comput complexity 3 (1993), 392–401 1016-3328/93/040392-10 $1.50+0.20
©1993 Birkhäuser Verlag, Basel

TWO TAPES VERSUS ONE FOR OFF-LINE TURING MACHINES

Wolfgang Maass, Georg Schnitger, Endre Szemerédi and György Turán

Abstract. We prove the first superlinear lower bound for a concrete, polynomial time recognizable decision problem on a Turing machine with one work tape and a two-way input tape (also called off-line 1-tape Turing machine).

In particular, for off-line Turing machines we show that two tapes are better than one and that three pushdown stores are better than two (both in the deterministic and in the nondeterministic case).

Key words. off-line 1-tape Turing machines; two tapes; lower bounds; time; nondeterminism.

Subject classifications. 68Q05, 68Q25.

1. Introduction

A 1-tape off-line Turing machine (see Hennie 1965, p.166) is a Turing machine (TM) with one work tape and an additional two-way input tape, i.e., an input tape with end markers 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 relatively powerful 2-tape TM.

No non-trivial lower bounds are known for the recognition of polynomial time computable languages on 2-tape Turing machines. On the other hand, lower bound arguments for concrete languages on restricted TM's have progressed from 1-tape TM's without input tape (Hennie 1965, Rabin 1963) to

1-tape TM's with a 1-way input tape (i.e., the input head is not allowed to back up) in Duris *et al.* (1983), Li & Vitanyi (1988), Li *et al.* (1986), and Maass (1985). However, noone has been able to prove a superlinear time bound for the computation of any concrete, polynomial time recognizable language on the 1-tape off-line model.

Some progress has recently been made with respect to lower time bounds for the computation of *functions* on 1-tape off-line TM's. Optimal lower bounds for matrix transposition on this model have been shown both for the regular version (Maass & Schnitger 1986) and the more powerful model with an additional two-way output tape (Dietzfelbinger & Maass 1986). Unfortunately, these new lower bound arguments can not be applied to decision problems since they utilize the information content of the output. We present in this paper a different lower bound technique for 1-tape off-line TM's 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). Our technique is based on graph theoretic separation arguments coupled with information theoretic reasoning. 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 three and off-line TM's with two 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 suppress those edges that are caused by the movement of the input head and represent instead the positions of the input head as labels for 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 has to be exchanged between these pieces.

The language SMT (*sparse matrix transposition*) which separates the above mentioned complexity classes is defined as follows. We say that a boolean matrix $A = (a_{i,j})_{1\leq i,j\leq m}$ is *sparse* if $a_{i,j} \neq 0$ implies that both i and j are multiples of $\lceil \log_2 m \rceil$. We code each boolean matrix A by a string over $\{0,1,*\}$ that lists the entries of A in row-wise order, with $*$ used as separation marker between successive rows. Let code(A) denote the above coding of matrix A. Finally, let A^t denote the transpose of A. Then,

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

In the following, n denotes the input size.

THEOREM 1.1. *The language SMT can be accepted by a deterministic Turing machine with two work tapes (even without a special input tape) in time $O(n)$. SMT cannot be accepted by any nondeterministic 1-tape off-line Turing machine in time $o(n\log_2 n)$.*

COROLLARY 1.2. *A linear time deterministic 2-tape Turing machine, with or without special input tape, cannot be simulated by a nondeterministic 1-tape off-line Turing machine in time $o(n\log_2 n)$.*

REMARK 1.3.

(a) *Both in the deterministic and nondeterministic case, for off-line TM's, this yields the first superlinear lower bound for the simulation of two tapes by one. The best known upper bound is $O(\frac{n^2}{\log n})$ (Dietzfelbinger 1989), improving the upper bound of Hartmanis and Stearns (see Hopcroft & Ullman 1979).*

(b) *One can easily construct from SMT and its complement a language L that is accepted by a deterministic 2-tape TM in linear time, but where neither L nor its complement can be accepted by a nondeterministic 1-tape off-line TM in time $o(n\log_2 n)$.*

Our techniques also allow us to separate off-line Turing machines with three pushdown stores and those with two pushdown stores.

