An Improved Exponential-time Algorithm for k-SAT

Ramamohan Paturi*, Pavel Pudlák†,
Michael E. Saks‡, and Francis Zane§

Abstract

We propose and analyze a simple new randomized algorithm, called ResolveSat, for finding satisfying assignments of Boolean formulas in conjunctive normal form. The algorithm consists of two stages: a preprocessing stage in which resolution is applied to enlarge the set of clauses of the formula, followed by a search stage that uses a simple randomized greedy procedure to look for a satisfying assignment. We show that, for each k, the running time of ResolveSat on a k–CNF formula is significantly better than 2^n, even in the worst case. In particular, we show that the algorithm finds a satisfying assignment of a general satisfiable 3–CNF in time $O(2^{.448n})$ with high probability; where the best previous algorithm [13] has running time $O(2^{.562n})$. We obtain a better upper bound of $2^{(2\ln 2-1)n+o(n)} = O(2^{0.387n})$ for 3-CNF that have exactly one satisfying assignment (unique k-SAT). For each k, the bounds for general k-CNF are the best currently known for the worst-case complexity of finding a satisfying solution for k-SAT. As in [9], the ideas used to analyze the algorithm can be applied to obtain lower bounds on circuit size. Here, we exhibit a function f such that any depth–3 AND-OR circuit with bottom fan–in bounded by k requires $\Omega(2^{c_k n/k})$ gates (with $c_k > 1$). This is the first such lower bound with $c_k > 1$. In addition, we show that any depth–3 AND-OR circuit computing f requires at least $2^{1.282\sqrt{n}}$ gates.

*University of California, San Diego; This research is supported by NSF grant CCR-9734911 from Theory of Computing Program

†Mathematical Institute ČSAV, Žitna 25, Praha 1, Czech Republic. Supported by grant no. A1019901 of the Academy of Sciences of the Czech Republic, and grant INT-9600919/ME 103(1997) under the cooperation of MŠMT, Czech Republic and NSF, USA

‡Research supported by NSF grant CCR-9700239. Department of Mathematics–Hill Center, Rutgers University, 110 Frelinghuysen Road, Piscataway, NJ 08854. saks@math.rutgers.edu.This work was done while on sabbatical at University of Washington.

§Bell Laboratories, Lucent Technologies

1 Introduction

1.1 k-CNF satisfiability

Given that the problem of deciding whether a given k–CNF formula is satisfiable is NP-complete for $k \geq 3$, it is natural to look for algorithms that improve significantly on the worst case running time of the naive exhaustive search algorithm, which is poly$(n)2^n$ for a formula on n variables. Monien and Speckenmeyer [8] gave the first real improvement by giving a simple algorithm whose running time is $2^{(1-\varepsilon_k)n}$, with $\varepsilon_k > 0$ for all k. Algorithms with increasingly better running times (larger values of ε_k) have been analyzed recently. Prior to this paper, the best known algorithm for $k = 3$ was a deterministic algorithm due to Kullmann with running time time $O(2^{.589n})$ [7]. For $k > 3$, the randomized algorithm in [9] was the fastest known, with a running time of $O(2^{(1-1/k)n})$.

In this paper, we present and analyze a new randomized algorithm, called ResolveSat, for finding satisfying assignments of k–CNF formula. For each value of k, the algorithm is faster than any currently known algorithm. For example, we show that, with high probability, the algorithm finds a satisfying assignment of any satisfiable 3-CNF formula in time $O(2^{.448n})$. We emphasize that, although the analysis is intricate, the algorithm itself is very simple. (Since the publication of a preliminary version of the present paper, Schöning [16] has obtained another randomized algorithm which is simpler and has a considerably simpler analysis than ours and has running time $(\frac{2k}{k+1})^n$, which is not as fast as ours but compares favorably to the algorithms that preceded ours.) There are also improved results on the complexity of k-SAT as a function of K, the number of clauses and L, the length of the input formula [6]. These bounds are of the form $2^{c_1 K}$ and $2^{c_2 L}$ for some constants $c_1, c_2 > 0$.

1.2 The Algorithm

First, we need a few definitions. For our purposes, a CNF boolean formula $F(x_1, x_2, \ldots, x_n)$ is viewed as both a boolean function and a set of clauses. We say that F is a k-CNF if all the clauses have size at most k. For a clause C, we write $vars(C)$ for the set of variables appearing in C. If $v \in vars(C)$, the *orientation* of v is positive if the literal v is in C and is negative if $\bar{v}$ is in C. Recall that if F is a CNF boolean formula on variables $(x_1, x_2, \ldots, x_n)$ and a is a partial assignment of the variables, the *restriction* of F by a is defined to be the formula $F' = F\lceil_a$ on the set of variables that are not set by a, obtained by treating each clause C of F as follows: if C is set to 1

by a then delete C, and otherwise replace C by the clause C' obtained by deleting any literals of C that are set to 0 by a. Finally, by a *unit clause*, we mean a clause that contains exactly one literal.

The following simple subroutine takes as input an arbitrary assignment and tries to modify it to a satisfying assignment of formula F by considering the variables one by one, in the order given by permutation π.

Procedure **Modify**(CNF formula $G(x_1, x_2, \ldots, x_n)$,
permutation π of $\{1, 2, \ldots, n\}$, assignment y)
$G_0 = G$.
for $i = 1$ **to** n
if G_{i-1} contains the unit clause $x_{\pi(i)}$
then $u_{\pi(i)} = 1$
else if G_{i-1} contains the unit clause $\bar{x}_{\pi(i)}$
then $u_{\pi(i)} = 0$
else $u_{\pi(i)} = y_{\pi(i)}$
$G_i = G_{i-1}\lceil_{x_{\pi(i)}=u_{\pi(i)}}$
end /* end for loop */
return u;

The algorithm **Search** is obtained by running **Modify**(G, π, y) on many pairs (π, y) where π is a random permutation and y is a random assignment.

Search(CNF-formula F, integer I)
repeat I times
π = uniformly random permutation of $1, \ldots, n$
y = uniformly random vector $\in \{0, 1\}^n$
u = **Modify**(F, π, y);
if u satisfies F
then output(u); **exit**;
end/* end repeat loop */
output('Unsatisfiable');

The algorithm **Search** was analyzed in [9]; we summarize the results in Theorem 1. The algorithm we investigate here is obtained by combining **Search** with a preprocessing step consisting of *bounded resolution*. We recall the definition of resolution. If C_1 and C_2 are two clauses we say that C_1 and C_2 *conflict* on variable v if one of them contains v and the other contains $\bar{v}$. C_1 and C_2 is a *resolvable pair* if they conflict on exactly one variable v. For

such a pair their *resolvent*, denoted $R(C_1, C_2)$ is the clause $C = D_1 \vee D_2$ where D_1 and D_2 are obtained by deleting v and $\bar{v}$ from C_1 and C_2. It is easy to see that any assignment satisfying C_1 and C_2 also satisfies C. Hence if F is a satisfiable CNF formula containing the resolvable pair C_1, C_2 then the formula $F' = F \wedge R(C_1, C_2)$ has the same satisfying assignments as F. We say that the resolvable pair C_1, C_2 is *s-bounded* if $|R(C_1, C_2)| \leq s$. The following subroutine extends a formula F to a formula F_s by applying as many steps of s-bounded resolution as possible.

Resolve(CNF Formula F, integer s)
  $F_s = F$.
  while F_s has an s-bounded resolvable pair C_1, C_2
      with $R(C_1, C_2) \notin F_s$
    $F_s = F_s \wedge R(C_1, C_2)$.
  return (F_s).

In this paper we analyze the following simple combination of **Resolve** and **Search**:

ResolveSat(CNF-formula F, integer s, positive integer I)
  $F_s =$ **Resolve**(F, s).
  Search(F_s, I).

1.3 Analysis of the algorithm Search

The algorithm **Search** was analyzed in [9]. It is easily seen that **Search**(F, I) runs in time $I|F|\text{poly}(n)$. It is also clear that **Search**(F, I) always answers "unsatisfiable" if F is unsatisfiable, and the problem of interest is to upper bound the error probability in the case that F is satisfiable. For a formula F and assignment z write $\tau(F, z)$ to be the probability, over random π and y, that **Modify**(F, π, y) returns the assignment z. Define $\tau(F)$ to be the sum of $\tau(F, z)$ over z that satisfy F, i.e., $\tau(F)$ is the probability that **Modify**(F, π, y) finds some satisfying assignment. Then for a satisfiable F the error probability of **Search**(F, I) is equal to $(1 - \tau(F))^{|I|} \leq e^{-|I|\tau(F)}$, which is at most e^{-n} provided that $|I| \geq n/\tau(F)$. The main result about **Search** in [9] is:

Theorem 1 *For any satisfiable k-CNF formula F on n variables, $\tau(F) \geq 2^{-(1-\frac{1}{k})n}$. Thus the algorithm* **Search** *with $I = n2^{(1-\frac{1}{k})n}$ has error probability $O(e^{-n})$ and runs in time $2^{(1-\frac{1}{k})n}\text{poly}(n)$.*

In particular, for $k = 3$ the running time is $2^{\frac{2}{3}n}\text{poly}(n)$ which is not as good as the $O(2^{.589n})$ algorithm of [7] mentioned in the introduction. For $k \geq 4$, Theorem 1 gave the best previously known upper bound.

The analysis of **Search** in [9] proceeds in two steps. First, Theorem 1 is proved for the special case of uniquely satisfiable F. Then an averaging argument is used to prove the result for all satisfiable F. The key idea in [9] is a simple relationship between the structure of the formula and the structure of the space of satisfying solutions expressed in terms of *critical clauses* and *isolated solutions*. We will review this idea in detail later in the paper.

It is also shown in [9] that the succinct description of *isolated solutions* used in proving Theorem 1 for uniquely satisfiable F can also be used to obtain better lower bounds on the size of depth 3 boolean circuits needed to compute certain functions.

1.4 Main results

In this paper we generalize the approach of [9] to analyze **ResolveSat**(F, s, I).

The running time of **ResolveSat**(F, s, I) can be bounded as follows. **Resolve**(F, s) adds at most $O(n^s)$ clauses to F and can be implemented easily in time $n^{2s}|F|\text{poly}(n)$ (this time bound can be improved, but this will not affect the asymptotics of our main results). **Search**(F_s, I) runs in time $I(|F| + n^s)\text{poly}(n)$. Hence the overall running time of **ResolveSat**(F, s, I) is crudely bounded from above by $I(|F| + n^{2s})\text{poly}(n)$. Provided that $s = o(n/\log n)$, $n^{2s} = 2^{o(n)}$ and we can bound this by $I|F|2^{o(n)}$. For our purposes, it will be sufficient to choose s to be a *slowly growing* function of n, by which we mean that $s(n)$ tends to infinity with n but is $o(\log n)$.

The error probability of **ResolveSat**(F, s, I), is just the error probability of **Search**(F_s, I) which, as noted above, is $(1 - \tau(F_s))^I$. So in extending the analysis of **Search** to **ResolveSat**, we want to lower bound $\tau(F_s)$ rather than $\tau(F)$. Trivially, $\tau(F_s) \geq \tau(F)$; the main contribution of this paper is to provide a way to quantify the advantage provided by the multiple critical clauses provided by the resolution preprocessing step. Following the approach of [9], we first upper bound $\tau(F_s)$ for *uniquely satisfiable formulas*. For $k \geq 1$, define:

$$\mu_k = \sum_{j=1}^{\infty} \frac{1}{j(j + \frac{1}{k-1})}.$$

Theorem 2 *Let $k \geq 3$, and let $s(n)$ be a slowly growing function. Then for any uniquely satisfiable k-CNF formula F on n variables,*

$$\tau(F_s) \geq 2^{-(1-\frac{\mu_k}{k-1})n-o(n)}$$

Hence, **ResolveSat**(F, s, I) *with* $I = 2^{(1-\frac{\mu_k}{k-1})n+o(n)}$ *has error probability* $o(1)$ *and running time* $2^{(1-\frac{\mu_k}{k-1})n+o(n)}$ *on any uniquely satisfiable k-CNF formula.*

It is not hard to show that $\mu_3 = 4 - 4\ln 2 > 1.226$, and hence for $k = 3$ the running time of the algorithm on any uniquely satisfiable 3-CNF formula is at most $2^{.387n+o(n)}$. It is easily seen that μ_k is an increasing function of k, and for large k, μ_k approaches $\sum_{j=1}^{\infty} \frac{1}{j^2} = (\frac{\pi^2}{6}) \approx 1.644$.

Next we consider general k-CNF formulas. In this case, we are able to extend the result for the uniquely satisfiable case, provided that $k \geq 5$:

Theorem 3 *Let $k \geq 5$, let $s(n)$ be a slowly growing function. Then for any satisfiable k-CNF formula F on n variables,*

$$\tau(F_s) \geq 2^{-(1-\frac{\mu_k}{k-1})n-o(n)}$$

Hence, **ResolveSat**(F, s, I) *with* $I = 2^{(1-\frac{\mu_k}{k-1})n+o(n)}$ *has error probability* $o(1)$ *and running time* $2^{(1-\frac{\mu_k}{k-1})n+o(n)}$ *on any satisfiable k-CNF formula, provided $k \geq 5$.*

For the case that $k = 3, 4$, we don't yet know whether the result for unique k-SAT can be generalized to general k-SAT, but we do get a substantial improvement over the best previous algorithms.

Theorem 4 *Let $s = s(n)$ be a slowly growing function. For any satisfiable n variable 3-CNF formula, $\tau(F_s) \geq 2^{-0.448n}$ and so* **ResolveSat**(F, s, I) *with $I = n2^{0.448n}$ has error probability $o(1)$ and running time $2^{0.448n+o(n)}$.*

Theorem 5 *Let $s = s(n)$ be a slowly growing function. For any satisfiable n variable 4-CNF formula, $\tau(F_s) \geq 2^{-0.581n}$ and so* **ResolveSat**(F, s, I) *with $I = n2^{0.581n}$ has error probability $o(1)$ and running time $2^{0.581n+o(n)}$.*

These results establish new upper bounds on the running time of algorithms for the k-SAT problem. The techniques we use to prove these upper bounds also provide some characterizations of the sets of satisfying assignments of k-CNF formulas; as in [9], using these characterizations, we prove lower bounds on the size of circuits computing explicit Boolean functions.

Our bounds pertain to the restricted circuit classes Σ^3, the set of unbounded fanin depth-3 circuits whose output gate is an OR gate, and Σ^3_k, the set of Σ^3 circuits with bottom fanin (ie, the fanin of gates closest to the inputs) at most k. Circuits of this form have been studied extensively. One motivation for studying such circuits is that sufficiently strong lower bounds on Σ^3 circuits would resolve other long-standing questions. In particular, using a technique of Valiant [17], it can be shown that a lower bound of $2^{n/\log\log n}$ on $\Sigma^3_{n^\epsilon}$ circuits would imply nonlinear lower bounds on the size of constant fanin, logarithmic depth circuits.

