INFORMATIQUE THÉORIQUE ET APPLICATIONS

MATTHIAS KRAUSE

Separating $\oplus L$ from L, NL, co-NL, and $AL = P$ for oblivious Turing machines of linear access

Informatique théorique et applications, tome 26, nº 6 (1992), p. 507-522

<http://www.numdam.org/item?id=ITA_1992__26_6_507_0>

Informatique théorique et Applications/Theoretical Informatics and Applications
(vol. 26, n° 6, 1992, p. 507 à 522)

SEPARATING $\oplus L$ FROM L, NL, co-NL, AND $AL=P$ FOR OBLIVIOUS TURING MACHINES OF LINEAR ACCESS (*)

by Matthias Krause

Communicated by J. Berstel

Abstract. – *Using algebraic arguments we derive lower bounds for parity branching programs. Several exponential lower bounds on the width of linear length bounded oblivious parity branching programs are obtained, so for the* graph accessibility problem *of directed graphs, and the word problem of free groups and one-relator-groups of finite rank. Using well-known results on the simulation of logspace-bounded Turing machines by sequences of branching programs we complete the separation of the complexity classes L, NL,* co-*NL*, $\oplus$ *L*, *AL = P for oblivious Turing machines of linear access time.*

Résumé. – *Nous utilisons des techniques algébriques pour obtenir des bornes inférieures sur des programmes à branchements particuliers. Des minorations exponentielles sur la largeur de programmes « oblivious » de longueur linéaire sont obtenus; il en est ainsi pour le* problème d'accessibilité *dans les graphes orientés, et pour le problème des mots dans les groupes libres, et les groupes à un relateur de rang fini. En employant des résultats bien connus sur la simulation de machines de Turing à espace logarithmique, nous complétons la séparation des classes de complexité L, NL,* co-*NL*, $\oplus$ *L*, *AL = P pour les machines de Turing « oblivious » à temps d'accès linéaire.*

INTRODUCTION

It is a main goal of theoretical computer science to separate complexity classes defined by Turing-machines which work according to various types of acceptation. In this context besides the classical complexity classes such as L, NL, P the study of complexity classes coming from Turing-machines with parity-acceptation, such as $\oplus L$ and $\oplus P$ has become increasingly important [PZ83, M88, CH89]. We show that in the case of oblivious Turing-machines of logarithmic space and linear access time the classes of decision problems computable by deterministic, nondeterministic, co-nondeterministic,

(*) Received August 1990, accepted September 1991.

[1] Fachbereich 4, Universität Dortmund, Postfach 500500, D-W 46 Dortmund 50, Germany.

parity-, and alternating machines, respectively, are strongly separated from each other.

Hereby, access time counts the steps at which the machine is really reading (*i.e.*, which correspond to branching points in the computation graph) and not the moving steps at the tape. To make the difference between access- and usual time clearer we use the following *random access input* variation of the Turing machine model similar to that defined in [Ru81].

In this model instead of a input-head the Turing machine has a special *index tape* and a special *read state*. Whenever it enters the read state with the natural number i written on the index tape, the i-th input bit is available. The access time counts the number of entering the read state and corresponds to the length of the simulating oblivious Ω-branching program. Logarithmic-space bounded deterministic Turing-machines of this type compute exactly those languages in L, where each language of P can be decided by a logspace bounded alternating Turing machines of linear access-time. Note that the well-known crossing sequence arguments which provide, e. g., that *palindroms* cannot be recognized within linear time and logarithmic space (*see*, e. g. [Co66]) do not apply to the case of linear access-time. It is quite obvious that the decision wether a given word is a palindrom can be made within logarithmic space and linear access-time.

Our investigations are based on the observation that the problem of separating some of the most interesting complexity classes can be formulated as lower bound problem for Ω-branching programs. Ω-branching programs were introduced by Meinel [M88]. They differ from ordinary branching programs by having additional operation nodes evaluating binary Boolean operations from a given set $\Omega \subseteqq B_2$. For each subset $\Omega \subseteqq B_2$ we denote by $\mathcal{P}_{\Omega-\mathrm{BP}}$ the set of all languages computable by sequences of polynomial-size Ω-branching programs. As usual, Ω-branching are called deterministic, disjunctive, conjunctive, parity, or alternating programs if $\Omega = \varnothing$, $\Omega = \{ \vee \}$, $\Omega = \{ \wedge \}$, $\Omega = \{ \oplus \}$, or $\Omega = \{ \vee, \wedge \}$, respectively.

The computational power of $\Omega-$branching programs is essentially characterized by the following two facts.

Fact 1 [M88]: Each decision problem computable by a sequence of polynomial-size Ω-branching programs for some $\Omega \subseteqq B_2$ belongs to one the classes $\mathcal{P}_{\mathrm{BP}}$, $\mathcal{P}_{\{\vee\}-\mathrm{BP}}$, $\mathcal{P}_{\{\wedge\}-\mathrm{BP}}$, $\mathcal{P}_{\{\oplus\}-\mathrm{BP}}$, and $\mathcal{P}_{\{\vee, \wedge\}-\mathrm{BP}}$. ■

Fact 2: Each logarithmic space bounded deterministic, nondeterministic, co-nondeministic, parity- or alternating Turing-machine can be simulated by a sequence of polynomial-size deterministic, disjunctive, conjunctive, parity or alternating branching programs, respectively. Thus,

$$L \subseteqq \mathscr{P}_{\mathrm{BP}}, \qquad NL \subseteqq \mathscr{P}_{\{\vee\}-\mathrm{BP}}, \qquad \text{co-}NL \subseteqq \mathscr{P}_{\{\wedge\}-\mathrm{BP}},$$
$$\oplus L \subseteqq \mathscr{P}_{\{\oplus\}-\mathrm{BP}}, \qquad \text{and} \qquad AL = P \subseteqq \mathscr{P}_{\{\vee,\wedge\}-\mathrm{BP}}$$

(*see*, e. g., [M88] for details). ■

Let us still mention here that the Immermann-Szelepszeny result $NL = \text{co-}NL$ [I87], [S87] provides $\mathscr{P}_{\{\vee\}-\mathrm{BP}} = \mathscr{P}_{\{\wedge\}-\mathrm{BP}}$.

