Tally Languages Accepted by Alternating Multitape Finite Automata

Dainis Geidmanis[1], Jānis Kaņeps[1], Kalvis Apsītis[2], Daina Taimiņa[3] and Elena Calude[4]

[1] Institute of Mathematics and Computer Science, University of Latvia, Raina bulv. 29, Riga, Latvia, dainisg@cclu.lv, janisk@konts.lv ‡

[2] Department of Computer Science University of Maryland College Park, MD 20742, U.S.A; kalvis@cs.umd.edu §

[3] Department of Mathematics Cornell University Ithaca, NY 14853, U.S.A.; dtaimina@math.cornell.edu ¶

[4] Computer Science Department University of Auckland Private Bag 92109 Auckland, New Zealand; elena@cs.auckland.ac.nz ‖

Abstract. We consider k-tape 1-way alternating finite automata (k-tape 1afa). We say that an alternating automaton accepts a language $L \subseteq (\sum^*)^k$ with $f(n)$-bounded maximal (respectively, minimal) leaf-size if arbitrary (respectively, at least one) accepting tree for any $(w_1, w_2, ..., w_k) \in L$ has no more than $f(\max_{1 \le i \le k} |w_i|)$ leaves. The main results of the paper are the following. If k-tape 1afa accepts language L over one-letter alphabet with $o(\log n)$-bounded maximal leaf-size or $o(\log \log n)$-bounded minimal leaf-size then the language L is semilinear. Moreover, if a language L is accepted with $o(\log \log(n))$-bounded minimal (respectively, $o(\log(n))$-bounded maximal) leaf-size then it is accepted by constant-bounded minimal (respectively, maximal) leaf-size by the same automaton. To show that this bound is optimal we prove that 4-tape 1afa can accept a non-semilinear languages over one-letter alphabet with $O(\log \log n)$-bounded minimal leaf-size. For maximal leaf-size our bound is optimal due to King's results.

1 Introduction

In this paper we consider languages over one-letter input alphabet accepted by k-tape 1-way alternating finite automata (k-tape 1afa).

There are many reasons why to investigate languages over one-letter alphabet separately, specially accepted by k-tape and k-head alternating automata. The

‡ Research supported by Grant No.96.0282 from the Latvian Council of Science

§ Supported by NSF Grant 9301339.

¶ Supported by NSF Grant 9501712

‖ Supported by Auckland University Research Grants A18/XXXXX/62090/3414012, A18/XXXXX/62090/F3414022

alternation adds very much computational power in contrast to nondeterministic automata investigated by Ibarra [Ib78] has shown that k-tape languages over one letter alphabet accepted by 2-way nondeterministic automata with restricted number of reversals of input heads and with finite number reversal restricted counters and unrestricted pushdown are semilinear languages. The languages accepted by such automata are very close to problems concerning vector addition systems with states [HRHY86].

King [Ki81] has shown that non-regular languages can be accepted already by 2-head 1afa.

In [Ge88] it was shown that the class of languages accepted by k-head 1afa is very complicated and for $k > 2$, the emptiness problem is undecidable. In [Pe 95] the result has been improved by proving undecidability for $k = 2$.

There are many examples showing that capabilities of automata in one letter alphabet differ from one in multi-letter alphabet. Therefore it also is of interest to investigate k-tape one-letter languages.

One of the most interesting characteristics of computation by alternating automata is the leaf-size [Ki81]. Leaf-size, in a sense, reflects the number of processors which run in parallel when reading a given input. Many results regarding hierarchies of languages over multi-letter alphabet accepted by different leaf-size bound have been got. There are two definitions of leaf-size bound — *minimal* and *maximal*. See more formal definition below.

Hromkovič [Hr85] has shown that there is a language that can be accepted by 3-head 1afa with $n^{1/2}$-bounded leaf-size and can not be accepted with any k-head 1afa with $o(n/\log(n))$ bounded leaf-size ($k \geq 1$). Minimal leaf-size bound is used.

Geidmanis [Ge88] has proved a hierarchy for k-tape languages accepted by constant-bounded leaf-size ($k \geq 2$). Minimal leaf-size bound is used.

