BRICS

Basic Research in Computer Science

BRICS RS-03-45 Miltersen et al.: On converting CNF to DNF

On converting CNF to DNF

Peter Bro Miltersen
Jaikumar Radhakrishnan
Ingo Wegener

BRICS Report Series **RS-03-45**

ISSN 0909-0878 **December 2003**

See back inner page for a list of recent BRICS Report Series publications. Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK–8000 Aarhus C
Denmark

Telephone:	**+45 8942 3360**
Telefax:	**+45 8942 3255**
Internet:	**BRICS@brics.dk**

BRICS publications are in general accessible through the World Wide Web and anonymous FTP through these URLs:

`http://www.brics.dk`
`ftp://ftp.brics.dk`
This document in subdirectory `RS/03/45/`

On Converting CNF to DNF

Peter Bro Miltersen*[1], Jaikumar Radhakrishnan**[2], and Ingo Wegener***[3]

[1] Department of Computer Science, University of Aarhus, Denmark,
e-mail: bromille@daimi.au.dk.
[2] School of Technology and Computer Science,
Tata Institute of Fundamental Research, Mumbai 400005, India
e-mail:jaikumar@tifr.res.in
[3] FB Informatik LS2
University of Dortmund, 44221 Dortmund, Germany.
e-mail: wegener@ls2.cs.uni-dortmund.de.

Abstract. We study how big the blow-up in size can be when one switches between the CNF and DNF representations of boolean functions. For a function $f : \{0,1\}^n \rightarrow \{0,1\}$, $\mathsf{cnfsize}(f)$ denotes the minimum number of clauses in a CNF for f; similarly, $\mathsf{dnfsize}(f)$ denotes the minimum number of terms in a DNF for f. For $0 \le m \le 2^{n-1}$, let $\mathsf{dnfsize}(m,n)$ be the maximum $\mathsf{dnfsize}(f)$ for a function $f : \{0,1\}^n \rightarrow \{0,1\}$ with $\mathsf{cnfsize}(f) \le m$. We show that there are constants $c_1, c_2 \ge 1$ and $\epsilon > 0$, such that for all large n and all $m \in [\frac{1}{\epsilon}n, 2^{\epsilon n}]$, we have

$$2^{n - c_1 \frac{n}{\log(m/n)}} \le \mathsf{dnfsize}(m,n) \le 2^{n - c_2 \frac{n}{\log(m/n)}}.$$

In particular, when m is the polynomial n^c, we get $\mathsf{dnfsize}(n^c, n) = 2^{n - \theta(c^{-1} \frac{n}{\log n})}$.

1 Introduction

Boolean functions are often represented as disjunctions of terms (i.e. in DNF) or as conjunctions of clauses (i.e. in CNF). Which of these representations is preferable depends on the application. Some functions are represented more succinctly in DNF whereas others are represented more succinctly in CNF, and switching between these representations can involve an exponential increase in size. In this paper, we study how big this blow-up in size can be.

We recall some well-known concepts (for more details see Wegener [16]). The set of variables is denoted by $X_n = \{x_1, \ldots, x_n\}$. Literals are variables and negated variables. Terms are conjunctions of literals. Clauses are disjunctions of literals. Every Boolean function f can be represented as a conjunction of clauses,

$$\bigwedge_{i=1}^{s} \bigvee_{\ell \in C_i} \ell, \tag{1}$$

* Supported by **BRICS**, Basic Research in Computer Science (www.brics.dk), funded by the Danish National Research Foundation.
** Work done while the author was visiting Aarhus.
*** Supported by DFG-grant We 1066/9.

as well as a disjunction of terms,

$$\bigvee_{i=1}^{s} \bigwedge_{\ell \in T_i} \ell, \tag{2}$$

where T_i and C_i are sets of literals. The form (1) is usually referred to as *conjunctive normal form* (CNF) and the form (2) is usually referred to as *disjunctive normal form* (DNF), although it would be historically more correct to call them conjunctive and disjunctive forms and use *normal* only when the sets C_i and T_i have n literals on distinct variables. In particular, this would ensure that *normal forms* are *unique*. However, in the computer science literature such a distinction is not made, and we will use CNF and DNF while referring to expressions such as (1) or (2) even when no restriction is imposed on the sets C_i and T_i, and there is no guarantee of uniqueness. The *size* of a CNF is the number of clauses (the parameter s in (1)), and $\mathsf{cnfsize}(f)$ is the minimum number of clauses in a CNF for f. Similarly, $\mathsf{dnfsize}(f)$ is the minimum number of terms in a DNF for f.

