Two tapes are better than one for off-line Turing machines

WOLFGANG MAASS[1], GEORG SCHNITGER[2], ENDRE SZEMEREDI[3]

Abstract: We prove the first superlinear lower bound for a concrete decision problem in P on a Turing machine with one work tape and a two-way input tape (also called: off-line 1-tape Turing machine). In particular we show for off-line Turing machines that 2 tapes are better than 1 and that 3 pushdown stores are better than 2 (both in the deterministic and in the nondeterministic case).

[1]Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago. Supported in part by NSF Grant DCR-8504247.

[2]Department of Computer Science, Pennsylvania State University. Supported in part by NSF Grant DCR-8407256.

[3]Department of Computer Science, Rutgers University, and Hungarian Academy of Sciences, Budapest.

A 1-tape off-line Turing machine (see Hopcroft, Ullman [4]) is a Turing machine (TM) with one work tape and an additional two-way input tape, i.e., an input tape with endmarkers ¢, $ on which the associated read-only input head can move without restriction in both directions. These TM's are used as the standard model for the analysis of the space-complexity of TM-computations. In addition they are of interest as an intermediate model between the relatively slow 1-tape TM without input tape and the very powerful (and so far intractable) 2-tape TM.

Preceding lower bound arguments for concrete languages on restricted TM's have progressed from 1-tape TM's without input tape in [10], [3] to 1-tape TM's with a one-way input tape (i.e., the input head is not allowed to back up) in [2], [7], [5] (it should be pointed out that some authors have called this latter model a 1-tape off-line TM). However one has not been able to prove for any concrete language in P a superlinear lower bound on the computation time for the here considered model of a 1-tape off-line TM with a two-way input tape. In particular the old question whether there is any language that can be recognized faster on a 2-tape TM than on a 1-tape off-line TM has remained open for many years.

Some progress has recently been made with regard to lower bounds on the computation time for the computation of functions on 1-tape off-line TM's. Optimal lower bounds for matrix transposition on this type of TM have been shown both for the regular version ([8]) and the more powerful model with an additional two-way output tape ([1]). Unfortunately these new lower bound arguments (via Kolmogorov complexity) can not be applied to decision problems. We present in this paper a different lower bound technique for 1-tape off-line TM's (via graph theoretic separation arguments) that yields a superlinear lower bound for a concrete decision problem (the problem of deciding for two given matrices A and B whether B is the transpose of A). This allows us to separate complexity classes for 2-tape TM's and 1-tape off-line TM's (both in the deterministic and the nondeterministic case). It also yields the first separation between off-line TM's with 3 and with 2 pushdown stores.

Our lower bound argument is based on a combinatorial analysis of computation graphs for off-line 1-tape TM's. It turns out that in spite of the fact that these graphs are in general not planar (because of the presence of the input tape) one can surpress those edges that are caused by the movement of the input head and represent instead the positions of the input head as labels on the nodes of a planar graph. We then cut the resulting labeled planar graph into a large number of small pieces and analyze how much communication about different input segments (= labels) has to be transmitted between these pieces.

The language SMT (= sparse matrix transposition), which separates the mentioned complexity classes, is defined as follows. In this paper we say that a Boolean matrix $A = (a_{ij})_{1 \leq i,j \leq m}$ is sparse if $a_{ij} \neq 0$ implies that both i and j are multiples of $\lceil \log m \rceil$ (we write $\log m$ instead of $\lceil \log m \rceil$ in the following). We code each Boolean matrix A by a string over $\{0,1,*\}$ that lists the entries of A in rowwise order, with * used as separation marker between successive rows. A^t is the transpose of matrix A.

$$SMT = \{A{*}{*}B \mid A \text{ and } B \text{ are sparse Boolean } m \times m \text{ matrices (for some } m \in N) \text{ and } A^t = B\}.$$

THEOREM 1. The language SMT can be accepted by a deterministic Turing machine with 2 work tapes (even without input tape) in time $O(n)$. SMT can not be accepted by any nondeterministic 1-tape off-line Turing machine (i.e., a TM with one work tape and a two-way input tape) in time $o(n \cdot \log n / \log \log n)$.