Nasyrov [Na89] has proved a hierarchy for 1-tape 1-way alternating pushdown automata by constant-bounded leaf-size and hierarchy for $f(n)$-bounded leaf-size for some functions $f(n)$ in the gap between $\log(n)$ and $n/\log(n)$. Minimal leaf-size bound is used.

In [ITH89] a hierarchy by leaf-size is proved for languages accepted by 2-dimensional automata with small space and leaf-size smaller than $\log(n)$. Maximal leaf-size bound is used.

In this paper we show that for languages over one-letter alphabet maximal and minimal leaf-size bounds for the same language can differ exponentially.

Some further results demonstrate that these bounds are optimal.

2 Definitions

One-way finite k-tape automaton has a finite set of states and k read-only tapes and k input heads which can never shift left (one head on each of tapes). On each tape is written a word in input alphabet Σ and on the right of each word is written the end marker $\# \notin \Sigma$. Languages accepted by these automata are subsets $L \subseteq (\Sigma^*)^k$.

We consider one-letter case $\Sigma = \{0\}$ only. A language $L \subseteq (\{0\}^*)^k$ is called semilinear if the set $\{(n_1, \ldots, n_k) | (0^{n_1}, \ldots, 0^{n_k}) \in L\}$ is semilinear subset of N^k

It is known that semilinear one-letter languages can be accepted by nondeterministic one-way finite multitape automata. Alternating one-way finite multitape automata can accept a non-semilinear language $\{(0^{2^n}, 0^{2^n})\}$ (it follows from [Ki81a]).

Let $N = \{0, 1, 2, \ldots\}$ the set of natural numbers, and $[0, \infty)$ be the set of all nonnegative real numbers.

Definition 1. Let $\mathcal{A}$ be an alternating automaton and $f : N \to [0, \infty)$ be a function. We say that $\mathcal{A}$ accepts a language $L \subseteq (\sum^*)^k$ with $f(n)$-bounded minimal leaf-size if

1. there is no accepting tree for any input vector of words $(w_1, w_2, ..., w_k) \notin L$,
2. there is an accepting tree for each $(w_1, w_2, ..., w_k) \in L$, and
3. at least one accepting tree for each $(w_1, w_2, ..., w_k) \in L$ has no more than $f(\max_{1 \le i \le k} |w_i|)$ leaves.

Let $[0, \infty] = [0, \infty) \cup \{\infty\}$.

Definition 2. Let $\mathcal{A}$ be an alternating automaton and $f : N \to [0, \infty]$ be a function. We say that $\mathcal{A}$ accepts a language $L \subseteq (\sum^*)^k$ with $f(n)$-bounded maximal leaf-size if

1. there is no accepting tree for each $(w_1, w_2, ..., w_k) \notin L$,
2. there is an accepting tree for each $(w_1, w_2, ..., w_k) \in L$, and
3. arbitrary accepting tree for each $(w_1, w_2, ..., w_k) \in L$ has no more than $f(\max_{1 \le i \le k} |w_i|)$ leaves.

(In the last definition we allow the values of $f(n)$ being ∞ since for some input words there can be infinitely many accepting trees without maximum of the number of leaves.)

Minimal leaf-size [Ki81, ITT83, MITT85, Hr85, Ge88, Na89] is more widely used rather than the maximal one [ITH89]. We consider both measures because minimal nontrivial complexities being different for minimal and maximal leaf-sizes gives an interesting contrast.

3 Semilinear Sets

Let P be a finite subset of N^k and $c \in N^k$. Let $L(c, P) = \{c + \sum_{p \in P} i_p \cdot p | i_p \in N\}$.

We say that subset $L \subseteq N^k$ is a linear set, iff $L = L(c, P)$ for some $c \in N^k$ and some finite subset P of N^k . We say that c is the constant of L, and P is the period system of L. An element $p \in P$ is a period of L. A subset $S \subseteq N^k$ is a semilinear set iff S is a finite union of linear sets.

Let $\mathcal{L}_r(c_1, c_2)$ be the class of linear r-dimensional sets with a constant from $\{0, \ldots, c_1\}^r$ and periods from $\{0, \ldots, c_2\}^r$ and $\mathcal{SL}_r(c_1, c_2)$ be the class of semilinear r-dimensional sets which are finite unions of sets from $\mathcal{L}_r(c_1, c_2)$.