We are interested in the maximal blow-up of size when switching from the CNF representation to the DNF representation (or vice versa). For $0 \le m \le 2^{n-1}$, let $\mathsf{dnfsize}(m, n)$ be the maximum $\mathsf{dnfsize}(f)$ for a function $f : \{0,1\}^n \to \{0,1\}$ with $\mathsf{cnfsize}(f) \le m$. Since $\wedge$ distributes over $\vee$, a CNF with m clauses each with k literals can be converted to a DNF with k^m terms each with at most m literals. If the clauses do not share any variable, this blow-up cannot be avoided. If the clauses don't share variables, we have $km \le n$, and the maximum $\mathsf{dnfsize}(f)$ that one can achieve by this method is $2^{\frac{n}{2}}$. Can the blow-up be worse? In particular, we want to know the answer to the following question:

> *For a function* $f : \{0,1\}^n \to \{0,1\}$*, how large can* $\mathsf{dnfsize}(f)$ *be if* $\mathsf{cnfsize}(f)$ *is bounded by a fixed polynomial in* n*?*

The problem is motivated by its fundamental nature: $\mathsf{dnfsize}(f)$ and $\mathsf{cnfsize}(f)$ are fundamental complexity measures. Practical circuit designs like programmable logic arrays (PLAs) are based on DNFs and CNFs. Lower bounds on unbounded fan-in circuits are based on the celebrated switching lemma of Håstad (1989) which is a statement about converting CNFs to DNFs where some variables randomly are replaced by constants. Hence, it seems that the exact relationship between CNFs and DNFs ought to be understood as completely as possible. Fortunately, CNFs and DNFs have simple combinatorial properties allowing the application of current combinatorial arguments to obtain such an understanding. In contrast, the results of Razborov and Rudich [13] show that this is not likely to be possible for complexity measures like circuit size and circuit depth.

Another motivation for considering the question is the study of SAT algorithms and heuristics with "mild" exponential behaviour; a study which has gained a lot of momentum in recent years (e.g., Monien and Speckenmeyer[10], Paturi et al. [11], Dantsin et al. [4], Schöning [14], Hofmeister et al. [7], and Dantsin et al. [5]). Despite many successes, the following fundamental question is still open: Is there an algorithm that decides SAT of a CNF with n variables and m clauses (without any restrictions on the length of clauses) in time $m^{O(1)}2^{cn}$ for some constant $c < 1$? The obvious brute force algorithm solves the problem in time $m^{O(1)}2^n$. One method for solving SAT is to convert the CNF to a DNF, perhaps using sophisticated heuristics to keep the final DNF and any intermediate results small (though presumably not optimally small, due to the hardness of such a task). Once converted to a DNF, satisfiability of the formula is trivial to decide. A CNF-DNF conversion method for solving SAT, phrased in a more general constraint satisfaction framework was recently studied experimentally by Katajainen and Madsen [8]. Answering the question above limits the worst case complexity of any algorithm obtained within this framework.

The monotone case: Our final motivation for considering the question comes from the monotone version of the problem. Let $\mathsf{dnfsize}^+(m, n)$ denote the maximum $\mathsf{dnfsize}(f)$ for a *monotone* function $f : \{0, 1\}^n \to \{0, 1\}$. In this case (see, e.g., Wegener [16, Chapter 2, Theorem 4.2]), the number of prime clauses of f is equal to $\mathsf{cnfsize}(f)$ and the number of prime implicants of f is equal to $\mathsf{dnfsize}(f)$. Our problem can then be modelled on a hypergraph H_f whose edges are precisely the prime clauses of f. A *vertex cover* or *hitting set* for a hypergraph is a subset of vertices that intersects every edge of the hypergraph. The number of prime implicants of f is precisely the number of minimal vertex covers in H_f. The problem of determining $\mathsf{dnfsize}^+(m, n)$ then immediately translates to the following problem on hypergraphs: *What is the maximum number of distinct minimal vertex covers in a hypergraph on n vertices with m distinct edges? In particular, how many minimal vertex covers can a hypergraph with $n^{O(1)}$ edges have?*

Previous work: Somewhat surprisingly, the exact question we consider does not seem to have been considered before, although some related research has been reported. As mentioned, Håstad's switching lemma can be considered as a result about approximating CNFs by DNFs. The problem of converting polynomial-size CNFs and DNFs into representations by restricted branching programs for the purpose of hardware verification has been considered since a long time (see Wegener [17]). The best lower bounds for ordered binary decision diagrams (OBDDs) and read-once branching programs (BP1s) are due to Bollig and Wegener [3] and are of size $2^{\Omega(n^{1/2})}$ even for monotone functions representable as disjunctions of terms of length 2.

The results in this paper: In Section 2, we show functions where the the blow-up when going from CNF to DNF is large:

$$\text{for } 2 \le m \le 2^{n-1}, \quad \mathsf{dnfsize}(m, n) \ge 2^{n-2\frac{n}{\log(m/n)}};$$
$$\text{for } 2 \le m \le \binom{n}{\lceil n \rceil 2}, \; \mathsf{dnfsize}^+(m, n) \ge 2^{n-n\frac{\log\log(m/n)}{\log(m/n)}-\log(m/n)}.$$

In particular, for $m = n^{O(1)}$, we have

$$\mathsf{dnfsize}(m, n) = 2^{n-O(\frac{n}{\log n})} \text{ and } \mathsf{dnfsize}^+(m, n) = 2^{n-O(\frac{n\log\log n}{\log n})}.$$

In Section 3, we show that functions with small CNFs do not need very large DNFs. There is a constant $c > 0$ such that for all large n and all $m \in [10^4 n, 2^{10^{-4}n}]$,

$$\mathsf{dnfsize}(m, n) \leq 2^{n - c\frac{n}{\log(m/n)}}.$$

In particular, for $m = n^{O(1)}$, we have $\mathsf{dnfsize}(m, n) = 2^{n-\Omega(n/\log n)}$.