To find methods for proving superpolynomial lower bounds on the complexity of unrestricted Ω-branching programs seems to be a very hard problem. Today such methods are known for some kinds of depth-restricted programs such as

– deterministic, disjunctive, conjunctive and parity-branching trees [W84, DM89],

– deterministic, disjunctive, and conjunctive read-once-only branching programs [W84, A&86, J86, KMW88], and

– determinstic, disjunctive, and oblivious branching programs of linear length [KW89, KWM89].

By using these methods the following separation results have been achieved so far.

(1) In the case of Ω-branching trees the five classes mentioned in Fact 1 are strongly separated from each other [DM89].

(2) In the case of read-once-only Ω-branching programs the deterministic, disjunctive, conjunctive and alternating classes are strongly separated [KWM88]. Observe that in the sense of theorem 2 the uniform counterpart of read-once-only Ω-branching programs are Eraser Turing machines introduced in [A&86].

(3) The same as in (2) holds for oblivious Ω-branching programs of linear length [KWM89].

In the sense of Fact 2 oblivious Ω-branching programs correspond exactly to the machine model described above, where the access time corresponds to the length of the simulating oblivious Ω-branching program (*see* e. g., [M89] for details).

The lower bound argument providing the separation result (3) has been achieved by using a special "Cut & Paste"-method. It has been shown that if the program is too small then there exists an accepting (rejecting) path for some input word which has to be rejected (accapted). It is quite obvious that this argument does not work in the case of parity programs. That is why completing the separation of the main complexity classes mentioned in Fact 2 remained an open problem in case (2), as well as in case (3).

We solve this problem for oblivious Ω-branching programs of linear length (case 3) by giving a lower bound technique for parity branching programs which is based on multilinear algebra over the field GF(2). As all upper bounds used in [KMW89] and in this paper are constructive this includes also the complete separation of the corresponding uniform classes.

The paper is organized as follows. In Section 2 we describe the lower bound technique. In section 3 we obtain exponential lower bounds on the width of linear length bounded oblivious parity branching programs which compute the *graph accessibility problem*, the *word problem of free groups of finite rank*, and some special decision problem called EQUALITY*. In section 4 we establish the separation result by considering another special decision problem, INNER PRODUCT*.

1. PRELIMINARIES

1.1. Ω-Branching programs

Let Σ denote a finite alphabet, $X_n=\{x_1, \ldots, x_n\}$ a set of n variables taking values from Σ, and Ω a subset of B_2. An Ω-branching program P over X_n is a directed, acyclic, labelled, finite graph fulfilling the following properties.

– Nodes of outdegree zero are called sinks.

– The set of non-sink nodes contains query nodes of outdegree $\# \Sigma$, and operation nodes having outdegree two. There is one distinguished non-sink of indegree zero called the source.

– Sinks are labelled by constants 0 or 1, query nodes and operation nodes have labels from x_n and Ω, respectively.

– Edges leaving query nodes are labelled by literals from Σ, where distinct edges leaving the same query node have distinct labels.

Given an input word $w=(w_1, \ldots, w_n)\in\Sigma^n$ a path p in P will be called *w-consistent* if for all query nodes v occuring in p the following is true. If the label of v is x_i, $1\leqq i\leqq n$, then p takes the edge leaving v which is labelled by w_i. A w-consistent path is called *w-accepting* (*w-rejecting*) if it leads from the source to the 1-sink (0-sink).

We denote by P^w the subgraph of P formed by all w-consistent paths leading from the source to one of the sinks. Obviously, the branching nodes in P^w are given by the operation nodes belonging to P^w. The input w assigns some Boolean constant δ_v to each node v of P^w in the following way.

If v is a sink then δ_{v} is given by the label of v.

– If v is query node with successor v' in P^w then $\delta_v = \delta_{v'}$.

– If v is an operation node labelled $\omega \in \Omega$ and having successors v' and v'' then $\delta_v = \omega(\delta_{v'}, \delta_{v''})$.

By definition, the function value $P(w) \in \{0, 1\}$ computed by P on input w is given by the constant assigned to the source. Obviously, taking $\Omega = \varnothing$ and $\Sigma = \{0, 1\}$ we obtain the well-known deterministic branching program model.

Further observe that each disjunctive (conjunctive) branching program P over x_n computes 1 (0) on a given input w if and only if there is some w-accepting (w-rejecting) path in P. If P is a parity program it will compute 1 on w if and only if the number of w-accepting paths in P is odd.

Significant complexity measures for Ω-branching programs are size and depth, *i.e.*, the number of non-sink nodes and the maximal length of a directed path of the program, respectively.

1.2. Oblivious Ω-branching programs

An Ω-branching program P is called *levelled* if the set of nodes can be divided into disjoint levels $L_1, L_2, \ldots, L_{d+1}$, $d = \mathit{depth}(P)$, so that for all edges (v, v') there is some i, $1 \leqq i < d$, with $v \in L_i$ and $v' \in L_{i+1}$.

Levelled programs will be called *oblivious* if non-sink nodes nodes on the same level have always the same label. For oblivious programs P let *width* (P) denote the maximal number of nodes in any level and *length* (P) the number of query-node-levels. Let $\Omega \subseteqq B_2$ and P be an oblivious Ω-branching program over X_n. For given natural numbers i, j with $1 \leqq i < j \leqq \mathit{depth}(P) + 1$ we denote by *segment* (i, j) of P the subgraph consisting of the levels $i, \ldots, j-1$ of P.

