On the Incompressibility of Monotone DNFs

Matthias P. Krieger*
Johann Wolfgang Goethe-Universität Frankfurt am Main
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 formulas 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

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

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

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 the function $f \wedge T_t^n \vee T_{t+1}^n$, where T_t^n is the t-th threshold function of n variables. The function T_t^n assumes the value 1 if and only if at least t of its n inputs are 1. A superpolynomial lower bound on the *monotone* complexity of a slice function implies a lower bound of the same order on 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. However, we also give an upper bound that shows that some of these DNFs are still highly compressible in the case of general monotone circuits.

We now sketch our main results and describe the organization of the paper. In Section 2 we introduce union-free functions and multilinear circuits. A monotone Boolean function is *union-free* if the union of any two of its prime implicants does not contain a new prime implicant. We identify two prominent union-free functions: the clique function and the so-called polynomial function (which is union-free for suitable parameters). A Boolean circuit is multilinear if the inputs to each of its AND gates are computed from disjoint sets of variables. Multilinear circuits are a generalization of nondeterministic read-once branching programs and ordered binary decision diagrams, which have received much attention (see e.g. monograph [9]). Hence, multilinear circuits are capable of computing numerous functions efficiently. We show that multilinear circuits for union-free functions are incompressible. Thus, multilinear circuits are inefficient for union-free functions, although they are efficient for many other functions.

In Section 3 we show that our lower bounds for multilinear circuits cannot be extended to unrestricted monotone circuits. We prove that cliques of size $n-1$ in an n-vertex graph

can be detected by monotone formulas with $O(\log n)$ OR gates. The DNF of this clique function has $n-1$ OR gates. Since the clique function is union-free, multilinear circuits for this clique function are incompressible and require $n-1$ OR gates as well. Hence, general monotone formulas for this clique function can be much more efficient than multilinear circuits. By exploiting that this particular clique function is a projection of almost all clique functions, we are able to show that general monotone circuits require less OR gates than DNFs for clique functions in general. The formulas we construct for proving this upper bound are Π_3-formulas, i.e. they are conjunctions of disjunctions of monoms.

In Section 4 we prove lower bounds for monotone circuits of bounded depth. Specifically, we show that monotone Σ_4-circuits for a certain class of polynomial functions are incompressible, i.e. they require at least as many OR gates as the DNFs of the respective functions. The class of Σ_4-circuits includes the Π_3-formulas, for which we proved the upper bound in the previous section. This means that the polynomial functions studied in this section are in a certain sense harder to compute than clique functions, whose Π_3-formulas are compressible. We still do not know any non-trivial upper bound for the polynomial function.

In Section 5 we prove that our lower bounds for multilinear circuits and monotone Σ_4-circuits also hold for certain pseudoslice functions. We show how the proofs we gave for our lower bounds can easily be adapted to make them work for pseudoslices.

We first make some preliminary remarks before discussing the results in detail. In this paper we consider Boolean circuits consisting of AND and OR gates. Sometimes we also introduce gates that assume the constant values 0 and 1. The 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. Therefore, we can compare monoms as we compare sets. For example, for monoms m_1 and m_2 we write $m_1 \subseteq m_2$ if every variable and negated variable of m_1 also belongs to m_2. 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* 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 $PI(f)$. We call a monotone function *k-homogeneous* if each of its prime implicants has k variables. A *disjunctive normal form (DNF)* is a disjunction of monoms. In this paper we always presume that a DNF is minimal, i.e. the DNF consists of a minimal number of monoms. The minimal DNF of a monotone function is the disjunction of all the prime implicants.

2 Multilinear Circuits Are Inefficient for Union-Free Functions

In this section we introduce multilinear circuits and union-free functions. We show that multilinear circuits for union-free functions are incompressible.

2.1 Union-Free Functions

The following definition of union-free functions will allow us to prove optimal lower bounds for multilinear circuits.

Definition 1. A monotone Boolean function is *union-free* if the union of any two of its prime implicants does not contain a new prime implicant.

The clique function $CLIQUE(n,s)$ is a function of $\binom{n}{2}$ variables representing the edges of an undirected n-vertex graph G. The function $CLIQUE(n,s)$ assumes the value 1 iff G contains an s-clique. This 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. □

Let $POLY(q,s)$ be the polynomial function introduced by Andreev [10]. 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 $f(z)$ of degree at most $s-1$ over $GF(q)$ such that $x_{i,f(i)} = 1$ for all $i \in GF(q)$. For certain parameters the polynomial function is another example of a union-free function.