For the class of CNF-DNF conversion based SAT algorithms described above, our results imply that no algorithm within this framework has complexity $m^{O(1)}2^{cn}$ for some constant $c < 1$, though we cannot rule out an algorithm of this kind with complexity $m^{O(1)}2^{n-\Omega(n/\log n)}$ which would still be a very interesting result.

2 Functions with a Large Blow-up

In this section, we show functions with small $\mathsf{cnfsize}$ but large $\mathsf{dnfsize}$. Our functions will be the conjunction of a small number of parity and majority functions. To estimate the $\mathsf{cnfsize}$ and the $\mathsf{dnfsize}$ of such functions, we will need a lemma. Recall, that a prime implicant t of a boolean function f is called an *essential* prime implicant if there is an input x such that $t(x) = 1$ but $t'(x) = 0$ for all other prime implicants t' of f. We denote the number of essential prime implicants of f by $\mathsf{ess}(f)$.

Lemma 1. *Let $f(x) = \bigwedge_{i=1}^{\ell} g_i(x)$, where the g_i's depend on disjoint sets of variables and no g_i is identically 0. Then,*

$$\mathsf{cnfsize}(f) = \sum_{i=1}^{\ell} \mathsf{cnfsize}(g_i) \quad \textit{and} \quad \mathsf{dnfsize}(f) \geq \mathsf{ess}(f) = \prod_{i=1}^{\ell} \mathsf{ess}(g_i).$$

Proof. First, consider $\mathsf{cnfsize}(f)$. This part is essentially Theorem 1 of Voigt and Wegener [15]. We recall their argument. Clearly, we can put together the CNFs of the g_i's and produce a CNF for f with size at most $\sum_{i=1}^{\ell} \mathsf{cnfsize}(g_i)$. To show that $\mathsf{cnfsize}(f) \geq \sum_{i=1}^{\ell} \mathsf{cnfsize}(g_i)$, let $\mathcal{C}$ be the set of clauses of the smallest CNF of f. We may assume that all clauses in $\mathcal{C}$ are prime clauses of f. Because the g_i's depend on disjoint variables, every prime clause of f is a prime clause of exactly one g_i. Thus we obtain a natural partition $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_\ell\}$ of $\mathcal{C}$ where each clause in $\mathcal{C}_i$ is a prime clause of g_i. Consider a setting to the variables of g_j $(j \neq i)$ that makes each such g_j take the value 1 (this is possible because no g_j is identically 0). Under this restriction, the function f reduces to g_i and all clauses outside $\mathcal{C}_i$ are set to 1. Thus, $g_i \equiv \bigwedge_{c \in \mathcal{C}_i} c$, and $|\mathcal{C}_i| \geq \mathsf{cnfsize}(g_i)$. The first claim follows from this.

It is well known since Quine [12] (see also, e.g., Wegener [16, Chapter 2, Lemma 2.2]) that $\mathsf{dnfsize}(f) \geq \mathsf{ess}(f)$. Also, it is easy to see that any essential prime implicant of f is the conjunction of essential prime implicants of g_i and every conjunction of essential prime implicants of g_i is an essential prime implicant of f. Our second claim follows from this. $\square$

We will apply the above lemma with the parity and majority functions as g_i's. It is well-known that the parity function on n variables, defined by

$$\mathsf{Par}_n(x) \triangleq \bigoplus_{i=1}^{n} x_i = \sum_{i=1}^{n} x_i \pmod 2,$$

has $\mathsf{cnfsize}$ and $\mathsf{dnfsize}$ equal to 2^{n-1}. For monotone functions, it is known that for the majority function on n variables, defined by

$$\mathsf{Maj}(x) = 1 \Leftrightarrow \sum_{i=1}^{n} x_i \geq \frac{n}{2},$$

has $\mathsf{cnfsize}$ and $\mathsf{dnfsize}$ equal to $\binom{n}{\lceil n/2 \rceil}$.

