INFORMATIQUE THÉORIQUE ET APPLICATIONS

M. KRAUSE
CH. MEINEL
ST. WAACK

Separating complexity classes related to certain input oblivious logarithmic space-bounded Turing machines

Informatique théorique et applications, tome 26, nº 4 (1992), p. 345-362
<http://www.numdam.org/item?id=ITA_1992__26_4_345_0>

Informatique théorique et Applications/Theoretical Informatics and Applications
(vol. 26, n° 4, 1992, p. 345 à 362)

SEPARATING COMPLEXITY CLASSES RELATED TO CERTAIN INPUT OBLIVIOUS LOGARITHMIC SPACE-BOUNDED TURING MACHINES (*)

by M. KRAUSE (1), Ch. MEINEL (2) and St. WAACK (3)

Communicated by Jean BERSTEL

Abstract. – *In the following we prove that input oblivious simultaneously linear access-time and logarithmic space-bounded nondeterministic Turing machines are more powerful than deterministic ones. Moreover, we separate all the corresponding complexity classes* $L_{0,\,\mathrm{lin}}$, $NL_{0,\,\mathrm{lin}}$, co-$NL_{0,\,\mathrm{lin}}$ *and* $P = AL_{0,\,\mathrm{lin}}$ *from each other.*

Résumé. – *Dans cet article, nous prouvons que les machines de Turing non déterministes à lecture insensible à la donnée, à temps d'accès linéaire et en space borné logarithmiquement sont plus puissantes que les machines de Turing déterministes de même nature. De plus, nous séparons les classes de complexité correspondantes les unes des autres.*

INTRODUCTION

One of the most important problems in complexity theory is to separate complexity classes (or to prove their coincidence). For example, in order to separate the classes L, NL or P one has to prove that logarithmic space-bounded nondeterministic or alternating Turing machines are more powerful than deterministic ones. In the following we investigate this question and give an affirmative answer to simultaneously linear access-time and logarithmic space-bounded *input oblivious* Turing machines (*i.e.*, Turing machines for which the order to read the input bits depends merely on the length of the input and not on the input itself). Moreover, we establish strong differences in

(*) Received December 1990, final version November 1991.
(1) Universität Dortmund, Fachbereich Informatik, Postfach 500 500, W-4600 Dortmund, Germany.
(2) Universität Trier, Fachbereich IV, Abt. Informatik, Postfach 3825, W-5500 Trier, Germany.
(3) Karl-Weierstraß-Institut für Mathematik, Mohrenstr. 39, 0-1086 Berlin, Germany.

0988-3754/92/04 345 18/$3.80/© AFCET-Gauthier-Villars

the computational power of such deterministic, nondeterministic, co-nondeterministic, and alternating Turing machines. One of the interesting consequences of these results is the proof that the technique of inductive counting [Im87, Sz87] does not work under the mentioned constraints.

Whereas this fact shows that nondeterministic as well as co-nondeterministic input oblivious, simultaneously linear access-time and logarithmic space-bounded Turing machines are less powerful than solely logarithmic space-bounded ones, the question arises whether at least each problem of L can be computed by one of these restricted machines. However, this question has to be negated. We prove that the GRAPH ACCESSIBILITY PROBLEM for graphs of outdegree 1 which, of course, belongs to L can be computed neither by input oblivious nondeterministic nor by input oblivious co-nondeterministic logarithmic space-bounded Turing machines within linear access-time. Since all problems belonging to L can be computed by input oblivious simultaneously linear access-time and logarithmic space-bounded alternating Turing machines, this shows in addition to the consequences mentioned before that input oblivious simultaneously linear access-time and logarithmic space-bounded nondeterministic and co-nondeterministic Turing machines together are not able to solve all problems computable within linear access-time and logarithmic space by input oblivious alternating Turing machines. However, the GRAPH ACCESSIBILITY PROBLEM for monotone graphs of outdegree 1 which is also p-projection complete in L [Me86] can be computed within these access-time and space restrictions already by input oblivious deterministic Turing machines [Kr91].

In order to prove these results we consider the corresponding nonuniform complexity classes which can be described by means of certain Ω-branching programs [Me88]. In detail, nonuniform, input oblivious, simultaneously linear access-time and logarithmic space-bounded deterministic, nondeterministic, co-nondeterministic, and alternating Turing machines correspond to oblivious ordinary, disjunctive, conjunctive, and alternating branching programs of polynomial size and linear length, respectively. Investigating such oblivious Ω-branching programs instead of the corresponding nonuniform Turing machines we are able to establish exponential lower bounds as well as polynomial upper bounds for the sizes of certain Ω-branching programs which imply similar bounds for the Turing machine access-time and space. The proof technique we apply to obtain our exponential lower bounds for certain oblivious Ω-branching programs of linear length generalizes that for ordinary oblivious branching programs [AM86, Kr88, KW91]. In detail, considering the SEQUENCE EQUALITY PROBLEM we prove exponential lower bounds and polynomial upper bounds for oblivious disjunctive and for oblivious conjunctive branching programs of linear length which imply the separation of all the nonuniform complexity classes under consideration. However, since nonuniform lower bounds are stronger than uniform ones, and since our upper bounds can be described uniformly we obtain similar separation results for the corresponding uniform classes. Finally, we give some exponential lower bounds for the GRAPH ACCESSIBILITY PROBLEM for graphs of outdegree 1 which prove that neither input oblivious nondeterministic nor input oblivious co-nondeterministic logarithmic space-bounded Turing machines are able to compute this problem within linear access-time.

