On the Structure and the Number of Prime Implicants of $k$-CNF Formulas

Navid Talebanfard*
Department of Computer Science, Aarhus University
ntalebanfard@gmail.com

Abstract

We show that the maximum number of prime implicants over 2-CNF formulas on $n$ variables is between $3^{n/3}$ and $(1 + o(1))3^{n/3}$. For large $k$ we show that this number is at least $3^{(1-O(\log k/k))n}$, and any $k$-CNF formula in which each variable appears in a constant number of clauses has at most $3^{(1-\Omega(1/k))n}$ prime implicants.

1 Introduction

A prime implicant of a Boolean function $f$ is a maximal subcube contained in $f^{-1}(1)$. It is known that among Boolean functions on $n$ variables the maximum number of prime implicants is between $\Omega(\frac{2^n}{n})$ and $O(\frac{2^n}{\sqrt{n}})$ (see [1]). It is interesting to give finer bounds for restricted classes of functions. This problem has indeed been studied for DNF with a bounded number of terms. It is known that a DNF with $k$ terms has at most $2^k - 1$ prime implicants (see e.g. [2]).

In this paper we consider this problem for the class of $k$-CNF functions and prove that the number of prime implicants in a $k$-CNF formula on $n$

*The author acknowledges support from the Danish National Research Foundation and The National Science Foundation of China (under the grant 61061130540) for the Sino-Danish Center for the Theory of Interactive Computation, within which this work was performed. The author also acknowledges support from the Center for Research in Foundations of Electronic Markets (CFEM), supported by the Danish Strategic Research Council.
variable in which every variable appears at most $r$ times is at most $3^{(1-\frac{1}{r})n}$. Indeed understanding the structure of the set of satisfying assignments of $k$-CNF formulas has been a crucial subject in computational complexity, in particular in developing $k$-SAT algorithms and bounded depth circuit lower bounds. Notable examples are the characterization of satisfying assignments of a 2-CNF which yields a polynomial time algorithm for 2-SAT (see e.g. [3]), and the Satisfiability Coding Lemma of Paturi, Pudlák and Zane [4]. In fact we obtain our main result by a generalization of this lemma.

To state our results, we need a few definitions as follows. A restriction on a set of variables $X$ is a mapping $\rho : X \to \{*, 0, 1\}$. We call a variable free if it is assigned $*$, and we call it fixed otherwise. The size of $\rho$ is defined to be the number of variables fixed to 0 or 1 and we denote it by $|\rho|$. A sub-restriction $\rho'$ of $\rho$ is one such that $\rho(x) = *$ implies $\rho'(x) = *$ and we denote this by $\rho' \sqsubseteq \rho$. We also define $\rho^{[x\rightarrow *]}$ to be the restriction obtained from $\rho$ by unspecifying $x$. For a function $f$, we define $f|_{\rho}$ to be the subfunction after setting values to the fixed variables according to $\rho$. An implicant of $f$, is any restriction $\rho$ such that $f_{\rho} \equiv 1$. Furthermore, if unspecifying any fixed variable does not yield the constant 1 function, we say that $\rho$ is a prime implicant of $f$. If a prime implicant leaves some variable free we call it partial. We denote the set of prime implicants of $F$ by $\Pi(F)$. A read-$r$ formula is one in which every variable appears at most $r$ times. The set of variables of a formula $F$ is represented by $\text{vbl}(F)$.

A $(d, k)$-CSP is set of constraints each on at most $k$ variables, where every variable can take a value from a set of size $d$. For example a $(2, k)$-CSP is just a $k$-CNF. We say that an assignment $\alpha$ is an isolated solution for a CSP $F$, if $\alpha$ satisfies $F$ and changing the value of any single variable, falsifies some constraint. In what follows we give a generalization of the Satisfiability Coding Lemma to bound the number of isolated solutions in a $(d, k)$-CSP.

We have two main results in this section. Using an old characterization of satisfying assignments of 2-CNFs, we give an essentially tight bound for the number of prime implicants of 2-CNFs. For $k$-CNFs in general with $k \geq 3$, we obtain a bound assuming that every variable appears in at most a bounded number of clauses.
2 \quad k = 2

In this section we prove sharp bounds for the number of prime implicants of 2-CNF formulas.

