stichting
mathematisch
centrum

MC

AFDELING INFORMATICA
(DEPARTMENT OF COMPUTER SCIENCE)

IW 226/83 APRIL

P.M.B. VITÁNYI

ON THE SIMULATION OF MANY STORAGE HEADS BY A SINGLE ONE

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~~ XX
1982 CR. Categories: F.1.1, F.1.3, F.2.3, B.3.2, E.1, E.4
69F11, 69F13, 69F22, 69B31, 69E40

ON THE SIMULATION OF MANY STORAGE HEADS BY A SINGLE ONE*

(Extended Abstract)

Paul M.B. Vitányi

ABSTRACT

Each multitape Turing machine, of which the storage heads scan $O(\log n)$ distinct squares in each interval of n steps, for all $n \geq 1$, can be real-time simulated by an oblivious one-head tape unit. There exist multitape Turing machines, e.g. the normal pushdown store, for which the fastest on-line simulation by an oblivious one-head tape unit requires $\Omega(n \sqrt{n})$ time.

Keywords & Phrases: multitape Turing machines, on-line simulation by single head tape units, upper bounds, lower bounds, real-time simulation by oblivious one-head tape unit, uniform logarithmic space, multicounter machines, augmented counter machines, pushdown stores.

* This report is a preliminary version to be presented at the 10th International Colloquium on Automata, Languages and Programming, held in Barcelona, Spain, the 3rd week of July, 1983. A full version, containing complete proofs, will be available shortly.

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. Analogous results hold with all heads placed on the same tape [V1,PSS], head-to-head jumps added [PSS], and for multihead finite automata with and without head-to-head jumps [J,SV]. Recently it was shown that k-tape Turing machines require nonlinear time to on-line simulate $k+1$-tape Turing machines [P]. With respect to upper bounds there are essentially two facts known. Each multitape machine can be on-line simulated by a one-head tape unit in square time [HU], and also by a two-tape Turing machine in time* $O(n \log n)$ [HS]. Both of these simulations can be made oblivious [PF], retaining the same simulation time. In [PF] it was furthermore shown that each oblivious multitape Turing machine on-line simulating a single pushdown store requires $\Omega(n \log n)$ time. Thus, for on-line simulation of multitape Turing machines by one-head tape units the fastest simulation time is somewhere in between a nonlinear lower bound and a square upper bound, while for on-line simulation by oblivious one-head tape units the lower bound is $n \log n$ and the upper bound n^2. We improve this situation in two ways. First, we show that for a restricted class of multitape Turing machines, viz. machines of which the storage heads scan $O(\log n)$ distinct squares in each interval of n steps, for all $n \leqslant 1$, each member can be real-time simulated by an oblivious one-head tape unit belonging to that class. Second, it is demonstrated that each oblivious one-head tape unit, on-line simulating a single pushdown store, requires $\Omega(n \sqrt{n})$ time.

Turing machines, simulation and obliviousness. 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 *on-line simulates* a machine B 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. A Turing machine is *oblivious* if the movements of the storage tape heads are fixed functions of time, independent of the particular inputs to the machines, see e.g. [PF]. There are many reasons why one may want to restrict attention to oblivious computations. For instance, oblivious Turing machine computations translate efficiently to combinational logic networks, while ordinary Turing machine computations do not. We mention two less often cited motives, of a more heuristic nature, for focussing attention on oblivious computations, of which the second one is pure conjecture. Suppose we can simulate some abstract

* We use the mnemonic "order of magnitude" notations as follows:
$f(n) \in O(g(n))$ if there is a positive constant c such that $|f(n)| \leqslant c\, g(n)$ for all n.
$f(n) \in \Omega(g(n))$ if there is a positive constant c such that $f(n) \geqslant c\, g(n)$ for infinitely many n.
$f(n) \in \Theta(g(n))$ if $f(n) \in O(g(n)) \bigcap \Omega(g(n))$.
$f(n) \in o(g(n))$ if $f(n) \in O(g(n)) - \Omega(g(n))$.

storage device S in time $T(n)$ by an oblivious one-head tape unit M. Then we can also simulate k copies of S, say $S_1, S_2, \ldots, S_k$, interacting through a common finite control, by dividing M's tape into k tracks, modifying M's finite control, and letting the head on each track do the same job as it formerly did on the total tape. Thus, the resulting oblivious one-head tape unit M', with modified finite control and expanded tape alphabet, uses the same time and space as did M.