The paper is organized as follows. In Section 1 we recall the definition of an Ω-branching program and review the relations between deterministic ($\Omega=\varnothing$), disjunctive ($\Omega=\{\vee\}$), conjunctive ($\Omega=\{\wedge\}$), and alternating ($\Omega=\{\vee, \wedge\}$) branching programs and logarithmic space-bounded deterministic, nondeterministic, co-nondeterministic and alternating Turing machines (Theorem 1), respectively. Then we introduce the restricted model of oblivious Ω-branching programs which are related to the corresponding types of input oblivious Turing machines (Theorem 2), respectively. In Section 2 we develop the technique for proving exponential lower bounds for oblivious disjunctive branching programs of linear length. Then, in Section 3 we consider the SEQUENCE EQUALITY PROBLEM and prove an exponential lower bound (Proposition 2) for oblivious disjunctive branching programs and a polynomial upper bound (Proposition 3) for conjunctive ones. Due to these bounds we separate the corresponding Turing machine classes (Theorem 4) in Section 4. The concluding Section 5 is devoted to the investigation of the GRAPH ACCESSIBILITY PROBLEM.

Generally, w.l.o.g. we assume $A \subseteq \{0,1\}^*$ for all languages A under consideration. Throughout this paper we make no difference between A and its characteristic function denoted by A, too.

1. BRANCHING PROGRAM DESCRIPTIONS

Our investigation of restricted logarithmic space-bounded deterministic, nondeterministic, co-nondeterministic, and alternating Turing machines are based on descriptions by certain types of branching programs. Following [Me88] we can relate these machines to polynomial size deterministic, disjunctive, conjunctive, and alternating branching programs, respectively. Unifying this approach we consider polynomial size Ω-branching programs, $\Omega \subseteq \mathbb{B}_2$.

In detail, a *branching program* is a directed acyclic graph where each node has outdegree 2 or 0. Nodes with outdegree 0 are called *sinks* and are labelled by Boolean constants. The remaining nodes are labelled by Boolean variables taken from a set $X=\{x_1, \ldots, x_n\}$. There is a distinguished node, called the *source*, which has indegree 0. An *Ω-branching program P* is a branching program some of whose non-sink nodes are labelled by 2-argument Boolean functions $\omega \in \Omega \subseteqq \mathbb{B}_2$ instead of Boolean variables. The Boolean values assigned to the sinks of P extend to Boolean values associated with all nodes of P in the following way: if both successor nodes v_0, v_1 of a node v of P carry the Boolean values δ_0, δ_1 and if v is labelled by a Boolean variable x_i we associate with v the value δ_0 or δ_1 according to $x_i=0$ or $x_i=1$. If v is labelled by a Boolean function ω then we associate with v the value $\omega(\delta_0, \delta_1)$. P is said to *accept* (*reject*) an input $w \in \{0,1\}^n$ if the source of P associates with 1 (0) under w. An Ω-branching program P is called a *disjunctive*, a *conjunctive*, or an *alternating* branching program if $\Omega=\{\vee\}$, $\Omega=\{\wedge\}$, or $\Omega=\{\vee, \wedge\}$, respectively. In the case $\Omega=\{\vee\}$ acceptance reduces to the existance of an accepting computation path. Ordinary branching programs correspond to $\Omega=\emptyset$.

The most important complexity measure of an Ω-branching program P is the number of its non-sink nodes, the *size* of P. By $\mathscr{P}_{\Omega-BP}$, $\Omega \subseteqq \mathbb{B}_2$, we denote the set of all languages acceptable by sequences of polynomial size Ω-branching programs.

In order to relate Turing machine classes and Ω-branching program classes, $\Omega \subseteqq \mathbb{B}_2$, we have to consider the nonuniform counterparts *L/poly*, *NL/poly*, co-*NL/poly*, and *AL/poly* of the classes *L*, *NL*, co-*NL*, and *AL* consisting of all languages $A \subseteqq \{0,1\}^*$ for which there exists a polynomial length-restricted advice $\alpha : \mathbb{N} \to \{0,1\}^*$ and a $\log n$ space-bounded deterministic, nondeterministic, co-nondeterministic, or alternating Turing machine M such that M accepts $w \# \alpha(|w|)$ iff $w \in A$.

THEOREM 1 [Me88]: *It holds*

$$\mathscr{P}_{BP}=L/poly, \; \mathscr{P}_{\{\wedge\}-BP}=\text{co-}NL/poly,$$

$$\mathscr{P}_{\{\vee\}-BP}=NL/poly, \qquad \textit{and} \qquad \mathscr{P}_{\{\vee, \wedge\}-BP}=AL/poly. \quad \blacksquare$$

Now we introduce the restricted model of input oblivious Turing machines which is the subject of this paper. A deterministic, non-deterministic, co-nondeterministic, or alternating Turing machine is said to be *input oblivious* if the order to read the input bits in the course of a computation depends merely on the length of the input and not on the input itself. We shall use a *random access input* variation of Turing machines similar to that defined in [Ru81]. In this model the Turing machine has no input head. Instead it 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. By $L_{0,\,\mathrm{lin}}$, $NL_{0,\,\mathrm{lin}}$, co-$NL_{0,\,\mathrm{lin}}$, and $AL_{0,\,\mathrm{lin}}$ we denote the classes of all languages acceptable by input oblivious, simultaneously linear access-time and logarithmic space-bounded deterministic, nondeterministic, co-nondeterministic and alternating Turing machines, respectively, where the *access-time* counts the number of entering the read state.

The notion of an input oblivious Turing machine is well-known: its input-behaviour does not depend on the course of the computation.

Nonuniform input oblivious Turing machines are related to oblivious Ω-branching programs, $\Omega \subseteq \mathbb{B}_2$. An Ω-branching program is said to be *oblivious* if it is leveled (*i.e.*, all paths from the source of the program to any one of its sink-nodes are of the same length) and if all nodes of any level are labelled either by Boolean functions $\omega \in \Omega$ or by one and the same input variable. As usual, the *width* of an Ω-branching program is the maximal size of its levels. A level which contains an input variable is called an *input level.* The *length* of an oblivious Ω-branching program is the number of its input levels. By $\mathscr{P}_{\mathrm{lin}\,\Omega-BP_0}$ we denote the class of all languages which are acceptable by sequences of polynomial size and linear length oblivious Ω-branching programs.