Let W and W' be disjoint subsets of X_n. A sequence $S = (j_1, i_2, j_2, \ldots, i_k, j_k)$ with $1 < j_1 < i_2 < \ldots < j_k \leqq \mathit{depth}(P) + 1$ is called (W, W')*-segmentation* of P, if for all l, $1 \leqq l \leqq k$, *segment* (i_l, j_l) and *segment* (j_l, i_{l+1}) does not contain any label from W' and W, respectively. [Set $i_1 = 1$ and $i_{k+1} = \mathit{depth}(P) + 1$.]

The *alternation depth* of P with respect to W and W' of X_n (*altdepth* (P, W, W')) is defined as to be the length of a shortest (W, W')-segmentation of P. Lower bound arguments for oblivious Ω-branching programs of linear length make use of the following Ramsey-theoretic observation [AM86, KW89].

Fact 3.: For all $n \in \mathbb{N}$ let P_n denote an oblivious Ω-branching program over X_n, where $length(P_n) \leqq D \cdot n$ for some $D \in \mathbb{N}$. Then there are numbers $\delta \in (0, 1/2)$ and $k \in \mathbb{N}$ so that for each balanced partition (Y_n, Z_n) of X_n there are subsets $W_n \subseteq Y_n$ and $W'_n \subseteq Z_n$ with $\# W_n = \# W'_n = \delta . n$ so that

$$altdepth(P_n, W_n, W'_n) \leqq k. \quad \blacksquare$$

1.3. Technical definitions

For all subsets Y of X_n we denote by Σ^Y the sets of assignments $c: Y \to \Sigma$. For disjoint $Y, Y' \subseteq X_n$, and $c \in \Sigma^Y$, $c' \in \Sigma^{Y'}$, we denote by $c \cup c'$ the unique assignment of $Y \cup Y'$ which coincides with c and c' on Y and Y', respectively.

Let $f = f(x_1, \ldots, x_n): \Sigma^n \to \{0, 1\}$, and (Y, Z) be a balanced partition of X_n (*i.e.*, $Y \cap Z = \emptyset$, $Y \cup Z = X_n$, and $|\# Y - \# Z| \leqq 1$). We denote by $M^f_{(Y, Z)}$ the *communication matrix* of f with respect to (Y, Z), *i.e.*, rows and columns of $M^f_{(Y, Z)}$ are labelled by the elements from Σ^Y and Σ^Z, respectively, and for all $c \in \Sigma^Y$ and $c' \in \Sigma^Z$ it holds

$$M^f_{(Y, Z)}(c, c') = f(c, c').$$

1.4. Decision Problems

We consider the *graph accessibility problems* GAP and GAPr, $r \in \mathbb{N}$, for directed graphs as to be sequences of Boolean functions $(\mathrm{GAP}_{n.n})_{n \in \mathbb{N}}$ and $(\mathrm{GAP}\,r_{n.n})_{n \in \mathbb{N}}$, respectively. For given $n \in \mathbb{N}$ we regard each Boolean $n \times n$-matrices A as to be the adjacency matrix of a directed graph $G(A)$ over a fixed set $\{v_1, \ldots, v_n\}$ of vertices. Let $\mathrm{GAP}_{n.n}(A) = 1 \Leftrightarrow$ there is a directed path in $G(A)$ connecting v_1 and v_n, $\mathrm{GAPr}_{n.n}(A) = 1 \Leftrightarrow \mathrm{GAP}_{n.n}(A) = 1$ and the outdegree of any vertice in $G(A)$ is at most r. Further let us define the decision problems EQUALITY and INNER PRODUCT (for short EQ and IP, respectively), where $\mathrm{EQ} = (\mathrm{EQ}_{2n})_{n \in \mathbb{N}}$, $\mathrm{IP} = (\mathrm{IP}_{2n})_{n \in \mathbb{N}}$ and EQ_{2n}, $\mathrm{IP}_{2n}: \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1\}$ are defined as follows.

For all $w, w' \in \{0, 1\}^n$ let

$$\mathrm{EQ}_{2n}(w, w') = 1 \quad \Leftrightarrow \quad w = w',$$

and

$$\mathrm{IP}_{2n}(w, w') = 1 \quad \Leftrightarrow \quad w_1 w'_1 + \ldots + w_n w'_n = 1 \mod 2.$$

For all sequences $F = (f_{2n}: \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1\})_{n \in \mathbb{N}}$ let us define the decision problem $F^* = (f^*_{2n}: \{0, 1, 2\}^n \times \{0, 1\})_{n \in \mathbb{N}}$ over the alphabet $\{0, 1, 2\}$. For all words $w \in \{0, 1, 2\}^*$ we denote by $red(w) \in \{0, 1\}^*$ the reduced word of w obtained from w by deleting all occurences of the literal "2".

For example, $red(202112102122)=(011101)$.

Now for all $n \in \mathbb{N}$ and all words $w, w' \in \{0, 1, 2\}^n$ let

$f^*_{2n}(w, w')=1 \Leftrightarrow red(w)$ and $red(w')$ have equal positive length m and $f_{2m}(red(w), red(w'))=1$.
For example, $\mathrm{EQ}^*_{12}(2120121, 1120202)=\mathrm{EQ}_8(1011, 1100)=0$, but $\mathrm{IP}^*_{12}(2120121, 1120202)=\mathrm{IP}_8(1011, 1100)=1+0+0+0=1$.

