



TRADEOFFS FOR LANGUAGE RECOGNITION ON ALTERNATING MACHINES

Juraj HROMKOVIČ

*Department of Theoretical Cybernetics and Mathematical Informatics, Comenius University, 842 15
Bratislava, Czechoslovakia*

Communicated by M. Nivat

Received July 1986

Abstract. The alternating machine having a separate input tape with k two-way, read-only heads, and a certain number of internal configurations, $AM(k)$, is considered as a parallel computing model. For the complexity measure $TIME \cdot SPACE \cdot PARALLELISM$ (TSP), the optimal lower bounds $\Omega(n^2)$ and $\Omega(n^{3/2})$ respectively are proved for the recognition of specific languages on $AM(1)$ and $AM(k)$ respectively. For the complexity measure $REVERSALS \cdot SPACE \cdot PARALLELISM$ (RSP), the lower bound $\Omega(n^{1/2})$ is established for the recognition of a specific language on $AM(k)$. This result implies a polynomial lower bound on $PARALLEL TIME \cdot HARDWARE$ of parallel RAM's.

Lower bounds on the complexity measures $TIME \cdot SPACE$ and $REVERSALS \cdot SPACE$ of nondeterministic machines are direct consequences of the result introduced above.

All lower bounds obtained are substantially improved in the case that $SPACE \geq n^\epsilon$ for $0 < \epsilon < 1$. Several strongest lower bounds for two-way and one-way alternating (deterministic, nondeterministic) multihead finite automata are obtained as direct consequences of these results. The hierarchies for the complexity measures TSP, RSP, TS and RS can be immediately achieved too.

1. Introduction

One of the hardest problems in the theory of computations and computational complexity is to prove nontrivial lower bounds on different complexity measures for some specific problems. Much effort was invested and no expressive success was arriving. For example, we have no greater than a quadratic lower bound on the time complexity of any NP-complete problem [10], and we have no nonlinear lower bound on the combinational complexity of a specific Boolean function [2] (despite of the fact that we know that almost all Boolean functions have exponential combinational complexity [32, 33]).

Besides the nontrivial lower bounds on time [14, 17, 34-37, 42, 51-53], space [5, 40, 44, 54], or other single complexity measures (see for example [1, 2, 6, 13, 20-22, 29, 41, 45, 54]), attempts have been made to prove lower bounds on some convex functions of time and space. The motivation of the study of such complexity measures can be found in the fact that lower bounds on a single complexity measure (time, space) give no information about the behaviour of the other complexity measures.

The investigation of some convex functions of complexity measures brings knowledge of how the decrease of a complexity measure can be compensated by the increase of another complexity measure. This can be considered in general, and in relation to any specific problem, too.

In 1966 Cobham [9] proved $\text{TIME} \cdot \text{SPACE} \geq cn^2$ for the recognition of the language $L_R = \{wcw^R \mid w \in \{0, 1\}^*\}$. The machine model considered in [9] was a machine with the input on a separate tape with one two-way, read-only head, and with a number of internal configurations. The fact that L_R can be recognized in linear time by a two-way, two-head finite automaton led to considering a more general model of sequential computing with several jumping read-only heads on the input tape. Borodin and Cook [3] showed that $\text{TIME} \cdot \text{SPACE} \in \Omega(n^2/\log_2 n)$ for sorting n numbers in the range $[1, n^2]$ on this general model of sequential computations. The shortcoming of this result pointed out by the authors themselves in [3] is that the large number of output bits was essential for the proof technique used. This was overcome by Āuriš and Galil [12] who proved $\text{TIME}^2 \cdot \text{SPACE} \in \Omega(n^3)$ for the recognition of a specific language on a sequential, nondeterministic computing model with two-way, read-only heads on the separate input tape.

The computing model investigated in this paper is a generalisation of alternating devices, and the studied convex functions of complexity measures include the “parallel” complexity measure. Alternation is considered as introduced in [7], and the definition of the parallel complexity of alternating devices introduced in [22] is used. We note that a similar complexity measure was independently introduced in [29]. The only lower bounds on the parallel complexity of alternating devices were obtained in [29] for one-way alternating simple multihead finite automata, in [22] for one-way alternating multihead finite automata, and in [24] for multihead multitape alternating Turing machines.

Our computing model called *multihead alternating machine* is in fact the alternating version of the sequential computing model used in [12]. We shall prove several first and/or strongest lower bounds for language recognition on the convex functions of the complexity measures TIME , SPACE , REVERSALS , and PARALLELISM . The lower bounds on $\text{TIME} \cdot \text{SPACE}$ proved in [9, 12] are direct consequences of results obtained in this paper. The proof technique used is a generalisation of the proof techniques in [12, 19, 21, 22, 30] based on the idea of Rivest and Yao [46].

The paper consists of five sections. Section 2 involves the definition of the alternating computing model that we shall call multihead alternating machine (MAM), the definitions of the complexity measures of these machines, and the definition of a new type of “honest” functions. The lower bounds $\Omega(n^2)$ and $\Omega(n^{3/2})$ on the complexity measure $\text{TIME} \cdot \text{SPACE} \cdot \text{PARALLELISM}$ (TSP) are proved in Section 3. Further, the optimality of these lower bounds is investigated, and the hierarchy for TSP is established. The lower bound $\Omega(n^{3/2})$ is essentially improved in the case that $\text{SPACE} \geq n^\epsilon$, for $0 < \epsilon < 1$. In Section 4 the lower bound $\Omega(n^{1/2})$ on the complexity measure $\text{REVERSALS} \cdot \text{SPACE} \cdot \text{PARALLELISM}$ (RSP) is established, and similar questions as for the complexity measure TSP in Section 3

are studied for the complexity measure RSP. Because the multitape Turing machines are a very special case of our multihead alternating machine, using the extended parallel computation thesis [16] we obtain a lower bound $\Omega(n^a)$ for $a > 0$ on the complexity measure PARALLEL TIME · HARDWARE. Section 5 involves several lower bounds on the complexity measures of different versions of multihead finite automata that can be obtained as direct consequences of the assertions proved in Sections 3 and 4. Some motivations for further research in Section 6 conclude this paper.

2. Definitions

We shall consider, for any positive integer k , a k -head alternating machine, $AM(k)$, as a parallel computing model. An $AM(k)$ consists of a separate input tape with k two-way read-only heads, and a countable state control. The countable set of states of the $AM(k)$ is partitioned into two disjoint sets K_E (of existential states) and K_U (of universal states) with the same sense as in all alternating devices [7]. A step of $AM(k)$ M is made according to the state of M and the k symbols read by the k heads on the input tape. Using this information M can branch the computation into a finite number of computations and independently, for each branch of the computation, change the state and the positions of the heads by 1. We give only one restriction on M , namely that there must be a constant d_M such that branching from any universal state of M is bounded by d_M .

Clearly, the multihead alternating machines (MAMs) include a large number of different types of computing models. For example, a MAM is the generalisation of the multitape alternating Turing machine (ATM) in the following two directions.

(1) A MAM can have an arbitrary large number of heads on the input tape, ATM only one.

(2) A MAM has an arbitrary organisation of the memory (in fact, MAM can see the whole contents of its memory in each step of the computation).

Now, let us give the formal definition of the multihead alternating machines.

Definition 2.1. A k -head alternating machine $AM(k)$ is a 8-tuple $M = (K, \Sigma, K_U, \delta, q_0, F, d, k)$, where

- (1) K is the nonempty, countable set of states (internal configurations);
- (2) $q_0 \in K$ is the initial state;
- (3) $K_U \subseteq K$ is the set of universal states, $K_E = K - K_U$ is the set of existential states;
- (4) $F \subseteq k$ is the set of accepting states;
- (5) Σ is a finite, nonempty set called input alphabet, ϕ and $\$ \notin \Sigma$ are the endmarkers;
- (6) $\delta \subseteq (K^* \times (\Sigma \cup \{\phi, \$\})^k) \times (K \times \{-1, 0, 1\}^k)$ is the next-move relation, where $-1, +1$, and 0 denote the direction of the head move (left, right, stationary respectively); for $((q, (a_1, \dots, a_k)), (p, (\gamma_1, \gamma_2, \dots, \gamma_k))) \in \delta$ the following is required: if

$a_j \equiv \phi$ for some $j \in \{1, \dots, k\}$, then $\gamma_j \in \{0, 1\}$; if $a_i \equiv \$$ for some $i \in \{1, \dots, k\}$, then $\gamma_i \in \{-1, 0\}$;

(7) d is a positive integer such that, for $\forall q \in K_U, \forall x \in (\Sigma \cup \{\phi, \$\})^k$, there exist at most d different tuples (p, α) , where $p \in K, \alpha \in \{-1, 0, 1\}^k$, such that $((q, x), (p, \alpha)) \in \delta$.

