On the Incompressibility of Monotone DNFs

Matthias P. Krieger[*]

Johann Wolfgang Goethe-Universität Frankfurt
Institut für Informatik
Lehrstuhl für Theoretische Informatik
Robert-Mayer-Straße 11–15
D-60054 Frankfurt am Main, Germany
mkrieger@cs.uni-frankfurt.de

Abstract We prove optimal lower bounds for multilinear circuits and for monotone circuits with bounded depth. These lower bounds state that, in order to compute certain functions, these circuits need exactly as many OR gates as the respective DNFs. The proofs exploit a property of the functions that is based solely on prime implicant structure. Due to this feature, the lower bounds proved also hold for approximations of the considered functions that are similar to slice functions. Known lower bound arguments cannot handle these kinds of approximations. In order to show limitations of our approach, we prove that cliques of size $n-1$ can be detected in a graph with n vertices by monotone formulae with $O(\log n)$ OR gates.
Our lower bound for multilinear circuits improves a lower bound due to Borodin, Razborov and Smolensky for nondeterministic read-once branching programs computing the clique function.

1 Introduction

In this paper we consider Boolean circuits consisting of AND and OR gates. These circuits have variables and negated variables as inputs. Unless otherwise noted, all gates have fanin 2. A circuit without any negated inputs is called *monotone*. A circuit whose gates have fanout 1 is a *formula*. A *monom* is a conjunction of variables and negated variables. In this paper we regard monoms also as sets. An *implicant* of a Boolean function f is a monom that does not evaluate to 1 unless f does. An implicant is a *prime implicant* (minterm) if no new implicant can be obtained by removing variables or negated variables from the conjunction. For a Boolean function f, we denote the set of its prime implicants by $PI(f)$.

Until now the best known lower bounds for non-monotone circuits are linear. However, there has been considerable success in proving superpolynomial lower bounds for *monotone* circuits. Nowadays we have several powerful techniques to prove lower bounds for monotone circuits: the method of approximations (Razborov [1]); the method of probabilistic amplifications for estimating the

[*] Partially supported by DFG grant SCHN 503/2-2.

depth of monotone circuits (Karchmer and Wigderson [2]); the rank argument for formulas (Razborov [3]) and span programs (Gál [4], Gál and Pudlák [5]).

Also, it is known that negation is almost powerless for so-called slice functions (see e.g. monographs [6,7,8]). The t-slice function of f is a function of the form $f \wedge T_t^n \vee T_{t+1}^n$, where T_t^n is the t-th threshold function in n variables. A super-polynomial lower bound for the *monotone* complexity of a slice function implies a lower bound of the same order for its non-monotone complexity. Unfortunately, the currently available arguments for proving monotone lower bounds seem to be incapable of yielding sufficient lower bounds for slice functions. Therefore it is justified to seek new methods for proving monotone lower bounds.

One property of t-slice functions which seems to make the known arguments unsuitable for them is that they accept *all* inputs with more than t ones. The available proof methods rely on adequate sets of inputs which are mapped to 0 by the function considered. That t-slice functions accept all inputs with more than t ones seems to be an obstacle to constructing adequate sets of rejected inputs. Therefore it is justified to seek lower bound arguments for functions of the form $f \vee T_{t+1}^n$ that share this problematic property with slice functions; because of this similarity, we will refer to functions of the form $f \vee T_{t+1}^n$ as *t-pseudoslice functions* in the sequel.

In this paper we make some steps in this direction. We propose proof methods for some restricted circuit models that avoid these shortcomings. In particular, the properties of functions that we exploit are based solely on the prime implicant structure and do not rely on any additional information about prime clauses or rejected inputs. In this sense our lower bound arguments are "asymmetric". Unlike the currently available arguments, they are applicable to certain pseudoslice functions as well.

Moreover, the lower bounds we prove are optimal for the circuit classes considered. They state that multilinear circuits and circuits with sufficiently small alternation depth require exactly as many OR gates as the DNFs of the considered functions. This means that by using these circuit types instead of DNFs, we cannot even save a single OR gate! In other words, the DNFs are "incompressible" when we restrict ourselves to the respective circuit classes.

2 Results

A Boolean circuit is *multilinear* if the inputs to each of its AND gates are computed from disjoint sets of variables. To be more precise, for a gate g let $var\,(g)$ be the set of variables that occur in the subcircuit rooted at g. A Boolean circuit is multilinear if $var\,(g_1) \cap var\,(g_2) = \emptyset$ for each of its AND gates g with inputs g_1 and g_2. Multilinear circuits have been studied in [9,10] ([10] uses a slightly less restrictive definition). Multilinear circuits are a generalization of nondeterministic read-once branching programs, which have received much attention (see e.g. monograph [11]). Boolean multilinear circuits are related to arithmetic multilinear circuits which are characterized by the restriction that the highest power of the polynomials computed at their gates is no larger than 1. Arithmetic multilinear circuits have been studied in [12,13,14]. The direct arithmetic counterpart to Boolean multilinear circuits are syntactic multilinear circuits, defined by Raz [14].

