Informatique théorique et applications

Klaus Kriegel
Stephan Waack

Lower bounds on the complexity of real-time branching programs

Informatique théorique et applications, tome 22, nº 4 (1988), p. 447-459.

<http://www.numdam.org/item?id=ITA_1988__22_4_447_0>

Informatique théorique et Applications/Theoretical Informatics and Applications
(vol. 22, n° 4, 1988, p. 447 à 459)

LOWER BOUNDS ON THE COMPLEXITY OF REAL-TIME BRANCHING PROGRAMS (*)

by Klaus KRIEGEL (1) and Stephan WAACK (1)

Communicated by J. BERSTEL

Abstract. – *A* $(2m)^{n/24}$ *lower bound is given for the real-time decision graph complexity of the Dyck language* D_m^*. *Furthermore, a* $2^{n/48}$ *lower bound for the real-time branching program complexity of an encoding of the Dyck language* D_2^* *is proved. Previously known similar lower bounds are* $2^{c.n}$, $c \approx 10^{-13}$, *for one-time-only branching programs (a less powerful model), and* $2^{\Omega(\sqrt{n})}$ *for real-time branching programs.*

Résumé. – *Dans cet article, nous montrons que le nombre de nœuds d'un arbre de décision en temps réel pour le langage de Dyck* D_m^* *est borné inférieurement par* $(2m)^{n/24}$. *On donne également une borne inférieure en* $2^{n/48}$ *pour la complexité des programmes temps réel pour un codage du langage de Dyck* D_2^*. *Les bornes précédemment connues étaient en* $2^{c.n}$ *avec* $c \approx 10^{-13}$ *pour les programmes à un seul branchement (un modèle moins puissant) et en* $2^{\Omega(\sqrt{n})}$ *pour les programmes à branchements temps réel.*

1. INTRODUCTION

A Σ-*decision graph* T, for Σ a finite alphabet, is a directed acyclic graph with the following properties:

– it has exactly one source, i. e. vertex with indegree 0;

– every vertex has outdegree 0 or $|\Sigma|$;

– sinks, i. e. vertices with outdegree 0, are labelled 0 or 1;

(*) Received in July 1986, revised in August 1987.

(1) Karl-Weierstraβ-Institut für Mathematik der Akademie der Wissenschaften der DDR, DDR 1086 Berlin, Mohrenstr. 39, Postfach 1304.

Informatique théorique et Applications/Theoretical Informatics and Applications
0296-1598 88/04 447 13/$3.30/© Gauthier-Villars

— branching vertices, i. e. vertices with outdegree $|\Sigma|$ are labelled i, for some $1 \leqq i \leqq n$, and the $|\Sigma|$ outgoing edges are labelled with the elements of Σ, where each $\sigma \in \Sigma$ occurs exactly once.

To every word $w_1 w_2 \ldots w_n = w \in \Sigma^n$ there corresponds a unique path p_w leading from the source to a sink (at a branching vertex labelled i it chooses the edge labelled $w_i \in \Sigma$). The decision graph T decides a set $L^{(n)} \in \Sigma^n$ iff for every $w \in \Sigma^n$ the sink at the end of the path p_w is labelled $L^{(n)}(w)$. [Throughout this work we make no difference between $L^{(n)}$ and its characteristic function denoted by $L^{(n)}$, too.] The size of a decision graph T, which we denote by SIZE (T), is the number of branching vertices of T.

A decision graph is said to the *real-time*, if for every w the length of p_w is less than or equal to n.

A $\{0, 1\}$-decision graph is called a *branching program*. Branching programs compute Boolean functions. They have been studied more extensively than decision graphs over a larger alphabet, although the latter ones are more adapted in many cases. However, the difference is not important.

The logarithm of the size of a smallest decision graph deciding a language is a lower bound on space requirement for any reasonable sequential model of computation.

Nonlinear lower bounds ($\Omega(n^2/\log^2 n)$) have already been given by Nechiporuk [7] (in the more general framework of contact schemes). In order to obtain larger lower bounds, bounded width branching programs have been studied in Borodin-Dolev-Fich-Paul [2], Chandra-Furst-Lipton [4], Pudlak [8] and Yao [11]. Another restricted model is the *one-time-only* branching program studied by Wegener [10], Zak [12], Dunne [5] and Ajtai *et al.* [1]. It imposes the constraint that any computation path may examine every input letter at most once. Wegener, Zak and Dunne have given $2^{\Omega(\sqrt{n})}$ lower bounds, whereas in [1] a 2^{cn} lower bound has been proved, for $c \approx 10^{-13}$. The Boolean functions studied in these works are determined by graph properties. The property given by Hajnal, Szemeredi and Turan in [1] is "G has an even number of triangles".