Definition 1. *Let the set of n variables $\{x_1, x_2, \ldots, x_n\}$ be partitioned into $\ell = \lceil n/k \rceil$ sets $S_1, \ldots S_\ell$ where $|S_i| = k$ for $i < \ell$. The functions $f_{k,n}, h_{k,n} : \{0,1\}^n \to \{0,1\}$ are defined as follows:*

$$f_{k,n}(x) = \bigwedge_{i=1}^{\ell} \bigoplus_{j \in S_i} x_j \quad \textit{and} \quad h_{k,n}(x) = \bigwedge_{i=1}^{\ell} \mathsf{Maj}(x_j : j \in S_i).$$

Theorem 1. *Suppose $1 \leq k \leq n$. Then*

$$\mathsf{cnfsize}(f_{k,n}) \leq \left\lceil \frac{n}{k} \right\rceil \cdot 2^{k-1} \quad \textit{and} \quad \mathsf{dnfsize}(f_{k,n}) = 2^{n - \lceil n/k \rceil};$$

$$\mathsf{cnfsize}(h_{k,n}) \leq \left\lceil \frac{n}{k} \right\rceil \cdot \binom{k}{\lceil k/2 \rceil} \quad \textit{and} \quad \mathsf{dnfsize}(h_{k,n}) \geq \binom{k}{\lceil k/2 \rceil}^{\lfloor n/k \rfloor}.$$

Proof. As noted above $\mathsf{cnfsize}(\mathsf{Par}_k) = 2^{k-1}$ and $\mathsf{cnfsize}(\mathsf{Maj}_n) = \binom{k}{\lceil k/2 \rceil}$. Also, it is easy to verify that $\mathsf{ess}(\mathsf{Par}_k) = 2^{k-1}$ and $\mathsf{ess}(\mathsf{Maj}_n) = \binom{k}{\lceil k/2 \rceil}$. Our theorem follows easily from this using Lemma 1. $\square$

Remark: One can determine the $\mathsf{dnfsize}$ of $f_{k,n}$ and $h_{k,n}$ directly using a general result of Voigt and Wegener [15], which states that the $\mathsf{dnfsize}(g_1 \wedge g_2) = \mathsf{dnfsize}(g_1) \cdot \mathsf{dnfsize}(g_2)$ whenever g_1 and g_2 are symmetric functions on disjoint sets of variables. This is not true for general functions g_1 and g_2 (see Voigt and Wegener [15]).

Corollary 1. *1. Let $2n \leq m \leq 2^{n-1}$. There is a function f with*

$$\mathsf{cnfsize}(f) \leq m \textit{ and } \mathsf{dnfsize}(f) \geq 2^{n - 2n/\log(m/n)}.$$

2. Let $4n \leq m \leq \binom{n}{\lceil n/2 \rceil}$. Then, there is a monotone function h with

$$\mathsf{cnfsize}(h) \leq m \textit{ and } \mathsf{dnfsize}(h) \geq 2^{n - n\frac{\log\log(m/n)}{\log(m/n)} - \log(m/n)}.$$

Proof. The first part follows from Theorem 1, by considering $f_{k,n}$ for $k = \lfloor \log_2(m/n) \rfloor$. The second part follows from the Theorem 1, by considering $h_{k,n}$ with the same value of k. We use the inequality $2^k/k \le \binom{k}{\lceil k/2 \rceil} \le 2^{k-1}$ (valid for $k \ge 2$). $\square$

Let us understand what this result says for a range of parameters, assuming n is large.

Case $m = cn$: There is a function with linear cnfsize but exponential dnfsize. For $\epsilon > 0$, by choosing $c = \theta(2^{2/\epsilon})$, the dnfsize can be made at least $2^{(1-\epsilon)n}$.

Case $m = n^c$: We can make $\mathsf{dnfsize}(f) = 2^{n - O(c^{-1} \frac{n}{\log n})}$. By choosing c large we obtain in the exponent an arbitrarily small constant for the $(n/\log n)$-term.

Case $m = 2^{o(n)}$: We can make $\mathsf{dnfsize}(f)$ grow at least as fast as $2^{n-\alpha(n)}$, for each $\alpha = \omega(1)$.

Monotone functions: We obtain a monotone function whose cnfsize is at most a polynomial $m = n^c$, but whose dnfsize can be made as large as $2^{n - \varepsilon \frac{n \log \log n}{\log n}}$. Here, $\varepsilon = O(c^{-1})$.

3 Upper Bounds on the Blow-up

In this section, we show the upper bound on $\mathsf{dnfsize}(m, n)$ claimed in the introduction. We will use restrictions to analyse CNFs. So, we first present the necessary background about restrictions, and then use it to derive our result.

3.1 Preliminaries

Definition 2 (Restriction). *A restriction on a set of variables V is a function $\rho : V \to \{0, 1, \star\}$. The set of variables in V assigned $\star$ by ρ are said to have been left free by ρ and denoted by $\mathsf{free}(\rho)$; the remaining variables $\mathsf{set}(\rho) = V - \mathsf{free}(\rho)$ are said to be set by ρ. Let $S \subseteq V$. We use $\mathcal{R}_S^V$ to denote the set of all restrictions ρ with $\mathsf{set}(\rho) = S$. For a Boolean function f on variables V and a restriction ρ, we denote by f_ρ the function with variables $\mathsf{free}(\rho)$ obtained from f by fixing all variables $x \in \mathsf{set}(V)$ at the value $\rho(x)$.*

The following easy observation lets us conclude that if the subfunctions obtained by applying restrictions have small dnfsize then the original function also has small dnfsize.

