for the promotion of pure and applied mathematics and computer science. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

1980 Mathematics subject classification: 68C40, 68C25, 68C05, 94B60, 10-00
1982 CR. Categories: F.1.1, F.1.3, F.2.3

AN $N^{1.618}$ LOWER BOUND ON THE TIME TO SIMULATE ONE QUEUE OR TWO PUSHDOWN STORES BY ONE TAPE*

by

Paul M.B. Vitányi

ABSTRACT

To simulate two pushdown stores, or one queue, on-line by a one-head tape unit requires $\Omega(n^{1.618})$ time. (This improves the known $\Omega(n \log^{1/2} n)$ lower bound for the on-line simulation of two-tape Turing machines by one-tape Turing machines.)

Keywords & Phrases: multitape Turing machines, pushdown stores, queues, time complexity, lower bounds, on-line simulation by single-head tape units, Kolmogorov complexity

---

* This work will be published elsewhere.

1. Introduction

It is generally the case, that additional access pointers in storage enhance computing power. In real-time, $(k+1)$-tape Turing machines are more powerful than k-tape Turing machines [Aa]. Analogous results hold with all heads placed on the same tape [PSS, Vi1]. Recently it was shown that k-tape Turing machines require nonlinear time to simulate $(k+1)$-tape Turing machines on-line [Pa2]. More in particular, that the on-line simulation of two pushdown stores by one tape requires $\Omega(n \log^{1/2} n)$ time*. In [Va] it was shown that the simulation of a queue by a real-time one-tape Turing machine is impossible. In [Vi2] it appeared that to simulate a pushdown store by an *oblivious* one-head tape unit on-line requires $\Omega(n\sqrt{n})$ time. Here we combine some ideas in [Vi2] with Kolmogorov complexity [Ko, PSS] arguments and an adversary demon to demonstrate an $\Omega(n^{1.618})$ lower bound on the simulation time for one queue, or two pushdown stores, by a (nonoblivious) one-tape Turing machine. This considerably narrows the gap with the best known upper bound, viz., each multitape machine can be simulated on-line by a one-head tape unit in square time [HU]. Square time is also the best known upper bound for the simulation of one queue or two pushdown stores by a one-head tape unit.

Turing machines and simulation. We regard machines as *transducers*, that is, as abstract storage devices connected with input- and output terminals. Thus we consider the machine as hidden in a black box, and the presented simulation results concern the input/output behaviour of black boxes and are independent of input/output conventions or whether we want to recognize or to compute. By a k-tape Turing machine we mean an abstract storage device, consisting of a finite control connected with k single-head linear storage tapes, and an input- and an output terminal. A one-tape Turing machine is the same as a one-head tape unit. The transducers effect a transduction from input strings to output strings by producing the i-th output just before reading the $(i+1)$-th input command. A machine A *simulates* a machine B *on-line* in time $T(n)$ if, for all $n>0$, the input/output behaviour of B, during the first n steps, is exactly mimicked by A within the first $T(n)$ steps. That is, for each input sequence $i_1, i_2, \ldots, i_k, \cdots$, read from the input terminal, the output sequences written to the output terminal are the same for A and B, and if $t_1 \leqslant t_2 \leqslant \cdots \leqslant t_k \leqslant \cdots$ are the steps at which B reads or writes a symbol, from or to the terminals, then there are corresponding steps $t_1' \leqslant t_2' \leqslant \cdots \leqslant t_k' \leqslant \cdots$ at which A reads or writes the same symbols and $t_i' \leqslant T(t_i)$, for all $i \geqslant 1$. In the sequel we write *simulation* for *on-line simulation*. (Simulation in time $T(n) = n$ is called *real-time* simulation; simulation in time $T(n) \in O(n)$ is called *linear time* simulation.)