Previously, the strongest lower bounds on the size of such circuits for explicity functions were proven for the parity function: Σ^3 circuits of size $\Theta(n^{1/4}2^{\sqrt{n}})$ and Σ^3_k circuits of size $2^{n/k+o(n)}$ are necessary and sufficient to compute parity [9]. Here, we obtain slightly better bounds for the membership function for an error correcting code. Although these lower bounds are only slightly better, they have the significant property that $f\log_2 S/n > 1$ where f is the bottom fanin of the circuit and S is the size of the circuit. Earlier lower bound techniques [4, 12, 15, 5] do not seem to be able to give such lower bounds.

Recall that a set $E \subseteq \{0,1\}^n$ is an *error correcting code* with minimum distance d if $e_1, e_2 \in E, e_1 \neq e_2$ implies that the Hamming distance between e_1 and e_2 is at least d. Elements of E are referred to as *codewords*. Abusing notation slightly, let $E(x)$ be the function which is 1 iff $x \in E$.

Our lower bounds will apply to any code E with minimum distance $d > \log n$ and at least $2^{n-(\sqrt{n}/\log n)}$ codewords. These are fairly lenient choices of parameters, and constructions of codes with these properties are well known. For example, a BCH code with designed distance $\log n$ has at least $2^{n-\log^2 n}$ codewords. We prove:

Theorem 6 *Let E be an error correcting code of minimum distance $d > \log n$ and size at least $2^{n-(\sqrt{n}/\log n)}$ codewords. If C is a Σ^3_k circuit computing E, then C has at least $2^{\frac{\mu_k}{k-1}n - \frac{5n}{\log n}}$ gates.*

For example, we obtain a lower bound of $2^{0.612n}$ on the size of Σ^3_3 circuits computing E. In addition, we obtain

Theorem 7 *Let E be an error correcting code of minimum distance $d > \log n$ and size at least $2^{n-(\sqrt{n}/\log n)}$ codewords. If C is a Σ^3 circuit computing E, then C has at least $2^{\sqrt{\frac{\pi^2 n}{6}} - \frac{5\sqrt{n}}{\log n}}$ gates.*

Note that since $\sqrt{\frac{\pi^2}{6}} > 1.282$, this proves a lower bound of $2^{1.282\sqrt{n}}$ on the size of such circuits for n sufficiently large.

2 Lower bounding $\tau(G, z)$

Recall that $\tau(G, z)$ is the probability with respect to random π and y, that **Modify**(G, π, y) returns z. For our main results we want to lower bound $\tau(G)$, the sum of $\tau(G, z)$ over all satisfying assignments z of G. Here we present a lemma which lower bounds $\tau(G, z)$ in terms of the clause structure of G. This lemma formalizes one of the key ideas in [9].

Consider the run of **Modify**(G, π, y). Recall that each variable x_i is assigned so as to satisfy some unit clause, or is set to y_i. A variable whose assignment is determined by a unit clause is said to be *forced* (with respect to π and y). Let Forced(G, π, y) denote the set of variables that are forced. We first observe:

Proposition 1 *Let z be a satisfying assignment of G, and let π be a permutation of $[n]$ and y be any assignment to the variables. Then* **Modify**$(G, \pi, y) = z$ *if and only if y and z agree on all variables outside of* Forced(G, π, z).

Proof. If y agrees with z on all variables outside of Forced(G, π, z), then a simple induction on i shows that in the execution of **Modify**(G, π, y), the output bit $u_{\pi(i)}$ is z_i. If the variable $x_{\pi(i)}$ is not in Forced(G, π, z) then the $u_{\pi(i)}$ is set to $y_{\pi(i)} = z_{\pi(i)}$. If variable $x_{\pi(i)}$ is in Forced(G, π, z) then, by induction, for $j < i$, $u_{\pi(j)} = z_{\pi(j)}$ and so the clauses of G force output bit $u_{\pi(i)}$ to be $z_{\pi(i)}$.

Now suppose y disagrees with z on some variable outside of Forced(G, π, z) and let i be the first index such that $y_{\pi(i)} \neq z_{\pi(i)}$. Then it is easy to see that the output bit $u_{\pi(i)}$ will equal $y_{\pi(i)}$ and hence **Modify**$(G, \pi, y) \neq z$. ■

Thus, for fixed π, the number of y for which **Modify**(G, π, y) outputs z is $2^{|\text{Forced}(G,\pi,z)|}$, and we have:

Proposition 2 *Let z be a satisfying assignment of the formula G. Then*

$$\begin{aligned} \tau(G, z) &= \frac{1}{2^n n!} \sum_{\pi} 2^{|\text{Forced}(G,\pi,z)|} \\ &= 2^{-n} \mathbf{E}_\pi[2^{|\text{Forced}(G,\pi,z)|}]. \end{aligned}$$

where $\mathbf{E}_\pi$ denotes expectation with respect to random π.

By the concavity of the exponential function, the last expression can be lower bounded by $2^{-n+\mathbf{E}_\pi[|\text{Forced}(G,\pi,z)|]}$. We next find an alternative expression for $\mathbf{E}_\pi[|\text{Forced}(G, \pi, z)|]$. If v is a variable of formula G and z

is a satisfying assignment we say that a clause C is *critical* for (v, G, z) if C is in G, $v \in vars(C)$, and under the assignment z, the only true literal in C is the one corresponding to v. Suppose that C is critical for (v, G, z), and that π is a permutation such that v appears last among the variables of C. Then, in the run **Modify**(G, π, z), by the time v is assigned, all of the other literals in C have been falsified and so $v \in \text{Forced}(G, \pi, z)$ (conversely, if $v \in \text{Forced}(G, \pi, z)$ then v must appear last in some critical clause for (v, G, z)). Let $\text{Last}(v, G, z)$ be the set of permutations π of the variables such that for at least one critical clause C for (v, G, z), v appears last among all variables in $vars(C)$, and let $P(v, G, z)$ denote the probability that a random permutation π belongs to $\text{Last}(v, G, z)$, which is equivalent to the probability that $v \in \text{Forced}(G, \pi, z)$ for random π. By linearity of expectation, $\mathbf{E}_\pi[|\text{Forced}(G, \pi, z)|] = \sum_v P(v, G, z)$. Putting things together we have:

Lemma 3 *For any satisfying assignment z of the CNF formula G:*

$$\tau(G, z) \geq 2^{-n + \sum_v P(v, G, z)}.$$

In particular, if $P(v, G, z) \geq p$ for all variables v then $\tau(G, z) \geq 2^{-(1-p)n}$.

Hence, to lower bound $\tau(G, z)$ it suffices to lower bound $P(v, G, z)$. It is important to emphasize that while the function **Modify** depends on random π and y, the probability represented by $P(v, G, z)$ depends only on π and is independent of y.

3 Unique SAT

3.1 Overview

Using Lemma 3, the proof of Theorem 1 for the case of uniquely satisfiable F is nearly immediate. Indeed, let z be the unique satisfying assignment for the k-CNF F. Then for each variable v, F must contain at least one critical clause C_v for (v, F, z), otherwise the assignment obtained from z by complementing the value in position v is also a satisfying assignment. Since C_v has at most k variables, we conclude that for a random permutation π, v appears last in C_v with probability at least $1/k$, i.e., $P(v, F, z) \geq 1/k$ and so Lemma 3 implies $\tau(F) = \tau(F, z) \geq 2^{-(1-\frac{1}{k})n}$ as required. In fact, this argument actually proves a more general result: that if F is a k-CNF, and z is an isolated satisfying assignment, in the sense that no assignment differing from z in only one position satisfies F, then $\tau(F, z) \geq 2^{-(1-\frac{1}{k})n}$.

Note that the argument uses only the fact that each variable has at least one critical clause; if there are more critical clauses for each variable we could get a better lower bound. For example, suppose $z = 1^n$ is an isolated satisfying assignment for the formula F, and that F contains the two clauses $(x_1 \vee \bar{x}_2 \vee \bar{x}_3)$ and $(x_1 \vee \bar{x}_4 \vee \bar{x}_5)$ which are both critical for (x_1, F, z). In this case, the probability that a random permutation of the variables puts x_1 last in some critical clause is at least $7/15$, rather than $1/3$ obtained using only a single critical clause.

In general, even if F is uniquely satisfiable, F need contain only one critical clause per variable, so we can't hope for a general improvement of this kind. However, what we will show is that for appropriately chosen s, F_s contains many critical clauses for each variable. As an example, consider a formula F which contains the clauses $C_1 = (x_1 \vee \bar{x}_2 \vee \bar{x}_3)$ and $C_2 = (x_2 \vee \bar{x}_4 \vee \bar{x}_5)$, which are critical for x_1, x_2, respectively and for the assignment 1^n. Resolution on the variable x_2 will produce the clause $C = (x_1 \vee \bar{x}_3 \vee \bar{x}_4 \vee \bar{x}_5)$, which is also a critical clause for x_1. Thus F_4 contains both C_1 and C, and the probability that x_1 will be forced increases.

As in this example, our proof will show that adding clauses generated by resolution also adds new critical clauses for each variable. In our analysis, we will need to assume that z satisfies a stronger property than the isolation property used to show the existence of one critical clause for each variable. We show that if z is *d-isolated (in F)* for integer $d \geq 1$, that is, if there is no other satisfying assignment within Hamming distance d of z, then F_s contains many critical clauses for each variable. As a consequence, we will prove:

Theorem 8 *Let F be a k-CNF formula on n-variables and suppose that z is a d-isolated satisfying assignment of F. Then for $s \geq k^d$,*

$$\tau(F_s, z) \geq 2^{-(1-\frac{\mu_k}{k-1}+\epsilon_k^{(d)})n},$$

where for each k, μ_k is as defined before Theorem 2 and $\epsilon_k^{(d)} = \frac{3}{(d-1)(k-2)+2}$.

If z uniquely satisfies F, then z is d-isolated for any d, and so Theorem 2 follows.

To prove the above theorem we must somehow quantify the increase in the probability that a variable is forced that results from adding clauses produced by resolution. This is complicated by the fact that new clauses obtained in this manner may have different lengths, and the events [x_1 is forced because of C_i] are not independent. We will formulate a precise

statement about the structure of critical clauses for (v, F_s, z). This structure will be described in terms of a rooted tree with some of the nodes labeled by variables. We'll need some definitions. The *degree* of a node in a rooted tree is the number of its children. The *depth* of a node is its distance from the root. The *min-depth of tree* T is the minimum depth of any leaf. A subset A of nodes is a *cut* if it does not include the root, and every path from the root to a leaf includes a node of A. (This definition is not quite what one might expect, e.g., the set of all leaves is a cut by this definition). If A is a set of nodes, write $L(A)$ for the set of variables that appear as labels of nodes of A.

A rooted tree is said to be *admissible* with respect to a given set of boolean variables if it has the following properties:

- The root is labeled by a variable
- Each node in the tree is either labeled by a variable or unlabeled
- For any path P from the root to a leaf, no two nodes have the same label. In other words, if node a is an ancestor of node b and both are labeled, then they have different labels.

A tree is said to be a *critical clause tree* for variable v, formula G and satisfying assignment z if it is admissible and in addition satisfies

- The root label is v
- For any cut A of the tree, G has a critical clause $C(A)$ for (v, G, z) such that $vars(C(A)) \subseteq L(A) \cup \{v\}$.

We will show that if F is uniquely satisfiable, then for some appropriately large s, there is a "large" critical clause tree for (v, F_s, z). This critical clause tree represents multiple critical clauses for (v, F_s, z). This will enable us to derive a better lower bound on the probability that v is forced.

We prove two lemmas that together imply Theorem 8 and hence Theorem 2.

Lemma 4 *Let F be a k-CNF formula and z be a d-isolated satisfying assignment of F. If v is any variable then for any $s \geq k^d$, there exists a critical clause tree for (v, F_s, z) of min-depth d and maximum degree $k-1$.*

The second lemma asserts that the existence of a sufficiently deep critical clause tree for (v, G, z) of bounded degree implies a lower bound on $P(v, G, z)$.

Lemma 5 *Let G be a formula, z a satisfying assignment, and v a variable. If there is a critical clause tree for (v, G, z) of min-depth d and maximum degree $k-1$, then:*

$$P(v, G, z) \geq \frac{\mu_k}{k-1} - \epsilon_k^{(d)},$$

where for each k, $\epsilon_k^{(d)} = \frac{3}{(d-1)(k-2)+2}$.

Putting $G = F_s$ and combining these two lemmas with Lemma 3 immediately yields Theorem 8. So it remains to prove these two lemmas.

3.2 Existence of a deep, bounded-degree critical clause tree

In this subsection we prove Lemma 4. Fix a k-CNF formula F, and d-isolated assignment z; we will assume without loss of generality that $z = 1^n$. In this case, for each variable v, the critical clause of F for v consists of v and a set of negated variables.

To get some intuition for our construction, let's consider an example with $k = 3$. Suppose that for $j < n/2$, the critical clause for x_j in F is $C_j = x_j \vee \bar{x}_{2j} \vee \bar{x}_{2j+1}$. We build a critical clause tree for (x_1, F, z). We start with a root labeled x_1, and two children labeled by x_2 and x_3, representing the critical clause $C_1 = x_1 \vee \bar{x}_2 \vee \bar{x}_3$. Now, since $C_2 = x_2 \vee \bar{x}_4 \vee \bar{x}_5$ is a critical clause, we make x_4 and x_5 children of x_2, and since $C_3 = x_3 \vee \bar{x}_6 \vee \bar{x}_7$ is a critical clause, we make x_6 and x_7 children of x_3. Note that the four distinct minimal cuts of this tree correspond to the clause C_1, the resolvent of C_1 and C_2, the resolvent of C_1 and C_3 and the resolvent of C_1, C_2 and C_3. We can continue in this way growing a tree with up to $n/2 - 1$ internal nodes: at each step we grow the tree from a leaf labeled x_j by adding two children labeled by the variables that appear negatively in C_j.

This example provides useful intuition, but is not general. One important property of the example that made it work was that the sets of negated variables corresponding to the critical clauses used in the construction were pairwise disjoint. If instead the negated variables appearing in different C_j overlap then the resulting tree will satisfy all properties of the critical clause tree except the property that variables along any root to leaf path are distinct, and this property is crucial for proving Lemma 5. We need a more subtle construction to maintain this property.

We need one additional piece of notation: for a set of variables U, $z \oplus U$ denotes the assignment obtained from z by complementing the variables of U.