Lemma 2. *For all $S \subseteq V$ and all boolean functions f with variables V,*

$$\mathsf{dnfsize}(f) \le \sum_{\rho \in \mathcal{R}_S^V} \mathsf{dnfsize}(f_\rho).$$

Proof. Let Φ_{f_ρ} denote the smallest DNF for f_ρ. For a restriction $\rho \in \mathcal{R}_S^V$, let $t(\rho)$ be the term consisting of literals from variables in S that is made 1 by ρ and 0 by all other restrictions in $\mathcal{R}_S^V$. (No variables outside S appears in $t(\rho)$. Every variable in S appears in $t(\rho)$: the variable x appears unnegated if and only if $\rho(x) = 1$.) Then, $\Phi = \bigvee_{\rho \in \mathcal{R}_S^V} t(\rho) \wedge \Phi_{f_\rho}$ gives us a DNF for f of the required size. $\square$

In light of this observation, to show that the $\mathsf{dnfsize}$ of some function f is small, it suffices to somehow obtain restrictions of f that have small $\mathsf{dnfsize}$. Random restrictions are good for this. We will use random restrictions in two ways. If the clauses of a CNF have a small number of literals, then the *switching lemma* of Håstad[6] and Beame [1] when combined with Lemma 2 immediately gives us a small DNF (see Lemma 4 below). We are, however, given a general CNF not necessarily one with small clauses. Again, random restrictions come to our aid: with high probability large clauses are destroyed by random restrictions (see Lemma 5).

Definition 3 (Random restriction). *When we say that ρ is a random restriction on the variables in V leaving ℓ variables free, we mean that ρ is generated as follows: first, pick a set S of size $|V| - \ell$ at random with uniform distribution; next, pick ρ with uniform distribution from $\mathcal{R}_S^V$.*

We will need the following version of the switching lemma due to Beame [1].

Lemma 3 (Switching Lemma). *Let f be a function on n variables with a a CNF whose clauses have at most r literals. Let ρ be a random restriction leaving ℓ variables free. Then*

$$\Pr[f_\rho \text{ does not have a decision tree of depth } d] < (7r\ell/n)^d.$$

We can combine Lemma 2 and the switching lemma to obtain small DNFs for functions with CNFs with small clauses.

Lemma 4. *Let $1 \le r \le \frac{n}{100}$. Let f have a CNF on n variables where each clause has at most r literals. Then, $\mathsf{dnfsize}(f) \le 2^{n - \frac{1}{100} \cdot \frac{n}{r}}$.*

Proof. Let V be the set of variables of f. Let ρ be a random restriction on V that leaves $\ell = \left\lfloor \frac{1}{15} \cdot nr \right\rfloor$ variables free. By the switching lemma, with probability more than $1 - 2^{-d}$, f_ρ has a decision tree of depth at most d. We can fix $S \subseteq V$ so that this event happens with this probability even when conditioned on $\mathsf{set}(\rho) = S$, that is, when ρ is chosen at random with uniform distribution from $\mathcal{R}_S^V$. If f_ρ has a decision tree of depth at most d, then it is easy to see that $\mathsf{dnfsize}(f_\rho) \le 2^d$. In any case, $\mathsf{dnfsize}(f_\rho) \le 2^{\ell-1}$. Thus, by Lemma 2, we have

$$\mathsf{dnfsize}(f) \ \le \ 2^{n-\ell} \cdot 2^d + 2^{n-\ell} \cdot 2^{-d} \cdot 2^{\ell-1}.$$

Set $d = \left\lfloor \frac{\ell}{2} \right\rfloor$. Then, $\mathsf{dnfsize}(f) \le \sum_{\rho \in R_S^V} \mathsf{dnfsize}(f_\rho) \le 2^{n - \frac{\ell}{2} + 1} \le 2^{n - \frac{1}{100} \cdot \frac{n}{r}}$. $\square$

Lemma 5. *Let V be a set of n variables, and K a set of literals on distinct variables. Let $|K| = k$. Let ρ be a random restriction that leaves $\left\lfloor \frac{n}{2} \right\rfloor$ variables free. Then,*

$$\Pr_\rho[\textit{no literal in } K \textit{ is assigned } 1] \le 2e^{-\frac{k}{8}}.$$

Proof. Let W be the set of variables that appear in K either in negated or non-negated form. Using estimates for the tail of the hypergeometric distribution [2], we see first have

$$\Pr[|W \cap \mathsf{set}(\rho)| \le \frac{k}{4}] \;\le\; \exp(-\frac{k}{8}).$$

Furthermore,

$$\Pr[\text{no literal in } K \text{ is assigned } 1 \mid |W \cap \mathsf{set}(\rho)| \ge \frac{k}{4}] \le 2^{-\frac{k}{4}}.$$

Thus,

$$\Pr_{\rho}[\text{no literal in } K \text{ is assigned } 1] \le e^{-\frac{k}{8}} + 2^{-\frac{k}{4}} < 2e^{-\frac{k}{8}}.$$

□

3.2 Small DNFs from Small CNFs

We now show that the blow-up obtained in the previous section (see Corollary 1) is essentially optimal.