2. Kolmogorov Complexity

The ideas on descriptional complexity below were developed independently by Kolmogorov [Ko] and Chaitin [Ch]. We closely follow the discussion in [PSS]. Consider the problem of describing a vector $\bar{x}$ of strings x_i over 0's and 1's. The string entries of the vector can be separated by ¢'s so that the vector is a string too. That is, $\bar{x} \in \{0, 1, ¢\}^*$. Any computable function f from vectors of strings over 0's and 1's to such vectors, together with a vector $\bar{y}$, such that $f(\bar{y}) = \bar{x}$, is such a description. A descriptional complexity K_f of $\bar{x}$, relative to f and $\bar{y}$, is defined by

$$K_f(\bar{x} \mid \bar{y}) = \min\{ \mid d \mid \mid d \in \{0,1\}^* \ \& \ f(d¢\bar{y}) = \bar{x} \} \ .$$

For the *universal* computable partial function f_0 we have that, for all vectors $\bar{x}, \bar{y}$, $K_{f_0}(\bar{x} \mid \bar{y}) \leqslant K_f(\bar{x} \mid \bar{y}) + c_f$, for all f with appropriate constant c_f. So the canonical relative descriptional complexity $K(\bar{x}, \bar{y})$ can be set equal to $K_{f_0}(\bar{x} \mid \bar{y})$. Define the *descriptional complexity* of

---

* We use the mnemonic "order of magnitude" notations as follows:
$f(n) \in O(g(n))$ if there are positive constants c and n_0 such that $\mid f(n) \mid \leqslant c\, g(n)$ for all $n \geqslant n_0$;
$f(n) \in \Omega(g(n))$ if there are positive constants c and n_0 such that $f(n) \geqslant c\, g(n)$ for all $n \geqslant n_0$.

$\bar{x}$ as $K(\bar{x}) = K(\bar{x} \mid \epsilon)$. ($\epsilon$ denotes the empty string.) Since there are 2^n binary strings of length n, but only $2^n - 1$ possible shorter descriptions d, it follows that $K(x) \geq |x|$ for some binary string x of each length. Following [PSS] further, we call such strings *incompressible*. It also follows that $K(x \mid y) \geq |x|$ for some binary string x of each length. Since similarly $K(x) \geq (1-\delta)|x|$ for $2^{\delta|x|}$ strings over $\{0,1\}$, which thus cannot be compressed to less than $(1-\delta)|x|$ bits, such "nearly" incompressible strings are abundant. Note that, for example, a string $x = uvw$ can be specified by v, $|x|$, $|u|$ and the bits of uw. Thus,

$$K(x) \leq K(v) + O(\log |x|) + |uw| \quad ,$$

so that with $K(x) \geq (1-\delta)|x|$ we obtain

$$K(v) \geq |v| - \delta|x| - O(\log |x|) \ .$$

3. Improved lower bound on the time to simulate multitape Turing machines by one-head tape units

Without loss of generality, we assume that all tape units below have semi-infinite tapes. That is, the squares of the tapes can be enumerated from left to right by the natural numbers. The 0th square is called the *start* square. Assume further, also without loss of generality, that the tape units write only 0's and 1's in the storage squares which introduces an extra constant delay. The Theorem below improves the known lower bounds. In the proof we make extensive use of *crossing sequences*. For a one-head tape unit we assume that, when it makes a move, it first overprints the symbol scanned and changes state, then moves the head. Thus, for any pair of adjacent tape squares, we can list the sequence of states in which the unit *crosses* from one to another. The first crossing must always be from left to right; after that crossings alternate in direction. The sequence of states so related to an intersquare boundary, or square, is called a crossing sequence. Early use of the concept is found in [Ra].

Theorem 1. *A one-head tape unit for simulating one queue requires* $\Omega(n^{1.618})$ *time. (Exact:* $\Omega(n^{(1+\sqrt{5})/2} / \log n)$. *The proportion* $(1+\sqrt{5})/2$, *or rather its reciprocal, is the "Golden Ratio".)*

Proof. Let M be a one-head tape unit simulating a virtual queue Q in time $T(n)$. It is known that $T(n) \in O(n^2)$. Without loss of generality, M has a semi-infinite tape and writes only 0's and 1's. In order to show that M loses a lot of time to either reach or transport earlier stored data we concentrate on a subproblem. Consider a sufficiently long incompressible string $x \in \{0,1\}^*$ of length n. We use M to obtain impossibly compressed descriptions. To store a substring of x of length at least m, M needs to use $m - O(\log n) - c_M$ work tape squares, with c_M a fixed constant depending only on M, by the incompressibility of x. Divide the tape in four segments $[0, n/6)$, $[n/6, 2n/6)$, $[2n/6, 3n/6)$ and $[3n/6, \infty)$. An *adversary demon* supplies the sequence of input commands.