2. THE LOWER BOUND ARGUMENT

Suppose that P is a $\{\oplus\}-\mathrm{BP}_0$ over $X_n=\{x_1, \ldots, x_n\}$ computing $f: \Sigma^n \rightarrow \{0, 1\}$. Further let (Y, Z) be a balanced partition of X_n and $k=altdepth(P, Y, Z)$. This section is devoted to the proof of the following

THEOREM 1: *$width(P) \geqq [rank_{GF(2)}(M^f_{(Y, Z)})]^{1/(2k-1)}$.*

Let $W=width(P)$. Suppose that the nodes at any level i of P are numbered as $v_{i,1}, v_{i,2}, v_{i,3}, \ldots$

Further, if we are given nodes v and v' of P and some, maybe partial, assignment c of X we denote by $\#(v \xrightarrow{c} v')$ the number of c-consistent paths leading from v to v'. For each input $w \in \Sigma^n$ and all natural numbers r, $1 \leqq r < depth(P)$, we define the *top vector* and the *bottom vector* $\mathrm{top}_r(w)$ and $\mathrm{bot}_r(w)$, respectively, from $GF(2)^W$ as follows. For all i, $1 \leqq i \leqq W$, let

$$[\mathrm{top}_r(w)]_i = \#(v_0 \xrightarrow{w} v_{r,i}) \mod 2,$$

$$[\mathrm{bot}_r(w)]_i = \#(v_{r,i} \xrightarrow{w} s_1) \mod 2,$$

if level r contains not less then i nodes. Otherwise let

$$[\mathrm{top}_r(w)]_i = [\mathrm{bot}_r(w)]_i = 0.$$

For all $w \in \Sigma^n$ and each pair (r, s), $1 \leqq r < s \leqq depth(P)$ let the *segment matrix* $\mathrm{seg}_{r,s}(w) \in \mathbb{M}_W(GF(2))$ *be defined as follows. For all i, j,* $1 \leqq i, j \leqq W$, let

$$[\mathrm{seg}_{r,s}(w)]_{i,j} = \#(v_{r,i} \xrightarrow{w} v_{s,j}) \mod 2,$$

if $v_{r,i}$ and $v_{s,j}$ do exist. Otherwise let

$$[\mathrm{seg}_{r,s}(w)]_{i,j} = 0.$$

(Hereby, $\mathbb{M}_r(\mathrm{GF}(2))$ denotes the $GF(2)$ – algebra of $r \times r$-matrices over GF (2) with the usual matrix product.)

LEMMA 2.1: *For all r, s, $1 \leqq r < s < depth(P)$, and all $w \in \Sigma_n$ it holds*

$$f(w) = \mathrm{IP}_{2W}(\mathrm{top}_r(w), \mathrm{bot}_r(w)), \tag{1}$$

$$\mathrm{bot}_r(w) = \mathrm{seg}_{r,s}(w) \,.\, \mathrm{bot}_s(w). \tag{2}$$

Proof: Let $w \in \Sigma^n$ be arbitrarily fixed. For all i, $1 \leqq i \leqq W$, let P^i denote the sub-program of P with starting node $v_{r,i}$. Observe that

$$\mathrm{bot}_r(w) = (P^1(w), \ldots, P^W(w)). \tag{I}$$

Now the proof of (1) follows directly as P is a parity branching program.

Further observe that for all i, $1 \leqq i \leqq W$, the i-th row of the matrix $\mathrm{seg}_{r,s}(w)$ equals the top vector of w with respect to level $s-r$ of P^i. Thus, (2) is a straightforward consequence of (I) and (1). ■

Recall some facts about multilinear mappings. Let $A_1, \ldots, A_k$ and $B_1, \ldots, B_k (k \in \mathbb{N})$ be $GF(2)$-vector spaces, where

$$\dim_{GF(2)}(A_i) = \dim_{GF(2)}(B_i) = d^{(i)} \qquad \text{for all} \quad 1 \leqq i \leqq k.$$

Further let ϕ denote a $2k$-linear mapping ϕ:

$$A_1 \times \ldots \times A_k \times B_1 \times \ldots \times B_k \rightarrow GF(2).$$

Denote by $M(\phi)$ the following matrix from $\mathbb{M}_\alpha(GF(2))$, where α denotes the cardinality of the cartesian product $A_1 \times \ldots \times A_k$, *i.e.*, $\alpha = \prod_{i=1}^{k} \# A_i = \prod_{i=1}^{k} 2^{d^{(i)}}$.

Suppose that rows and columnes of $M(\phi)$ are labelled by elements from $A_1 \times \ldots \times A_k$ and $B_1 \times \ldots \times B_k$, respectively. Then for all

$$a = (a_1, \ldots, a_k) \in A_1 \times \ldots \times A_k \qquad \text{and} \qquad b = (b_1, \ldots, b_k) \in B_1 \times \ldots \times B_k$$

let

$$M(\phi)_{a,b} = \phi\,(a_1, \ldots, a_k, b_1, \ldots, b_k).$$

As Φ is linear in each component the rank of $M(\Phi)$ cannot exceed the minimum of the dimensions of the vector spaces $A_1 \times \ldots \times A_k$ and $B_1 \times \ldots \times B_k$. Consequently, LEMMA 2.2: $rank_{GF(2)}(M(\phi)) \leqq \prod_{i=1}^{k} d^{(i)}$. ■

The Proof of Theorem 1: Fix an (Y, Z)-segmentation $(j_1, i_2, j_2, \ldots, i_k, j_k)$ of P. For all inputs $w \in \{0, 1\}^n$ let

$$\alpha(w) = \mathrm{top}_{j_1}(w),$$

$$\beta(w) = \mathrm{bot}_{j_k}(w)$$

and, for all r, $2 \leqq r \leqq k$,

$$a_r(w) = \mathrm{seg}_{i_r,\, j_r}(w),$$

$$b_r(w) = \mathrm{seg}_{j_{r-1},\, i_r}(w).$$

Let $A_1 = B_1 = GF(2)^W$ and $A_2 = B_2 = \ldots = A_k = B_k = \mathbb{M}_W(GF(2))$, and let the $2k$-linear mapping $\phi_0 : A_1 \times \ldots \times A_k \times B_1 \times \ldots \times B_k \rightarrow GF(2)$ be defined as

$$\phi_0(\alpha, a_2, \ldots, a_k, \beta, b_2, \ldots, b_k) = \mathrm{IP}_W(\alpha, b_2 \circ a_2 \circ \ldots \circ b_k \circ a_k \circ \beta).$$

Lemma 2.1. yields that for each input vector $w \in \{0, 1\}^n$

$$\begin{aligned} f(w) &= \mathrm{IP}_W(\alpha(w), \mathrm{bot}_{j_1}(w)) \\ &= \mathrm{IP}_W(\alpha(w), b_2(w) \circ a_2(w) \circ \ldots \circ b_k(w) \circ a_k(w) \circ \beta(w)) \\ &= \phi_0(\alpha(w), a_2(w), \ldots, a_k(w), \beta(w), b_2(w), \ldots, b_k(w)). \end{aligned}$$

On the other hand observe that for all inputs

$$w \in \{0, 1\}^n (\alpha(w), a_2(w), \ldots, a_k(w)) \quad \text{and} \quad (\beta(w), b_2(w), \ldots, b_k(w))$$

depend merely on $w|_Y$ and $w|_Z$, respectively.