THEOREM 2: *It holds*

$$L_{0,\,\mathrm{lin}}/poly = \mathscr{P}_{\mathrm{lin}\; BP_0},$$

$$NL_{0,\,\mathrm{lin}}/poly = \mathscr{P}_{\mathrm{lin}\,\{\vee\}-PB_0},$$

$$\text{co-}NL_{0,\,\mathrm{lin}}/poly = \mathscr{P}_{\mathrm{lin}\,\{\wedge\}-BP_0},$$

and

$$AL_{0,\,\mathrm{lin}}/poly = \mathscr{P}_{\mathrm{lin}\,\{\vee,\,\wedge\}-BP_0}.$$

The proof can be obtained by similar arguments as of [Me88]. ■

Let us only remark that, because of the well-known equality ALogSpace $= P$ due to Chandra, Kozen, and Stockmeyer, it is not difficult to show that the class $AL_{0,\,\mathrm{lin}}/poly = \mathscr{P}_{\mathrm{lin}\,\{\vee,\,\wedge\}-BP_0}$ coincides with the class of languages acceptable by nonuniform polynomial time-bounded (deterministic) Turing machines.

PROPOSITION 1:

$$P/poly = \mathscr{P}_{\mathrm{lin}\{\vee,\wedge\}-BP_0} = AL_{0,\,\mathrm{lin}}/poly. \quad \blacksquare$$

2. THE LOWER BOUND TECHNIQUE

We start with an observation made by Alon and Maass [AM86].

Put $[n] = \{1, 2, \ldots, n\}$ and let $\underline{i} := (i_1, i_2, \ldots, i_r)$ be a sequence of elements of $[n]$. Let $S_j \subseteq [n]$, $j = 1, 2$, be two disjoint subsets of $[n]$. We say that an $\{S_1, S_2\}$-alternation occurs at index j in the sequence $\underline{i}$ if i_j belongs to S_1 (S_2) and if the minimal element i_k, $k > j$, which belongs to $S_1 \cup S_2$ is an element of S_2 (S_1). The number of indices j at which there occurs an $\{S_1, S_2\}$-alternation is called the *alternation length* of $\underline{i}$ *with respect to* $\{S_1, S_2\}$.

The following lemma is a straightforward consequence of a Ramsey-theoretic lemma given in [AM86].

LEMMA 1: *Assume that in the sequence $\underline{i}$ each $i \in [n]$ appears at most k times. Then for any preassigned partition $[n] = C_1 \dot{\cup} C_2$ of $[n]$ into two disjoint sets there are two subsets $S_j \subseteq C_j$, $j = 1, 2$, such that*

– $\# S_j \geq \# C_j . 2^{-(2k-1)}$, $j = 1, 2$, *and*

– *the alternation length of $\underline{i}$* with respect to $\{S_1, S_2\}$ *is less than or equal to* $2.k$. $\blacksquare$

Now let us return to oblivious Ω-branching programs. Let P be an oblivious Ω-branching program of length λ which computes a set $A^n \subseteq \{0, 1\}^n$. We associate with P a sequence $\underline{i} = (i_1, i_2, \ldots, i_\lambda)$ of indices, where i_j is the index of that input variable the nodes of the j-th input level of P are labelled with. The sequence $\underline{i}$ is called the *index sequence* of P. The *alternation length of P* is defined as that of $\underline{i}$.

The notion which plays the central role in our lower bound proofs is that of a *sheaf.* Sheafs are projection reductions from palindrome-like sets into the problem under consideration. Recall, a mapping

$$\pi_m: \quad \{y_1, y_2, \ldots, y_n\} \to \{x_1, \bar{x}_1, x_2, \bar{x}_2, \ldots, x_m, \bar{x}_m, 0, 1\}$$

is a *projection reduction* [SV81] from a set $M \subseteq \{0, 1\}^m$ to a set $N \subseteq \{0, 1\}^n$ iff

$$M(x_1, x_2, \ldots, x_m) = N(\pi_m(y_1), \pi_m(y_2), \ldots, \pi_m(y_n)).$$

Equivalently, this means that $M=(\pi_m^*)^{-1}(N)$, where

$$\pi_m^*:\quad \{0,1\}^m \to \{0,1\}^n$$

is the canonical map resulting from π_m.

DEFINITION: Let S_1 and S_2 be two disjoint subsets of the set $[n]$, and let $A^n \subseteq \{0,1\}^n$. $\{S_1, S_2\}$ is called a *sheaf* in A^n of *thickness* τ iff there is a projection reduction $\pi_{2\tau}$,

$$\pi_{2\tau}:\quad \{y_1, y_2, \ldots, y_n\} \to \{x_1, \bar{x}_1, x_2, \bar{x}_2, \ldots, x_{2\tau}, \bar{x}_{2\tau}, 0, 1\},$$

from the set $\mathrm{QUA}^{2\tau} := \{ww \mid w \in \{0,1\}^\tau\}$ to A^n such that

$$\pi_{2\tau}^{-1}(\{x_1, \bar{x}_1, \ldots, x_\tau, \bar{x}_\tau\}) = \{y_i \mid i \in S_1\},$$

and

$$\pi_{2\tau}^{-1}(\{x_{\tau+1}, \bar{x}_{\tau+1}, \ldots, x_{2\tau}, \bar{x}_{2\tau}\}) = \{y_i \mid i \in S_2\},$$

or vice versa.

The following lemma supplies a lower bound for oblivious disjunctive branching programs in terms of sheaves of the problems under consideration. Similar methods were developed for ordinary input oblivious branching programs in [AM86, Kr91, KW91].

LEMMA 2: *Let P_n be an oblivious disjunctive branching program of width ω and length λ deciding a set A^n. Let α be the alternation length of P_n with respect to $\{S_1, S_2\}$, where S_1 and S_2 are disjoint subsets of* $[n]$.