- If M's head is on $[0, 2n/6)$ when the next input command is read then the next unread bit of $|x|$ is read and appended to the queue.
- If M's head is on $[2n/6, \infty)$ when the next input command is read, and the queue is nonempty, then the current head of the queue is unstored; if the queue is empty then the next unread bit of $|x|$ is read and appended to the queue.
- Having used all of x, the demon unstores the entire queue.

Let $T(n) \leq cn^{\alpha}$ for large enough n. Initially, M stores a prefix u of x with $n/4 \geq |u| \geq (n/6c)^{1/\alpha}$ while on $[0, n/6)$. Note that $K(u) \geq |u| - O(\log n)$.

It is immediately clear that given x the demon must have used all of x within a $2n$-length sequence of input commands.

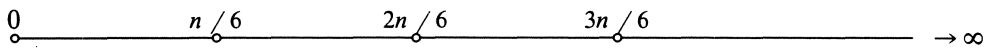

Figure. Division of M's tape in consecutive segments.

Case 1. The initially stored prefix u is retrieved one symbol at a time, before the demon has used all of x, by commands read while M's head is on $[2n/6, \infty)$.

In our description of $x = uv$ we give the u in terms of M's operation and v literally as suffix.

- A description of this discussion in O(1) bits.
- A description of M in O(1) bits.
- The value of n in $O(\log n)$ bits.
- The location of *any* square of $[n/6, 2n/6)$ in $O(\log n)$ bits.
- The crossing sequence at that square.
- v in $n - |u|$ literal bits.

For *any* square in $[n/6, 2n/6)$, the crossing sequence associated with that square contains, for each crossing, the time of crossing in $O(\log n)$ bits and the state of M in O(1) bits. To recover u, exhaustively test *every* bit string of length $|u|$ for consistency with the selected crossing sequence. If some other candidate $u' \neq u$ were consistent with it, then M would erroneously retrieve the same word in the first $|u|$ unstore commands for both $x = uv$ and $x' = u'v$ under the strategy of the adversary demon. Let the minimal number of crossings in a crossing sequence on $[n/6, 2n/6)$ be m. Then the description of x takes

$$O(1) + O(\log n) + O(m \log n) + n - |u|$$

bits. Consequently,

$$|u| \geq (n/6c)^{1/\alpha} \quad \& \quad |u| \in O(m \log n) \ .$$

Thus, $m \in \Omega(n^{1/\alpha}/\log n)$ and summing the crossing sequences of $[n/6, 2n/6)$ yields

$$T(2n) \in \Omega\left(\frac{n^{1+1/\alpha}}{\log n}\right) \ .$$

Since by assumption $T(n) \in O(n^{\alpha})$ we obtain $\alpha \geq (1+\sqrt{5})/2$ and therefore

$$T(n) \in \Omega(n^{1.618}) \ . \tag{1}$$

Case 2. Suppose that not all of u has been retrieved before the demon has used all of x. This implies that M has read less than $|u|$ input commands while its head was on $[2n/6, \infty)$. Therefore, M must have read $|x|$ input commands while its head was on $[0, 2n/6)$. In our description of $x = uv$ below we give u literally as a prefix and v in terms of M's operation.

- u in $|u|$ literal bits.
- A description of this discussion in O(1) bits.
- A description of M in O(1) bits.
- The value of n in $O(\log n)$ bits.

- The location of *any* square of $[2n/6, 3n/6)$ in $O(\log n)$ bits. Let this be square p.
- The *final* tape contents of the segment $[0,p)$ in at most $n/2$ bits.
- The crossing sequence at square p.