For $v_1, \ldots, v_m \in N^k$ where $v_i = (x_{i1}, \ldots, x_{ik})$ we denote by $(v_1, \ldots, v_m)$ the vector $(x_{11}, \ldots, x_{1k}, x_{21}, \ldots, x_{2k}, \ldots, x_{m1}, \ldots, x_{mk}) \in N^{mk}$.

For $v \in N^k$ we denote by $mult_m(v)$ the vector $(v, v, \ldots, v) \in N^{mk}$.

For sets $A, B \subseteq N^k$ we define

$$mult_m(A) = \{mult_m(v) | v \in A\},$$
$$A + B = \{v_1 + v_2 | v_1 \in A \ \& \ v_2 \in B\}.$$

For each $m, k \geq 1$ and each set $S \subseteq N^{mk}$ we define the diagonal set $diag_k(S) = \{v \in N^k | mult_m(v) \in S\}$.

In the next section we use some lemmas following from the results by Dung T. Huynh [Th82].

Lemma 3. *For any $k, a \geq 1$ there is a number $b \geq 1$ such that*

$$(\forall m \geq 1)(\forall S \in \mathcal{SL}_{mk}(a^m, a)) \ diag_k(S) \in \mathcal{SL}_k(b^m, b^m).$$

Proof uses Corollary 2. 7 of the paper [Th82].

Lemma 4. *For arbitrary $r \geq 1$ there is a constant $c \geq 1$ such that*

$$(\forall n \geq 0)(\forall A, B \in \mathcal{SL}_r(n, n)) \ A \setminus B \in \mathcal{SL}_r(c^{n^c}, c^{n^c}).$$

Proof of this lemma is based on Theorem 3. 31 in [Th82].

4 Lower Bounds for Minimal and Maximal Leaf-Size

In this section we show semilinearity of languages over one-letter alphabet accepted by k-tape 1afa with a small amount of leaves. The main results of the paper are two subsequent theorems.

Theorem 5. *If a language $L \subseteq (\{0\}^*)^k$ is accepted by k-tape 1afa with $f(n) = o(\log n)$-bounded maximal leaf-size then L is semilinear.*

Theorem 6. *If a language $L \subseteq (\{0\}^*)^k$ is accepted by k-tape 1afa with $f(n) = o(\log \log n)$-bounded minimal leaf-size then L is semilinear.*

These theorems will follow from Lemma 7 and Theorems 8, 9.

Lemma 7. *If a language $L \subseteq (\{0\}^*)^k$ is accepted by k-tape 1afa with const-bounded minimal leaf-size (const $< \infty$) then L is semilinear.*

Theorem 8. *If a language $L \subseteq (\{0\}^*)^k$ is accepted by k-tape 1afa $\mathcal{A}$ with $f(n) = o(\log n)$-bounded maximal leaf-size then L is accepted by $\mathcal{A}$ also with $g(n)$-bounded maximal leaf-size where $g(n) \leq const < \infty$ for all sufficiently large n.*

Theorem 9. *If a language $L \subseteq (\{0\}^*)^k$ is accepted by k-tape 1afa $\mathcal{A}$ with $f(n) = o(\log\log n)$-bounded minimal leaf-size then L is accepted also by $\mathcal{A}$ with const-bounded minimal leaf-size (const $< \infty$).*

We present only sketches for proofs of results 7 — 9 due to lack of place.

By technical reasons, we define k-tape 1afa in a non-traditional way. We assume that the input tape is infinite to the right. We describe current heads positions by a vector $v \in N^k$. The initial position is zero-vector. We say that a vector of words $(0^{n_1}, 0^{n_2}, \ldots, 0^{n_k})$ is accepted by a computation path if the automaton enters an accepting configuration at the heads positions $(n_1, \ldots, n_k) \in N^k$. Hence according to our definition an automaton has no endmarker. In the same time the definition does not restrict capabilities of a k-tape 1afa because reading of the endmarker can be simulated by nondeterministic "guessing".

We define a configuration of a k-tape 1afa as a pair (q, v) where $q \in Q$ (Q is a set of states of the automaton) and $v \in N^k$ is a vector of heads positions.