COROLLARY 1 ("2 TAPES VERSUS 1"). A linear time deterministic 2-tape Turing machine, with or without input tape, cannot be simulated by a deterministic or nondeterministic 1-tape off-line Turing machine in time $o(n \cdot \log n / \log \log n)$.

Remarks:

1. Both in the deterministic and the nondeterministic case this yields for off-line TM's the first superlinear lower bound for the simulation of two tapes by one (the best known upper bound is $O(n^2)$ [4], due to Hartmanis and Stearns).

2. The lower bound for off-line 1-tape TM's in Theorem 1 holds of course also for the more natural decision problem of deciding for arbitrary Boolean $m \times m$ matrices A and B whether

$A^t = B$. The sparseness requirement in SMT is only needed to get a linear upper bound on a 2-tape TM.

3. One can easily construct from SMT and $\overline{SMT}$ a language L that is accepted by a deterministic 2-tape TM in linear time, but where <u>neither</u> L <u>nor</u> $\overline{L}$ can be accepted by a nondeterministic 1-tape off-line TM in time $o(n \cdot \log n / \log\log n)$.

<u>Proof of Theorem 1</u>: For the upper bound we first note that a 2-tape TM (with or without input tape) can compute in $O(n \log n)$ steps the function $A \to A^t$ (where A is a Boolean $m \times m$ matrix and $n = \theta(m^2)$ is the length of the input string). This algorithm exploits the fact that the 2-tape TM can move its two read/write heads in such a way that with $\log m$ simultaneous sweeps of both heads (where each sweep takes $O(n)$ steps) it realizes the butterfly graph between the m blocks of length $\theta(m)$ on the input area of tape 1 (which initially hold the rows of A) and the m blocks of length $\theta(m)$ on the output area on tape 2 (which hold in the end the columns of A). The matrix transposition proceeds accordingly in $\log m$ phases. At the beginning of each phase the considered areas of length n on both tapes contain the same "partially converted" matrix. During each phase the TM exchanges matrix elements between those pairs of blocks on the two tapes that are simultaneously visited by the two heads. At the end of each phase the TM copies the new matrix on both of its tapes (this requires another simultaneous sweep of both heads of length n).

In order to check in linear time on a 2-tape TM whether a given input $x \in \{0,1,*\}^*$ lies in SMT, one first checks whether x has the form A**B for two sparse $m \times m$ matrices A and B (obviously this can be done in linear time). If this is the case, the 2-tape TM "collapses" in linear time the sparse $m \times m$ matrics A and B to their nontrivial $(m / \log m) \times (m / \log m)$ submatrices $\tilde{A}$, $\tilde{B}$. Then it computes $\tilde{A}^t$ with the previously described algorithm in $O((m / \log m)^2 \cdot \log m) = O(n)$ steps and checks whether $\tilde{A}^t = \tilde{B}$ (this is equivalent to $A^t = B$).

In order to prove the lower bound one assumes for a contradiction that there exists a nondeterministic off-line 1-tape TM M that accepts SMT in time $n \cdot c(n)$ with $c(n) = o(\log n / \log\log n)$. Fix some sufficiently large n of the form $2m \cdot (m + 1))$. For simplicity we write c for trices A, B the input string A**B has length $2m \cdot (m + 1))$. For simplicity we write c for $c(n)$ and we delete "$\lfloor\ \rfloor$" and "$\lceil\ \rceil$".

Let X_0 be the set of all $2^{(m / \log m)^2}$ inputs $A{*}{*}A^t \in$ SMT of length n. For each input $I \in X_0$ fix an accepting computation C_I of M with $\le n \cdot c(n)$ steps.

A simple counting argument shows that there exists for every input $I \in X_0$ a partition P_I of both the input- and work-tape of M into blocks of length $b = 2 \cdot (m + 1) \cdot \log m$ such that the heads of M cross at most $2nc/b$ times a block boundary of partition P_I during computation C_I (there are b different partitions of both tapes which result from shifting all block boundaries by an equal number of cells; the TM M crosses at each step of computation C_I for at most two different partitions a block boundary on the input tape or on the work tape). Choose a set $X_1 \subseteq X_0$ of size $|X_1| = |X_0|/b = 2^{(m / \log m)^2} / b$ so that the tape partitions P_I are identical for all $I \in X_1$.

