comput. complex. **14** (2005), 362 – 375
1016-3328/05/040362–14
DOI 10.1007/s00037-005-0203-0

computational complexity

ON THE COMPUTATIONAL POWER OF BOOLEAN DECISION LISTS

Matthias Krause

Abstract. We study the computational power of decision lists over AND-functions versus threshold-$\oplus$ circuits. AND-decision lists are a natural generalization of formulas in disjunctive or conjunctive normal form. We show that, in contrast to CNF- and DNF-formulas, there are functions with small AND-decision lists which need exponential size unbounded weight threshold-$\oplus$ circuits. Consequently, it is questionable if the polynomial learning algorithm for DNFs of Jackson (1994), which is based on the efficient simulation of DNFs by polynomial weight threshold-$\oplus$ circuits (Krause & Pudlák 1994), can be successfully applied to functions with small AND-decision lists. A further result is that for all $k \geq 1$ the complexity class defined by polynomial length AC^0_k-decision lists lies *strictly* between AC^0_{k+1} and AC^0_{k+2}.

Keywords. Decision lists, AC^0, learnability, lower bounds.
Subject classification. 68Q15, 94C05, 94C10, 06E30.

1. Introduction

A *decision list* L of input size n and length m for computing a function $f \in B_n$ is a sequence of m instructions of the form **if** $f^i(x) = a^i$ **then output** $f(x) = b^i$ **and stop**, followed by the single instruction **output** $f(x) = \neg b^m$ **and stop**. For $i = 1, \ldots, m$ the functions $f^i \in B_n$ are called *query functions*, and a^i and b^i are Boolean constants. B_n denotes the set of all Boolean functions in n variables. If the query functions are defined to belong to a function basis $S \subseteq B_n$ then L is called an S-*decision list*.

Decision lists are a special kind of decision trees and form a basic and natural model of computation. The standard models of decision lists and decision trees, where the query functions ask for variables, have been studied in numerous papers. With the results given here we intend to initiate a more systematic study of decision lists and decision trees which are defined over different sets of query functions like AND-functions, AC^0_k-functions, MOD-functions and threshold functions. We think that this could be a promising way to extend the known reservoir of lower bound arguments, learning rules for Boolean concept classes and efficient data structures and algorithms for implementing and manipulating Boolean functions.

Our results concern the computational power of AND-*decision lists* (i.e., decision lists over AND-functions), width(k)-*decision lists* (i.e., decision lists for which all query functions depend on at most k variables, k a constant), and AC^0_k-*decision lists* (i.e., decision lists which query AC^0_k-functions).

The first group of results is devoted to comparing the computational power of width(k)-decision lists and AND-decision lists with the power of threshold-$\oplus$ circuits. AND-decision lists are a natural generalization of formulas in disjunctive or conjunctive normal form (for short, DNF- and CNF-formulas) which can be considered as monotone AND-decision lists. The comparison function $\mathrm{COMP}_n(x, y)$, which outputs 1 iff the n-bit number x is not smaller than the n-bit number y, is a natural witness for the fact that AND-decision lists are strictly more powerful than DNF- and CNF-formulas. COMP_n does not belong to AC^0_2 [1], but it has linear length AND-decision lists, it even has linear length width(2)-decision lists (see Section 2).

Threshold-$\oplus$ circuits (i.e., unbounded fan-in depth-2 circuits with $\oplus$-gates at the bottom level and a threshold output gate at the top) were extensively studied during the last decade (see, e.g., Alon & Bruck 1994, Aspnes *et al.* 1991, Bruck 1990, Bruck & Smolensky 1990, Goldmann *et al.* 1992, Krause & Pudlák 1994, 1995). One reason for the importance of this computational model is that representing a Boolean function f by threshold-$\oplus$ circuits is equivalent to representing f as the sign of a polynomial with integer coefficients. This allows us to characterize the complexity of f with respect to threshold-$\oplus$ circuits in terms of the spectral coefficients of f (Bruck 1990, Bruck & Smolensky 1990, Linial *et al.* 1989). Another nice property is that polynomial weight threshold-$\oplus$ circuits are polynomially PAC-learnable with respect to the uniform distribution (Jackson 1994), if membership queries are allowed. As polynomial size CNF- and DNF-formulas can be efficiently simulated by polynomial weight threshold-$\oplus$ circuits (Krause & Pudlák 1994), Jackson's learning algorithm yields the only known polynomial learning algorithm for CNF- and DNF-formulas. Let, as usual, $\hat{PT}_1$ and PT_1 denote the complexity classes containing those sequences of Boolean functions which have polynomial weight threshold-$\oplus$ circuits and polynomial size threshold-$\oplus$ circuits, respectively.

[1]Note that any monomial accepting (x, x) and (y, y) accepts also (x, y) and (y, x). Consequently, each DNF-formula F for COMP_n has at least 2^n monomials, as each monomial in F is allowed to accept at most one input (x, x). A similar argument works for CNF-formulas for COMP_n.