Lemma 1. *If we can simulate a pushdown store by an oblivious one-head tape unit in time $T(n)$ then we can simulate each multitape Turing machine by an oblivious one-head tape unit in time $T(n)$, using just the same space.*

Proof. Replace each tape of the multitape Turing machine by two pushdown stores and apply the preceding argument to multipushdown store machines. □ □

Conjecture. If we can simulate each multitape Turing machine by a one-head tape unit in time $T(n)$ then we can also simulate each multitape Turing machine by an oblivious one-head tape unit in time $T(n)$.

2. Uniform space and fast simulation of many heads by a single oblivious one

For on-line computations (viz. the transducer type of computations) it is, perhaps, unreasonable that the workspace accessed in any length input interval may be arbitrary large, for unbounded storage complexity, if the machine has been computing long enough previously. For example, if a real-time Turing machine M has storage complexity $\Theta(\log n)$ then in the 1-step interval, from the 2^n-th step through the 2^n+1-th step, the machine M can access $\Theta(n)$ storage squares. Thus, we propose a space complexity measure, independent from the origin of the time scale, and only depending on the size of the intervals of steps.

Definition. Let M be a multitape Turing machine, and let for any unbounded input sequence ω the interval of steps by M, executed in processing ω, from the $(m+1)$-th through $(m+i)$-th *step*, be denoted by $I_{m,i}^{\omega}$, $m \geqslant 0$ and $i \geqslant 1$. The *uniform space complexity* $U(n)$ of M is defined as

$$U(n) = \sup\{U_m^{\omega} \mid U_m^{\omega} \textit{ is the number of distinct squares, on all of M's storage tapes, visited by M's heads during } I_{m,n}^{\omega}, \textit{for } m \geqslant 0 \textit{ and } \omega \textit{ an unbounded input sequence } \}.$$

Thus, finite automata correspond to Turing machines with uniform space complexity $O(1)$. If the (ordinary) space complexity of some Turing machine is $S(n)$ and the time complexity $T(n)$, then its uniform space complexity $U(n) \in \Omega(S(T^{-1}(n))) \cap O(n)$. Recall that finite automata have the exceptional property that a storage facility consisting of a collection of k finite automata, interacting through a common finite control, can be replaced by a single finite automaton, without slowing down the computation, see e.g. [HU]. Below we show that additional tapes, or nonobliviousness, likewise do not increase the power of an (oblivious) one-head tape unit in uniform logarithmic space.

Lemma 2. *Each multitape Turing machine of uniform space complexity $O(\log n)$ can be real-time simulated by an oblivious one-head tape unit of also uniform space complexity $O(\log n)$.*

Proof Sketch. The proof uses a complicated tape manipulation technique developed in [V2], to real-time simulate multicounter machines by oblivious one-head tape units. Without going into details, an oblivious one-head tape unit M can be constructed such that, for each $i \geqslant 1$, the pair of squares, or rather square contents or *cells*, originally in positions i, $i+1$ is scanned at least once in each time interval of c^i steps for some small constant c, say c is about 3. Moreover, the head recognizes such pairs when they are scanned (knows they are cells i, $i+1$ for some $i \geqslant 1$) and has always cells 1, 2 under scan. The tape unit M works by, in each step, interchangeing cells residing on the currently simultaneously scanned tape squares. (M's fat head scans a few adjacent cells simultaneously.) In this process, the identity of the underlying squares is not important, the identity (index i above) of the cells, however, is fixed wherever they end up. The oblivious one-head tape unit M has uniform space complexity $\Theta(\log n)$. By Lemma 1 we only have to show that any pushdown store P of uniform space complexity $O(\log n)$ can be real time simulated by the described oblivious one-head tape unit. So, let P be a pushdown store which does not change its stack height by more than $O(\log i)$ elements in each interval of steps $I_{m,i}^{\omega}$, for all $m \geqslant 0$, $i \geqslant 2$ and any ω. In the simulating M each cell (square contents) can contain an ordered segment of P's stack consisting of 0, d, or $2d$ elements, and the first cell can contain an initial segment of P's stack of in between 0 and $2d$ elements. Each cell $i \geqslant 1$ strives for an occupancy of stack elements as follows. If it contains $2d$ elements when cells i, $i+1$ are scanned then the last d elements are shifted to cell $i+1$. If it contains 0 elements when cells i, $i+1$ are scanned, and cell $i+1$ contains d or $2d$ elements, then the first d elements are shifted from cell $i+1$ to cell i. Cell 1, being distinguished, shifts d elements out, if it contains $2d$ elements, and shifts d elements in, if it contains $d-1$ (or less) elements, to and from cell 2. According to the current input, elements are added/deleted from the segment in cell 1 in each step. Thus, a segment of d stack elements can be shifted from the 1-th cell to the i-th cell, or vice versa, in $\sum_{j=1}^{i-1} c^j < c^i$ $(c \geqslant 2)$ steps, and thus in c^i steps id elements can be pushed or popped. Starting with an empty stack, it can be proved that at all times $t \geqslant 0$, for any input,