Lemma 2. *If $s \leq q/2$, then the function $POLY(q,s)$ is union-free.*

Proof. The prime implicants of the function $POLY(q,s)$ are of the form $\bigwedge_{i \in GF(q)} x_{i,f(i)}$ for some polynomial $f(z)$ of degree at most $s-1$. The function $POLY(q,s)$ is *s-disjoint*, i.e. two distinct prime implicants of this function cannot have s variables in common. Otherwise, two distinct polynomials of degree at most $s-1$ would assume the same values at s points, which is impossible.

To prove the lemma, assume that p_1, p_2 and p_3 are distinct prime implicants of $POLY(q,s)$ and that $p_3 \subseteq p_1 \cup p_2$. Then p_3 must have $q/2 \geq s$ variables in common with p_1 or p_2, a contradiction. □

2.2 Multilinear Circuits

Since the term "multilinear" has been first used to describe a restriction on arithmetic circuits, we discuss arithmetic multilinear circuits before turning to Boolean multilinear circuits. An *arithmetic circuit* performs computations in a field. The gates of the circuit compute the field operations + and ×. The inputs of the circuit are variables and field elements.

A polynomial is multilinear if in each of its monomials the power of every variable is at most one. An arithmetic circuit is multilinear if every polynomial computed by some gate of the circuit is multilinear. Multilinear arithmetic circuits were defined in [11]. Raz [12] proved a superpolynomial gap between the size of multilinear arithmetic circuits and the size of multilinear arithmetic formulas.

Raz [13] introduced syntactic multilinear circuits which are slightly more restricted than multilinear circuits. In order to define syntactic multilinear circuits, let $var(g)$ be the set of variables that occur in the subcircuit rooted at the gate g of some circuit. An arithmetic circuit is syntactic multilinear if $var(g_1) \cap var(g_2) = \emptyset$ for each of its $\times$-gates with inputs g_1 and g_2. Every syntactic multilinear circuit is multilinear, but not vice versa. Raz [13] showed that multilinear *formulas* can be converted to syntactic multilinear formulas without an increase in size.

We now turn to Boolean multilinear circuits. In order to define Boolean multilinear circuits, let $var(g)$ again be the set of variables that occur in the subcircuit rooted at the gate g of some circuit.

Definition 2. 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.

In other words, a Boolean circuit is multilinear if the inputs to each of its AND gates are computed from disjoint sets of variables. Our definition of Boolean multilinear circuits is equivalent to the definition of multilinear circuits in [14]. This notion of Boolean multilinear circuits closely mimics the definition of arithmetic syntactic multilinear circuits. In [15] a slightly less restrictive definition of Boolean multilinear circuits is used which resembles the concept of arithmetic multilinear circuits more closely. While it may be possible to prove our results about multilinear circuits using the more general definition of [15], this appears to require significantly more sophisticated proofs, so we decide to limit ourselves to the more restrictive notion of multilinearity.

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. Multilinear Boolean circuits are a generalization of nondeterministic read-once branching programs and ordered binary decision diagrams. A simulation of nondeterministic branching programs by circuits is given in [16] (there nondeterministic branching programs are referred to as directed switching networks). Thus, 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 $O(nk)$. The construction of an efficient ordered binary decision diagram for T_k^n can be found in [9, chapter 4]. Hence, 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.

Let $CONN(n)$ be the function whose argument is the adjacency matrix of a directed n-vertex graph and assumes the value 1 if and only if the graph is connected. Sengupta and Venkateswaran have proved the following theorem.

Theorem 1 ([14]). *Multilinear circuits for $CONN(n)$ have at least $\sqrt{\frac{1}{n} \cdot \left(\frac{4}{3}\right)^{n-1}}$ gates.*

Since the function $CONN(n)$ can be computed by monotone circuits of polynomial size, this shows that the gap between multilinear complexity and monotone complexity is also exponential. Let $BPM(n)$ be the function which decides if an n-vertex bipartite graph has a perfect matching. The following theorem is from Ponnuswami and Venkateswaran.

Theorem 2 ([15]). *Multilinear circuits for $BPM(n)$ have at least $\Omega\left(2^{0.459n}\right)$ gates.*

In the next subsection we prove lower bounds for multilinear circuits of some other functions. While we use a slightly less general notion of multilinearity than in [15], we are able to prove stronger lower bounds which are even optimal.