Consequently, there is a submatrix M^* of $M(\phi_0)$ with the property that

$$rank_{GF(2)}(M^*) = rank_{GF(2)}(M^f_{(Y,Z)}).$$

The rows and columnes of M^* are determined by exactly those

$$a = (\alpha, a_2, \ldots, a_k) \in A_1 \times \ldots \times A_k \quad \text{and} \quad b = (\beta, b_2, \ldots, b_k) \in B_1 \times \ldots \times B_k,$$

respectively, for which there are assignments c to Y and c' to Z with $a=(\alpha(c), a_2(c), \ldots, a_k(c))$ and $b=(\beta(c'), b_2(c'), \ldots, b_k(c')$. Hence, according to Lemma 2.2.,

$$rank_{GF(2)}(M^f_{(Y,Z)}) \leqq rank_{GF(2)}(M(\phi_0)) \leqq W.W^{2(k-1)} = W^{2k-1},$$

This completes the proof of Theorem 1. ■

To obtain a lower bound argument for length-restricted oblivious parity branching programs we formulate this result for subfunctions.

COROLLARY 2.3: *Let W and W' be disjoint subsets of X_n and $k = altdepth(P, W, W')$. Then for all assignments c of $X_n \setminus (W \cup W')$ it holds*

$$width(P) \geqq [rank_{GF(2)}(M^{f^c}_{(Y,Z)})]^{1/(2k-1)}. \quad ■$$

Straightforward application of Fact 3 and Corollary 2.3 yields

COROLLARY 2.4: *Let $g: \mathbb{N} \to \mathbb{N}$ be a nondecreasing function. Further, for all $n \in \mathbb{N}$ let f_n be a decision function over $x_n = \{x_1, \ldots, x_n\}$ which fulfils the following condition. There is a balanced partition (Y_n, Z_n) of X_n so that for all m, $1 \leqq m \leqq n/2$, and all $W \subseteqq Y_n$ and $W' \subseteqq Z_n$ of cardinality m there is an assignment c to $X_n \setminus (W \cup W')$ so that*

$$rank_{GF(2)}[M^{f^c}_{(W,W')}] \geqq 2^{g(m)}.$$

Then all sequences of $\{\oplus\}$-BP_0's which compute $(f_n)_{n \in \mathbb{N}}$ within linear length have width $\exp(\Omega(g(n)))$. ■

3. LOWER BOUNDS

In [KW89] there is developed the following method for proving lower bounds on the size of oblivious disjunctive branching programs. Let $f = f(x_1, \ldots, x_n): \Sigma^n \to \{0, 1\}$ be a decision function and (Y, Z) be a balanced partition of $X_n = \{x_1, \ldots, x_n\}$.

DEFINITION 3.1: A set $c \subseteqq \Sigma^n$ is called *sheaf* of f with support (Y, Z) if there are pairwise distinct assignments $c_1, \ldots, c_T$ to Y and $c'_1, \ldots, c'_T$ to Z with

(1) $C = \{c_i \cup c'_i;\ i = 1, \ldots, T\}$,

(2) $f(c_i \cup c'_j) = \delta_{ij}$ for all $i, j \in \{1, \ldots, \mathrm{T}\}$.

(Hereby, $\delta = \delta_{ij}$ denotes the Kronecker-symbol.)

Let P be an oblivious Ω-branching program computing f and $k = altdepth(P, Y, Z)$. Further suppose that there is a sheaf C of f with support (Y, Z).

LEMMA 3.1 [KW89]: *If P is disjunctive then $width(P) \geqq \# C^{1/(2k-1)}$.* ■

In section 4 a more general version of Lemma 3.1, will be proved.

Due to Proposition 2.1 the same is true for $\{\oplus\}$-BP$_0$'s.

LEMMA 3.2: *If P is a parity program then $width(P) \geq \# C^{1/(2k-1)}$.*

Proof: Let $T = \# C$ and fix assignments $c_1, \ldots, c_T$ of Y and $c'_1, \ldots, c'_T$ to Z so that condition (1) and (2) in Definition 3.1. are fulfilled. By Proposition 2.1. it is sufficient to show that $rank_{GF(2)}[M^f_{(Y,Z)}] \geqq T$.

But this is quite obvious. Consider the submatrix M' of $M^f_{(Y,Z)}$ formed by those rows and columnes which are labelled $c_1, \ldots, c_T$ and $c'_1, \ldots, c'_T$, respectively. Using the fact that $C = \{c_i \cup c'_1;\ i = 1, \ldots, T\}$ is a sheaf of f it is easy to see that M' is a permutation matrix, *i.e.*, $rank_{GF(2)}(M') = T$. ■

According to Corollary 2.3. we can derive exponential lower bounds on the width of $\{\oplus\}$-BP$_0$ of linear length by finding sheafs of exponential size for certain subfunctions. Thus, the results in [KW89] yield the following lower bounds.