Definition 2.2. A *descriptive configuration* of an $AM(k)$ machine $M = (K, \Sigma, K_U, \delta, q_0, F, d, k)$ is any element from

$$\Sigma^* \times K \times (\mathbb{N} \cup \{0\})^k,$$

where \mathbb{N} denotes the set of natural numbers.

Informally, a descriptive configuration $(w, q, (i_1, i_2, \dots, i_k))$ describes the situation in which the $AM(k)$ is in state q , has word w on the input tape, and the j th head is on the i_j th position of the input tape involving $\phi w \$$. Obviously, we assume that $0 \leq i_1, i_2, \dots, i_k \leq |w| + 1$, where $|w|$ denotes the length of the word w .

Definition 2.3. A *configuration* of an $AM(k)$ $M = (K, \Sigma, K_U, \delta, q_0, F, d, k)$ is an element from $K \times (\mathbb{N} \cup \{0\})^k$. For all $x \in \Sigma^*$, $I_M(x) = (x, q_0, (0, 0, \dots, 0))$ is the *initial descriptive configuration*. We shall say that the descriptive configuration $(x, q, (i_1, \dots, i_k))$ is *universal*, *existential*, and *accepting* respectively if q is a universal, existential, and accepting state respectively.

In what follows we define the notions “step” and “computation” of multihead alternating machines.

Definition 2.4. Let $M = (K, \Sigma, K_U, \delta, q_0, F, d, k)$ be an $AM(k)$. Let C and C' be two descriptive configurations. We shall say that M can go from C to C' in one step, $C \vdash C'$, if C' can be obtained from C by applying the next-move relation δ . A *sequential computation* of M on x is a sequence $C_0 = I_M(x) \vdash C_1 \vdash \dots \vdash C_m, m \geq 0$. In what follows we shall often write C_0, C_1, \dots, C_m only.

A *computation (computation tree* if we want to draw attention to the structure of the computation) of M on a word x is a finite, nonempty, labelled tree with the following properties:

- (1) each node v of the tree is labelled by a descriptive configuration $l(v)$;
- (2) if v is an internal node (a non-leaf) of the tree, $l(v)$ is universal, and $\{C \mid l(v) \vdash C\} = \{C_1, \dots, C_m\}$, then π has exactly k children u_1, \dots, u_m such that $l(u_i) = C_i$;
- (3) if v is an internal node of the tree and $l(v)$ is existential, then v has exactly one child u such that $l(v) \vdash l(u)$.

An *accepting computation (tree)* of M on an input word x is a computation (tree) whose root is labelled with $I_M(x)$ and whose leaves are all labelled with accepting descriptive configurations. We say that M *accepts* x if there is an accepting computation (tree) of M on input x . We define $L(M) = \{x \in \Sigma^* \mid M \text{ accepts } x\}$ as the *language accepted by* M .

In what follows we shall often consider the computation as a tree labelled by configurations instead of desriptional configurations. It will cause no confusion because it will be clear which input word is considered. For the recognition of different languages we shall define the notion "prominent configurations" according to the given language. If V is the set of prominent configurations, then we define, for each accepting computation, the *pattern of the accepting computation* as a tree U with the following properties:

- (1) the root of U is the root of D ;
- (2) the rest nodes are the nodes of D labelled by the prominent configurations from V ;
- (3) the nodes u and v are connected by an edge in U iff D involves a path from u to v that involves no node labelled by a prominent configuration.

Now, let us define the complexity measures for multihead alternating machines. Let A be an $AM(k)$ accepting a language $L(A)$.

The *space complexity* of A is a function of the input word length $S_A(n) = \log_2(C_A(n))$, where $C_A(n)$ is the number of all different states (internal configurations) used in all accepting computations on words from $L(A) \cap \Sigma^n$. We note that the number of all configurations used in accepting computations on inputs with length n can be at most $(n+2)^k C_A(n)$, where $(n+2)^k$ is the number of all different positions of the heads on the input tape.

For an accepting computation D of A we denote by $T_A(D)$ ($R_A(D)$) the maximum number of steps (head reversals) performed in the sequential computations from the root of D to the leaves of D . The *time* and *reversal complexity* measure respectively are defined in the obvious way as the following function: $X_A(n) = \max\{X_A(D) | D \text{ is an accepting computation on an input of the length } n\}$, where $X \in \{T, R\}$.

The parallel complexity measure is defined as introduced in [22] for alternating devices. The definitions of similar complexity measure called leaf-size can be found in [29]. Let $P_A(D)$ denote the number of universal states in the accepting computation D . Clearly, $P_A(D)$ is an upper bound on branchings in D . The *parallel complexity* of A is the function $P_A(n) = \max\{P_A(D) | D \text{ is an accepting computation on an input of the length } n\}$.

Let \mathcal{R} denote the set of all positive, real numbers. For arbitrary functions f and g from \mathbb{N} to \mathcal{R} , $f(n) \in \Omega(g(n))$ is equivalent to $\exists c \in \mathcal{R}, \exists m \in \mathbb{N}$, such that, for $\forall n \geq m, f(n) \geq cg(n)$, and $f(n) \in O(g(n))$ is equivalent to $\exists c \in \mathcal{R}, \exists m \in \mathbb{N}$ such that, for $\forall n \geq m, f(n) \leq cg(n)$. We shall write $f(n) = \Theta(g(n))$ iff $f(n) \in O(g(n))$ and $f(n) \in \Omega(g(n))$. The cardinality of a set K will be denoted by $|K|$. $\lfloor d \rfloor$ for a $d \in \mathbb{N}$ is the greatest natural number m such that $d \geq m$. If we shall write, for example, $\text{TIME} \cdot \text{SPACE} \cdot \text{PARALLELISM} \in \Omega(n^{3/2})$ in what follows, then it means that $T_A(n)S_A(n)P_A(n) \in \Omega(n^{3/2})$ for each device A of the computing model considered.

Now, giving some restrictions on MAMs we define multihead deterministic and nondeterministic machines.

Definition 2.5. Let $A = (K, \Sigma, K_U, \delta, q_0, F, d, k)$ be an $AM(k)$. We shall say that A is a k -head nondeterministic machine, $NM(k)$, if $K_U = \emptyset$. We shall say that A is a k -head deterministic machine, $DM(k)$, if the next-move relation is a function.

Different types of multihead finite automata can be defined by requiring the set of states of the multihead machine to be finite (and, maybe, some further properties). We omit these definitions because it is clear that they lead to the definitions currently used for multihead finite automata. For $Y \in \{1, 2\}$, $X \in \{D, N, A\}$, $k \in \mathbb{N}$, we denote by $YXFA(k)$ the Y -way k -head deterministic ($X = D$) (nondeterministic ($X = N$), alternating ($X = A$)) finite automaton.

For each type of device M (for example, $2AFA(k)$), we denote by $\mathcal{M}(M)$ the class of devices of type M , and by $\mathcal{L}(M) = \{L(A) \mid A \in \mathcal{M}(M)\}$ the family of languages accepted by devices of type M .

For each type of device M and functions t_1, t_2, t_3, t_4 from \mathbb{N} to \mathcal{R} , we define the types of devices $M-T(t_1)-S(t_2)-P(t_3)-R(t_4)$ and $M-TSPR(t_1)$, where $A \in \mathcal{M}(M-T(t_1)-S(t_2)-P(t_3)-R(t_4))$ iff $A \in \mathcal{M}(M)$ and $T_A(n) \in O(t_1(n))$, $S_A(n) \in O(t_2(n))$, $P_A(n) \in O(t_3(n))$, $R_A(n) \in O(t_4(n))$, and $B \in \mathcal{M}(M-TSPR(t_1))$ iff $B \in \mathcal{M}(M)$ and $T_B(n)S_B(n)P_B(n)R_B(n) \in O(t_1(n))$. If some of the symbols T, S, P, R are missing, it means that no upper bound is given for the corresponding complexity measure. So, a $2AFA(k)-T(f_1)-P(f_2)$ automaton B is a $2AFA(k)$ automaton with $T_B(n) \in O(f_1(n))$, $P_B(n) \in O(f_2(n))$.