(i) no cell contains more than $2d$ stack elements;

(ii) if any cell contains stack elements then cell 1 contains stack elements,

provided the stack height does not change more than id elements in I_{m,c^i}^{ω} , for all m, i, ω. Choosing d appropriately, which is possible since the stack height varies $O(\log i)$ elements in each interval $I_{m,i}^{\omega}$, for all m, i, ω, (i) and (ii) show that the arrangement can real-time implement a uniform $O(\log n)$ space pushdown store. □ □

The next question is which computations, or problems, are in uniform logarithmic space. [V2] shows that each multicounter computation is of this space complexity. Uniform log space is, however, more extensive. Recall that multicounter machines consist of a set of counters numbered say, $1,2, \ldots, k$, which can execute one-step arithmetic/boolean instructions as "add [subtract] 1 from counter i" and "test counter i for 0", $1 \leqslant i \leqslant k$. Several other one-step instructions can be synthesized, by using concealed auxiliary counters, such as tests for equality amongst counters (by maintaining all the differences on extra counters). Instructions for which it is known [FMR] that they cannot be so synthesized as one-step instructions are "set counter i to 0" or "set counter i to the value of counter j". Call multicounter machines with those one-step instructions added *augmented counter machines* [*ACMs*]. It can be proved [V3] that:

Lemma 3. *Each augmented counter machine can be real-time simulated by a uniform log space oblivious one-head tape unit.*

None the less, uniform log space computations are *not very* powerful if we impose time restrictions.

Lemma 4. *There are (ordinary) log space/real-time Turing machines such that the fastest uniform log space Turing machines simulating them use exponential time.*

Proof Sketch. Take as an example the on-line recognition of

$$L = \{\star w_1 \star w_2 \star \cdots \star w_{2^i} \star \cdots \mid w_j \textit{ is a palindrome in } \{0,1\}^*, j \geqslant 1, \ \& \\ \& \ \# w_{2^i} = \# w_{2^i+1} = \cdots = \# w_{2^{i+1}-1} = 2(i+1), i \geqslant 0 \}.$$

Recognition in real-time/ $\log n$ *space:* on one track of its single storage tape the recognizer maintains a binary count of the number of $\star$'s. During the update of the $\star$ count, it writes the first half of the current word $w_{current}$ on another track and, while proceeding back to the origin, compares it with the second half.

Recognition in uniform $\log n$ *space:* to on-line check whether w_{2^i} is a palindrome, the machine must access $\Omega(i)$ tape squares. By definition it takes $\Omega(2^i)$ steps to do so. □ □

Obviously, any real-time multitape Turing machine computation can be simulated on a uniform $\log n$ space Turing machine in exponential time. Thus, by Lemma's 1-4:

Theorem 1. *(i). In real-time, multicounter machines are less powerful than uniform* $\log n$ *space Turing machines.*
(ii). In real-time, uniform $\log n$ *space multitape Turing machines are equally powerful as uniform* $\log n$ *space oblivious one-head tape units.*
(iii). There are $\log n$ *space/real-time one-tape Turing machines which cannot be simulated by uniform* $\log n$ *space multitape Turing machines in less than exponential time.*

The question to consider next is: 'what is the most extensive uniform space complexity class for which nonobliviousness and extra storage heads do not increase the computation power of the device under the real-time restriction?'. We can give an upper bound on the uniform space complexity allowing linear time simulation, of multitape Turing machines, by oblivious one-head tape units.