It is clear that every Boolean function f can be computed by a multilinear circuit with $|PI(f)| - 1$ OR gates: just take the DNF of f. Many functions commonly referred to have multilinear circuits that are much smaller than their DNFs. Consider the threshold function T_k^n as an example. The threshold function T_k^n has $\binom{n}{k}$ prime implicants, but can be computed by a multilinear circuit of size $O(nk)$ [11, chapter 4]. Thus, the gap between the size of a smallest multilinear circuit which computes a certain function and the size of the DNF of this function can be exponential. It is also known that the gap between multilinear complexity and monotone complexity is exponential [9].

We identify a class of functions whose multilinear circuits require exactly as many OR gates as their DNF, the so-called union-free functions. We call a monotone Boolean function *union-free* if the union of any two of its prime implicants does not contain a new prime implicant.

Theorem 1. *Let f be a monotone union-free function. Then any multilinear circuit for f must have at least $|PI(f)| - 1$ OR gates.*

In the proof of this theorem we establish the following property of union-free functions: among the optimal (with respect to the number of OR gates) circuits there is one which is a formula, and for each of its AND gates, at least one input to the gate computes a monom.

The clique function $CLIQUE(n, s)$ is the function on $\binom{n}{2}$ variables representing the edges of an undirected graph G whose value is 1 iff G contains an s-clique. The clique function is a prominent example of a union-free function.

Lemma 1. *The function $CLIQUE(n, s)$ is union-free.*

Proof. Suppose the union of two distinct s-cliques A and B contains all edges of some third clique C. Since all three cliques are distinct and have the same number of vertices, C must contain a vertex u which does not belong to A and a vertex v which does not belong to B. This already leads to a contradiction because either the vertex u (if $u = v$) or the edge $\{u, v\}$ (if $u \neq v$) of C would remain uncovered by the cliques A and B. □

Corollary 1. *Multilinear circuits for $CLIQUE(n, s)$ require $\binom{n}{s} - 1$ OR gates (just as many as the DNF of this function).*

Because nondeterministic read-once branching programs can be simulated by multilinear circuits in a natural way, Corollary 1 improves the lower bound of $\exp(\Omega(\min(s, n - s)))$ given in [15] for nondeterministic read-once branching programs computing $CLIQUE(n, s)$.

Our lower bound for multilinear circuits also holds for certain pseudoslices of union-free functions. We call a monotone function *k-homogeneous* if each of its prime implicants has k variables.

Theorem 2. *Let f be a monotone k-homogeneous union-free function. Then any monotone multilinear circuit which computes the t-pseudoslice of f such that $t \geq 2k$ must have at least $|PI(f)| - 1$ OR gates.*

The next result we discuss shows that the union-freeness property is not sufficient for proving good lower bounds for *general* monotone circuits. By Corollary 1, $CLIQUE(n, n-1)$ requires $n-1$ OR gates to be computed by a multilinear circuit. On the other hand, we have the following upper bound.

Theorem 3. *The function $CLIQUE(n, n-1)$ can be computed by a monotone formula with $O(\log n)$ OR gates.*

This is apparently the first non-trivial upper bound for the monotone complexity of the clique function. The only other upper bound for the clique function that we are aware of is given in [6] and is only for its non-monotone complexity.

A circuit has *alternation depth d* iff d is the highest number of blocks of OR gates and blocks of AND gates on paths from input to output gates. A Σ_d-circuit (respectively, Π_d-circuit) is a circuit with alternation depth d such that the output gate is an OR gate (AND gate, respectively). We give incompressibility results, similar to those for multilinear circuits, also for monotone Σ_4-circuits. A Boolean function is *s-disjoint* if any two of its prime implicants do not have s variables in common.

Theorem 4. *Let f be a monotone k-homogeneous s-disjoint function such that $|PI(f)| \leq (k/2s)^{k/2s}$. Then every monotone Σ_4-circuit for f must have at least $|PI(f)| - 1$ OR gates.*

The same also holds for any t-pseudoslice of f such that $t \geq k^2/2s$.