If $\{S_1, S_2\}$ is a sheaf in A^n of thickness τ then

$$\omega \geqq 2^{\tau/\alpha}.$$

Proof: Let $\underline{i}=(i_1, \ldots, i_\lambda)$ be the index sequence of P_n of alternation length α with respect to $\{S_1, S_2\}$, let π be the projection reduction which ensures $\{S_1, S_2\}$ to be a sheaf in A^n, and let $L_{\alpha(1)}, \ldots, L_{\alpha(\alpha)}$ be those levels of P_n where an $\{S_1, S_2\}$-alternation occurs at index $\alpha(j)$, $1 \leqq j \leqq \alpha$, in the sequence $\underline{i}$.

A set $H \subseteq \{0,1\}^\tau$ is said to satisfy the *sheaf property with respect to a node v of P_n* iff for all w, $w' \in H$ there is a computation path for $\pi^*(ww')$ from the source v_0 via v to a sink of P_n which is accepting just in case of $w=w'$.

Now, the assertion is a consequence of the following three claims:

Claim (i): $H_0 = \{0, 1\}^\tau$ satisfies the sheaf property with respect to the node v_0.

Claim (ii): If $H_i \subseteq \{0, 1\}^\tau$ satisfies the sheaf property for some node $v_i \in L_{\alpha(i)}$, $1 \leqq i < \alpha$, then there is a node $v_{i+1} \in L_{\alpha(i+1)}$ and a subset $H_{i+1} \subseteq H_i$ such that H_{i+1} satisfies the sheaf property with respect to v_{i+1}, and $\# H_{i+1} \geqq \# H_i / \omega$.

Claim (iii): If $H_{\alpha-1} \subseteq \{0, 1\}^\tau$ satisfies the sheaf property for some node $v_{\alpha-1} \in L_{\alpha(\alpha-1)}$, then

$$\omega \geqq \# L_{\alpha(\alpha)} \geqq \# H_{\alpha-1}.$$

Claim (i) is trivial. Since the proofs of claims (ii) and (iii) are similar, we outline the proof of claim (iii) only. Assume, there are two different words w, $w' \in H_{\alpha-1}$ such that there are accepting computation paths of $\pi^*(ww)$ and of $\pi^*(w'w')$ from v_0 via $v_{\alpha-1}$ having a node $v_\alpha \in L_{\alpha(\alpha)}$ in common. Since the last $\{S_1, S_2\}$-alternation occurs at $\alpha(\alpha)$, the program would accept $\pi^*(ww')$ as well as $\pi^*(w'w)$. Contradiction!

From these 3 claims we obtain

$$\omega \geqq \# L_{\alpha(\alpha)} \geqq 2^\tau / \omega^{\alpha-1},$$

and, consequently, $\omega \geqq 2^{\tau/\alpha}$. ■

The following theorem claims that the complexity of a language is high if it contains sheaves in rather general position.

THEOREM 3: *Let $s : \mathbb{N} \to \mathbb{N}$ be a nondecreasing function,* $\log n = o(s(n)) \leqq n$, *and let A, $A \subseteq \{0, 1\}^*$, be a language. Assume that for all ε, $0 < \varepsilon < 1/2$, there is a δ, $0 < \delta$, such that for infinitely many natural numbers n the following condition is fulfilled: There is a partition C_1, C_2 of $[n]$ with $\# C_1$, $\# C_2 \geqq [n/2]$, such that for any two (disjoint) subsets $C'_j \subseteq C_j$ with $\# C'_j \geqq \varepsilon . n$, $j = 1, 2$, there is a sheaf $\{S_1, S_2\}$ in A^n of thickness greater than or equal to $\delta . s(n)$ with $S_j \subseteq C'_j$.*

Then each sequence of oblivious disjunctive branching programs of length $O(n)$ which accepts A is of size $2^{\Omega(s(n))}$. In particular, it holds

$$A \notin \mathscr{P}_{\text{lin}\{\vee\}-BP_0}.$$

Proof: Let $(P_n)_{n \in \mathbb{N}}$ be a sequence of oblivious disjunctive branching programs of width ω and length $c . n$, where c is fixed. Let us pick an n for which the assumptions are fulfilled. Let i be the index sequence of P_n. Let C_1, C_2

be the partition of $[n]$ with $\# C_j \geqq [n/2]$. Obviously, there are subsets $C'_j \subseteqq C_j$, $j=1, 2$, $\# C'_j \geqq [n/4]$, such that each $i \in C'_1 \cup C'_2$ occurs in i at most $4.c$ times. Then by Lemma 1 there are disjoint subsets $C''_j \subseteqq C'_j$, $\# C''_j \geqq n.2^{-8c}$, $j=1,2$, such that the alternation length of i with respect to $\{C''_1, C''_2\}$ is bounded by $8.c$.

By the assumptions there is a $\delta>0$ and a sheaf $\{S_1, S_2\}$ in A^n of thickness greater than or equal to $\delta.s(n)$, where $S_1 \subseteqq C''_1$, and $S_2 \subseteqq C''_2$. Clearly, the alternation length of i with respect to $\{S_1, S_2\}$ is also bounded by $8.c$. By Lemma 2 it follows that $\mathrm{SIZE}(P_n) \geqq 2^{s(n).\delta/8c} = 2^{\Omega(s(n))}$. ■

3. A LOWER AND AN UPPER BOUND FOR THE SEQUENCE EQUALITY PROBLEM

In the following section we give an $\exp(\Omega(n))$ lower bound for the sizes of oblivious disjunctive branching programs of linear length which solve the SEQUENCE EQUALITY PROBLEM (Proposition 2). Additionally, we give polynomial size oblivious conjunctive branching programs of linear length (Proposition 3) which perform this task.

Let $w=(x_1, x_2, \ldots, x_{2n}) \in \{0, 1\}^{2n}$. By

$$\mathrm{red}(w)=(z_{i_1}, z_{i_2}, \ldots, z_{i_r})$$

we denote the *reduced sequence* of w which is described by the sequence $i_1, \ldots, i_r$ of those odd indices of $[2n]$, where $x_{i_j}+x_{i_j+1} \leqq 1.z_{i_j}$ is defined by $z_{i_j}=x_{i_j}+x_{i_j+1}$.