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. We give an alternative proof that uses the specific restrictions of multilinear circuits.

Lemma 3. *If f is a monotone function, then any optimal multilinear circuit for f is monotone.*

Proof. Let S be an optimal multilinear circuit for f. We take the notion of a parse-graph G of S from [14]: The parse-graph G includes the output of S; for any OR gate v of G, exactly one immediate predecessor of v is included as its only predecessor in G; and for any AND gate v included in G, both immediate predecessors are included as predecessors of v in G. The parse-graph G can be viewed as a kind of circuit that accepts a subset of the inputs that S accepts. Since S is multilinear, a variable can occur at most once in G, so a variable and its negation can never both appear in G. This means that the conjunction of all variables and negated variables in G is consistent, and an implicant of f. So the set of all non-negated variables in G must contain a prime implicant of f.

Every input that a circuit accepts is accepted by one of its parse-graphs. Therefore, we can set all inputs of a multilinear circuit for f that are fed from negated variables to 1. Clearly, the variable set of every parse-graph of the resulting circuit will still contain a prime implicant of f because f is monotone. □

2.3 The Lower Bound for Multilinear Circuits

We now give the lower bound for multilinear circuits:

Theorem 3. *Let f be a monotone union-free function. Then any multilinear circuit for f must have at least $|PI(f)| - 1$ OR gates.*

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, the bound of $\exp(\Omega(s \log(n/s)))$ given by Corollary 1 improves the bound of $\exp(\Omega(\min(s, n-s)))$ given in [18] for nondeterministic read-once branching programs computing $CLIQUE(n, s)$.

Corollary 2. *If $s \leq q/2$, then any multilinear circuit for POLY (q, s) has $q^s - 1$ OR gates (just as many as the DNF of this function).*

For the proof of Theorem 3 we first give a lemma that describes a restriction of multilinear circuits. This restriction leads to exponential lower bounds for certain monotone Boolean functions. Given a prime implicant p, we show that, depending on the circuit, certain variables of p can be substituted by some variables of another prime implicant p'. This yields a "derived" implicant of the function computed by the circuit. If the function is union-free, we are able to reason further about the derived implicant.

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.

Lemma 4. *For every gate g which is necessary for an implicant m of S, there is a path from the output of S to g which is consistent with m.*

Proof. First note that m is an implicant of the function computed by an OR gate h iff m is an implicant of one of the inputs to h. Analogously, the implicant m is an implicant of an AND gate h iff m is an implicant of both of the inputs to h.

We find a consistent path in S from the output to g by descending into the circuit starting at the output. Doing so, we compare the two circuits S and $S_{g\to 0}$ with each other. We require that our path consists of gates that are not implied by m in the modified circuit $S_{g\to 0}$. We start with the output gate as the first gate of the path. Assume we have followed the path $g_1, \ldots, g_i$ and we are not done since $g_i \neq g$. We must pick the next gate g_{i+1} on our path. If g_i is an OR gate, then we choose g_{i+1} as the input to g_i that is implied by m in S. Since both inputs to g_i are not implied by m in $S_{g\to 0}$, our choice of g_{i+1} is also not implied by m in $S_{g\to 0}$, as we require. If g_i is an AND gate, then we choose g_{i+1} as the input to g_i that is not implied by m in $S_{g\to 0}$. Since both inputs to g_i are implied by m in S, our choice of g_{i+1} is also implied by m in S, as we require.

Finally we must reach g while constructing the path, since every leaf node in $S_{g\to 0}$ which is not g does not differ from the corresponding node in S. □

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 5 (Exchange Lemma). *Let g be a gate in a monotone multilinear circuit S for a function f and p, p' be prime implicants in $PI_g(f)$. Let $m \subseteq p$ and $m' \subseteq p'$ be distinct prime implicants in $PI(g)$.*

(i) If w is a path from g to the output of S that is consistent with p, then 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.

Proof. (i) We first note that the substitution of the variables of m by the variables of m' is valid at gate g. Then we observe that the substitution remains valid along the path w due to the multilinearity of the circuit.