Let $POLY(q, s)$ be the polynomial function introduced by Andreev [16]. This function has $n = q^2$ variables corresponding to the points in the grid $GF(q) \times GF(q)$, where q is a prime power. The function $POLY(q, s)$ accepts a $q \times q$ 0-1 matrix $X = (x_{i,j})$ iff there is a polynomial $p(z)$ of degree at most $s-1$ over $GF(q)$ such that $x_{i,p(i)} = 1$ for all $i \in GF(q)$. If $s < q/2$, then $POLY(q, s)$ is another example of a union-free function.

The function $POLY(q, s)$ is q-homogeneous. This function is also s-disjoint because the graphs of two distinct polynomials of degree at most $s-1$ cannot share s points. This together with $|PI(POLY(q, s))| = q^s$ and Theorem 4 leads to the following corollary.

Corollary 2. *If $s \leq \sqrt{q}/2$, then any monotone Σ_4-circuit for $POLY(q, s)$ must have at least $q^s - 1$ OR gates (just as many as the DNF of this function).*

The construction used in the proof of Theorem 3 yields a Π_3-formula. Hence, Theorem 4 suggests that it is harder to prove upper bounds for sufficiently disjoint functions because an efficient monotone circuit for them must be more complicated than a Σ_4-circuit. It is not even clear whether these polynomial functions $POLY(q, s)$ with $s \leq \sqrt{q}/2$ can be computed by general monotone circuits that are smaller than the respective DNFs.

The rest of the paper is devoted to the proof of our theorems.

3 Lower Bounds for Multilinear Circuits

In this section we prove Theorems 1 and 2. The following lemma allows us to restrict ourselves to *monotone* multilinear circuits. It is a special case of a theorem given in [17] for read-once nondeterministic machines.

Lemma 2. *If f is a monotone function, then any optimal multilinear circuit for f is monotone.*

Our next lemma describes a restriction of multilinear circuits which leads to exponential lower bounds for certain monotone Boolean functions. Given a prime implicant p, we show that certain variables of p can be substituted by some variables of another prime implicant p', yielding a "derived" implicant of the function. We say a path from a gate to the output of a circuit is *consistent* with a monom m if m is an implicant of all the functions computed at the gates along this path. We call a gate g *necessary* for an implicant m of a circuit S if m is not an implicant of the circuit $S_{g \to 0}$ we obtain from S by replacing g with the constant 0. Clearly, for every gate g which is necessary for an implicant m of S, there is a path from g to the output of S which is consistent with m. Let $PI_g\,(f)$ denote the set of prime implicants of f that g is necessary for. By $PI\,(g)$ we denote the set of prime implicants of the function computed at gate g.

Lemma 3 (Exchange Lemma). *Let g be a gate in a monotone multilinear circuit S for a function f, w be a path from g to the output of S and $p, p' \in PI_g\,(f)$ such that w is consistent with p. Let $m \subseteq p$ and $m \subseteq p'$ be distinct prime implicants in $PI\,(g)$.*

(i) The path w is consistent with the derived monom $(p \setminus m) \cup m'$. This means in particular that the derived monom $(p \setminus m) \cup m'$ is also an implicant of f.

(ii) If f is union-free, then the identity $p = (p' \setminus m') \cup m$ holds.

(iii) If f is a t-pseudoslice of a monotone k-homogeneous union-free function f^* such that $t \geq 2k$ and p, p' are prime implicants of f^* as well, then the same identity $p = (p' \setminus m') \cup m$ also holds.

Proof. (i) We have to show that $(p \setminus m) \cup m'$ is an implicant of all functions computed along w ($g = g_1, \ldots, g_t$). We prove this by induction on the length of the path w. For $g_1 = g$ the claim is correct since $(p \setminus m) \cup m'$ is a superset of $m' \in PI\,(g_1)$. For the inductive step, assume that $q \in PI\,(g_i)$ such that $q \subseteq (p \setminus m) \cup m'$. If g_{i+1} is an OR gate, then q is an implicant of g_{i+1}. If g_{i+1} is an AND gate, then let h be the other gate feeding it. We know that p is an implicant of the function computed at g_{i+1}. Hence, there must be some $m_h \in PI\,(h)$ such that $m_h \subseteq p$. Because the circuit is multilinear, we have $var\,(g_i) \cap var\,(h) = \emptyset$. Gate g belongs to the subcircuit rooted at gate g_i. We conclude that $var\,(g) \subseteq var\,(g_i)$ and that $var\,(g) \cap var\,(h) = \emptyset$. Since a variable of a prime implicant of a gate must occur somewhere in the subcircuit rooted at that gate, we conclude from $m \in PI\,(g)$ and $m_h \in PI\,(h)$ that $m \cap m_h = \emptyset$. Now we can see that $q \cup m_h$, an implicant of the function computed at g_{i+1}, is a subset of $(p \setminus m) \cup m'$.