Another counting argument shows that there exists a set $X_2 \subseteq X_1$ of size

$$\frac{|X_1|}{2^{K_1 \cdot n \cdot c \cdot \log n / b}} = \frac{2^{(m/\log m)^2 - K_1 \cdot n \cdot c \cdot \log n / b}}{b}$$

(where $K_1 > 0$ is a constant) so that the following data are identical for all $I \in X_2$:

- the time points where a head of M crosses a block boundary of P_I during computation C_I
- the state of M at each such crossing
- the precise locations and the direction of movement for each head of M at each such crossing.

The time interval between two successive crossings of block boundaries of P_I will be called in the following a <u>time block</u> of computation C_I. Note that all computations C_I, $I \in X_2$ are "block-oblivious", i.e., the same tape blocks are examined during corresponding time blocks for each such computation.

Let $G = \langle V,E \rangle$ be the common "block computation graph" of the computations C_I, $I \in X_2$. Each node of G represents a time block of C_I, thus $|V| \leq 2nc/b$. The linear order of time induces a linear order "<" on the elements of V. Two nodes $v_1, v_2 \in V$ with $v_1 < v_2$ are connected by an edge of E if for some computation C_I, $I \in X_2$ (or equivalently; for <u>all</u> C_I, $I \in X_2$) the work head of M visits during time block v_2 a tape block that it had last visited during time block v_1.

This graph G does not reflect the complete information flow of the considered computations C_I because it ignores the input tape of M. However G has an important advantage that will be exploited extensively in the following: it is planar (because one can simulate the work tape of M by 2 pushdown stores and therefore graph G has pagenumber 2, see [9], [12]).

According to Ullman's refined version [11] of the Lipton-Tarjan Planar Separator Theorem [6] there exists a constant $K_2 > 0$ so that any planar graph $G = \langle V,E \rangle$ can be split into three vertex sets A, B, C with $|A|, |B| \leq \frac{|V|}{2}$, $|C| \leq K_2 \cdot \sqrt{|V|}$, so that there is in E no edge between A and B. By iterating this separator theorem j times one gets a set S_j of at most

$$\sum_{i=0}^{j-1} K_2 \cdot \sqrt{\frac{2nc}{b \cdot 2^i}} \cdot 2^i = K_2 \cdot \sqrt{\frac{2nc}{b}} \cdot \sum_{i=0}^{j-1} 2^{i/2}$$

$\leq K \cdot \sqrt{\frac{n \cdot c \cdot 2^j}{b}}$ vertices (for some constant $K > 0$) so that after removal of all vertices in S_j one can split the remaining vertices of V into 2^j pairwise disconnected vertex sets of size at most $\frac{2nc}{b \cdot 2^j}$. For the following we choose $j = \log(64c^2)$. For this value of j the separator $S_j =: S$ consists of $O\left(\sqrt{\frac{n \cdot c^3}{b}}\right)$ vertices and the resulting $2^j = 64c^2$ pairwise disconnected sets are each of size at most $\frac{n}{32 \cdot b \cdot c}$.

For any $I \in X_2$ and any time block v of computation C_I one needs $O(b)$ bits to describe the contents of the currently visited block on the work tape and the input tape (at the beginning and the end of v). Therefore we can select a set $X_3 \subseteq X_2$ of

$$\frac{2^{(m/\log m)^2 - K_1 \cdot \frac{n \cdot c \cdot \log n}{b} - K_3 \sqrt{nc^3 b}}}{b}$$

inputs I so that for all nodes v in the separator S these block contents are identical for all $I \in X_3$ ($K_3 > 0$ is a constant).

Fix a set B of $n/4b$ blocks (in partition P_I) from the left half of the input tape so that each block in B is visted during at most 8c time blocks of C_I (for $I \in X_3$). There is a set LEFT $\subseteq$ B of

$$\frac{n}{4b \cdot \binom{64c^2}{8c}} \geq \frac{n}{4b \cdot (64c^2)^{8c}} =: \ell$$

input blocks that are visited during the <u>same</u> 8c pieces $P_1,\ldots,P_8$ among the $64c^2$ pairwise disconnected pieces of the separated computation graph G. These 8c vertex sets $P_1,\ldots,P_8$ have access (via a visit of the input head during a time block $v \in P_i$) to at most $8c \cdot n/32bc = n/4b$ of the n/b input blocks. Therefore we can select a set RIGHT of $\ell = |\text{LEFT}|$ blocks from the right half of the input tape that are never visited by the input head during any of the considered 8c graph pieces $P_1,\ldots,P_8$.

