A note on unsatisfiable k-CNF formulas with few occurrences per variable

Shlomo Hoory*
Department of Computer Science
University of British Columbia
Vancouver, Canada
shlomoh@cs.ubc.ca

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

January 26, 2006

Abstract

The (k, s)-SAT problem 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 ($k \geq 3$). We prove that $f(k) = O(2^k \cdot \log k/k)$, improving upon the best know upper bound $O(2^k/k^\alpha)$, where $\alpha = \log_3 4 - 1 \approx 0.26$. The new upper bound is tight up to a $\log k$ factor with the best known lower bound $\Omega(2^k/k)$.

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 distinct literals and each variable occurs in *at most* s clauses of F. We denote the set of satisfiable formulas by SAT.

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 Kratochvíl, 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. the satisfiability problem restricted to formulas in $(k, s + 1)$-CNF is already NP-complete.

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

$$f(k) := \max\{\, s : (k, s)\text{-CNF} \subseteq \text{SAT} \,\}.$$

*Research is supported in part by an NSERC grant and a PIMS postdoctoral fellowship.

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 [7] that $f(3) = 3$ and $f(k) \geq k$ in general. Also, by [6], we know that $f(4) = 4$.

Upper and lower bounds for $f(k)$, $k = 5, \ldots, 9$, have been obtained in [2, 6, 1, 3]. For larger values of k, the best known lower bound, a consequence of Lovász Local Lemma, is due to Kratochvíl et al. [4]:

$$f(k) \geq \left\lfloor \frac{2^k}{ek} \right\rfloor . \tag{1}$$

Prior to this work, the best known upper bound has been by Savický and Sgall [5]. They constructed a family of unsatisfiable k-CNF formulas with 2^k clauses and small number of occurrences per variable. Their construction yields:

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

where $\alpha = \log_3 4 - 1 \approx 0.26$.

In this paper we asymptotically improve upon (2) and show

$$f(k) = O\left(\frac{2^k \log k}{k}\right). \tag{3}$$

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 (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 (2) and (3).

The remainder of the paper is organized as follows. In Section 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 3 we refine our construction and obtain the upper bound (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, their product $F_1 \times F_2$ is defined 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, log and ln denote logarithms to the base of 2 and e, respectively.

Lemma 1. $f(k) < 2^k \cdot \min_{1 \leq l \leq k} \left((1 - 2^{-l})^{\lfloor k/l \rfloor} + 2^{-l}\right)$.

Proof. We prove the lemma by constructing, for every l, an unsatisfiable (k, s)-CNF formula F where $s = 2^k \cdot ((1 - 2^{-l})^{\lfloor k/l \rfloor} + 2^{-l})$. 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:

$$\begin{aligned} F_0 &= \mathcal{K}(z_1, \ldots, z_v) \times \prod_{i=1}^{u} \mathcal{K}^-(x_1^{(i)}, \ldots, x_l^{(i)}), \\ F_i &= \mathcal{K}(y_1^{(i)}, \ldots, y_{k-l}^{(i)}) \times \{\{x_1^{(i)}, \ldots, x_l^{(i)}\}\} \quad \text{for } i = 1, \ldots, u. \end{aligned}$$

Therefore, F is a k-CNF formula with n variables and m clauses, where

$$n = k + u \cdot (k - l) \leq k^2/l, \tag{4}$$

$$m = 2^v \cdot (2^l - 1)^u + u \cdot 2^{k-l} = 2^k \cdot \left((1 - 2^{-l})^{\lfloor k/l \rfloor} + \lfloor k/l \rfloor \cdot 2^{-l}\right). \tag{5}$$

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 $z_j, y_j^{(i)}$, and $x_j^{(i)}$ occur $|F_0|, |F_i|$, and $|F_0| + |F_i|$ times, respectively. Since $|F_0| = 2^v \cdot (2^l - 1)^u = 2^k \cdot (1 - 2^{-l})^{\lfloor k/l \rfloor}$ 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

$$\begin{aligned} (1 - 2^{-l})^{\lfloor k/l \rfloor} \leq \exp(-2^{-l} \cdot \lfloor k/l \rfloor) &\leq \exp\left(-\frac{\log^2 k}{k \log e} \cdot (\frac{k}{l} - 1)\right) \\ &\leq e \cdot \exp\left(-\frac{\log^2 k}{l \log e}\right) \leq e \cdot \exp\left(-\frac{\log k}{\log e}\right) = \frac{e}{k}, \end{aligned}$$

where the last two inequalities follow from the fact that for $k \geq 4$ we have $\log^2 k < k \log e$ and $l \leq \log k$. Therefore, by Lemma 1 there exists an unsatisfiable k-CNF formula F where the number of occurrences of variables is bounded by

$$2^k \cdot \left(\frac{e}{k} + \frac{2 \log^2 k}{k \log e}\right).$$

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 $O(k^2/\log k)$. Thus, in comparison to the construction in [5], we pay for the better bound on k by a $O(\log k)$ factor in the number of clauses.