THEOREM 1.4. *SMT can be accepted in linear time by a deterministic off-line Turing machine with three pushdown stores (no input tape required), but SMT can not be accepted by a nondeterministic off-line Turing machine with two pushdown stores in time $o(n\log_2 n)$.*

The next section describes linear time decision procedures for 2-tape Turing machines and Turing machines with three pushdown stores. The proof of Theorem 1.1 is given in section 3. Section 4 contains a graph-theoretical lemma which is required for the proof of Theorem 1.1.

2. Recognizing SMT in Linear Time with two Tapes

In order to check with a 2-tape TM whether a given input $x \in \{0, 1, *\}^*$ belongs to SMT, one first checks whether x has the form code(A)**code(B) for two sparse $m \times m$ matrices A and B. Obviously, this can be done in linear time (i.e., in time $O(m^2)$).

If the input passes this test, then in linear time, the sparse $m \times m$ matrices A and B are collapsed to $(m/\log_2 m) \times (m/\log_2 m)$ matrices A' and B'. More precisely,

$$A'[i,j] = (j, i, A[i \log_2 m, j \log_2 m]) \text{ and } B'[i,j] = (i, j, B[i \log_2 m, j \log_2 m]).$$

Next, we observe that a 2-tape TM can sort s keys in time $O(s \log_2^2 s)$ assuming that all keys consist of at most $O(\log_2 s)$ bits. This is achieved, for instance, by simulating a non-recursive version of Mergesort. If we now sort the triples of A' lexicographically (in time $O((m/\log_2 m)^2 \log_2^2 m) = O(m^2)$), then we obtain the (row-wise representation of the) matrix A'' with

$$A''[i,j] = (i, j, A[j \log m, i \log m]).$$

But A'' represents the collapsed version of A^t and it now suffices to check whether A'' equals B'.

Thus, SMT can be accepted in linear time overall. Observe that the above algorithm also runs in linear time for TM's with three pushdown stores (since three pushdowns already allow an efficient implementation of Mergesort).

3. A Lower Bound for 1-Tape off-line Turing machines

Assume that there exists a nondeterministic off-line 1-tape TM M which recognizes SMT in time $n \cdot c(n)$ where $c(n) = o(\log_2 n)$. We will now show that such a machine M cannot exist.

Fix some sufficiently large n of the form $n = 2m(m+1) + 2$ (so n is the length of the input string code(A)**code(B) for boolean $m \times m$ matrices A and B). For simplicity we write c for $c(n)$ and we omit the floor and ceiling notation.

Let $k_0 = (m/\log_2 m)^2$ and let X_0 be the set of all 2^{k_0} inputs of length n belonging to SMT. For each input $I \in X_0$ fix an accepting computation C_I of M which runs in at most $n \cdot c$ steps.

Now, let us fix some input $I \in X_0$. Consider partitions of both the input and the work tape of M into blocks of $b = 2(m+1)\log_2 m$ adjacent cells. There are b different partitions of both tapes which result from shifting all block boundaries by an equal number of cells. At each step of computation C_I, M crosses a block boundary for at most two different partitions. Therefore, we will be able to find one partition P_I such that the heads of M cross block boundaries at most $2nc/b$ times during C_I.

Next choose a set $X_1 \subseteq X_0$ with $| X_1 | = |X_0|/b = 2^{k_0}/b$ so that the tape partitions P_I are identical for all $I \in X_1$.

Let $k_1 = C_1 \dfrac{n \cdot c \cdot \log_2 n}{b}$ where C_1 is a positive constant. Another counting argument shows that there exists a subset X_2 of X_1 of size at least

$$\frac{|X_1|}{2^{k_1}} = \frac{2^{k_0-k_1}}{b}$$

so that the following data are identical for all $I \in X_2$,

- the time steps when 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 location 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 *time block* of computation C_I. Note that all computations C_I for $I \in X_2$ are "block-oblivious", i.e., the same tape blocks are examined during corresponding time blocks for each such computation. Let us fix one input $I \in X_2$.

