Families of unsatisfiable $k$-CNF formulas with few occurrences per variable

Shlomo Hoory  
Department of Computer Science  
University of Toronto  
Toronto, Ontario, Canada  
shlomoh@cs.toronto.edu

Stefan Szeider  
Department of Computer Science  
University of Durham  
Durham, England, UK  
stefan.szeider@durham.ac.uk

November 9, 2018

Abstract

$(k,s)$-SAT is the satisfiability problem restricted to instances where each clause has exactly $k$ literals and every variable occurs at most $s$ times. It is known that there exists a function $f$ such that for $s \leq f(k)$ all $(k,s)$-SAT instances are satisfiable, but $(k,f(k)+1)$-SAT is already NP-complete ($k \geq 3$). The best known lower and upper bounds on $f(k)$ are $\Omega(2^{k}/k)$ and $O(2^{k}/k^{\alpha})$, where $\alpha = \log_{4} 3 - 1 \approx 0.26$. We prove that $f(k) = O(2^{k} \cdot \log k/k)$, which is tight up to a $\log k$ factor.

1 Introduction

We consider CNF formulas represented as sets of clauses, where each clause is a set of literals. A literal is either a variable or a negated variable. Let $k$, $s$ be fixed positive integers. We denote by $(k,s)$-CNF the set of formulas $F$ where every clause of $F$ has exactly $k$ literals and each variable occurs in at most $s$ clauses of $F$. We denote the sets of satisfiable and unsatisfiable formulas by SAT and UNSAT, respectively.

It was observed by Tovey [7] that all formulas in $(3,3)$-CNF are satisfiable, and that the satisfiability problem restricted to $(3,4)$-CNF is already NP-complete. This was generalized in Kratochvil, et al. [4] where it is shown that for every $k \geq 3$ there is some integer $s = f(k)$ such that

1. all formulas in $(k,s)$-CNF are satisfiable, and
2. $(k,s+1)$-SAT, the SAT problem restricted to $(k,s+1)$-CNF, is already NP-complete.

The function $f$ can be defined for $k \geq 1$ by the equation

$$f(k) := \max\{ s : (k,s)$-CNF $\cap$ UNSAT $= \emptyset \}.$$ 

Exact values of $f(k)$ are only known for $k \leq 4$. It is easy to verify that $f(1) = 1$ and $f(2) = 2$. It follows from [4] that $f(3) = 3$ and $f(k) \geq k$ in general. Also, by [6], we know that $f(4) = 4$. 

1
Upper and lower bounds for $f(k)$, $k = 5, \ldots, 9$, have been obtained in \cite{2, 6, 11, 13}. For larger values of $k$, the best known lower bound, a consequence of Lovász Local Lemma, is due to Kratochvıl et al. \cite{4}:

$$f(k) \geq \left\lfloor \frac{2^k}{ek} \right\rfloor.$$  

(1)

The best known upper bound, due to Savický and Sgall \cite{5}, is given by

$$f(k) \leq O\left(\frac{2^k}{k^\alpha}\right),$$

(2)

where $\alpha = \log_{3/4} \approx 0.26$.

In this paper we asymptotically improve upon \cite{2}, and show

$$f(k) = O\left(\frac{2^k \log k}{k}\right).$$

(3)

Our result reduces the gap between the upper and lower bounds to a $\log k$ factor. It turns out that the construction yielding the upper bound \cite{3} can be generalized. We present a class of $k$-CNF formulas, that is amenable to an exhaustive search using dynamic programming. This enables us to calculate upper bounds on $f(k)$ for values up to $k = 20000$ improving upon the bounds provided by the constructions underlying \cite{2} and \cite{3}.

The remainder of the paper is organized as follows. In Section\cite{2} we start with a simple construction that already provides an $O(2^k \log^2 k/k)$ upper bound on $f(k)$. In Section\cite{3} we refine our construction and obtain the upper bound \cite{3}. In the last section we describe the more general construction and the results obtained using computerized search.

## 2 The first construction

We denote by $\mathcal{K}(x_1, \ldots, x_k)$ the complete unsatisfiable $k$-CNF formula on the variables $x_1, \ldots, x_k$. This formula consists of all $2^k$ possible clauses. Let $\mathcal{K}^-(x_1, \ldots, x_k) = \mathcal{K}(x_1, \ldots, x_k) \setminus \{x_1, \ldots, x_k\}$. The only satisfying assignment for $\mathcal{K}^-(x_1, \ldots, x_k)$ is the all-False-assignment. Also, for two CNF formulas $F_1$ and $F_2$ on disjoint sets of variables, define their product $F_1 \times F_2$ as

$$\{c_1 \cup c_2 : c_1 \in F_1 \text{ and } c_2 \in F_2\}.$$  

Note that the satisfying assignments for $F_1 \times F_2$ are assignments that satisfy $F_1$ or $F_2$.