**Theorem 1.** For every \( n \), there exists a 2-CNF formula \( F \) on \( n \) variables with \( |\text{PI}(F)| \geq 3^n \).

**Proof.** Let \( n = 3m \) and consider the following formula on variable set \( \{x_1, \ldots, x_m, y_1, \ldots, y_m, z_1, \ldots, z_m\} \) suggested to us by Dominik Scheder:

\[
T(x, y, z) = \bigwedge_{i=1}^{m} (x_i \lor y_i) \land (y_i \lor z_i) \land (x_i \lor z_i).
\]

It is easy to see that every prime implicant of \( T \) and every \( 1 \leq i \leq m \) must set exactly two variables among \( x_i, y_i \) and \( z_i \) to 1. Therefore \( T(x, y, z) \) has \( 3^n \) prime implicants.

\( \square \)

**Theorem 2.** For any 2-CNF \( F \) on \( n \) variables, we have \( |\text{PI}(F)| \leq (1 + o(1))3^n \).

Let \( F \) be a 2-CNF on \( \{x_1, \ldots, x_n\} \) and let \( \rho \) be a prime implicant that fixes all the variables. We claim that \( \rho \) is an isolated satisfying assignment for \( F \), that is if we change the value of any single one of the variables, the formula evaluates to 0. To see this note that if changing the value of some variable \( x_i \) still satisfies \( F \), we can simply unspecify \( x_i \) and get a smaller restriction which yields the constant 1 function, contradicting the minimality of \( \rho \). We can now apply the Satisfiability Coding Lemma and bound the number of such prime implicants by \( 2^n \).

**Lemma 1** (The Satisfiability Coding Lemma [4]). Any \( k \)-CNF on \( n \) variables has at most \( 2(1 - \frac{1}{k})^n \) isolated satisfying assignments.

It thus remains only to bound the number of partial prime implicants. We need some terminology. We define the implication digraph of \( F \) which we denote by \( D(F) \) as follows. The vertex set consists of all literals \( x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n \). For every clause \( x \lor y \) in \( F \) we put two directed edge \( \overline{x} \to y \) and \( \overline{y} \to x \) in \( D(F) \). If there is a clause consisting of only one literal \( x \), we add the edge \( \overline{x} \to x \). It is folklore and very easy to see that one can characterize the set of satisfying assignments of 2-CNFs in terms of their implication digraphs.
Proposition 1. An assignment $\alpha$ satisfies a 2-CNF $F$ if and only if there is no edge in $D(F)$ going out of the set of true literals.

Assume without loss of generality that $F$ contains no clauses with only one literal, since the value of such literal is forced. We now give a similar characterization of partial prime implicants. For a restriction $\rho$ we partition the set of literals into three sets $A_\rho$, $B_\rho$ and $C_\rho$ containing false literals, true literals, and those that are free, respectively.

Proposition 2. A restriction $\rho$ is a partial prime implicant if and only if

1. there is no edge going out of $B_\rho$
2. there is no edge from $C_\rho$ to $A_\rho$
3. $C_\rho$ is a maximal independent set.

Proof. $\Rightarrow$: Assume that $\rho$ is a prime implicant. Assume for a contradiction that there is an edge going out of $B_\rho$, namely $u \rightarrow v$. If $v \in A_\rho$, by construction of $D(F)$ we have $\overline{u} \lor v$ as a clause in $F$. But $\rho$ leaves this clause unsatisfied which is a contradiction. If $v \in C_\rho$, since it is left free by $\rho$ we can set it to 0 and we will get a $1 \rightarrow 0$ edge, and hence a contradiction with the same argument. If there is an edge from $C_\rho$ to $A_\rho$, again since we are free to assign values on $C_\rho$ we can get a $1 \rightarrow 0$ edge and reach a contradiction. Furthermore, $C_\rho$ is an independent set, since otherwise there would be a clause which is left completely untouched by $\rho$, and its maximality follows from minimality of $\rho$.