COROLLARY 3.3: *For all $n \in \mathbb{N}$ let P_n be an oblivious parity branching program of length $O(n^2)$ which computes* GAP$_{n',n}$. *Then $width(P_n) = exp(\Omega(n^{1/2}))$.* ■

COROLLARY 3.4: *The same is true for the decision problem* GAP r, *where $r \in \mathbb{N}$ is arbitrarily fixed.* ■

Moreover, all exponential lower bounds proved in [KW89] on the width of linear-length-bounded $\{\vee\}$-BP$_0$'s for the word problems of free groups and one-relator-groups of finite rank hold also for oblivious parity branching programs.

Using the construction at the end of subsection 1.4. we obtain further decision problems which cannot be computed by oblivious parity programs within linear length and polynomial width.

Fact 4: Let $g : \mathbb{N} \to \mathbb{N}$ be a nondecreasing function. Suppose that for all $n \in \mathbb{N}$ the function $f_{2n} : \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1\}$ has a sheaf with support (Y_n, Z_n) and cardinality $2^{g(n)}$, where $Y_n = \{y_1, \ldots, y_n\}$, $Z_n = \{z_1, \ldots, z_n\}$ denote the variables corresponding to the first and second argument of f_n, respectively. Then all oblivious disjunctive and parity branching programs which compute $(f^*_{2n})_{n \in \mathbb{N}}$ within linear length have width $\exp(\Omega(g(n)))$.

Proof: For all $n \in \mathbb{N}$ let f^*_{2n} be defined over the variables $Y_n \cup Z_n$ taking values from $\{0, 1, 2\}$, where $Y_n = \{y_1, \ldots, y_n\}$ and $Z_n = \{z_1, \ldots, z_n\}$ denote the variables corresponding to the first and second argument of f^*_{2n}, respectively. Further, for all $n \in \mathbb{N}$ let P_n be an oblivious disjunctive or parity branching program computing f^*_{2n}.

By Fact 3, there is a constant k and subsets $W_n \subseteq Y_n$, $W'_n \subseteq Z_n$ with $\# W_n = \# W'_n = \Omega(n)$ so that $altdepth(P_n, W_n, W'_n) \leq k$.

Let c denote the constant-"2"-assignment of $(Y_n \cup Z_n) \setminus (W_n \cup W'_n)$.

Obviously, $(f^*_{2n})^c = f_{2n}$, where $m = \# W_n$. Now the proof is a straightforward consequence of Corollary 2.4 and Lemma 3.2. ■

We obtain

COROLLARY 3.5: *Disjunctive or parity oblivious branching programs which compute the decision problem* EQ* *within linear length have width* $\exp(\Omega(n))$.

Proof: The set $\{0, 1\}^n \times \{0, 1\}^n$ is a sheaf of EQ_{2n} of support (Y_n, Z_n). ■

4. SEPARATION OF COMPLEXITY CLASSES

For all sets Ω of binary Boolean operations let $\mathcal{P}_{\text{lin}-\Omega-\text{BP}}$ denote the set of languages over a finite alphabet computable by sequences of oblivious Ω-branching programs of linear length and polynomial width. This section is devoted to the proof of the following theorem which accomplishes the separation results in [KMW89].

THEOREM 2: (1) *The complexity class* $\mathcal{P}_{\text{lin}-\text{BP}}$ *is a proper subset of all* $\mathcal{P}_{\text{lin}-\{\oplus\}-\text{BP}}$, $\mathcal{P}_{\text{lin}-\{\vee\}-\text{BP}}$, *and* $\mathcal{P}_{\text{lin}-\{\wedge\}-\text{BP}}$.

(2) *The union of the complexity classes* $\mathcal{P}_{\text{lin}-\{\oplus\}-\text{BP}}$, $\mathcal{P}_{\text{lin}-\{\vee\}-\text{BP}}$, *and* $\mathcal{P}_{\text{lin}-\{\wedge\}-\text{BP}}$ *is properly contained in* $\mathcal{P}_{\text{lin}-\{\vee,\wedge\}-\text{BP}}$.

(3) *The complexity classes* $\mathcal{P}_{\text{lin}-\{\oplus\}-\text{BP}}$, $\mathcal{P}_{\text{lin}-\{\vee\}-\text{BP}}$, *and* $\mathcal{P}_{\text{lin}-\{\wedge\}-\text{BP}}$ *are pairwise uncomparable via inclusion.*

A considerable part of the proof of Theorem 2 has been already done. Item (2) is a consequence of the fact that GAP belongs to NL and, thus, to $P \subseteq \mathcal{P}_{\text{lin}-\{\vee,\wedge\}-\text{BP}}$. Corollary 3.3 implies that GAP can be solved neither by parity nor by disjunctive oblivious programs within linear length and polynomial width. In [KMW89] it is shown that also conjunctive programs of linear length and polynomial width can not do this.

The language EQ* can be solved by conjunctive oblivious programs of linear length and polynomial width [KW89]. Hence, the complement of EQ* belongs to $\mathscr{P}_{\mathrm{lin}-\{\vee\}-\mathrm{BP}}$. Consequently, Corollary 3.5 yields that $\mathscr{P}_{\mathrm{lin}-\mathrm{BP}}$ is a proper subset of $\mathscr{P}_{\mathrm{lin}-\{\vee\}-\mathrm{BP}}$, as well as of $\mathscr{P}_{\mathrm{lin}-\{\wedge\}-\mathrm{BP}}$. As $\mathscr{P}_{\mathrm{lin}-\{\oplus\}-\mathrm{BP}}$ is closed under complement we obtain by Corollary 3.5 that neither $\mathscr{P}_{\mathrm{lin}-\{\vee\}-\mathrm{BP}}$ nor $\mathscr{P}_{\mathrm{lin}-\{\wedge\}-\mathrm{BP}}$ is subset of $\mathscr{P}_{\mathrm{lin}-\{\oplus\}-\mathrm{BP}}$. We now complete the proof of Theorem 2 by showing that

$$\mathrm{IP}^* \in \mathscr{P}_{\mathrm{lin}-\{\oplus\}-\mathrm{BP}} \setminus (\mathscr{P}_{\mathrm{lin}-\{\wedge\}-\mathrm{BP}} \cup \mathscr{P}_{\mathrm{lin}-\{\vee\}-\mathrm{BP}}).$$

LEMMA 4.1: IP* *can be computed by an oblivious parity Turing machine within logarithmic space and linear access time.*

(By Fact 2 this implies $\mathrm{IP}^* \in \mathscr{P}_{\mathrm{lin}-\{\oplus\}-\mathrm{BP}}$.)