Clearly, the real-time model is more powerful than the one-time-only model. Again Zak [13] has proved a $2^{\Omega(\sqrt{n})}$ lower bound. In this paper we study the real-time decision graph complexity of the Dyck languages D_m^*. In view of the well-known Chomsky-Schützenberger Theorem, they are very interesting context-free languages. It is known that the membership problem for the Dyck language D_m^* is identical with the word problem of the free group on m distinct generators.

Comparing the Dyck language D_m^* with the graph property in [1] from the complexity point of view, we have:

(*a*) "G has an even number of triangles" as well as D_m^* (*see* [6]) can be decided within logspace;

(*b*) D_m^* can be decided in realtime, whereas this is not clear for "G has an even number of triangles".

In section 2 we give several definitions from the theory of partially ordered sets. We make use of them in section 3, where a general lower bound for real-time decision graphs is derived. This result is applied in section 4 to prove $(2m)^{n/24}$ lower bounds for real-time decision graphs of the Dyck languages D_m^*. Finally, in section 5 we encode D_2^* and show a $2^{n/48}$ lower bound for real-time branching programs. As we are only interested in some basic grouptheoretic properties of D_m^*, we shall consider D_m^* to be the word problem of the free group of rank m.

2. PRELIMINARIES

A subset of a partially ordered set (poset) P is *descending* iff $x \in S$ and $y \leqq x$ imply $y \in S$. An *ascending subset* of P is just a descending subset of the dual poset P^* obtained by reversing the order relation. If S is any subset of P, then

$$\mathrm{Cl}_P(S) := \{ x \mid \exists y \in S : x \leqq y \} \qquad (\text{resp. } \overline{\mathrm{Cl}}_P(S) := \{ x \mid \exists y \in S : x \geqq y \})$$

is the smallest descending (resp. ascending) subset of P containing S. $\mathrm{Cl}_P(S)$ is often called the *closure* of S in P.

A *chain* C is a totally ordered subset of P. A subset Q of P is said to be a *cutset* iff $C \subseteq P$, C maximal chain, implies $Q \cap C \neq \emptyset$.

The *product* $P \times Q$ of partially ordered sets P and Q is the set of all ordered pairs (p, q), where $p \in P$ and $q \in Q$, endowed with the order $(p, q) \leqq (r, s)$ whenever $p \leqq r$ and $q \leqq s$. The least upper bound $(p, q) \vee (r, s)$ exists iff both $p \vee r$ and $q \vee s$ exist. If they exist, then $(p, q) \vee (r, s) = (p \vee r, q \vee s)$. The dual assertion for the greatest lower bound also holds.

The product (f, g) of order-preserving maps $f: P \to P'$ and $g: Q \to Q'$ is the map $P \times Q \to P' \times Q'$ which assings to each pair (p, q) the pair $(f(p), g(q))$. Clearly, (f, g) is order preserving.

In line with [3] let us introduce the partially ordered set $\mathrm{Cond}(\Sigma^n)$ where Σ is a finite alphabet. The elements of $\mathrm{Cond}(\Sigma^n)$, the so called *conditions*, are all partial maps from $\{1, \ldots, n\}$ into Σ including the empty condition $\hat{0}$ which is defined nowhere. Condition c_1 is a *subcondition* of condition c_2 (we write $c_1 \leqq c_2$) iff the graph of c_1 is contained in the graph of c_2. The *graph of a condition* c is defined to be $\operatorname{graph}(c) := \{(i, \sigma) \mid i \in \operatorname{dom} c \text{ and } c(i) = \sigma\}$, where $\operatorname{dom} c = \{i \mid c(i) \text{ is defined}\}$.

It is very easy to check that Cond (Σ^n) is the poset of faces of a simplicial complex. First this means that two conditions c_1 and c_2 have a greatest lower bound $c_1 \wedge c_2$. We remark that graph $(c_1 \wedge c_2) = \operatorname{graph}(c_1) \cap \operatorname{graph}(c_2)$. Secondly each segment $[\hat{0}, c] = \{c' \mid \hat{0} \leqq c' \leqq c\}$ is isomorphic to the Boolean algebra of all subsets of graph (c). Consequently, if $c' \in [\hat{0}, c]$, then there is the complement $c - c'$ of c' in c.

The maximal elements of $\operatorname{Cond}(\Sigma^n)$ are the ordinary words of length n over Σ, i.e. the elements of Σ^n. We extend the natural length function for words to a *rank function* r of the entire poset defining

$$r(c) := |\operatorname{dom} c| = |\operatorname{graph}(c)|$$

for any condition c. Then, of course, the empty condition $\hat{0}$ has rank 0, all atomic conditions (i.e., all conditions covering the empty condition) have rank 1, all maximal conditions (i.e., all words of length n) are of rank n.

If two conditions c_1 and c_2 have an upper bound, then they are called *compatible*. In that case they have a least upper bound $c_1 \vee c_2$. Obviously graph $(c_1 \vee c_2) = $ graph $(c_1) \cup \operatorname{graph}(c_2)$ holds.