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

3 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. $f(k) < \min\{2^{k-l+1} : l \in \{0, \dots, k\} \text{ and } l \cdot 2^l \le \log e \cdot (k - 2l)\}$.

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

Set $d_j = k - l + j$ and $u_j = \lfloor (k - l + j)/(l - j + 1) \rfloor$. We proceed by induction on j. For $j = 0$, we define $F_0 = \mathcal{K}(x_1, \dots, x_{k-l})$. 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 $y_1, \dots, y_n$, satisfying the requirements. We define the formula $F_j = \bigcup_{i=0}^{u_j} F_{j,i}$ as follows:

$$F_{j,0} = \mathcal{K}(z_1, \dots, z_{d_j - u_j \cdot (l-j+1)}) \times \prod_{i=1}^{u_j} \mathcal{K}^-(x_1^{(i)}, \dots, x_{l-j+1}^{(i)}), \tag{6}$$

$$F_{j,i} = F'_{j-1}(y_1^{(i)}, \dots, y_n^{(i)}) \times \{\{x_1^{(i)}, \dots, x_{l-j+1}^{(i)}\}\} \cup F''_{j-1}(y_1^{(i)}, \dots, y_n^{(i)}) \quad \text{for } i = 1, \dots, u_j. \tag{7}$$

It is easy to verify 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 $x_1^{(i)}, \dots, x_{l-j+1}^{(i)}$ to False for some i. On the other hand, for any satisfying assignment to $F_{j,i}$, at least one of the variables $x_1^{(i)}, \dots, x_{l-j+1}^{(i)}$ 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, $|F'_{j-1}| \le 2^{k-l}$. Also,

$$\begin{aligned} |F'_j| &= |F_{j,0}| = 2^{d_j - u_j \cdot (l-j+1)} \cdot (2^{l-j+1} - 1)^{u_j} = 2^{d_j} \cdot (1 - 2^{-l+j-1})^{u_j} \\ &\le 2^{k-l+j} \cdot \exp(-2^{-l+j-1} \cdot u_j) \le 2^{k-l+j} \cdot \exp(-2^{-l+j-1} \cdot (k - 2l)/l). \end{aligned}$$

Taking logarithms, we get

$$\begin{aligned} \log |F_{j,0}| &\le k - l + j - \log e \cdot 2^{-l+j-1} \cdot (k - 2l)/l \\ &\le k - l + j - 2^{j-1} \le k - l. \end{aligned}$$

Therefore, F_j satisfies the induction hypothesis. For $j = l$ this implies that F_l is an unsatisfiable (k, s)-CNF formula for $s = 2^{k-l+1}$, as long as

$$l \cdot 2^l \le \log e \cdot (k - 2l). \tag{8}$$

$\square$

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

Corollary 4. $f(k) < 2^k \cdot 8 \ln k/k \quad \text{for } k \ge 2$.

4 Further generalization and experimental results

One way to derive better upper bounds on $f(k)$ is to generalize the constructions of Sections 2 and 3. To this end, we first define a special way to compose CNF formulas capturing the essence of these constructions.

Definition 5. *Let G_1, G_2 be unsatisfiable CNF formulas that have clauses of size at most k such that G'_i is a k_i-CNF formula for $i = 1, 2$. Also, assume that $k_1 \leq k_2 < k$. Then the formula $G_1 \circ G_2$ is defined as:*

$$\left(\bigcup_{c \in \mathcal{K}^-(x_1,\ldots,x_{k-k_2})} G'_{1,c} \times c \cup G''_{1,c} \right) \cup G'_2 \times \{\{x_1, \ldots, x_{k-k_2}\}\} \cup G''_2,$$

where the formulas $G_{1,c}$ are copies of G_1 on distinct sets of variables. We say that $G_1 \circ G_2$ is obtained by applying $\circ G_2$ to G_1, and we let $G_1 \circ_q G_2$ denote the formula obtained by applying $\circ G_2$ to G_1 q times.

It is not difficult to verify the following:

Lemma 6. *Let G_1, G_2 be formulas as above, where the number of occurrences of each variable is bounded by some number s satisfying $s \geq (2^{k-k_2} - 1) \cdot |G'_1| + |G'_2|$. Then $G = G_1 \circ G_2$ 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 $|G'| = (2^{k-k_2} - 1) \cdot |G'_1|$.*

Given k, s, we ask whether one 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 G is a derived formula such that $s \geq 2 \cdot |G'|$, then we can derive $G'_x \times \{\{x\}\} \cup G'_{\overline{x}} \times \{\{\overline{x}\}\} \cup G''_x \cup G''_{\overline{x}}$, where x is a new variable and G_x, $G_{\overline{x}}$ are two disjoint copies of G.
2. If G_1, G_2 are two derived formulas satisfying the conditions of Lemma 6, then we can derive the formula $G_1 \circ G_2$.