Proof: We define the functions *equal length*

$$\mathrm{EL}_{2.n}: \{0, 1, 2\}^n \times \{0, 1, 2\}^n \to \{0, 1\}$$

and, for all i, j, $1 \leqq i \leqq j \leqq n$, *opposite one components*

$$\mathrm{OOC}_{2.n}^{i,j}: \{0, 1, 2\}^n \times \{0, 1, 2\}^n \to \{0, 1\}.$$

For all $w, w' \in \{0, 1, 2\}^n$ let $\mathrm{EL}_{2.n}(w, w') = 1 \Leftrightarrow red(w)$ and $red(w')$ have equal length, and

$\mathrm{OOC}_{2.n}^{i,j}(w, w') = 1 \Leftrightarrow w_i = w'_j = 1$ and w_i and w'_j correspond to opposite components in $red(w)$ and $red(w)$, *i.e.*,

$$i - \# \{i;\ 1 \leq k < i \wedge w_k = 2\} = j - \# \{l;\ 1 \leqq l < j \wedge w_l = 2\}.$$

Obviously, for all $w, w' \in \{0, 1, 2\}^n$ it holds (I)

$$\mathrm{IP}_{2.n}(w, w') = \mathrm{EL}_{2.n}(w, w') \wedge \left(\sum_{i,j=1}^{M} \mathrm{OOC}_{2.n}^{i,j}(w, w')\right).$$

Thus, the following nondeterministic algorithm, A, which uses procedures for solving $\mathrm{EL}_{2.n}$ and $\mathrm{OOC}_{2.n}$, computes IP* via parity-acceptation.

```
If $\mathrm{EL}_{2.n}(w, w') = 0$ output $\mathrm{IP}^*(w, w') = 0$,
else choose (nondeterministically) $i$ and $j$ from $\{1, \ldots, n\}$,
    if $\mathrm{OOC}_{2.n}^{i,j}(w, w') = 1$ output $\mathrm{IP}^*(w, w') = 1$
    else output $\mathrm{IP}^*(w, w') = 0$.
```

It is easy to verify that both decision functions $\mathrm{EL}_{2.n}$ and $\mathrm{OOC}_{2.n}^{i,j}$ can be solved by deterministic logarithmic space bounded oblivious Turing-machines. ■

It remains to prove

LEMMA 4.2: *Linear length bounded disjunctive as well as conjunctive oblivious branching programs which compute* IP* *have width* $\exp(\Omega(n))$.

We use the following more general lower bound argument. Let (Y, Z) be a balanced partition of the set $X=\{x_1, \ldots, x_n\}$ of variables. A collection R of input vectors from Σ^n is called a *rectangle* of support (Y, Z) if there are assignments $c_1, \ldots, c_s$ to Y and $c'_1, \ldots, c'_t$ to Z so that $R=\{c_i \cup c'_j;\ 1 \leqq i \leqq s,\ 1 \leqq j \leqq t\}$.

Let $f=f(x_1, \ldots, x_n):\Sigma^n \to \{0, 1\}$, *and* $b \in \{0, 1\}$. We call a rectangle $R \subseteqq \{0, 1\}^n$ of support (Y, Z) *b-rectangle* for f if $f(w)=b$ for all $w \in R$. We prove

LEMMA 4.3: *Let P be a disjunctive or conjunctive oblivious branching program computing f and* $k=altdepth(P, Y, Z)$. *Then there are b-rectangles* $R_1, \ldots, R_T$ *of f with* $T \leqq width(P)^{2k-1}$ *so that* $f^{-1}(b)=R_1 \cup \ldots \cup R_T$, *where* $b=1$ *if P is disjunctive and* $b=0$ *if conjunctive.*

Proof: Assume that P is disjunctive. (If P is conjunctive the proof can be performed in a similar way.) Fix a (Y, Z)-segmentation $\{j_1, i_2, j_2, \ldots, i_k, j_k\}$ of P. Further denote by $\mathcal{N}$ the following set of $(2k-1)$-tuples $(v_1, v'_2, v_2, \ldots, v_k, v'_k)$ consisting of nodes of P, where for each l, $1 \leqq l \leqq k$, node v_l belongs to level i_l and node v'_l belongs to level j_l.

Obviously, $\# \mathcal{N} \leqq width(P)^{2k-1}$.

For all vectors $N \in \mathcal{N}$ let $R(N)$ denote the set of those input vectors $w \in \{0, 1\}^n$ for which there is an accepting path in P passing all nodes of N.

Obviously, $f^{-1}(1)= \bigcup_{N \in \mathcal{N}} R(N)$.

It remains to show that for all $N \in \mathcal{N}$ $R(N)$ is rectangle of support (Y, Z). But this follows from a standard "Cut & Paste"-argument.

Let $N \in \mathcal{N}$ and $w \neq w' \in R(N)$ be arbitrarily fixed, and denote

$$d=w|_Y \cup w'|_Z,\ d'=w'|_Y \cup w|_Z.$$

As $\{j_1, i_2, j_2, \ldots, i_k, j_k\}$ is a (Y, Z)-segmentation of P it is easy to verify that also d and d' have accepting paths which pass all nodes of N. ■

By the arguments presented in Fact 4 the proof of Proposition 4.2 is now a straightforward consequence of the following

Fact 5: For all $n \in \mathbb{N}$ consider IP_{2n} as to be defined over $Y_n \cup Z_n$, where $Y_n=\{y_1, \ldots, y_n\}$ and $Z_n=\{z_1, \ldots, z_n\}$ are the variables corresponding to the first and second argument of IP_{2n}, respectively. Further, let $R_1, \ldots, R_T$

be a collection of b-rectangles of support (Y_n, Z_n) for IP_{2n}, where $b \in \{0, 1\}$ is arbitrarily fixed, so that $\mathrm{IP}_{2n}^{-1}(b) = R_1 \cup \ldots \cup R_T$. Then $T = \exp(\Omega(n))$.