We call a condition c a *piece* iff the domain of c is a segment $[i, j] \subseteq \{1, 2, \ldots, n\}$. If $c \in \operatorname{Cond}(\Sigma^n)$ is a piece, $\operatorname{dom} c = [i, j]$, then we associate with c a word $w_c \in \Sigma^{j-i+1}$ by $w_c(k) = c(k+i-1)$. Condition $c' \leqq c$ is said to be a *part* of condition c iff c' is a maximal element of $\{c'' \mid c'' \leqq c,\ c'' \text{ is a piece}\}$. Clearly every condition is the least upper bound of its parts.

For n, $m \geqq 1$, there is a natural order-preserving isomorphism $\operatorname{Cond}(\Sigma^n) \times \operatorname{Cond}(\Sigma^m) \to \operatorname{Cond}(\Sigma^{n+m})$. If we denote by $c_1 c_2$ the image of (c_1, c_2) under this map, then dom $c_1 c_2 = \operatorname{dom} c_1 \cup (\{n\} + \operatorname{dom} c_2)$, and

$$c_1 c_2(i) := \begin{cases} c_1(i), & \text{if } i \leqq n \\ c_2(i-n), & \text{if } i > n \end{cases}$$

provided that $i \in \operatorname{dom} c_1 c_2$.

Let T be a Σ-decision graph. We assign to each edge e leading from vertex v_1 to vertex v_2 an element $(i, \sigma) \in \{1, 2, \ldots, n\} \times \Sigma$ where i is the label of v_1 and σ is the label of e itself. Extending this assignment each path p in T is mapped onto a subset $c(p)$ of $\{1, 2, \ldots, n\} \times \Sigma$. A path p is called a *computation path* iff p starts at the source and $c(p)$ is the graph of a condition. It is plain that if p is a computation path, if $w \in \Sigma^n$, and if $c(p) \leqq w$, then the path p_w along which the word is computed (*see* introduction) contains p as a prefix, i. e. as an initial subpath. In particular, we have that $c(p_w) = w$.

3. GENERAL LOWER BOUNDS

Let $L^{(n)}$ be a nonempty subset of Σ^n. The following three definitions are slight modifications of those occuring in [9], [10], respectively.

$L^{(n)}$ is called *k-sensitive* if for every condition c of rank $k-1$ there are words $w_1 \geqq c$ and $w_2 \geqq c$ such that $w_1 \in L^{(n)}$ and $w_2 \notin L^{(n)}$. A word $w \in L^{(n)}$ is said to be *critical* iff all other words $w' \in \Sigma^n$ with $r(w \wedge w') = n-1$ do not belong to $L^{(n)}$.

The language $L^{(n)}$ is called critical iff all of its elements are critical.

LEMMA 1: *If T is a real-time Σ-decision graph which decides $L^{(n)}$, and if a word $w \in L^{(n)}$ is critical, then the computation path in T corresponding to w examines each input exactly once.*

Proof: Let p be the path corresponding to w. Assume that there is an input which is not examined exactly once. We shall derive a contradiction. Since T is real-time, $r(c(p)) \leqq n-1$. As w is critical, there is a word w' with $w' \geqq c(p)$ and $w' \notin L^{(n)}$. Contradiction to the fact, that p is also computation path corresponding to w'. □

LEMMA 2: (i) *If T is a Σ-decision graph deciding $L^{(n)}$, and if $L^{(n)}$ is k-sensitive, then no computation path of length less than k leads to a sink.*

(ii) *If moreover T is real-time and $L^{(n)}$ critical, then each computation path of length k examines each input at most once. Consequently, in this case all paths of length k are computation paths.*

Proof: (i) Exactly the definition of k-sensitivity.

(ii) Assume that there is a computation path p of length k which examines an input at least twice. Then $r(c(p)) \leqq k-1$.

Since $L^{(n)}$ is k-sensitive, there is a word $w \geqq c(p)$ belonging to $L^{(n)}$. Let q be the computation path in T corresponding to w. Then p is prefix of q. Consequently $r(c(q)) \leqq n-1$, because T is real-time. Contradiction to lemma 1. □

The conditions c_1 and c_2 ar e said to be $L^{(n)}$-*equivalent* [we write $c_1 \sim c_2 \bmod L^{(n)}$] iff

(i) dom $c_1 = \text{dom } c_2$;

(ii) if w_1 and w_2 are words, if $w_1 \geqq c_1$ and $w_2 \geqq c_2$, and if

$$w_1 - c_1 = w_2 - c_2, \quad \text{then } w_1 \in L^{(n)} \quad \Leftrightarrow \quad w_2 \in L^{(n)}.$$

By definition two conditions are equivalent only if they have the same rank.

LEMMA 3: *Let T be any real-time Σ-decision graph computing $L^{(n)}$. Furthermore, let $L^{(n)}$ be $2k+1$-sensitive and critical, where $2k+1 \leqq n$.*

If p_1 and p_2 are computation paths of length k in T leading to one and the same node v, then $c(p_1) \sim c(p_2) \bmod L^{(n)}$.

