Survey on $k$-satisfiable Partial Satisfaction

Bachelor Thesis
Hu Qinheping
January 22, 2014

Advisors: Prof. Welzl, Dr. Hertli
Department of Computer Science, ETH Zürich
Abstract

A CNF formula $F$ is $k$-satisfiable if each $k$ clauses of $F$ can be satisfied simultaneously. Define $r_k$ to be the supremum of all real number $r$ so that every weighted $k$-satisfiable CNF formula has an assignment that satisfies at least $r$-fraction of the weight. Define $s_k$ analogously for unweighted CNF formulas.

We survey the results of the bound of $r_k$ and $s_k$, i.e., under which conditions imposed on a CNF formula which fraction of the clauses or weight is guaranteed to be simultaneously satisfied by some assignment, and consider whether such an assignment can be found algorithmically.

Furthermore, we discuss the problem of finding a assignment for $k$-satisfiable CNF formula satisfying at least given $r$-fraction of the total weight where $r > r_k$. 
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contents</td>
<td>iii</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2 Previous Results</td>
<td>5</td>
</tr>
<tr>
<td>2.1 1-satisfiable CNF formula</td>
<td>5</td>
</tr>
<tr>
<td>2.2 2-satisfiable CNF formula</td>
<td>6</td>
</tr>
<tr>
<td>2.3 3-satisfiable CNF formula</td>
<td>10</td>
</tr>
<tr>
<td>2.4 4-satisfiable CNF formula</td>
<td>16</td>
</tr>
<tr>
<td>2.5 k-satisfiable CNF formula</td>
<td>20</td>
</tr>
<tr>
<td>2.6 Summary</td>
<td>23</td>
</tr>
<tr>
<td>3 Algorithm Implementation</td>
<td>25</td>
</tr>
<tr>
<td>3.1 Algorithm for the probabilistic method</td>
<td>25</td>
</tr>
<tr>
<td>3.2 Algorithm ENUMERATE</td>
<td>26</td>
</tr>
<tr>
<td>4 The Hardness of &gt; r_k Problem.</td>
<td>29</td>
</tr>
<tr>
<td>Bibliography</td>
<td>33</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Satisfiability (SAT) is the problem of deciding whether a boolean formula in propositional logic has an assignment that evaluates to true. SAT occurs as a problem and is a tool in applications, e.g., Artificial Intelligence and circuits, and it is considered a fundamental problem in theory. A partial satisfaction problem is to consider how much fraction of a not-satisfiable formula can be satisfied. In this thesis, we discuss the satisfiability problem base a local satisfiability condition and we can see that how local satisfiability implies fractional global satisfiability.

In Boolean logic, a formula is in conjunctive normal form (CNF) if it is a conjunction of clauses, where a clause is a disjunction of literals. We use the simplified notation that

- a literal \( l \) is either a variable \( x \in V \) or its negation \( \overline{x} \);
- a clause \( C \) is a set of literals \( l \);
- a CNF formula \( F \) over a finite variables set \( V = \{x_1,...,x_n\} \) is a set \( \{C_1,...,C_m\} \) of disjunctive clauses.

We say that a clause is "\( m \)-clause" if it contains exactly \( m \) literals. We require that no clause contains a variable and its negation simultaneously because such clause is always satisfied and meaningless.

We define a vocabulary function as follow: if \( l \) is a literal, then vbl\((l)\) is its variable, i.e., vbl\((x)\)=vbl\((\overline{x})\) = \( x \) for \( x \in V \). In a formula \( F \), switching a variable \( x \) means replacing each occurrence of \( x \) in \( F \) by \( \overline{x} \) and vice versa.

For \( V \) a set of boolean variables, an assignment \( \alpha \) on \( V \) is a mapping \( \alpha : V \rightarrow \{0,1\} \), i.e. \( \alpha \in \{0,1\}^V \). We say that an assignment \( \alpha \) satisfies a literal \( l \) if either \( l = x \) for \( x \in V \) with \( \alpha(x) = 1 \) or \( l = \overline{x} \) with \( \alpha(x) = 0 \), an assignment \( \alpha \) satisfies a clause \( C \) if there exists a literal \( l \in C \) such that \( \alpha \) satisfies \( l \), an assignment \( \alpha \) satisfies a CNF formula \( F \) if \( \alpha \) satisfies all
clauses of $F$. A random assignment is a probability distribution over all possible assignments. We will restrict that in a random assignment each variable is assigned true with a certain probability, independently of the other variables. Thus a random assignment $\alpha_R$ is entirely specified by the probabilities $\{p_x\}_{x \in V}$, where $\Pr[\alpha_R(x) = 1] = p_x$.

Sometimes we just want to assign the value of some of the variables rather than all of them. So for a CNF formula $F$ and one of its variable $x$, we denote
\[
F[x \mapsto 1] = \{ C \setminus \{x\} \mid C \in F \text{ and } x \notin C \},
\]
\[
F[x \mapsto 0] = \{ C \setminus \{x\} \mid C \in F \text{ and } \overline{x} \notin C \}.
\]

Sometime the considered formula can not be globally satisfied. So we want the considered formula weighted such that we can evaluate how much of it can be satisfied at the same time. The idea of weighted formulas is to assign a real number (weight) to each clause such that for an assignment $\alpha$, we can see how much weight of the formula can be satisfied by this assignment. Here is a formal definition: for a CNF formula $F$ on a variable set $V$, we define a clause weight function $\mu : F \rightarrow \mathbb{R}^+$, and the function $\mu$ extends to subsets $G \subset F$ by $\mu(G) = \sum_{C \in G} \mu(C)$. Given a truth assignment $\alpha$, let $\mu[\alpha](F)$ denote $\mu(\{C \in F \mid \alpha \text{ satisfies } C\})$. Finally, let $\mu^*(F) := \max_\alpha \mu[\alpha](F)$. Note that an unweighted formula can be seen as a weighted formula in which each clause has same weight.

Instead of global satisfiable formulas, in this thesis we mainly discuss the formulas which are “locally satisfiable” in the sense that small subsets of the clauses are simultaneously, i.e., we deal with the notion of $k$-satisfiable CNF formula introduced and studied by Lieberherr and Specker [1].

**Definition 1.1** A CNF formula is $k$-satisfiable if any $k$ clauses of it can be simultaneously satisfied.

If a CNF formula is not satisfiable, it is sometimes desired to at least satisfy as much weight as possible. Here we use two ratios to investigate how much is the fraction of the weight and the clauses that can be always satisfied, given certain local satisfiable preconditions on the formulas.

**Definition 1.2** Define $r_k$ as follow:
\[
r_k := \inf \left\{ \frac{\mu^*(F)}{\mu(F)} \mid F \text{ is } k\text{-satisfiable}, \, \mu : F \rightarrow \mathbb{R}^+ \right\},
\]
and for formulas with unit clause weights, we define
\[
s_k := \inf \left\{ \frac{\mu^*(F)}{\mu(F)} \mid F \text{ is } k\text{-satisfiable}, \, \mu : F \rightarrow \mathbb{R}^+, \, \mu(C) = 1 \forall C \in F \right\}.
\]
Thus the ratio $s_k$ is the supremum of the set of reals such that in any $k$-satisfiable formula with $m$ clauses, at least $s_k m$ clauses can be simultaneously satisfied. Similarly, the ratio $r_k$ is the supremum of the set of reals such that in any weighted $k$-satisfiable formula, at least fraction $r_k$ of the total weight can be simultaneously satisfied.

In next chapter, we will discuss some previous results and their proofs of the value of $r_k$ and $s_k$ which shows how local satisfiability implies fractional global satisfiability. Then we try to implement algorithms to find the assignment which reach the results in chapter 2 for $k$-satisfiable CNF formulas.

At the last chapter, we discuss the hardness of the problems of finding an assignment for $k$-satisfiable CNF formula satisfying at least $r$-fraction of the total weight where $r > r_k$. 
In this chapter, we can see the previous results of the value or bound of $r_k$ and $s_k$. This table is an overview of this chapter:

<table>
<thead>
<tr>
<th>$k$</th>
<th>$r_k$</th>
<th>$s_k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
</tr>
<tr>
<td>2</td>
<td>$\sqrt{3} - \frac{1}{2}$</td>
<td>$\frac{2}{3}$</td>
</tr>
<tr>
<td>3</td>
<td>$\frac{3}{5} + \frac{\sqrt{69} - 11}{2}$</td>
<td>$\frac{7}{10}$</td>
</tr>
<tr>
<td>$\to \infty$</td>
<td>$\frac{3}{4}$</td>
<td>$\frac{3}{4}$</td>
</tr>
</tbody>
</table>

2.1 1-satisfiable CNF formula

We are not interesting in 0-satisfiable because every CNF formula is 0-satisfiable, it is no help! We start from the 1-satisfiable CNF formula. In fact, 1-satisfiable means that the formula contains no empty clause, that is, condition “1-satisfiable” is equivalent to “($\geq 1$)-CNF”. It is easy to see that at least half of the weight of a 1-satisfiable CNF formula can be satisfied.

**Theorem 2.1** $r_1 = s_1 = \frac{1}{2}$.

**Proof** Given a 1-satisfiable CNF formula $F$ on variables set $V$. We pick an arbitrary assignment $\alpha$ on $V$ and switch $\alpha$ to get $\overline{\alpha}$, i.e., each variable $x$ is set to be true by $\overline{\alpha}$ iff it is set to be false by $\alpha$. Thus for each clause $C$ which is not satisfied by $\alpha$, the assignment $\overline{\alpha}$ can satisfy $C$. If $\alpha$ just satisfies no more than $1/2$-fraction of the clauses, the assignment $\overline{\alpha}$ satisfies at least half of all clauses. So one of $\alpha$ and $\overline{\alpha}$ can satisfy at least half of the clauses in $F$. That is, a lower bound of $s_1$ is $\frac{1}{2}$.