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 Lemma 4, there is path from g to the output of S that is consistent with p, because g is necessary for p. Therefore, according to (i), the monom $(p \setminus m) \cup m'$ is an implicant of f. Clearly, we have $(p \setminus m) \cup m' \subseteq p \cup p'$. 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 $(p' \setminus m') \cup m \subseteq p$. From the two inclusions $p \subseteq (p' \setminus m') \cup m$ and $(p' \setminus m') \cup m \subseteq p$ we derive $p = (p' \setminus m') \cup m$. □

We now show how to transform a multilinear circuit for a union-free function into a normal form. 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 6. *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. If $PI(g_1)$ is not empty, then there exists a monom 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 some 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. Since g is an AND gate, the input g_1 is also necessary for p. Therefore we can apply Lemma 5(ii), which 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. Case 1 applies to p'' because $m_1' \not\subseteq p''$, and we conclude $m_2' \subseteq p''$. There must be some m_1'' in $PI\,(g_1)$ with $m_1'' \subseteq p''$. We use Lemma 5 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^* such that $PI\,(g) \neq \emptyset$. Then*

(i) there is some monom 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 Claim 1. Since (i) does not hold, $PI_g\,(f)$ cannot be empty. So there is some $p' \in PI_g\,(f)$ and, according to Lemma 4, 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 5 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 $m' \not\subseteq p''$. Case 1 applies to p'' because $m' \not\subseteq p''$, and we conclude that w' is consistent with p''. There must be some m'' in $PI(g)$ with $m'' \subseteq p''$. Lemma 5 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. The pathological case $PI(g) = \emptyset$ ($g = 0$) is trivial, so it suffices to discuss the two cases listed in Claim 2.

Case 1: There is some m in $PI(g)$ such that $m \subseteq p$ for all p in $PI_g(f)$. 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 $PI_g(f)$.

Case 2: There is some path w from g to the output of S^* that is consistent with all p in $PI_g(f)$. 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 $PI_g(f)$ 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 7. *Let F be a monotone broom-like formula computing f. Then F has at least $|PI(f)| - 1$ OR gates.*

Proof. We prove the lemma by induction on the size of the formula. If F does not contain any OR gates, it is clear that the claim holds. Let F_1 and F_2 be formulas computing the monotone functions f_1 and f_2, respectively. Since

$$|PI(f_1 \vee f_2)| \leq |PI(f_1)| + |PI(f_2)| ,$$

$$|PI(f_1 \vee f_2)| - 1 \leq ((|PI(f_1)| - 1) + (|PI(f_2)| - 1)) + 1 ,$$

so the claim holds for $F_1 \vee F_2$. So let us turn to the case of conjunction. W.l.o.g. let f_1 be a monom. Then

$$|PI(f_1 \wedge f_2)| \leq |PI(f_2)| ,$$

so the claim holds in this case too. □

Theorem 3 follows immediately from Lemma 6 together with Lemma 7. Recall that, according to Lemma 3, it is enough to consider monotone multilinear circuits.

3 An Upper Bound for the Clique Function

In this section we show that the union-freeness property is not sufficient for proving good lower bounds for unrestricted monotone circuits. By Corollary 1, the function $CLIQUE\,(n, n-1)$ requires $n-1$ OR gates to be computed by a multilinear circuit. On the other hand, we prove the following upper bound in this section.

Theorem 4. *The function $CLIQUE\,(n, n-1)$ can be computed by a monotone formula with $O\,(\log n)$ OR gates.*

Thus, general monotone circuits for the clique function can be much more efficient than multilinear circuits. 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.

We will use the following lemma for proving the upper bound.

Lemma 8. *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 to 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 to a common vertex. Choose arbitrary distinct edges e_1 and e_2 in $\overline{G}$. Let e_1 and e_2 be incident to the common vertex u. Since $\overline{G}$ is not a star, there is an edge e_3 which is not incident to u. Let e_2 and e_3 be incident to the common vertex $v \neq u$. The edges 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 $\overline{G}$. □

We are now ready to prove Theorem 4.

Proof of Theorem 4. To design the desired formula for $CLIQUE\,(n, n-1)$ we use an error correcting 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. [19]).