For *any* square of $[2n/6, 3n/6)$, the crossing sequence associated with that square consists of the times of crossing together with the states of M. To recover v, exhaustively test uv' for *every* bit string v' of length $|v|$, for consistency with the crossing sequence and final tape contents of $[0,p)$. If some other word $v' \neq v$ were consistent with those, then M would erroneously retrieve the same word in the first $|x|$ unstore commands for both $x = uv$ and $x = uv'$ under the strategy of the adversary demon. Let the shortest crossing sequence on $[2n/6, 3n/6)$ consist of m crossings. Then the description of x takes at most

$$|u| + O(1) + O(\log n) + n/2 + O(m \log n)$$

bits. Since x is incompressible, it follows that $m \in \Omega(n/\log n)$, and summing the crossing sequences of $[2n/6, 3n/6]$ yields

$$T(2n) \in \Omega(\frac{n^2}{\log n}) \ , \tag{2}$$

which, together with inequality (1) in Case 1, proves the Theorem. □

Since four one-head tape units can simulate a queue in real-time [LS] the lower bound on the time to simulate many storage tape units by one is set to $\Omega(n^{1.618})$. This lower bound holds also for the simulation of two pushdown stores.

Theorem 2. *A one-head tape unit for simulating two pushdown stores requires* $\Omega(n^{1.618})$ *time.*

Proof. The proof is essentially the same. Let M be a one-head tape unit simulating two virtual pushdown stores P_1 and P_2 in time $T(n)$. Without loss of generality M has a semi-infinite tape and writes only 0's and 1's. Consider a sufficiently long incompressible string $x \in \{0,1\}^*$ of length n. Divide the tape of M in four segments as previously, see Figure. An adversary demon supplies the sequence of input commands.

- Initially, while M's head is on $[0, 2n/6)$ for each next input command the next unread bit of x is pushed on P_1. M's head must range outside $[0, 2n/6)$ by the incompressibility of x. Ever after, when the next input command is read while M's head is on $[0, 2n/6)$ the next unread bit of x is pushed on P_2.
- If M's head is on $[2n/6, \infty)$ when the next input command is read then the top of P_1 is popped. If P_1 is empty then push instead the next unread bit of x on P_2.
- Having used all of x, the demon pops all of P_1 and P_2.

It is immediately clear that given x the demon must have used all of x within a $2n$-length sequence of input commands. Choose $T(n)$ and u as in the previous proof. The same reasoning in Cases 1 and 2 is *mutatis mutandis* applicable. □

Corollary. The result implies trivially that an *oblivious* one-head tape unit for simulating one pushdown store requires $\Omega(n^{1.618})$ time.

REFERENCES

Aa Aanderaa, S.O., On k-tape versus $(k+1)$-tape real-time computation, SIAM-AMS Proceedings, Vol. 7 (*Complexity of Computation*), 1974, 75-96.

Ch Chaitin, G.J., Algorithmic Information Theory, *IBM J. Res. Dev.,* **21** (1977) 350 - 359.

HU Hopcroft, J.E. & J.D. Ullman, *Formal languages and their relations to automata*, Addison-Wesley, 1969.

Ko Kolmogorov, A.N., Three approaches to the quantitative definition of information, *Problems in Information Transmission, Vol. 1, No. 1,* (1965) 1 - 7.

LS Leong, B.L., and J.I. Seiferas, New real-time simulations of multihead tape units, *J. Ass. Comp. Mach.* **28** (1981) 166 - 181.

Pa1 Paul, W.J., On heads versus tapes, 22nd IEEE Symp. on Foundations of Computer Science, 1981, 68 - 73.

Pa2 Paul, W.J., On-line simulation of $k+1$ tapes by k tapes requires nonlinear time, 23rd IEEE Symp. on Foundations of Computer Science, 1982, 53-56.

PSS Paul, W.J., J. Seiferas & J. Simon, An information-theoretic approach to time bounds for on-line computation, 12th ACM Symp. on Theory of Computing, 1980, 357-367.

Ra Rabin, M.O., Real-time computation, *Israel J. Math.* **1**, (1963) 203-211.

Va Valiev, M.K., Certain estimates of the time of computations on Turing machines with an input, *Cybernetica* **6** (1972) 734 - 741. Translated from: *Kibernetica* **6** (1970) 309 - 317.

Vi1 Vitányi, P.M.B., On the power of real-time Turing machines under varying specifications, 7th Int. Coll. on Automata, Languages and Programming, *Lecture Notes in Computer Science* **85** , Springer Verlag, Berlin, 1980, 658-671.

Vi2 Vitányi, P.M.B., On the simulation of many storage heads by one, 10th Int. Coll. on Automata, Languages and Programming, *Lecture Notes in Computer Science* **154** , Springer Verlag, Berlin, 1983, 687 - 694.