Proof: (i) Let $c_1 := c(p_1)$, $c_2 := c(p_2)$. We claim that dom $c_1 = \text{dom } c_2$.

Define c'_1 (resp. c'_2) to be the maximal subcondition of c_1 (resp. c_2) which is compatible with c_2 (resp. c_1).

Then

$$\text{dom}(c_1 - c'_1) = \text{dom}(c_2 - c'_2),$$
$$\text{dom}(c'_1 \vee c_2) = \text{dom}(c_1 \vee c'_2)$$

and

$$r(c'_1 \vee c_2) = r(c_1 \vee c'_2) \leqq 2k.$$

Furthermore $((c_1 \vee c'_2) - (c_1 - c'_1)) \vee (c_2 - c'_2) = c'_1 \vee c_2$.

Since $L^{(n)}$ is $2k+1$-sensitive, there is a word $w_1 \geqq c_1 \vee c_2$ belonging to $L^{(n)}$. The property $w_1 \geqq c_1 \vee c'_2 \geqq c_1$ implies that the path in T traced under w_1 to a sink labelled 1 equals $p_1 q$, where q leads from v to the sink. By lemma 1 $p_1 q$ examines each input exactly once. Hence $c(q) = w_1 - c_1$, and dom $c_1 = \{1, 2, \ldots, n\} - \text{dom}(c(q))$.

Let $w_2 := (w_1 - (c_1 - c'_1)) \vee (c_2 - c'_2)$. Obviously w_2 is a word. An easy calculation reveals that $w_2 \geqq c'_1 \vee c_2 \geqq c_2$ as well as $w_2 \geqq w_1 - c_1 = c(q)$. Hence $p_2 q$ is the path in T traced under w_2. Therefore $w_2 \in L^{(n)}$ and again $p_2 q$ examines each input exactly once. Thus dom $c_2 = \{1, \ldots, n\} - \text{dom}(c(q))$. So as claimed dom $c_1 = \text{dom } c_2$.

(ii) Let w_1 and w_2 be words such that $w_1 \geqq c_1$, $w_2 \geqq c_2$, $w_1 - c_1 = w_2 - c_2$, and $w_1 \in L^{(n)}$. We claim that $w_2 \in L^{(n)}$. Since $w_1 \geqq c_1$, $p_1 q$ is the path traced under w_1 to a sink labelled 1. Analogously to (i) we get that $p_2 q$ is the computation path corresponding to w_2. Hence $w_2 \in L^{(n)}$. $\square$

In order to formulate the general lower bound for real-time decision graphs we need some further notations. Let $\mathrm{Cl}(S)$ denote the closure of S with reference to $\mathrm{Cond}(\Sigma^n)$. The notation $\overline{\mathrm{Cl}}(S)$ also refers to $\mathrm{Cond}(\Sigma^n)$. Let $K^{(n)} \subseteq \Sigma^n$ be another non-empty subset and Q be a cutset of $\mathrm{Cl}(K^{(n)})$. We define:

$$[c]:=\{c' \in \mathrm{Cond}(\Sigma^n) \mid c \sim c' \bmod L^{(n)}\},$$

for any condition c,

$$m_1(L^{(n)}, Q):=\max\{\,|[c] \cap Q| \mid c \in Q\,\}$$
$$m_2(K^{(n)}, Q):=\max\{\,|\overline{\mathrm{Cl}}(\{c\}) \cap K^{(n)}| \mid c \in Q\,\}.$$

THEOREM 1: *Let $L^{(n)}$ be $2k+1$-sensitive $(2k+1<n)$ and critical. Assume that $K^{(n)}$ is nonempty, and Q is a cutset of $Cl(K^{(n)})$ such that $c \in Q$ implies $r(c) \leqq k$.*

If T is a real-time Σ-decision graph deciding $L^{(n)}$, then

$$\mathrm{SIZE}(T) \geqq \frac{|K^{(n)}|}{m_1(L^{(n)}, Q)\, m_2(K^{(n)}, Q)}.$$

Proof: Let π be the set of all computation paths p of T such that:

- length of p is less than or equal to k;
- $c(p) \in Q$;
- $c(p') \notin Q$ for each proper prefix p' of p.

By lemma 2 $r(c(p))$ equals the length of p, for all $p \in \pi$. It is easy to see that for all $p, q \in \pi$, $p \neq q$, $c(p)$ and $c(q)$ are not compatible with each other and consequently $\overline{\mathrm{Cl}}(c(p)) \cap \overline{\mathrm{Cl}}(c(q)) = \varnothing$.

Let $w \in K^{(n)}$, p_w the unique computation path corresponding to w. Again by lemma 2, the length of p_w is greater than k. Since Q is a cutset of $\mathrm{Cl}(K^{(n)})$ and $r(c) \leqq k$ for any $c \in Q$, there is a $p \in \pi$ such that p is a prefix of p_w. According to definitions each $p \in \pi$ is connected to at most $m_2(K^{(n)}, Q)$ words of $K^{(n)}$ in the above described way. Hence $|\pi| \geqq |K^{(n)}|/m_2(K^{(n)}, Q)$.