The main question which stimulated this research was to find out whether even polynomial length AND-decision lists can be simulated by polynomial weight threshold-$\oplus$ circuits. This would imply that Jackson's learning algorithm polynomially learns even functions with polynomial length AND-decision lists. One main result of this paper is a negative answer to this question. There are functions with polynomial length AND-decision lists for which even weighted threshold-$\oplus$ circuits have exponentially many nodes (Theorem 3.1).

Theorem 3.1 is based on analyzing the computational power of width(k)-decision lists versus threshold-$\oplus$ circuits of bounded weight. The main result here is that there are even functions with width(2)-decision lists which cannot be computed by threshold-$\oplus$ circuits of subexponential weight (Theorem 3.2). Theorem 3.2 is proved by using a result of Beigel (1994), providing a Boolean function f with a width(1)-decision list which cannot be computed by threshold-$\oplus$ circuits within subexponential weight and sublinear bottom fan-in, combined with a probabilistic argument.

Note that for any constant k, width(k)-decision lists can always be simulated by polynomial size (but exponential weight) threshold-$\oplus$ circuits. This is because each decision list of length m can be simulated by an m-ary exponential weight threshold function over the query functions (see relation (F) in Section 2). COMP$_n$, a function with 2-decision lists, *has* polynomial weight threshold-$\oplus$ circuits (Bruck 1990).

Another part of this paper concerns the computational power of AC_k^0-decision lists. For all integers $k \geq 1$ let DL^k denote the class of all Boolean functions having polynomial length AC_k^0-decision lists. Due to property (G) in Section 2 it follows that

$$AC_k^0 \subseteq DL^{k-1} \subseteq AC_{k+1}^0.$$

Our second main result is that all these inclusions are strict (Theorem 4.1). The proof is done via induction on k. The induction step is based on the switching lemma of Håstad (1986). Note that DL^1 coincides with the set of all Boolean functions having polynomial length AND-decision lists. Consequently, our result $DL^1 \not\subseteq PT_1$ strengthens the main results of Krause & Pudlák (1994) that $AC_2^0 \subseteq \hat{PT_1}$ but $AC_3^0 \not\subseteq PT_1$.

The paper is organized as follows. In Section 2 we introduce some more notations and discuss some basic properties of decision lists and threshold-$\oplus$ circuits. Section 3 contains our results about the computational power of AND-decision lists versus threshold-$\oplus$ circuits. Section 4 is devoted to the proof of Theorem 4.1.

2. Some more basics

If not stated otherwise, Boolean functions are defined to be functions which map $\{0,1\}^n$ to $\{0,1\}$ for some natural number n. As a compressed notation of decision lists we use expressions like $L = (f^1, a^1, b^1), \ldots, (f^m, a^m, b^m)$, where for $i = 1, \ldots, m$, f^i, a^i, b^i are defined as at the beginning of Section 1. Note for instance that the 2-decision list

$$(x_{n-1} > y_{n-1}?, 1, 1), (x_{n-1} < y_{n-1}?, 1, 0), \ldots, (x_0 > y_0?, 1, 1)(x_0 < y_0?, 1, 0)$$

computes the decision whether the n-bit number $x = (x_{n-1}, \ldots, x_0)$ is not smaller than the n-bit number $y = (y_{n-1}, \ldots, y_0)$, i.e., COMP_n. Observe that:

(A) If L computes the function f then $(\neg f^1, \neg a^1, b^1), \ldots, (\neg f^m, \neg a^m, b^m)$ also does.

(B) If L computes f then $(f^1, a^1, \neg b^1), \ldots, (f^m, a^m, \neg b^m)$ computes $\neg f$.

(C) $\bigvee_{k=1}^m f^k$ can be computed by $(f^1, 1, 1), \ldots, (f^m, 1, 1)$.

(D) $\bigwedge_{k=1}^m f^k$ can be computed by $(f^1, 0, 0), \ldots, (f^m, 0, 0)$.

(E) Each width(k)-decision list of length m can be simulated by an AND-decision list of length at most $2^{k-1}m$.

(F) If $L = (f^1, a^1, b^1), \ldots, (f^m, a^m, b^m)$ computes f then

$$f(x) = 1 \quad \text{iff} \quad 2\sum_{i=1}^{m} 2^{m-i}(-1)^{b^i}(f^i(x) = a^i) + (-1)^{1-b^m} \leq 0,$$

i.e., f can be written as an exponential weight threshold function over $f^1, \ldots, f^m$.