Similarly, if the formula $F$ is weighted, for any assignment $\alpha$ and its negation $\overline{\alpha}$ of $F$, one of $\alpha$ and $\overline{\alpha}$ can satisfy at least half of the weight in $F$, which implies that $r_1 \geq \frac{1}{2}$.
2. Previous Results

These two bounds are tight, as can be seen from

\[ F := \{ \{x_1\}, \{\overline{x}_1\}\}. \]

with all weights equal 1.

\[ \square \]

2.2 2-satisfiable CNF formula

Now we look into the 2-satisfiable CNF formulas. In this part we will see that “2-satisfiable” allows us to satisfy significantly more than half of the weight in the formula.

Lieberherr and Specker [1] first gave a bound of \( r_2 \) (in fact, they gave a proof to bound \( s_2 \), which is wrong for \( s_2 \) but correct for \( r_2 \)). They used symmetrization technique to show that \( r_2 \geq \sqrt{5/2} - 1/2 \) and defined a family \((F_i)_{i \in \mathbb{N}}\) of 2-satisfiable formulas for which \( \sqrt{5/2} - 1/2 \) is asymptotically tight.

The idea of symmetrization is to construct a symmetric formula (i.e., \( F_2 \) in the following proof) from an arbitrary 2-satisfiable formula \( F \) such that we can easily compute the fraction of the weight satisfied in the symmetric formula and then we use that fraction to bound the fraction of the weight satisfied in \( F \).

Since the proof of the lower bound of \( r_2 \) is constructive, we can modify the parameter of the symmetric formula (i.e., \( F_2 \)) to construct the family of 2-satisfiable CNF formulas which shows that the bound is tight.

In fact, there exists a simple probabilistic method raised by Yannakakis to prove the same bound, which we will see in next section. Here we just show the proof by Lieberherr and Specker.

**Theorem 2.2 (Lieberherr and Specker)\( r_2 = \frac{\sqrt{5} - 1}{2} \).**

**Proof** Choose an arbitrary weighted 2-satisfiable CNF formula \( F \) on variables set \( V \). We do following process to \( F \):

1. Switch variable \( x \in V \) if some 1-clause \( C \) contains the negated literal of \( x \), i.e, \( \overline{x} \in C \) and \( |C| = 1 \);
2. For each clause \( C \in F \), if \( C \) contains at least one positive literal, drop all except one positive literal;
3. For each clause \( C \in F \), if \( C \) contains only negative literals, drop all except two.

Because \( F \) contains neither the empty clause nor a pair \( \{x\}, \{\overline{x}\} \) of complementary 1-clauses, this process will terminate after a finite number of steps and we can get a new CNF formula, which we call \( F_1 \), at the end of the process. The formula \( F_1 \) is still 2-satisfiable because it has following properties:
2.2. 2-satisfiable CNF formula

- The clauses of length 1 only contain positive literal.
- The clauses of length 2 only contain negative literals.
- There are no other clauses.

Since \( F_1 \) is reduced from \( F \), each assignment \( \alpha \) of \( F \) corresponds to an assignment \( \alpha_1 \) of \( F_1 \) which satisfies at least as much weight as \( \alpha \) in \( F \).

Now we start to construct a symmetric formula. For a permutation \( \Pi \) on \( V \), we denote \( \Pi(F_1) \) a formula substituting \( \Pi(x) \) for each variable \( x \) of \( F_1 \) and keep the same weight for each clause. We define \( F_2 \) to be the concatenation of sequences \( \Pi(F_1) \) for all permutation \( \Pi \) in full-permutation group, i.e., for each \( \Pi \) we add all clauses in \( \Pi(F_1) \) into \( F_2 \) and here we allow the formula contains multiple same clauses. Then it is easy to see that \( F_2 \) has following properties:

- For each variable \( x \) in \( F_2 \), the formula \( F_2 \) contains same number of 1-clauses \( \{x\} \).
- For each variable pair \( \{x, y\} \), the formula \( F_2 \) contain same number of 2-clauses \( \{x, y\} \).

(In general, the formula \( F_2 \) has such properties if it is the concatenation of sequence \( \Pi(F_1) \) for all permutation \( \Pi \) in the permutation group \( PG \) where \( PG \) is a doubly transitive permutation group. A permutation group \( PG \) is doubly transitive if for all \( x_1 \neq x_2, y_1 \neq y_2 \) there is a permutation \( \Pi \) in \( PG \) such that \( \Pi(x_1) = y_1 \) and \( \Pi(x_2) = y_2 \). The full permutation group used in this proof is obviously a doubly transitive group.)

Therefore, the formula \( F_2 \) is symmetric, i.e. assignments setting \( k \) of variables to be true satisfies same fraction of weight in \( F_2 \). Assume that assignment \( \alpha_2 \) sets \( k \) variables to be true, then obviously \( k/n \) of all 1-clauses are satisfied and \( ((\binom{n}{2} - \binom{k}{2})/\binom{n}{2}) \) of the 2-clauses are satisfied. That is, the weight of 1-clauses satisfied is equal to \( ku \) and of 2-clauses satisfied is \( (\binom{k}{2} - \binom{k}{2})v \) where \( u \) is the total weight of all 1-clauses in \( F_1 \) and \( v \) is the total weight of 2-clauses in \( F_1 \). Note that the total weight of \( F_2 \) is \( nu + \binom{n}{2}v \). So the assignment \( \alpha \) satisfies fraction

\[
 f(n, a, k) := \frac{ka + \binom{k}{2} - \binom{k}{2}}{na + \binom{n}{2}} = \frac{ka + n^2/2 - n/2 - k^2/2 + k/2}{na + n^2/2 - n/2}
\]

of weight in \( F_2 \) where \( a := u/v \).

Now we consider the min-max problem

\[
 \min_{0 < a < n - 1} \max_{0 \leq k \leq n} f(n, a, k).
\]

Note that the function \( f(n, a, k) \) can be seen as a quadratic function of \( k \) and \( k \) is an integer. So the function \( f(n, a, k) \) reach its maximum when \( k = a \). We
2. Previous Results

substitute $a$ for $k$ in $f(n,a,k)$ and obtain

$$f_1(n,a) = \frac{a^2 + a + n^2 - n}{2na + n^2 - n}.$$  

Then considering the first derivative $\frac{\partial f_1(n,a,k)}{\partial a}$, we can easily know that the function $f_1(n,a)$ attains its minimum $\frac{\sqrt{5} - 1}{2}$ for $a/n = h$.

That is, there exists a natural number $k$ where $0 \leq k \leq n$ such that

$$ka + \left(\begin{array}{c} n \\ 2 \end{array}\right) - \left(\begin{array}{c} k \\ 2 \end{array}\right) > \frac{\sqrt{5} - 1}{2}.$$  

Therefore, there exists an assignment $a_2$ satisfying at least fraction $\frac{\sqrt{5} - 1}{2}$ weight of $F_2$.  

Because $F_2$ is the concatenation of all permutations of $F_1$, there exists at least one permutation $\Pi$ on $V$ such that $a_2$ also satisfies at least fraction $\frac{\sqrt{5} - 1}{2}$ weight of $\Pi(F_1)$. Then we can use the inverse permutation of $\Pi$ to get an assignment $a_1$ from $a_2$ such that $a_1$ satisfies at least fraction $\frac{\sqrt{5} - 1}{2}$ weight of $F_1$. At last, the lower bound of $r_2$ is at least $\frac{\sqrt{5} - 1}{2}$ since $F_1$ is reduced from $F$ and $F$ is an arbitrary 2-satisfiable CNF formula.

Then, for the sake of showing that this bound is tight, we define a sequence $F^1, F^2, ...$ of 2-satisfiable CNF formula. We want to show that: for any $r = r_2 + \delta > r_2$, there is one CNF formula $F$ in this sequence such that no assignment can satisfy at least fraction $r$ weight of $F$, which implies that the bound of $r_2$ is tight. We use the idea of $F_2$ to define $F^n$: CNF formula $F^n$ has $n$ variable and following properties:

- For each variable $x$ in $F_2$, the formula $F_2$ contains same number of 1-clauses $\{x\}$;
- For each variable pair $\{x, y\}$, the formula $F_2$ contain same number of 2-clauses $\{x, \overline{y}\}$;

and the quotient of the total weight of all 1-clauses in $F^n$ and the total weight of all 2-clauses in $F^n$ is equal to $a_n$. We set

$$a_n = n \cdot \frac{F^n}{F_{n+1}},$$

where $F_n$ is the $n$th Fibonacci number.

Recall that an assignment setting $k$ variables to be true in $F^n$ can satisfy fraction $f(n,a,k)$ weight of $F^n$. And the function $f(n,a,k)$ attains its maximum when $a/n = \frac{\sqrt{5} - 1}{2}$.  


By the first derivative, we know that \( f_1(n,a,k) \) as the function of \( a \) is decreasing as \( a \) decreasing and reach the minimum \( \frac{\sqrt{5} - 1}{2} \) when \( t = \frac{\sqrt{5} - 1}{2} \). Observe that \( \frac{a_{n+1}}{a_n} > \frac{a_n}{n} \) and
\[
\lim_{n \to \infty} \frac{a_n}{n} = \frac{\sqrt{5} - 1}{2}.
\]
Therefore, the fraction of satisfied weight in \( F_n \) is decreasing as \( n \) increasing and has an limit \( \frac{\sqrt{5} - 1}{2} \). So for this CNF formula sequence, the maximal fraction of weight which can be satisfied converges to \( r_2 \), implying that the bound of \( r_2 \) is tight.