Let V be the set of non-sink vertices of T. We consider the map $\Theta : \pi \to V$ which assigns to each p its terminal node. We claim that:

$$\mathrm{SIZE}(T) = |V| \geqq |\mathrm{image}\,\Theta| \geqq \frac{|K^{(n)}|}{m_1(L^{(n)}, Q)\, m_2(K^{(n)}, Q)}$$

Assume that $v \in V$ is a vertex such that $\Theta^{-1}(v)$ is maximal, and p is a fixed path mapped onto v via Θ. Then by lemma 3

$$\Theta^{-1}(v) \subseteq \{ q \in \pi \mid c(q) \sim c(p) \bmod L^{(n)} \}.$$

Hence

$$|\Theta^{-1}(v)| \leqq |[c(p)] \cap Q| \leqq m_1(L^{(n)}, Q).$$

Since $\{ \Theta^{-1}(v) \mid v \in \operatorname{image} \Theta \}$ is a partition of π, our claim follows. □

4. LOWER BOUNDS FOR DECISION GRAPHS

Let $X = \{ x_1, \ldots, x_m \}$, $m \geqq 2$. Assume that $\langle X \rangle$ is the free group on X. Then each element of $\langle X \rangle$ can be represented as a word over $\underline{X} = X \cup \{ x_1^{-1}, \ldots, x_m^{-1} \}$. Given two words w_1 and w_2 over $\underline{X}$, it is well-known that w_1 is *freely equal* to w_2, i. e. w_1 and w_2 define one and the same element in $\langle X \rangle$, iff w_1 can be transformed into w_2 by a finite sequence of the following rules: (1) replace $x_i x_i^{-1}$ by 1, (2) replace $x_i^{-1} x_i$ by 1, (3) the inverse of (1), (4) the inverse of (2), where 1 is the empty word ($i = 1, 2, \ldots, m$).

The *word problem* of $\langle X \rangle$ is the following formal language:

$$W(\langle X \rangle) := \{ w \in \underline{X}^* \mid w = 1 \text{ in } \langle X \rangle \}.$$

If

$$W^{(n)}(\langle X \rangle) := W(\langle X \rangle) \cap \underline{X}^n,$$

then $W^{(n)}(\langle X \rangle) = \varnothing$ if n is odd. Hence we assume n to be even.

Aword w is called *reduced* iff neither rule (1) nor rule (2) can be applied to w. Obviously, each group element of $\langle X \rangle$ has a unique reduced representation over $\underline{X}$. It is plain that X is a set of reduced words, and $W^{(n)}(\langle X \rangle)$ as well as X^* are closed under cyclic permutation.

LEMMA 4: (i) *Two reduced words are equal iff they are freely equal.*

(ii) *Let* $c \in \operatorname{Cond}(\underline{X}^n)$, $r(c) \leqq n/2$.

Then there is a condition $c' \geqq c$ *such that:*

– $r(c') \leqq 2r(c)$;

– *if u is a part of c', then the word w_u associated with u is freely equal to* 1.

Proof: Easy. □

COROLLARY: *$W^{(n)}(\langle X \rangle)$ is $(n/2)+1$-sensitive.*

LEMMA 5: *$W^{(n)}(\langle X \rangle)$ is critical.*

Proof: The assertion follows from the trivial observation that

$$x_{i_1}^{e_1} x_{i_2}^{e_2} \dots x_{i_n}^{e_n} = 1$$

in $\langle X \rangle$, for $e_i = \pm 1$, implies

$$x_{i_j}^{e_j} = (x_{i_1}^{e_1} \dots x_{i_{j-1}}^{e_{j-1}})^{-1} (x_{i_{j+1}}^{e_{j+1}} \dots x_{i_n}^{e_n})^{-1} \quad \text{in } \langle X \rangle. \quad \square$$

Throughout the remainder of this section let

$$\mathrm{Cl}(S) := \mathrm{Cl}_{\mathrm{Cond}(\underline{X}^n)}(S)$$
$$\overline{\mathrm{Cl}}(S) := \overline{\mathrm{Cl}}_{\mathrm{Cond}(\underline{X}^n)}(S),$$

for $S \subseteq \mathrm{Cond}(\underline{X}^n)$.

Define $Q_{n,k} := \{ c \in \mathrm{Cl}(X^n);\ r(c) = k \}$.

LEMMA 6: (i) *The sets $Q_{n,k}$ are cutsets of* $\mathrm{Cl}(X^n)$.

(ii) *Assume that n is divisible by $6l$, and $l \geqq 2$ is an integer. Then* $m_1(W^{(n)}(\langle X \rangle), Q_{n, n/2l}) \leqq m^{n/3l}$.