$\Leftarrow$: We first show that $\rho$ satisfies the formula. Notice that all the clauses are hit by $\rho$, since $C_\rho$ is an independent set. To see that the formula is indeed satisfied, consider an arbitrary clause $u \lor v$. If both $u$ and $v$ are fixed by $\rho$, since there is no edge from a true literal to a false literal, the clause should evaluate to true. If $u$ is fixed but $v$ is left free, $u$ has to be set to true, since otherwise there will be an edge from $\overline{u}$ to $v$ contradicting the fact that there is no edge going out of $A_\rho$. To show that $\rho$ is a prime implicant, note that if we can unspecify some literal and still satisfy the formula, this new set of free variables should be an independent set, because otherwise some clause will be untouched and hence unsatisfied.

Claim 1. For any two distinct partial prime implicants $\rho$ and $\rho'$ we have $C_\rho \neq C_{\rho'}$. 

4
Proof. For any partial prime implicant $\rho$ since $C_\rho$ is a maximal independent set, we have $A_\rho = N^-(C_\rho)$ and $B_\rho = N^+(C_\rho)$, where for any $S$, $N^-(S)$ and $N^+(S)$ denote the set of in-neighbors and out-neighbors of $S$, respectively. Therefore if two partial prime implicants define the same $C$, they should also define the same $A$ and $B$, and hence they should be the same. \qed

We say that an independent set $S \subseteq D(F)$ is reflexive if for any literal $u$, $u \in S$ if and only if $\overline{u} \in S$. By Claim 1 for each partial prime implicant $\rho$, $C_\rho$ is a maximal reflexive independent set. It is thus enough to bound the number of maximal reflexive independent sets in $D(F)$. From $F$ we construct another graph $G(F)$ on vertices $x_1, \ldots, x_n$ as follows: we include an edge $(x_i, x_j)$ if and only if there is some clause in $F$ which includes literals on both $x_i$ and $x_j$. We claim that if $S$ is a maximal reflexive independent set in $D(F)$, the set of all variables $T$ appearing in $S$ also forms a maximal independent set in $G(F)$. Assume that there is an edge $(x_i, x_j)$ in $T$. By construction of $G(F)$ this implies that there is an edge between some literal on $x_i$ and some literal on $x_j$. But since both literals appear in $S$, this means that there is an edge in $S$ and hence a contradiction. To see that $T$ is maximal, assume for a contradiction that we can add a variable $x_k$ to $T$ without forming any edge. This means that there is no edge between any literal on any variable in $T$ with any literal on $x_k$. But this implies that we can add $x_k$ and $\overline{x_k}$ to $S$ and have an independent set, contradicting the maximality of $S$.

It thus remains to bound the number of maximal independent sets in $G(F)$, which is just a simple undirected graph on $n$ vertices. This is a very well-known problem for which sharp bounds are known.

**Theorem 3** (Moon, Moser [7]). In any simple graph on $n$ vertices, there are at most $3^\frac{n}{2}$ maximal independent sets.

The set of prime implicants consists of those that are partial and those that fix all the variables. Therefore the total of number of prime implicants is bounded by $3^\frac{n}{2} + 2^\frac{n}{2}$.

**3 $k \geq 3$**

In this section we study the number of prime implicants of $k$-CNF formulas when $k \geq 3$. We first prove the following lower bound.
**Theorem 4.** For large enough \( n \) and \( k \), there exists \( k \)-CNF formula \( F \) on \( n \) variables such that \(|\text{PI}(F)| \geq 3^{(1-O\left(\frac{\log k}{k}\right))n}\) prime implicants.

**Proof.** We follow the construction of Chandra and Markowsky [1]. We divide the set of \( n \) variables in \( n/k \) parts, each of size \( k \). On each of these parts, we represent the Chandra-Markowsky function as a \( k \)-CNF, that is the disjunction of all conjunctions of \( 2k/3 \) variables, exactly \( k/3 \) of which are negated. Formula \( F \) would then be obtained by conjuncting all these function together. In [1] it was shown that each block has at least \( \Omega(3^k/k^k) \) prime implicants. It is easy to see that prime implicants of \( F \) are obtained by concatenating prime implicants of the blocks. Therefore the total number of prime implicants is at least \( \Omega((3^k/k^k)^{n/k}) = 3^{(1-O(\log k/k))n} \). \( \square \)

Generalizing the Satisfiability Coding Lemma [4], we give a scheme to encode prime implicants of \( k \)-CNF formulas. Unfortunately our argument goes through only as long as the formula is read-\( O(1) \).

