Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

P.M.B. Vitányi

One queue or two pushdown stores take square time on a one-head tape unit

Department of Computer Science Report CS-R8406 March

ONE QUEUE OR TWO PUSHDOWN STORES TAKE SQUARE TIME ON A ONE-HEAD TAPE UNIT

PAUL M.B. VITANYI
Centre for Mathematics and Computer Science, Amsterdam

To simulate one virtual queue or two virtual pushdown stores by a one-head tape unit takes at least square time. Since each multitape Turing machine can be trivially simulated by a one-head tape unit in square time this result is optimal.

69F11, 69F13, 69F23
1980 MATHEMATICS SUBJECT CLASSIFICATION: 68C40, 68C25, 68C05, 94B60, 10-00
1982 CR CATEGORIES: F.1.1, F.1.3, F.2.3
KEYWORDS & PHRASES: multitape Turing machines, pushdown stores, queues, time complexity, square lower bounds, on-line simulation by single-head tape units, Kolmogorov complexity
NOTE: This report will be submitted for publication elsewhere.

Report CS-R8406
Centre for Mathematics and Computer Science,
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

1. Introduction

Each multitape Turing machine can be simulated by a one-head tape unit in square time [HU]. We can derive a lower bound which matches this upper bound on the simulation time. So in this case the obvious simulation is also optimal. In particular, it takes $\Omega(n^2)$ time to simulate one queue or two pushdown stores by a one-head tape unit. The concept of simulation used is that of the implementation of one virtual queue or two virtual pushdown stores on a one-head tape unit. A previous lower bound on this simulation time was $n^{1.618}$ in [Vi2]. Recall, that in an *oblivious* Turing machine the movement of the storage tape heads is independent of the input, and is a function of time alone, see for instance [Vi4]. For simulation by oblivious one-head tape units we had closed the gap between the lower bound and upper bound already in [Vi3] by exhibiting a square lower bound on the time to simulate one pushdown store. A one-head tape unit is used as a synonym for a Turing machine with one single-head storage tape.

The proof uses Kolmogorov Complexity as in [PSS, Vi2, Vi3, Ma, Li]. It also uses an adversary demon as did [Vi2, Vi3]. The basic idea used descends from [Vi1] and is a straightforward extension of [Vi3]. In the meantime Joel Seiferas has drawn my attention to the almost simultaneous independent results of Wolfgang Maass and Ming Li in this area. In [Ma] it is proved that to simulate two single-head tapes by one single-head tape takes $\Omega(n^2 / \log n)$ time. The stronger result due to [Li] shows the optimal square lower bound on the time for single one-head tape simulation of two stacks, which is derived here by different methods.

2. Kolmogorov Complexity

The ideas on descriptional complexity below were developed independently by Kolmogorov [Ko] and Chaitin [Ch]. We follow [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 f with appropriate constant c_f, for all vectors $\bar{x},\bar{y}$, $K_{f_0}(\bar{x} \mid \bar{y}) \leq K_f(\bar{x} \mid \bar{y}) + c_f$. 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 $\bar{x}$ 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 $K(x) \geq \mid x \mid$ for some binary string x of each length. We call such strings *incompressible*. It also follows that $K(x \mid y) \geq \mid x \mid$ for some binary string x of each length. Since similarly $K(x) \geq (1-\delta) \mid x \mid$ for $2^{\delta \mid x \mid}$ strings over $\{0,1\}$, which thus cannot be compressed to less than $(1-\delta) \mid x \mid$ bits, such "nearly" incompressible strings are abundant. Note that a string $x = uvw$ can be specified by v, $\mid x \mid$, $\mid u \mid$ and the bits of uw. Thus,

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

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

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

3. The square lower bound

Without loss of generality, we assume that the 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 and relax the *real-time* requirement to *constant delay*. A computation is of *constant delay* if there is a fixed constant c such that there are at most c computation steps in between processing the nth and the $(n+1)$th inputsymbol, for all n. Thus, constant delay with $c=1$ is the same as real-time, and it is not difficult to see that each computation of constant delay can be sped up to a real-time computation by expanding the storage alphabet and the size of the finite control.

Theorem. *The fastest simulation of two pushdown stores by a one-head tape unit takes* $\Theta(n^2)$ *time.*

Proof. The only thing we have to prove is the square lower bound; the square upper bound is trivial. Consider two pushdown stores P_1 and P_2. Assume, by way of contradiction, that M is a one-head tape unit simulating the virtual pushdown stores P_1 and P_2 in time $T(n) \in o(n^2)$. Without loss of generality M has a semi-infinite tape and writes only 0's and 1's. An adversary demon supplies the sequence of input commands. The adversary demon first determines an $n/8$-length initial segment $[0, n/8)$ of the tape for a given n. It subsequently determines the sequence of n polled input commands as follows. Let $z = xy$ be an incompressible word of length $2n$ with a prefix x of length n.