(G) If $L = (f^1, a^1, b^1), \ldots, (f^m, a^m, b^m)$ computes f then

$$\begin{aligned} f(x) &= \bigvee_{i,\, b^i=1} \Big(\bigwedge_{j=1}^{i-1}(f^j(x) \neq a^j) \wedge (f^i(x) = a^i)\Big) \vee (\neg b_m) \bigwedge_{j=1}^{m}(f^j(x) \neq a^j) \\ &= \bigwedge_{i,\, b^i=0} \Big(\bigvee_{j=1}^{i-1}(f^j(x) = a^j) \vee (f^i(x) \neq a^i)\Big) \wedge \Big(\bigvee_{j=1}^{m}(f^j = a^j) \vee \neg b^m\Big). \end{aligned}$$

Properties (A)–(D) follow directly from the definitions and the De Morgan laws. Property (E) can be obtained by replacing all query functions f^k by a monotone sublist simulating a DNF-formula for f^k according to (C). The coefficients of the threshold representation in (F) are constructed in such a way that the computational mode of a decision list is simulated. In particular the sign is $(-1)^{b^i}$ for the smallest i for which $f^i(x) = a^i$, or $(-1)^{1-b^m}$ if $f^i(x) \neq a^i$ for all i. The formulas in (G) reflect the fact that L accepts x iff x reaches an accepting sink iff there is an i, $1 \leq i \leq m$, with $b^i = 1$ such that $f^i(x) = a^i$ and $f^j(x) \neq a^j$ for all $1 \leq j < i$, or if $f^j(x) \neq a^j$ for all $1 \leq j \leq m$ and $b^m = 0$. Conversely, L accepts x iff x does not reach any rejecting sink iff for all i with $b^i = 0$ we have $f^i(x) \neq a^i$ or there is a $j < i$ such that $f^j(x) = a^j$, and $b_m = 0$ or there is a $j \leq m$ with $f^j(x) = a^j$.

For all constants k let us denote by WIDTH$^k DL$ the set of all functions having width(k)-decision lists. As for each decision list it makes no sense to pose a query twice, width(k)-decision lists can always be supposed to have polynomial length.

Typical representatives of these classes are functions $d_{k,n} : \{0,1\}^{kn} \to \{0,1\}$ defined by $d_{k,n}(x_{1,1}, \dots, x_{1,k}, \dots, x_{n,1}, \dots, x_{n,k}) = 1$ iff the maximal i for which $\bigwedge_{j=1}^{k} x_{i,j} = 1$ is odd. An AND-decision list for $d_{k,n}$, n odd, is

$$\Big(\Big(\bigwedge_{i=1}^{k} x_{i,n}, 1, 1\Big), \Big(\bigwedge_{i=1}^{k} x_{i,n-1}, 1, 0\Big), \dots, \Big(\bigwedge_{i=1}^{k} x_{i,2}, 1, 0\Big), \Big(\bigwedge_{i=1}^{k} x_{i,1}, 1, 1\Big)\Big).$$

Note that $(d_{k,n})_{n\in\mathbb{N}}$ is WIDTH$^k DL$-complete and that $(d_{n,n})_{n\in\mathbb{N}}$ is DL^1-complete with respect to projections. (Projection reducibility is defined as follows. Let K denote a complexity class consisting of sequences of Boolean functions. A sequence $F = (f_n)_{n\in\mathbb{N}}$ of Boolean functions is called *K-complete with respect to projections* if all $G = (g_n)_{n\in\mathbb{N}}$ in K are *projection reducible* to F, i.e., there is a polynomially bounded function $m : \mathbb{N} \to \mathbb{N}$ such that for all $n \in \mathbb{N}$, g_n is a projection of $f_{m(n)}$. A Boolean function $g = g(x_1, \dots, x_n)$ is called a *projection* of a Boolean function $f(y_1, \dots, y_m)$, if there is a mapping $\pi : \{y_1, \dots, y_m\} \to \{x_1, \neg x_1, \dots, x_n, \neg x_n\} \cup \{0,1\}$ such that

$$g(x_1, \dots, x_n) = f(\pi(y_1), \dots, \pi(y_m)).$$

It can be easily derived that each Boolean function with a decision list of length l over k-ary AND-functions can be written as a projection of $d_{k,n}$ for some $n \leq 2l$.)

A Boolean function $f \in B_n$ is called a *threshold function* if it can be computed by a single threshold gate, i.e., if there are integers $a_0, \dots, a_n$ such that $f(x) = 1$ iff $a_1x_1 + \dots + a_nx_n \geq a_0$.

The numbers $a_0, \ldots, a_n$ are called *edge weights* and the value $|a_0|+\cdots+|a_n|$ is called the *weight* of the gate. The systematic study of threshold functions started with the work on *perceptrons* by Minsky & Papert (1968). It was shown there that all n-ary threshold functions can be written as threshold functions of weight $\exp(n \log n)$. There are threshold functions which cannot be written as polynomial weight threshold functions: see, for instance, COMP_n. It is known that unbounded weight polynomial size threshold-$\oplus$ circuits can do more than polynomial weight threshold-$\oplus$ circuits, i.e. $\hat{PT}_1 \subset PT_1$ (Goldmann *et al.* 1992).

Computing a Boolean function $f \in B_n$ by a threshold-$\oplus$ circuit containing s $\oplus$-gates of fan-in at most d and one threshold top gate of weight w is the same as representing f as the sign of a *voting polynomial*, i.e., a degree-d integer polynomial in n variables with s monomials, where the sum of the absolute values of the coefficients is w. As voting polynomials allow a more elegant presentation of our proof techniques we will use them in the following and give the corresponding notations.