Theorem 2. *Each real-time multitape Turing machine, which can be linear time simulated by an oblivious one-head tape unit, has uniform space complexity* $o(n / \log n)$.

Proof Sketch. Similar to the overlap argument in [PF] that an oblivious Turing machine, simulating a single pushdown store, needs $\Omega(n \log n)$ time. The argument leads in fact to the more general trade-off:

$$\sum_{i=1}^{\log n} 2^{-i} U(2^i) \in O(T(n)/n),$$

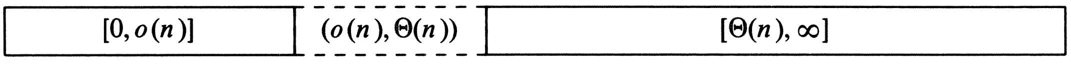
with $U(n)$ the uniform space complexity of the simulated real-time machine and $T(n)$ the simulation time of the oblivious simulator. □

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

In view of Lemma 1, any lower bound on the simulation time of multitape Turing machines by oblivious one-head tape units also holds for the simulation of a single pushdown store by the same (and obviously, vice versa). The following Theorem improves the known lower bounds. The proof makes extensive use of *crossing sequence* arguments, as is to be expected. It depends on a fairly precise selection and interdependence of the involved constants of proportionality. The proof sketch below gives the intuition behind the proof.

Theorem 3. *Any oblivious one-head tape unit simulating a (typical) pushdown store requires* $\Omega(n \sqrt{n})$ *time.*

Proof Sketch. Let M be the fastest oblivious one-head tape unit for simulating a typical pushdown store in, say, time $T(n)$. It is known that $T(n) \in \Omega(n \log n) \cap O(n^2)$. Let $I_{m,i}$ denote the interval of steps by machine M to process the $(m+1)$-th through $(m+i)$-th *input command.* (Do not confuse these intervals with the I-intervals of the previous section. The subscripts refer to the sequence of input commands, instead of the sequence of steps, and since M is oblivious a superscript referring to particular input sequences is unnecessary, and therefore suppressed. For simplicity we assume that in an oblivious machine the steps at which it reads or writes a symbol are the same for all input streams. However, it is not necessary to assume this subtility in order to derive the desired results.) Assume that M's tape is one-way infinite, and let $[0,o(n)]$ be some initial $o(n)$ length tape segment, let $[\Theta(n),\infty]$ be a final tape segment, consisting of all of the tape but an initial $\Theta(n)$-length tape segment, and let $(o(n),\Theta(n))$ be the elusive segment in between. See the picture below.



The idea is to show that there is an integer function $m(n)$, $m(n) \in \Theta(n)$, such that the storage head stays on the right tape segment $[\Theta(n),\infty]$ during the interval of steps $I_{m(n),\sqrt{m(n)}}$. Consequently, all information originally recorded on the initial tape segment $[0,o(n)]$, must have been transported over the intervening tape segment $(o(n),\Theta(n))$, in order to be accessed during the interval of steps $I_{m(n),\sqrt{m(n)}}$. This then entails long crossing sequences associated with *each* square of $(o(n),\Theta(n))$, the sum of which yields the claimed lower bound on $T(n)$. Below we define three functions, viz. S, P and Q, from the cartesian product of the positive rationals and the natural numbers into the powerset of the natural numbers. Let

$$S(c,n) \overset{def}{=} \{j \mid 1 \leq j \leq n \textit{ and the tape segment visited in } I_{j,1} \textit{ is contained in } [0,cn]\ \} .$$

With $[0,cn]$ we mean the initial tape segment consisting of the first $\lceil cn \rceil$ squares. Similar statements are tacitly assumed in all cases hereafter where comparable issues crop up.

Claim 1. If $T(n) \in o(n^2)$ then $\# S(c,n) \in O(cn)$.