The SEQUENCE EQUALITY PROBLEM $\mathrm{SEQ}=\{\mathrm{SEQ}_n\}$ is defined by

$$\mathrm{SEQ}_n(w, w')=1 \qquad \text{iff} \quad \mathrm{red}(w)=\mathrm{red}(w').$$

for any $w, w' \in \{0, 1\}^{2n}$, $n \in \mathbb{N}$.

PROPOSITION 2: *Every oblivious disjunctive branching program of linear length which computes* SEQ_n *is of size* $2^{\Omega(n)}$. *In particular,*

$$\mathrm{SEQ} \notin \mathscr{P}_{\mathrm{lin}\{\vee\}-BP_0}.$$

Proof: Let C_1, C_2 be the partition of $[4n]$ into $C_1=[2n]$ and $C_2 \subseteqq [4n]-[2n]$. For any ε, $0<\varepsilon<1/2$, let C'_j, $j=1,2$, be two (disjoint) subsets $C'_j \subseteqq C_j$ with $\# C'_j \geqq \varepsilon.4n$. Due to Theorem 3 we have to show that there exists a sheaf $\{S_1, S_2\}$, $S_j \subseteqq C'_j$, $j=1, 2$, of thickness greater than or equal to $\delta.n$ for some $\delta>0$.

A subset $A \subseteqq [4n]$ is called *admissible* if it contains at most one of the two elements $2j-1$ and $2j$ for each j, $1 \leqq j \leqq 2n$. For $j=1, 2$ let S_j be a maximal admissible subset of C'_j. Obviously, it holds

$$\# S_j \geqq (1/2) . \# C'_j \geqq \varepsilon . 2n.$$

*W.*l.o.g. we assume $\# S_1 = \# S_2$. Let $n' := \# S_1 \geqq \varepsilon . 2n$.

Let $S_1 = \{ i_1, \ldots, i_{n'} \}$ and $S_2 = \{ i'_1, \ldots, i'_{n'} \}$. We consider the following projection reduction $\pi = \pi_{2n'}$,

$$\pi(y_i) = \begin{cases} 1 & \text{if } \{2[i/2]-1, 2[i/2]\} \cap (S_1 \cup S_2) = \varnothing, \\ x_r & \text{if } i \in S_1 \text{ and } i = i_r, \\ x_{n'+r} & \text{if } i \in S_2 \text{ and } i = i'_r, \\ 0 & \text{otherwise.} \end{cases}$$

Then we have

$$\operatorname{red}(\pi(y_1), \ldots, \pi(y_{2n})) = (x_{i_1}, \ldots, x_{i_{n'}}),$$

and

$$\operatorname{red}(\pi(y_{2n+1}), \ldots, \pi(y_{4n})) = (x_{i'_1}, \ldots, x_{i'_{n'}}).$$

Hence, it holds

$$\mathrm{SEQ}_n(\pi(y_1), \ldots, \pi(y_{2n}), \pi(y_{2n+1}), \ldots, \pi(y_{4n})) = 1$$

iff

$$(x_{i_1}, \ldots, x_{i_{n'}}) = (x_{i'_1}, \ldots, x_{i'_{n'}})$$

iff

$$\mathrm{QUA}^{2n'}(x_{i_1}, \ldots, x_{i_{n'}}, x_{i'_1}, \ldots, x_{i'_{n'}}) = 1.$$

Since $\pi^{-1}(\{1, \ldots, n'\}) = S_1$ and $\pi^{-1}(\{n'+1, \ldots, 2n'\}) = S_2$, π is a projection reduction which proves that $\{S_1, S_2\}$ is a sheaf of thickness n'. ■

COROLLARY 1: (i) *Every oblivious conjunctive branching program of linear length which computes* $\neg\, \mathrm{SEQ}_n$ *is of size* $2^{\Omega(n)}$. *In particular,*

$$\neg\, \mathrm{SEQ} \notin \mathscr{P}_{\mathrm{lin}\{\wedge\}-BP_0}.$$

(ii) *Every oblivious branching program of linear length which computes* SEQ_n *or* $\neg\, \mathrm{SEQ}_n$ *is of size* $2^{\Omega(n)}$. *In particular,*

$$\mathrm{SEQ}, \neg\, \mathrm{SEQ} \notin \mathscr{P}_{\mathrm{lin}\, BP_0}. \quad ■$$

Proof: (i) For every oblivious conjunctive branching program of linear length computing $\neg\, \mathrm{SEQ}_n$ we obtain an oblivious disjunctive one of equal size and length which computes $\neg(\neg \mathrm{SEQ}_n) = \mathrm{SEQ}_n$ if we replace conjunctive $\wedge$-nodes by disjunctive $\vee$-nodes, 1-sinks by 0-sinks and 0-sinks by 1-sinks. Hence, Proposition 2 implies the assertion.

Claim (ii) is an immediate consequence of Theorem 2, Proposition 2 and claim (i). ■

Whereas oblivious disjunctive branching programs of polynomial size and linear length do not possess enough computational power for computing SEQ the corresponding conjunctive branching programs do.

PROPOSITION 3: SEQ_n *can be computed by means of an oblivious conjunctive branching program of linear length and polynomial size*, i. e.

$$\mathrm{SEQ} \in \mathscr{P}_{\mathrm{lin}\{\wedge\}-BP_0}.$$

Proof: It is easy to check that SEQ_n can be written as

$$\mathrm{SEQ}_n = \bigwedge_{i,j=1}^{n} S_{ij},$$

where the value $S_{ij}(w, w')$ is defined, for all $(i,j) \in [n]^2$ and each $w = (x_1, x_2, \ldots, x_{2n})$, $w' = (x'_1, x'_2, \ldots, x'_{2n}) \in \{0, 1\}^{2n}$, by

$$\begin{aligned} S_{ij}(w, w') = (x_{2i-1} + x_{2i} = 2) \vee (x'_{2j-1} + x'_{2j} = 2) \\ \vee (x_{2i-1} + x_{2i} = x'_{2j-1} + x'_{2j}) \\ \vee (\#\{k \mid k < i, x_{2k-1} + x_{2k} < 2\} \\ \neq \#\{l \mid l < j, x'_{2k-1} + x'_{2k} < 2\}). \end{aligned}$$