Proof: We use a combinatorial result cited in [H&87] saying that for all rectangles $R \subseteqq \{0, 1\}^n \times \{0, 1\}^n$ of support (Y_n, Z_n)

$$\left| \#(\mathrm{IP}_{2n}^{-1}(0) \cap R) - \#(\mathrm{IP}_{2n}^{-1}(1) \cap R) \right| \leqq (\# R . 2^n)^{1/2}.$$

Hence, if R is a b-rectangle of IP_{2n} for some $b \in \{0, 1\}$ then R contains at most 2^n elements. Now, Fact 5 is a straightforward consequence of the following

Fact 6: For both $b=0$ and $b=1$ $\#(\mathrm{IP}_{2n}^{-1}(b)) \geqq 3^{n-1}$ holds.

Proof: Denote by A_n the communication matrix of IP_n with respect to (Y_n, Z_n). It is not hard to verify that A_n can be written as

$$A_n = \begin{bmatrix} A_{n-1} & A_{n-1} \\ A_{n-1} & \bar{A}_{n-1} \end{bmatrix} \quad \text{and} \quad A_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

For all $b \in \{0, 1\}$ let $\# A_{n, b} = \# \{(w, w') \in \{0, 1\}^n \times \{0, 1\}^n ; (w, w') = b\}$.

We obtain $\# A_{1, 1} = 1$, $\# A_{1, 0} = 3$, and $A_{n, b} \geqq 3 . A_{n-1, b}$ for all $n \geqq 1$ and $b \in \{0, 1\}$. Thus, $\# A_{n, 0} > \# A_{n, 1} \geqq 3^{n-1}$.

This completes the proof of Proposition 4.2. and Theorem 2. ■

ACKNOWLEDGEMENTS

I would like to thank Carsten Damm, Christoph Meinel, and Stephan Waack for helpfull discussion.

REFERENCES

[A&86] M. Ajtai, L. Babaj, P. Hajnal, J. Komlos, P. Pudlak, V. Rödel, E. Szemeredi and G. Turan, Two Lower Bounds for Branching Programs, *Proc.* 18 *ACM-STOC*, 1986, pp. 30-39.

[AM86] N. Alon and W. Maass, Meanders, Ramsey Theory and Lower Bounds for Branching Programs, *Proc. 27th ACM-STOC*, 1986, pp. 30-39.

[CH89] J. Cai and L. Hemachandra, On the Power of Parity Polynomial Time, *Proc. STACS'89*, *Springer Verlag*, *L.N.C.S.*, *No. 349*, pp. 229-239.

[Co66] H. Cobham, The Recognisation Problem for Perfect Square, *Proc. 7-th I.E.E.E. Symp. on SWAT*, 1966, pp. 78-87.

[DM89] C. Damm and Ch. Meinel. Separating Completely Complexity Classes Related to Polynomial Size Ω-Decision Trees, *Proc. of FCT'89*, *L.N.C.S.*, *No. 380*, pp. 127-136.

[H&87] A. Hajnal, W. Maass, P. Pudlak, M. Szegedy and G. Turan, Threshold Circuits of Bounded Depth, *Proc.* 28 *I.E.E.E. Symp. F.O.C.S.*, pp. 99-110.

[I87] N. Immerman, Nondeterministic Space is Closed Under Complement, *Techn. Report* 552, Yale University, 1987.

[J86] S. Jukna, Lower Bounds on the Complexity of Local Circuits, *Proc. MFCS'*86, 1986, *L.N.C.S. No. 233*, pp. 440-448.

[KMW88] M. Krause, Ch. Meinel and S. Waack, Separating the Eraser Turing Machine Classes *L*, *NL*, co-*NL* and *P*, *Proc. of M.F.C.S.'*88, *Lect. Notes in Compct. Sci, No. 324*, pp. 405-413, *Theoret. Comput. Sci.* (to appear).

[KMW89] M. Krause, S. Waack and Ch. Meinel, Separating Complexity Classes Related to Certain Input-Oblivious Logarithmic-Space Bounded Turing-Machines, *Proc. I.E.E.E. Structure & Complexity*, 1989, Eugenie, U.S.A.

[KW88] M. Krause and S. Waack, On Oblivious Branching Programs of Linear Length, *Proc. of FCT'*89, *L.N.C.S.*, *No. 380*, pp. 287-296 *Inform. Comput.* (to appear).

[M88] Ch. Meinel, The Power of Polynomial Size Ω-Branching Programs, *Proc. STACS'*88, Bordeaux, *L.N.C.S. No. 294*, pp. 81-90, *Inform. Comput.* (to appear).

[M89] Ch. Meinel, Modified Branching Programs and their Computational Power, Berlin, 1988, in *L.N.C.S. No. 370*.

[PZ83] C. Papadimitriou and S. Zachos, Two Remarks on the Power of Counting, *Proc.* 6 *th GI Conf. on Theor. Comp. Sci.*, *Springer Verlag*, *L.N.C.S.*, *No. 145*, pp. 269-276.

[Ru81] W. Ruzzo, On Uniform Circuit Complexity, *J. Comp. System Sci.*, 1981, *22*, (3), pp. 236-283.

[S87] R. Szelepcsenyi, The Method of Forcing for Nondeterministic Automata, *Bulletin of the E.A.T.C.S.*, 1987, *33*, pp. 96-99.

[W84] I. Wegener, On the Complexity of Branching Programs and Decision Trees for Clique Functions, Interner Bericht der Univ. Frankfurt, 1984, in *J. of the A.C.M.*, 1988, *35*, pp. 461-471.