Proof Sketch. Assume $T(n) \in o(n^2)$ and $\# S(c,n) \in \Theta(n)$, that is, there is a constant $\delta > 0$ such that, for each $c > 0$,we have $\# S(c,n) \geqslant \delta n$ for infinitely many n. By a crossing sequence argument we can show that then $T(n) \in \Omega(n^2)$: contradiction. □ □

Let

$$P(c,n) \overset{def}{=} \{j \mid 1 \leqslant j \leqslant n \textit{ and the tape segment visited in } I_{j,\sqrt{j}} \textit{ has at least length } cj \ \}.$$

Claim 2. If $T(n) \in o(n\sqrt{n})$ then $\# P(c,n) \in O(cn)$.

Proof Sketch. Assume $T(n) \in o(n\sqrt{n})$ and $\# P(c,n) \in \Theta(n)$, that is, there is a constant $\delta > 0$ such that, for each $c > 0$, we have $\# P(c,n) \geqslant \delta n$ for infinitely many n. Thus, for infinitely many such n, there are δn values of j, $1 \leqslant j \leqslant n$, for which $T(j+\sqrt{j}) - T(j) \geqslant cj$. It therefore follows that, for such n:

$$T(n) \geqslant \sum_{j=1}^{\delta n} \frac{c\sqrt{j}}{2} ,$$

and hence $T(n) \in \Omega(n\sqrt{n})$: contradiction. □ □

Assume $T(n) \in o(n\sqrt{n})$. Then from Claims 1 and 2 it follows that, for

$$Q(c,n) \overset{def}{=} \{j \mid 1 \leqslant j \leqslant n \textit{ and the tape segment visited in } I_{j,\sqrt{j}} \textit{ is contained in } [cn,\infty] \ \},$$

we have

$$Q(c,n) \supseteq \{1,2, \ldots , n\} - S(2c,n) - P(c,n) .$$

Therefore, $\# Q(c,n) \geqslant n - \epsilon n$, for arbitrary small $\epsilon > 0$, depending on the choice of c. Hence, for $m(c,n) \overset{def}{=} \max(Q(c,n))$ we can, for each $\epsilon > 0$, choose a $\delta > 0$ such that $m(\delta,n) \geqslant n - \epsilon n$. Since $T(n) \in o(n\sqrt{n})$, we have $T(\sqrt{n}) \in o(n^{3/4})$. Considering the input ensemble

$$\{push\ 0, push\ 1\}^{\sqrt{n}} \{skip\}^{m(\delta,n)-\sqrt{n}} \{pop\}^{\sqrt{n}} ,$$

we must pop an arbitrary sequence of 0's and 1's of length $\sqrt{n}$, originally recorded on the initial $o(n)$ length tape segment $[0,o(n)]$, completely, while never leaving the tape segment $[\Theta(n),\infty]$, see picture. Again using a crossing sequence argument, we obtain the contradictory $T(n) \in \Omega(n\sqrt{n})$, and hence the Theorem. □

REFERENCES

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

[FMR] Fischer, P.C., A.R. Meyer & A.L. Rosenberg, [1968], Counter machines and counter languages, *Math. Systems Theory*, **2** , 265-283.

[HS] Hennie, F.C. & R.E. Stearns, [1966], Two-tape simulation of multitape Turing machines, *J. ACM*, **13** , 533-546.

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

[J] Janiga, L., [1979], Real-time computations of two-way multihead finite automata, Fundamentals of Computation Theory (FCT '79), L. Budach ed., Akademie Verlag, Berlin, DDR, 1979, 214-218.

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

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

[PF] Pippenger, N. & M.J. Fischer, [1979], Relations among complexity measures, *J. ACM*, **26** , 361-384.

[SV] Savitch, W.J. & P.M.B. Vitányi, [1982], On the power of real-time two-way multihead finite automata with jumps. Tech. Rept. CS-056, Dept. EECS, Univ. of California, San Diego.

[V1] Vitányi, P.M.B., [1980], On the power of real-time Turing machines under varying specifications, 7th Coll. on Automata, Languages and Programming (ICALP '80), *Lecture Notes in Computer Science*, **85** , Springer Verlag, Berlin, 658-671.

[V2] Vitányi, P.M.B., [1982], Real-time simulation of multicounters by oblivious one-tape Turing machines, 14th ACM Symp. on Theory of Computing, 27-36.

[V3] Vitányi, P.M.B., [1983], An optimal simulation of counter machines: the ACM case. Tech. Rept. IW 225, Mathematisch Centrum, Amsterdam, March 1983.