(iii) $m_2(X^n, Q_{n,k}) = m^{n-k}$.

Proof: Claim (i) and (iii) are obvious.

(ii) Let $c_1, c_2 \in Cl(X^n)$, $r(c_1) = r(c_2) = (n/2l)$, $c_1 \neq c_2$ such that $c_1 \sim c_2$ mod $W^{(n)}(\langle X \rangle)$. Let $I := \mathrm{dom}\ c_1 = \mathrm{dom}\, c_2$. Since both $W^{(n)}(\langle X \rangle)$ and X^n are invariant under cyclic permutations, we can restrict ourselves to the case that $|I \cap \{1, 2, \dots, n/3\}| \geqq (1/6l)n$.

It is sufficient to show that there is an $i \in I$, $i > n/3$, such that $c_1(i) \neq c_2(i)$. Suppose that this is not the case. We shall derive a contradiction. Let $c_1 = c'_1 d$ and $c_2 = c'_2 d$, where $c'_1, c'_2 \in \mathrm{Cond}(\underline{X}^{n/3})$ and $d \in \mathrm{Cond}(\underline{X}^{2n/3})$. Let $w'_1, w'_2 \in X^{n/3}$ be words such that

$$c'_1 \leqq w'_1, \quad c'_2 \leqq w'_2 \qquad \text{and} \qquad w'_2 - c'_2 = w'_1 - c'_1.$$

Let $d' \in \mathrm{Cond}(\underline{X}^{2n/3})$, $d' \geqq d$ be a condition the existence of which is ensured by lemma 4(ii). Since $r(d') \leqq (2/3l)^n \leqq n/3$, all parts of d' are associated with words which are freely equal to 1, and $|w'_1| = n/3$, there is a $w_2 \in \underline{X}^{2n/3}$ such that $w_2 \geqq d'$ and $w'_1 w_2 = 1$ in X. The $W^{(n)}$ $(\langle X \rangle)$-equivalence of c_1 and c_2 implies $w'_2 w_2 = 1$ in $\langle X \rangle$. Then w'_1 and w'_2 are freely equal and consequently equal. It follows that $c'_1 = c'_2$. Contradiction to $c_1 \neq c_2$. $\square$

THEOREM 2: *Assume that T is a real-time $\underline{X}$-decision graph deciding $W^{(n)}(\langle X \rangle)$, for 12 dividing n and $|X| \geqq 2$.*

Then $\mathrm{SIZE}(T) \geqq |\underline{X}|^{tn/12} \geqq |\underline{X}|^{n/24}$, *where* $t = \ln|X|/(\ln|X| + \ln 2)$.

Proof: Applying theorem 1 for

$$L^{(n)} = W^{(n)}(\langle X \rangle), \quad K^{(n)} = X^n \quad \text{and} \quad Q = Q_{n,\, n/4}$$

We obtain

$$\mathrm{SIZE}(T) \geqq \frac{m^n}{m^{n/6}\, m^{n-n/4}} = m^{n/12} = (2m)^{(\ln m/\ln 2m) . n/12}. \quad \square$$

5. LOWER BOUNDS FOR BRANCHING PROGRAMS

Let

$$X = \{x_1, x_2\}, \quad X^{-1} = \{x_1^{-1}, x_2^{-1}\}, \quad \underline{X} = X \cup X^{-1}, \quad \varphi: \underline{X} \to \{0, 1\}^2$$

defined by

$$x_1 \mapsto 10, \quad x_2 \mapsto 01, \quad x_1^{-1} \mapsto 11, \quad x_2^{-1} \mapsto 00.$$

First we observe that φ can be extended to $\varphi_n: \underline{X}^n \to \{0, 1\}^{2n}$ in the straightforward way. Second, we can extend φ to an injective order-preserving map $\varphi_{1*}: \mathrm{Cond}(\underline{X}^1) \to \mathrm{Cond}(\{0, 1\}^2)$ simply by setting $\varphi_{1*}(\hat{0}) = \hat{0}$.

We know from section 2 that $\mathrm{Cond}(\underline{X}^n)$ is isomorphic to $\mathrm{Cond}(\underline{X}^1) \times \ldots_{(n)} \ldots \times \mathrm{Cond}(\underline{X}^1)$, and $\mathrm{Cond}(\{0, 1\}^{2n})$ is isomorphic to $\mathrm{Cond}(\{0, 1\}^2) \times \ldots_{(n)} \ldots \times \mathrm{Cond}(\{0, 1\}^2)$. We define the order-preserving injection $\varphi_{n*}: \mathrm{Cond}(\underline{X}^n) \to \mathrm{Cond}(\{0, 1\}^{2n})$ to be $(\varphi_{1*}, \ldots, \varphi_{1*})$. It is obvious that $r(\varphi_{n*}(c)) = 2r(c)$. We remark that φ_{n*} restricted to $\underline{X}^n$ is identical to φ_n.