(ii) According to (i), the monom $(p \setminus m) \cup m'$ is an implicant of f. Clearly, $(p \setminus m) \cup m' \subseteq p \cup p'$. Since f is union-free, this implies $p \subseteq (p \setminus m) \cup m'$ or $p' \subseteq (p \setminus m) \cup m'$. Because m and m' are distinct prime implicants, we have $m \not\subseteq m'$ and $m \not\supseteq m'$. The inclusion $p \subseteq (p \setminus m) \cup m'$ is impossible because $m \not\subseteq m'$. So $p' \subseteq (p \setminus m) \cup m'$ holds, this implies $m' \supseteq p' \setminus p$.

Since its assumptions are symmetrical, claim (i) also implies that $(p' \setminus m') \cup m$ is an implicant of f. Arguing in the same way as above we conclude that $p \subseteq (p' \setminus m') \cup m$. Since $m' \supseteq p' \setminus p$, we have $(p' \setminus m') \cup m \subseteq p$. Because p is a prime implicant of f, this means $p = (p' \setminus m') \cup m$.

(iii) We observe that the assumptions allow us to reason the same way as in (ii). Again, the monom $(p \setminus m) \cup m'$ is an implicant of f. We have $|(p \setminus m) \cup m'| \leq 2k$ because $|p| = k$ and $|m'| \leq |p'| = k$. Thus, $(p \setminus m) \cup m'$ must also be an implicant of f^*. Since f^* is union-free, we can conclude in the same way as in (ii) that $m' \supseteq p' \setminus p$. As in (ii), $(p' \setminus m') \cup m$ is an implicant of f and also of f^* since $|(p' \setminus m') \cup m| \leq 2k$. We may now proceed as in (ii) and conclude that $p = (p' \setminus m') \cup m$. □

We call a monotone circuit *broom-like* if, for each of its AND gates with inputs g_1 and g_2, $|PI(g_1)| = 1$ or $|PI(g_2)| = 1$ (or both). Thus, broom-like circuits have a particularly simple structure, and there is a direct correspondence between their prime implicants and their OR gates.

Lemma 4. *Every monotone multilinear circuit S for a union-free function f can be transformed into a broom-like formula for f with at most as many OR gates as S.*

Proof. We first transform S into a broom-like multilinear *circuit* for f without an increase in the number of OR gates. For this we need to know the following.

Claim 1. *Let g be an AND gate with inputs g_1 and g_2. Then there exists m in $PI(g_1) \cup PI(g_2)$ such that $m \subseteq p$ for all $p \in PI_g(f)$.*

Proof. Suppose there is no suitable m in $PI(g_1)$. We show that then there must be an m in $PI(g_2)$ such that $m \subseteq p$ for all p in $PI_g(f)$. Since there is no suitable m in $PI(g_1)$, $PI_g(f)$ cannot be empty. We pick some arbitrary p' in $PI_g(f)$. Because p' is an implicant of the function computed at g, there must be some m'_2 in $PI(g_2)$ such that $m'_2 \subseteq p'$. We prove that in fact

$$m'_2 \subseteq p \text{ for all } p \in PI_g(f) \ .$$

We distinguish two cases. First note that there must be an m'_1 in $PI(g_1)$ such that $m'_1 \subseteq p'$.

Case 1: $m'_1 \not\subseteq p$. Then there is some m_1 in $PI(g_1)$ such that $m_1 \subseteq p$, since p is an implicant of the function computed at g. Lemma 3(ii) yields that $p = (p' \setminus m'_1) \cup m_1$. Hence, $m'_2 \subseteq p$ because $m'_2 \subseteq p'$ and $m'_1 \cap m'_2 = \emptyset$ due to the multilinearity of the circuit.

Case 2: $m'_1 \subseteq p$. Note that there must be some $p'' \in PI_g(f)$ such that $m'_1 \not\subseteq p''$ because, by our initial assumption, $m'_1 \in PI(g_1)$ cannot be a suitable choice of m. There must be some m''_1 in $PI\,(g_1)$ with $m''_1 \subseteq p''$. Case 1 applies to p'' because $m'_1 \not\subseteq p''$, and we conclude $m'_2 \subseteq p''$. We use Lemma 3 again and find that $p = (p'' \setminus m''_1) \cup m'_1$. Hence, $m'_2 \subseteq p$ because $m'_2 \subseteq p''$ and $m''_1 \cap m'_2 = \emptyset$ due to the multilinearity of the circuit. □