Theorem 2. *There is a constant $c > 0$, such that for all large n, and $m \in [10^4 n, 2^{10^{-4}n}]$,*

$$\mathsf{dnfsize}(m, n) \le 2^{n - c\frac{n}{\log(m/n)}}.$$

Proof. Let f be a Boolean function on a set V of n variables, and let Φ be a CNF for f with at most m clauses. We wish to show that f has a DNF of small size. By comparing the present bound with Lemma 4, we see that our job would be done if we could somehow ensure that the clauses in Φ have at most $O(\log(m/n))$ literals. All we know, however, is that Φ has at most m clauses. In order to prepare Φ for an application of Lemma 4, we will attempt to destroy the large clauses of Φ by applying a random restriction. Let ρ be a random restriction on V that leaves $\lfloor \frac{n}{2} \rfloor$ variables free. We cannot claim immediately that all large clause are likely to be destroyed by this restriction. Instead, we will use the structure of the surviving large clauses to get around them. The following predicate will play a crucial role in our proof.

> $\mathcal{E}(\rho)$: *There is a set $S_0 \subseteq \mathsf{free}(\rho)$ of size at most $n/10$ so that every clause of Φ that is not killed by ρ has at most $r \triangleq \lfloor 100 \log(m/n) \rfloor$ free variables outside S_0.*

Claim. $\Pr_\rho[\mathcal{E}(\rho)] \ge 1 - 2^{-\frac{n}{100}}$.

Before we justify this claim, let us see how we can exploit it to prove our theorem. Fix a choice of $S \subseteq V$ such that $\Pr[\mathcal{E}(\rho) \mid \mathsf{set}(\rho) = S] \ge 1 - 2^{-\frac{n}{100}}$. Let $F = V - S$. We will concentrate only on ρ's with $\mathsf{set}(\rho) = S$, that is, ρ's from the set $\mathcal{R}_S^V$. We will build a small DNF for f by putting together the DNFs for the different f_ρ's. The key point is that whenever $\mathcal{E}(\rho)$ is true, we will be able to show that f_ρ has a small DNF.

$\mathcal{E}(\rho)$ **is true:** Consider the set $S_0 \subseteq \mathsf{free}(\rho)$ whose existence is promised in the definition of $\mathcal{E}(\rho)$. The definition of S_0 implies that for each $\sigma \in \mathcal{R}^F_{S_0}$ all clauses of $\Phi_{\sigma\circ\rho}$ have at most r literals. By Lemma 4, $\mathsf{dnfsize}(f_{\sigma\circ\rho}) \leq 2^{|F|-|S_0|-\frac{|F|-|S_0|}{100r}}$, and by Lemma 2, we have

$$\mathsf{dnfsize}(f_\rho) \leq \sum_{\sigma\in\mathcal{R}^F_{S_0}} \mathsf{dnfsize}(f_{\sigma\circ\rho}) \leq 2^{|S_0|}2^{|F|-|S_0|-\frac{|F|-|S_0|}{100r}} \leq 2^{|F|-\frac{|F|-|S_0|}{100r}}.$$

$\mathcal{E}(\rho)$ **is false:** We have $\mathsf{dnfsize}(f_\rho) \leq 2^{|F|-1}$.

Using these bounds for $\mathsf{dnfsize}(f_\rho)$ for $\rho \in \mathcal{R}^V_S$ in Lemma 2 we obtain

$$\mathsf{dnfsize}(f) \leq 2^{|S|} \cdot 2^{|F|-\frac{|F|-|S_0|}{100r}} + 2^{|S|}2^{-\frac{n}{100}}2^{|F|-1} = 2^n(2^{-\frac{|F|-|S_0|}{100r}} + 2^{-\frac{n}{100}}).$$

The theorem follows from this because $|F| - |S_0| = \Omega(n)$ and $r = O(\log(m/n))$. We still have to prove the claim.

Proof of claim. Suppose $\mathcal{E}(\rho)$ is false. We will first show that there is a set of at most $\lceil n/(10(r+1))\rceil$ surviving clauses in Φ_ρ that together involve at least $n/10$ variables. The following sequential procedure will produce this set of clauses. Since $\mathcal{E}$ does not hold, there is some (surviving) clause c_1 of Φ_ρ with at least $r+1$ variables. Let T be the set of variables that appear in this clause. If $|T| \geq n/10$, then we stop: $\{c_1\}$ is the set we seek. If $|T| < n/10$, there must be another clause c_2 of Φ_ρ with $r+1$ variables outside T, for otherwise, we could take $S_0 = T$ and $\mathcal{E}(\rho)$ would be true. Add to T all the variables in c_2. If $|T| \geq n/10$, we stop with the set of clauses $\{c_1, c_2\}$; otherwise, arguing as before there must be another clause c_3 of Φ_ρ with $r+1$ variables outside T. We continue in this manner, picking a new clause and adding at least $r+1$ elements to T each time, as long as $|T| < \frac{n}{10}$. Within $\lceil n/(10(r+1))\rceil$ steps we will have $|T| \geq \frac{n}{10}$, at which point we stop.