LEMMA 7: *Assume that* c *and* c' *belong to the closure of* X^n *in* $\mathrm{Cond}(\underline{X}^n)$, *i. e. to* $\mathrm{Cl}(X^n)$. *Then* $\varphi_{n*}(c \wedge c') = \varphi_{n*}(c) \wedge \varphi_{n*}(c')$.

Proof: Notice that the Hamming-distance of $\varphi(x_1)$ and $\varphi(x_2)$ is equal to 2. This implies that

$$\varphi_{1*}(d \wedge d') = \varphi_{1*}(d) \wedge \varphi_{1*}(d') \quad \text{for} \quad d, d' \in X \cup \{\hat{0}\} \subseteq \mathrm{Cond}(\underline{X}^1).$$

It is known that an isomorphism of partially ordered sets respects least upper bounds as well as greatest lower bounds. Let

$$c = c_1 c_2 \ldots c_n, \quad c' = c'_1 c'_2 \ldots c'_n \quad \text{where} \quad c_j, c'_i \in X \cup \{\hat{0}\},$$

for $i = 1, \ldots, n$. Then

$$c \wedge c' = (c_1 \wedge c'_1)(c_2 \wedge c'_2) \ldots (c_n \wedge c'_n).$$

Consequently

$$\varphi_{n*}(c \wedge c') = \varphi_{1*}(c_1 \wedge c'_1)\,\varphi_{1*}(c_2 \wedge c'_2)\ldots\varphi_{1*}(c_n \wedge c'_n)$$
$$= \varphi_{n*}(c) \wedge \varphi_{n*}(c'). \quad \square$$

We define $f_{2n} := W^{(n)}(\langle X \rangle) . \varphi_n^{-1}$. (Remember, that we identify a formal language with its characteristic function.)

LEMMA 8: *The Boolean function f_{2n}, for even n, is $(n/2)+1$ sensitive and critical.*

Proof: Immediate consequence of lemma 4 and 5, resp. $\square$

Throughout the remainder of this section let $\mathrm{Cl}_1(A)$ denote the closure of A in $\mathrm{Cond}(\underline{X}^n)$, whereas let $\mathrm{Cl}_2(B)$ denote the closure of B in $\mathrm{Cond}(\{0, 1\}^{2n})$. In section 4 we considered the subsets

$$Q_{n,k} = \{ c \in \mathrm{Cl}_1(X^n) \mid r(c) = k \}, \qquad k \geqq 1,$$

of $\mathrm{Cond}(\underline{X}^n)$. Now we are interested in cutsets of the closure of $\varphi_{n*}(\underline{X}^n)$ in $\mathrm{Cond}(\{0, 1\}^{2n})$. Clearly, the sets $\varphi_{n*}(Q_{n,k})$ do not have this property. Define

$$K^{(2n)} := \varphi_{n*}(X^n)$$
$$R_{n,k} := \mathrm{Cl}_2(\varphi_{n*}(Q_{n,k})) - \mathrm{Cl}_2(\varphi_{n*}(Q_{n,k-1})).$$

LEMMA 9: (i) *The sets $R_{n,k}$ are cutsets of* $\mathrm{Cl}_2(K^{(2n)})$.

(ii) *If $6l$ divides n, for an integer $l \geqq 2$, then*

$$m_1(f_{2n}, R_{n, n/2l}) \leqq 2^{n/3l}$$

(iii) $m_2(K^{(2n)}, R_{n,k}) \leqq 2^{n-k}$.

Proof: It follows from the definition of $R_{n,k}$ that if $d \in R_{n,k}$ then there is a $c \in Q_{n,k}$ such that $d \leqq \varphi_{n*}(c)$. We show first, that this condition c is uniquely determined. If c, $c' \in Q_{n,k}$ such that $d \leqq \varphi_{n*}(c)$, and $d \leqq \varphi_{n*}(c')$, then

$$d \leqq \varphi_{n*}(c) \wedge \varphi_{n*}(c') = \varphi_{n*}(c \wedge c')$$

by lemma 8. Since

$$d \notin \mathrm{Cl}_2(\varphi_{n*}(Q_{n,k-1})), \qquad r(c \wedge c') > k-1.$$

Hence $c = c'$. Notation: $\mu(d) := c$.

It is plain that, for d, $d' \in R_{n,k}$ with dom d = dom d', $\mu(d) = \mu(d')$ implies $d = d'$.

(i) Obvious.

(ii) We shall show that if $d \sim d' \bmod f_{2n}$, for $d, d' \in R_{n, n/2l}$ then $\mu(d) \sim \mu(d')$ mod $W^{(n)}(\langle X \rangle)$.