Let v be an arbitrary variable of F. We will construct a critical clause tree for (v, F_s, z). We "grow" the tree by the following process. Start with a tree T_0 consisting of one node labeled v. We construct a sequence of trees $T_1, T_2, \ldots$ as follows. Having constructed T_{i-1}, if all leaves have depth d stop. Otherwise, choose a leaf b_i of depth less than d, and let P_i be the set of nodes appearing on the path from b_i to the root (including b_i and the root). Since z is d-isolated, and $|P_i| \leq d$, $z \oplus L(P_i)$ does not satisfy F; choose a clause C_i that is not satisfied by $z \oplus L(P_i)$. For each variable w of $vars(C_i) - L(P_i)$, give b_i a child labeled w. If $vars(C_i) - L(P_i)$ is empty, give b an unlabeled child. Let N_i denote the set of nodes corresponding to the children of b_i.

Clearly the above procedure terminates with an admissible tree of min-depth d and maximum degree $k-1$. The total number of nodes in the final tree is bounded above by k^d, and hence any clause whose variables all appear as node labels in the tree has size at most s.

We will prove by induction on i that T_i is a critical clause tree for (v, F_s, z), i.e., that for any cut A, F_s contains a critical clause $C(A)$ for (v, F_s, z) whose variable set is contained in $L(A) \cup \{v\}$. The result is vacuously true for T_0, which has no cuts. For T_1, the tree has only one cut. T_1 is constructed from a clause C which is not satisfied by $z \oplus v$, i.e., a critical clause for v at z.

Now suppose $i \geq 2$ and the result holds for T_{i-1}. T_i consists of T_{i-1} together with the node set N_i. Let A be a cut of T_i, and let $A' = A - N_i$. Let the nodes of P_i be denoted $v = a_0, a_1, \ldots, a_t = b_i$. Note that for each $1 \leq j \leq t$, the set $A_j = A' \cup a_j$ is a cut of T_i and a cut of T_{i-1}. Hence, by the induction hypothesis, there is a critical clause $C(A_j)$ for (v, F_s, z) such that $vars(C(A_j)) \subseteq L(A_j)$. If for some $j \in [t]$, $L(A_j) \subseteq L(A')$ then we could choose $C(A)$ to be $C(A_j)$. Thus we may assume that $L(A_j) \not\subseteq L(A')$, which means that a_j is labeled by a variable r_j and $r_j \notin L(A')$. (We also define $r_0 = L(a_0)$, which is just v). Thus for each $j \in T$ we may write $C(A_j)$ as $B_j \vee \bar{r}_j \vee v$ where B_j is a clause consisting of negated variables from A' (and hence contains none of the variables r_h).

Now consider the clause C_i of F that is used to construct T_i from T_{i-1}. We can write C_i in the form $R \vee U$ where R is a clause consisting of negated variables from $L(N_i)$ and U is a clause consisting of positive variables from $r_0, \ldots, r_t$.

We now prove, by reverse induction on $j \in \{0, \ldots, t\}$ that there is a clause D_j in F_s of the form $D_j = R \vee S_j \vee U_j$ where R is defined in the last paragraph, S_j is a clause consisting of negations of variables from A' and U_j is a clause consisting of (positive occurrences of) of variables from

$\{r_0, \ldots, r_j\}$. D_0 must then be of the form $R \vee S_0 \vee U_0$ where U_0 is empty or $U_0 = v$. It can't be that U_0 is empty, since then D_0 would consist only of negated variables, and such a clause can not be in F_s since $z = 1^n$ satisfies F_s. Thus we may take $C(A)$ to be D_0.

To prove the existence of D_j, for the basis we take $D_t = C_i$, with S_t being the empty clause and $U_t = U$. For $j < t$, assuming we have constructed D_{j+1}, if r_{j+1} does not appear in U_{j+1} then take $D_j = D_{j+1}$. Otherwise the clause D_{j+1} can be resolved with $C(A_{j+1})$ to eliminate r_{j+1} and yield D_j. Since each of the D_j has size at most s, each is in F_s.

3.3 Lower Bounding $P(v, G, z)$

We now proceed to the proof of Lemma 5. We are given a critical clause tree for (v, G, z) of min-depth d and maximum degree $k-1$ and we want to deduce a lower bound on $P(v, G, z)$, the probability with respect to a random permutation π, that v appears last in some critical clause.

For the analysis, it will be useful to view the permutation π as a random variable on the following continuous probability space. A *placement* of the variables is a function α that maps each variable to $(0, 1)$. We take the set of placements as our sample space and consider the uniform distribution, i.e., the values $\alpha(u)$ are independent and uniformly distributed on $(0, 1)$. Given a placement α we define a permutation $\pi = \pi_\alpha$ obtained by ranking the variables according to their α values, breaking ties by some (any) arbitrary rule. Since α is one-to-one with probability 1, π is uniformly distributed over all permutations. Henceforth, all probabilities we compute are with respect to this probability space. An event is this probability space is then a (measurable) set of placements.

For an admissible tree T with root labeled by variable v, we define the event Cut_T to consist of all placements α such that for some cut A of T, $\alpha(w) < \alpha(v)$ for all $w \in L(A)$. We define $\mathrm{Cut}_T(r)$ for $r \in (0, 1)$ to consist of all α such that for some cut A of T, $\alpha(w) < r$ for all $w \in L(A)$. We define Q_T (resp. $Q_T(r)$) to be the probability Cut_T (resp. $\mathrm{Cut}_T(r)$) occurs. From the definitions we have:

Proposition 6 *If T is a critical clause tree for (v, G, z), then $P(v, G, z) \geq Q_T$.*

So to prove the lemma, it suffices to lower bound Q_T in the case that T is a min-depth d tree of maximum degree $k-1$.

Now it is easy to see that $Q_T(r)$ is just equal to the conditional probability of Cut_T given that $\alpha(v) = r$ (here we need to use the fact that in an admissible tree, no other node has the same label as the root). Hence:

$$Q_T = \int_0^1 Q_T(r)dr.$$

We say that T is trivial if it consists of one node; in this case $Q_T = Q_T(r) = 0$ for all r. Otherwise, for each child a of the root, the subtree $T(a)$ rooted at a is admissible if and only if a is labeled. We have the following recursive lower bound on $Q_T(r)$:

Lemma 7 *Let T be an admissible tree with more than one node with root labelled by v and $r \in (0,1)$. Let $T_1, T_2, \ldots, T_t$ be the admissible subtrees rooted at the labeled children of the root of T. Then:*

$$Q_T(r) \geq \prod_{i=1}^{t}(r + (1-r)Q_{T_i}(r)),$$

where an empty product is interpreted as 1.

Proof. If none of the children of the root of T is labeled, then T has a cut A such that $L(A) = \emptyset$ and so Cut_T occurs with probability 1. Otherwise, let v_i be the variable labeling the root of T_i. The event $\mathrm{Cut}_T(r)$ can be written as the intersection of the events $\{K_i : 1 \leq i \leq t\}$ where $K_i = \mathrm{Cut}_{T_i}(r) \vee (\alpha(v_i) < r)$.

By the admissibility of T_i, v_i does not appear elsewhere in T_i, so the events Cut_{T_i} and $\alpha(v_i) < r$ are independent. Thus $\mathbf{Prob}[K_i] = r + (1 - r)Q_{T_i}(r)$.

Finally we need that $\mathbf{Prob}[\wedge_i K_i] \geq \prod_i \mathbf{Prob}[K_i]$. This follows from standard correlation inequalities. For a placement α, let $W(\alpha, r)$ be the subset of variables w such that $\alpha(w) < r$. Then each event K_i depends only on $W(\alpha, r)$. Let $\mathcal{W}_i$ denote the set of all subsets U of variables such that $W(\alpha, r) = U$ implies K_i. From the definition of K_i, $\mathcal{W}_i$ is monotone, i.e., if $U \in \mathcal{W}_i$ then so is any superset of U. Finally, for $2 \leq i \leq t$, let $\mathcal{V}_i = \cap_{j<i}\mathcal{W}_j$. Note that the families $\mathcal{V}_i$, being intersections of monotone increasing families, are themselves monotone increasing.

Now, we make use of the following special case of of Theorem 3.2 from Chapter 6 of [1].

Lemma 8 *Let $\mathcal{A}$ and $\mathcal{B}$ be two monotone increasing families of subsets of $X = \{x_1, \ldots, x_n\}$. For any $0 \leq p \leq 1$, if $Y \subseteq X$ is chosen by selecting each x_i independently with probability p, then*

$$\Pr[Y \in \mathcal{A} \cap \mathcal{B}] \geq \Pr[Y \in \mathcal{A}] \Pr[Y \in \mathcal{B}]$$

For $2 \leq i \leq t$, we apply the above lemma with $p = r$, $A = \mathcal{V}_i$, and $B = \mathcal{W}_i$ to obtain:

$$\Pr[Y \in \mathcal{V}_{i+1}] \leq \Pr[Y \in \mathcal{V}_i] \Pr[Y \in \mathcal{W}_i]$$

and so by induction on i

$$\Pr[Y \in \mathcal{V}_i] \leq \prod_{j<i} \Pr[Y \in \mathcal{W}_j],$$

which implies that $\mathbf{Prob}[\wedge_{i\in[t]} K_i] \geq \prod_{i\in[t]} \mathbf{Prob}[K_i]$, as required. ■

We apply this lemma in the case that T is a tree of degree at most $k-1$ and min-depth at least d. We need some definitions. Fix $k \geq 3$ and $r \in (0,1)$.

- Let $f_k(x;r) = (r + (1-r)x)^{k-1}$,
- Define the sequence $(Q_k^{(d)}(r) : d \geq 0)$ by the recurrence $Q_k^{(0)}(r) = 0$ and $Q_k^{(d)}(r) = f_k(Q_k^{(d-1)}(r); r)$.
- Define $Q_k^{(d)} = \int_0^1 Q_k^{(d)}(r)dr$.

Lemma 7 together with induction on d yields:

Lemma 9 *If T is an admissible tree of degree at most $k-1$ and min-depth at least d then for all $r \in (0,1)$:*

$$Q_T(r) \geq Q_k^{(d)}(r).$$

Hence:

$$Q_T \geq Q_k^{(d)}.$$

We will prove:

Lemma 10 *Let $k \geq 3$. Then for each $d \geq 1$,*

$$Q_k^{(d)} \geq \frac{\mu_k}{k-1} - \frac{3}{(d-1)(k-2)+2}.$$

Proposition 6, and Lemmas 9 and 10 give $P(v, G, z) \geq Q_T \geq \frac{\mu_k}{k-1} - \frac{3}{(d-1)(k-2)+2}$, as required for Lemma 5. So it remains to prove this lemma.

3.4 Proof of Lemma 10

In order to lower bound $Q_k^{(d)}$, we will show that for each k and r, the sequence $(Q_k^{(d)}(r) : d \geq 0)$ converges and also give bounds on the rate of convergence. We need the following definitions.

- For $r \in [0,1]$ define $R_k(r)$ to be the smallest nonnegative x that satisfies $f_k(x;r) = x$. $R_k(r)$ is well defined and has range $[0,1]$, since for each r, $f_k(x;r) - x$ is a polynomial in x, having 1 as a root.
- For $t \in [0,1)$ define

$$S_k(t) = \frac{t^{\frac{1}{k-1}} - t}{1-t}.$$

We first establish:

Proposition 11 *Fix $k \geq 3$.*

- *For $r \in [\frac{k-2}{k-1}, 1]$, $R_k(r) = 1$.*
- *On the interval $[0, \frac{k-2}{k-1}]$, $R_k(r)$ is an increasing continuous map onto $[0,1]$, and its inverse is $S_k(t)$.*

Proof. Fix r and let $\epsilon = 1 - R_k(r)$, so $\epsilon \in [0,1]$. Substituting into $R_k(r) = f_k(R_k(r);r)$, yields $g(\epsilon) = 1$ where $g(\epsilon) = (1-(1-r)\epsilon)^{k-1} + \epsilon$. If $\epsilon > 0$, then $g(\epsilon) > 1$ for $r \in (\frac{k-2}{k-1}, 1]$, a contradiction. Thus for $r \in [\frac{k-2}{k-1}, 1]$, we have $\epsilon = 0$ and thus $R_k(r) = 1$.

Now, for fixed $0 < r < \frac{k-2}{k-1}$, $g(\epsilon; r)$ is a continuous function of ϵ. Since $g(0;r) - 1 = 0$, $g'(0;r) < 0$ and $g(1;r) - 1 > 0$, we conclude that for all $r < \frac{k-2}{k-1}$ there is a positive ϵ^* satisfying $g(\epsilon^*) = 0$, which implies that $R_k(r) < 1$.

Thus for $r \in [0, \frac{k-2}{k-1})$, $R_k(r) \in [0,1)$. Then the equation $f_k(R_k(r);r) = R_k(r)$ can be solved for r to obtain $r = S_k(R_k(r))$. By elementary calculus, $S_k(t)$ is a continuous increasing function of t mapping $[0,1)$ onto $[0, \frac{k-2}{k-1})$, from which it follows that its inverse $R_k(r)$ is a continuous increasing function mapping $[0, \frac{k-2}{k-1})$ to $[0,1)$. ■

Let $R_k = \int_0^1 R_k(r)\, dr$. Next we define the sequences $\{\Delta_k^{(d)}(r) : d \geq 0\}$ and $\{\Delta_k^{(d)} : d \geq 0\}$ by:

- $\Delta_k^{(d)}(r) = R_k(r) - Q_k^{(d)}(r)$,

- $\Delta_k^{(d)} = R_k - Q_k^{(d)}$.

Lemma 10 follows immediately from the next two lemmas.

Lemma 12 *Let $k \geq 3$. For all $d \geq 1$,*

$$0 \leq \Delta_k^{(d)} \leq \frac{3}{(d-1)(k-2)+2}$$

.

Lemma 13 *For each $k \geq 3$, $R_k = \frac{\mu_k}{k-1}$.*

Proof of Lemma 12: For simplicity of notation, we omit the subscript k. For each $r \in [0,1]$, $\Delta^{(d)}(r) \geq 0$ since $Q^{(d)}(r)$ is monotnic increasing in d and $R_k(r) = Q^{(\infty)}$. It then follows $\Delta^{(d)} = \int_0^1 \Delta^{(d)}(r) \geq 0$. To upper bound $\Delta^{(d)}(r)$ we first prove bounds on $R(r) - f(x;r)$ for $x \in [0, R(r)]$.

Proposition 14 *Let $r \in (0,1)$ and $k \geq 3$. Let $\gamma \in [0, R(r)]$. Then:*

$$0 \leq R(r) - f(R(r) - \gamma; r) \leq \gamma \frac{(k-1)(1-r)R(r)}{r + (1-r)R(r)}$$

Proof. The first inequality follows since for fixed $r \in (0,1)$, $f(x;r)$ is an increasing function of x and $f(R(r);r) = R(r)$. For the second inequality:

$$\begin{aligned}
R(r) - f(R(r)-\gamma;r) &= f(R(r);r) - f(R(r)-\gamma;r) \\
&= (r + (1-r)R(r))^{k-1} - (r + (1-r)(R(r)-\gamma))^{k-1} \\
&= \gamma(1-r)\sum_{j=0}^{k-2}(r + (1-r)R(r))^{k-2-j}(r + (1-r)(R(r)-\gamma))^j \\
&\leq \gamma(1-r)(r + (1-r)R(r))^{k-2})(k-1) \\
&= \gamma\frac{(1-r)R(r)}{r + (1-r)R(r)}(k-1).
\end{aligned}$$

■

It follows immediately from the above Proposition, and induction on d that

$$\Delta^{(d)}(r) \leq \left(\frac{(k-1)(1-r)R(r)}{r + (1-r)R(r)}\right)^d.$$

Hence:

$$\Delta^{(d)} \leq \int_0^1 \left(\frac{(k-1)(1-r)R(r)}{r+(1-r)R(r)} \right)^d dr.$$

We split the range of integration into two intervals. For $r \in [\frac{k-2}{k-1}, 1]$ we have $R(r) = 1$ and the integral over this range is easily calculated to be $\frac{1}{(d+1)(k-1)}$.