We describe a modification that can be applied to every AND gate g which prevents S from being broom-like. Let g_1 and g_2 be the gates that feed g. The gate g prevents S from being broom-like, so $|PI\,(g_1)| > 1$ and $|PI\,(g_2)| > 1$. Let m be the monom in $PI\,(g_i)$ $(i \in \{1, 2\})$ given by Claim 1. We add a new gate h that computes m (along with the corresponding subcircuit for this computation). Then we disconnect g from g_i and feed g from h instead of g_i. Clearly, the resulting circuit S' rejects all the inputs that the original circuit rejected, since we are dealing with monotone circuits. Because S' accepts all inputs that $S_{g\to 0}$ accepts, g must be necessary for any prime implicant p of S that is not a prime implicant of S'. But according to Claim 1, after the modification every such p remains an implicant of the function computed at g. This way we obtain a broom-like multilinear circuit S^* for f without an increase in the number of OR gates.

We now describe a way of transforming a broom-like multilinear circuit S^* for f into a broom-like formula F for f without an increase in the number of OR gates.

Claim 2. *Let g be a gate in S^*. Then*

(i) there is some m in $PI\,(g)$ such that $m \subseteq p$ for all p in $PI_g\,(f)$, or

(ii) there is some path w from g to the output of S^* that is consistent with all $p \in PI_g\,(f)$.

Proof. We show that if (i) does not hold, then (ii) follows. This proof has a similar structure compared to the proof of the first claim. Since (i) does not hold, $PI_g\,(f)$ cannot be empty. So there is some $p' \in PI_g\,(f)$ and some path w' from g to the output of S^* that is consistent with p'. We prove that in fact

$$w' \text{ is consistent with } p \text{ for all } p \in PI_g\,(f)\ .$$

We distinguish two cases. First note that there is some $m' \in PI\,(g)$ with $m' \subseteq p'$ because p' is an implicant of the function computed at g.

Case 1: $m' \not\subseteq p$. There must be some $m \in PI\,(g)$ such that $m \subseteq p$ because p is an implicant of the function computed at g. Lemma 3 yields that $p = (p' \setminus m') \cup m$ and that w' is consistent with p.

Case 2: $m' \subseteq p$. Because (i) does not hold, there is some p'' in $PI_g\,(f)$ such that $m' \not\subseteq p''$. There must be some m'' in $PI\,(g)$ with $m'' \subseteq p''$. Case 1 applies to p'' because $m' \not\subseteq p''$, and we conclude that w' is consistent with p''. Lemma 3 tells us that $p = (p'' \setminus m'') \cup m'$ and that w' is consistent with p. □

We now describe a modification that we carry out for every gate g of S^* with fanout larger than 1 in order to reduce its fanout to 1. As with the modification for making the circuit broom-like, we only have to check the prime implicants for which g is necessary. We distinguish two cases according to Claim 2.

Case 1: There is some m in $PI\left(g\right)$ such that $m \subseteq p$ for all p in $PI_g\left(f\right)$. We remove g from the circuit and replace all wires from g by subcircuits that each compute m. The resulting circuit computes a function that is clearly implied by all prime implicants p in $PI_g\left(f\right)$.

Case 2: There is some path w from g to the output of S^* that is consistent with all p in $PI_g\left(f\right)$. We then cut all wires stemming from g that are not on path w, i.e. we replace inputs to other gates from g by the constant 0. All prime implicants in $PI_g\left(f\right)$ are preserved because after the modification w is still consistent with all of them. To see this, note that, due to the multilinearity of the circuit, every AND gate on w can have at most one input that depends on g (such an input must be on w itself). □

The following lemma enables us to count the prime implicants of monotone functions by counting the OR gates of their monotone broom-like formulas.

Lemma 5. *Let F be a monotone broom-like formula computing f. Then F has at least $\left|PI\left(f\right)\right| - 1$ OR gates.*

Proof. The lemma can be proved by induction on the size of the formula. The details are omitted. □

Theorem 1 follows immediately from Lemma 4 together with Lemma 5.

To verify Theorem 2, we use Lemma 3(iii) in place of Lemma 3(ii). The construction of Lemma 4 then yields a broom-like formula for a function $\widetilde{f}$ such that $PI\left(\widetilde{f}\right) \supseteq PI\left(f\right)$. The lower bound then follows with Lemma 5.

4 An Upper Bound for the Clique Function

We will use the following lemma.