We assign to each vertex x (and hence, to each $(n-1)$-clique $V \setminus \{x\}$) its own codeword $code\,(x) \in C$. 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,

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

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 claim that the formula

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

computes $CLIQUE\,(n, n-1)$. Clearly, this formula has $k\,(|A|-1) = O(\log n)$ OR gates. Using distributivity we obtain the following representation of the function computed by F:

$$F = \bigvee_{(a_1,\ldots,a_k)\in A^k} \bigwedge_{i=1}^{k} m_{i,a_i}\,. \tag{2}$$

Every $(n-1)$-clique $V \setminus \{x\}$ with $code\,(x) = (a_1, \ldots, a_k)$ 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, by (2) 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 by (2) 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

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

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

$$|P_x \cap P_y| \leq k - d < k/4\,. \tag{3}$$

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 $i = 1, \ldots, k$, $code\,(x)$ or $code\,(y)$ has symbol a_i at position i. So we have

$$P_x \cup P_y = [k] = \{1, \ldots, k\}\ . \tag{4}$$

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 8.

Assume first that $\overline{G}$ contains a triangle with vertices u, v and w. By (4), 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 (3), 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 (3).

Assume now that $\overline{G}$ contains a pair of vertex disjoint edges $\{u, v\}$ and $\{x, y\}$. By (4), 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 (3). □

Theorem 4 tells us that monotone circuits for the function $CLIQUE\,(n, n-1)$ are compressible, i.e. for sufficiently large n they require less OR gates than the respective DNF. We now show that monotone circuits for most other clique functions are also compressible. Thus, our optimal lower bounds for the clique function from the previous section cannot be extended to general monotone circuits in any way.

Corollary 3. *There exists some s_0 such that, for all $s \geq s_0$ and $n > s$, the function $CLIQUE\,(n, s)$ has monotone circuits with less OR gates than the DNF of this function.*

Proof. We pick s_0 such that every function $CLIQUE\,(s+1, s)$ with $s \geq s_0$ is compressible. This is possible according to Theorem 4. Now we show that a clique function $CLIQUE\,(n, s)$ with $s \geq s_0$ and $n > s$ and is compressible. We choose arbitrary $s+1$ vertices in the n-vertex graph taken by $CLIQUE\,(n, s)$. Let $p_1, \ldots, p_{s+1}$ be the prime implicants that correspond to the s-cliques of the $s+1$ chosen vertices. We denote the other prime implicants of $CLIQUE\,(n, s)$ by $q_1, \ldots, q_k$, so we can write the DNF of $CLIQUE\,(n, s)$ as

$$CLIQUE\,(n, s) = \left(\bigvee_{i=1}^{s+1} p_i\right) \vee \left(\bigvee_{i=1}^{k} q_i\right) . \tag{5}$$

Here the disjunction $\bigvee_{i=1}^{s+1} p_i$ is the DNF of the function $CLIQUE\,(s+1, s)$, so this term can be computed by a monotone circuit with less than s OR gates. Hence, according to (5) the function $CLIQUE\,(n, s)$ can be computed by a monotone circuit with less OR gates than the DNF of this function. □

4 Lower Bounds for Monotone Σ_4-Circuits

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 gates to the output. A Σ_d-circuit (respectively, Π_d-circuit) is a circuit with alternation depth at most d such that the output gate is an OR gate (AND gate, respectively).

In this section we contrast the upper bound for the clique function proved in the previous section with a lower bound for functions that are even harder than the clique function in a certain sense. We introduced the polynomial function $POLY\,(q, s)$ in Section 2.1. For some polynomial functions we give incompressibility results, similar to those for multilinear circuits, also for monotone Σ_4-circuits. We show that monotone Σ_4-circuits for these functions require at least as many OR gates as the respective DNFs. The construction used in the proof of Theorem 4 yields a monotone Π_3-formula. A monotone Π_3-formula is a simple kind of monotone Σ_4-circuit. Thus, to prove upper bounds for the functions we study in this section, we would have to give a more elaborate construction than we did for the clique function. A monotone circuit for any of these functions that is more efficient than the DNF would have to be more complicated than a Σ_4-circuit. Therefore, these hard polynomial functions we investigate here are an interesting starting point for

looking for new lower bounds. It is not even clear whether these polynomial functions can be computed by unrestricted circuits that are smaller than the respective DNFs.

A Boolean function is *s-disjoint* if any two of its prime implicants do not have s variables in common. 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 for monotone Σ_4-circuits.

Lemma 9. *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. We restrict these circuits to depth 3. We require the output gate to be an AND gate (possibly with only one input) and the inputs to this gate to be OR gates. The *top fanin* is the fanin of the output gate. The *bottom fanin* is the maximal fanin of the AND gates representing Π_1-subcircuits (if there are no such subcircuits, we define the bottom fanin to be 1).