For $b \in \{0,1\}$ we write $b^* = (-1)^b$. For $x = (x_1, \ldots, x_n) \in \{0,1\}^n$ we denote by $x^* \in \{1,-1\}^n$ the vector $x^* = (x_1^*, \ldots, x_n^*)$, and for a function $f : X \to \{0,1\}$ we define $f^* : X \to \{1,-1\}$ by $f^*(x) = (f(x))^*$. For a nonzero integer z we denote by $\operatorname{sgn}(z) \in \{1,-1\}$ the sign of z. Observe that $f \in B_n$ is a threshold function iff there are integers $b_0, \ldots, b_n$ such that $f^*(x) = \operatorname{sgn}(b_0 + b_1x_1 + \cdots + b_nx_n)$ iff there are integers $c_0, \ldots, c_n$ such that $f^*(x) = \operatorname{sgn}(c_0+c_1x_1^*+\cdots+c_nx_n^*)$. For each subset $I \subseteq [n]$ we denote by m_I the monomial $m_I = \prod_{i \in I} x_i$, where $m_\emptyset$ corresponds to the constant-1 function.

We call p a *voting polynomial* for f if $f^*(x) = \operatorname{sgn}(p(x^*))$ for all $x \in \{0,1\}^n$.

Note that $m_I(x^*) = (\bigoplus_{i \in I} x_i)^*$ for each $I \subseteq [n]$. Thus, voting polynomials correspond to threshold-$\oplus$ circuits in a straightforward way. The relevant cost measures of integer polynomials $p = \sum_{I \subseteq [n]} a_I m_I$ are the degree (the maximal length of a monomial occurring in p), the weight (the sum of the absolute values of the coefficients), and the length (the number of monomials occurring in p).

We get the complexity measures $\deg(f)$, $\mathrm{weight}(f)$, $\mathrm{length}(f)$ for Boolean functions f, defined as the minimal degree, minimal weight and minimal length of a voting polynomial for f, respectively. Moreover we need to consider the measure $\mathrm{weight}^d(f)$ defined as the minimal weight of a degree-d voting polynomial for f.

Note that due to (F), $d_{1,n}$ is a threshold function for all n. The following technical result on writing $d_{1,n}$ as a threshold function can be verified quite straightforwardly.

LEMMA 2.1. $d^*_{1,n} = \operatorname{sgn}(1 + b\sum_{i=1}^{n}(-a)^i x_i)$ *for all natural* $a \geq 2$ *and* $b \geq 1$.□

It follows that for all $f \in \mathrm{WIDTH}^k DL$ we have $\deg(f) \leq k$, and thus $\mathrm{WIDTH}^k DL \subseteq PT_1$.

3. AND-decision lists versus threshold-⊕ circuits

This section is devoted to the proof of

THEOREM 3.1. $\mathrm{length}(d_{n,n}) \in 2^{n^{\Omega(1)}}$, *i.e.*, $DL^1 \not\subseteq PT_1$.

The proof is performed in two steps. The first is to show the following

THEOREM 3.2. $\mathrm{weight}(d_{2,n}) \in 2^{n^{\Omega(1)}}$, *i.e.*, $\mathrm{WIDTH}^2 DL \not\subseteq \hat{PT}_1$.

The second step is to deduce Theorem 3.1 from Theorem 3.2. As a byproduct we show that AC^0_3 and $\mathrm{WIDTH}^3 DL$ contain *heavy* threshold functions, i.e., threshold functions which do not belong to $\hat{PT}_1$. The first example of a heavy threshold function (which is not an AC^0-function) was given by Goldmann *et al.* (1992).

PROOF OF THEOREM 3.2. The proof uses first a nontrivial tradeoff result between weight and degree for voting polynomials for $d_{1,n}$. Using a probabilistic argument Bruck and Smolensky (1990) showed that $d_{1,n} \in \hat{PT}_1$, i.e., $d_{1,n}$ has polynomial weight voting polynomials. On the other hand, $d_{1,n}$ is an exponential weight threshold function, i.e. $\deg(d_{1,n}) = 1$. Beigel (1994) observed that threshold-∧ circuits for $d_{1,n}$ cannot have polynomial weight and small bottom fan-in simultaneously. In particular, there is a constant C such that for all n and each depth-2 circuit for $d_{1,n}$ with s ∧-gates of fan-in at most d at the bottom level and a threshold gate with edge weights of absolute value at most w at the top we have the relation

$$w \;\geq\; \frac{1}{s}\, 2^{Cn/d^2}.$$

Now, let d, $2 \leq d \leq n$, be arbitrary, consider a degree-d voting polynomial p of weight $\mathrm{weight}^d(d_{1,n})$ for $d_{1,n}$, and let $W = \max|w|$ over all weights w occurring in p. Observe that p has at most n^d monomials. By means of the formula

$$\bigoplus_{j=1}^{d} x_j \;=\; \sum_{I\subseteq[d],\, I\neq\emptyset} 2^{|I|} \bigwedge_{i\in I} x_i,$$