It is not hard to verify that all the ingredients of these S_{ij} can be computed by means of input oblivious ordinary branching programs of linear length and quadratic width testing the variables x_i and x'_j in the same order. ■

In analogy with the Corollary 1 we obtain

COROLLARY 2: $\neg\,\mathrm{SEQ}_n$ *can be computed by means of an oblivious disjunctive branching program of linear length and polynomial size*, i. e.

$$\neg\,\mathrm{SEQ} \in \mathscr{P}_{\mathrm{lin}\{\vee\}-BP_0}. \quad ■$$

4. THE SEPARATION RESULT

Due to Theorem 2 and the lower and upper bounds given in Section 3 for the SEQUENCE EQUALITY PROBLEM we can separate the oblivious, simultaneously linear access-time and logarithmic space-bounded Turing machine classes $L_{0,\,\mathrm{lin}}$, $NL_{0,\,\mathrm{lin}}$, co-$NL_{0,\,\mathrm{lin}}$ and $AL_{0,\,\mathrm{lin}}$ from each other.

THEOREM 4: *It holds*

$$\begin{array}{ccccc} & \subsetneqq & NL_{0,\,\mathrm{lin}} & \subsetneqq & \\ L_{0,\,\mathrm{lin}} & & \not\subseteq \quad \not\supseteq & & AL_{0,\,\mathrm{lin}} \\ & \subsetneqq & \text{co-}NL_{0,\,\mathrm{lin}} & \subsetneqq & \end{array}$$

Proof: Trivially, it holds

$$L_{0,\,\mathrm{lin}} \subseteq NL_{0,\,\mathrm{lin}}, \text{co-}NL_{0,\,\mathrm{lin}} \subseteq AL_{0,\,\mathrm{lin}}.$$

The corresponding nonuniform separation results are a consequence of Theorem 2 and the results of Section 3.

Since nonuniform lower bounds are stronger than uniform ones, and since the upper bound of Proposition 3 can be described uniformly we obtain the claimed separation results for the uniform classes, too. ■

COROLLARY 3:

(i) $NL_{0,\,\mathrm{lin}} \subsetneqq \mathrm{NSPACE}(\log n) = NL$;

(ii) co-$NL_{0,\,\mathrm{lin}} \subsetneqq$ co-NSPACE$(\log n) = NL$.

Proof: Claim (i) and claim (ii) follow immediately from Theorem 3 and from NL = co-NL [Im87, Sz87]. ■

5. LOWER BOUNDS FOR A GRAPH ACCESSIBILITY PROBLEM

In this final section we give $\exp(\Omega(n))$ lower bounds for the sizes of oblivious disjunctive as well as of oblivious conjunctive branching programs of linear length which solve the GRAPH ACCESSIBILITY PROBLEM for directed graphs of outdegree 1 (Proposition 4 and 5). Hence, this GRAPH ACCESSIBILITY PROBLEM does not belong to $NL_{0,\,\mathrm{lin}} \cup$ co-$NL_{0,\,\mathrm{lin}}$. Since it is known to belong to the complexity class $L = \mathrm{SPACE}(\log n)$ we obtain, for example, that L is not contained in $NL_{0,\,\mathrm{lin}} \cup$ co-$NL_{0,\,\mathrm{lin}}$ (Theorem 5).

The GRAPH ACCESSIBILITY PROBLEM $\mathrm{GAP1} = \{\mathrm{GAP1}_n\}$ for directed graphs of outdegree 1 consists in the decision whether there is a path in a given directed Graph $G = (V = \{v_1, \ldots, v_n\}, E)$ of outdegree 1 which leads from node v_1 to node v_n. As usual, let G be given by its adjacency matrix $A(G)=(a_{ij})_{1 \leqq i,\, j \leqq n,\, i \neq j}$ with

$$a_{ij}=a(i,j)=\begin{cases} 1 & \text{if } (v_i, v_j) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

Then, $\mathrm{GAP1}_n : \{0, 1\}^{n(n-1)} \to \{0, 1\}$ is defined by

$$(a_{ij})_{i,j} \mapsto \begin{cases} 1 & \text{if there is a path in the graph} \\ & G \text{ of outdegree 1 from } v_1 \text{ to } v_n, \\ 0 & \text{otherwise.} \end{cases}$$

It is well-known that GAP1 can be computed by logarithmic space bounded Turing machines and that it is complete in L with respect to different reduction concepts [*see* for example Me87]. However, in the following we prove that neither input oblivious simultaneously linear access-time and logarithmic space-bounded nondeterministic Turing machines nor co-nondeterministic ones are able to compute GAP1. We will prove this assertion by establishing exponential lower bounds for the sizes of the corresponding disjunctive and conjunctive branching programs.

Let us start with a technical lemma.

LEMMA 3 [KW91]: *Let* E *be a subset of* $\{(i,j) \mid 1 \leqq i,\ j \leqq n,\ i \neq j\}$, $\# E \geqq \zeta . n(n-1)$ *with* $0 < \zeta \leqq 1$. *Let* $F \subseteq [n]$ *be a set of "forbidden" numbers such that* $1 \leqq \# F \leqq \tau . n$, *where* τ *is another constant*, $0 < \tau < 1$, *with* $\zeta - 2\tau > 0$.

Then there is a set $E' \subseteq \{1, 2, \ldots, n\}^3$ *such that*

(i) $\# E' \geqq ((\zeta - \tau)/6) . n - 1$,

(ii) $(h, i, j), (k, l, m) \in E'$ implies $\# \{h, i, j, k, l, m\} = 6$,

(iii) $(i, j, k) \in E'$ implies that $\{i, j, k\} \cap F = \varnothing$, and

(iv) $(i, j, k) \in E'$ implies $(i, j) \in E$ and $(i, k) \in E$. ■

Now we are prepared to prove our lower bounds.