In what follows, all logarithms are to the base of 2.

**Lemma 1.** $f(k) \leq 2^k \cdot \min_{1 \leq l \leq k} ((1 - 2^{-l})^{[k/l]} + 2^{-l})$.

**Proof.** We prove the lemma by constructing an unsatisfiable $(k, s)$-CNF formula $F$ where $s = 2^k \cdot ((1 - 2^{-l})^{[k/l]} + 2^{-l})$. Let $k, l$ be two integers such that $1 \leq l \leq k$, and let $u = \lfloor k/l \rfloor$ and $v = k - l \cdot u$. Define the formula $F$ as the union $F = F_0 \cup F_1 \cup \ldots \cup F_u$, where:

$$F_0 = \mathcal{K}(z_1, \ldots, z_u) \times \prod_{i=1}^{u} \mathcal{K}^-(x^{(i)}_1, \ldots, x^{(i)}_l),$$

$$F_i = \mathcal{K}(y^{(i)}_1, \ldots, y^{(i)}_{k-l}) \times \{x^{(i)}_1, \ldots, x^{(i)}_l\} \quad \text{for } i = 1, \ldots, u.$$  

2
Therefore, $F$ is a $k$-CNF formula with $n$ variables and $m$ clauses, where
\begin{align*}
n &= k + u \cdot (k - l) \leq l + k^2/l, \\
m &= 2^u \cdot (2^l - 1)^u + u \cdot 2^{k-l} = 2^k \cdot \left(1 - 2^{-l} \right)^{\lfloor k/l \rfloor} + \lfloor k/l \rfloor \cdot 2^{-l}. 
\end{align*}

(4) \hspace{1cm} (5)

To see that $F$ is unsatisfiable observe that any assignment satisfying $F_0$ must set all the variables $x_1^{(i)}, \ldots, x_l^{(i)}$ to False for some $i$. On the other hand, any satisfying assignment to $F_i$ must set at least one of the variables $x_1^{(i)}, \ldots, x_l^{(i)}$ to True.

To bound the number of occurrences of a variable note that the variables $|F_i|$, and $|F_0| + |F_i|$ times, respectively. Since $|F_0| = 2^u \cdot (2^l - 1)^u = 2^k \cdot (1 - 2^{-l})^{\lfloor k/l \rfloor}$ and $|F_i| = 2^{k-l}$, we get the required result.

\[ \square \]

For $k \geq 4$, let $l$ be the largest integer satisfying $2^l \leq k \cdot \log e / \log^2 k$. If follows that
\[ (1 - 2^{-l})^{\lfloor k/l \rfloor} \leq \exp(-2^{-l} \cdot \lfloor k/l \rfloor) \leq \exp(- \frac{\log^2 k}{k \log e} \cdot \left(\frac{k}{l} + 1\right)) \]
\[ \leq \frac{1}{\sqrt{e}} \cdot \exp(- \frac{\log^2 k}{l \log e}) \leq \frac{1}{\sqrt{e}} \cdot \exp(- \frac{\log k}{\log e}) = \frac{1}{k \sqrt{e}}, \]

where the last inequality follows from the fact that $l \leq \log k$ for $k \geq 4$. Therefore, by Lemma [1] there exists an unsatisfiable $k$-CNF formula $F$ where the number of occurrences of variables is bounded by
\[ 2^k \cdot \left(\frac{1}{k \sqrt{e}} + \frac{2 \log^2 k}{k \log e}\right). \]

It may be of interest that by [4] and [5], the number of clauses in $F$ is $O(2^k \cdot \log k)$ and the number of variables is $O(k^2 / \log k)$.

**Corollary 2.** $f(k) = O(2^k \cdot \log^2 k / k)$.

\section{A better upper bound}

To simplify the subsequent discussion, let us fix a value of $k$. We will only be concerned with CNF formulas $F$ that have clauses of size at most $k$. We call a clause of size less that $k$ an incomplete clause and denote $F' = \{c \in F : |c| < k\}$. A clause of size $k$ is a complete clause, and we denote $F'' = \{c \in F : |c| = k\}$.