Let $G = (V, E)$ be the "block computation graph" of the computation C_I. Each node of G represents a time block of C_I, thus $|V| \leq 2nc/b$. The pair (v_1, v_2) is an edge of G if and only if (for C_I) during time block v_2 the work head of M visits a tape block that it had last visited during time block v_1. Observe that the graph G captures the work tape/time dependence for **all** inputs $I \in X_2$ since their computations are block-oblivious.

This graph G does not reflect the complete information flow of the computation 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 a planar graph, since the work tape of M can be simulated by two pushdown stores (Paul *et al.* 1983).

Before exploiting the planarity, we capture the actions of the read-only head. Label every node v of G with the index of the input tape block which has been visited during time block v. Next, select $n/4b$ tape blocks from the left half of the input tape and the same number from the right half such that each such block appears at most $8c$ times as a label in G. Let Z (resp. Z') be the set of nodes corresponding to the chosen blocks from the left (resp. right) half.

By a repeated application of the planar separator theorem of Lipton & Tarjan (1979), the graph G can be cut into relatively few pieces with few edges connecting the pieces. We would then like to combine the pieces into two disjoint groups, with each group only accessing parts of the input not accessible to the other. This leads us to a problem of Communication Complexity. Namely the language SMT, restricted to inputs in X_2, has to be recognizable by exchanging a number of bits not larger than the product of $O(b)$ and the number of edges connecting the two groups. Lemma 3.1 allows us to carry out the graph-theoretical part of the above process.

LEMMA 3.1. *Let $G = (V, E)$ be a planar graph, Z and Z' be disjoint sets of s labels, and $\alpha : V \to Z \bigcup Z'$ be a (partial) labeling of G such that each label occurs at most k times. Then, there are subsets $Z_0 \subseteq Z$, $Z'_0 \subseteq Z'$ and $V^* \subseteq V$ such that the following conditions hold:*

(a) $|Z_0| = |Z'_0| \geq s/(9^k)$,

(b) $|V^*| = O(k\sqrt{|V|})$ *and*

(c) after removing V^*, none of the remaining connected components contain labels from both Z_0 and Z'_0.

The proof of Lemma 3.1 is given in the next section. If we apply Lemma 3.1 with $s = n/4b$ and $k = 8c$, we obtain subsets Z_0 and Z'_0 of size

$$z = \frac{n}{4 \cdot b \cdot 9^{8c}}$$

and a subset V^* of size $O(8c\sqrt{|V|}) = O(\sqrt{nc^3/b})$. The input blocks in Z_0 (Z'_0) will only appear as labels in connected components that do not contain an input block from Z'_0 (Z_0) as a label.

Now we can remove the influence of nodes in V^* on our communication problem. 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 index of the input block visited during v. This description, for all the nodes of V^*, costs us not more than $k_2 = C_2\sqrt{nc^3b}$ bits (for some positive constant C_2). Thus, we can select a set $X_3 \subseteq X_2$ of $2^{k_0-k_1-k_2}/b$ inputs I so that for all nodes v of V^* and for all $I \in X_3$, the corresponding block contents and input block addresses are identical.

Since each input block has length $2(m+1)\log_2 m$, a block belonging to Z_0 holds for input I=code(A)**code(A^t) at least one complete row r_i of matrix A whose index i is a multiple of $\log_2 m$. Analogously, each block of Z_0' holds at least one complete column c_j of A where j is a multiple of $\log_2 m$. Thus, with z equal to the size of Z_0 (and Z_0'), we obtain a sequence $(i_1, ..., i_z)$ of row numbers and a sequence $(j_1, ..., j_z)$ of column numbers which are multiples of $\log_2 m$ and which define for every input I=code(A)**code(A^t) $\in X_3$ a $z \times z$ submatrix A'. Moreover, all matrix elements of A that belong to A' appear on the input tape exclusively in those tape blocks that belong to Z_0 or Z_0'.