We assume that each universal state has exactly two choices and for each state a pair of choices is determined by a function $\psi : U \longrightarrow Q \times Q$, where Q is the set of states, $U \subset Q$ is a set of universal states of the automaton, and heads in a universal state are not moved. Obviously this assumption does not limit possibilities of an automaton.

We assume also that all states in Q are ordered, and for each $q \in U$ the pair (q', q'') returned by the function $\psi(q)$ is ordered: $q' < q''$.

We consider an arbitrary accepting tree whose nodes are labeled with configurations $(q, \bar{v}) \in Q \times N^k$. Using ordering of edges in universal states defined by function ψ we may define an order for the set $\{\pi_1, \pi_1, ..., \pi_m\}$ of all leaves as a strict order: $\pi_1 < \pi_1 < ... < \pi_m$.

Let $q_0 \in Q$ be the initial state and $q_{acc} \in Q$ be the only accepting state.

We introduce a generalization of an accepting tree (called a $(q, \bar{q}) - tree$) as a finite tree that may be obtained by the given automaton, starting at an arbitrary state and whose leaves may have different configurations. It is not associated with any specific input of words. Instead we fix an a specific state $q \in Q$ in the root, an amount of leaves m and a vector of states $\bar{q} \in Q^m$. In contrast to the usual definition of the accepting tree each leaf in the $(q, \bar{q}) - tree$ may correspond to a different input of words. Obviously, $(q, \bar{q}) - tree$ is an accepting tree in a regular sense when $q = q_0$ and there is one fixed configuration $(q_{acc}, \bar{v})$ labeling all the leaves of the tree, therefore $\bar{q} = (q_{acc}, ..., q_{acc})$ and $\bar{v}$ is the vector of heads positions for all leaves, i.e. it corresponds to the same input words.

For $q \in Q$, $\bar{q} = (q_1, q_2, ..., q_m) \in Q^m$ we say that an automaton $(q, \bar{q}) - accepts$ a vector $(v_1, v_2, ..., v_m) \in N^{mk}$ where $v_i \in N^k$ iff there is a $(q, \bar{q}) - tree$ with leaves $\pi_1 < \pi_2 < ... < \pi_m$ such that a leave $\pi_i (i = 1, 2, ..., m)$ is labeled with the configuration (q_i, v_i).

Proof of Lemma 7, Theorem 8 and Theorem 9 is based on Lemma 3 and Lemma 4 from the previous section and the following Lemma 10.

Lemma 10. *For any k-tape 1afa $\mathcal{A}$ there is a constant $c \geq 1$ such that for arbitrary $m \geq 1, q \in Q, \bar{q} \in Q^m$ the set of $(q, \bar{q}) - accepted$ vectors belongs to $\mathcal{SL}_{mk}(cm, c)$.*

Outline of proof. For sets S, of all $(q, \bar{q}) - accepted$ vectors where $\bar{q} \in Q^1$, the constant c is chosen to ensure S$\in \mathcal{SL}_k(c, c)$. The constant can be found due to results of [Ib78] on nondeterministic automata. Next we use induction on m. The idea is to split an m-leaf three T into three following trees: $1-leaf$ tree T_0 (where $root(T_0) = root(T)$) with only existential non-leaf states, an $m' - leaf$ tree T_1 ($m' < m$), and an $(m-m')-leaf$ tree T_2 (where $root(T_1)$ and $root(T_2)$) are successors of the leaf of the tree T_0). The proof also uses the following obvious properties of semilinear sets: $\begin{matrix} A \in \mathcal{SL}_n(a,c) \& B \in \mathcal{SL}_n(b,c) \Rightarrow A + B \in \mathcal{SL}_n(a+b,c) \\ A \in \mathcal{SL}_n(a,c) \& B \in \mathcal{SL}_n(a,c) \Rightarrow A \cup B \in \mathcal{SL}_n(a,c) \end{matrix}$.□

5 Accepting Non-Regular Languages with $O(\log\log(n))$ - Bounded Minimal Leaf-Size

In this section we show that there is a non-regular language over one-letter alphabet accepted by 4-head 1-way alternating finite automaton with $f(n) = O(\log\log(n))$-bounded minimal leaf-size. We remark that automaton does not feel coincidence of heads.