PROPOSITION 4: *Every oblivious disjunctive branching program of linear length which computes* $\mathrm{GAP1}_n$ *is of size* $2^{\Omega(n)}$. *In particular,*

$$\mathrm{GAP1} \notin \mathscr{P}_{\mathrm{lin}\{\vee\}-BP_0}.$$

Proof: We shall carry out the proof on the basis of Theorem 3.

$\mathrm{GAP1}_n$ is a Boolean function depending on $n(n-1)$ Boolean variables. The index set used is $\mathscr{I} = \{(i,j) \mid (i,j) \in \{1, \ldots, n\}^2,\ i \neq j\}$.

Let C_1 and C_2 be two disjoint subsets of elements of $\mathscr{I}$ such that $\# C_i \geqq \zeta . n(n-1)$, $0 \leqq \zeta \leqq 1$, $i=1, 2$. Using Lemma 3 there is an $m=\Omega(n)$ and subsets S_1 and S_2 of C_1 and C_2, respectively, such that

- $\# S_i = 2m$, $i=1, 2$,
- $(r, s) \in S_1 \cup S_2$ implies $\{1, n\} \cap \{r, s\} = \emptyset$,
- $\# \{k \mid k$ is incident to an element (r, s) of $S_1 \cup S_2\} = 7 . m$.

Let

$$S_1 = \{(a_i, a_i'), (a_i, a_i'') \mid i=1, 2, \ldots, m\},$$

and

$$S_2 = \{(d_i, e_i), (f_i, g_i) \mid i=1, 2, \ldots, m\}.$$

Now it remains to define a projection reduction $\pi = \pi_{2m}$

$$\pi : \{y_i \mid i \in \mathscr{I}\} \rightarrow \{x_1, \bar{x}_1, \ldots, x_{2m}, \bar{x}_{2m}, 0, 1\}$$

from QUA^{2m} to $\mathrm{GAP1}_n$, with

$$\pi^{-1}(\{x_1, \bar{x}_1, \ldots, x_m, \bar{x}_m\}) = \{y_i \mid i \in S_1\},$$
$$\pi^{-1}(\{x_{m+1}, \bar{x}_{m+1}, \ldots, x_{2m}, \bar{x}_{2m}\}) = \{y_i \mid i \in S_2\}.$$

Writing $y(r, s)$ instead of y_{rs} we define the required projection reduction π by

$$\pi(y(r, s)) := \begin{cases} 1 & \text{if } (r, s) \in \{(a_i', d_i), (a_i'', f_i), \\ & (e_i, a_{i+1}), (g_i, a_{i+1}), (1, a_1)\}, \\ x_i & \text{if } (r, s) = (a_i, a_i'), \\ \bar{x}_i & \text{if } (r, s) = (a_i, a_i''), \\ x_{2m+i} & \text{if } (r, s) = (d_i, e_i), \\ \bar{x}_{2m+i} & \text{if } (r, s) = (f_i, g_i), \\ 0 & \text{otherwise}, \end{cases}$$

where $1 \leqq i \leqq m$ and $a_{m+1} := n$.

Figure 1 illustrates the projection reduction $\pi = \pi_{2m}$ in the case $m=3$. The dotted arrows depend on the literal they are labelled with. For example, the edge (a_i, a_i'') exists iff $\bar{x}_i = 1$. All other edges are fixed. We observe that the triples (a_i, a_i', a_i'') serve as switches. ■

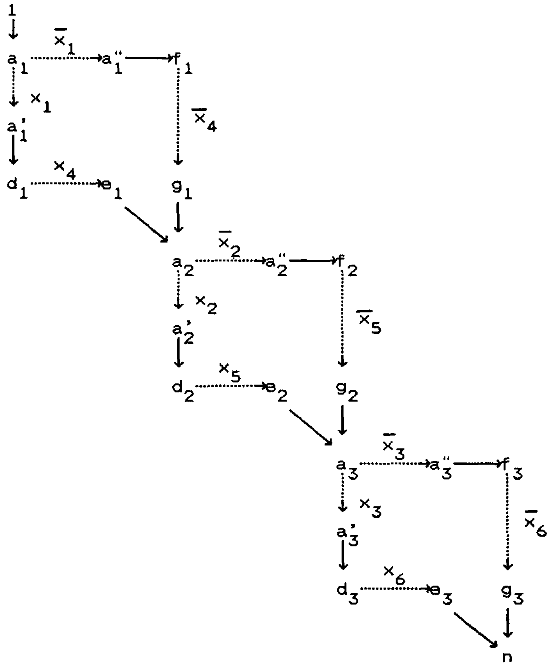

Figure 1

PROPOSITION 5: *Every oblivious conjunctive branching program of linear length which computes* $\mathrm{GAP1}_n$ *is of size* $2^{\Omega(n)}$. *In particular,*

$$\mathrm{GAP1} \notin \mathscr{P}_{\mathrm{lin}\{\wedge\}-BP_0}.$$

Proof: According to the construction in the proof of Corollary 1 of Section 3 it suffices to consider $\neg$GAP1 and to proof an exponential lower bound for oblivious disjunctive branching programs of linear length.

$\neg\mathrm{GAP1}_n$ is a Boolean function which depends on $n(n-1)$ Boolean variables with indices from the set $\mathscr{I} = \{(i,j) \mid 1 \leqq i, j \leqq n, i \neq j\}$. Again let C_1 and C_2 be two disjoint subsets of $\mathscr{I}$ such that $\# C_i \geqq \zeta . n(n-1)$, $0 < \zeta \leqq 1$, $i = 1, 2$. Due to Lemma 3 there is an $m = \Omega(n)$ and subsets S_1 and S_2 of C_1 and C_2, respectively, such that

- $\# S_1 = 2m$, $\# S_2 = 4m$,
- $(r, s) \in S_1 \cup S_2$ implies $\{1, n\} \cap \{r, s\} = \varnothing$,