For $r \in [0, \frac{k-2}{k-1}]$, $R(r)$ maps the interval bijectively to $[0,1]$, and $r = S(R(r))$. Make the substitution $u = R(r)^{1/(k-1)}$, so $r(u) = \frac{u-u^{k-1}}{1-u^{k-1}}$ and $\frac{(1-r)R(r)}{r+(1-r)R(r)} = \frac{(1-u)u^{k-2}}{1-u^{k-1}}$. Also $\frac{dr}{du} = \frac{1-(k-1)u^{k-2}+(k-2)u^{k-1}}{(1-u^{k-1})^2}$ which, for $u \in [0,1]$, is nonnegative and (by a routine calculation) is at most $\frac{(1-u)(k-1)}{1-u^{k-1}}$ This gives:

$$\begin{aligned}
\int_0^{\frac{k-1}{k-2}} \left(\frac{(k-1)(1-r)R(r)}{r+(1-r)R(r)} \right)^d dr &\leq \int_0^1 (k-1)^{d+1} \frac{(1-u)^{d+1} u^{(k-2)d}}{(1-u^{k-1})^{d+1}} du \\
&= \int_0^1 \frac{u^{(k-2)d}}{(\frac{1}{k-1}(1+u+\cdots+u^{k-2}))^{d+1}} du \\
&\leq \int_0^1 \frac{u^{(k-2)d}}{u^{(\frac{k-2}{2})(d+1)}} du \\
&= \frac{2}{kd-2d-k+4} \leq \frac{2}{(k-2)(d-1)+2}.
\end{aligned}$$

The next-to-last inequality follows from the arithmetic-geometric mean inequality.

Summing the two integrals over the two ranges gives the desired upper bound $\Delta_d \leq \frac{3}{(k-2)(d-1)+2}$. ■

Proof of Lemma 13: For $k = 3$, we can explicitly solve for $R_3(r)$ to get

$$R_3(r) = \begin{cases} (\frac{r}{1-r})^2 & r < 1/2 \\ 1 & r \geq 1/2 \end{cases}$$

Integrating this from 0 to 1 yields $R_3 = 2 - 2\ln 2 \geq 0.6137$.

For general $k \geq 3$, we evaluate $\int_0^1 R_k(r)dr$ by a change of variables. As we noted, $R_k(r) = 1$ on $[\frac{k-2}{k-1}, 1]$, and is continuous and strictly increasing function on $[0, \frac{k-2}{k-1}]$ with inverse given by $S_k(t)$. One can then see (either geometrically, or by a change of variables) that $\int_0^1 R_k(r)dr = \int_0^1 (1 - S_k(t))dt$.

We have:

$$1 - S_k(t) \;=\; 1 - \frac{t^{\frac{1}{k-1}} - t}{1-t} \;=\; \sum_{i=0}^{\infty} t^i - t^{i+\frac{1}{k-1}}$$

Integrating this from 0 to 1 yields:

$$\begin{aligned} R_k &= \int_0^1 (1 - S_k(t))dt \;=\; \sum_{i=0}^{\infty} \frac{1}{i+1} - \frac{1}{i+1+\frac{1}{k-1}} \\ &= \sum_{j=1}^{\infty} \frac{1}{j} - \frac{1}{j+\frac{1}{k-1}} \;=\; \frac{1}{k-1} \sum_{j=1}^{\infty} \frac{1}{j(j+\frac{1}{k-1})} \;=\; \frac{\mu_k}{k-1}, \end{aligned}$$

completing the proof. ■

4 General k-SAT

We now proceed to the analysis of general k-CNF formulas. Theorem 8 applies to any formula that has a sufficiently isolated satisfying assignment, but a satisfiable formula need not have such an assignment. Intuitively, though, if F has few satisfying assignments, then it should be close to the unique-SAT case, and if it has many satisfying assignments, then finding one should be easy. Our aim is to formalize this intuition.

As described in the Sections 1.3 and 1.4, upper bounding the running time of **Search**(F_s, I) is accomplished by lower bounding $\tau(F_s)$, the probability that **Modify**(F_s, π, y) returns a satisfying assignment. In the uniquely satisfiable case, we proved such a lower bound, and we will prove a similar lower bound in the general case. However, several new technical difficulties arise. In order to formulate our main lemma for lower bounding $\tau(F_s)$, we need some additional definitions.

We consider nondecreasing, continuous, functions H mapping $[0,1]$ to $[0,1]$ with $H(0) = 0$ and $H(1) = 1$, that are differentiable at all but at most finitely many points, and whose derivative is uniformly bounded on $[0,1]$. Such functions are distribution functions associated to probability distributions on $[0,1]$, and we call them *nice distribution functions*.

For a nice distribution function $H(r)$ having derivative $h(r) = \frac{dH}{dr}$, we define:

$$\begin{aligned}
\beta_H &= \int_0^1 h(r)\log_2(h(r))dr \\
\gamma_H(k) &= \int_0^1 H(r)^{k-1}dr \\
\nu_H(k) &= \frac{(k-1)\beta_H}{k\gamma_H(k)+k\beta_H-1}
\end{aligned}$$

It is routine to show that the (Riemann) integrals are defined and bounded for nice distribution functions. Moreover, it is not hard to show [1]

$$\beta_H = \lim_{M\to\infty}\left(\log_2 M + \sum_{i=1}^{M} g_i \log_2 g_i\right) \tag{1}$$

where $g_i = H(\frac{i}{M}) - H(\frac{i-1}{M})$. Note that g_i is the probability function of a discrete random variable on $[M]$, and the sum in equation (1) is the negative of the entropy of this random variable, which is at most $\log_2 M$. Hence $\beta_H \geq 0$. The negative of β_H is known as *differential entropy of* H (see, for example, [2]).

Our main tool for lower bounding $\tau(F_s)$ is:

Lemma 15 *Let F be a k-CNF formula with $k \geq 3$, let $d = o(\log n)$ and let $s \geq k^d$. Let $\epsilon_k^{(d)} = \frac{3}{(d-1)(k-2)+2}$. Let H be a nice distribution function satisfying $H(r) \leq R_k(r)^{1/(k-1)}$ for $r \in [0,1]$. Then:*

1. $$\tau(F_s) \geq 2^{-\max\{1-\gamma_H(k),\beta_H\}n-\epsilon_k^{(d)}n-o(n)}.$$

2. *If $\beta_H \leq 1-\gamma_H(k)$ then*
$$\tau(F_s) \geq 2^{-(1-\gamma_H(k))n-\epsilon_k^{(d)}n-o(n)}.$$

3. $$\tau(F_s) \geq 2^{-\max\{\nu_H(k),1-\gamma_H(k)\}n-\epsilon_k^{(d)}n-o(n)}$$

[1] Let B be the uniform bound on the derivative. Divide $[0,1]$ into M equal-sized intervals. By the continuity of H and the fact that H has a derivative at all but finitely many points, for each sub-interval $[\frac{i-1}{M}, \frac{i}{M}]$, the quantity $Mg_i = M(H(\frac{i}{M}) - H(\frac{i-1}{M}))$ is between the minimum and maximum value of the derivative on that subinterval. Thus $\log_2 M + \sum_{i=1}^{M}(H(\frac{i}{M}) - H(\frac{i-1}{M}))\log_2(H(\frac{i}{M}) - H(\frac{i-1}{M}))$ is in between the lower and upper Riemann sums for $h(x)\log_2 h(x)$.

In the next subsection we prove this lemma, and we then apply it in subsections 4.2 and 4.3 for specific choices of the nice distribution function H. This leads to the main results stated in Theorems 3, 4 and 5.

4.1 Proof of Lemma 15

The method for bounding $\tau(G)$ outlined in Section 2 and applied in the uniquely satisfiable case focused on the probability $\tau(G, z)$ of accepting a particular assignment z, where z was d-isolated. We need to refine this approach. We start with a simple combinatorial lemma. If a is a partial assignment to the variables $\{x_1, \ldots, x_n\}$, the *subcube defined by* a is the set of all assignments that extend a.

Lemma 16 *Let A be a nonempty set of assignments (i.e., points in $\{0,1\}^n$). Then $\{0,1\}^n$ can be partitioned into a family $(B_z : z \in A)$ of disjoint subcubes so that $z \in B_z$ for each $z \in A$.*

Proof. If $|A| = 1$, the result is trivial. Otherwise, there are at least two assignments, and one variable u which occurs both as 0 and 1 in S. Split $\{0,1\}^n$ into two subcubes, one with $u = 0$ and one with $u = 1$, and recursively partition each. ■

We call such a partition an *A-dissection* of $\{0,1\}^n$. If G is a satisfiable formula, we apply this lemma in the case that A is the set $sat(G)$ of satisfying assignments of the formula G, and henceforth we fix some $sat(G)$-dissection $\{B_z : z \in sat(G)\}$. We will analyze the probability $\tau(G)$ that **Modify**(G, π, y) finds some satisfying assignment by conditioning according to the subcube B_z that contains y. For satisfying assignments w and z write $\tau(G, w|B_z)$ for the probability that **Modify**(G, π, y) returns w given $y \in B_z$. We write $\tau(G|B_z)$ for the sum of $\tau(G, w|B_z)$ over all satisfying assignments w. We then have:

$$\begin{aligned} \tau(G) &= \sum_{z \in sat(G)} \tau(G|B_z)\mathbf{Prob}[y \in B_z] \\ &\geq \sum_{z \in sat(G)} \tau(G, z|B_z)\mathbf{Prob}[y \in B_z], \end{aligned}$$

from which we conclude:

Proposition 17 *For any satisfiable formula G, $\tau(G) \geq \min_{z \in sat(G)} \tau(G, z|B_z)$.*

So, to lower bound $\tau(G)$, we take a generic satisfying assignment z and lower bound the probability $\tau(G, z|B_z)$ that **Modify**(G, π, y) returns z given that $y \in B_z$.

Let $D = D_z$ be the set of variables that defines the subcube B_z (i.e., the set of variables which are constant over that subcube) and let $N = N_z$ be the remaining variables. The variables in D_z are referred to as the *defining* variables of z and those in N_z are referred to as *nondefining*.

Now, to lower bound $\tau(G, z|B_z)$, we try to generalize the argument leading to Lemma 3. Given that $y \in B_z$, we have that y agrees with z on the defining variables, so **Modify**(G, π, y) returns z if and only if the non-defining variables are set according to z. Writing $\text{Forced}_z(G, \pi, y)$ for the set of non-defining variables in $\text{Forced}(G, \pi, y)$, we have the following generalization of Proposition 2:

Proposition 18 *Let z be a satisfying assignment of the formula G. Then*

$$\tau(G, z|B_z) = 2^{-|N_z|}\mathbf{E}[2^{|\text{Forced}_z(G,\pi,z)|}].$$

Continuing the argument, one obtains the following generalization of Lemma 3: if the average of $P(v, G, z)$ over defining variables is at least p then $\tau(G, z|B_z) \geq 2^{-(1-p)|N_z|}$. Since $P(v, G, z) \geq \frac{1}{k}$, we have:

Lemma 19 *Let F be a boolean formula and z a satisfying assignment. Then:*

$$\tau(G, z|B_z) \geq 2^{-(1-\frac{1}{k})|N_z|}$$

This is enough to finish the proof of the general case of Theorem 1, since it implies

$$\tau(G) \geq \min_{z \in sat(G)} \tau(G, z|B_z) \geq 2^{-(1-\frac{1}{k})|N_z|} \geq 2^{-(1-\frac{1}{k})n}.$$

In fact, since the number of nondefining variables is strictly less than n if z is a non-unique solution, the result in this case is strictly better than the bound for the uniquely satisfiable case. In particular if $D_z \geq \delta n$ for some constant $\delta > 0$ then the lower bound on $\tau(G)$ will be $2^{-(1-\frac{1}{k})(1-\delta)n}$. In general, however, this is not good enough to extend Theorem 2 to the general case, nor even to get a non-trivial improvement over the $2^{-(1-\frac{1}{k})n}$ bound,

So, we again consider the effect of adding clauses by resolution and try to generalize Theorem 8. The difficulty comes when we try to construct

the critical clause tree of min-depth d. Recall the proof of Lemma 4, where to extend the tree from a given leaf b_i we used a clause C_i that was not satisfied by $z \oplus L(P_i)$, and the existence of such a clause was guaranteed by the fact that $z \oplus L(P_i)$ does not satisfy the formula (because of the d-isolation of z). Now, however, such a clause C_i need not exist; since z need not be d-isolated, $z \oplus L(P_i)$ might satisfy G. But we do know that if $L(P_i)$ consists only of nondefining variables then $z \oplus L(P_i)$ does not satisfy G since z is the only satisfying assignment in B_z. We now modify the rule for building the tree to say that we never try to expand the tree from a leaf labeled by a defining variable. In this way, we maintain the property that defining variables appear only at leaves, and so by the above observation, it is always possible to expand any other leaf. Thus we conclude the following counterpart to lemma 4.

Lemma 20 *Let F be a k-CNF formula and z an arbitrary satisfying assignment, and let B_z be as defined above. If v is any nondefining variable, and d is any integer, and $s \geq k^d$, there exists a critical clause tree for (v, F_s, z) of maximum degree $k-1$ such that (i) the only nodes labeled by defining variables are leaves, (ii) any leaf labeled by a nondefining variable is at depth d.*

A tree satisfying the conclusion of the lemma is said to be of *min-depth d with respect to the set N_z*. Such trees are nice, but not good enough to directly get a result such as Lemma 5. For instance, it could be that the tree consists of a root together with $k-1$ children all labeled by defining variables. In this case, we get only $1/k$ as the probability Q_T, leading to a bound no better than that given in Lemma 19. As we noted, this bound gives a good improvement on the unique-sat analysis if $|D_z|$ is a large enough fraction of n. The "bad case" for this bound is when D_z is not too big a fraction of n and there are many shallow leaves labeled by defining variables.