Now, for each function z from \mathbb{N} to \mathcal{R} , $1 \leq z(n) \leq n$, we define the computability of z in an unusual way. This way shows to be more suitable for multihead devices because it does not require any additional space in which the value of $z(n)$ is coded.

Definition 2.6. Let M be a type of device introduced above (for example, $AM(3)-T(n^2)$), and let $1 \leq z(n) \leq n$ be a function from \mathbb{N} to \mathcal{R} . We shall say that z is M -computable if there is a machine $A \in \mathcal{M}(M)$ having a special accepting state s such that

- (1) $0^n \in L(A)$ for each $n \in \mathbb{N}$;
- (2) each accepting computation on 0^n has exactly one leaf labelled by a configuration with state s , and the first head is positioned on the $z(n)$ th square of the input tape in this configuration.

Finishing this section we define the languages studied in this paper. Let \oplus be the Boolean sum operator (sum mod 2).

$$L = \{w2^i w \mid w \in \{0, 1\}^*, i \geq 1\}, \quad L' = \{w2^m w \mid w \in \{0, 1\}^m, m \geq 1\},$$

$$S = \left\{ x_1 2^m x_2 2^m \dots 2^m x_r 2^z \mid x_i \in \{0, 1\}^m, \sum_{i=1}^r \oplus x_i = 0^m, m \geq 1, r \geq 1, z \geq 1 \right\},$$

where $\sum_{i=1}^r \oplus x_i = 0^m$ means that $x_{1j} \oplus x_{2j} \oplus \dots \oplus x_{rj} = 0$ for $j = 1, \dots, m$ and $x_i = x_{i1} x_{i2} \dots x_{im}$ for $i = 1, \dots, r$.

We note that a language similar to S was used in [12]. To obtain optimal lower bounds in this paper, we shall consider some special subsets of S defined in the following way. Let f and g be some functions from \mathbb{N} to \mathbb{N} such that $n - 2f(n)g(n) \geq 0$ for all $n \in \mathbb{N}$. Then,

$$S(f, g) = \left\{ x_1 2^{g(n)} x_2 2^{g(n)} \dots 2^{g(n)} x_{f(n)} 2^{g(n)+z(n)} \mid n \geq 1, x_i \in \{0, 1\}^{g(n)} \right. \\ \left. \text{for } i = 1, \dots, f(n), \sum_{i=1}^{f(n)} \oplus x_i = 0^{g(n)}, 0 \leq z(n) \leq n - 2f(n)g(n) \right\}.$$

3. TIME-SPACE-PARALLELISM tradeoff for language recognition

In this section we prove lower bounds $\text{TIME} \cdot \text{SPACE} \cdot \text{PARALLELISM} \in \Omega(n^2)$, for AM(1)s accepting L , and $\text{TIME} \cdot \text{SPACE} \cdot \text{PARALLELISM} \in \Omega(n^{3/2})$, for MAMs accepting S . We show that the lower bounds obtained are optimal in some sense which implies several hierarchy results for different complexity classes. Proving upper bounds for the recognition of some subsets of S we show, for linear time, that parallelism can compensate for a decrease of space complexity and vice versa. In the case that the inequality $\text{SPACE} \geq n^\varepsilon$ for $0 < \varepsilon < 1$ holds, we essentially improve the lower bound for the recognition of S .

First, we give results for AM(1)s.

Theorem 3.1. *Let A be an AM(1) machine such that $L' \subseteq L(A) \subseteq L$. Then*

$$T_A(n)S_A(n)P_A(n) \in \Omega(n^2).$$

Proof. To prove Theorem 3.1, we shall show that if an AM(1) machine A accepts all words in L' and does not accept any word in $\{0, 1, 2\}^* - L$, then $T_A(n)S_A(n)P_A(n) \in \Omega(n^2)$. We prove this by contradiction.

Let $A = (K, \Sigma, K_U, \delta, q_0, F, d, 1)$ be an AM(1) such that $L' \subseteq L(A) \subseteq L$ and $T_A(n)S_A(n)P_A(n) \notin \Omega(n^2)$, i.e., for $\forall a \in \mathbb{N}$, $\forall m \in \mathbb{N}$, $\exists s \geq m$ such that $T_A(s)S_A(s)P_A(s) < as^2$. In what follows we shall show that there is a word in $L(A) - L$, which will be the contradiction.

For each accepting computation D_w on the input $w2^m w \in L$, we define the *prominent configuration* as follows:

(1) The initial configuration is the prominent configuration.

(2) A configuration C in which the head is adjusted on the first or the last symbol of the subword 2^m is the prominent configuration iff the head crossed the whole subword 2^m in the sequential part of the computation D_w between the immediately preceding prominent configuration and C .

Using the notion of prominent configuration defined above we can, for any accepting computation D , construct the *pattern of D* in the way described in Section 2. Let $L_n = \{w2^m w \mid w \in \{0, 1\}^m, n = 3m\}$ for each $n \in \mathbb{N}$, $n = 3m$. Let us fix, for each word $x = w2^m w$ in $\bigcup_{n \in \mathbb{N}} L_n$, an accepting computation D_w . We define the *pattern of x* , \bar{D}_w , as the pattern of the fixed accepting computation D_w on x . Now, we shall prove the following fact.

Fact 3.1.1. For all $n = 3m$, $m \in \mathbb{N}$, the number of different patterns of words in L_n is bounded by

$$e(n) = 2^{3dT_A(n)(S_A(n)+1)P_A(n)/n}.$$

Proof. Each pattern can be transformed to a sequence containing the concatenation of all (at most $dP_A(n)$) paths from the root of the pattern to the leaves of the pattern. We note that having such a sequence of prominent configurations we can unambiguously construct the original pattern.

The length of each sequence corresponding to a pattern is bounded by $3dT_A(n)P_A(n)/n$ because A must make at least $m = \frac{1}{3}n$ steps in the part of each sequential computation between two prominent configurations. Since the number of all prominent configurations is bounded by $2 \cdot 2^{S_A(n)}$, the number of all patterns of words in L_n is bounded by

$$(2^{S_A(n)+1})^{3dT_A(n)P_A(n)/n} = 2^{3dT_A(n)(S_A(n)+1)P_A(n)/n}. \quad \square$$

Proof of Theorem 3.1 (continued). The number of words in L_n is $2^{n/3}$. Using the assumption that $T_A(n)S_A(n)P_A(n) \notin \Omega(n^2)$ we obtain that there exists a positive integer s such that

$$3dT_A(s)(S_A(s)+1)P_A(s) < \frac{1}{3}s.$$

It follows that there are two distinct words $w_1 2^{s/3} w_1$, and $w_2 2^{s/3} w_2$ in L_s having the same pattern \bar{D} .

Now, let us show that there is an accepting computation of A on the word $y = w_1 2^{s/3} w_2$ not in L , which proves Theorem 3.1.

The construction of an accepting computation (tree) on y is based on the fact that, during the computation on the words $w_1 2^{s/3} w_1$ and $w_2 2^{s/3} w_2$, A did not read the twins of subwords w_i in $w_i 2^{s/3} w_i$ at the same time because A has only one head on the input tape. Let us construct an accepting computation on y from the pattern \bar{D} in the following way. For each node u in the pattern \bar{D} , let X_u^1, X_u^2 be the subtrees of the accepting computations of D_{w_1}, D_{w_2} respectively from u (i.e., with the root u) to the prominent configurations in which an edge leads from u in D . Then, for every node u in \bar{D} , we replace the node u with the edges leading from u by one of the subtrees X_u^1, X_u^2 . The determination which of X_u^1, X_u^2 is chosen is given below.

If the head reads the subword $w_1 2^m$ of y between the prominent configuration u and the following ones, then X_u^1 is chosen. If the head reads the subword $2^m w_2$ of y between u and the following prominent configurations, then X_u^2 is chosen. Clearly, this completes the proof. \square

The direct consequence of Theorem 3.1 is the following result similar to Cobham's result in [9].

Corollary 3.2. Let A be an $NM(1)$ such that $L' \subseteq L(A) \subseteq L$. Then

$$T_A(n)S_A(n) \in \Omega(n^2).$$

Now, using a deterministic multitape Turing machine we show that the lower bounds obtained in Theorem 3.1 and Corollary 3.2 are optimal.