Observe that it is sufficient to show that there are two *different* inputs I=code(A)**code(A^t) and J=code(B)**code(B^t) in X_3 whose matrices A and B agree outside of the considered submatrix. This follows by a conventional cut-and-paste argument. We combine these inputs I and J to a new input code(A)**code(B^t) that is accepted by M. This is done by first "cutting" the accepting computations C_I and C_J at the nodes of V^* and then by replacing in C_I those disconnected subgraphs of the computation graph where the input head visits a block of Z_0' by the corresponding subgraphs of C_J.

In order to show that there are two different inputs I, J in X_3 that agree outside of the considered submatrix we prove that the cardinality of X_3 is larger than $2^{k_0 - z^2}$, which is the number of sparse boolean matrices that differ in an entry not contained in the considered $z \times z$ submatrix. Since $|X_3| = 2^{k_0 - k_1 - k_2 - \log_2 b}$, it suffices to show that $k_1 + k_2 + \log_2 b = o(z^2)$.

Since $b = \theta(m \log_2 m)$ and $n = \theta(m^2)$, we have

$$k_1 + \log_2 b = C_1 \frac{n \cdot c \cdot \log_2 n}{b} + \log_2 b = O(c\sqrt{n}) = O(\sqrt{nc^3 b}).$$

But, $k_2 = C_2\sqrt{nc^3 b}$. Thus, $k_2 = O(n^{4/5})$ and it suffices to show that $n^{4/5} = o(z^2)$.

Since $z = \dfrac{n}{4 \cdot b \cdot 9^{8c}}$ and $c(n) = o(\log_2 n)$, we get $z^2 = \theta(\dfrac{n}{9^{16c}\log_2 n})$ and z^2 grows faster than $n^{4/5}$, as required. This proves Theorem 1.1. □

For Theorem 1.4 we have to show that SMT can be recognized by a deterministic off-line Turing machine with three pushdown stores in linear time, but SMT cannot be recognized by a nondeterministic off-line Turing machine with two pushdown stores.

We already discussed the linear time recognition in the previous section. Observe that our lower bound argument only utilizes the planarity of the computation graph. Since the computation graph of a two-pushdown Turing machine is planar (Paul *et al.* *1983*), our lower bound also applies to this machine model. □

4. Separating Planar Graphs

For the proof of Lemma 3.1, we need the following result of Savage (1984). To describe it, we have to introduce partition trees. A binary tree T (where every interior node has exactly two children) is a partition tree for the set V if its nodes are labeled with subsets of V (i.e., node w is labeled with V_w), the root has label V and for every interior node w of T (V_{w0}, V_{w1}) is a partition of V_w.

FACT 4.1. *Let $G = (V, E)$ be a planar graph, $V' \subseteq V$, and $1 \leq p \leq |V'|$. Then, there is a partition tree T_p of V with p leaves such that*

(a) for every leaf w, $\frac{|V'|}{4p} \leq |V' \bigcap V_w| \leq 4\frac{|V'|}{p}$,

(b) for every node w there is a set $S_w \subseteq V$ such that $|S_w| = O(\sqrt{|V|})$ and no edges join V_w and $V - (V_w \bigcup S_w)$.

We also need a result of Babai *et al.* (1990) which generalizes a result of Alon & Maass (1988) from sequences to trees.

FACT 4.2. *Let T be a binary tree, Z and Z' be disjoint sets of size s, and β be a partial labeling of the nodes of T by elements of $Z \bigcup Z'$ such that each label occurs at most k times. Then, there are subsets $Z_0 \subseteq Z$ and $Z'_0 \subseteq Z'$ and a set F of edges of T such that*

(a) $|Z_0| = |Z'_0| \geq \frac{s}{9^k}$,

(b) $|F| \leq 2k - 1$, and

(c) after removing the edges of F from T none of the remaining components contain labels from both Z_0 and Z'_0.

Now, to prove Lemma 3.1, apply Fact 4.1 to G with V' being the set of labeled nodes and $p = |V'|$. Let T_p be the partition tree guaranteed by Fact 4.1. For each leaf w of T_p, the set V_w contains at least one node from V'. As there are $|V'|$ many leaves, each leaf contains exactly one labeled node. Label the leaf w with the label of this node.