So, we need to back up somewhat and generalize an earlier part of the argument. Let's return to Proposition 18 which generalized Proposition 2. The next step in the argument was to use concavity to bring the expectation inside the exponential. This step is still valid, but now it gives too much away. Intuitively here's why. For a random variable X, the inequality $\mathbf{E}[2^X] \geq 2^{\mathbf{E}[X]}$ is tight if X is constant and becomes very loose if X has a small but non-negligible probability of being very large. Now consider the random variable $\text{Forced}_z(G, \pi, z)$, which depends on the random permutation π (which depends in turn on the placement α). If we consider the "bad case" in the development sketched above, we see that permutations

for which the defining variables appear early in the permutation will tend to force many more nondefining variables than permutations for which the defining variables appear later. This creates exactly the situation where the concavity bound is loose.

So we proceed as follows. Suppose we identify a set Γ of placements having "fairly large" probability with the property that the average of $|\text{Forced}_z(G,\pi,z)|$ conditioned on $\alpha \in \Gamma$ is much larger than the overall average. We then have

$$\mathbf{E}[2^{|\text{Forced}_z(G,\pi,z)|}] \geq \mathbf{Prob}[\alpha \in \Gamma]\mathbf{E}_\Gamma[2^{|\text{Forced}_z(G,\pi,z)|}]$$

where $\mathbf{E}_\Gamma$ denotes the conditional expectation given that $\alpha \in \Gamma$. Following the argument as in Section 2, letting $P_\Gamma(v,G,z)$ denote the probability that $\pi \in \text{Last}(v,G,z)$ given that $\alpha \in \Gamma$, and repeating the argument in Section 2 we obtain the following generalization of Lemma 3:

Lemma 21 *Let Γ be a (measurable) subset of placements. For any satisfying assignment z of the CNF formula G if $P_\Gamma(v,G,z) \geq p$ for all nondefining variables v then $\tau(G,z|B_z) \geq 2^{-(1-p)|N_z|}\mathbf{Prob}[\alpha \in \Gamma]$.*

One choice for Γ is the set of all placements α for which all defining variables appear before all nondefining variables. For $\alpha \in \Gamma$, write $\pi(\alpha)$ for the associated permutation and $\pi_N = \pi(\alpha)_N$ for the restriction of π to N_z. Note that, conditioned on $\alpha \in \Gamma$, π_N is uniformly distributed over all permutations of N_z. Let $a = a_z$ denote the partial assignment to D_z that defines B_z. Then, conditioned on $\alpha \in \Gamma$, a variable $v \in N_z$ is last in some critical clause of (v,G,z) if and only, when we consider the restricted function $G\lceil_a$ obtained by fixing the defining variables, then v appears last in some critical clause of $(v,G\lceil_a,z)$. Thus $P_\Gamma(v,G,z) = P(v,G\lceil_a,z)$. Now $G\lceil_a$ is uniquely satisfiable, and so we can apply the lower bound on $P(v,G,z) \geq \frac{\mu_k}{k-1} - \epsilon_k^{(d)}$ obtained in the uniquely satisfiable case. But now, to lower bound $\tau(F_s,z|B_z)$ we must multiply $2^{(1-\frac{\mu}{k-1}+\epsilon_k^{(d)})|N_z|}$ by $\mathbf{Prob}[\alpha \in \Gamma]$, which for this choice of Γ, is $1/\binom{n}{|D_z|} = \Theta\left(\frac{|D_z|}{n}\right)^{|D_z|}$.

Lemma 22 *Let F be a k-CNF formula and z a satisfying assignment. Let $d \geq 1$ and $s \geq k^d$. Then:*

$$\tau(F_s,z|B_z) \geq 2^{-(1-\frac{\mu_k}{k-1}+\epsilon_k^{(d)})|N_z|}\left(\frac{c_0|D_z|}{n}\right)^{|D_z|},$$

where c_0 is a constant and for each k, $\epsilon_k^{(d)}$ is a function that tends to 0 as d gets large.

If $|D_z| = o(n)$ the expression for $\mathbf{Prob}[\alpha \in \Gamma]$ is $2^{-o(n)}$ and the bound in this lemma gives what we want. But for $|D_z| = \Omega(n)$, $\mathbf{Prob}[\alpha \in \Gamma] = 2^{-\Omega(n)}$ and the quality of the lower bound on $\tau(F_s, z|B_z)$ is significantly reduced.

So we can handle the case that D_z is small and also the case that D_z is large. To handle intermediate sizes of D_z, we will aim to choose Γ so that the defining elements *tend* to occur earlier than the nondefining ones, but so that Γ still has reasonably high probability. The sets Γ that we use are defined in terms of the nice distribution functions introduced in the previous subsection.

Let H be a nice distribution function and let $\lambda > 0$. Let D be a set of variables. We say that a placement α is (H, λ)-*good with respect to* D provided that, for each $r > \lambda$, the number of defining variables that are mapped to $[0, r]$ is at least $H(r)|D|$. For fixed D, we define $\Gamma_{H,\lambda}(D)$ to be the set of all placements that are (H, λ)-good. We will apply Lemma 21 with $\Gamma = \Gamma_{H,\lambda}(D)$ for an appropriately chosen H and for sufficiently small λ. In order to apply this lemma for a specific H we need to obtain lower bounds on both $\mathbf{Prob}[\alpha \in \Gamma_{H,\lambda}(D)]$, and on $P_{\Gamma_{H,\lambda}(D)}(v, G, z)$. These bounds, stated below, are expressed in terms of the values β_H and $\gamma_H(D)$ defined in the previous subsection.

Lemma 23 *Let H be a nice distribution function and $\lambda > 0$. Then for a set D of variables,*

$$\mathbf{Prob}[\alpha \in \Gamma_{H,\lambda}(D)] \geq 2^{-\beta_H |D| - o(|D|)}.$$

Lemma 24 *Let F be a k-CNF formula on n variables, and suppose that a dissection of $sat(F)$ is given. Let $z \in sat(F)$, such that $|D_z|, |N_z| \geq \sqrt{n}$. Let v be a nondefining variable. Let $H(r)$ be a nice distribution function satisfying $H(r) \leq R_k(r)^{1/(k-1)}$ for $r \in [0, 1]$, and let $\Gamma = \Gamma_{H,\lambda}(D_z)$, for any $\lambda > 0$. Then for any $d \geq 0$, $d = o(\log n)$, and $s \geq k^d$, as n gets large we have:*

$$P_\Gamma(v, F_s, z) \geq \gamma_H(k) - \epsilon_k^{(d)} - \lambda - o(1),$$

Before proving these two lemmas, we complete the proof of Lemma 15.

Proof of Lemma 15. From the definition of $\gamma_H(k)$ and the fact that $\int_0^1 R_k(r)dr = \frac{\mu_k}{k-1}$ we have:

Proposition 25 *If $H(r)$ is a nice distribution function satisfying $H(r) \leq R_k(r)^{1/(k-1)}$ then $\gamma_H \leq \frac{\mu_k}{k-1}$ with equality if $H(r) = R_k(r)^{1/(k-1)}$.*

By Proposition 17, it suffices to fix $z \in sat(F)$ and lower bound $\tau(F_s, z|B_z)$. If $|D_z| \leq \sqrt{n}$, then Lemma 22 implies that $\tau(F_s, z|B_z) \geq 2^{-(1-\frac{\mu_k}{k-1})n - \epsilon_k^{(d)} n - o(n)} \geq 2^{-(1-\gamma_H)n - \epsilon_k^{(d)} n - o(n)}$, as required. Also, since $\tau(F_s, z|B_z) \leq 2^{-|N_z|}$ holds trivially, we may assume that $|N_z| \geq \sqrt{n}$ (here we could assume more, but we don't need it).

So we may assume that $|D_z|, |N_z| \geq \sqrt{n}$. Let $\delta = \frac{|D_z|}{n}$. We may then apply Lemma 24 to conclude that $P_\Gamma(v, G, z) \geq \gamma_H(k) - \epsilon_k^{(d)} - \lambda - o(1)$, as n gets large.

Applying Lemmas 21 and 23 and writing γ for $\gamma_H(k)$ and β for β_H, we have:

$$\begin{aligned} \tau(F_s, z|B_z) &\geq 2^{-(1-\gamma+\epsilon_k^{(d)}+\lambda+o(1))(1-\delta)n} 2^{-\beta\delta n - o(\delta n)} \\ &\geq 2^{-((1-\gamma)+(\beta+\gamma-1)\delta)n - \epsilon_k^{(d)} n - \lambda n - o(n)}. \end{aligned}$$

Since $\delta \in [0,1]$, the quantity $(1-\gamma)+(\beta+\gamma-1)\delta$ is at most $\max\{1-\gamma, \beta\}$. Furthermore, this bound holds for all $\lambda > 0$, implying

$$\tau(F_s, z|B_z) \geq 2^{-(\max\{1-\gamma,\beta\})n - \epsilon_k^{(d)} n - o(n)}$$

proving the first and second parts of the lemma.

For the third part of the Lemma, note that Lemma 19 implies that $\tau(F_s, z|B_z) \geq 2^{-\frac{k-1}{k}(1-\delta)n}$. Letting $\eta = \min\{(1-\gamma)+(\beta+\gamma-1)\delta, \frac{k-1}{k}(1-\delta)\}$, we have $\tau(F_s, z|B_z) \geq 2^{-\eta n - \epsilon_k^{(d)} n - o(n)}$. We complete the proof by showing that $\eta \leq \max\{\nu_H(k), 1-\gamma_H(k)\}$, where $\nu_H(k)$ is as defined in the previous subsection. If $\beta+\gamma-1 \leq 0$ then the first term in the minimum defining η is at most $1-\gamma$. Also, if $1-\gamma \geq \frac{k-1}{k}$ then $\eta \leq \frac{k-1}{k} \leq 1-\gamma$. Finally, if $\beta+\gamma-1 > 0$ and $1-\gamma < \frac{k-1}{k}$, then η is maximized (as a function of δ) when the two terms in the min are equal, which occurs when $\delta = \frac{k\gamma-1}{k\gamma+k\beta-1}$ and so $\eta \leq \frac{(k-1)\beta}{k\gamma+k\beta-1}$, which is how we defined $\nu_H(k)$. Thus $\eta \leq \max\{\nu_k(H), 1-\gamma_H(k)\}$, and we obtain the desired lower bound on $\tau(F_s, z|B_z)$. ■

Thus to complete the proof of Lemma 15, it remains to prove Lemmas 23 and 24.

Proof of Lemma 23. Throughout the proof the distribution function H is fixed. Note that the event $\alpha \in \Gamma_{H,\lambda}(D)$ depends only on the placement of the set D of defining variables. We show that for every $\epsilon > 0$, if $|D|$ is sufficiently large then $\mathbf{Prob}[\alpha \in \Gamma_{H,\lambda}(D)] \geq 2^{-(\beta_H+2\epsilon)|D| - o(|D|)}$ (where o does not depend on ϵ). The lemma then follows.

We will choose an integer M and consider a partition of $[0,1]$ into M equal-sized intervals, separated by the points $m_i = \frac{i}{M}$, $0 \leq i < M$. We refer to the interval $[m_i, m_{i+1}]$ as the i^{th} bin. Define $g_i = H(m_i) - H(m_{i-1})$. Let $M = \lfloor \sqrt{|D|} \rfloor$. We assume that $|D|$ is large enough such that:

1. $M > \frac{1}{\lambda}$
2. $H(\frac{2}{M}) < \epsilon$
3. $|\beta_H - (\log_2 M + \sum_{i=1}^{M} g_i \log_2 g_i)| < \epsilon$

The second condition can be satisfied since $H(0) = 0$ and H is continuous. The third can be satisfied since equation (1) implies that the expression inside the absolute value goes to 0 as $M \to \infty$.

For a placement α, define the *M-profile* of the placement to be the sequence $a_1(\alpha), a_2(\alpha), \ldots, a_M(\alpha)$ where $a_i(\alpha)$ is the number of defining variables mapped to bin $i-1$. A sufficient condition for α to be in $\Gamma_{H,\lambda}$ is that for all $j \in [M-1]$, $\sum_{i=1}^{j} a(\alpha_i) \geq |D| H(m_{j+1})$, since this means that for $r \geq \lambda > m_1$, if j is the index such that $m_j \leq r \leq m_{j+1}$, then the number of defining variables v for which $\alpha(v) < r$ is at least $|D| H(\frac{j+1}{M}) \geq |D| H(r)$. Define the sequences $b_1, \ldots, b_M$ and $a_1, \ldots, a_M$ of nonnegative integers by $b_i = \lceil |D| H(m_i) \rceil - \lceil |D| H(m_{i-1}) \rceil$, and $a_1 = b_1 + b_2$, $a_i = b_{i+1}$ for $2 \leq i \leq M-1$ and and $a_M = 0$. Note that $\sum b_i = \sum a_i = D$ and that the above sufficient condition implies that any placement α whose profile is $a_1, \ldots, a_M$ will be (H, λ)-good. Thus we can lower bound the probability that α is (H, λ)-good by the probability that α has profile $a_1, \ldots, a_m$, which is:

$$M^{-|D|} \frac{|D|!}{a_1! \ldots a_M!} \geq a_1^{-a_1} \ldots a_M^{-a_M} \left(\frac{|D|}{eM} \right)^{(} |D| - M)$$

Here we used $|D|! \geq (\frac{|D|}{e})^{|D|}$ to lower bound the numerator and $x! \leq (\frac{x}{e})^{x+1}$ to upper bound each term in the denominator for which $a_i \geq 1$. After cancellation, the denominator is equal to $\prod_{i=1}^{M} a_i^{a_i}$ times the product over $a_i \geq 1$ of $\frac{a_i}{e}$. By the arithmetic-geometric mean inequality, this latter product is at most $\left(\frac{|D|}{Me}\right)^M$. Simplifying gives the above expression on the right.

Let $p_i = \frac{a_i}{|D|}$ and $q_i = \frac{b_i}{D}$. The expression on the right is:

$$2^{|D|(-\log M - \sum_i p_i \log p_i) + o(|D|)}.$$

We now claim that $|\sum_{i=1}^{M} p_i \log p_i - \sum_{i=1}^{M} g_i \log g_i| \leq \epsilon + o(1)$, (where the $o(1)$ term is as D gets large). Assuming this claim, and using the third condition on M, we would have that the above expression is at least $2^{-|D|(\beta_H + 2\epsilon) + o(|D|)}$, as required. So it suffices to prove the claim. We have:

$$|\sum p_i \log p_i - \sum g_i \log g_i| \leq |\sum p_i \log p_i - \sum q_i \log q_i| + |\sum q_i \log q_i - \sum g_i \log g_i|.$$

Now $|\sum p_i \log p_i - \sum q_i \log q_i| = |(q_1+q_2)\log(q_1+q_2) - q_1 \log q_1 - q_2 \log g_2|$. Letting $\gamma = q_1/(q_1+q_2)$, this is equal to $(q_1+q_2)|\gamma \log_2 \gamma + (1-\gamma)\log_2(1-\gamma)|$ which is at most $q_1 + q_2 \leq H(m_2) + \frac{1}{|D|} \leq \epsilon + \frac{1}{|D|}$.