**Theorem 5.** Let \( F \) be an \( r \)-read \( k \)-CNF on \( n \) variables. We have \(|\text{PI}(F)| \leq 3^{(1-\frac{1}{r})n}\).

**Proof.** Given a \( k \)-CNF \( F \) on \( x_1, \ldots, x_n \), we construct a CSP \( F' \) on \( x'_1, \ldots, x'_n \) as follows. We set \( \Sigma = \{0, 1, \ast\} \). Each assignment \( \alpha \) on \( F' \) naturally induces a restriction \( \sigma_\alpha \) on \( F \) as follows: for all \( 1 \leq i \leq n \), \( \sigma_\alpha(x_i) = \alpha(x'_i) \). We want to add constraints to \( F' \) so that \( F'(\alpha) = 1 \) if and only if \( \sigma_\alpha \) is a prime implicant of \( F \). We claim it is possible to define such \( F' \) by a \((3, rk)\)-CSP. We can define \( F' \) as follows

\[
F'(\alpha) = 1 \iff F|_{\sigma_\alpha} \equiv 1 \land \forall \sigma' (\sigma' \sqsubseteq \sigma_\alpha, |\sigma'| = |\sigma_\alpha| - 1 \Rightarrow F|_{\sigma'} \neq 1).
\]

The first conjunct can easily be expressed by a \((3, k)\)-CSP. Since

\[
F|_{\sigma_\alpha} \equiv 1 \iff \forall C \in F, C|_{\sigma_\alpha} \equiv 1,
\]

for each clause \( C \in F \) we can introduce a constraint \( C' \) in \( F' \) such that \( C'(\alpha) = 1 \) if and only if \( C|_{\sigma_\alpha} \equiv 1 \). For the second conjunct note that if \( \sigma' \) is obtained from \( \sigma_\alpha \) by unspecifying a variable \( x_i \), \( F|_{\sigma'} \neq 1 \) can be written as \( \bigvee_{C \ni x_i} C|_{\sigma'} \neq 1 \). But as \( x_i \) appears in at most \( r \) clauses each of size at most \( k \), the total number of variables appearing in this expression is at most \( rk \).
and thus we can represent it by a constraint on \( r k \) variables. However we do not need this constraint for \( n x_j \) which is not fixed by \( \sigma_\alpha \).

\[
D_i := (\alpha(x'_i) = \ast) \lor \bigvee_{C \ni x_i} C|_{\sigma_\alpha[x_i \rightarrow \ast]} \not\equiv 1
\]

Thus

\[
F'(\alpha) = (\land_{C \in FC'} C') \land (\land_{i=1}^n D_i)
\]

**Lemma 2.** All satisfying assignments of \( F' \) are isolated.

*Proof.* The construction guarantees that \( F'(\alpha) = 1 \) if and only if \( \sigma_\alpha \) is a prime implicant of \( F \). Let \( \alpha \) be such an assignment. We show that if we change the value of any single variable, the restriction induced by the resulting assignment is not a prime implicant. Let \( x'_i \) be any variable in \( F' \). Assume \( \alpha(x'_i) = \ast \). If we fix \( x'_i \) to either 0 or 1, this will correspond to a subcube strictly contained in the one defined by \( \sigma_\alpha \), and hence contradicting the maximality of \( \sigma_\alpha \). If on the other hand \( \alpha(x'_i) = 0 \) or 1, then changing this value to \( \ast \) corresponds to inducing a subcube, strictly containing the one defined by \( \alpha \), again contradicting maximality. \( \square \)

It thus remains to bound the number of isolated solutions of CSPs.

**Lemma 3.** In any \((d, k)\)-CSP on \( n \) variables, the number of isolated solutions is at most \( 3^{(1-\frac{1}{2})^n} \).