Lemma 6. *Let G be a graph with n vertices. If its complement $\overline{G}$ does not contain a triangle and does not have two edges which are not incident with a common vertex, then G has an $(n-1)$-clique.*

Proof. Suppose G does not have an $n-1$-clique. Then $\overline{G}$ is not a star. Suppose $\overline{G}$ does not have two edges which are not incident with a common vertex. Choose arbitrary distinct edges e_1 and e_2 in $\overline{G}$. Let e_1 and e_2 be incident with the common vertex u. Since $\overline{G}$ is not a star, there is an edge e_3 which is not incident with u. Let e_2 and e_3 be incident with the common vertex $v \neq u$. e_1 and e_3 must share the common vertex w, which is distinct from u and v. Hence, u, v and w form a triangle in $\overline{G}$. □

Proof of Theorem 3 To design the desired Π_3-formula for $CLIQUE\left(n, n-1\right)$ we use a code $C \subseteq A^k$ for some k over an alphabet A with a constant number of symbols (independent of n) such that $|C| \geq n$ and the minimal distance d of

C is larger than $3k/4$. The existence of such a code of length $k = O(\log n)$ is guaranteed by the Gilbert bound (see e.g. [18]).

We assign to each vertex x (and hence, to each $(n-1)$-clique $V \setminus \{x\}$) its own codeword $\mathit{code}\,(x) \in C$. For each $1 \leq i \leq k$ and $a \in A$, let $S_{i,a}$ be the intersection of all $(n-1)$-cliques whose codes have symbol a in the i-th position. Hence,

$$S_{i,a} = V \setminus \{x \in V \mid \mathit{code}\,(x) \text{ has symbol } a \text{ in position } i\} \ . \tag{1}$$

Let $m_{i,a}$ be the monom consisting of all variables which correspond to edges having both their endpoints in $S_{i,a}$ (if $|S_{i,a}| \leq 1$, we set $m_{i,a} = 1$). We give the following Π_3-formula F for $\mathit{CLIQUE}\,(n, n-1)$:

$$F = \bigwedge_{i=1}^{k} \bigvee_{a \in A} m_{i,a} \,.$$

Every $(n-1)$-clique $V \setminus \{x\}$ with $\mathit{code}\,(x) = (a_1, \ldots, a_k)$ is accepted by the monom $\bigwedge_{i=1}^{k} m_{i,a_i}$ because the clique $V \setminus \{x\}$ contains all the cliques S_{i,a_i}, $i = 1, \ldots, k$. Hence, every $(n-1)$-clique is accepted by F. It remains to show that F does not accept any graph without an $(n-1)$-clique.

Let G be a graph accepted by F. Then there is a sequence $a_1, \ldots, a_k$ of symbols in A such that G is accepted by the monom $\bigwedge_{i=1}^{k} m_{i,a_i}$. For a vertex $x \in V$, let

$$P_x = \{i \mid \mathit{code}\,(x) \text{ has symbol } a_i \text{ in position } i\} \ .$$

Since the code C has minimal distance $d > 3k/4$, this implies that for every two distinct vertices x and y,

$$|P_x \cap P_y| \leq k - d < k/4 \,. \tag{2}$$

Let $\{x, y\}$ be an edge of the complement graph $\overline{G}$. Then the edge $\{x, y\}$ cannot belong to any of the monoms $m_{1,a_1}, \ldots, m_{k,a_k}$, implying that $x \notin S_{i,a_i}$ or $y \notin S_{i,a_i}$ for all $i = 1, \ldots, k$. According to (1) this means that for all $i = 1, \ldots, k$, $\mathit{code}\,(x)$ or $\mathit{code}\,(y)$ has symbol a_i at position i. So we have

$$P_x \cup P_y = [k] = \{1, \ldots, k\} \ . \tag{3}$$

Now we are able to show that G must contain an $(n-1)$-clique. We do so by showing that its complement $\overline{G}$ does not contain a triangle and does not contain a pair of vertex disjoint edges. The result then follows with Lemma 6.

Assume first that $\overline{G}$ contains a triangle with vertices u, v and w. By (3), we have that $P_u \cup P_w = [k]$ and $P_v \cup P_w = [k]$. Taking the intersection of these two equations yields

$$(P_u \cap P_v) \cup P_w = [k] \,.$$

But by (2), we have that $|P_u \cap P_v| < k/4$, so $|P_w| > 3k/4$. Similarly we obtain $|P_u| > 3k/4$, implying that $|P_u \cap P_w| > k/2$, a contradiction with (2).

Assume now that $\overline{G}$ contains a pair of vertex disjoint edges $\{u, v\}$ and $\{x, y\}$. By (3), we have $P_u \cup P_v = [k]$ and $P_x \cup P_y = [k]$. Assume w.l.o.g. that $|P_u| \geq |P_v|$. Then $|P_u| \geq k/2$. We know that

$$P_u = P_u \cap [k] = P_u \cap (P_x \cup P_y) = (P_u \cap P_x) \cup (P_u \cap P_y) \,.$$

Assume w.l.o.g. that $|P_u \cap P_x| \geq |P_u \cap P_y|$. Then $|P_u \cap P_x| \geq |P_u|/2 \geq k/4$, a contradiction with (2). □

5 Lower Bounds for Monotone Σ_4-Circuits

The following lemma shows that the union-freeness property is a special case of the disjointness property. This lemma names the properties of sufficiently disjoint functions that we exploit when proving the lower bound of Theorem 4.