p defines a threshold-$\wedge$ circuit for $d_{1,n}$ of at most $2^d n^d$ $\wedge$-gates of degree at most d at the bottom level, and the absolute value of any edge weight does not exceed $W2^d$. We obtain the relation

$$2^d W \;\geq\; 2^{-(\log(n)+d)} 2^{Cn/d^2},$$

i.e.,

$$\text{weight}^d_{\oplus}(d_{1,n}) \;\geq\; 2^{Cn/d^2-\log(n)-2d}. \tag{3.3}$$

Next, we use the following construction from Krause & Pudlák (1994). For all $f \in B_n$ let $f^{\mathrm{op}} \in B_{3n}$ be defined by

$$f^{\mathrm{op}}(x_1,\ldots,x_n,y_1,\ldots,y_n,z_1,\ldots,z_n) = f(u_1,\ldots,u_n),$$

where $u_i = \bar{z}_i x_i \vee z_i y_i$ for all $i,\ 1 \leq i \leq n$. Theorem 3.2 can be derived from the following technical lemma.

Lemma 3.4. *For all $f \in B_n$ we have* $\text{weight}(f^{\mathrm{op}}) \geq \text{weight}^d(f)$ *for all d with* $2^d \geq \text{weight}(f^{\mathrm{op}})$.

We complete the proof of Theorem 3.2 using Lemma 3.4. First observe that $d^{\mathrm{op}}_{1,n} \in \mathrm{WIDTH}^2 DL$. Now suppose that there exist numbers $n_0 \geq 1$ and $\epsilon > 0$ such that $\text{weight}(d^{\mathrm{op}}_{1,n}) < 2^{n^{1/3-\epsilon}}$ for all $n \geq n_0$. Then, by putting $d = n^{1/3-\epsilon}$ into Lemma 3.4 we see by (3.3) that

$$2^{n^{1/3-\epsilon}} > \text{weight}^d(d_{1,n}) \geq 2^{Cn^{1/3+2\epsilon}-2n^{1/3-\epsilon}-\log(n)}$$

for all $n \geq n_0$, which is obviously wrong. □

Proof of Lemma 3.4. We use a probabilistic argument which is similar to the one in Lemma 2.3 of Krause & Pudlák (1994). Fix $f \in B_n$ and a voting polynomial $p = \sum_{I\subseteq X\cup Y\cup Z} a_I m_I = \sum_{I\in M} a_I m_I$ of weight W for f^{op}, and let $d' = d+1$, where d denotes the minimal natural number for which $2^d > W$. Let M denote the set of all I for which $a_I \neq 0$. Observe that $|M| \leq W$. Let f^{op} depend on the sets of variables $X = \{x_1,\ldots,x_n\}$, $Y = \{y_1,\ldots,y_n\}$ and $Z = \{z_1,\ldots,z_n\}$.

Now observe that each assignment c of the Z-variables defines a partition $X \cup Y = U(c) \cup V(c),\ |U(c)| = |V(c)| = n$, such that $(f^{\mathrm{op}})^c$ depends only on the variables of $U(c)$. This means that for any fixed assignment $u \in \{0,1\}^{U(c)}$ we have $f^*(u) = (f^{\mathrm{op}*})^c(u,v)$ for all assignments $v \in \{0,1\}^{V(c)}$. This implies

$f^*(u) = \mathbf{E}_v[(f^{\mathrm{op}*})^c(u,v)]$, where $\mathbf{E}_v$ denotes the expected value for v randomly chosen with respect to the uniform distribution on $\{0,1\}^{V(c)}$. Consequently,

$$f^*(u) = \mathbf{E}_v\Big[\mathrm{sgn}\Big(\sum_{I\in M} a_I m_I^c(u^*,v^*)\Big)\Big] = \mathrm{sgn}\Big(\sum_{I\in M} a_I \mathbf{E}_v[m_I^c(u^*,v^*)]\Big).$$

Note that $\mathbf{E}_v[m_I^c(u^*,v^*)] = 0$ if $I \cap V(c) \neq \emptyset$. Correspondingly, we say that c *destroys a monomial* m_I of p if $I \cap V(c) \neq \emptyset$. Each $c \in \{0,1\}^Z$ defines a $\{1,-1\}$-voting polynomial $p|_c$ of weight $\leq W$ for f by removing all those monomials which are destroyed by c.

Call a monomial m_I *large* if I contains at least d' variables from $X \cup Y$. Clearly, p contains at most W large monomials. The probability (with respect to $c \in_U \{0,1\}^Z$) that a large monomial is not destroyed by c is at most $2^{-d'}$. Consequently, the probability that there is an assignment $c \in \{0,1\}^Z$ that destroys all large gates is at least $1 - W2^{-d'}$, which is positive because of the choice of d'. The resulting $\{1,-1\}$-voting polynomial for f contains only monomials of length at most d, and we conclude that $\mathrm{weight}^d(f) \leq W$. □

Theorem 3.2 shows that for $k = 2$ there are sequences $f = (f_n)_{n\in\mathbb{N}}$ of Boolean functions with width(k)-decision lists for which width(f_n) grows exponentially in n. It is an interesting question if there are constants k such that there are *threshold* functions with this property.