Each block of the input tape that belongs to LEFT holds for input $I = A{*}{*}A^t$ at least one complete row r_i of matrix A whose index i is a multiple of $\log m$. Analogously each block of RIGHT holds at least one complete column c_j of A where j is a multiple of $\log m$. Thus there is a sequence $\langle i_1,\ldots,i_\ell\rangle$ of row numbers and a sequence $\langle j_1,\ldots,j_\ell\rangle$ of column numbers which are multiples of $\log m$ and which define for every input $I = A{*}{*}A^t \in X_3$ an $\ell \times \ell$ submatrix $\tilde{A} = (a_{i_r j_s})_{1\leq r,s\leq\ell}$ of $A = (a_{ij})_{1\leq i,j\leq m}$ so that all matrix elements a_{ij} of A that belong to the submatrix $\tilde{A}$ appear on the input tape exclusively in those tape blocks that belong to LEFT or RIGHT. In order to achieve the desired contradiction it is sufficient to show that there are two <u>different</u> inputs $I = A{*}{*}A^t$ and $I' = A'{*}{*}A'^t$ in X_3 that agree outside of the considered submatrix (i.e., $a_{ij} = a'_{ij}$ if $i \notin \{i_1,\ldots i_\ell\}$ or $j \notin \{j_1,\ldots j_\ell\}$). We then amalgamate these inputs I, I' to a new input $\tilde{I} = A{*}{*}A'^t \notin$ SMT that is accepted by M because one can "cut" the accepting computations C_I and $C_{I'}$ at the separator nodes and replace in C_I those pieces of the computation graph where the input head visits a block of RIGHT by corresponding pieces of $C_{I'}$.

In order to show that there are two different inputs I, I' in X_3 that agree outside of the considered submatrix we prove that the cardinality $|X_3|$ of X_3 is larger than $2^{(m/\log m)^2 - \ell^2}$, which is the number of sparse Boolean matrices that are pairwise different outside of the considered $\ell \times \ell$ submatrix.

By assumption there exists for every $\varepsilon > 0$ some $n_\varepsilon \in \mathbb{N}$ so that $c(n) \leq \varepsilon \cdot \log n / \log\log n$ for $n \geq n_\varepsilon$. This implies for $\varepsilon \leq 1/8$ and $n \geq n_\varepsilon$ that

$$(64c^2)^{16c} \cdot c^{3/2} \leq$$

$$(64 \cdot \varepsilon^2 \cdot \log^2 n)^{16\varepsilon \cdot \log n / \log\log n} \cdot (\log n)^{3/2} \leq$$

$$2^{2 \cdot \log\log n \cdot 16\varepsilon \cdot \log n / \log\log n} \cdot (\log n)^{3/2} \leq$$

$$n^{32\varepsilon} \cdot (\log n)^{3/2} \leq n^{33\varepsilon}.$$

On the other hand

$$\frac{n^{3/2}}{b^{5/2}} = \Theta\left(\frac{n^{3/2}}{n^{5/4} \cdot (\log n)^{5/2}}\right) = \Omega(n^{1/8}), \quad \text{and}$$

therefore the preceding implies that

$$(64c^2)^{16c} \cdot c^{3/2} = o\left(\frac{n^{3/2}}{b^{5/2}}\right).$$

Hence

$$\sqrt{nc^3 b} = o\left(\frac{n^2}{b^2 \cdot (64c^2)^{16c}}\right),$$

which implies together with

$$\frac{n \cdot c \cdot \log n}{b} + \log b = o(\sqrt{nc^3 b})$$

that

$$K_1 \cdot \frac{n \cdot c \cdot \log n}{b} + K_3 \cdot \sqrt{nc^3 \cdot b} + \log b$$
$$= o\left(\frac{n^2}{b^2 \cdot (64c^2)^{16c}}\right).$$

Thus for sufficiently large n one has $|X_3| =$

$$2^{(m / \log m)^2} - K_1 \cdot \frac{n \cdot c \cdot \log n}{b} - K_3 \cdot \sqrt{nc^3 b} - \log b$$
$$> 2^{(m / \log m)^2 - \frac{n^2}{16 \cdot b^2 \cdot (64c^2)^{16c}}}$$
$$= 2^{(m / \log m)^2 - \ell^2}.$$

This completes the proof of Theorem 1.