Lemma 10. *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 $s-1$, and let F be the function computed by S. 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 distinct 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 9(ii). Lemma 9(ii) then yields a monom m which, according to the Lemma, is not an implicant of h, but for which we show that it is an implicant of F. This way we obtain $F \neq h$ and contradict our assumption.

We pick the x_i's 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, the monom $m = m_r$ is the desired implicant of F needed for the contradiction with Lemma 9(ii).

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_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, since the circuit has a bottom fanin of at most $s-1$. 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

$$\frac{(s-1)\,|A_{t-1}|}{k-(s-1)\,(r-1)} \leq \frac{|A_{t-1}|}{k/2s}$$

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 10. 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. □

Since Σ_4-circuits can be broken up naturally into Π_3-circuits, our lower bound for monotone Σ_4-circuits follows easily from the previous lemma about monotone Π_3-circuits. We only have to pay attention to a few technicalities.

Theorem 5. *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.*

Proof. 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.

Without loss of generality we can assume that no Π_1-subcircuit of S depends on more than $s-1$ variables, i.e. S has bottom fanin at most $s-1$ when regarded as a circuit of unbounded fanin. We can do so because a monom m computed by a Π_1-subcircuit with more than $s-1$ variables can be implied by at most one prime implicant $p \in PI(f)$ (f is s-disjoint). We can remove this Π_1-subcircuit and, if m is implied by $p \in PI(f)$, add p to the top level disjunction of S. The function computed after the modification has the same prime implicants as the original one. This modification is allowed because it leaves the total number of OR gates (with fanin two) unchanged, and we are only interested in this number.

The function f can be represented as a disjunction of functions f_i which are computed by Π_3-circuits: $f = \bigvee f_i$. Let f_i be computed by the Π_3-circuit S_i. 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 $S_i = H$.

We claim that $2 \leq R < k/2s$ cannot hold. To see this, assume the contrary. View H as a Π_3-circuit of unbounded fanin. We can apply Lemma 10 to H. Lemma 10 yields that H must have a top fanin of at least $(k/2s)^R \geq (k/2s)^2 \geq R^2$. Note that we may assume w.l.o.g. that at most one of the inputs to the top level conjunction of H computes a monom. (We can replace several such inputs by one input computing the conjunction of the monoms.) Using this assumption, we conclude that H requires at least $R^2 - 1$ OR gates. However, a plain disjunction (DNF) of the prime implicants that h shares with f could do the same job that H does in S, and requires only $R - 1 < R^2 - 1$ OR gates of fanin 2. This contradicts our assumption that S has the smallest possible number of OR gates.

To finish the proof of Theorem 5, we distinguish the two remaining cases.

Case 1: $R = 1$. Then S is essentially a DNF and needs $|PI(f)| - 1$ OR gates.

Case 2: $R \geq k/2s$. Again we view H as a circuit with gates of unbounded fanin. Applying Lemma 10 with $r = k/2s$ yields that H must have a top fanin of at least $(k/2s)^{k/2s} \geq |PI(f)|$ (this inequality is stated as an assumption of the theorem). When built of fanin-2 gates, H requires at least $|PI(f)| - 1$ OR gates since again we may assume w.l.o.g. that at most one of the inputs to the top level conjunction of H consists of a single monom. □

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 5 leads to the following lower bound.

Corollary 4. *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).*

5 Lower Bounds for Pseudoslice Functions

The t-slice function of f is the function $f_t = f \wedge T_t^n \vee T_{t+1}^n$, where T_t^n is the t-th threshold function of n variables. Slice functions are studied because a superpolynomial lower bound on the monotone complexity of a slice function implies a lower bound of the same order on its non-monotone complexity (see e.g. monographs [6, 7, 8]). Thus, a superpolynomial lower bound for non-monotone circuits could be proved by proving a superpolynomial lower bound for a slice function. Known arguments for superpolynomial lower bounds still fail for slice functions. We suggest to approach this problem by studying the complexity of functions that are similar to slice functions. We call these functions *pseudoslice* functions. The t-pseudoslice function of f is the function $f'_t = f \vee T_{t+1}^n$. Let $|x|$ denote the number of ones in the Boolean vector x. Then an equivalent definition of the t-pseudoslice f'_t of f is

$$f'_t(x) = \begin{cases} f(x) & \text{for } |x| < t \\ f(x) & \text{for } |x| = t \\ 1 & \text{for } |x| > t \end{cases} .$$

On the other hand, the t-slice f_t of f assumes the following values:

$$f_t(x) = \begin{cases} 0 & \text{for } |x| < t \\ f(x) & \text{for } |x| = t \\ 1 & \text{for } |x| > t \end{cases} .$$

Thus, the t-slice and the t-pseudoslice function of f only differ for arguments with less than t ones.