Lieberherr and Specker [1] also provided an algorithm ENUMERATE to find an assignment satisfying at least \( r_2 \) fraction of the weight in 2-satisfiable CNF formula follow this constructive proof. The idea of the algorithm is using a doubly transitive permutation group with polynomial size rather then full-permutation whose size is \( n! \) to construct \( F_2 \) in this proof. We will see this algorithm in next chapter.

Surely, \( s_k \geq r_k \) for any integer \( k \) since weighted formulas are generalize of unweighted. We have seen that for the 1-satisfiable CNF formulas, the ratio \( r_1 \) and \( s_1 \) have same value. But for the 2-satisfiable CNF formula, the ratio \( r_2 \) and \( s_2 \) have different value, i.e., \( s_2 = 2/3 > \frac{\sqrt{5} - 1}{2} = r_2 \) as shown in the following.

Käppeli and Scheder [4] showed that \( s_2 \geq \frac{2}{3} \) and provided a certain 2-satisfiable CNF formula which achieve this bound. The proof is a combinatorial proof, using two different way to count the total size of all clauses which contain only negative literals.

**Theorem 2.3 (Käppeli and Scheder)** \( s_2 = \frac{2}{3} \).

**Proof** Given an arbitrary 2-satisfiable formula \( F \), we just need to show that there exists an assignment satisfying at least \( 2/3 \) number of clauses in \( F \). For a variable \( x \), we define
\[
d^+(x) := |\{C \in F \mid x \in C \text{ and } C \setminus \{x\} \text{ contains only negative literals}\}|
\]
\[
d^-(x) := |\{C \in F \mid \overline{x} \in C \text{ and } C \text{ contains only negative literals}\}|.
\]
That is, \( d^+(x) \) is the set of clauses containing only one positive literal \( x \) and several negative literals, and \( d^-(x) \) is the set of clauses containing only negative literals including \( \overline{x} \). We define these two set because we want to argue that a all-1 assignment can satisfy at least \( s_2 \) fraction of clauses after the following process. Now as long as possible, apply the following rules:

1. If \( d^-(x) > d^+(x) \), switch \( x \);
2. Previous Results

2. If \( d^{-}(x) = d^{+}(x) \) and \( \{\overline{x}\} \in F \), switch \( x \).

This process terminates after a finite number of steps. In the end of the process, for any variable \( x \) of \( F \), we have \( d^{+}(x) \geq d^{-}(x) + 1 \) if \( \{x\} \in F \) and \( d^{+}(x) \geq d^{-}(x) \).

Then let

\[
F^- := \{ C \in F | C \text{ contains only negative literals} \}; \\
F^+ := \{ C \in F | C \text{ contains exactly one positive literal} \}.
\]

We calculate

\[
|F^+| = \sum_{x \in V, \{x\} \in F} d^{+}(x) + \sum_{x \in V, \{x\} \notin F} d^{+}(x) \\
\geq \sum_{x \in V, \{x\} \in F} (1 + d^{-}(x)) + \sum_{x \in V, \{x\} \notin F} d^{-}(x) \\
= |\{\{\overline{x}\} \in F\}| + \sum_{x \in V} d^{-}(x) \\
= |\{\{\overline{x}\} \in F\}| + \sum_{C \in F^-} |C|
\]

Note that \( \sum_{C \in F^-} |C| \) contains two parts: the clauses in \( F^- \) with length 1, denoted by \( F^{-}_{1} \), and the clauses with length at least 2, denoted by \( F^{-}_{2} \). And the part \( |\{\{\overline{x}\} \in F\}| \) is exactly equal to \( \sum_{C \in F^{-}_{1}} |C| \). Therefore,

\[
|F^+| = |\{\{\overline{x}\} \in F\}| + \sum_{C \in F^-} |C| \\
= 2 \sum_{C \in F^{-}_{1}} |C| + \sum_{C \in F^{-}_{2}} |C| \\
\geq 2 \sum_{C \in F^-} 1 = 2|F^-|.
\]

Hence \( |F^-| \leq |F|/3 \), and the assignment \( \alpha = (1, \ldots, 1) \) satisfies at least \( \frac{2}{3} \) of all clauses of \( F \).

And this bound is tight, which can be showed by the formula

\[
\{\{x\}, \{y\}, \{\overline{x}, \overline{y}\}\}.
\] (2.1)

\[\square\]

2.3 3-satisfiable CNF formula

A 3-satisfiable contains neither empty clause, a pair of complementary 1-clauses nor a subset in form \( \{\{x\}, \{y\}, \{\overline{x}, \overline{y}\}\} \). Yannakakis [3] proved lower
2.3. 3-satisfiable CNF formula

bounds of \( r_2 \) and \( r_3 \) by a simple probabilistic method and by considering
the conditional probability we can get a simpler algorithm to find a proper
assignment.

The idea of the proof is to construct a random assignment \( \alpha \) for a 3-satisfiable
CNF formula such that the expected weight satisfied by \( \alpha \) in \( F \) is at least \( r_3 \)
fraction. Then there exists a deterministic assignment can also satisfy at
least \( r_3 \) of the total weight because a random assignment is a probability
distribution on all possible assignment.

Käppeli and Scheder [4] showed that the bound is tight by a family of for-
mulas.

**Theorem 2.4** \( r_3 = \frac{2}{3} \).

**Proof** Here we first show that \( r_3 \geq \frac{2}{3} \) by the probabilistic method, and we
will see this bound is tight in the following section. For \( r_2 \) and \( r_3 \), we fix
some \( p, 0 \leq p \leq 1 \) and choose a random truth assignment \( \alpha \) with

\[
\Pr[\alpha(x) = 1] = \begin{cases} 
  p, & \text{if } \{x\} \in F, \\
  1 - p, & \text{if } \{\overline{x}\} \in F, \\
  \frac{1}{2}, & \text{otherwise},
\end{cases}
\]

independently for each variable. Since the considered formula will not con-
tain a pair \( \{x\}, \{\overline{x}\} \) of complementary 1-clauses, the probability distribution
is well-defined.

For \( F \) a 3-satisfiable CNF formula, we choose \( p = \frac{2}{3} \) and consider the satis-
fied probability of each clause

- Each 1-clause \( \{l\} \) in \( F \) is satisfied by the random assignment with the
  probability exact \( p = \frac{2}{3} \) because \( \Pr[l = 1] = p \).
- For each 2-clause \( C = \{l_1, l_2\} \), since formula \( F \) is 3-satisfiable, there can
  not be both clauses \( \overline{l_1} \) and \( \overline{l_2} \) in \( F \) at the same time. Therefore, the
  clause \( C \) is satisfied by the random assignment with the probability at
  least \( 1 - \frac{1}{2} \cdot \frac{2}{3} = \frac{2}{3} \).
- Obviously, each \( (> 2) \)-clause is satisfied with the probability at least
  \( 1 - (\frac{2}{3})^3 \geq \frac{2}{3} \).

Therefore, the expected value of the fraction of clauses satisfied by the ran-
don assignment is at least \( \frac{2}{3} \). Since a random assignment is a probability
distribution on all assignment, there exists a deterministic assignment that
satisfies at least \( \frac{2}{3} \) of weight. Thus,

\[ r_3 \geq \frac{2}{3} \quad (2.2) \]

\[ \square \]
2. Previous Results

Similarly, we can use the same method to bound $r_2$ with a proper choosing of the value of $p$. For $F$ 2-satisfiable, we choose $p = \frac{\sqrt{5} - 1}{2}$ and observe that each clause is satisfied with probability at least $p$. So the expected value of the fraction of satisfied clauses is also at least $p$. That is, the lower bound of $r_2$ is at least $\frac{\sqrt{5} - 1}{2}$.

In [4], Käppeli and Scheder introduced a proof idea of $r_3 \leq \frac{2}{3}$, but they did not provide a complete proof. In this part, we first present a complete construction of a 3-satisfiable formula $\text{SAT}_3(n)$. Then we show that the bound $\frac{2}{3}$ of $r_3$ is asymptotically tight.

The formula $\text{SAT}_3(n)$ for $n$ even is defined as follows: it has $n$ variables $x_i$, $1 \leq i \leq n$, and $2^n$ variables $y_{UV}$ where $UV$ is one of $2^n$ partitions on $X := \{x_1, \ldots, x_n\}$, i.e., $X = U \uplus V$. The clauses of the formula $\text{SAT}_3(n)$ are the following:

- $n$ clauses $\{x_i\}$ for $1 \leq i \leq n$ each of weight $\frac{1}{n}$.
- $2^n$ sets of clauses $G_{UV} := \{\{\overline{x}, y_{UV}\} | u \in U\} \cup \{\{x, \overline{y}_{UV}\} | v \in V\}$. Each of these $n2^n$ clauses has the weight equal to $2/\binom{n}{2}$, i.e., the sum of their weights is equal to 2.

Note that the total weight of $\text{SAT}_3(n)$ is equal to 3. Now we state two lemmas on formulas $\text{SAT}_3(n)$. We first show that it is always 3-satisfiable such that we can use it to bound $r_3$.