To upper bound $|\sum q_i \log q_i - \sum g_i \log g_i|$ we use the inequality $|y \log y - x \log x| \leq |y - x| \log |y - x|$ and $|g_i - q_i| \leq \frac{1}{|D|}$ to bound the sum by $\frac{M \log |D|}{|D|}$. Since $M \leq \sqrt{|D|}$, this is $o(1)$ in $|D|$.

■

Proof of Lemma 24. We want to lower bound $P_\Gamma(v, G, z)$, where $v \in N = N_z$ and $\Gamma = \Gamma_{H,\lambda}(D_z)$. We need a counterpart to Proposition 6. For an admissible tree T, let $Q_{T,\Gamma}(r) = \mathbf{Prob}_\Gamma[\mathrm{Cut}_T(r)]$ and $Q_{T,\Gamma} = \mathbf{Prob}_\Gamma[\mathrm{Cut}_T]$. Exactly as before we have:

$$P_\Gamma(v, G, z) \geq Q_{T,\Gamma} = \int_0^1 Q_{T,\Gamma}(r) dr.$$

In the case that z was d-isolated, we could assume that there was an admissible tree T with root labeled by z, that had min-depth at least d and degree at most $k-1$. We then showed that $\mathbf{Prob}[\mathrm{Cut}_T(r)] \geq Q_k^{(d)}(r)$ which approaches $R_k(r)$. Now, things are more complicated. First of all, the critical clause tree promised by Lemma 20 is only min-depth d *with respect to* N_z; leaves labeled by variables in D_z may be shallower. Secondly, $\mathrm{Cut}_T(r)$ is determined by the set of variables v for which $\alpha(v) \leq r$, and in the unconditioned case, each variable belongs to this set independently with probability r, while in the conditioned case, this is only true of the non-defining variables. The following Lemma provides the needed extension of Lemma 9.

Lemma 26 *Let F be a formula and suppose that a dissection relative to $sat(F)$ is given. Let z be a satisfying assignment and let D and N be the defining and non-defining variables corresponding to z. Suppose that $v \in N$ and that there is a critical clause tree T that is degree $k-1$ and min-depth d*

with respect to N. Let $H(r)$ be a nice distribution function and let $r \in [\lambda, 1]$. Then

$$Q_{T,\Gamma_{H,\lambda}}(r) \geq \min\{H(r)^{k-1}, Q_k^{(d)}(r)\} - \rho(H(r))$$

where $\rho(x) = 0$ for $x \in \{0,1\}$ and $\rho(x) = \min\{\frac{k^{2d}}{|D|}\left(\frac{1}{x(1-x)}\right), 1\}$ for $x \in (0,1)$.

Proof. Let $w(r) = \lceil |D|H(r) \rceil$, $D(r) = \{v : v \in D, \alpha(v) < r\}$, and $N(r) = \{v : v \in N, \alpha(v) < r\}$. Observe that $\alpha \in \Gamma_{H,\lambda}$ implies that $|D(r)| \geq w(r)$ for $r \geq \lambda$. If we condition on $|D(r)| = w$ then $D(r)$ is uniformly distributed among all subsets of D of size w. Furthermore note that conditioning further on $\alpha \in \Gamma_{H,\lambda}$ does not further change the distribution of $D(r)$. This implies that for $w \geq w(r)$,

$$\mathbf{Prob}[\mathrm{Cut}_T(r) : \alpha \in \Gamma_{H,\lambda}(D), |D(r)| = w] = \mathbf{Prob}[\mathrm{Cut}_T(r) : |D(r)| = w].$$

Thus we have:

$$\begin{aligned} Q_{T,\Gamma}(r) &= \sum_{w=w(r)}^{|D|} \mathbf{Prob}[\mathrm{Cut}_T(r) : |D(r)| = w]\mathbf{Prob}_\Gamma[|D(r)| = w] \\ &\geq \mathbf{Prob}[\mathrm{Cut}_T(r) : |D(r)| = w(r)]. \end{aligned}$$

The last inequality is true because $\sum_{w=w(r)}^{|D|} Prob_\Gamma[|D(r)| = w] = 1$ and $\mathbf{Prob}[\mathrm{Cut}_T(r) : |D(r)| = w]$ is monotone in w.

Thus it suffices to lower bound $\mathbf{Prob}[\mathrm{Cut}_T(r) : |D(r)| = w(r)]$. Fix T and r, and let D^T and N^T be the defining and nondefining variables appearing in T, and let $D^T(r) = D^T \cap D(r)$ and $N^T(r) = N^T \cap N(r)$. The event $\mathrm{Cut}_T(r)$ depends only on $(N^T(r), D^T(r))$. The distribution $\Gamma_{H,\lambda}(D)$ conditioned on $|D(r)| = w(r)$ induces a distribution $\Lambda = \Lambda_H(D)$ on $(N^T(r), D^T(r))$: $N^T(r)$ and $D^T(r)$ are chosen independently. $N^T(r)$ has the binomial distribution $B(|N^T|, r)$. The distribution of $D^T(r)$ is the distribution induced by choosing a subset C' of $D(r)$ uniformly at random from the sets of size $w(r)$ and intersecting it with D^T. The resulting distribution satisfies that for a given subset $C \subseteq D^T$,

$$\mathbf{Prob}_\Lambda[D^T(r) = C] = \frac{\prod_{i=1}^{|C|}(w(r)+1-i)\prod_{i=1}^{|D^T|-|C|}(|D|-w(r)+1-i)}{\prod_{i=1}^{|D^T|}(|D|+1-i)}$$

Abusing notation, let us write $Q_{T,\Lambda}(r)$ for $\mathbf{Prob}_\Lambda[\mathrm{Cut}_T(r)]$. Then our lower bound on $Q_{T,\Gamma}(r)$ can be written $Q_{T,\Gamma}(r) \geq \mathbf{Prob}_\Lambda[\mathrm{Cut}_T(r)]$.

So now we want to lower bound $Q_{T,\Lambda}(r)$. To do this, we approximate the distribution Λ by the simpler distribution Λ' on $(N(r), D(r))$ where for each $v \in D$, we put v in $D(r)$ independently with probability $H(r)$. We show two things:

$$Q_{T,\Lambda}(r) \geq Q_{T,\Lambda'}(r) - \rho(H(r)). \tag{2}$$

$$\text{If } T \text{ is min-depth } d \text{ with respect to } N(z) \text{ then} Q_{T,\Lambda'}(r) \geq \min\{H(r)^{k-1}, Q_k^{(d)}(r)\}. \tag{3}$$

These two facts together imply the conclusion of the Lemma.

So let us prove (2). If $H(r) = 1$, then $D^T(r) = D^T$ under both distributions Λ and Λ', and the distributions are identical. Similarly if $H(r) = 0$ then $D^T(r) = \emptyset$ under both distributions and again the distributions are identical. So assume $H(r) \in (0, 1)$.

Observe that for any $C \subseteq D^T$, the distributions Λ and Λ' conditioned on $D^T(r) = C$ are identical. Thus:

$$\begin{aligned}
\frac{Q_{T,\Lambda'}(r)}{Q_{T,\Lambda}(r)} &= \frac{\sum_{C \subseteq D^T} \mathbf{Prob}_\Lambda[\mathrm{Cut}_T(r) : D^T(r) = C]\mathbf{Prob}_{\Lambda'}[D^T(r) = C]}{\sum_{C \subseteq D^T} \mathbf{Prob}_\Lambda[\mathrm{Cut}_T(r) : D^T(r) = C]\mathbf{Prob}_\Lambda[D^T(r) = C]} \\
&\leq \max_{C \subseteq D^T} (\mathbf{Prob}_{\Lambda'}[D^T(r) = C]/\mathbf{Prob}_\Lambda[D^T(r) = C]).
\end{aligned}$$

Here the inequality follows from the fact that if $a_1, \ldots, a_t, b_1, \ldots, b_t$ are nonnegative reals then $(\sum_i b_i)/(\sum_i a_i) \leq \max_i b_i/a_i$.

Now for each $C \subseteq D^T$:

$$\begin{aligned}
\mathbf{Prob}_\Lambda[D^T(r) = C] &= \frac{\prod_{i=1}^{|C|}(w(r)+1-i)\prod_{i=1}^{|D^T|-|C|}(|D|-w(r)+1-i)}{\prod_{i=1}^{|D^T|}(|D|+1-i)} \\
&\geq \left(\frac{H(r)|D|-|C|}{|D|}\right)^{|C|}\left(\frac{(1-H(r))|D|-(|D^T|-|C|)}{|D|}\right)^{|D^T|-|C|} \\
&= H(r)^{|C|}(1-H(r))^{|D^T|-|C|}\left(1-\frac{|C|}{H(r)|D|}\right)^{|C|}\left(1-\frac{|D^T|-|C|}{(1-H(r))|D|}\right)^{|D^T|-|C|} \\
&\geq \mathbf{Prob}_{\Lambda'}[D^T(r) = C]\left(1-\min\{\frac{|D^T|^2}{|D|}\left(\frac{1}{H(r)}+\frac{1}{1-H(r)}\right), 1\}\right) \\
&\geq \mathbf{Prob}_{\Lambda'}[D^T(r) = C](1-\rho(H(r))).
\end{aligned}$$

Hence, we conclude that $Q_{T,\Lambda'}(r) \geq Q_{T,\Lambda}(r)(1-\rho(H(r))) \geq Q_{T,\Lambda}(r) - \rho(H(r))$ as required.

Next we prove (3). We first need a variant of Lemma 7. That lemma expressed $Q_T(r)$ as a product of $r + (1-r)Q_{T_i}(r)$ where T_i ranged over all subtrees rooted at labeled children of the root of T. Now, we must take into account the possibility that some of the children of the root might be leaves labeled by defining variables.

Lemma 27 *Let T be an admissible tree with more than one node, and with root labeled by v. Let $T_1, \ldots, T_t$ be the admissible subtrees rooted at the labeled children of the root of T, and let $T_1, \ldots, T_s$ be those whose root is labeled by a defining variable. Then:*

$$Q_{T,\Lambda'}(r) \geq H(r)^s \left(\prod_{i=s+1}^{t} (r + (1-r)Q_{T_i,\Lambda'}(r)) \right).$$

Proof. The proof is a slight variant of that of Lemma 7. Using the notation of that proof, let $K' = \wedge_{i \in [s]} K_i$.

Using Lemma 8, we write $\mathbf{Prob}_{\Lambda'}[\wedge_{i\in[t]} K_i] \geq \mathbf{Prob}_{\Lambda'}[K'] \left(\prod_{i=s+1}^{t} \mathbf{Prob}_{\Lambda'}[K_i] \right)$. We bound $\mathbf{Prob}_{\Lambda'}[K'] \geq \mathbf{Prob}_{\Lambda'}[\wedge_{i\in[s]}(v_i \in D(r))] \geq H(r)^s$ (the inequality is strict only if some of the v_i are equal) and for $s+1 \leq i \leq t$, $\mathbf{Prob}_{\Lambda'}[K_i] \geq r + (1-r)Q_{T_i,\Lambda'}(r)$. ■

Now, we prove (3) by induction on d. If $d = 0$, the conclusion is true since $Q_k^{(0)}(r) = 0$. So assume $d \geq 1$ and that T is min-depth d with respect to N_z. Let $T_1, \ldots, T_t$ be the subtrees rooted at labeled children of the root, and let $T_1, \ldots, T_s$ be those whose root is in D_z. Then for $s+1 \leq i \leq t$, T_i is an admissible tree of min-depth $t-1$ with respect to N_z, and hence $Q_{T_i,\Lambda'} \geq \min\{H(r)^{k-1}, Q_k^{(d-1)}(r)\}$.

Let $y = \min\{H(r)^{k-1}, Q_k^{(d-1)}(r)\}$. By applying Lemma 27, and recalling the definition of the function $f_k(x;r)$ we have $Q_{T,\Lambda'}(r) \geq H(r)^s f_k(y;r)^{(t-s)/(k-1)} \geq H(r)^s f_k(y;r)^{(k-1-s)/(k-1)}$.

We claim that $f_k(y;r) \geq \min\{H(r)^{k-1}, Q_k^{(d)}(r)\}$. Assuming the claim, it follows immediately that $Q_{T,\Lambda'}(r) \geq \min\{H(r)^{k-1}, Q_k^{(d)}(r)\}$ as required.

To prove the claim, we have by definition and the fact that $f_k(x;r)$ is an increasing function of x on $[0,1]$, that $f_k(y;r) = \min\{f_k(H(r)^{k-1};r), f_k(Q_k^{(d-1)};r)\}$. The latter term in the min is just $Q_k^{(d)}(r)$. Since $f_k(x;r) \geq x$ for $x \in [0, R_k(r)]$, and $f_k(x;r) \geq R_k(r)$ for $x \geq R_k(r)$ we have that $f_k(H(r)^{k-1}) \geq \min(H(r)^{k-1}, R_k(r))$. Since $Q_k^{(d)}(r) \leq R_k(r)$, we conclude that $f_k(y;r) \geq \min\{H(r)^{k-1}, Q_k^{(d)}(r)\}$.

■

We are now ready to finish the proof of Lemma 24. We are trying to lower bound $P_\Gamma(v, F_s, z)$, and for this it suffices to lower bound $\int_0^1 Q_{T,\Gamma}(r)dr$. By Lemma 26, for $r \geq \lambda$, we can lower bound the integrand by $\min\{H(r)^{k-1}, Q_k^{(d)}(r)\} - \rho(H(r))$.

Since $Q_{T,\Gamma}$ (and thus any lower bound to it) is at most 1, then

$$P_\Gamma(v, F_s, z) = \int_0^1 Q_{T,\Gamma}(r)dr \geq \int_\lambda^1 Q_{T,\Gamma}(r)dr \geq \int_0^1 \min\{H(r)^{k-1}, Q_k^{(d)}(r)\} - \rho(H(r))dr - \lambda$$

and we can focus on computing this last integral.

The hypothesis of the present lemma gives $H(r)^{k-1} \leq R_k(r)$ and thus $Q_k^{(d)}(r) = R_k(r) - \Delta_k^{(d)}(r) \geq H(r)^{k-1} - \Delta_k^{(d)(r)}$. Thus $\min\{H(r)^{k-1}, Q_k^{(d)(r)}\} \geq H(r)^{k-1} - \Delta_k^{(d)}(r)$ and $Q_{T,\Gamma}(r) \geq H(r)^{k-1} - \Delta_k^{(d)}(r) - \rho(H(r))$. Integrating this, we obtain that

$$\int_0^1 \min\{H(r)^{k-1}, Q_k^{(d)}(r)\} - \rho(H(r))dr \geq \gamma_H(k) - \epsilon_k^{(d)} - \int_0^1 \rho(H(r))dr$$

All that remains is to show that the integral of $\rho(H(r))$ is $o(1)$ as n gets large.