THEOREM 3.5. WIDTH^3*DL contains heavy threshold functions, i.e., a sequence* $t = (t_n)_{n\in\mathbb{N}}$ *of threshold functions with* $\mathrm{weight}(t_n) \in 2^{n^{\Omega(1)}}$.

PROOF. Due to the WIDTH^2*DL*-completeness of the function $d_{2,n}$ we know by Theorem 3.2 that

$$\mathrm{weight}(d_{2,n}) \in 2^{n^{\Omega(1)}}.$$

Observe that for all $a \geq 2$, by Lemma 2.1,

$$\begin{aligned} d^*_{2,n} &= \mathrm{sgn}\Big(1 + 2\sum_{i=1}^{n}(-a)^i x_{i,1}x_{i,2}\Big) \\ &= \mathrm{sgn}\left(1 + 2\sum_{i=1}^{n}(-a)^i \frac{1}{2}(x_{i,1} + x_{i,2} - (x_{i,1} \oplus x_{i,2}))\right). \end{aligned}$$

Thus $T_n^*(x,y,z) = \mathrm{sgn}(1 + \sum_{i=1}^n (-a)^i(x_i + y_i - z_i))$ defines a threshold function $T_n \in B_{3n}$ with $\mathrm{weight}(T_n) \in 2^{n^{\Omega(1)}}$.

It remains to show that $T_n \in \mathrm{WIDTH}^3 DL$. If $x_i + y_i = z_i$ for all $i = 1, \ldots, n$ then $T_n(x, y, z) = 0$. We have $T_n(x, y, z) = 1$ if, for $r = \max\{i : x_i + y_i \neq z_i\}$, $x_r + y_r > z_r$ and r is odd, or $x_r + y_r < z_r$ and r even.

A width(3)-decision list for T_n consists of n consecutive blocks $D_n, D_{n-1}, \ldots, D_1$. Inside D_i, all inputs satisfying $x_i + y_i < z_i$, corresponding to $\bar{x}_i \bar{y}_i z_i = 1$, are mapped to b_i and all inputs satisfying $x_i + y_i > z_i$, corresponding to $x_i \bar{z}_i \vee y_i \bar{z}_i \vee x_i y_i = 1$, are mapped to $\bar{b}_i$, where $b_i = 1$ if i is even and $b_i = 0$ if i is odd. The remaining inputs are forwarded to D_{i+1} if $i < n$, and mapped to 0 if $i = n$. □

Proof of Theorem 3.1 using Theorem 3.5. We use the main result from Krause & Pudlák (1995), Theorem 1.4, saying that for all threshold functions $t_n \in B_n$, if $\mathrm{weight}(t_n) \in 2^{n^{\Omega(1)}}$ then $\mathrm{length}(t_n \circ \mathrm{AND}_{n,n}) \in 2^{n^{\Omega(1)}}$, where $t_n \circ \mathrm{AND}_{n,m} \in B_{nm}$ is defined as

$$t_n \circ \mathrm{AND}_{n,m}(x_{1,1}, \ldots, x_{1,m}, \ldots, x_{n,1}, \ldots, x_{n,m}) = t_n\Big(\bigwedge_{i=1}^{m} x_{1,i}, \ldots, \bigwedge_{i=1}^{m} x_{n,i}\Big).$$

As a consequence, $\mathrm{length}(T_n \circ \mathrm{AND}_{n,n}) \in 2^{n^{\Omega(1)}}$, where T_n denotes the heavy threshold function considered in Theorem 3.5. As obviously $T_n \circ \mathrm{AND}_{n,n} \in DL^1$ the theorem follows. □

4. The decision list hierarchy and the AC^0-hierarchy

In this section we prove

Theorem 4.1. *$AC^0_k \subset DL^{k-1} \subset AC^0_{k+1}$ for all $k \geq 1$.*

The proof is by induction on k. The induction step makes use of an argument from Theorem 5 (p. 17) of Håstad (1986), in which the levels of the AC^0-hierarchy are separated. The more advanced technique of Håstad (1989) can be used in an equivalent way to prove our theorem. Note that also elsewhere in the literature the recursive argument of Håstad (1986, 1989) was used to prove hierarchy results (see, e.g., Berg & Ulfberg 1998).

On page 15 (Figure 7) of Håstad (1986) a Boolean function g^k_m is defined. We define a function h^k_m in a slightly modified way by giving the definition of a circuit H^k_m computing h^k_m. For $k \geq 3$ let H^k_m consist of k layers consisting of gates of fan-in $4.4m^2, 4.4m^3, \ldots, 4.4m^{k-1}, 1.1m^k, m^k$. H^2_m consists of two layers consisting of gates of fan-in $1.1m^2$ and m^2. The top gate is a gate computing $d_{1,4.4m^2}$ (resp. $d_{1,1.1m^2}$ for $k = 2$), followed by a level of AND-gates, followed by a level of OR-gates, followed by a level of AND-gates and so on. The bottom gates are ANDs, resp. ORs, over pairwise disjoint blocks $B_j^{k,m}$, $j = 1, 2, \ldots$, containing m^k variables each. Observe that h_m^k belongs to DL^{k-1}. We show that $h_m^k \notin AC_0^k$ by the following modification of Theorem 5 in Håstad (1986).