Lemma 7. *Let $p_1, ..., p_r$ be prime implicants of a monotone Boolean function f and m be an implicant of f. Let f be k-homogeneous and k/r-disjoint.*

(i) If $\bigcup_{i=1}^{r} p_i \supseteq m$, then $m \supseteq p_i$ for some i.

(ii) If $x_1, \ldots, x_r$ are variables such that $x_i \in p_i$ and $x_i \notin p_j$ for $i \neq j$, then $\bigcup_{i=1}^{r} (p_i \setminus \{x_i\})$ is not an implicant of f.

Proof. (i) There must be some prime implicant p of f with $m \supseteq p$. Since $\bigcup_{i=1}^{r} p_i \supseteq p$, p must share at least k/r variables with some p_i. Because f is k/r-disjoint, this implies $p = p_i$. Claim (ii) is a direct consequence of (i). □

The following lemma deals with Π_3-circuits with gates of *unbounded* fanin.

Lemma 8. *Let f be a monotone k-homogeneous and s-disjoint function. If $r \leq k/2s$ and h is a function such that $h \leq f$ (i.e., f evaluates to 1 if h does) and $|PI\,(h) \cap PI\,(f)| \geq r$, then any monotone Π_3-circuit for h with bottom fanin at most $s-1$ must have top fanin at least $(k/2s)^r$.*

Proof. Let S be a monotone Π_3-circuit with top fanin a and bottom fanin at most $s-1$. Let $F_1, \ldots, F_a$ be the functions computed by the Σ_2-subcircuits of S that are inputs to the AND gate which is the output gate of S. The function F computed by S can be represented in the form

$$F = \bigwedge_{i=1}^{a} F_i \,.$$

Let $a < (k/2s)^r$. We now show that the circuit S must then make an error, i.e. that $F \neq h$. For the sake of contradiction, assume that $F = h$. We choose arbitrary prime implicants $p_1, \ldots, p_r \in PI\,(h) \cap PI\,(f)$. Our goal is to pick $x_1 \in p_1, \ldots, x_r \in p_r$ suitable for Lemma 7(ii), yielding $F \neq h$.

We pick the x_is in the order indicated by their indices. During this process we consider the preliminary monoms

$$m_t = \bigcup_{i=1}^{t} (p_i \setminus \{x_i\}) \;,\; t = 1, \ldots, r \,.$$

The preliminary monom m_t is available after the t-th step of the process. Finally, m_r is the desired implicant needed for the contradiction with Lemma 7(ii). Let A_t denote the set of indices of the functions F_i which are not implied by m_t, i.e. $i \in A_t$ iff m_t is not an implicant of F_i.

Claim 3. *There is always a choice of x_t in order to make*

$$|A_t| \leq \frac{|A_{t-1}|}{k/2s}\,.$$

Proof. We describe a choice of x_t that makes A_t sufficiently small. For every i in A_{t-1} we choose some $m_i \in PI\,(F_i)$ with $p_t \supseteq m_i$. Every F_i has such a prime implicant because p_t is a prime implicant of $h = F$. As x_t, we pick a variable of p_t that does not belong to any other of the prime implicants $p_1, \ldots, p_r$. Since each of the prime implicants can share at most $s-1$ variables with each of the other $r-1$ prime implicants, the prime implicant p_t has at least $k-(s-1)\,(r-1)$ variables which do not belong to any of the other prime implicants. Of these "private" variables of p_t, at most $s-1$ can belong to some particular monom m_i we chose. If we add all the occurrences of the private variables of p_t in the monoms m_i together, we count at most $(s-1)\,|A_{t-1}|$ occurrences. Using that p_t has at least $k-(s-1)\,(r-1)$ private variables, we find that at least one of these variables is in not more than $|A_{t-1}|\,/\,(k/2s)$ of the chosen monoms. This sufficiently "rare" variable is our choice of x_t. Since only those $i \in A_{t-1}$ remain in A_t for which x_t belongs to the chosen monom m_i, the desired bound for $|A_t|$ follows. □

We now finish the proof of Lemma 8. We start with $|A_0| = a < (k/2s)^r$. According to the claim, we can always choose the $x_1, \ldots, x_r$ such that A_r is empty. This means the finally constructed monom m_r is in fact an implicant of F. □

Proof of Theorem 4 (Sketch) Let S be a monotone Σ_4-circuit with gates of fanin 2 which computes a monotone k-homogeneous s-disjoint function f. We assume that S has the smallest possible number of OR gates. The function f can be represented, according to the structure of S, as a disjunction of functions f_i which are computed by the Π_3-subcircuits of S: $f = \bigvee f_i$. Let f_i be computed by the Π_3-circuit S_i. Without loss of generality we can assume that no Π_1-subcircuit of any S_i depends on more than $s-1$ variables, i.e. every S_i has bottom fanin at most $s-1$ when regarded as a circuit of unbounded fanin.