Remark: The proof of Theorem 1 shows that the same upper and lower bound also holds for each of the following languages SMT_d. Let $d : \mathbb{N} \to \mathbb{N}$ be some function with $(\log m)^{1/2} \leq d(m) \leq m^{(1/5) - \delta}$ for some $\delta > 0$ so that the map $1^m \to 1^{d(m)}$ from $\{1\}^*$ into $\{1\}^*$ can be computed on a 2-tape TM in time $O(m^2)$. Call a Boolean $m \times m$ matrix $A = (a_{ij})_{1 \leq i,j \leq m}$ d-sparse if $a_{ij} \neq 0$ implies that both i and j are multiples of $d(m)$. Define

$SMT_d = \{A**B \mid A$ and B are d-sparse quadratic matrices with $A^t = B\}$.

THEOREM 2. ("3 PUSHDOWN STORES ARE BETTER THAN 2 FOR OFF-LINE TM's"). SMT can be accepted in linear time by a deterministic off-line Turing machine with 3 pushdown stores (no input tape is required), but SMT can not be accepted by a nondeterministic off-line Turing machine with 2 pushdown stores in time $o(n \cdot \log n / \log \log n)$.

Remark: This yields both in the deterministic and the nondeterministic case the first separation result for off-line TM's with different numbers of pushdown stores.

The proof of Theorem 2 relies on the fact that the lower bound argument of Theorem 1 also holds if M has 2 pushdown stores instead of 1 work tape. Furthermore the described linear time algorithm for SMT on 2-tape TM's also yields a linear time algorithm for TM's with 3 pushdown stores.

REFERENCES

[1] M. Dietzfelbinger, W. Maass, The complexity of matrix transportation on one-tape off-line Turing machines with output tape, Research Report in Computer Science, University of Illinois at Chicago (March 1987).

[2] P. Duris, Z. Galil, W.J. Paul, R. Reischuk, Two nonlinear lower bounds, Proc. 15th ACM STOC, (1983), 127-132.

[3] F.C. Hennie, One-tape off-line Turing machine computations, Inf. and Control, 8 (1965), 553-578.

[4] J.E. Hopcroft and J.D. Ullman, Introduction to Automata Theory, Languages and Computation, Addison-Wesley (Reading, 1979).

[5] M. Li, L. Longpre, P. Vitanyi, The power of the queue, Proc. of Structure in Complexity Theory, Berkeley, 1986, Lecture Notes in Computer Science vol. 223 (Springer, 1986), 219-233.

[6] R.J. Lipton, R.E. Tarjan, A separator theorem for planar graphs, SIAM J. Appl. Math., 36 (1979), 177-189.

[7] W. Maass, Combinatorial lower bound arguments for deterministic and nondeterministic Turing machines, Trans. of the Amer. Math. Soc., 292(1985), 675-693.

[8] W. Maass, G. Schnitger, An optimal lower bound for Turing machines with one work tape and a two-way input tape, Proc. of Structure in Complexity Theory, Berkeley 1986, Lecture Notes in Computer Science vol. 233 (Springer 1986), 249-264.

[9] W.J. Paul, N. Pippenger, E. Szemeredi, W. Trotter, On determinism versus nondeterminism and related problems, Proc. 24th IEEE FOCS, (1983), 429-438.

[10] M.O. Rabin, Real time computation, Israel J. of Math., 1(1963), 203-211.

[11] J.D. Ullman, Computational Aspects of VLSI, Computer Science Press (1984).

[12] M. Yannakakis, Four pages are necessary and sufficient for planar graphs, Proc. of the 18th Annual ACM STOC, (1986), 104-108.