Lemma 4.2. *There are no depth $k+1$ AND, OR circuits computing h_m^k with bottom fan-in $\leq m/10$ and size $\leq 2^{m/10}$ for $m > m_0$, some absolute constant.*

This lemma is proved by induction on k. It is quite straightforward to check that the induction step is similar to the proof of Theorem 5 in Håstad (1986). In particular, consider the probability distribution $R^k(m)$ on the set of partial assignments to the variables of h_m^k. For each block $B_j^{k,m}$ of the input variables of h_m^k do independently the following experiment.

With probability $1/m^{k-1}$ set all $B_j^{k,m}$ variables to 1. With probability $1 - 1/m^{k-1}$ choose a random $x \in B_j^{k,m}$, set all $x' \in B_j^{k,m} \setminus \{x\}$ to 1, and with probability $1/m$ set x to $*$ and with probability $1 - 1/m - 1/m^{k-1}$ to 0.

The induction step follows from the two observations:

(1) Applying a random restriction, distributed according to $R^k(m)$, $k \geq 3$, to an AND, OR circuit of depth $k+1$, size $2^{m/10}$, and bottom fan-in $m/10$ leads to a circuit of depth k and the same bottom fan-in with probability $1 - 2^{m/10}\alpha^{m/10}$, where $\alpha \leq 0.42$ for m large enough (Lemma 5 in Håstad 1986).

(2) Applying a random restriction, distributed according to $R^k(m)$, to h_m^k, $k \geq 3$, yields a function which contains h_m^{k-1} as a subfunction with probability at least $2/3$. This can be proved in the same manner as Lemma 8 in Håstad (1986).

Consequently, for m large enough and $k \geq 2$ the existence of an AND, OR circuit of depth $k+1$, size $2^{m/10}$ and bottom fan-in $m/10$ computing h_m^k implies the existence of an AND, OR circuit of depth k of at most the same size and bottom fan-in for h_m^{k-1}. We have to show the base case $k = 2$. Observe that we can write

$$h_m^2(B_1, \ldots, B_{1.1m^2}) = d_{1,1.1m^2}\Big(\bigwedge_{x_i \in B_1} x_i, \ldots, \bigwedge_{x_i \in B_{1.1m^2}} x_i\Big).$$

Assume that there is a depth-3 AND,OR circuit C of size $\leq 2^{m/10}$ and bottom fan-in $\leq m/10$ computing h_m^2. Observe that, under $R^2(m)$, with probability $(1 - m^{-2})^{1.1m^2} \approx e^{-1.1} > 0.3$ no bottom AND of H_m^2 is set to constant 1 and that the expected number of bottom ANDs which are set to $*$ is $1.1m$. Consequently, if m is large enough then with probability at least 0.3 at least m bottom ANDs are set to $*$ and no bottom AND is set to 1. Thus, if m is large enough our assumption implies that there is a depth two AND, OR circuit of bottom fan-in $m/10$ computing $d_{1,m}$. This would imply that either $d_{1,m}^{-1}(1)$ or $d_{1,m}^{-1}(0)$ can be covered by prime implicants of length at most $m/10$. But this can be disproved by observing that

$$d_{1,m} = \bigwedge_{r \leq m \text{ odd}} \bar{x}_1 \cdots \bar{x}_{r-1} x_r, \quad \text{resp.} \quad \neg d_{1,m} = \bigwedge_{r \leq m \text{ even}} \bar{x}_1 \cdots \bar{x}_{r-1} x_r$$

are the only possibilities to cover $d_{1,m}^{-1}(1)$ (resp. $d_{1,m}^{-1}(0)$) by prime implicants.

To show that $DL^{k-1} \subset AC_0^{k+1}$ we prove a more general result. Define a Th(k)-*circuit* to be a circuit consisting of an unbounded weight threshold gate on the top followed by $k-1$ levels of AND, OR gates. As $d_{1,n}$ is a threshold function, for all $k \geq 2$ all problems in DL^{k-1} have polynomial size Th(k)-circuits. Consider a function f_m^k which is defined via a circuit F_n as follows. F_n is a tree consisting of k levels which contain AND and OR gates alternately. For $k > 2$ the top gate is an AND gate of fan-in $4.4m^4$ followed by levels of gates of fan-in $4.4m^5, \ldots, 4.4m^{k+1}, 1.1m^{k+2}, m^{k+2}$. The gates at the bottom level are defined over pairwise disjoint blocks of m^{k+2} variables each. The function f_m^2 is an AND over $1.1m^4$ OR-gates of fan-in m^4. The proof of the theorem will be finished by applying the following lemma of Berg and Ulfberg (1998). For the sake of completeness we give a sketch of the proof.