*Proof.* Given \( F \), a \((d, k)\)-CSP on \( z_1, \ldots, z_n \) that can take values in \( \Sigma \), and an isolated assignment \( \alpha \) we give an encoding of \( \alpha \) with respect a permutation \( \pi \) of the variables.

\begin{verbatim}
encode(\alpha, \pi, F)
1: \beta := the empty string
2: for z_i \in vbl(F) in the order given by \pi do
3:    if F \not\equiv z_i = \alpha_i then
4:        append \alpha_i to \beta
5:    end if
6:    F := F[z_i \rightarrow \alpha_i]
7: end for
8: return \beta
\end{verbatim}

7
Lemma 4. $E_\pi[|\text{encode}(\alpha, \pi, F)|] \leq (1 - \frac{1}{k})n$.

Proof. For each $z_i$, there exists a constraint $C_i$ such that $C_i(\alpha) = 1$, but if we change the value of $z_i$, $C$ is falsified. Therefore if $z_i$ appears after all other variables in $C_i$, the encoding procedure is forced to set the value of $z_i$ according to $\alpha_i$ and hence $z_i$ does not appear in the encoding. The expected number of such $z_i$s is at least $n/k$, and hence the lemma goes through. \qed

Lemma 5. Let $S \subseteq \Sigma^*$ be a prefix-free code with average code length at most $\ell$. We have $|S| \leq |\Sigma|^\ell$.

Proof. Let $d = |\Sigma|$. By Kraft’s inequality we have $\sum_{w \in S} d^{-|w|} \leq 1$. On the other hand we have $\ell \geq \sum_{w \in S} \frac{|w|}{|S|}$. We can thus write

\[
\ell - \log_d |S| \geq \sum_{w \in S} \frac{1}{|S|} (|w| - \log_d |S|) \\
= - \sum_{w \in S} \frac{1}{|S|} (\log_d d^{-|w|} + \log_d |S|) \\
= - \sum_{w \in S} \frac{1}{|S|} \log_d(|S|d^{-|w|}) \\
\geq - \log_d(\sum_{w \in S} d^{-|w|}) \\
\geq 0.
\]

\qed

Let $S$ to be the set of isolated solutions of $F$. Let $S_\pi$ be the set of encodings of elements of $S$ under $\pi$. By Lemma 4 it follows that there exists some $\pi^*$ such that the average code length in $S_{\pi^*}$ is at most $(1 - \frac{1}{k})n$. Then by Lemma 5 we can bound $|S_{\pi^*}| \leq |\Sigma|^{(1 - \frac{1}{k})n}$. But $|S| = |S_{\pi^*}|$ and we are done. \qed

All satisfying assignments of $F'$ are isolated and since $F'$ is a $(3, rk)$-CSP, we can use Lemma 4 and bound the number of prime implicants of $F$ by $3^{(1 - \frac{1}{rk})n}$, proving Theorem 5. \qed
4 Acknowledgements

I am grateful to Dominik Scheder and Stefan Schneider for useful discussions and to György Turán for correspondence. I would also like to thank Kristoffer Arnsfelt Hansen for pointing out an error in an earlier draft.

References

[1] A. K. Chandra, G. Markowsky, On the number of prime implicants, Discrete Mathematics 24 (1) (1978) 7–11.

[2] R. H. Sloan, B. Szörényi, G. Turán, On $k$-term DNF with the largest number of prime implicants, SIAM J. Discrete Math. 21 (4) (2008) 987–998.

[3] B. Aspvall, M. F. Plass, R. E. Tarjan, A linear-time algorithm for testing the truth of certain quantified boolean formulas, Inf. Process. Lett. 8 (3) (1979) 121–123.

[4] R. Paturi, P. Pudlák, F. Zane, Satisfiability coding lemma, in: FOCS, 1997, pp. 566–574.

[5] R. Paturi, P. Pudlák, M. E. Saks, F. Zane, An improved exponential-time algorithm for $k$-SAT, J. ACM 52 (3) (2005) 337–364.

[6] P. B. Miltersen, J. Radhakrishnan, I. Wegener, On converting CNF to DNF, Theor. Comput. Sci. 347 (1-2) (2005) 325–335.

[7] J. Moon, L. Moser, On cliques in graphs, Israel Journal of Mathematics 3 (1) (1965) 23–28.