**Lemma 2.5** The formula $\text{SAT}_3(n)$ is 3-satisfiable for all $n$ even.

**Proof** Let $W$ be a minimal subformula of a formula $\text{SAT}_3(n)$, which cannot be simultaneously satisfied. By the minimality of $W$, any variable either does not appear in the clauses of $W$ at all or it has both a positive and a negative occurrence in the clauses of $W$, if not, i.e., a variable just have positive literals in $W$, then we let $W[\overline{x} \rightarrow 1]$ be a smaller subset in which all clauses cannot be simultaneously satisfied, which is a contradiction to the precondition that the size of $W$ is minimum. We can use this fact to prove that the size of $W$ should be greater than 3.

Assume for the sake of contradiction that $|W| \leq 3$. Since $W$ cannot be satisfied, it contain at least one 1-clause $\{x_k\}$ for some $1 \leq k \leq n$. Therefore $W$ should also have a clause containing the negation of $x_k$, i.e., clause $\{\overline{x}_k, y_{UV}\}$ (or clause $\{\overline{x}_k, \overline{y}_{UV}\}$) for some $UV$. Then $W$ should have a clause containing the negation of $y_{UV}$, i.e., clause $\{\overline{x}_l, \overline{y}_{UV}\}$. Because a variable appear in one partition just once, we know that $x_k \neq x_l$. At last, the variable $x_l$ appear in $W$ just with its negative occurrence, which is contradiction to the minimality of $W$. \qed
Then we show that every assignment satisfy almost same fraction of the weight in the formula $SAT_3(n)$.

**Lemma 2.6** The maximum weight of the satisfied clauses of a formula $SAT_3(n)$ is equal to $2 + O(n^{-\frac{1}{4}})$.

**Proof** Consider a fixed assignment which set $k$ of the variables $x_i$, $1 \leq i \leq n$ to true. We deal with each type of clause contained in $SAT_3(n)$ separately and see the satisfied probability:

- $\{x_i\}$ for $1 \leq i \leq n$

  There are exactly $k$ clauses satisfied. Hence the weight of the satisfied clauses of this type is exactly $\frac{k}{n}$, note that the total weight of this type is 1.

- Clauses in all $G_{UV}$,

  Choose a random partition $UV$ such that for each $x_i \in X$, variable $x_i$ is in the part $U$ with probability $\frac{1}{2}$. Let $T$ be the number of variables of $U$ which are set to be true by the considered assignment. Note that $T$ is a random variable and it follows a binomial distribution with the expected value equal to $\frac{k}{2}$.

  To satisfy a maximum number of clauses in $G_{UV}$, we set $y_{UV}$ to 1 if $T \geq \frac{k}{2}$ and 0 otherwise. For the random partition $UV$, let $E_{UV}[unsatisfied]$ be the expected value of the number of unsatisfied clauses in $G_{UV}$.

  Then for $0 < \delta < 1$ we have

  \[
  E_{UV}(unsatisfied) = \sum_{t=0}^{k} \Pr[T = t] \min(t, k - t) \geq \sum_{|t-\frac{k}{2}| < \delta \frac{k}{2}} \Pr[T = t] \min(t, k - t) \geq (1 - \delta)^{\frac{k}{2}} \sum_{|t-\frac{k}{2}| < \delta \frac{k}{2}} \Pr[T = t].
  \]

  Chernoff bounds can be used to bound the probability

  \[
  \sum_{|t-\frac{k}{2}| < \delta \frac{k}{2}} \Pr[T = t] = \Pr_{UV} \left[ \left| T - \frac{k}{2} \right| < \delta \frac{k}{2} \right] \geq 1 - e^{-\delta^2 k/6} - e^{-\delta^2 k/4}.
  \]

  We set $\delta = k^{-1/3}$ and observe that

  \[
  e^{k^{\frac{3}{2}}/6} \geq 2k^{\frac{1}{2}} \quad \Rightarrow \quad 1 - 2e^{-k^{\frac{3}{2}}/6} \geq 1 - k^{-\frac{1}{3}} \quad \Rightarrow \quad 1 - e^{-k^{\frac{3}{2}}/6} - e^{-k^{\frac{3}{2}}/4} \geq 1 - k^{-\frac{1}{3}} \quad \Rightarrow \quad E_{UV}(unsatisfied) \geq (1 - k^{-\frac{1}{3}}) 2 \frac{k}{2}
  \]
Therefore the weight of the satisfied clauses of this type is
\[ 2^n \cdot \frac{2}{n^{2n}} \cdot (n - \text{EUV(unsatisfied)}) \leq 2 - \frac{k}{n}(1 - k^{-\frac{3}{2}})^2. \]

Now the total weight of the satisfied clauses in \( \text{SAT}_3(n) \) is
\[ w := \frac{k}{n} + 2 - \frac{k}{n}(1 - k^{-\frac{3}{2}})^2. \]

Suppose that \( k \leq n^{\frac{3}{4}} \), it is easy to see that the total weight \( w = 2 + O(n^{-\frac{1}{4}}) \).
Otherwise, i.e., \( k > n^{\frac{3}{4}} \), the total weight of the satisfied clauses
\[ w = 2 + \frac{k}{n} - \frac{k}{n} + O(k^{-\frac{3}{2}}) = 2 + O(n^{-\frac{1}{4}}). \]

These two lemmas lead us to show that the bound of \( r_3 \) is tight.

**Theorem 2.7** For each \( \varepsilon > 0 \), there exists a 3-satisfiable formula \( F \) with weight function \( \mu \) such that \( \mu^*(F) \leq \frac{2}{3} \mu(F) + \varepsilon. \)

**Proof** From the Lemma 2.5 and the Lemma 2.6, we can choose the formula \( F := \text{SAT}_3(n) \) for a sufficiently large number \( n \) such that the error term \( O(n^{-1/4}) \) from the Lemma 2.6 is smaller than the considered \( \varepsilon \). Therefore, the maximum total weight of satisfied clauses of \( \text{SAT}_3(n) \) is greater than \( \frac{2}{3} \mu(F) + \varepsilon. \)

Note that Theorem 2.7 indicate that the bound \( r_3 \geq 2/3 \) is tight.

To the best of our knowledge, it is still an open question to determine the exact value of \( s_3 \). Kääpeli and Scheder [4] bounded \( s_3 \) by a probabilistic method with introducing dependence among variables. The idea of the proof is using two different random assignment \( \alpha \) and \( \beta \), where these two random assignment will not have poor performance at the same time. So we can use the mix of \( \alpha \) and \( \beta \) to show the lower bound of \( s_3 \).

**Theorem 2.8** (Kääpeli and Scheder) \( \frac{21}{31} \leq s_3 \leq \frac{7}{10}. \)

**Proof** To show that \( s_3 \leq \frac{7}{10} \), consider the following formulas:
\[ \{ \{a\}, \{b\}, \{c\}, \{\bar{a}, w\}, \{\bar{b}, w\}, \{\bar{c}, \bar{w}\}, \{\bar{d}, \bar{w}\}, \{\bar{a}, \bar{b}, \bar{w}\}, \{\bar{c}, \bar{d}, \bar{w}\} \}. \]

This formula is 3-satisfiable and has 10 clauses, but no assignment satisfies more than 7 clauses. Note that the assignment \((a, b, c, d, w) \mapsto (1, 1, 1, 0, 1)\)
To prove the lower bound, let $F$ by any 3-satisfiable formula. We can assume $F$ contains only positive unit clauses (if $\{x\}$, switch $x$). Let $X := \{x \in V \mid \{x\} \in F\}$. Write $F = F_1 \uplus F_2$ with

$$F_1 := \{\{x\} \in F\} \cup \{\{x, l\} \in F \mid x \in X, \text{vbl}(l) \in V \setminus X\}$$

and $F_2 := F \setminus F_1$. We will define two random assignment $\alpha$ and $\beta$.

Define $\alpha$ as follow: the variable $x \in X$ is set to be true with the probability $\frac{2}{3}$ and the variable $u \in V \setminus X$ is set to be true with the probability $\frac{1}{2}$. It is not difficult to see that $\Pr[\alpha \text{satisfies } C] = \frac{2}{3}$ for $C \in F_1$ and $\Pr[\alpha \text{satisfies } D] \geq \frac{19}{27}$ for $D \in F_2$.

Next, we define the second random assignment $\beta$. If $x \in X$ and $\overline{x}$ does not occur in $F_1$, set $\Pr[\beta(x) = 1] = \frac{3}{4}$. Otherwise, there is a clause $C_x = \{\overline{x}, l\} \in F_1$. For each clause $C_x = \{\overline{x}, l\}$ we call $\text{vbl}(l)$ the master variable of $x$ and $C_x$ the master clause of $x$. Then for each $u \in V \setminus X$ set the probability $\Pr[B(u) = 1] = 1/2$ independently. Depending on $u \in V \setminus X$, we set each variable $x$ with a master $u$ such that its master clauses $\{\overline{x}, u\}$ is satisfied, i.e., if $u \mapsto 0$ we set $x \mapsto 0$ otherwise we set $x \mapsto 1$. Note that we introduce a dependency between variables when we construct $\beta$, the “random assignment” $\beta$ is not perfect meet our definition of random assignment because it assign value to variables not independently. But this dependency does not effect the bound of expected value following.