One can sometimes replace $G_1 \circ G_2$ in the second rule by a more compact formula $G_1 \circ' G_2$ that avoids duplicating G_1. Namely, the formula $G'_1 \times \mathcal{K}^-(x_1, \ldots, x_{k-k_2}) \cup G''_1 \cup G'_2 \times \{\{x_1, \ldots, x_{k-k_2}\}\} \cup G''_2$. Although this can never reduce the number of occurrences of variables, this modification reduces the number of clauses and variables. The constructions presented in Sections 2 and 3 are special cases of the above derivation rule. Indeed, $\mathcal{K}(x_1, \ldots, x_v)$ can be obtained by applying the first rule v times to $\{\emptyset\}$. The formula of Section 2 is just

$$F = \mathcal{K}(z_1, \ldots, z_v) \circ'_u \mathcal{K}(y_1, \ldots, y_{k-l}).$$

The formula of Section 3 is inductively obtained by

$$F_0 = \mathcal{K}(z_1, \ldots, z_{k-l}),$$
$$F_j = \mathcal{K}(z_1, \ldots, z_{d_j - u_j \cdot (l-j+1)}) \circ'_{u_j} F_{j-1} \qquad \text{for } j = 1, \ldots, l.$$

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 $f_2(k)$ is larger than s using a simple dynamic programming algorithm. The algorithm keeps an array $a_0, \ldots, a_k$, where eventually a_l contains the minimal size of F' for a derivable formula F such that F' is an l-CNF formula.

Initialize $a_0 = 1,\ \ a_1 = \cdots = a_k = \infty$
Repeat until no more changes are made to $a_1, \ldots, a_k$
  For $l = 0, \ldots, k-1$
    If $s \geq 2l$ then $a_{l+1} \leftarrow \min(\, 2a_l,\, a_{l+1}\,)$
  For $k_2 = 0, \ldots, k-1$
    For $k_1 = 0, \ldots, k_2$
      If $s \geq (2^{k-k_2} - 1) \cdot a_{k_1} + a_{k_2}$ then $a_{k_1+k-k_2} \leftarrow \min(\,(2^{k-k_2} - 1) \cdot a_{k_1},\, a_{k_1+k-k_2}\,)$
If $a_k < \infty$ then output "$f_2(k) \leq s$" else output "$f_2(k) > s$"

This algorithm 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 1.

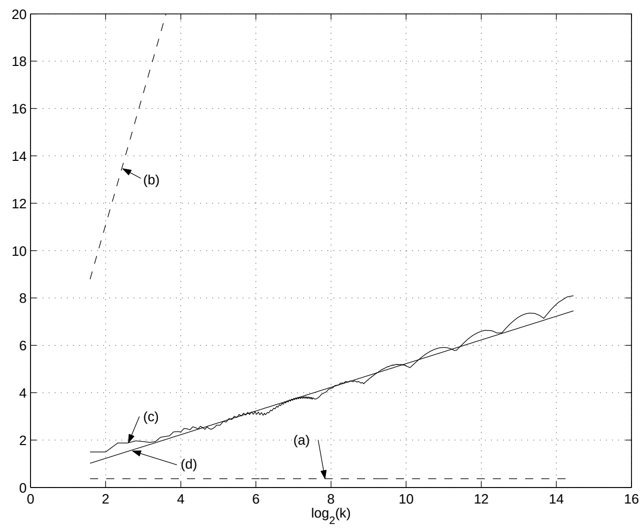

Figure 1: The bounds on $f(k) \cdot k/2^k$. (a) Lower bound of Kratochvíl et al. [4], $1/e$. (b) Upper bound (3) obtained in Section 3 of the present paper, $8 \ln k$. (c) Upper bound $f_2(k) \cdot k/2^k$, calculated by a computer program. (d) The line $0.5 \log(k) + 0.23$.

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

$$f_2(k) \cdot k/2^k = 0.5 \log(k) + o(\log(k)) \tag{9}$$

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 $f(k)$.

References

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

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

[3] S. Hoory and S. Szeider. Computing unsatisfiable k-SAT instances with few occurrences per variable. *Theoret. Comput. Sci.*, 337(1-3):347–359, 2005.

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

[5] P. Savický and J. 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] C. A. Tovey. A simplified NP-complete satisfiability problem. *Discr. Appl. Math.*, 8(1):85–89, 1984.