For a set $\mathcal{C}$ of clauses of Φ, let $K(\mathcal{C})$ be a set of literals obtained by picking one literal for each variable that appears in some clause in $\mathcal{C}$. By the discussion above, for $\mathcal{E}(\rho)$ to be false, there must be some set $\mathcal{C}$ of clauses of Φ such that $|\mathcal{C}| \leq \lceil n/(10(r+1))\rceil \stackrel{\Delta}{=} a$, $K(\mathcal{C}) \geq \frac{n}{10}$ and no literal in $K(\mathcal{C})$ is assigned 1 by ρ. Thus, using Lemma 5, we have

$$\begin{aligned}
\Pr_\rho[\neg\mathcal{E}(\rho)] &\leq \sum_{\mathcal{C},|\mathcal{C}|\leq a,|K(\mathcal{C})|\geq\frac{n}{10}} \Pr_\rho[\text{no literal in } K(\mathcal{C}) \text{ is assigned 1 by } \rho] \\
&\leq \sum_{j=1}^{a}\binom{m}{j}\cdot 2e^{-\frac{n}{80}} \\
&\leq a\binom{m}{a}\cdot 2e^{-\frac{n}{80}} \\
&\leq a\left(\frac{em}{\lceil n/(10r)\rceil}\right)^{\lceil n/(10r)\rceil} 2e^{-\frac{n}{80}}
\end{aligned}$$

$$
\begin{aligned}
&\leq a\left(\frac{em10r}{n}\right)^{\lceil n/(10r)\rceil} 2e^{-\frac{n}{80}}\\
&\leq a\left(\frac{m}{n}\cdot 1000e\log(\frac{m}{n})\right)\lceil n/(10r)\rceil 2e^{-\frac{n}{80}}\\
&\leq a\left(\frac{m}{n}\cdot 1000e\log(\frac{m}{n})\right)\lceil n/(10r)\rceil 2e^{-\frac{n}{80}}\\
&\leq \left(\frac{m}{n}\right)^{2\lceil n/(10r)\rceil} 2e^{-\frac{n}{80}}\\
&\qquad \textit{Note: } \tfrac{m}{n}\geq 1000e\log(\tfrac{m}{n}) \text{ follows from } 10^4 n\leq m\\
&\leq a2^{2\log(\frac{m}{n})[\frac{n}{1000\log(\frac{m}{n})}+1]} 2e^{-\frac{n}{80}}\\
&\leq a2^{\frac{n}{500}+2\log(\frac{m}{n})} 2e^{-\frac{n}{80}}\\
&\leq a2^{\frac{n}{400}} 2e^{-\frac{n}{80}}\\
&\quad 2\log(m/n)\leq \tfrac{n}{5000} \text{ follows from } m\leq 2^{10^{-4}n}\\
&\leq 2^{\frac{n}{400}} 2^{-\frac{n}{80}} = 2^{-\frac{n}{100}}.
\end{aligned}
$$

This completes the proof of the claim. □

Conclusion and Open Problems

We have shown lower and upper bounds for $\mathsf{dnfsize}(m, n)$ of the form $2^{n-c\frac{n}{\log(m/n)}}$. The constant c in the lower and upper bounds are far, and it would be interesting to bring them closer, especially when $m = An$ for some constant A. Our bounds are not tight for monotone functions. In particular, what is the largest possible blow-up in size when converting a polynomial-size monotone CNF to an equivalent optimal-size monotone DNF? Equivalently, what is the largest possible number of distinct minimal vertex covers for a hypergraph with n vertices and $n^{O(1)}$ edges? We have given an upper bound $2^{n-\Omega(n/\log n)}$ and a lower bound $2^{n-O(n\log\log n/\log n)}$. Getting tight bounds seems challenging.

Acknowledgements: An earlier version of this paper was submitted to the conference on Mathematical Foundations of Computer Science (MFCS). We thank the referees of MFCS for their comments on that version. The conference version of this paper appeared in [9].

References

1. Beame, P.: A switching lemma primer. Technical Report UW-CSE-95-07-01, Department of Computer Science and Engineering, University of Washington (November 1994). Available online at `www.cs.washington.edu/homes/beame/`.
2. Chvátal, V.: The tail of the hypergeometric distribution. Discrete Mathematics **25** (1979) 285–287.
3. Bollig, B. and Wegener, I.: A very simple function that requires exponential size read-once branching programs. Information Processing Letters **66** (1998) 53–57.