Note that unit clauses $\{x\}$, where $x$ having a master variable $u$, are satisfied with probability 1/2 since $\Pr[u = 1] = 1/2$, and master clauses are satisfied with probability 1. Each remaining 2-clause is satisfied with probability $3/4$ (here we use that if $\{\overline{x}, l\}$ is the master clause of $x$, $\{\overline{x}, l\} \notin F$, because 3-satisfiability). A clause in $F_2$ is satisfied with probability $\geq 1/2$. Hence, the expected number of satisfied clauses is $3/4|F_1| + 1/2|F_2|$. We summarize:

$$\mathbb{E}[\mu^{[\alpha]}(F)] \geq \frac{2}{3}|F_1| + \frac{19}{27}|F_2|,$$

$$\mathbb{E}[\mu^{[\beta]}(F)] \geq \frac{3}{4}|F_1| + \frac{1}{2}|F_2|.$$

Then with probability $p = 27/31$, choose $\alpha$, and with probability $4/31$, choose $\beta$. The expected number of satisfied clauses is $21/31|F|$. We obtain

$$s_3 \geq \frac{21}{31} \tag{2.3}$$
Using a more sophisticated approach, they derived a better lower bound of $s_3 \geq 57/82$.

### 2.4 4-satisfiable CNF formula

For the 4-satisfiable CNF formulas, the concrete value of $r_4$ is a bit complicated. And no non-trivial bound for $s_4$ has been found.

Král [5] provided a sophisticated probabilistic argument of the lower bound of $r_4$, then showed it is tight by a certain 4-satisfiable formula. The main idea of the proof is dividing clauses to different type and using the random assignment to discuss the satisfied probability of each type of clauses.

**Theorem 2.9 (Král)** \[ r_4 = \frac{3}{5 + \sqrt[3]{\sqrt{69} - 11} - \sqrt[3]{\sqrt{69} + 11}}. \]

**Proof** We first define two useful constant $p_0$ and $q_0$ and they will be used when choosing the random assignment.

Let $q_0$ be the unique real solution of the following equation $q_0^3 = (1 - q_0)^2$, and define $p_0 := \sqrt{q_0}$, such that

\[ q_0 = \sqrt[3]{\frac{3\sqrt{69} + 11}{2}} - \sqrt[3]{\frac{3\sqrt{69} - 11}{2}} + \frac{1}{3} \approx 0.57 \]

and $p_0 \approx 0.755$.

Then we first show the lower bound of $r_4$:

\[ r_4 \geq \frac{1}{2 - p_0^2} = \frac{3}{5 + \sqrt[3]{\sqrt{69} - 11} - \sqrt[3]{\sqrt{69} + 11}} \approx 0.6992. \]

Consider a 4-satisfiable formula $F$ on variables set $V$ and a weight function $\mu$. If $F$ contains no clauses with length 1, we can choose a random assignment $\alpha$ in which each variable is set to be true independently with probability $\frac{1}{2}$. Therefore the expected fraction of the weight satisfied by $\alpha$ is at least

\[ \frac{1}{2 - p_0^2} = \frac{3}{5 + \sqrt[3]{\sqrt{69} - 11} - \sqrt[3]{\sqrt{69} + 11}} \approx 0.6992. \]

So we assume that $F$ contains clauses of length 1 and all the occurrences of variables in 1-clauses are positive (if $\{x\} \in F$, switch $x$).

Now let $X$ be the set of variables which occur in some 1-clauses, i.e., $X = \{x \in V \mid \{x\} \in F\}$. Then if there exists some clause $\{x, y\}$ for $x \in X$, we switch $y$; this process terminates after a finite number of steps, otherwise
there exists clauses \{x, y\}, \{x\}, \{y, z\}, \{z\} in the 4-satisfiable CNF formula \(F\).

Let \(Y\) be the set of variable \(y\) which occur in some clauses of type \{\(x, y\)\} for \(x \in X\) and which are not contained in \(X\), i.e., \(Y = \{y \in V \setminus X \mid \{x, y\} \in F, x \in X\}\).

Finally, let \(Z\) be the set of the remaining variables contained in \(F\), i.e., \(Z = V \setminus (X \cup Y)\).

Then we define four types of clauses:

- \(A_1\): clauses containing only literals of the form \(x\) for \(x \in X\), i.e., \(\{x_1, x_2, \ldots\}\);
- \(A_2\): clauses containing a literal \(x\) for \(x \in X\) and a literal \(y\) for \(y \in Y\) but no literal \(x'\) for \(x' \in X\), i.e., \(\{x, y, \ldots\}\);
- \(A_3\): clauses containing only literals of the form \(x\) for \(x \in X\) and literals of the form \(y\) for \(y \in Y\), i.e., \(\{x_1, \ldots, x_t, y_1, \ldots, y_s\}\);
- \(A_4\): clauses containing only literals of the form \(y\) for \(y \in Y\) and literals with the variables of \(Z\).

Formalize the total weights of all clauses containing literal \(x \in X\) by multiplying all weights of clauses a suitable constant. Denote \(\sigma\) the total weight of all clauses in \(A_1\), \(A_2\), \(A_3\) and \(A_4\). Let’s consider the following probability distribution: Set each variable of \(X\) to be true with the probability \(p_0\), each variable of \(Y\) to be true with the probability \(q_0\) and each variable of \(Z\) to be true with the probability \(1/2\).

Now if \(\sigma \leq 1/p_0^3\), then the expected weight of the satisfied clauses with this distribution is at least \(\mu(F)\), this can be showed by discussing the satisfied probability of each type of clauses. For example a clause of type 1 is satisfied with probability at least \(1 - p_0^4\) because it consists of at least four literals (4-satisfiable) \(\overline{x}\) for \(x \in X\), and a clause of type 2 is satisfied with probability at least \(1 - p_0(1 - q_0) = 1 - p_0^4\). After some similar case analysis, we find that a clause of type \(A_i\) are satisfied with probability at least \(1 - p_0^i\), the clause containing literal \(x \in X\) is satisfied with probability \(p_0\) and the other clause is satisfied with probability at least \(1/(2 - p_0^3)\). So we just need to show

\[
\frac{1 \cdot p_0 + \sigma \cdot (1 - p_0^4)}{1 + \sigma} \geq \frac{1}{2 - p_0^3}
\]

from the condition \(\sigma \leq 1/p_0^3\).

If \(\sigma > 1/p_0^3\), then we consider a new probability distribution which set all variables of \(X\) and \(Y\) to false with probability 1 and variables of \(Z\) to
be true with probability $\frac{1}{2}$. Either this new distribution or the distribution introduced in the previous satisfies at least $\frac{\mu(F)}{2-P_0}$ of the total weight in the formula $F$. Here, some details of the computation and cases analysis (just like the situation $\sigma \leq 1/p_0^3$) are omitted.

Hence we get that $r_4 \geq \frac{1}{2-P_0}$.

Then we show that this bound is tight by a sequence of 4-satisfiable CNF formulas.

We define formula $SAT_4(n, \alpha, \beta, \gamma)$ as follows: It has $n$ variables $x_i, 1 \leq i \leq n$, and $n^k$ variables $y_{\vec{a}}$ where $\vec{a}$ ranges over all ordered $k$-tuples of numbers $1, ..., n$ for $k = \lfloor n^{1/3} \rfloor$. We say that two ordered $k$-tuples $\vec{a}$ and $\vec{b}$ have a common entry if there is $i$ which is an entry both of $\vec{a}$ and $\vec{b}$. The clauses of the formula $SAT_4(n, \alpha, \beta, \gamma)$ are the following:

- $n$ clauses $\{x_i\}$ for $1 \leq i \leq n$ each of weight $1/n$;
- $kn^k$ clauses $\{x_i, y_{\vec{a}}\}$ for all pairs $i$ and $\vec{a}$ such that $i$ is contained in $\vec{a}$, each with weight $\alpha/(kn^k)$. Note that if $i$ is contained in $\vec{a}$ several times, then the formula contains as many clauses $\{x_i, y_{\vec{a}}\}$ as the number of occurrences of $i$ in $\vec{a}$;
- clauses $\{y_{\vec{a}}, y_{\vec{b}}\}$ for all ordered pairs of $k$-tuples $\vec{a}$ and $\vec{b}$ which do not have a common entry. Each of this type of clauses has same weight and the sum of all is equal to $\beta$;
- $\binom{n}{4}$ clauses $\{\overline{x}_{i_1}, \overline{x}_{i_2}, \overline{x}_{i_3}, \overline{x}_{i_4}\}$, each with weight $\gamma/\binom{n}{4}$.

It is easy to see that such a formula $SAT_4(n, \alpha, \beta, \gamma)$ is 4-satisfiable for all possible parameters. Then consider a fixed assignment which sets $p$ fraction of variables $x_i$ to be true and $q$ fraction of variables $y_{\vec{a}}$ to be true (since the formula $SAT_4(n, \alpha, \beta, \gamma)$ is symmetric, we don’t care about the value of specified variable). Now we see how much weight is satisfied in each type of clauses by this assignment:

- $\{x_i\}$
  There are $pn$ clauses in this type are satisfied, the total satisfied weight is exactly $p$.
- $\{x_i, y_{\vec{a}}\}$
  Let $\tau(\vec{a})$ be the number of entries of $\vec{a}$ which correspond to the variables $x_i$ set to be true by the assignment. Then a Chernoff bound can be used to bound the number of ordered $k$-tuples $\vec{a}$ for which
2.4. 4-satisfiable CNF formula

\[ \tau(\overrightarrow{a}) \] differs significantly from the "average value" \( kp \):

\[
\Pr[\tau(\overrightarrow{a}) \geq (1 + \lambda)pk] \leq e^{-\lambda^2 pk/3} n^k,
\]
\[
\Pr[\tau(\overrightarrow{a}) \leq (1 - \lambda)pk] \leq e^{-\lambda^2 pk/2} n^k.
\]