Theorem 3.3. *There is a multitape Turing machine B such that $L(B) = L$ and $T_B(n)S_B(n) \in O(n^2)$.*

Proof. Let B be a deterministic Turing machine with separate input tape and one working tape. Obviously, B is a special version of AM(1), where the internal configuration is the state of B and the contents of the working tape. B having an input word $x = w_1 2^i w_2$ on the separate input tape writes w_1 on the working tape, and compares the contents of the working tape with w_2 . Using its states B can find out the fact $x \notin \{0, 1\}^* \{2\}^* \{0, 1\}^*$ in real time. \square

Corollary 3.4. *Let g, f be functions from \mathbb{N} to \mathbb{N} fulfilling the following conditions:*

- (1) $1 \leq f(n) \leq n^2$;
- (2) f is AM(1)-T(n)-S(f) computable;
- (3) $g(n) \notin \Omega(f^2(n))$.

Then $\mathcal{L}(\text{AM}(1)\text{-TSP}(g(n))) \subsetneq \mathcal{L}(\text{AM}(1)\text{-TSP}(f(n)n))$.

Proof. Let us consider the language $L(f) = \{x = w 2^i w 2^j \mid w \in \{0, 1\}^*, |w 2^i w| = f(|x|)\}$. Following the proof of Theorem 3.1 we obtain that $L(f) \notin \mathcal{L}(\text{AM}(1)\text{-TSP}(g))$, where $g(n) \notin \Omega(f^2(n))$. The obvious fact that $L(f)$ can be recognized by an AM(1) in linear time and $O(f(n))$ space completes the proof. \square

Similarly as for Cobham's result, one can easily see that L can be recognized by a 1DFA(2) in real time, i.e., by an AM(2) M with $T_M(n)S_M(n)P_M(n) \in O(n)$. It implies an interesting fact claiming that one reading head on the separate input tape cannot be compensated by nondeterminism connected with $o(n)$ increase of the product of time, space and parallelism.

In what follows we shall consider the multihead alternating machine as a more general model of parallel computing and we shall prove nonlinear lower bounds on TSP for language recognition.

Theorem 3.5. *Let A be an MAM such that $S(f, g) \subseteq L(A) \subseteq S$ for functions $f(n) = \lfloor n^{1/2} \rfloor$ and $g(n) = \lfloor \frac{1}{2} n^{1/2} \rfloor$. Then*

- (a) $T_A(n)S_A(n)P_A(n) \in \Omega(n^{3/2}/\log_2 n)$;
- (b) *if $S_A(n) \geq \log_2 n$, then $T_A(n)S_A(n)P_A(n) \in \Omega(n^{3/2})$.*

Proof. We prove Theorem 3.5 by contradiction. Let, for a $k \in \mathbb{N}$, $A = (K, \Sigma, K_U, \delta, q_0, F, d, k)$ be an AM(k) such that $S(f, g) \subseteq L(A)$ and $T_A(n)S_A(n)P_A(n) \notin \Omega(n^{3/2}/\log_2 n)$ ($\Omega(n^{3/2})$ in the case $S_A(n) \geq \log_2 n$). We shall show that A accepts a word $y \notin S$ which proves Theorem 3.5.

Since $T_A(n)S_A(n)P_A(n) \notin \Omega(n^{3/2}/\log_2 n)$ ($\Omega(n^{3/2})$), there is a positive integer s with the property

$$(i) \quad 16k^3 dT_A(s)S_A(s)P_A(s) + 1 < sg(s)/\log_2 s \quad (sg(s)).$$

Let $S_s(f, g) = \{w \in S(f, g) \mid |w| = s\}$. In what follows we shall consider accepting computations on words in $S_s(f, g)$. Let $w = x_1 2^m x_2 2^m \dots 2^m x_r 2^{m+z} \in S_s(f, g)$. We shall say that A compares the pair of subwords (x_i, x_j) in a computation D_w on w iff there exists a configuration in D_w such that one of the k heads is positioned on x_i , and another head is positioned on x_j .

Let, for $v = 1, \dots, f(s)$, the v -prominent configuration of an accepting computation on a word in $S_s(f, g)$ be a configuration in which one of the heads is adjusted on the first or the last symbol of the subword x_v after crossing the whole subword $2^{g(n)}$. A prominent configuration is any v -prominent configuration, where v is in $\{1, 2, \dots, f(n)\}$.

Now, using property (i) of s we prove an important property of accepting computations on words in $S_s(f, g)$.

Fact 3.5.1. For each accepting computation D_w on $w \in S_s(f, g)$ there is a pair (i, j) , $1 \leq i < j \leq f(s)$, such that

- (1) the subwords x_i and x_j are not compared in D_w ;
- (2) D_w involves at most $4dkT_A(s)P_A(s)/s$ h -prominent configurations for $h \in \{i, j\}$.

Proof. In a sequential computation D_w can contain at most $kT_A(s)/g(s)$ prominent configurations. So, D_w involves at most $dkT_A(s)P_A(s)/g(s)$ prominent configurations which implies that there exist at least $\frac{1}{2}f(n)$ subwords x_h of w such that D_w involves at most $2dkT_A(s)P_A(s)/g(s)f(s)$ h -prominent configurations. The number of all pairs chosen from these $\frac{1}{2}f(s)$ subwords is $\binom{\frac{1}{2}f(s)}{2} \geq s/16$.

Considering the upper bound on the number of prominent configurations in D_w we obtain that at most $dk^3T_A(s)P_A(s)/g(s)$ pairs of subwords of w are compared in D_w . Property (i) of s implies $dk^3T_A(s)P_A(s)/g(s) < s/16 - 1$. So, we can find two words x_i and x_j among these $\frac{1}{2}f(s)$ subwords of w considered above such that x_i and x_j are not compared in D_w . This completes the proof of Fact 3.5.1. \square

Proof of Theorem 3.5 (continued). Let, for each $w \in S_s(f, g)$, D_w be a fixed accepting computation on w . The number of words in $S_s(f, g)$ is $2^{g(s)(f(s)-1)}$. Using Fact 3.5.1 we obtain that there exist positive integers h and r , $1 \leq h < r \leq f(s)$, such that the pair (h, r) fulfils conditions (1) and (2) of Fact 3.5.1 for at least $2^{g(s)(f(s)-1)}/f^2(n)$ accepting computations D_w on words w in $S_s(f, g)$.

In what follows, let the pattern of $w \in S_s(f, g)$ be the pattern \bar{D}_w of the fixed accepting computation D_w on w according to h -prominent and r -prominent configurations.

Fact 3.5.2. The number of all different patterns of the words in $S_s(f, g)$ is bounded by

$$e(s) = 2^{(S_A(s)+k \log_2 s)4kdT_A(s)P_A(s)/s}.$$

Proof. The number of all prominent configurations is bounded by

$$n^k 2^{S_A(s)} = 2^{S_A(s) + k \log_2 s}.$$

Now, following the proof of Fact 3.1.1, the proof of Fact 3.5.2 can be completed. \square

Proof of Theorem 3.5 (continued). Using property (i) of s we obtain $e(s)f^2(s) < 2^{g(s)} - 1$. It implies that there are two distinct words (for $m = g(s)$)

$$u = y_1 2^m y_2 \dots y_{h-1} 2^m x_h 2^m \dots y_{r-1} 2^m x_r 2^m y_{r+1} 2^m \dots 2^m y_{f(s)} 2^{m+z(s)},$$

$$u' = y_1 2^m y_2 \dots y_{h-1} 2^m x'_h 2^m \dots y_{r-1} 2^m x'_r 2^m y_{r+1} 2^m \dots 2^m y_{f(s)} 2^{m+z(s)}$$

in $S_s(f, g)$ having the following properties:

- (1) $x_h \neq x'_h$ and $x_r \neq x'_r$;
- (2) u and u' have the same pattern;
- (3) the pairs of subwords (x_h, x_r) and (x'_h, x'_r) are not compared in the accepting computations D_u and $D_{u'}$ respectively.