Let $r_0 = \sup\{r : H(r) = 0\}$ and $r_1 = \inf\{r : H(r) = 1\}$ and let $\zeta > 0$. Then:

$$\begin{aligned}
\int_0^1 \rho(H(r))dr &= \int_{r_0}^{r_1} \rho(H(r))dr \\
&\leq \int_{r_0}^{r_0+\zeta} dr + \int_{r_0+\zeta}^{r_1-\zeta} \frac{k^{2d}}{|D|} \left(\frac{1}{H(r)(1-H(r))} \right) dr + \int_{r_1-\zeta}^{r_1} dr \\
&\leq 2\zeta + \frac{k^{2d}}{|D|} \left(\frac{1}{H(r_0+\zeta)(1-H(r_1-\zeta))} \right).
\end{aligned}$$

Since H is fixed, for each fixed $\zeta > 0$, and $d = o(\log n)$, we can take n sufficiently large so that using $|D| \geq \sqrt{n}$, the second term in the last expression is at most ζ, and thus the last expression is at most 3ζ. Since this is true for any positive ζ, we conclude that as n tends to ∞, this is $o(1)$. ■

4.2 Applying Lemma 15: Proof of Theorems 3 and 5

Now we would like to select H so as to obtain the best lower bound we can on $\tau(F_s)$. One natural candidate for H is the function $H_k(r) = R_k(r)^{1/(k-1)}$, which is the largest it can be to apply Lemma 15. We show:

Lemma 28 *For each $k \geq 3$, H_k is a nice distribution function and*

$$\beta_{H_k} = \int_0^1 \log_2 \frac{(1-y^{k-1})^2}{1+(k-2)y^{k-1}-(k-1)y^{k-2}} dy$$

Proof. Fix $k \geq 3$. Using Proposition 11, we see that H_k is an increasing function on $[0, \frac{k-2}{k-1}]$ that maps onto $[0,1]$ and is 1 for $r \in [\frac{k-2}{k-1}, 1]$. Also, on the interval $[0, \frac{k-2}{k-1}]$, the inverse of $y = H_k(r)$ is $r = S_k(y^k)$ (as in Proposition 11), and so:

$$\begin{aligned} H_k^{-1}(y) &= \frac{y-y^{k-1}}{1-y^{k-1}} \\ h(r) = \frac{dy}{dr} &= \frac{(1-y^{k-1})^2}{1+(k-2)y^{k-1}-(k-1)y^{k-2}} \end{aligned}$$

where the second equality follows from the first by differentiating with respect to r and solving for $\frac{dy}{dr}$.

It is straightforward to show that $y \in [0,1]$ implies $1 \leq h(r) \leq \frac{(k-1)^2}{k-2}$ (Divide both numerator and denominator by $(1-y)^2$ and compare the resulting polynomials). Thus H_k is differentiable and has uniformly bounded derivative at all points of $[0,1]$ except $\frac{k-2}{k-1}$, and is thus a nice distribution function. Therefore β_{H_k} is well defined. Noting that $h(r) = 0$ for $r \geq \frac{k-2}{k-1}$ and making the change of variables $r = H_k^{-1}(y)$ (so that $h(r)dr = dy$) we have:

$$\begin{aligned} \beta_{H_k} &= \int_0^1 h(r)\log_2 h(r) dr = \int_0^{\frac{k-2}{k-1}} \log_2 h(r)(h(r)dr) \\ &= \int_0^1 \log_2 \frac{(1-y^{k-1})^2}{1+(k-2)y^{k-1}-(k-1)y^{k-2}} dy \end{aligned}$$

■

The following table of numerical values for β_{H_k} (using Lemma 28), $1-\gamma_{H_k} = 1 - \frac{\mu_k}{k-1}$ (using Proposition 25), and $\nu_H(k)$.

k	β_H	$1-\gamma_H(k)$	$\nu_H(k)$
3	1.115	0.387	0.533
4	0.666	0.556	0.581
5	0.478	0.649	0.784
6	0.373	0.711	0.339

From this, we immediately obtain Theorem 5 and a weaker version of Theorem 4 by applying the third part of Lemma 15. Under the assumptions of that lemma, we obtain

$$\tau(F_s) \geq 2^{-.533n} 2^{-\epsilon_k^{(d)} n - o(n)}$$

for $k = 3$ and

$$\tau(F_s) \geq 2^{-.581n} 2^{-\epsilon_k^{(d)} n - o(n)}$$

for $k = 4$, from which the theorems follow.

For $k \geq 5$, the following lemma yields Theorem 3, again by applying the third part of Lemma 15.

Lemma 29 *For each $k \geq 5$, $\beta_{H_k} \leq (1 - \gamma_{H_k}) = 1 - \frac{\mu_k}{k-1}$.*

which is itself a consequence of the values in the table above and the following two lemmas.

Lemma 30 *β_H decreases monotonically with k.*

Lemma 31 *$\frac{\mu_k}{k-1}$ decreases monotonically with k.*

Proof of Lemma 30. For $y \in [0,1]$, let $\theta_k(y) = \frac{(1-y^{k-1})^2}{1+(k-2)y^{k-1}-(k-1)y^{k-2}}$. We show that $\theta_k(y) > \theta_{k+1}(y)$ pointwise. The lemma follows by noting that $\log x$ is an increasing function of x and then integrating.

First, define

$$\begin{aligned} f_k = f_k(y) &= 1 + 2y + 3y^2 + \ldots + (k-2)y^{k-3} \\ g_k = g_k(y) &= 1 + y + y^2 + \ldots + y^{k-2} \end{aligned}$$

With these definitions

$$\theta_k = \frac{(1-y)^2 g_k^2}{(1-y)^2 f_k} = \frac{g_k^2}{f_k}$$

Thus, we can show that $\theta_k \geq \theta_{k+1}$ by showing that $f_{k+1} g_k^2 - f_k g_{k+1}^2 \geq 0$

$$\begin{aligned} f_{k+1} g_k^2 - f_k g_{k+1}^2 &= f_{k+1} g_k^2 - f_k g_k^2 - f_k g_{k+1}^2 + f_k g_k^2 \\ &= g_k^2 (f_{k+1} - f_k) - f_k (g_{k+1}^2 - g_k^2) \\ &= (y^{k-2} g_k)((k-1) g_k - f_k (2y + \frac{y^k}{g_k})) \\ &= (y^{k-2} g_k)(A - B) \end{aligned}$$

where we define

$$\begin{aligned} A &= (k-1)g_k - 2yf_k \\ B &= \frac{f_k y^k}{g_k} \end{aligned}$$

Since $(y^{k-2}g_k) > 0$ for $y \in (0,1)$, we need only show that $A \geq B$ to prove the lemma. To do so, we show $A \geq k-1 \geq B$.

$$\begin{aligned} A &= (k-1)g_k - 2yf_k \\ &= (k-1) + (k-3)y + (k-5)y^2 + \ldots - (k-3)y^{k-2} \\ &= (k-1) + (k-3)(y - y^{k-2}) + (k-5)(y^2 - y^{k-3}) + \ldots \\ &\quad \ldots + (k - 2\lfloor k/2 \rfloor + 1)(y^{\lfloor k/2 \rfloor} - y^{\lceil k/2 \rceil}) \\ &\geq k-1 \end{aligned}$$

while $B \leq k-1$ follows from

$$(k-1)g_k \geq g_k^2 \geq f_k \geq f_k y^k$$

Thus $A \geq B$, $\theta_k(y) - \theta_{k+1}(y) \geq 0$, and the lemma holds. ■

Proof of Lemma 31. To show that

$$\frac{\mu_k}{k-1} = \frac{1}{k-1} \sum_{j=1}^{\infty} \frac{1}{j(j + \frac{1}{k-1})}$$

decreases monotonically with k, we view this expression as a function of a (real-valued) variable k and differentiate with respect to k to obtain

$$-\sum_{j=1}^{\infty} \left(\frac{j}{1 + j^2(k-1)} \right)^2$$

which is obviously negative, implying that $\frac{\mu_k}{k-1}$ is a decreasing function of k. ■

4.3 Applying Lemma 15: Improving the bound for $k = 3$

Since it is possible that the formula has only one satisfying assignment, we do not expect to prove better bounds for general formula than we do for

uniquely satisfiable ones. However, for $k = 3, 4$, there is a gap separating what we can prove in the two cases. Here, we will focus on improving the results for $k = 3$.

Lemma 15 gives us a procedure for proving lower bounds. For any nice distribution function H satisfying the constraint $H(r) \leq R_k(r)^{1/(k-1)}$ it gives a bound on $\tau(F_s)$ in terms of the quantities β_H and $\gamma_H(k)$. Thus far, we have made use of one specific distribution function $H_k(r) = R_k(r)^{1/(k-1)}$ in proving bounds using Lemma 15. For, $k \geq 5$, this distribution produces the same value of $\gamma_H(k)$ as we obtained in the uniquely satisfiable case, which is the best we could hope for. Unfortunately, this distribution H is very skewed for small k: $\beta_H(3) = 1.115$, so this distribution is very unlikely to occur. To obtain stronger bounds for the case $k = 3$, we want to choose a function H that more closely balances β_H and $1 - \gamma_H(3)$.

To this end fix an integer M. Given a vector $\vec{a}$ indexed by $\{0, 1 \ldots, M\}$, we associate it to the unique function a on $[0, 1]$ satisfying (i) $a(\frac{i}{M}) = \vec{a}(i)$ for $i \in \{0, 1, \ldots, M\}$ and (ii) a is piecewise linear on each subinterval $[\frac{i-1}{M}, \frac{i}{M}]$, so for $i \in \{1, \ldots, M\}$ and $\alpha \in [0, 1]$, $a(\frac{i-1+\alpha}{M}) = (1 - \alpha)a(\frac{i-1}{M}) + \alpha a(\frac{i}{M})$. Define the difference vector $\Delta\vec{a}$ by $\Delta\vec{a}(i) = \vec{a}(i) - \vec{a}(i-1)$. The derivative a' of the function a is constant on the ith subinterval and is equal to $M\Delta\vec{a}(i)$.

We consider piecewise linear distribution functions H of this form. Thus we describe H by $\vec{H}$ that satisfies $0 = \vec{H}(0) \leq \vec{H}(1) \leq \cdots \leq \vec{H}(M) = 1$. As we show below the quantities β_H and $\gamma_H(k)$, can be easily computed from $\vec{H}$.

To compute β_H, note that for $r \in [\frac{i-1}{M}, \frac{i}{M}]$, $H'(r) = M\Delta\vec{H}(i)$. Thus the integral for β_H can be expressed as a summation

$$\sum_{i=1}^{M} \frac{1}{M}(M\Delta\vec{H}(i)) \log(M\Delta\vec{H}(i)) = \log M + \sum_{i=1}^{M} \Delta\vec{H}(i) \log(\Delta\vec{H}(i))$$

.

Thus, defining

$$\vec{\beta}(i) = \Delta\vec{H}(i) \log(\Delta\vec{H}(i)) + \frac{\log M}{M},$$

we have $\beta_H = \sum_i \vec{\beta}(i)$.

Similarly, we get a summation formula for $\gamma_H(3)$. For $r \in (\frac{i-1}{M}, \frac{i}{M})$, write $r = \frac{i-1+x}{M}$ with $x \in (0, 1)$. Then $H(r) = \vec{H}(i) + x\Delta\vec{H}(i)$, and $H(r)^2 = \vec{H}(i)^2 + 2x\Delta\vec{H}(i)\vec{H}(i) + x^2\Delta\vec{H}(i)^2$. Then

$$\int_0^1 H(r)^{k-1} dr \quad = \quad \sum_i \frac{1}{M} \int_0^1 (\vec{H}(i)^2 + 2x\Delta\vec{H}(i)\vec{H}(i) + x^2\Delta\vec{H}(i)^2) dx$$

$$= \sum_i \frac{1}{M}(\vec{H}(i)^2 + \Delta\vec{H}(i)\vec{H}(i) + \frac{\Delta\vec{H}(i)^2}{3})$$

Defining

$$\vec{\gamma}(i) = \frac{1}{M}(\vec{H}(i)^2 + \Delta\vec{H}(i)\vec{H}(i) + \frac{\Delta\vec{H}(i)^2}{3}),$$

we have $\gamma_H(3) = \sum_i \vec{\gamma}(i)$.

We also need to be able to check that H satisfies the constraint $H(r) \leq R_3(r)^{1/2}$. Let $Q(r) = R_3(r)^{1/2}$ Recall that $Q(r) = \frac{r}{1-r}$ on $[0, \frac{1}{2}]$ and is 1 for $r \geq 1/2$, so we need only consider $r \leq \frac{1}{2}$. Given $\vec{H}$, we can easily check this inequality at the points $\frac{i}{M}$, but because $Q(r)$ is convex on $[0, \frac{1}{2}]$ and H is piecewise linear, it is not sufficient to only check the inequality at the interval boundaries. Define the vector $\vec{S}$ by

$$\vec{S}(i) = Q(\frac{i-1}{M}) + \frac{1}{M}Q'(\frac{i-1}{M}),$$

where Q' is the derivative of Q and we define $Q(\frac{-1}{M}) = Q'(\frac{-1}{M}) = 0$.

We claim that the associated piecewise linear function S derived from $\vec{S}$ satisfies $S(r) \leq Q(r)$ for all r. Assuming the claim, to check that $H(r) \leq Q(r)$ it suffices to check that $H(r) \leq S(r)$ for all r, and for this it is enough to check that $\vec{H}(i) \leq \vec{S}(i)$ for $i = 1, \ldots, M$. To verify the claim, we prove by induction on $i \geq 1$ that $S(r) \leq Q(r)$ is satisfied in the interval $[\frac{i-1}{M}, \frac{i}{M}]$. For $\alpha \in [0, 1]$, the piecewise linearity of S implies:

$$\begin{aligned} S(\frac{i-1+\alpha}{M}) &= (1-\alpha)S(\frac{i-1}{M}) + \alpha S(\frac{i}{M}) \\ &= (1-\alpha)S(\frac{i-1}{M}) + \alpha Q(\frac{i-1}{M}) + \frac{\alpha}{M}Q'(\frac{i-1}{M}). \end{aligned}$$

For $i = 1$, the first two terms on the righthand side are 0, so the righthand side is equal to $\frac{\alpha Q'(0)}{M}$, which is at most $Q(\frac{\alpha}{M})$ by the convexity of Q, as required. For $i \geq 1$, the induction hypothesis implies that $Q(\frac{i-1}{M}) \geq S(\frac{i-1}{M})$ and so the righthand side is at most $Q(\frac{i-1}{M}) + \frac{\alpha}{M}Q'(\frac{i-1}{M}) \leq Q(\frac{i-1+\alpha}{M})$ by the convexity of Q.