Every prime implicant of f must be a prime implicant of at least one of the f_i. Let R be the largest number of prime implicants of f that are prime implicants of one particular $f_i = h$. Let h be computed by the Π_3-circuit $S_i = H$. Under our assumption that S is optimal with respect to the number of OR gates used, we conclude that the case $2 \leq R < k/2s$ cannot occur. Otherwise, by Lemma 8, H would require at least $(k/2s)^R - 1 \geq R^2 - 1$ OR gates and could be replaced by a simple two-level circuit requiring only $R-1$ OR gates.

In the case $R = 1$ the circuit S is essentially a DNF and needs $|PI(f)| - 1$ OR gates. In the remaining case $R \geq k/2s$ Lemma 8 yields that H has a top fanin of at least $(k/2s)^{k/2s}$. The inequality $(k/2s)^{k/2s} \geq |PI(f)|$ is assumed by Theorem 4, so the desired lower bound for the number of OR gates in S follows.

Acknowledgement I am grateful to Stasys Jukna for helpful discussions.

References

1. Razborov, A.: Lower bounds for the monotone complexity of some Boolean functions. Sov. Math., Dokl. **31** (1985) 354–357
2. Karchmer, M., Wigderson, A.: Monotone circuits for connectivity require superlogarithmic depth. SIAM J. Discrete Math. **3** (1990) 255–265
3. Razborov, A.: Applications of matrix methods to the theory of lower bounds in computational complexity. Combinatorica **10** (1990) 81–93
4. Gál, A.: A characterization of span program size and improved lower bounds for monotone span programs. Comput. Complexity **10** (2001) 277–296
5. Gál, A., Pudlák, P.: A note on monotone complexity and the rank of matrices. Inf. Process. Lett. **87** (2003) 321–326
6. Wegener, I.: The complexity of Boolean functions. Wiley-Teubner Series in Computer Science. John Wiley & Sons Ltd., Chichester (1987)
7. Dunne, P.E.: The complexity of Boolean networks. Volume 29 of APIC Studies in Data Processing. Academic Press Ltd., London (1988)
8. Savage, J.E.: Models of computation: Exploring the power of computing. Addison-Wesley Publishing Company, Reading, MA (1998)
9. Sengupta, R., Venkateswaran, H.: Multilinearity can be exponentially restrictive (preliminary version). Technical Report GIT-CC-94-40, Georgia Institute of Technology. College of Computing (1994)
10. Ponnuswami, A.K., Venkateswaran, H.: Monotone multilinear boolean circuits for bipartite perfect matching require exponential size. In Lodaya, K., Mahajan, M., eds.: FSTTCS. Volume 3328 of Lecture Notes in Computer Science., Springer (2004) 460–468
11. Wegener, I.: Branching programs and binary decision diagrams. Theory and applications. SIAM Monographs on Discrete Mathematics and Applications (2000)
12. Nisan, N., Wigderson, A.: Lower bounds on arithmetic circuits via partial derivatives. Comput. Complexity **6** (1996/97) 217–234
13. Raz, R.: Multi-linear formulas for permanent and determinant are of super-polynomial size. In Babai, L., ed.: STOC, ACM (2004) 633–641
14. Raz, R.: Multilinear-$NC_1 \neq$ Multilinear-NC_2. In: FOCS, IEEE Computer Society (2004) 344–351
15. Borodin, A., Razborov, A.A., Smolensky, R.: On lower bounds for read-k-times branching programs. Comput. Complexity **3** (1993) 1–18
16. Andreev, A.: On a method for obtaining lower bounds for the complexity of individual monotone functions. Sov. Math., Dokl. **31** (1985) 530–534
17. Grigni, M., Sipser, M.: Monotone complexity. In Paterson, M.S., ed.: Boolean function complexity. Volume 169 of London Mathematical Society Lecture Note Series. Cambridge University Press, Cambridge (1992) 57–75
18. van Lint, J.H.: Introduction to coding theory. Volume 86 of Graduate Texts in Mathematics. Springer-Verlag, New York (1982)