Lemma 4.3 (Berg & Ulfberg 1998). *For all $k \geq 2$ and $m > m_0$ some absolute constant,* Th(k)*-circuits computing f_m^{k+1} with bottom fan-in $\leq m/10$ and size $\leq 2^{m/10}$ do not exist.*

This lemma is proved by induction on k again. The induction step can be performed exactly in the same way as described above, i.e.. as in Håstad (1986). We consider the base case $k = 2$ and suppose that there is a Th(2)-circuit $C = t(C_1, \ldots, C_r)$ of size $\leq 2^{m/10}$ and bottom fan-in $\leq m/10$ computing f_m^3. Here, t denotes a threshold function corresponding to the top gate of C. By Lemma 5 from Håstad (1986), a random restriction from $R^5(m)$ with probability $1 - 2^{m/10}\alpha^{m/10} \geq 1 - 0.84^{m/10}$ converts all functions $C_i,\ i = 1, \ldots, r$, into functions for which the function itself and its negation have prime implicants of size at most $m/10$. This implies that all these restricted functions have decision trees of depth $(m/10)^2$ (see, e.g., Wegener 2000, Theorem 2.5.11), and thus can be written as rational polynomials of degree at most $(m/10)^2$. Consequently, with probability at least $1 - 0.84^{m/10}$ we have $\deg(C|_\rho) \leq (1/100)m^2$, where ρ is $R^5(m)$-distributed. On the other hand, $f_m^3|\rho$ has f_m^2 as a subfunction with probability at least 2/3. This contradicts a result of Minsky and Papert (1968) saying that $\deg(\bigwedge_{i=1}^n \bigvee_{j=1}^{4n^2} x_{i,j}) = n$, i.e., $\deg(f_m^2) \geq \frac{1}{2}m^2$.

Acknowledgements

I would like to thank the unknown referees, and Daniel Mölle and Ingo Wegener for helpful discussions and for finding bugs in earlier versions of this paper. This research was partly supported by DFG grant Kr 1521/3-2 and Kr 1521/7-1.

References

N. Alon & J. Bruck (1994). Explicit constructions of depth-2 majority circuits for comparison and addition. *SIAM J. Discrete Math.* **7**, 1–8.

J. Aspnes, R. Beigel, M. Furst & S. Rudich (1991). The expressive power of voting polynomials. In *Proc. 23rd ACM Symposium on the Theory of Computing (STOC)* 1991, 402–409. See also *Combinatorica* **14** (1994), 1–14.

R. Beigel (1994). Perceptrons, PP, and the polynomial hierarchy. *Comput. Complexity* **4**, 339–349.

C. Berg, S. Ulfberg (1998). A lower bound for perceptrons and an oracle separation of the PP^PH hierarchy. *J. Comput. System Sci.* **56**, 263–271.

J. Bruck (1990). Harmonic analysis of polynomial threshold functions. *SIAM J. Discrete Math.* **3**, 168–177.

J. Bruck & R. Smolensky (1990). Polynomial threshold functions, AC^0-functions, and spectral norms. In *Proc. 31th IEEE Symposium on Foundations of Computer Science (FOCS)*, 632–641. See also *SIAM J. Comput.* **21** (1992), 33–42.

M. Goldmann, J. Håstad & A. A. Razborov (1992). Majority gates versus general weighted threshold gates. *Comput. Complexity* **2**, 277–300.

J. Håstad (1980). Almost optimal lower bounds for small depth circuits. In *Proc. 18th ACM Symposium on the Theory of Computing (STOC)*, 6–20.

J. Håstad (1989). Randomness and computation. In *Advances in Comp. Research*, Vol. 5, S. Micali (ed.), JAI Press, 143–170.

J. Jackson (1994). An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. In *Proc. 35th IEEE Symposium on Foundations of Computer Science (FOCS)*, 42–53. See also *J. Comput. System Sci.* **55** (1997), 414–440.

M. Krause & P. Pudlák (1994). On the computational power of depth-2 circuits with threshold and modulo gates. In *Proc. 26th ACM Symposium on the Theory of Computing (STOC)*, 49–59. See also *Theoret. Comput. Sci.* **174** (1997), 137–156.

M. Krause & P. Pudlák (1995). Computing Boolean functions by polynomials and threshold circuits. In *Proc. 36th IEEE Symposium on Foundations of Computer Science (FOCS)*, 682–691. See also *Comput. Complexity* **7** (1998), 346–370.

N. Linial, Y. Mansour & N. Nisan (1989). Constant depth circuits, Fourier transform, and learnability. In *Proc. 30th IEEE Symposium on Foundations of Computer Science (FOCS)*, 574–579. See also *J. ACM* **4** (1993), 607–620.

M. Minsky & S. Papert (1968). *Perceptrons*. MIT Press, Cambridge (expanded edition, 1988).

I. Wegener (2000). *Branching Programs and Binary Decision Diagrams*. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia.

Manuscript received 18 November 2003

Matthias Krause
Lehrstuhl Theoretische Informatik
Universität Mannheim
68131 Mannheim, Germany
krause@th.informatik.uni-mannheim.de