Now, we shall consider the word

$$y = y_1 2^m y_2 \dots y_{h-1} 2^m x_h 2^m y_{h+1} 2^m \dots y_{r-1} 2^m x'_r 2^m y_{r+1} 2^m \dots 2^m y_{f(s)} 2^{m+z(s)}$$

that does not belong to S . Realizing property (3), there is no doubt that the accepting computation on y can be constructed in the same way as in the proof of Theorem 3.1. This completes the proof of Theorem 3.5. \square

Since $NM(k)$ s are $AM(k)$ s without universal states, the following assertion follows directly from Theorem 3.5.

Corollary 3.6. *Let A be an $AN(k)$ for a $k \in \mathbb{N}$ such that $S(\lfloor n^{1/2} \rfloor, \lfloor n^{1/2}/2 \rfloor) \subseteq L(A) \subseteq S$. Then*

- (a) $T_A(n)S_A(n) \in \Omega(n^{3/2}/\log_2 n)$;
- (b) *If $S_A(n) \geq \log_2 n$, then $T_A(n)S_A(n) \in \Omega(n^{3/2})$.*

It can be simply seen that there is a deterministic multitape Turing machine recognizing $S(\lfloor n^{1/2} \rfloor, \lfloor \frac{1}{2} n^{1/2} \rfloor)$ in linear time and $O(n^{1/2})$ space. So we have tight lower and upper bounds for the recognition of this language. In the following assertion we shall show a more interesting upper bound for the recognition of $S(\lfloor n^{1/2} \rfloor, \lfloor \frac{1}{2} n^{1/2} \rfloor)$ that gives information about the relation between space complexity and parallel complexity.

Theorem 3.7. *Let h and q be functions from \mathbb{N} to \mathbb{N} satisfying the following conditions:*

- (1) $h(n)q(n) = \lfloor n^{1/2} \rfloor + z(n)$, where $0 \leq z(n) \leq q(n)$;
- (2) h and q are $AM(3)$ - $T(n)$ - $S(h)$ - $P(q)$ computable.

Then there is an $AM(6)$ C , recognizing $S(\lfloor n^{1/2} \rfloor, \lfloor \frac{1}{2} n^{1/2} \rfloor)$ with $T_C(h) \in O(n)$, $S_C(n) \in O(h(n))$ and $P_C(n) \in O(q(n))$.

Proof. We outline only the idea of the proof because using it there is a simple exercise to complete the proof. In the first part of the computation, C deterministically verifies whether the input word y has the following form $x_1 2^{m_1} x_2 2^{m_2} \dots x_{f-1} 2^{m_{f-1}} x_f 2^{m_f}$. In the second part of the computation, C computes h and q and checks whether $|x_1| = \lfloor \frac{1}{2} f \rfloor$ and $|x_1| = m_1 = |x_2| = m_2 = \dots = m_{f-1} = |x_f| \leq m_f$. In the third part, C gradually branches the computation in $q(|y|)$ parallel computations, and using $h(|y|)$ space C checks in the j th parallel computation whether $\sum_{i=1}^f x_{ji} = 0^{h(|y|)}$, where x_{ji} is the j th subword of x_i of the length $h(n)$. \square

Now, following the lower and upper bounds obtained above, we can formulate the following hierarchy results.

Corollary 3.8. *Let h, h', q and q' be increasing functions from \mathbb{N} to \mathbb{N} fulfilling the following conditions:*

- (1) h and q are MAM-T(n)-S(h)-P(q) computable;
- (2) $h(n)q(n) \leq \lfloor n^{1/2} \rfloor$;
- (3) $h(n) \geq h'(n) \geq \log_2 n$;
- (4) $h'(n)q(n) = o((h(n)q(n))^3/n)$ and $h(n)q'(n) = o((h(n)q(n))^3/n)$.

Then

$$\begin{aligned} \mathcal{L}(\text{MAM-T}(n)\text{-S}(h')\text{-P}(q)) &\subsetneq \mathcal{L}(\text{MAM-T}(n)\text{-S}(h)\text{-P}(q)), \\ \mathcal{L}(\text{MAM-T}(n)\text{-S}(h)\text{-P}(q')) &\subsetneq \mathcal{L}(\text{MAM-T}(n)\text{-S}(h)\text{-P}(q)), \\ \mathcal{L}(\text{MAM-T}(n)\text{-SP}(h(n)q'(n))) &\subsetneq \mathcal{L}(\text{MAM-T}(n)\text{-SP}(h(n)q(n))). \end{aligned}$$

Proof. Let us consider the language $S(f, g)$, where $2f(n) = 2h(n)q(n) = g(n)$. Following the proof of Theorem 3.5 we have $\text{TSP} \in \Omega((f(n)g(n))^{3/2}) = \Omega((h(n)q(n))^3)$. Now, following the proof of Theorem 3.8, we see that $S(f, g)$ separates all pairs of complexity classes compared in Corollary 3.9. \square

We note that one can obtain several other upper bounds that can imply many different hierarchies for distinct types of computing models. Concluding this section we prove a stronger lower bound on TSP for the MAMs using at least n^ε space for a number $\varepsilon \in \mathcal{R}$.

Theorem 3.9. *Let A be an MAM such that $S_A(n) \geq n^\varepsilon$ for a $\varepsilon \in \mathcal{R}$, $\varepsilon < 1$. Let $f_\varepsilon(n) = \lfloor n^{(1-\varepsilon)/2} \rfloor$ and $g_\varepsilon(n) = \lfloor \frac{1}{2} n^{(1+\varepsilon)/2} \rfloor$ be functions from \mathbb{N} to \mathbb{N} . Then $S(f_\varepsilon, g_\varepsilon) \subseteq L(A) \subseteq S$ implies*

$$T_A(n)S_A(n)P_A(n) \in \Omega(n^{(3+\varepsilon)/2}).$$

Proof. Since the proof is very similar to the proof of Theorem 3.5, we shall make a sketch only referring to the same procedures from the proof of Theorem 3.5.

Let $S(f_\varepsilon, g_\varepsilon) \subseteq L(A)$ and $T_A(n)S_A(n)P_A(n) \notin \Omega(n^{(3+\varepsilon)/2})$. Let v -prominent configurations, prominent configurations, $S_n(f_\varepsilon, g_\varepsilon)$ for each $n \in \mathbb{N}$, and the pattern of any word in $S(f_\varepsilon, g_\varepsilon)$ be defined as in the proof of Theorem 3.5.

Since, for a suitable s ,

$$\binom{\frac{1}{2}f_\varepsilon(s)}{2} \geq \frac{1}{8}s^{1-\varepsilon}$$

Fact 3.5.1 holds in the proof of Theorem 3.9, too. According to the fact that the number of different patterns of words in $S_s(f_\varepsilon, g_\varepsilon)$ is bounded by

$$e(s) = 2^{aT_A(s)S_A(s)P_A(s)/g_\varepsilon(s)f_\varepsilon(s)}$$

for a constant a , we obtain that the number of patterns of words in $S_s(f_\varepsilon, g_\varepsilon)$ is smaller than $2^{g_\varepsilon(s)/f_\varepsilon^2(s)}$. Now, the proof can be completed in the same way as the proof of Theorem 3.5. \square

Corollary 3.10. *Let A be an $NM(k)$ such that $L(A) = S$ and $S_A(n) \geq n^\varepsilon$ ($S_A(n) = \Theta(n^\varepsilon)$) for an $\varepsilon \in \mathcal{R}$, $0 < \varepsilon < 1$. Then*

$$T_A(n)S_A(n) \in \Omega(n^{(3+\varepsilon)/2}) \quad (T_A(n) \in \Omega(n^{(3-\varepsilon)/2})).$$

It is easy to see that the language $S(f_\varepsilon, g_\varepsilon)$ can be recognized in linear time, $O(n^{(1-\varepsilon)/2})$ parallelism, and $O(n^\varepsilon)$ space. So, the following result follows.

Corollary 3.11. *Let $\varepsilon', \varepsilon$ be positive rational numbers such that $0 \leq \varepsilon < \varepsilon' \leq 1$. Then*

$$\mathcal{L}(\text{MAM-T}(n)\text{-S}(n^\varepsilon)\text{-P}(n^{(1-\varepsilon')/2})) \subseteq \mathcal{L}(\text{MAM-T}(n)\text{-S}(n^\varepsilon)\text{-P}(n^{(1-\varepsilon)/2})).$$

Note that, by showing other upper bounds for the recognition of the languages $S(f_\varepsilon, g_\varepsilon)$, several further hierarchies can be established.