Now apply Fact 4.2 to T_p. Consider the separating edge set F of T_p. For each edge $e \in F$ let α_e be the endpoint of e farther away from the root. Set

$$V^* = \bigcup_{e \in F} S_{\alpha_e},$$

where the sets S_α are given by Fact 4.1, part (b). Furthermore, let Z_0 and Z_0' be the sets provided by Fact 4.2. It follows directly that these sets satisfy the requirements. Thus, Lemma 3.1 is proved. □

Acknowledgements

A preliminary version, proving a slightly weaker result, appears in Maass *et al.* (1987). Georg Schnitger was partially supported by NSF grant CCR-9114545.

References

N. ALON AND W. MAASS, Meanders and their application in lower bound arguments. *J. Comput. System Sci.* **37** (1988), 118–129.

L. BABAI, P. PUDLÁK, V. RÖDL, AND E. SZEMERÉDI, Lower bounds to the complexity of symmetric boolean functions. *Theoret. Comput. Sci.* **74** (1990), 313–323.

M. DIETZFELBINGER, The speed of copying on one-tape off-line turing machines. *Inform. Process. Lett.* **33** (1989), 83–89.

M. DIETZFELBINGER AND W. MAASS, The complexity of matrix transposition on one-tape off-line turing machines with output tape, 1986. To appear in *Theoret. Comput. Sci.*

P. DURIS, Z. GALIL, W. J. PAUL, AND R. REISCHUK, Two nonlinear bounds. In *Proc. Fifteenth Ann. ACM Symp. Theor. Comput.*, 1983, 127–132.

F. C. HENNIE, One-tape off-line turing machine computations. *Inform. and Control* **8** (1965), 553–578.

J. E. HOPCROFT AND J. D. ULLMAN, *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley, 1979.

M. LI AND P. M. B. VITANYI, Tape versus queue and stacks: the lower bounds. *Inform. and Comput.* **78** (1988), 56–85.

M. LI, L. LONGPRE, AND P. M. B. VITANYI, The power of the queue. *Proc. of Structure in Complexity Theory, Lecture Notes in Computer Science* **223** (1986), 219–233.

R. J. LIPTON AND R. E. TARJAN, A separator theorem for planar graphs. *SIAM J. Appl. Math.* **36** (1979), 177–189.

W. MAASS, Combinatorial lower bound arguments for deterministic and nondeterministic turing machines. *Trans. Amer. Math. Soc.* **292** (1985), 675–693.

W. MAASS AND 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, Lecture Notes in Computer Science* **223** (1986), 249–264.

W. MAASS, G. SCHNITGER, AND E.SZEMERÉDI, Two tapes are better than one for off-line turing machines. In *Proc. Nineteenth Ann. ACM Symp. Theor. Comput.*, 1987, 94–100.

W. J. PAUL, N. PIPPENGER, E. SZEMERÉDI, AND W. TROTTER, On determinism versus nondeterminism and related problems. In *Proc. 24th Ann. Symp. Found. Comput. Sci.*, 1983, 429–438.

M.O. RABIN, Real time computation. *Israel J. of Math.* **1** (1963), 203–211.

J.E. SAVAGE, The performance of multilective vlsi algorithms. *J. Comput. System Sci.* **29** (1984), 243–272.

Manuscript received 12 September 1991

WOLFGANG MAASS
Institute for Theoretical
Computer Science
Technische Universität Graz
Klosterwiesgasse 32/2
A-8010 Graz, AUSTRIA
maass@igi.tu-graz.ac.at

GEORG SCHNITGER
FB 17, Mathematik-Informatik
Universität Paderborn
Warburger Strasse 100
33095 Paderborn, GERMANY
georgs@uni-paderborn.de

ENDRE SZEMERÉDI
Department of Computer Science
Rutgers University
New Brunswick, NJ 08903, USA
szemered@cs.rutgers.edu

GYÖRGY TURÁN
Department of Mathematics,
Statistics and Computer Science
University of Illinois at Chicago
Chicago, IL 60680,USA
U11557@uicvm.uic.edu