Then set \( \lambda = n^{1/4} / pk \geq n^{-1/12} \). Observe that except for at most \( e^{-\Theta(n^{1/6})} n^k \) number of tuples \( \overrightarrow{a} \), all tuples \( \overrightarrow{a} \) satisfy that the number of their entries \( i \) with correspond to the variables \( x_i \) set to be true is with difference of \( n^{1/4} \) from \( pk \). Therefore there fraction

\[
(1 - p + I(n^{-1/12})) k
\]

of clauses with form \( \{x_i, y_{\overrightarrow{\alpha}}\} \) are satisfied for the variable \( y_{\overrightarrow{\alpha}} \) false. Hence, the ratio of the satisfied clauses of the type \( \{x_i, y_{\overrightarrow{\alpha}}\} \) is equal to

\[
q + (1 - q)(1 - p + O(n^{-1/12})) + e^{-\Theta(n^{-1/6})} = 1 - p + pq + O(n^{-1/12}).
\]

- \( \{y_{\overrightarrow{\alpha}}, y_{\overrightarrow{\beta}}\} \)

Consider all ordered pairs of literals \( y_{\overrightarrow{\alpha}} \) and \( y_{\overrightarrow{\beta}} \), exactly \( 1 - q^2 \) of them contain a satisfied literal. However only some of them correspond to real clauses. Note that the number of clauses of this type is at least \( n^{k} (n - k)^{k} \) and the number of all ordered pairs of literals \( y_{\overrightarrow{\alpha}} \) and \( y_{\overrightarrow{\beta}} \) is \( n^{2k} \). Hence the error make by approximating the ratio of the satisfied clauses by \( 1 - q^2 \) is of order \( 1 - (n-k)^k = O(n^{-1/3}) \).

- \( \binom{n}{4} \) clauses \( \{x_{i_1}, x_{i_2}, x_{i_3}, x_{i_4}\} \)

Consider all ordered quadruples of literals \( x_i \); exactly \( (1 - p^4)n^4 \) of them contain a satisfied literal. The number of ordered quadruples corresponding to the clauses of the formula is \( 4!(\binom{n}{4}) \). Hence at most

\[
n^4 - n(n - 1)(n - 2)(n - 3) = \Theta(n^3)
\]

of the quadruples do not correspond to the clauses and the fraction of the satisfied clauses of the type is

\[
1 - p^4 + O(n^{-1}).
\]

So, for all possible \( n, \alpha, \beta, \gamma \), the total weight of satisfied clauses of a formula \( SAT_4(n, \alpha, \beta, \gamma) \) is equal to

\[
f = p + \alpha(1 - p + pq + O(n^{-1/12})) + \beta(1 - q^2 + O(n^{-1/3})) + \gamma(1 - p^4 + O(n^{-1})).
\]
This result lead us to computing the value of $r_4$.

For each $\varepsilon > 0$, we want to show that there exists a 4-satisfiable CNF formula $SAT_4(n, \alpha, \beta, \gamma)$ with a weight function $\mu$ such that $\mu^*(SAT_4(n, \alpha, \beta, \gamma)) \leq \mu(SAT_4(n, \alpha, \beta, \gamma))/(2 - p_0^2) + \varepsilon$.

Set $\alpha$, $\beta$ and $\gamma$ to be the unique solution of the following system of equations:

\begin{align*}
\alpha p_0^4 + \beta p_0^4 + \gamma p_0^4 - p_0 &= 0 \quad (2.4) \\
1 - \alpha + \alpha q_0 - 4\gamma p_0^3 &= 0 \quad (2.5) \\
ap_0 - 2\beta q_0 &= 0. \quad (2.6)
\end{align*}

We choose a sufficiently large number $n$ such that the error term $\alpha O(n^{-1/12}) + \beta O(n^{-1/3}) + \gamma O(n^{-1})$ from the result $f$ is smaller than the considered $\varepsilon$.

Then consider the first partial derivatives according to $p$ and $q$ of the rest part of function $f$:

\begin{align*}
\frac{\partial}{\partial p} (p + \alpha (1 - p + pq) + \beta (1 - q^2 + \gamma (1 - p^4))) &= 1 - \alpha + \alpha q - 4\gamma p^3 \\
\frac{\partial}{\partial q} (p + \alpha (1 - p + pq) + \beta (1 - q^2 + \gamma (1 - p^4))) &= ap - 2\beta q.
\end{align*}

From the equation 2.5 and 2.6 and the derivative result, we see that $f$ as the function of variable $p$ and $q$ attains its maximum $(\alpha + \beta + \gamma)$ when $p = p_0$ and $q = q_0$. Then using the equation 2.4, we get that:

\begin{align*}
\frac{\mu^*(SAT_4(n, \alpha, \beta, \gamma))}{\mu(SAT_4(n, \alpha, \beta, \gamma))} &\leq \frac{\alpha + \beta + \gamma}{1 + \alpha + \beta + \gamma + \varepsilon} = \frac{1}{1 + p_0^2} + \varepsilon, \quad (2.7) \\
 &\leq \frac{1}{2 - p_0^2} + \varepsilon. \quad (2.8)
\end{align*}

\section{2.5 k-satisfiable CNF formula}

Lieberherr and Specker [1] conjectured that $\lim_{k \to \infty} r_k = 1$, which was disproved by Huang and Lieberherr in [9], who showed that $\lim_{k \to \infty} r_k \leq 3/4$. Trevisan [6] proved that $\lim_{k \to \infty} r_k \geq 3/4$ by a probabilistic method which use the idea of ranking to attribute probabilities to the variables. These were later improved by Král[5].

**Theorem 2.10 (Trevisan)** For any $r$ such that $1/2 < r < 3/4$, a $k$ exists (depending on $r$) such that for any $k$-satisfiable formula $F$ we can find in polynomial time a probability distribution over the variables in such a way that any clauses is satisfied with probability at least $r$. 


Before the proof we define some notion for the probability distribution.

**Definition 2.11** For any real r \( \neq 0 \), we define the sequence \( \{ a_i^r \}_{i \geq 1} \) as follows:

1. \( a_1^r = r \);
2. \( a_{i+1}^r = 1 - (1 - r)/a_i^r \).

Because the equation \( x = 1 - (1 - r)/x \) has no real root when \( r < 3/4 \), if we start from a number \( r < 3/4 \), the sequence eventually goes below 1/2.

**Definition 2.12** Given a CNF formula \( F \),

1. If \( \{ l \} \in F \) then \( l \) is 1-provable in \( F \).
2. If \( \{ l_1, ..., l_h \} \in F \) and \( l_j \) is \( i_j \)-provable in \( F \) for \( j = 1, ..., h - 1 \), then \( l_h \) is \( (1 + \max\{ i_1, ..., i_{h-1} \}) \)-provable in \( F \).

In fact, each literal \( l \) in \( F \) may appear in different clauses and it is different \( i \)-provable from each clause. To eliminate the ambiguity, for each literal \( l \) we just choose the smallest integer \( i \) such that \( l \) is \( i \)-provable in \( F \).

The notion of provability is used to describe the dependence among different literals. We state a lemma about the “provable” and “\( k \)-satisfiable”

**Lemma 2.13** Let \( F \) be a formula with clauses of length at most 4. If \( x \) is \( i \)-provable in \( F \) and \( x \) is \( j \)-provable in \( F \), then \( F \) is not \( (3^{i+1} + 3^{j+1} - 2) \)-satisfiable.

**Proof** Since the length the clauses in \( F \) is at most 4, at most 3 of \( (i - 1) \)-provable literals can be in the same clause \( C \) with \( l \). We can use the induction to show that: for a literal \( l \) is \( i \)-provable in \( F \), then a set \( S_l \) of at most \( 3^{i+1} - 1 \) clauses of \( F \) exists such that any assignment that satisfies all the clauses in \( S_l \) must also satisfy \( l \).

Therefore, the set \( S_x \cup S_{\bar{x}} \) has at most \( (3^{i+1} + 3^{j+1} - 2) \) clauses, and no assignment can satisfy all of them. That is, \( F \) is not \( (3^{i+1} + 3^{j+1} - 2) \)-satisfiable. \( \Box \)

In this lemma, the motivation of the condition that the length of clauses in \( F \) is at most 4 is that a clause with length 5 is satisfied with probability at least \( 1 - (3/4)^5 > 3/4 \) if we assign each variable true with probability \( r \) between 1/2 and 3/4.

Having the definition and the lemma, now we start the proof of the theorem 2.7:

**Proof** We use the random assignment \( \alpha \) to show this theorem, that is, we assign a probability \( p_x \) to variable \( x \) such that \( \Pr[\alpha(x) = 1] = p_x \). The probability \( p_x \) for each \( x \) should be between \( r \) and \( 1 - r \). So we don’t need to consider any \( (\geq 5) \)-clauses in \( F \) because such clause is satisfied by \( \alpha \) with
probability at least $1 - (3/4)^5 > 3/4$. We only need to care about unary, binary, ternary and 4-ary clauses, so denote $F_4$ the subset of $F$ containing all clauses with length at more 4.

Let $k = 2 \cdot 3^{h(r) + 1} + 1$ where $h(r)$ is the smallest integer such that $a'_{h(r)} < 1/2$. Then for a literal $l$, if $l$ is $i$-provable in $F$ for $i < h(r)$, its complement $\overline{l}$ cannot be $j$-provable for $j < h(r)$ because here $(3^{i+1} + 3^{i+1} - 2) < 3^{2h(r)} - 1 = k$, contradiction to that $F$ is $k$-satisfiable.