4. REVERSALS-SPACE-PARALLELISM tradeoff for language recognition

We prove a lower bound $\Omega(n^{1/2})$ on REVERSALS · SPACE · PARALLELISM for MAMs accepting S in this section. We establish a tight upper bound to this lower bound, and we improve the lower bound in the case that SPACE $\geq n^\varepsilon$ for a real number ε , $0 < \varepsilon < 1$. Considering the “extended parallel computation theses” of Dymond and Cook [16] we obtain, for a number $b \in \mathcal{R}$, an $\Omega(n^b)$ lower bound on HARDWARE · PARALLEL TIME of a very large class of parallel computing models.

Theorem 4.1. *Let f and g be functions from \mathbb{N} to \mathbb{N} such that $f(n) = \lfloor n^{1/2} \rfloor$ and $g(n) = \lfloor \frac{1}{2}n^{1/2} \rfloor$. Let A be an MAM fulfilling $S(f, g) \subseteq L(A) \subseteq S$. Then*

- (a) $T_A(n)S_A(n)P_A(n) \in \Omega(n^{1/2}/\log_2 n)$;
- (b) if $S_A(n) \geq \log_2 n$, then $T_A(n)S_A(n)P_A(n) \in \Omega(n^{1/2})$.

Proof. We prove Theorem 4.1 by contradiction. Let, for some $k \in \mathbb{N}$, $A = (K, \Sigma, K_U, \delta, q_0, F, d, k)$ be an $AM(k)$ such that $S(f, g) \subseteq L(A)$ and $R_A(n)S_A(n)P_A(n) \notin \Omega(n^{1/2}/\log_2 n)$ ($\Omega(n^{1/2})$ in the case $S_A(n) \geq \log_2 n$).

Analogously to the proofs of Theorem 3.1 and 3.5, we shall show that A accepts a word $y \notin S$ which proves Theorem 4.1.

Since $R_A(n)S_A(n)P_A(n) \notin \Omega(n^{1/2}/\log_2 n)$ ($\Omega(n^{1/2})$ if $S_A(n) \geq \log_2 n$), there is a positive number s with the properties

$$(ii) \quad dk^3 P_A(s)R_A(s) + 1 < \lfloor s^{1/2} \rfloor = f(s),$$

$$(iii) \quad (k \log_2(s+2) + S_A(s))4kdR_A(s)P_A(s) + 1 < \lfloor \frac{1}{2}s^{1/2} \rfloor = g(s).$$

Let us consider the accepting computations on the words $w = x_1 2^m x_2 2^m \dots 2^m x_r 2^{m+z} \in S_s(f, g)$. Let, for all $i, j = 1, 2, \dots, r$ ($r = f(s)$), the comparison of a pair of subwords (x_i, x_j) in a computation D_w on w , and an i -prominent configuration of an accepting computation on w be defined as in the proof of Theorem 3.5. Now, using property (ii) of s we prove the following fact.

Fact 4.1.1. *Let D be an accepting computation on a word $w \in S_s(f, g)$. Then there exist positive integers $i, j \in \{1, \dots, f(s)\}$ such that x_i and x_j are not compared in D .*

Proof. A pair of heads can compare at most $kf(s)$ pairs of subwords of w in any part of a sequential computation without reversals. So, k heads can compare at most $\binom{k}{2}kR_A(s)f(s) \leq k^3R_A(s)f(s)$ in any sequential computation from the root of D to a leaf of D . Since the number of leaves is bounded by $dP_A(s)$, we obtain that there are at most $dk^3R_A(s)P_A(s)f(s)$ pairs of subwords (x_h, x_v) compared in D . Property (ii) of s and the fact that the number of pairs (x_h, x_v) is $\binom{f(s)}{2} \geq f^2(s)/16$ completes the proof of Fact 4.1.1. \square

Proof of Theorem 4.1 (continued). Let, for each $w \in S_s(f, g)$, D_w be a fixed accepting computation on w . The number of words in $S_s(f, g)$ is $2^{g(s)(f(s)-1)}$. Using Fact 4.4.1 we obtain that there exist positive integers a and b , $1 \leq a < b \leq f(s)$, such that the subwords x_a and x_b are not compared in at least

$$2^{g(s)(f(s)-1)}/f^2(s)$$

accepting computations on different words in $S_s(f, g)$. Let, for each $w \in S_s(f, g)$, the pattern of w be the pattern \bar{D}_w of the fixed accepting computation D_w on w according to a -prominent and b -prominent configurations.

Fact 4.1.2. *The number of all different patterns of words in $S_s(f, g)$ is bounded by*

$$e(s) = 2^{(k \log_2(s+2) + S_A(s))4kdR_A(s)P_A(s)}.$$

Proof. Similarly as in the proof of Fact 3.1.1, any pattern \bar{D}_w can be viewed as a sequence S_w of prominent configurations that is the concatenation of at most $dP_A(s)$ paths leading from the root of \bar{D}_w to the leaves of \bar{D}_w .

Since each part of any sequential computation without any reversal can contain at most $4k$ prominent configurations, each S_w has length at most $4kdR_A(s)P_A(s)$. Realizing that the number of different configurations in the computations on words in $S_s(f, g)$ is bounded by

$$(s+2)^k C_A(s) = 2^{k \log_2(s+2) + S_A(s)},$$

the proof is finished. \square

Proof of Theorem 4.1 (continued). Using property (iii) of s we obtain $e(s)f^2(s) < 2^{g(s)} - 1$. Now, the proof can be completed in the same way as in Theorem 3.5. \square

Corollary 4.2. *Let A be an $NM(k)$, for a $k \in \mathbb{N}$ such that $S(\lfloor n^{1/2} \rfloor, \lfloor \frac{1}{2}n^{1/2} \rfloor) \subseteq L(A) \subseteq S$. Then*

- (a) $R_A(n)S_A(n) \in \Omega(n^{1/2}/\log_2 n)$;
- (b) if $S_A(n) \geq \log_2 n$, then $R_A(n)S_A(n) \in \Omega(n^{1/2})$.

As it was already noted in Section 3 there is a deterministic multitape Turing machine recognizing $S(\lfloor n^{1/2} \rfloor, \lfloor \frac{1}{2}n^{1/2} \rfloor)$ in linear time and $O(n^{1/2})$ product of space and parallelism. We note that this Turing machine can work without using any reversal. So, using this upper bound or proving other upper bounds one can establish several hierarchy results whose formulation is omitted.

Dymond and Cook [16] state the extended parallel computation thesis, claiming that space and the number of reversals of sequential computations (deterministic multitape Turing machines) are simultaneously polynomially related to the requirements on time and hardware of parallel computing models (for example, of parallel RAMs). Using this thesis we obtain the following result.

Theorem 4.3. *For each parallel machine class fulfilling the extended parallel computation thesis, there is a constant b such that*

$$\text{PARALLEL TIME} \cdot \text{HARDWARE} \in \Omega(n^b)$$

for the recognition of the language S .

Concluding this section we prove a stronger lower bound on RSP of the MAMs using at least n^ε space for a number $\varepsilon \in \mathcal{R}$.

Theorem 4.4. *Let, for an $\varepsilon, 0 < \varepsilon < 1$, f_ε and g_ε be functions from \mathbb{N} to \mathbb{N} such that $f_\varepsilon(n) = \lfloor n^{(1-\varepsilon)/2} \rfloor$ and $g_\varepsilon(n) = \lfloor n^{(1+\varepsilon)/2} \rfloor$. Let A be an MAM fulfilling $S(f_\varepsilon, g_\varepsilon) \subseteq L(A) \subseteq S$ and $S_A(n) \geq n^\varepsilon$. Then*

$$R_A(n)S_A(n)P_A(n) \in \Omega(n^{(1+\varepsilon)/2}).$$

Proof. We prove this result by contradiction. Let, for a $k \in \mathbb{N}$, $A = (K, \Sigma, K_U, \delta, q_0, F, d, k)$ be an $AM(k)$ such that $S(f_\varepsilon, g_\varepsilon) \subseteq L(A)$ and $R_A(n)S_A(n)P_A(n) \notin \Omega(n^{(1+\varepsilon)/2})$. Following the proof of Theorem 4.1 we shall show that A accepts a word $y \notin S$.