**Lemma 3.** $f(k) \leq \min\{2^{k-l+1} : l \in \{0, \ldots, k\} \text{ and } l \cdot 2^l \leq \log e \cdot (k - 2l)\}$.

**Proof.** Let $l$ be in $\{0, \ldots, k\}$, satisfying $l \cdot 2^l \leq \log e \cdot (k - 2l)$, and set $s = 2^{k-l+1}$. We will define a sequence of CNF formulas, $F_0, \ldots, F_l$. We require that (i) $F_j$ is unsatisfiable, (ii) $F'_j$ is a $(k - l + j)$-CNF formula, (iii) $|F''_j| \leq 2^{k-l}$, and that (iv) the maximal number of occurrences of a
variable in \( F_j \) is bounded by \( s \). It follows that \( F_i \) is an unsatisfiable \((k, s)\)-CNF formula, implying the claimed upper bound.

Set \( k_j = k - l + j \) and \( u_j = \lfloor (k - l + j)/(l - j + 1) \rfloor \). We proceed by induction on \( j \). For \( j = 0 \), we define \( F_0 = \mathcal{K}(x_1, \ldots, x_{k-1}) \). It can be easily verified that \( F_0 \) satisfies the above four requirements. For \( j > 0 \), assume a formula \( F_{j-1} \) on the variables \( y_1, \ldots, y_n \), satisfying the requirements. We define the formula \( F_j = \bigcup_{i=0}^{u_j} F_{j,i} \) as follows:

\[
F_{j,0} = \mathcal{K}(z_1, \ldots, z_{k_j-u_j}(l-j+1)) \times \prod_{i=1}^{u_j} \mathcal{K}^-(x_1^{(i)}, \ldots, x_{l-j+1}^{(i)}),
\]

\[
F_{j,i} = F'_{j-1}(y_1^{(i)}, \ldots, y_n^{(i)}) \times \{ \{ x_1^{(i)}, \ldots, x_{l-j+1}^{(i)} \} \cup F''_{j-1}(y_1^{(i)}, \ldots, y_n^{(i)}) \} \quad \text{for } i = 1, \ldots, u_j.
\]

It is easy to check that \( F'_j \) is a \((k - l + j)\)-CNF formula. To see that \( F_j \) is unsatisfiable, observe that any assignment satisfying \( F_{j,0} \), must set all the variables \( x_1^{(i)}, \ldots, x_{l-j+1}^{(i)} \) to False for some \( i \). On the other hand, for any satisfying assignment to \( F_{j,i} \), at least one of the variables \( x_1^{(i)}, \ldots, x_{l-j+1}^{(i)} \) must be set to True.

Let us consider the number of occurrences of a variable in \( F_j \). Consider first the \( y \)-variables. These variables occur only in the \( u_j \) duplicates of \( F_{j-1} \) and therefore occur the same number of times as in \( F_{j-1} \), which is bounded by \( s \) by induction. The number of occurrences of an \( x \)- or \( z \)-variable is \(|F'_{j-1}| + |F_{j,0}| \) or \(|F_{j,0}| \) respectively. By induction, \(|F'_{j-1}| \leq 2^{k-l} \). Also,

\[
|F_{j,0}| = |F_{j,0}| = 2^{k_j-u_j(l-j+1)} \cdot (2^{l-j+1}-1)^{u_j} = 2^{k_j} \cdot (1 - 2^{l-j-1})^{u_j} \\
\leq 2^{k-j} \cdot \exp(-2^{l-j-1} \cdot u_j) \leq 2^{k-l+j} \cdot \exp(-2^{l-j-1} \cdot (k - 2l)/l).
\]

Taking logarithms, we get

\[
\log |F_{j,0}| \leq k - l + j - \log e \cdot 2^{-l+j-1} \cdot (k - 2l)/l \\
\leq k - l + j - 2^{l-1} \leq k - l.
\]

Therefore, \( F_i \) is an unsatisfiable \((k, s)\)-CNF formula for \( s = 2^{k-l+1} \), as long as

\[
l \cdot 2^l \leq \log e \cdot (k - 2l). \tag{8}
\]

Let \( l \) be the largest integer satisfying \( 2^l \leq \log e \cdot k/(2 \log k) \). Then (8) holds for \( k \geq 2 \) and therefore we get the following:

**Corollary 4.** \( f(k) \leq 2^k \cdot 8 \log e \cdot k/k \) for \( k \geq 2 \).

## 4 Even better upper bounds

One way to derive better upper bounds on \( f(k) \) is to generalize the construction of Section 3. To this end, we first define a special way to compose CNF formulas capturing the essence of that construction.
Definition 5. Let $F_1, F_2$ be unsatisfiable CNF formulas that have clauses of size at most $k$ such that $F_i'$ is a $k_i$-CNF formula for $i = 1, 2$. Also, assume that $k_1 \leq k_2 < k$. Then the formula $F_1 \circ F_2$ is defined as:

$$
\bigcup_{c \in \mathcal{K}^-} (F_1' \times c \cup F_{1,c}^{'}) \cup (F_2' \times \{x_1, \ldots, x_{k-k_2}\}) \cup F_2^{'},
$$

where the formulas $F_{1,c}$ are copies of $F_1$ on distinct sets of variables.