Define a sequence of probability $p_1, ..., p_{h(r) - 1}, 1/2$ (here the subscript is integer rather than variable). Let $p_i = a'_i$ for $1 \leq i \leq h(r) - 1$ and $p_{h(r)} = 1/2$. Then the probability distribution is as follows.

$$
\Pr[x = 1] = \begin{cases} 
p_i, & \text{if } x \text{ is exactly } i\text{-provable in } F_4, \text{ for } i \leq h(r) - 1, \\
1 - p_i, & \text{if } \overline{x} \text{ is exactly } i\text{-provable in } F_4, \text{ for } i \leq h(r) - 1, \\
\frac{1}{2}, & \text{otherwise.}
\end{cases}
$$

Note that for a $i$-provable literal $x$ in $F_4$, the larger is $i$, the smaller is $\Pr[x = 1]$.

Then let’s consider the satisfaction probability of each clause in $F_4$. First for a 1-clause, it is easy to see that the satisfied probability is at least $p_1 = r$.

Then denote a clause $C = \{l_1, ..., l_k\}$ with 2 or more (but less than 5) literals. Without losing of generality, we assume that

$$
\Pr[l_1 = 0] \leq ... \leq \Pr[l_k].
$$

We consider two cases:

- If $\Pr[l_2 = 0] \leq 1/2$, just considering the first two literals, the satisfied probability of $C$ is at least

$$
1 - \Pr[l_1 = 0] \Pr[l_2 = 0] \geq 1 - \left(\frac{1}{2}\right)^2 > r.
$$

If $\Pr[l_2 = 0] > 1/2$, assume that $\overline{l_1}$ is $i_2$-provable in $F_4$. According to the definition, literal $l_1$ should be at most $i_1$-provable where $i_1 \leq i_2 + 1$. Therefore $\Pr[l_1 = 0] = 1 - p_{i_1} = (1 - r)/a_{i_1 - 1}$ and $\Pr[l_2 = 0] = p_{i_2} = a_{i_2} \leq a_{i_1 - 1}$. Thus,

$$
\Pr[C \text{ is satisfied}] \geq 1 - \Pr[l_1 = l_2 = 0] \geq 1 - (1 - r)/a_{i_1 - 1} \cdot a_{i_1 - 1} = r.
$$

At last, all of the clauses are satisfied by the random assignment with the probability at least $r$. □
2.6 Summary

From the proofs in this section, we can see that the random assignment is useful when we study the problem of $k$-satisfiable CNF formula. Most of the proofs in this section use the technic that classify the variables to different type according to the clauses they belong to. And assign true probability to each variable according to their type.

This idea work well when $k$ is small, because there is not much dependence among variables. Therefore we can use a simple probability to bound the value of $r_2$ and $r_3$. But when we want to deal with the 4-satisfiable CNF-formulas, it become much more complicated. We have to analyze many cases and different type of clauses and variables.

In the proof of the theorem 2.9, we saw idea of provability which is a generalized method to classify different literals. We use this notion to build the literal-hierarchy such that we can discuss the case for a general $k$. 
Chapter 3

Algorithm Implementation

In this section, we discuss the algorithm for $k$-satisfiable CNF formula to find the assignment satisfying at least $r_k$ fraction of the total weight. In fact, most of the proofs in last section are constructive, so we can use the idea of the proof to implement the algorithm.

3.1 Algorithm for the probabilistic method

In the proof of the Theorem 2.4, we have seen a probabilistic method which uses a random assignment to show the bound of $r_2$ and $r_3$. The key idea of this method is that the expected fraction is at least $r_k$ implying that there exists a deterministic assignment can satisfy also at least $r_k$. So here we use this idea to implement an algorithm, that is, we use the technique of derandomization via conditional probabilities. The input of this algorithm is a 2-satisfiable CNF formula $F$ on variables set $V$ and its weight function $\mu$. The output of this algorithm is an assignment $\alpha$ which satisfies at least $r_2 = \frac{\sqrt{5} - 1}{2}$ of the total weight in $F$.

Here is the description of this algorithm:

- **Step 0:** Maintain a probability table for each variable of $F$ using the probability distribution in the proof of Theorem 2.4.

- **Step 1:** Choose a variable $x \in V$; calculate the sum of weight of all clauses containing $x$, denoted by $u_0$, and the sum of weight of all clauses containing $\overline{x}$, denoted by $u_1$; let $F_0 := F[x \rightarrow 0], F_1 := F[\overline{x} \rightarrow 1]$ and $V := V \setminus \{x\}$.

- **Step 2:** Use the probability table to calculate the sum of the expected satisfied weight for all clauses in $F_0$, denoted by $w_0$; and the sum of the expected satisfied weight for all clauses in $F_1$, denoted by $w_1$. 


3. Algorithm Implementation

- Step 3: If $u_0 + w_0 > u_1 + w_1$, add $x \mapsto 0$ to the output assignment $\alpha$ and let $F := F_0$; otherwise, add $x \mapsto 1$ to the output assignment $\alpha$ and let $F := F_1$. Go to step 1 if $|V| \geq 1$.

- Step 4: Output assignment $\alpha$.

This algorithm can output an assignment satisfying at least $r_2$ of the total weight because in every loop we choose the value for $x$ to keep the remaining expected satisfied weight still at least $r_2$ of the total weight by the conditional probability. In each step, the running time is $O(n)$ and the number of loops is also $n$ where $n$ is the size of variables set $V$. Therefore the total running time of this algorithm is $O(n^2)$.

This algorithm is correct because of the following inequality:

$$r_2 \cdot \mu(F) < E_{a}[\mu[\alpha](F)] = (w_0 + u_0) \Pr[x \mapsto 0] + (w_1 + u_1) \Pr[x \mapsto 1] < \max\{w_0 + u_0, w_1 + u_1\}.$$ 

Also, it is easy to adapt this algorithm to find a proper assignment for a 3-satisfiable CNF formula by using the probability distribution for 2-satisfiable CNF formulas.

Note that, in the proof of the Theorem 2.9, we also used two random assignments. So we can also use the derandomization to implement an algorithm to find an assignment satisfying at least $r_4$ of weight in 4-satisfiable CNF formula. We use the derandomization algorithm to two random assignments separately to get two deterministic assignment then return the better one.

3.2 Algorithm ENUMERATE

We use the symmetrization idea from the proof of Theorem 2.2 to implement an algorithm which can find a proper assignment for 2-satisfiable CNF formulas. However the proof is not polynomially constructive, since we used the full-permutation group for symmetrizing. Note that the size of full-permutation is $n!$. In fact, there exist some doubly transitive permutation groups with "small" size which can be used to symmetrize in our algorithm. In next paragraph, we describe a double transitive permutation group $\text{DT}[p]$ whose size is polynomial.

For a prime $p$, the permutation group $\text{DT}[p]$ contains $p(p-1)$ permutations. Each permutation in $\text{DT}[p]$ has two parameter $q$ and $r$ where $1 \leq q \leq p$ and $1 \leq r \leq p-1$. And a variable $x$ maps to $x'$ in the permutation $\text{DT}[p](q, r)$ iff

$$x = qx' + r \pmod{p}.$$
This group was introduced and proved to be doubly transitive permutation group in the 67th chapter of [7].

Then we describe a set $A(n)$ of assignment on the variables set $V$ where $|V| = n$. Set

$$A(n) = \bigcup_{k=0}^{p} \bigcup_{q=1}^{p-1} \bigcup_{r=1}^{p} a(n, k, q, r),$$

where $p$ is the first prime greater than or equal to $n$ and the assignment $a(n, k, q, r)$ defined as follow: each variable $x_i \in V$ is set to be true iff

$$q \cdot i + r \leq k \pmod{p}.$$ 

That is, a variable $x_i$ is set to be true by $a(n, k, q, r)$ if it is in the first $k$ variables after the permutation $DT[p](q, r)$. And the parameter $k$ means that there are exactly $k$ variables are set to be true in the assignment $a(n, k, q, r)$.

The following algorithm, called $ENUMERATE$, construct an assignment satisfying at least $r_2$ of the weight in 2-satisfiable clause $F$.

1. For each variable $x \in V$, switch $x$ in there exists some $C \in F$ and $x \in C$.
2. Compute the first prime $p$ greater than or equal to $n$.
3. Enumerate the set $A(n)$ of assignment of $V$, and choose an assignment which satisfies the maximal weight in $F$.

This algorithm is a polynomial algorithm in the total size of clauses of $F$ because the size of the enumerated assignment set $A(n)$ is

$$p(p-1)(p+1) < p^3,$$

and $p$ will not be greater than $2n$. We can find such $p$ in time $O(n^{3/2})$ by the Postulate of Bertrand [8]. Thus the overall time of this algorithm is polynomial.

Each assignment in the set $A(n)$ corresponds to a permutation $\Pi$ of $DT[p]$ and each permutation correspond a formula $\Pi[F_1]$ which is used to construct the symmetric formula $F_2$ in the proof of Theorem 2.2. And since there exists an assignment satisfying at least $r_2$ of weight in $F_2$, a corresponding assignment in $A(n)$ can satisfy at least $r_2$ of weight in $F$. That is, the algorithm can return correctly.
Chapter 4

The Hardness of $r_k$ Problem.

In this section, we discuss the hardness of a sequence of $> r_k$ problems. The $> r_k$ problem is to determine for a $k$-satisfiable formula if there exists an assignment satisfying more than $r$ fraction of weight where $r > r_k$. In [1], Lieberherr first raised a conjecture about the unweighted version of this problem and gave a proof idea for the 2-satisfiable case. In the following part, we will formally define a weighted version of the conjecture and show that this conjecture is hold for $k = 2, 3$.