Since from k-head automata accepting language $L \subseteq \Sigma^*$ one can easily construct the appropriate k-tape automaton accepting language $L' = \{(v, \dots, v) \in (\Sigma^*)^k | v \in L\}$ we can extend this result to 4-tape 1-way alternating finite automata as well.

Theorem 11. *There is a non-regular language L over one letter alphabet that can be accepted by 1-way finite alternating automaton with 4 heads with $O(\log\log(n))$-bounded minimal leaf-size and with $O(\log(n))$-bounded maximal leaf-size .*

Proof. Let

$$L = \{0^n \mid \quad n \geq 1 : \exists b \in \{2^k \mid k = 1, 2, ...\}, \\ \exists d \in \{1, ..., b-1\} : (not\ (d \mid n))\ and\ (b \mid n)\}.$$

The language L is similar to one defined in [AM75]. We denote by COH the configuration of heads (being a 4-tuple of integers): $COH = (h_1, h_2, h_3, h_4)$ where h_i is the distance from the $i - th$ head to the end marker.

To prove the theorem we describe a program of an 4−head automaton $\mathcal{A}$ that accepts the language L. We denote by $\forall$ points of a universal branching in the program.

In the construction of the automaton $\mathcal{A}$ we will use a subautomaton B which verifies whether the initial COH is of the form (n, n, d, q) where $n = dq$. Lets by $B(h_1, h_2, h_3, h_4)$ denote the fact that the automaton B is started on the 4-tape word $(0^{h_1}, 0^{h_2}, 0^{h_3}, 0^{h_4})$ in the one-letter alphabet $\{0\}$. In the end of the proof we construct the automaton B and we show that an algorithm of the B for $B(n, n, d, q)$ has exactly one accepting tree with $O(log\ d)$ leaves if $n = dq$, and no accepting tree if $n \neq dq$.

Now we describe the automaton $\mathcal{A}$. Let all the heads of $\mathcal{A}$ are on the first symbol of the input word 0^n,i.e. we have $COH = (n, n, n, n)$, $n > 0$.

Step #0: From $COH = (n, n, n, n)$ existentially get $COH = (n, n, b, n)$ where $b < n$; goto #1;

Step #1: ∀
- ↗ [verify $b \in \{2^k \mid k = 1, 2, ...\}$ using algorithm from [Ki81a]; accept iff belongs; **Stop**]
- → **goto #2;**
- ↘ **goto #3;**

Step #2: From $COH = (n, n, b, n)$ existentially get $COH = (n, n, b, z)$ where $z < n$; Start $B(n, n, b, z)$ verifying $n = bz$; accept iff $n = bz$ **Stop**

Step #3: From $COH = (n, n, b, n)$ existentially get $COH = (n, n, d, n)$ where $0 < d < b$; **goto #4;**

Step #4: From $COH = (n, n, d, n)$ existentially get $COH = (n, n - r, d, q)$ where $r > 0$, $0 < q < n$; **goto #5;**

Step #5: ∀
- ↗ [verify (n-r)+d>n; accept iff r<d; **Stop**]
- ↘ [from $COH = (n, n - r, d, q)$ existentialy get $COH = (n - r', n - r, d, q)$ where r'>0; **goto #6;**]

Step #6: ∀
- ↗ [verify (n-r)=(n-r'); accept iff r=r'; **Stop**]
- ↘ [start $B(n - r, n - r, d, q)$ verifying $n - r = dq$; accept iff $n = dq + r$; **Stop**]

Note that the step #2 verifies whether $(b|n)$ where $b \in \{2^k \mid k = 1, 2, ...\}$; steps #4 — 6 verify whether *not* $(d \mid n)$ where $0 < d < b$. Therefore the automaton A accepts the language L.

Now we estimate the leaf-size. Let $0^n \in L$. We have an accepting tree for each b, d where $b \in \{2^k \mid k = 1, 2, ...\}$is chosen in the step #0 and $d \in \{1, 2, ..., b-1\}$is chosen in the step #3 such that *not* $(d \mid n)$ *and* $(b \mid n)$. Algorithm from [Ki81a] used in step #1 requires $O(log\ b)$ leaves.