It is not difficult to verify the following:

Lemma 6. Let $F_1, F_2$ be formulas as above, where the number of occurrences of a variable is bounded by $s \geq (2^{k-k_2} - 1) \cdot |F_1'| + |F_2'|$ and let $G = F_1 \circ F_2$. Then $G$ is an unsatisfiable CNF formula where each variable occurs at most $s$ times. Furthermore, $G'$ is a $(k_1 + k - k_2)$-CNF formula, and $|G'| = (2^{k-k_2} - 1) \cdot |F_1'|$.

Given $k, s$, we ask whether we can obtain a $k$-CNF formula using the following derivation rules. We start with the unsatisfiable formula $\{\emptyset\}$ as an axiom (this formula consists of one empty clause). For a set of derivable formulas, one can apply one of the following rules:

1. If $F$ is a derived formula such that $s \geq 2 \cdot |F'|$, then we can derive $F' \times \{x\} \cup F''$, where $x$ is a new variable.

2. If $F_1, F_2$ are two derived formulas satisfying the conditions of Lemma 6 then we can derive the formula $F_1 \circ F_2$.

Note 7. One can sometimes replace $F_1 \circ F_2$ in the second rule by a more compact formula $F_1 \circ' F_2$ that avoids duplicating $F_1$. Namely, the formula $F_1' \times \mathcal{K}^- \times \{x_1, \ldots, x_{k-k_2}\} \cup F_2'' \times \{x_1, \ldots, x_{k-k_2}\} \cup F_2''$. Although this can never reduce the number of occurrences of variables, this modification reduces the number of clauses and variables. In the construction of Section 3 we always use $\circ'$ instead of $\circ$.

Since any $k$-CNF formula obtained using the above procedure is an unsatisfiable $(k, s)$-CNF, one can define $f_2(k)$ as the maximal value of $s$ such that no $k$-CNF formula can be obtained using the above procedure (clearly $f(k) \leq f_2(k)$). It turns out that the function $f_2(k)$ is appealing from an algorithmic point of view. Given a value for $s$, one can check if $f_2(k) \leq s$ using a simple dynamic programming algorithm. For all $l = 0, \ldots, k - 1$, the algorithm keeps as state the minimal size of $F'$ for a derivable unsatisfiable formula $F$ where $F'$ is an $l$-CNF formula. This approach yields an algorithm that works well in practice and we were able to calculate $f_2(k)$ for values up to $k = 20000$ to get the results depicted by the graph in Figure 11.

The computed numerical values of $f_2(k)$ seem to indicates that

$$
f_2(k) \cdot k/2^k = 0.5 \log(k) + o(\log(k))
$$

which is better than our upper bound by a constant factor of about 11. If (9) indeed holds, then a better analysis of the function $f_2$ may improve our upper bound by a constant factor. However, such an approach cannot improve upon the logarithmic gap left between the known upper and lower bounds on $f(k)$.
Figure 1: The bounds on $f(k) \cdot k/2^k$. (a) Lower bound of Kratochvíl et al. $1/e$. (b) Upper bound obtained in Section 3 of the present paper, $8 \log_e k$. (c) Upper bound $f_2(k) \cdot k/2^k$, calculated by a computer program. (d) The line $0.5 \log(k) + 0.23$.

References

[1] Piotr Berman, Marek Karpinski, and Alex D. Scott. Approximation hardness and satisfiability of bounded occurrence instances of SAT. Technical Report TR03-022, Electronic Colloquium on Computational Complexity (ECCC), 2003.

[2] Olivier Dubois. On the $r,s$-SAT satisfiability problem and a conjecture of Tovey. *Discrete Appl. Math.*, 26(1):51–60, 1990.

[3] S. Hoory and S. Szeider. Computing unsatisfiable $k$-SAT instances with few occurrences per variable. Submitted, 2004.

[4] Jan Kratochvíl, Petr Savický, and Zsolt Tuza. One more occurrence of variables make satisfiability jump from trivial to NP-complete. *Acta Informatica*, 30:397–403, 1993.

[5] P. Savický and Jiří Sgall. DNF tautologies with a limited number of occurrences of every variable. *Theoret. Comput. Sci.*, 238(1-2):495–498, 2000.

[6] J. Stříbrná. Between combinatorics and formal logic. Master’s thesis, Charles University, Prague, 1994.

[7] Craig A. Tovey. A simplified NP-complete satisfiability problem. *Discrete Appl. Math.*, 8(1):85–89, 1984.