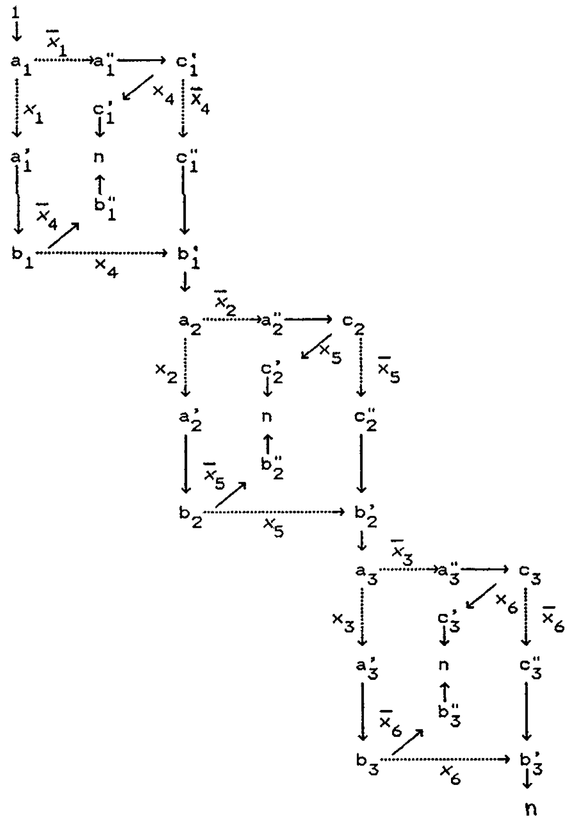

Figure 2

– $\#\{k \mid k$ is incident to an element (r, s) of $S_1 \cup S_2\} = 9 . m.$

Let

$$S_1 = \{(a_i, a_i'), (a_i, a_i'') \mid i = 1, 2, \ldots, m\},$$

and

$$S_2 = \{(b_i, b_i'), (b_i, b_i''), (c_i, c_i'') \mid i = 1, 2, \ldots, m\}.$$

Now we define the required projection reduction $\pi = \pi_{2m}$

$$\pi : \{y(i) \mid i \in \mathscr{I}\} \to \{x_1, \bar{x}_1, \ldots, x_{2m}, \bar{x}_{2m}, 0, 1\}$$

from QUA^{2m} to $\neg\mathrm{GAP1}_n$, with

$$\pi^{-1}(\{\bar{x}_1, x_1, \ldots, \bar{x}_m, x_m\}) = \{y_i \mid i \in S_1\},$$
$$\pi^{-1}(\{x_{m+1}, \bar{x}_{m+1}, \ldots, x_{2m}, \bar{x}_{2m}\}) = \{y_i \mid i \in S_2\}.$$

Setting $a_{m+1} := 1$ we define π by

$$\pi(y(r,s)) := \begin{cases} 1 & \text{if } (r,s) \in \{(1,a_1), (a_i', b_i), (b_i'', n), \\ & (a_i'', c_i), (c_i', n), (c_i'', b_i'), (b_i', a_{i+1})\} \\ x_i & \text{if } (r,s) = (a_i, a_i'), \\ \bar{x}_i & \text{if } (r,s) = (a_i, a_i''), \\ x_{2m+i} & \text{if } (r,s) \in \{(b_i, b_i'), (c_i, c_i')\} \\ \bar{x}_{2m+i} & \text{if } (r,s) \in \{(b_i, b_i''), (c_i, c_i'')\} \\ 0 & \text{otherwise,} \end{cases}$$

where $1 \leqq i \leqq m$.

Figure 2 illustrates this projection reduction π in the case $m=3$. ■

THEOREM 5:

$$L \subseteqq NL_{0,\mathrm{lin}} \cup \text{co-}NL_{0,\mathrm{lin}} \subsetneqq NL = \text{co-}NL.$$

Proof: Since GAP1 as well as $\neg$GAP1 belong to $L \subseteqq NL = \text{co-}NL$, Propositions 4 and 5 imply the claim. ■

REFERENCES

[AM86] N. ALON and W. MAASS, Meanders, Ramsey Theory and Lower Bounds, *Proc. 27th I.E.E.E. FOCS* 410-417, 1986.

[Im87] N. IMMERMAN, Nondeterministic Space is Closed Under Complement, Techn. Report 552, Yale Univ., 1987.

[KL80] R. M. KARP and R. J. LIPTON, Some Connections Between Nonuniform and Uniform Complexity Classes, *Proc. 12th A.C.M. Symp. on Theory of Computing*, 302-309, 1980.

[KMW91] M. KRAUSE, Ch. MEINEL and S. WAACK, Separating the Eraser Turing Machine Classes L_e, NL_e, co-NL_e and P_e, *Theoret. Comput. Sci.*, 1991, *86*, pp. 267-275.

[Kr91] M. KRAUSE, Lower Bounds for Depth-Restricted Branching programs, *Inform. and Comput.*, 1991, *91*, pp. 1-14.

[KW91] M. KRAUSE and S. WAACK, On Oblivious Branching Programs of Linear Length, *Inform. and Comput.*, 1991, *94*, pp. 232-249.

[Me86] Ch. MEINEL, *p*-Projection Reducibility and the Complexity Classes L (Nonuniform) and NL (Nonuniform), *Proc. 12th MFCS, Bratislava, LNCS* 233, pp. 527-535; revised and extended version in EIK, 1987, *23*, 10/11, pp. 545-558.

[Me88] Ch. MEINEL, The Power of Polynomial Size Ω-Branching Programs, *Proc. STAC'88, Bordeaux, L.N.C.S. n°* 294, pp. 81-90.

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

[SV81] S. SKYUM and L. VALIANT, A Complexity Theory Based on Boolean Algebra, *Proc. 22th I.E.E.E. F.O.C.S.*, 1981, pp. 244-253.

[Sz87] R. SZELEPCSENYI, The Method of Forcing for Nondeterministic Automata, *Bulletin of the E.A.T.C.S.*, 1987, *33*, pp. 96-99.

[We87] I. WEGENER, The Complexity of Boolean Functions, *Teubner*, Stuttgart, 1987.