We are able to show that our lower bounds for multilinear circuits and monotone Σ_4-circuits also hold for certain pseudoslice functions. We first show how to extend our lower bound for multilinear circuits to pseudoslice functions. The key step is to prove a variant of Lemma 5(ii) for pseudoslice functions.

In the proof of Lemma 5(ii), the union-freeness of f is only used to assert that the union of the prime implicants p and p' does not contain any new prime implicant. This allows us to state the following generalization of Lemma 5(ii).

Corollary 5 (Generalization of Lemma 5(ii)). *Let g be a gate in a monotone multilinear circuit for a function f and p, p' be prime implicants in $PI_g(f)$. Let $m \subseteq p$ and $m' \subseteq p'$ be distinct prime implicants in $PI(g)$. If f has no prime implicant other than p and p' which is a subset of $p \cup p'$, then the identity $p = (p' \setminus m') \cup m$ holds.*

With this corollary, it is easy to prove a variant of Lemma 5(ii) for pseudoslice functions.

Lemma 11 (Exchange Lemma for Pseudoslice Functions). *Let g be a gate in a monotone multilinear circuit for the t-pseudoslice f'_t of a monotone k-homogeneous union-free function f such that $t \geq 2k$. Let p and p' be prime implicants of f that are in $PI_g(f'_t)$. If $m \subseteq p$ and $m' \subseteq p'$ are distinct prime implicants in $PI(g)$, then the identity $p = (p' \setminus m') \cup m$ holds.*

Proof. We apply corollary 5 to f'_t. According to corollary 5, all we need to show is that f'_t has no prime implicant other than p and p' which is a subset of $p \cup p'$. The prime implicants of f'_t are the prime implicants of f, which have length k, and the prime implicants of T^n_{t+1}, which have length $t+1 \geq 2k+1$. Let $q \subseteq p \cup p'$ be a prime implicant of f'_t. We have $|p| = |p'| = k$ because f is k-homogeneous. We conclude $2k \geq |p \cup p'| \geq |q|$. Thus, the prime implicant q of f'_t must also be a prime implicant of f. Because f is union-free, this implies $q = p$ or $q = p'$. □

This lemma easily yields the lower bound for pseudoslice functions.

Theorem 6. *Let f be a monotone k-homogeneous union-free function. Then any multilinear circuit which computes the t-pseudoslice of f such that $t \geq 2k$ must have at least $|PI(f)| - 1$ OR gates (just as many as the DNF of this function).*

Proof. We adapt the proof of Lemma 6. The idea is to ignore the long prime implicants of the pseudoslice functions. We use Lemma 11 in place of Lemma 5(ii). This yields the following imitations of Claim 1 and Claim 2:

Claim 4. *Let g be an AND gate in a monotone multilinear circuit for the t-pseudoslice f'_t of f with inputs g_1 and g_2. If $PI(g_1)$ is not empty, then there exists a monom m in $PI(g_1) \cup PI(g_2)$ such that $m \subseteq p$ for all prime implicants $p \in PI(f) \cap PI_g(f'_t)$.*

Claim 5. *Let g be a gate in a monotone multilinear circuit for the t-pseudoslice f'_t of f such that $PI(g) \neq \emptyset$. Then*

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

(ii) there is some path w from g to the output that is consistent with all prime implicants $p \in PI(f) \cap PI_g(f'_t)$.

Using the same kind of circuit modifications as in the proof of Lemma 6, we are able to transform the original circuit for f'_t into a broom-like formula for a function $\widetilde{f}$ such that $PI\left(\widetilde{f}\right) \supseteq PI(f)$. The lower bound then follows with Lemma 7. □

Since the function $CLIQUE(n,s)$ is $s(s-1)/2$-homogeneous, we obtain the following lower bound.

Corollary 6. *Any multilinear circuit which computes the t-pseudoslice of $CLIQUE(n,s)$ such that $t \geq s(s-1)$ must have at least $\binom{n}{s} - 1$ OR gates.*

The function $POLY(q,s)$ is by definition q-homogeneous.