Verification whether $(b \mid n)$ in the step #2 requires $O(log\ b)$ leaves. Verification whether *not* $(d \mid n)$ in steps #4 — 6 requires $O(log\ d)$ leaves.

Total number of required leaves is $O(log\ b + log\ d)$. Since $d < b \leq n$, the number of leaves of an arbitrary accepting tree does not exceed $O(log\ n)$, i.e. the language L is accepted by the automaton $\mathcal{A}$ with $O(log\ n)$ – bounded maximal leaf-size.

Let us estimate the minimal leaf-size. For $0^n \in L$ we can choose $d = d_{\min}$, $b = 2^{\lceil \log_2 d_{\min} \rceil}$ where $d_{\min}$ is the minimal integer not dividing n. Therefore we have an accepting tree with $O(log_2 d_{\min})$ leaves. It is known that $d_{\min} = O(log\ n)$. Therefore we get that the language L is accepted by the automaton $\mathcal{A}$ with $O(\log\log\ n)$ – bounded minimal leaf-size.

To conclude the proof we construct the automaton B verifying for $COH = (n, n, d, q)$ whether $n = dq$. It is technically easier to describe an automaton C verifying for $COH = (n + d, n + d, d, q)$ whether $n = dq$, and it requires $O(\log d)$ leaves. Then after $B(n, n, d, q)$ can be easily reduced to $C(n, n, d, q - 1)$.

Let in the beginning $COH = (n + d, n + d, d, q)$. If $d = 1$, the computation is trivial. Let $d > 1$. We denote:

$$s = \begin{cases} 1\ , \text{ if } d \text{ is odd;} \\ 0\ , \text{ if } d \text{ is even;} \end{cases}, \quad d' = (d-s)/2, \;\; n' = (n - sq)/2.(***)$$

Obviously, $n = dq$ iff $n' = d'q$. We shall reduce $C(n+d, n+d, d, q)$ to $C(n'+d', n'+d', d', q)$ with constant number of universal branching. Since $d' \leq d/2$, the total number of leaves for accepting tree will not exceed $O(\log d)$.

Reduction of $C(n+d, n+d, d, q)$ to $C(n'+d', n'+d', d', q)$.

Step #0: [Existentially guess $s \in \{0, 1\}$; From $COH = (n+d, n+d, d, q)$ moving the 1st head existentially get $COH = (n+d', n+d, d, q)$ where $d' < d$; **goto #1**;]

Step #1: $\forall$
- ↗ [Verify $n + d' = (n+d) - (d+s)/2$; and **accept** iff the values of s and d' are as defined in (***) **Stop**]
- ↘ [Considering the other $\forall$ – choice the current COH may be denoted by $(n+d', n+d, d, q)$; Moving the 3-rd head existentially guess $COH = (n+d', n+d, y, q)$ where $y < d$; **goto #2**;]

Step #2: $\forall$
- ↗ [Verify $y = (n+d) - (n+d') - s$; **accept** iff $y = d'$; **Stop**]
- ↘ [Considering the other $\forall$ – choice the current COH may be denoted by $(n+d', n+d, d', q)$; Moving the 2-nd head existentially guess $COH = (n+d', z, d', q)$ where $z < n+d$; **goto #3**;]

Step #3: $\forall$
- ↗ [Verify $z = n + d'$; **accept** iff $z = n + d'$; **Stop**]
- ↘ [Considering the other $\forall$ – choice the current COH may be denoted by $(n+d', n+d', d', q)$; Moving the 1-st head existentially guess $COH = (u, n+d', d', q)$ where $u < n+d'$; **goto #4**;]

Step #4: $\forall$
- ↗ Verify $u = (n+d')/2 - d'/2 - sq/2$; **accept** iff $u = n' + d'$; **Stop**
- ↘ [Considering the other $\forall$ – choice the current COH may be denoted by $(n'+d', n+d', d', q)$; Moving the 2-nd head existentially guess $COH = (n'+d', v, d', q)$ where $v < n+d'$; **goto #5**;]

Step #5: $\forall$
- ↗ [Verify $v = n' + d'$; **accept** iff $v = n' + d'$; **Stop**]
- ↘ [Considering the other $\forall$ – choice the current COH may be denoted by $(n'+d', n'+d', d', q)$; Start $C(n'+d', n'+d', d', q)$ that verifies: $n' = d'q$; **Stop**]

□

References

[AM75] H.ALT and K.MEHLHORN, A language over a one symbol alphabet requiring only $O(\log\log n)$ space, SIGACT news 7(4) (1975), P.31-33.