4. Dantsin, E., Goerdt, A., Hirsch, E.A., and Schöning, U.: Deterministic algorithms for k-SAT based on covering codes and local search. Proceedings of the 27th International Colloquium on Automata, Languages and Programming. Springer. LNCS 1853 (2000) 236–247.
5. Dantsin, E., Goerdt, A., Hirsch, E.A., Kannan, R., Kleinberg, J., Papadimitriou, C., Raghavan, P., and Schöning, U.: A deterministic $(2-2/(k+1))^n$ algorithm for k-SAT based on local search. Theoretical Computer Science, to appear.
6. Håstad, J.: Almost optimal lower bounds for small depth circuits. In: Micali, S. (Ed.): *Randomness and Computation.* Advances in Computing Research, **5** (1989) 143–170. JAI Press.
7. Hofmeister, T., Schöning, U., Schuler, R., and Watanabe, O.: A probabilistic 3-SAT algorithm further improved. Proceedings of STACS, LNCS 2285 (2002) 192–202.
8. Katajainen, J. and Madsen, J.N.: Performance tuning an algorithm for compressing relational tables. Proceedings of SWAT, LNCS 2368 (2002) 398–407.
9. Miltersen, P.B., Radhakrishnan, J., and Wegener, I.: On converting CNF to DNF. Proceedings of MFCS, LNCS 2747 (2003) 612–621.
10. Monien, B. and Speckenmeyer, E.: Solving satisfiability in less than 2^n steps. Discrete Applied Mathematics **10** (1985) 287–295.
11. Paturi, R., Pudlàk, P., Saks, M.E., and Zane, F.: An improved exponential-time algorithm for k-SAT. Proceedings of the 39th IEEE Symposium on the Foundations of Computer Science (1998) 628–637.
12. W. V. O. Quine: On cores and prime implicants of truth functions. American Mathematics Monthly **66** (1959) 755–760.
13. Razborov, A. and Rudich, S.: Natural proofs. Journal of Computer and System Sciences **55** (1997) 24–35.
14. Schöning, U.: A probabilistic algorithm for k-SAT based on limited local search and restart. Algorithmica **32** (2002) 615–623.
15. Voigt, B., Wegener, I.: Minimal polynomials for the conjunctions of functions on disjoint variables an be very simple. Information and Computation **83** (1989) 65–79.
16. Wegener, I.: *The Complexity of Boolean Functions.* Wiley 1987. Freely available via http://ls2-www.cs.uni-dortmund.de/∼wegener.
17. Wegener, I.: *Branching Programs and Binary Decision Diagrams – Theory and Applications.* SIAM Monographs on Discrete Mathematics and Applications 2000.

Recent BRICS Report Series Publications

RS-03-45 **Peter Bro Miltersen, Jaikumar Radhakrishnan, and Ingo Wegener.** ***On converting CNF to DNF*****. December 2003. 11 pp. A preliminary version appeared in Rovan and Vojtás, editors,** ***Mathematical Foundations of Computer Science: 28th International Symposium*****, MFCS '03 Proceedings, LNCS 2747, 2003, pages 612–621.**

RS-03-44 **Anna Gál and Peter Bro Miltersen.** ***The Cell Probe Complexity of Succinct Data Structures*****. December 2003. 17 pp. An early version of this paper appeared in Baeten, Lenstra, Parrow and Woeginger, editors,** ***30th International Colloquium on Automata, Languages, and Programming*****, ICALP '03 Proceedings, LNCS 2719, 2003, pages 332–344.**

RS-03-43 **Mikkel Nygaard and Glynn Winskel.** ***Domain Theory for Concurrency*****. December 2003. 45 pp. To appear in a** ***Theoretical Computer Science*** **special issue on Domain Theory.**

RS-03-42 **Mikkel Nygaard and Glynn Winskel.** ***Full Abstraction for HOPLA*****. December 2003. 25 pp. Appears in Amadio and Lugiez, editors,** ***Concurrency Theory: 14th International Conference*****, CONCUR '03 Proceedings, LNCS 2761, 2003, pages 383–398.**

RS-03-41 **Malgorzata Biernacka, Dariusz Biernacki, and Olivier Danvy.** ***An Operational Foundation for Delimited Continuations*****. December 2003. 21 pp.**

RS-03-40 **Andrzej Filinski and Henning Korsholm Rohde.** ***A Denotational Account of Untyped Normalization by Evaluation*****. December 2003. 29 pp.**

RS-03-39 **Jörg Abendroth.** ***Applying π-Calculus to Practice: An Example of a Unified Security Mechanism*****. November 2003. 35 pp.**

RS-03-38 **Henning Böttger, Anders Møller, and Michael I. Schwartzbach.** ***Contracts for Cooperation between Web Service Programmers and HTML Designers*****. November 2003. 23 pp.**

RS-03-37 **Claude Crépeau, Paul Dumais, Dominic Mayers, and Louis Salvail.** ***Computational Collapse of Quantum State with Application to Oblivious Transfer*****. November 2003. 30 pp.**