Corollary 7. *Any multilinear circuit which computes the t-pseudoslice of $POLY(q,s)$ such that $s \leq q/2$ and $t \geq 2q$ must have at least $q^s - 1$ OR gates.*

Next we show that our lower bounds for monotone Σ_4-circuits also hold for certain pseudoslices.

Theorem 7. *Let f'_t be the t-pseudoslice of a monotone k-homogeneous s-disjoint function f such that $|PI(f)| \leq (k/2s)^{k/2s}$ and $t \geq k^2/2s$. Then every monotone Σ_4-circuit for f'_t must have at least $|PI(f)| - 1$ OR gates.*

Proof. First we prove a version of Lemma 10 that also holds for functions $\widetilde{f}$ whose prime implicants are the prime implicants of f and perhaps some additional prime implicants of length more than t. We can proceed as in the proof of Lemma 10. We only need to deal with prime implicants of f. We determine an implicant m_r of the given Π_3-circuit for $\widetilde{f}$ in the same way. This implicant has at most $rk \leq k^2/2s$ variables. Hence, the monom m_r must also be an implicant of f, and we again find a contradiction with Lemma 9(ii).

We now adapt the proof of Theorem 5 in order to make it work for the pseudoslices we are dealing with here. We can basically leave it unchanged. Again, we only deal with prime implicants of f. In the proof of Theorem 5 we assume w.l.o.g. that no Π_1-subcircuit depends on more than $s-1$ variables. We give instructions there for modifying the circuit to make it meet this requirement. In the case of computing pseudoslices, these modifications may alter the function computed by the circuit, but we always preserve the prime implicants of f. As a result, we obtain a Σ_4-circuit that computes a function $\widetilde{f}$ as described above. So we can apply the modified version of Lemma 10 in the same way as we applied Lemma 10 in the proof of Theorem 5. This yields the lower bound. □

Corollary 8. *Any monotone Σ_4-circuit which computes the t-pseudoslice of $POLY(q,s)$ such that $s \leq \sqrt{q}/2$ and $t \geq q^2/2s$ must have at least $q^s - 1$ OR gates.*

6 Conclusion

We prove optimal lower bounds on the number of OR gates for multilinear circuits and monotone Σ_4-circuits. These kinds of circuits need as many OR gates as the DNFs of the functions considered. This incompressibility is an interesting property of the functions we study here, namely the clique function and the polynomial function. When dealing with more general circuit models, this may make it easier to prove lower bounds for the clique function and the polynomial function. We give an upper bound for the clique function in order to show that monotone circuits for the clique function require less OR gates than the respective DNFs in general. Hence, our incompressibility results for multilinear circuits computing the clique function cannot be extended to unrestricted monotone circuits. While our upper bound for the clique function also holds for monotone Σ_4-circuits, we give a class of polynomial functions whose monotone Σ_4-circuits are also incompressible. Thus, these polynomial functions are in this sense even harder to compute than clique functions. This observation makes the polynomial function interesting to study when looking for new lower bounds. It is an open problem to find a non-trivial upper bound for the polynomial function. Finally, we note that our lower bounds for multilinear circuits and monotone Σ_4-circuits also hold for certain pseudoslice functions. Since known lower bound arguments for unrestricted monotone circuits seem to fail for pseudoslice functions, our lower bounds

could be a starting point for the improvement of lower bounds for unrestricted monotone circuits.

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 super-logarithmic 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] Wegener, I.: Branching programs and binary decision diagrams. Theory and applications. SIAM Monographs on Discrete Mathematics and Applications (2000)

[10] Andreev, A.: On a method for obtaining lower bounds for the complexity of individual monotone functions. Sov. Math., Dokl. **31** (1985) 530–534

[11] Nisan, N., Wigderson, A.: Lower bounds on arithmetic circuits via partial derivatives. Comput. Complexity **6** (1996/97) 217–234

[12] Raz, R.: Multilinear-$NC_1 \neq$ Multilinear-NC_2. In: FOCS, IEEE Computer Society (2004) 344–351

[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] 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)

[15] 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

[16] Pudlák, P.: The hierarchy of Boolean circuits. Comput. Artificial Intelligence **6** (1987) 449–468

[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] Borodin, A., Razborov, A.A., Smolensky, R.: On lower bounds for read-k-times branching programs. Comput. Complexity **3** (1993) 1–18

[19] van Lint, J.H.: Introduction to coding theory. Volume 86 of Graduate Texts in Mathematics. Springer-Verlag, New York (1982)