[CK81] A.K.CHANDRA, D.C.KOZEN and L.J.STOCKMEYER, Alternation, JACM 28 (1981), P.114–133.

[DHKRS89] J.DASSOV, J.HROMKOVIČ, J.KARHUMÄKI, B.ROVAN, and A.SLOBODOVA, On the power of synchronization in parallel computations, in Proc. 14th MFCS, LNCS 379, Springer-Verlag, 1989, P.196–206.

[Fr81] R.FREIVALDS, Projections of Languages Recognizable By Probabilistic And Alternating Finite Multitape Automata, Information Processing Letters, v.13, 1981, P.195–198.

[Ge88] D.GEIDMANIS, On the capabilities of alternating and nondeterministic multitape automata, Proc. Found. of Comp. Theory, LNCS 278, (Springer, Berlin, 1988), P.150–154.

[Hr85] J.HROMKOVIČ, On the power of alternation in automata theory, Journ. of Comp.and System Sci., Vol.31, No.1, 1985, P.28–39

[Hr86] J.HROMKOVIČ, How to organize the communications among parallel processes in alternating computations, Manuscript, January 1986

[HI68] M.A.HARRISON and O.H.IBARRA, Multi-tape and multi-head pushdown automata, Information and Control, 13, 1968, P.433–470.

[HRHY86] R.R.HOWELL, L.E.ROSIER, D.T.HUYNH and H.-C. YEN, Some complexity bounds for problems concerning finite and 2-dimensional vector addition systems with states, Ther.Comp.Sci. 46 (1986). P.107–140.

[Ib78] O.H.IBARRA, Reversal bounded multicounter machines and their decision problems, Journ. of ACM, Vol. 25, No.1, 1978, P.116–133.

[IT91] O.H.IBARRA and N.Q.TRAN, On space-bounded synchronized alternating Turing machines, LNCS 529 (Springer, Berlin, 1991) P.248–257.

[ITT83] K.INOUE, I.TAKANAMI and H.TANIGUCHI, Two-dimensional alternating Turing machines, Theoret.Comput.Sci.27 (1983) P.61–83.

[ITH89] K.INOUE, I.TAKANAMI and J.HROMKOVIČ, A leaf-size hierarchy of two dimensional alternating Turing machines, Theoretical Computer Science, 67, 1989, P.99–110.

[Ki81] K.N.KING, Measures of parallelism in alternating computation trees, Proc.13th Ann.ACM Symp.on Theory of Computing (1981) P.189–201.

[Ki81a] K.N.KING, Alternating multihead finite automata, LNCS 115 (Springer, Berlin, 1981) P.506–520.

[LLS78] R.E.LADNER, R.J.LIPTON, and L.J.STOCKMEYER, Alternating pushdown automata, Conf. Rec. IEEE 19th Ann. Symp. on Found. of Comp. Sci. (1978), P.92–106.

[MITT85] H.MATSUNO, K.INOUE, H.TANIGUCHI and I.TAKANAMI, Alternating simple multihead finite automata, Theoret.Comput.Sci. 36(1985) P.291–308.

[Na89] I.R.NASYROV, On some complexity measure for alternating automata with pushdown storage, Izvestija VUZ, Matematika, 1989, P.37–45. (in Russian)

[Pe 95] H.PETERSEN, Alternation in simple devices, Lecture Notes in Computer Science 944, ICALP95 (1995) P.315-323.

[PPR80] W.J.PAUL, E.J.PRAUSS and R.REISHUK, On alternation, Acta Informatica, 14 (1980), P.243–255.

[PR80] W.J.PAUL and R.REISHUK, On alternation II, Acta Informatica, 14 (1980), P.391–403.

[Th82] THIET-DUNG HUYNH, The complexity of semilinear sets, Elektronische Informationsverarbeitung und Kybernetik EIK 18(1982), 6, P.291–338.