Let $d = d_1 d_2 \ldots d_n$, and $d' = d'_1 d'_2 \ldots d'_n$ where, for $i = 1, \ldots, n$, d_i and d'_i belong to Cond $(\{0, 1\}^2)$. Then there are $w, w' \in \{0, 1\}^{2n}$ such that:

- $w \geqq \varphi_{n^*}(\mu(d)) \geqq d$, $w' \geqq d'$;
- $w \in K^{(2n)} (\Leftrightarrow \varphi_n^{-1}(w) \in X^n)$;
- $w - d = w' - d'$;
- $f_{2n}(w) = 1 (\Leftrightarrow \varphi_n^{-1}(w) = 1 \text{ in } \langle X \rangle)$;
- $f_{2n}(w') = 1 (\Leftrightarrow \varphi_n^{-1}(w') = 1 \text{ in } \langle X \rangle)$.

Let $w = w_1 w_2 \ldots w_n$, and $w' = w'_1 w'_2 \ldots w'_n$ where $w_i, w'_i \in \{0, 1\}^2$. Obviously, then $w_i \geqq d_i$, and $w'_i \geqq d'_i$, for $i = 1, \ldots, n$. Set $I := \{i \mid r(d_i) = r(d'_i) = 1, d_i \neq d'_i\}$. Assume that $I \neq \varnothing$. Then, for any $i \in I$, $\varphi^{-1}(w_i) \in X$ (since $w > \varphi_{n^*}(\mu(d))$), and $\varphi^{-1}(w'_i) \in X^{-1}$. But this implies $\varphi_n^{-1}(w) \neq \varphi_n^{-1}(w') \bmod N_=$, where $N_=$ is the normal closure of the element $x_1 x_2^{-1}$ in $\langle X \rangle$. Contradiction to $w = w' = 1$ in $\langle X \rangle$.

Hence we have proved: If $r(d_i) = r(d'_i) = 1$, then $d_i = d'_i$. The fact that $\mu(d) \sim \mu(d')$ mod $W^{(n)}(\langle X \rangle)$ follows from the definition of $W^{(n)}(\langle X \rangle)$-equivalence and f_{2n}-equivalence. Consequently,

$$m_1(f_{2n}, R_{n, n/2l}) \leqq m_1(W^{(n)}(\langle X \rangle), Q_{n, n/2l}).$$

But $m_1(W^{(n)}(\langle X \rangle), Q_{n, n/2l}) \leqq 2^{n/3l}$ by lemma 6.

(iii) Let $d \in R_{n, k}$.

$$|\mathrm{Cl}_2^*(\{d\}) \cap K^{(2n)}| = |\mathrm{Cl}_1(\{\mu(d)\}) \cap X^n| = 2^{n-k}, \quad \square$$

THEOREM 3: *Assume that T is a real-time branching program computing f_n, for 24 dividing n.*

Then $\mathrm{SIZE}(T) \geqq 2^{n/48}$.

Proof: Immediate consequence of theorem 1 and lemma 9. $\square$

REFERENCES

1. M. Ajtai, L. Babai, P. Hajnal, J. Komlos, P. Pudlak, V. Rödel, E. Szemeredi and G. Turan, *Two lower bounds for branching programs*, 18th ACM STOC, 1986, pp. 30-39.
2. A. Borodin, D. Dolev, F. E. Fich and W. Paul, *Bounds for width two branching programs*, 15th ACM STOC, 1983, pp. 87-93.

3. L. BUDACH, *Lower bounds for the number of nodes in a decision tree*, EIK 21, 1985, No 4/5, pp. 221-228.
4. A. K. CHANDRA, M. L. FURST and R. J. LIPTON, *Multiparty protocols*, 15th ACM STOC, 1983, pp. 94-99.
5. P. E. DUNNE, *Lower bounds on the complexity of* 1-*time-only branching programs*, FCT Proc., *Lect. Notes in Comp. Sci.*, Vol. 199, 1985, pp. 90-99.
6. R. J. LIPTON and Y. ZALCSTEIN, *Word problems solvable in logspace*, Journal of the ACM, Vol. 24, No. 3, 1977, pp. 522-526.
7. E. I. NECHIPORUK, *On a Boolean function*, Dokl. Akad. Nauk USSR, Vol. 169, No. 4, 1966, pp. 765-766.
8. P. PUDLAK, *A lower bound on the complexity of branching programs*, Preprint, Univ. Prague, 1983.
9. U. VISHKIN and A. WIGDERSON, *Trade-offs between depth and width in parallel computation*, SIAM J. Comput., Vol. 14, No. 2, 1985, pp. 303-314.
10. I. WEGENER, *On the complexity of branching programs and decision trees for clique functions*, Univ. of Frankfurt, Fachbereich Informatik, Interner Bericht, 5/84, 1984.
11. A. C. YAO, *Lower bounds by probabilistic arguments*, 24th IEEE FOCS, 1983, pp. 420-428.
12. S. ZAK, *An exponential lower bound for one-time-only branching programs*, MFCS Proc., Lect. Notes in Comp. Sci., Vol. 176, 1984, pp. 562-566.
13. S. ZAK, *An exponential lower bound for real-time branching programs*, manuscript.