Since $R_A(n)S_A(n)P_A(n) \notin \Omega(n^{(1+\varepsilon)/2})$, there is a positive integer s such that

$$(iv) \quad 64k^3 d S_A(s) R_A(s) P_A(s) + 1 < \lfloor \frac{1}{2} s^{(1+\varepsilon)/2} \rfloor = g_\varepsilon(s)$$

$$(v) \quad s^\varepsilon \geq k \log_2(s+2)$$

hold. Since $S_A(s) \geq s^\varepsilon$, inequality (iv) implies

$$(vi) \quad 32k^3 d R_A(s) P_A(s) < \lfloor s^{(1+\varepsilon)/2} / s^\varepsilon \rfloor = \lfloor s^{(1-\varepsilon)/2} \rfloor = f_\varepsilon(s).$$

Using (vi) we obtain that Fact 4.1.1 holds in this proof, too. Following the proof of Fact 4.1.2 and (v) we have that the number of all different patterns of words in

$S_\varepsilon(f_\varepsilon, g_\varepsilon)$ is bounded by

$$e(s) = 2^{8kdS_A(s)R_A(s)P_A(s)}.$$

Using (iv) one can simply prove $e(s)f^2(s) < 2^{g_\varepsilon(s)} - 1$. Now, the proof can be completed in the same way as in Theorems 3.5 and 4.1. \square

Corollary 4.5. *Let, for a real number ε , $0 < \varepsilon < 1$, f_ε and g_ε be functions from \mathbb{N} to \mathbb{N} such that $f_\varepsilon(n) = \lfloor n^{(1-\varepsilon)/2} \rfloor$ and $g_\varepsilon(n) = \lfloor n^{(1+\varepsilon)/2} \rfloor$. Let A be an MAM fulfilling $S(f_\varepsilon, g_\varepsilon) \subseteq L(A) \subseteq S$ and $S_A(n) \geq n^\varepsilon$ ($S_A(n) = \Theta(n^\varepsilon)$). Then*

$$R_A(n)S_A(n) \in \Omega(n^{(1+\varepsilon)/2}) \quad (R_A(n) \in \Omega(n^{(1-\varepsilon)/2})).$$

5. Lower bounds for multihead finite automata

Multihead finite automata are computation devices which have no additional working space (i.e., they are multihead machines with finite state control). There were several reasons for the extensive study of them (see, for example, [4, 5, 8, 11, 15, 18, 19, 21–23, 25–31, 38, 39, 43, 46–50, 54]). One of the most important properties of two-way multihead finite automata according to complexity theory is that they characterise the basic complexity classes—deterministic and nondeterministic logarithmic space [54], and polynomial deterministic time [7].

We shall give several nontrivial lower bounds for multihead automata that are immediate consequences of the results obtained in Sections 3 and 4. First, using Theorem 3.5 we obtain the first lower bound for the complexity measure TIME · PARALLELISM of two-way alternating multihead finite automata.

Theorem 5.1. *Let A be a 2AFA(k) for a $k \in \mathbb{N}$ such that $S(\lfloor n^{1/2} \rfloor, \lfloor \frac{1}{2}n^{1/2} \rfloor) \subseteq L(A) \subseteq S$. Then*

$$T_A(n)P_A(n) \in \Omega(n^{3/2}/\log_2 n).$$

Corollary 5.2. *Let A be a 2NFA(k) for a $k \in \mathbb{N}$ such that $S(\lfloor n^{1/2} \rfloor, \lfloor \frac{1}{2}n^{1/2} \rfloor) \subseteq L(A) \subseteq S$. Then*

$$T_A(n) \in \Omega(n^{3/2}/\log_2 n).$$

We note that the result formulated in Corollary 5.2 was already established in [12]. We draw attention to the fact that one can easily construct a two-way deterministic multihead finite automaton recognizing $S(\lfloor n^{1/2} \rfloor, \lfloor \frac{1}{2}n^{1/2} \rfloor)$ in $O(n^{3/2})$ time, which shows that the lower bounds obtained in Theorem 5.1 and Corollary 5.2 are nearly optimal.

Now, using Theorem 5.1 we give the strongest lower bound for the complexity measure REVERSALS · PARALLELISM of two-way alternating multihead finite automata. The lower bound $RP \in \Omega(n^{1/3}/\log_2 n)$ was established in [24] for a language different from S .

Theorem 5.3. *Let A be a 2AFA(k) for a $k \in \mathbb{N}$ such that $S(\lfloor n^{1/2} \rfloor, \lfloor \frac{1}{2}n^{1/2} \rfloor) \subseteq L(A) \subseteq S$. Then*

$$T_A(n)P_A(n) \in \Omega(n^{1/2}/\log_2 n).$$

Corollary 5.4. *Let A be a 1AFA(k) for a $k \in \mathbb{N}$ such that $S(\lfloor n^{1/2} \rfloor, \lfloor \frac{1}{2}n^{1/2} \rfloor) \subseteq L(A) \subseteq S$. Then $P_A(n) \in \Omega(n^{1/2}/\log_2 n)$.*

Corollary 5.5. *Let A be a 2NFA(k) for a $k \in \mathbb{N}$ such that $S(\lfloor n^{1/2} \rfloor, \lfloor \frac{1}{2}n^{1/2} \rfloor) \subseteq L(A) \subseteq S$. Then $R_A(n) \in \Omega(n^{1/2}/\log_2 n)$.*

Corollary 5.6. $S \notin \mathcal{L}(1NFA(k))$ for any $k \in \mathbb{N}$.

We note that, for the language of reversals L_R introduced in [47], the stronger (than in Corollary 5.4) lower bound $P_A(n) \in \Omega((n/\log_2 n)^{1/2})$ was achieved in [22]. The strongest lower bound known until now on the reversal complexity of two-way nondeterministic multihead finite automata, $R_A(n) \in \Omega(n^b)$ for $0 < b < \frac{1}{3}$, was established for the language $(L_R)^*$ in [21]. The first lower bounds on reversal complexity measures were established in [49], where languages having nonconstant reversal complexity are presented.

It can be simply seen that there are a two-way deterministic multihead finite automaton B recognizing $S(\lfloor n^{1/2} \rfloor, \lfloor \frac{1}{2}n^{1/2} \rfloor)$ with $R_B \in O(n^{1/2})$, and a one-way alternating multihead finite automaton C recognizing $S(\lfloor n^{1/2} \rfloor, \lfloor \frac{1}{2}n^{1/2} \rfloor)$ with $P_C(n) \in O(n^{1/2})$. So, the lower bounds introduced in Theorem 5.3 and Corollaries 5.4 and 5.5 are tight to the upper bounds for the recognition of $S(\lfloor n^{1/2} \rfloor, \lfloor \frac{1}{2}n^{1/2} \rfloor)$. We note that, for the lower bounds established in [12, 21, 22], no tight upper bounds are known.

6. Conclusion

The main results of this paper are the lower bounds on different complexity measure of multihead machines. An important fact is that we are able to give tight upper bounds to these lower bounds. The following observation is interesting, too. The proof technique developed in this paper cannot be used to obtain lower bounds greater than $\Omega(n^2)$. So, concluding this paper we give some motivations for further research.

To determine the significance of a lower bound established, the following two questions should be answered:

- (1) How high is the lower bound according to the known lower bounds?
- (2) How general is the computing model considered?

These two questions imply possible directions of further research. First, the effort can be made to obtain higher lower bounds on the complexity measures of multihead machines than the lower bounds introduced in this paper. Second, allowing the heads on the input tape of MAMs to jump, we obtain the most general model of

alternating computations. Our proof technique is not suitable for MAMs with jumping heads. The language S can be recognized in linear time and logarithmic space by a deterministic two-head machine with jumping heads. It implies that jumping heads cannot be compensated by $o(n^{1/2}/\log_2 n)$ increase of the product of time, space and parallelism. So, to prove a nontrivial lower bound for language recognition on multihead machines with jumping heads is of great importance. We note that the lower bound of Borodin and Cook [3] obtained for sorting on deterministic multihead machines with jumping heads is the only nontrivial lower bound obtained for the general model of sequential computations.