- In each input command the demon pushes the next unread bit of x on P_1.
- If M scans a square of the initial segment $[0, n/8)$, when polling an input command, then the demon pushes the next unread bit of y on P_2. The demon maintains a function $g(t)$ which is defined as the total number of input commands polled, up to time t, while M scanned a square of the initial segment $[0, n/8)$ at poll time.
- Let M's head be positioned on the final segment $[n/8, \infty)$ in the step polling an input command. Then (1) the demon neither pushes nor pops P_2 if the head is on the segment $[n/8, g(t))$ in that step, or (2) it pops P_2 if the head is on the segment $[g(t), \infty)$ in that step.

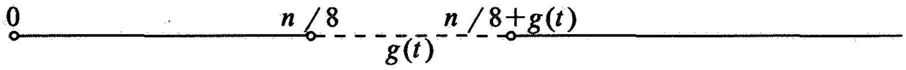

Figure 1.

Case 1. Suppose that at least n/c input command's are polled while the head scans a particular $n/2c$-length tapesegment $T = [a, a + n/2c)$, during the first n input commands issued by the demon. In the description of $z = xy$ below we give part of x, in concatenated literal form, in a suffix v and the concatenated remainder w of x in terms of M's operation. (Therefore x is a *shuffle* of v and w.) We concatenate y in literal form to this description of x. So the description of z is as follows.

- A description of this discussion in O(1) bits.
- A description of M in O(1) bits.

- The value of n and a in $O(\log n)$ bits.
- The location of *any* pair of squares (p,q) with p in $[a-n/8c,a)$ and q in $[a+n/2c,a+5n/8c)$. This takes $O(\log n)$ bits.
- The crossing sequence at that pair (p,q) of squares, as described below.
- The *final* contents of the tape segment $[p,q]$, after $2n$ input command's of M have been polled, processing all of x.
- The concatenated literal remainder v of x in not more than $n-n/c$ bits.
- The literal representation of y in n bits.

For *any* pair (p,q) of such squares, the crossing sequence associated with that pair contains for each crossing the state of M and whether we enter/leave $[p,q]$ from/to left or right in $O(1)$ bits. Associated with each *entrance* of $[p,q]$ we give the number of times the input is polled up to the corresponding *leave* of $[p,q]$; summed over all of the crossing sequence this does not take more than $O(l\log(n/l))$ bits, where l is the length of the crossing sequence. To recover x, start reading the literal representation v until the head of M enters $[p,q]$. Try all continuations which lead in the correct number of inputs to the correct exit of $[p,q]$ and continue with the literal representation v, and so on. Finally, after having processed n bits, which includes all of v, and matching the final contents of $[p,q]$, the resulting machine i.d. must store x, which can be retrieved by n pop P_1 commands. Let the minimal length of any crossing sequence for a pair (p,q) be $l(n)$. Then the description of $z=xy$ takes not more than:

$$O(1)+O(\log n)+O(l(n)\log(n/l(n)))+6n/8c+n-n/c+n$$

bits. Since this amount must be at least $K(z)\geqslant 2n$, it follows that $l(n)\log(n/l(n))\in\Omega(n)$ and therefore $l(n)\in\Omega(n)$. Summing the lengths of the crossing sequences of a set S of all pairs (p,q), such that if $(p_1,q_1),(p_2,q_2)\in S$ then $p_1\neq p_2$ and $q_1\neq q_2$, must give a lower bound on the running time. Therefore $T(n)\geqslant(n/8c)l(n)$. Hence, for each fixed constant $c>0$ we obtain $T(n)\in\Omega(n^2)$: contradiction

Case 2. Suppose that less than n/c input commands are polled on *any* $n/2c$-length contiguous tapesegment during the first n input commands provided by the demon. Thus manageing to avoid a square time consumption by the positioning of input polls with respect to P_1, the adversary input strategy with respect to P_2 will now force a square time consumption anyway.

Since there can all in all be at most $n/4$ inputs polled on the initial tapesegment $[0,n/8)$ by assumption, the value of $g(T(n))$ is not greater than $n/4$. So subsequent to n polled input commands the "skip" tapesegment $F=[n/8,n/8+g(T(n)))$ is contained in $[n/8,3n/8)$. By Case 1 the segment F can harbor not more than $n/2$ input command polls. Consequently, the amount of pops polled on $[3n/8,\infty)$ is at least $n/4$ out of n.

Because $T(n)\in o(n^2)$ by assumption, virtual pushdown store P_2 contains $\Omega(n^{1/2})$ bits by the time the head initially enters F (which it must by Case 1). Since there are at least as many pops of P_2 ($\geqslant n/4$) as pushes of P_2 ($\leqslant n/4$) in the first n inputcommands, at some time $t_1\leqslant T(n)$ the store of P_2 is emptied for the first time. Let m be the number of input commands polled by time t_1. By the demon's strategy and Case 1 the number of skips is at no time more than twice the number of pushes. Therefore the number of pops in the first m commands is at least $m/4$ and we know that $m/4\in\Omega(n^{1/2})\cap O(n)$. Let p be any square in $[n/8,(n+m)/8)$. See Figure 2.