Hence to apply Lemma 15, it suffices to find any $\vec{H}$ satisfying $\vec{H}(i) \leq \vec{S}(i)$ for all $i \in \{0, 1, \ldots, M\}$ and to compute $\beta_H = \sum_i \vec{\beta}(i)$ and $\gamma_H(3) = \sum_i \vec{\gamma}(i)$. In Table 4.3, we present a particular $\vec{H}$. Here $M = 25$, and each row of the table gives $\vec{H}(i)$, $\vec{S}(i)$, $\Delta\vec{H}(i)$, $\vec{\beta}(i)$ and $\vec{\gamma}(i)$.

We found this $\vec{H}$ by using the Differential Evolution program [11] to minimize $\max\{\sum_i \vec{\beta}(i), 1 - \sum_i \vec{\gamma}(i)\}$, subject to the constraint $\forall i, \vec{H}(i) \leq \vec{S}(i)$.

From this table and the discussion above, it is easy to see that H is a valid distribution function, and H satisfies the constraint $H(r) \leq R_k(r)^{1/(k-1)}$. Also, $\beta_H \leq 0.4471$ and $1 - \gamma_h(k) \leq 0.4471$. By applying the first part of Lemma 15 to this distribution function H, we obtain Theorem 4.

Naturally, one could consider improving this result by using a finer discretization or more thorough optimization. However, our experiments show only very modest improvements by such methods (changes only in the third decimal place of the exponent for $k = 3$ and changes on the order of 0.02 for similar experiments for $k = 4$), so significant improvements will require other ideas.

5 Lower Bounds for Depth-3 Circuits

In proving lower bounds on the size of Σ_k^3 and Σ^3 circuits needed to compute specific boolean functions, we follow the general approach used in [9]. In that paper, the key observation was that Theorem 1 places an upper bound on the number of isolated solutions that a k-CNF formula can have. Here we use Theorem 8 to make a similar observation. If F is a function, circuit or formula, let $I_d(F)$ denote the number of d-isolated points in $F^{-1}(1)$.

Lemma 32 *Let n, k, d be integers with $n \geq k, d \geq 1$. Let F be a CNF formula on n-variables and suppose that F has at most L clauses of size larger than k. Then:*

$$I_d(F) \leq 2^{(1-\frac{\mu_k}{k-1}+\epsilon_k^{(d)})n} + Ln^d 2^{n-k}$$

where $\epsilon_k^{(d)} = \frac{3}{(d-1)(k-2)+2}$.

Proof. First consider the case that $L = 0$, which means that F is a k-CNF. By Theorem 8, if $s = k^d$ then for each d-isolated satisfying assignment z to F, the probability $\tau(F_s, z)$ that **Modify**(F_s, π, y) outputs z is at least $2^{-(1-\frac{\mu_k}{k-1}+\epsilon_k^{(d)})n}$. Since **Modify**$(G, \pi, y)$ outputs at most one assignment, we have $1 \geq \sum_z \tau(F_s, z) \geq I_d(F) 2^{-(1-\frac{\mu_k}{k-1}+\epsilon_k^{(d)})n}$. The upper bound on $I_d(F)$ follows.

Now let F be a general CNF. Write $F = F_{small} \wedge F_{large}$ where F_{large} consists of the clauses of size more than k. For each clause C of F_{large} let Z_C

i	$\vec{H}$	$\vec{S}$	$\Delta\vec{H}$	$\vec{\beta}$	$\vec{\gamma}$
1	0.0400	0.0400	0.0400	0.0000	0.0000
2	0.0851	0.0851	0.0451	-0.0158	0.0002
3	0.1342	0.1342	0.0491	-0.0279	0.0005
4	0.1880	0.1880	0.0538	-0.0411	0.0010
5	0.2472	0.2472	0.0591	-0.0555	0.0019
6	0.3125	0.3125	0.0653	-0.0714	0.0031
7	0.3850	0.3850	0.0725	-0.0888	0.0049
8	0.4660	0.4660	0.0810	-0.1080	0.0073
9	0.5571	0.5571	0.0910	-0.1290	0.0105
10	0.6499	0.6602	0.0928	-0.1326	0.0146
11	0.7270	0.7778	0.0771	-0.0992	0.0190
12	0.7894	0.9133	0.0624	-0.0640	0.0230
13	0.8392	1.0000	0.0499	-0.0299	0.0265
14	0.8784	1.0000	0.0392	0.0027	0.0295
15	0.9088	1.0000	0.0304	0.0325	0.0319
16	0.9323	1.0000	0.0234	0.0588	0.0339
17	0.9502	1.0000	0.0180	0.0815	0.0354
18	0.9639	1.0000	0.0136	0.1012	0.0366
19	0.9742	1.0000	0.0103	0.1176	0.0376
20	0.9820	1.0000	0.0078	0.1311	0.0383
21	0.9879	1.0000	0.0059	0.1423	0.0388
22	0.9923	1.0000	0.0045	0.1510	0.0392
23	0.9956	1.0000	0.0033	0.1584	0.0395
24	0.9981	1.0000	0.0025	0.1642	0.0398
25	1.0000	1.0000	0.0019	0.1687	0.0399
Totals			1.0000	0.4470	0.5530

Table 1: A piecewise linear function H that gives $\max\{\beta_H, 1-\gamma_H\} \leq .4471$ for $k = 3$.

be the set of points that do not satisfy C and let $Z = \bigcup_{C \in F_{large}} Z_C$. Note that $F^{-1}(1) = F^{-1}_{small}(1) - Z$. If $x \in I_d(F)$ then (i) $x \in I_d(F_{small})$, or (ii) there is a point $x' \in Z$ that is within d of x. The number of points satisfying (i) is at most $2^{(1-\frac{\mu_k}{k-1}+\epsilon_k^{(d)})n}$ by the $L = 0$ case. To upper bound the number of points satisfying (ii), note that each point $x' \in Z$ has at most n^d points within distance d of it. Since for a large clause C, $|Z_C| = 2^{n-|C|} < 2^{n-k}$, we have that the number points satisfying (ii) is at most $L2^{n-k}n^d$ as required. ■

Before using this observation to prove circuit lower bounds, we review some simple facts about depth 3 circuits. We may assume that every Σ^3 circuit is constructed in a levelled fashion: the inputs are connected to OR gates, these OR gates are connected to AND gates, and these AND gates are connected to the output OR gate. A circuit which does not have this property can be easily converted into this form. Because the gates have unbounded fanin, two gates of the same type which are connected can simply be collapsed into one gate. If an input x_i is directly connected to an AND gate or to the output gate, we modify the circuit to maintain this levelled property by introducing new gates of fanin one at the levels which it skips. Since this process introduces at most $O(n)$ gates, it has negligible impact on the exponential lower bounds we prove, and we may assume that all circuits are levelled.

Call the AND gates of the circuit *CNF gates*, since they compute functions which are ANDs of ORs of inputs, and call the circuit rooted at such a gate a *CNF subcircuit*. Similarly, call the non-output OR gates of the circuit *clause gates*. As usual, given a circuit $\mathcal{C}$ or a formula F, we use $\mathcal{C}(x)$ and $F(x)$ to denote the Boolean functions they compute.

Lemma 32 implies an upper bound on the number of d-isolated points accepted by any Σ^3 circuit in terms of the number of gates:

Lemma 33 *Let $\mathcal{C}$ be an Σ^3 circuit on n variables and let k, d be positive integers. Let Q be the number of CNF gates and R be the number of clause gates having fan-in more than k. Then*

$$I_d(\mathcal{C}) \leq 2^n \left(Q2^{(-\frac{\mu_k}{k-1}+\epsilon_k^{(d)})n} + R2^{-k}n^d \right).$$

Proof. Let $G_1, \ldots, G_Q$ be the CNF gates of $\mathcal{C}$. Each d-isolated point of $F^{-1}(1)$ is accepted by at least one G_i and is still d-isolated in $G^{-1}(1)$. Hence $I_d(F) \leq \sum_{i=1}^{Q} I_d(G_i)$ and so Lemma 32 implies that this is at most $Q2^{(1-\frac{\mu_k}{k-1}+\epsilon_k^{(d)})n} + R2^{n-k}n^d$. ■

An immediate consequence of Lemma 33 is:

Corollary 34 *Let F be an n-variate boolean function. Any Σ^3_k circuit computing F has at least $I_d(F)2^{-(1-\frac{\mu_k}{k-1}+\epsilon_k^{(d)})n}$ CNF gates.*

Proof of Theorem 6. Recall that we want to prove a lower bound on the smallest Σ^3_k circuit computing membership in an error correcting code E with $|E| \geq 2^{n-(\sqrt{n}/\log n)}$ and distance greater than $\log n$. For $d = \lfloor \log n \rfloor$, every element of E is d-isolated and so by Corollary 34, any circuit computing $\mathcal{C}$ has at least $|E|2^{-(1-\frac{\mu_k}{k-1}+\epsilon_k^{(d)})n}$ gates. Since $\epsilon_k^{(d)} = \frac{3}{(d-1)(k-2)+2} \leq 4/\log n$ we conclude that $\mathcal{C}$ has at least $2^{\frac{\mu_k}{k-1}n - \frac{5n}{\log n}}$ gates. ■

Next we consider lower bounds on Σ^3 circuits with no restriction on fan-in.

Corollary 35 *Let F be an n-variate boolean function. Any Σ^3 circuit computing F has at least*

$$I_d(F)2^{-n+\min\{(\frac{\mu_k}{k-1}-\epsilon_k^{(d)})n, k-d\log n\}}$$

gates.

Proof. By Lemma 33, for any circuit $\mathcal{C}$, $I_d(F) \leq 2^n(Q+R)\max\{2^{(-\frac{\mu_k}{k-1}+\epsilon_k^{(d)})n}, 2^{-k+d\log n}\}$. Since $Q+R$ lower bounds the number of gates of $\mathcal{C}$, the corollary follows. ■

Proof of Theorem 7. Again we are looking at the membership function for the code E. In applying Corollary 35, we are free to choose k to make $\min\{(\frac{\mu_k}{k-1} - \epsilon_k^{(d)})n, k - d\log n\}$ as large as possible.

First we note:

$$\frac{\pi^2}{6} - \mu_k = \sum_{j=1}^{\infty} \frac{1}{j^2} - \frac{1}{j(j+\frac{1}{k-1})} = \sum_{j=1}^{\infty} \frac{\frac{1}{j(k-1)}}{j(j+\frac{1}{k-1})} \leq \sum_{j=1}^{\infty} \frac{\frac{1}{k-1}}{j(j+\frac{1}{k-1})} = \frac{\mu_k}{k-1}$$

and so $\frac{\mu_k}{k-1} \geq \frac{\pi^2}{6k}$

As for $\epsilon_k^{(d)}$, as long as k and d are both growing functions of n, then for sufficiently large n, $\epsilon_k^{(d)} = \frac{3}{(d-1)(k-2)+2} \leq \frac{4}{dk}$.

Since $d = \log n$, the first term in the min is at least $\frac{n}{k}(\frac{\pi^2}{6} - \frac{4}{\log n})$ and the second is at least $k - \log^2 n$. Choose $k = \lfloor \sqrt{\frac{\pi^2 n}{6}} \rfloor$. Then both terms in the

min are at least $\sqrt{\frac{\pi^2 n}{6}} - \frac{4\sqrt{n}}{\log n}$. Since $I_d(E) = |E| \geq 2^{n-\sqrt{n}/\log n}$, Corollary 35 gives a circuit size lower bound

$$2^{\sqrt{\frac{\pi^2 n}{6}} - \frac{5\sqrt{n}}{\log n}}$$

proving the theorem. ■

6 Acknowledgements

We thank Professor Ed Bender for his help in evaluation the integral in Section 3.4. We also thank Richard Beigel and Yuming Zhang for his helpful comments and corrections.

References

[1] Alon, N., Spencer, J., and Erdös, P., (1992), "The Probabilistic Method", John Wiley & Sons, Inc.

[2] Cover, T. and Thomas, J. (1991), "Elements of Information Theory", New York: Wiley.

[3] Davis, M., Logemann, G., and Loveland, D., (1962), A machine program for theorem proving, *Communications of the ACM*, 5:394–397.

[4] Håstad, J., (1986), Almost Optimal Lower Bounds for Small Depth Circuits, *in* " Proceedings of the 18th ACM Symposium on Theory of Computing", pp. 6–20.

[5] Håstad, J., Jukna, S., and Pudlák, P., (1995), Top–Down Lower Bounds for Depth 3 Circuits, Computational Complexity 5, pp. 99–112.

[6] E. A. Hirsch, Two New Upper Bounds for SAT, SIAM conference on Discrete Algorithms, 1997.

[7] Kullmann, O. New methods for 3-SAT decision and worst-case analysis. *To appear in* Theoretical Computer Science.

[8] Monien, B. and Speckenmeyer, E., (1985), Solving Satisfiability In Less Than 2^n Steps, Discrete Applied Mathematics **10**, pp. 287–295.

[9] Paturi, R., Pudlák, P., and Zane, F., (1997), Statisfiability Coding Lemma, *in* Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science, pp 566–574, October 1997.

[10] Paturi, R., Saks, M.E., and Zane F., (1997), Exponential Lower Bounds on Depth 3 Boolean Circuits, *in* Proceedings of the 29th Annual ACM Symposium on Theory of Computing", pp. 86-91

[11] Price, K. and Storn, R. (1997), Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, Journal of Global Optimization **11**, pp. 341–359.

[12] Razborov, A.A. (1986), Lower Bounds on the Size of Bounded Depth Networks over a Complete Basis with Logical Addition, *Mathematische Zametki* **41** pp. 598–607 (in Russian). English Translation in*Mathematical Notes of the Academy of Sciences of the USSR* **41**, pp. 333–338.

[13] Rodosek, R. (1996) A new approach on solving 3-satisfiability, Proc. 3rd Intern. Conf. on AI and Symbolic Math. Computation, Springer-Verlag LNCS 1138, pp. 197–212.

[14] Schiermeyer, I. (1993), Solving 3-Satisfiability in less than 1.579^n Steps, *in* Selected papers from CSL '92, LNCS Vol. 702, pp. 379-394.

[15] Smolensky, R. (1987), Algebraic Methods in the Theory of Lower Bounds for Boolean Circuit Complexity *in* "Proceedings of the 19th ACM symposium on Theory of Computing", pp. 77–82.

[16] Schöning, U. (1999), A probabilistic algorithm for k-SAT and constrain satisfaction problems, preprint

[17] Valiant, L.G., (1977), Graph–theoretic arguments in low–level complexity, in *Proceedings of the 6th Symposium on Mathematical Foundations of Computer Science*, Springer–Verlag, Lecture Notes in Computer Science, vol. 53, pp. 162–176.

[18] Yao, A. C–C. (1985), Separating the Polynomial Hierarchy by Oracles, *in* "Proceedings of the 31st Annual IEEE Symposium on Foundations of Computer Science", pp. 1–10.

[19] Zhang, W. (1996), Number of models and satisfiability of sets of clauses, Theoretical Computer Science **155**, pp. 277-288.