References

- [1] A.E. Andreev, On a method of obtaining lower bounds to the complexity of individual monotone functions, *Dokl. Akad. Nauk SSSR* **282**(5) (1985) 1033–1037 (in Russian).
- [2] N.A. Blum, Boolean functions requiring $3n$ network size, *Theoret. Comput. Sci.* **28** (1984) 337–345.
- [3] A.B. Borodin and S.A. Cook, A time-space tradeoff for sorting on a general model of computation, in: *Proc. 12th ACM STOC* (1980) 294–301.
- [4] F.J. Brandenburg, Three write heads are as good as k , *Math. Systems Theory* **14** (1981) 1–12.
- [5] F.J. Brandenburg, On one-way auxiliary pushdown automata, in: *Lecture Notes in Computer Science* **48** (Springer, Berlin, 1979) 132–144.
- [6] T. Chan, Reversal complexity of counter machines, in: *Proc. 13th Ann. ACM STOC* (1981) 147–157.
- [7] A.K. Chandra, D.C. Kozen and L.J. Stockmeyer, Alternation, *J. ACM* **28** (1981) 114–133.
- [8] M. Chrobak, Hierarchies of one-way multihead automata languages, in: *Proc. 12th ICALP '85, Lecture Notes in Computer Science* **194** (Springer, Berlin, 1985) 101–110.
- [9] A. Cobham, The recognition problem for perfect squares, in: *Proc. 7th IEEE Symp. on SWAT* (1966) 78–87.
- [10] S.A. Cook, An overview of computational complexity, *Comm. ACM* **26** (1983) 400–408.
- [11] P. Ďuriš and Z. Galil, Fooling a two-way automaton or one pushdown store is better than one counter for two-way machines, *Theoret. Comput. Sci.* **21** (1982) 39–53.
- [12] P. Ďuriš and Z. Galil, A time-space tradeoff for language recognition, *Math. Systems Theory* **17** (1984) 3–12.
- [13] P. Ďuriš and Z. Galil, On reversal-bounded counter machines and on pushdown automata with a bound on the pushdown store, *Inform. and Control* **54** (1982) 217–227.
- [14] P. Ďuriš, Z. Galil, W. Paul and R. Reischik, Two nonlinear lower bounds, in: *Proc. 15th ACM STOC* (1983) 127–132.
- [15] P. Ďuriš and J. Hromkovič, One-way simple multihead finite automata are not closed under concatenation, *Theoret. Comput. Sci.* **27** (1983) 121–125.
- [16] P.W. Dymond and S.A. Cook, Hardware complexity and parallel computation, in: *Proc. 21st Ann. IEEE FOCS* (1980) 360–372.
- [17] R. Freivalds, Relation between probabilistic and nondeterministic Turing machines, Unpublished communication presented at the 11th MFCS '84, Prague.
- [18] J.E. Hopcroft and J.D. Ullman, *Formal Languages and their Relation to Automata* (Addison-Wesley, Reading, MA, 1969).
- [19] J. Hromkovič, One-way multihead deterministic finite automata, *Acta Inform.* **19** (1983) 377–384.
- [20] J. Hromkovič, Hierarchy of reversal and zero-testing bounded multicounter machines, in: *Proc. 11th MFCS '84, Lecture Notes in Computer Science* **176** (Springer, Berlin, 1984) 312–321.
- [21] J. Hromkovič, Fooling a two-way nondeterministic multihead automaton with reversal number restriction, *Acta Inform.* **22** (1985) 589–594.
- [22] J. Hromkovič, On the power of alternation in automata theory, *J. Comput. System Sci.* **31** (1985) 28–39.
- [23] J. Hromkovič, On one-way two-head deterministic finite state automata, *Computers AI* **4** (1985) 503–526.

- [24] J. Hromkovič, Reversals-Space-Parallelism tradeoff for language recognition, Unpublished manuscript, 1985.
- [25] O.H. Ibarra, Characterizations of some tape and time complexity classes of Turing machines in terms of multihead and auxiliary stack automata, *J. Comput. System Sci.* **5** (1971) 81-117.
- [26] O.H. Ibarra, On two-way multihead automata, *J. Comput. System Sci.* **7** (1973) 28-37.
- [27] O.H. Ibarra and C.E. Kim, On 3-head versus 2-head finite automata, *Acta Inform.* **4** (1975) 173-200.
- [28] K. Inoue, I. Takanami, A. Nakamura and T. Ae, One-way simple multihead finite automata, *Theoret. Comput. Sci.* **9** (1979) 311-328.
- [29] K. Inoue, H. Matsuno, H. Taniguchi and I. Takanami, Alternating simple multihead finite automata, *Theoret. Comput. Sci.* **36** (1985) 291-308.
- [30] L. Janiga, Real-time computations of two-way multihead finite automata, in: L. Budach, ed., *Proc. FCT '79* (Akademie Verlag, Berlin 1979) 214-219.
- [31] K.N. King, Alternating multihead finite automata, in: *Proc. 8th ICALP '81*. Lecture Notes in Computer Science **115** (Springer, Berlin, 1981) 506-520.
- [32] O.B. Lupanov, Ob odnom metode sinteza schem, *Izv. Vuzov Radicfiz.* **1** (1958) (in Russian) 120-140.
- [33] O.B. Lupanov, O sinteze nekatorych klassov upravljajuščich sistem, *Problemy Kibernetiki* **10** (1963) (in Russian) 63-97.
- [34] W. Maas, Quadratic lower bounds for deterministic and nondeterministic one-tape Turing machines, in: *Proc. 16th Ann. ACM STOC* (1984) 401-408.
- [35] Li Ming, On one tape versus two stacks, Tech. Rept. 84-591, Dept. of Computer Science, Cornell University, Ithaca, New York, January 1984.
- [36] Li Ming, Simulating two pushdowns by one tape in $n^{1.5}(\log_2 n)^{0.5}$ time, in: *Proc. 26th Ann. IEEE FOCS* (1985) 56-64.
- [37] Li Ming and P.M.B. Vitányi, Tape versus queue and stack: The lower bounds, Centrum voor Wiskunde en Informatica, Rept. CS-R8519, September 1985.
- [38] B. Monien, Two-way multihead automata over a one-letter alphabet, *RAIRO Inform. Théor.* **14** (1980) 67-82.
- [39] B. Monien and I.H. Sudborough, The interface between language theory and complexity theory, in: R.V. Book, ed., *Formal Language Theory, Perspectives and Open Problems* (Academic Press, New York, 1981) 287-323.
- [40] E.J. Nečiporuk, A Boolean function, *Dokl. Akad. Nauk SSSR* **169** (1966) 765-766.
- [41] R.G. Nigmatulin, *The Complexity of Boolean Functions*, (Kazan State University Press, 1983) (in Russian).
- [42] W.J. Paul, On time hierarchies, *J. Comput. System Sci.* **19** (1979) 197-202.
- [43] T.F. Piatkowski, *N-head finite-state machines*, Ph.D. Dissertation, University of Michigan, 1963.
- [44] P. Pudlák, A lower bound on complexity of branching programs, in: *Proc. 11th MFCS '84*, Lecture Notes in Computer Science **176** (Springer, Berlin, 1984) 480-490.
- [45] A.A. Razborov, Lower bounds on the monotone complexity of some Boolean functions, *Soviet Math. Doklady* **281** (1985) 798-801.
- [46] R.L. Rivest and A.C. Yao, $k+1$ heads are better than k , *J. ACM* **25** (1978) 337-340.
- [47] A.L. Rosenberg, On multihead finite automata, *IBM J. Res. Develop.* **10** (1966) 388-394.
- [48] W.J. Sakoda and M. Sipser, Nondeterminism and the size of two-way finite automata, in: *Proc. 10th Ann. ACM STOC* (1978) 275-286.
- [49] I.H. Sudborough, Bounded-reversal multihead finite automata languages, *Inform. and Control* **25** (1974) 317-328.
- [50] I.H. Sudborough, One-way multihead writing finite automata, *Inform. and Control* **39** (1976) 1-20.
- [51] P.M.B. Vitányi, The simple roots of real-time computation hierarchies, in: *Proc. 11th ICALP '84*, Lecture Notes in Computer Science **172** (Springer, Berlin, 1984) 486-489.
- [52] P.M.B. Vitányi, Square time is optimal for simulation of one pushdown store or one queue by an oblivious one-head tape unit, *Inform. Process. Lett.* **21** (1985) 87-91.
- [53] P.M.B. Vitányi, An $n^{1.618}$ lower bound on the time to simulate one queue or two pushdown stores by one tape, *Inform. Process. Lett.* **21** (1985) 147-152.
- [54] K. Wagner and G. Wechsung, *Computational Complexity*, Mathematische Monographien (VEB Deutscher Verlag der Wissenschaften, Berlin 1986).