We give a description of $z=xy$ in terms of M's operation. In the description of y below we give part of y literally in a suffix v and part w of y in terms of M's operation. Now y is a *shuffle* of v and w. We give x literally as a suffix. The description of z is as follows.

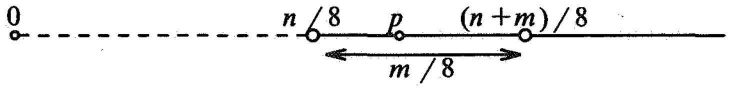

Figure 2.

- A description of this discussion in O(1) bits.
- A description of M in O(1) bits.
- The value of n and m in $O(\log n)$ bits.
- The location of *any* square p on $[n/8, (n+m)/8)$ in $O(\log n)$ bits.
- The crossing sequence at p.
- The concatenated bits of y popped on receiving pop P_2 commands *left* of p in not more than $m/8$ bits, by definition of g and m.
- The literally represented (possibly noncontiguous) part v of y in not more than $n - m/4$ bits.
- The literal representation of x in n bits.

The crossing sequence associated with p contains for each crossing the state of M and with each *entrance* of $[0,p)$ the number of times the input is polled up to the corresponding *leave* of $[0,p)$. Similar to Case 1, if l_p is the length of the crossing sequence at p then the crossing sequence can be denoted in not more than $O(l_p \log(m/l_p))$ bits. Let the minimum length of such a crossing sequence on $[n/8, (n+m)/8)$ at time t_1 be $l(m)$. Then the description of y takes not more than

$$O(1) + O(\log n) + O(l(m)\log(m/l(m))) + m/8 + n - m/4 + n$$

bits. To recover y, try all binary strings to see which n-length string matches:

- The crossing sequence at p.
- The literal sequence of bits popped by pops initiated left of p. Note that it is easy to determine at any time t what is the skip tape segment and what is the pop tape segment.
- The literal concatenated bits of y popped by pops initiated *right* of p. In case the bits are actually output *left* of p we add to the crossing sequence at p the bits of y for which the pop command was polled *right* of p and which were factually output on the segment $[0,p)$. This adds at most one bit to the description of each crossing, so the embellished description of the crossing sequence at p stays $O(l(p)\log(m/l(p)))$.
- The literally given suffix v of the description of y.
- The literal representation of x.

Since $K(xy) = 2n$ we have $l(m)\log(m/l(m)) \in \Omega(m)$. Therefore $l(m) \in \Omega(m)$. Minorizing the running time $T(m)$ by summing the lengths of the crossing sequences over all squares of $[n/8, (n+m)/8)$ to at least $l(m)m/8$ we obtain $T(m) \in \Omega(m^2)$. □

Corollary. To simulate a queue by a one-head tape unit requires $\Theta(n^2)$ time.

Proofsketch. Choose an incompressible string x of length n. Store the consecutive bits of x, one bit at a time, on polls at time t occurring on the initial tape segment $[0, (g(t)+n)/8)$ ($g(t)$ as above). At polls occurring at time t on the final tape segment $[(g(t)+n)/8, \infty)$ unstore the queue by one bit each such poll. Since by Case 1 above $g(t) \leq n/4$, there are in the n polled inputcommmands at least $n - 2(n+n/4)/8 = 11n/16$ unstores. Although the former *skip* segment now harbors *stores* it has shrunk so much to $[n/8, 5n/32)$ that the $11n/16$ *unstores* on the right adjoining final segment $[5n/32, \infty)$ must eventually unstore all bits stored in the left adjoining initial tape segment $[0, n/8)$. Furthermore, at all times the number of bits polled while on the middle segment is never more than $1/4$th of the number of bits polled while on the initial segment by Case 1 above. Consequently the same arguments as above hold *mutatis mutandis.* □

REFERENCES

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

HU Hopcroft, J.E., and 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.

Li Li, M(ing), On one tape versus two stacks. Manuscript, Computer Science Department, Cornell University, February 1984.

Ma Maass, W., Quadratic lower bounds for deterministic and nondeterministic one-tape Turing machines, Extended Abstract, Department of Mathematics and Computer Science Division, University of California, Berkeley, December 1983.

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

Vi1 Vitányi, On the simulation of many storage heads by one. Technical Report IW 228, Mathematisch Centrum, Amsterdam, June 1983. Extended Abstract in: Proceedings International Colloquium on Automata, Languages and Programming, *Lecture Notes in Computer Science* **154**, Springer Verlag, Berlin, 1983, 687 - 694.

Vi2 Vitányi, P.M.B., An $N^{1.618}$ lower bound on the time to simulate one queue or two pushdown stores by one tape, Technical Report IW 245, Mathematisch Centrum, Amsterdam, December 1983.

Vi3 Vitányi, P.M.B., Square time is optimal for the simulation of a pushdown store by an oblivious one-head tape unit, Technical Report CS-R8402, Centre for Mathematics and Computer Science (C.W.I.), Amsterdam, January 1984.

Vi4 Vitányi, P.M.B., An optimal simulation of counter machines, *SIAM J. Comput.,* in press.