**Conjecture 4.1** If $r = r_k + \varepsilon$ for $\varepsilon > 0$, the set of pairs $(F, \mu)$, where $\mu$ is the weight function on the $k$-satisfiable CNF formula $F$ which have an assignment satisfying at least $r\mu(F)$ weight, is NP-complete.

First we state a lemma as the reduce basis.

**Lemma 4.2** For $k$ constant, the set of $k$-satisfiable CNF formula which is globally satisfiable is NP-complete.

**Proof** Here we describe a reduction from the well-known NP-complete problem 3SAT to $k$-satisfiable and satisfiable CNF problem. That is, given a 3-CNf formula $F$, we can construct a $k$-satisfiable CNF formula $F'$ in polynomial time such that $F$ is satisfiable iff $F'$ is satisfiable. Assume that the size of the given 3-CNf formula should greater than $k$ to make the problem well defined.

In fact, we just need to check whether $F$ is $k$-satisfiable. If $F$ is $k$-satisfiable, we just let $F' := F$, otherwise let $F'$ be an arbitrary unsatisfiable $k$-satisfiable CNF formula.

There are $\binom{|F|}{k}$ number of subformulas of $F$ with size $k$, and for each of these subformulas there are at most $2^{3k}$ possible assignments. So we can check whether $F$ is $k$-satisfiable in $(\binom{|F|}{k})2^{3k}$ time, which is polynomial since $k$ is constant. □
Now we prove a lemma give us a process which can produced a 2-satisfiable CNF formula with a known maximum satisfied weight fraction \( r' < r \). The idea is to construct a sequence of 2-satisfiable CNF formulas whose maximum satisfied weight fraction can be calculated from the size \( n \) and that fraction converge to \( r_2 \). Then we can choose a proper \( n \) such that the fraction \( r' \) is less than the given \( r \).

**Lemma 4.3** For \( r = r_2 + \varepsilon \) where \( \varepsilon > 0 \), we can construct a 2-satisfiable CNF formula \( F \) with the weight function \( \mu \) and return \( \gamma' \) in time \( O(\varepsilon^{-2}) \), such that \( \mu^*(F) = r'\mu(F) \) for some \( r' < r \).

**Proof** We choose some variables set \( V \) with \( |V| = n \) and construct 2-satisfiable formula

\[
F := \{ \{ x \} \mid x \in V \} \cup \{ \{ x, \overline{y} \} \mid \{ x, y \} \in \binom{V}{2} \};
\]

and the weight function \( \mu \) assigns \( v = \frac{-1+\sqrt{5}}{2} n = \Phi n \) to every 1-clause and \( 1 \) to every 2-clause of \( F \). Note that the overall weight is \( nv + \binom{n}{2} \).

If we set \( k \) of variables in \( V \) to 1, then we can satisfy weight

\[
s(k) := kv + \binom{n}{2} - \binom{k}{2},
\]

which attains its maximum for \( k = v + \frac{1}{2} \) because \( s(k) \) is a quadratic function. Therefore no more than

\[
\max(s(v), s(v+1)) = \frac{v^2}{2} + \frac{v}{2} + \binom{n}{2}
\]

weight of \( F \) can be satisfied. That is,

\[
\frac{\mu^*(F)}{\mu(F)} = \frac{v^2 + v + n(n-1)}{2nv + n(n-1)} = \frac{\Phi^2 + 1 + \frac{\Phi-1}{n}}{2\Phi + 1 - \frac{1}{n}} > \frac{\Phi^2 + 1}{2\Phi + 1} = r_2,
\]

to make this fraction less than \( r \), we choose

\[
n := \left\lfloor \frac{r + \Phi - 1}{2\Phi r + r - \Phi^2 - 1} \right\rfloor + 1,
\]

note that this \( n \) is from the inequality \( \mu^*(f)/\mu(F) < r \). Since \( r_2 = \Phi \), we rewrite

\[
n = \left\lfloor \frac{\sqrt{5} - 2 + \varepsilon}{\sqrt{5} \varepsilon} \right\rfloor + 1 = O\left(\frac{1}{\varepsilon}\right).
\]

At last, we get the formula \( F \) over the variables set \( |V| = \left\lfloor \frac{r + \Phi - 1}{2\Phi r + \Phi - 1} \right\rfloor + 1 \) and the weight function \( \mu \) such that \( \mu^*(F) = r'\mu(F) \) for the ratio \( r' := \frac{\Phi^2 + 1 + \frac{\Phi-1}{n}}{2\Phi + 1 - \frac{1}{n}} \). Because the size of the formula is \( O(n + n^2) = O(\varepsilon^{-2}) \), this construction can finish in time \( O(\varepsilon^{-2}) \).
Now we prove the 2-satisfiable case of the conjecture using the lemma. The idea of the deduction is mixing two formulas with different maximum satisfied weight fraction to get a new formula with the maximum satisfied weight fraction exactly $r_2$.

**Theorem 4.4** For $r = r_2 + \varepsilon$, the set of pairs $(F, \mu)$, where $\mu$ is the weight function on the 2-satisfiable CNF formula $F$ which have an assignment satisfying at least $r\mu(F)$ weight, is NP-complete if $O(\varepsilon^{-2}) = \text{poly}(l)$ where $l$ is the total length of string $(F, \mu)$.

**Proof** Deduce from the Lemma 4.2, we just need to give a polynomial transformation $T$ which transforms a 2-satisfiable CNF $F'$ to a pair $T(F) = (F, \mu)$ where $F$ is 2-satisfiable CNF formula and $\mu$ is the weight function on $F$, such that $F'$ is satisfiable iff $F$ has an assignment satisfying at least the fraction $r$ of the total weight $\mu(F)$.

According to the Lemma 4.3, we can construct a 2-satisfiable CNF formula $F_0$ and the weight function $\mu_0$ on it in polynomial time since $O(\varepsilon^{-2}) = \text{poly}(l)$, such that $\mu_0^*(F_0) = r'\mu_0(F_0)$ for $r' < r$.

Then let $F$ contain $F_0$ concatenated with $F'$ and $\mu$ extend $\mu_0$ to formula $F'$ such that $\mu(C) = \mu_0(C)$ for all $C \in F_0$ and $\mu(F') = \frac{r - r'}{1 - r} \mu_0(F_0)$.

At last, the formula $F'$ is satisfiable iff we can satisfy the fraction
\[
\frac{\mu^*(F)}{\mu(F)} = \frac{\mu(F') + \mu^*(F_0)}{\mu(F)} = \frac{\frac{r - r'}{1 - r} \mu_0(F_0) + r' \mu_0(F_0)}{\frac{r - r'}{1 - r} \mu_0(F_0) + \mu_0(F_0)} = r
\]
of the total weight of $F$. \hfill \Box

In general, we can prove the $k$-satisfiable case of the conjecture 4.1 if we have two conditions:

- the set of $k$-satisfiable CNF formulas which is satisfiable is NP-complete;
- we can construct a $k$-satisfiable formula $F$ with weight function $\mu$ and return the fraction $r'$ "efficiently", such that $\mu^*(F')/\mu(F) = r$ for $r' < r$.

Then, similar to the Lemma 4.3, we construct a 3-satisfiable CNF formula using the same idea.

**Lemma 4.5** For $r = r_3 + \varepsilon$, we can construct a 3-satisfiable CNF formula $F$ with the weight function $\mu$ and return the fraction $r'$ in time $O(\varepsilon^{-4}2\varepsilon^{-4})$, such that $\mu^*(F) = r'\mu(F)$ for some $r' < r$.

**Proof** Let $F := \text{SAT}_3(n)$ which is introduced in the proof of theorem, and we just need to chose proper $n$ such that $\mu^*(F) = r'\mu(F)$ for some $r' < r$.

We have shown that at least $r_3 + cn^{-1/4}$ fraction of weight can be satisfied in $\text{SAT}_3(n)$ where $c$ is constant. So we can choose a large $n = O(1/\varepsilon^4)$ such
4. The Hardness of $> r_k$ Problem.

that $\mu^*(F) < r\mu(F)$. So the only problem is to calculate $r' = \frac{\mu^*(F)}{\mu(F)}$ for that $n$. Since $r$ is constant, the size $n$ is also constant. Assume that $k$ of variables $x_i$ are set to 1 in $SAT_3(n)$, the ratio

$$r' = \max_k \frac{\max_{a(k)} \mu^{[a(k)]}(F)}{\mu(F)} = \max_k \sum_{t=0}^{k} \binom{k}{t} \frac{1}{2^t} \min(t, k-t),$$

where $a(k)$ is an assignment which set exactly $k$ of $x_i$ to true. To get this $r'$ we need to calculate $n$ terms of $\sum_{t=0}^{k} \binom{k}{t} \frac{1}{2^t} \min(t, k-t)$ and find the maximum one, which need no more than $O(n^2)$ time. Note that the size of $SAT_3(n)$ is equal to $n(1 + 2^n) = O(n2^n)$. So at last, we construct a formula $F$ and weight function $\mu$ such that $\mu^*(F) = r'\mu(F)$ in time $O(n2^n) = O(\epsilon^{-4}2^{-4})$ and calculate $r'$ in time $O(n^2) = O(\epsilon^{-8})$.

\[\Box\]

**Theorem 4.6** For $r = r_3 + \epsilon$, the set of pairs $(F, \mu)$, where $\mu$ is the weight function on the 3-satisfiable CNF formula $F$ having an assignment which satisfies at least $r\mu(F)$ weight, is NP-complete if $O(\epsilon^{-4}2^{-4}) = \text{poly}(l)$ where $l$ is the total length of the string $(F, \mu)$.

**Proof** Use the same method in the proof of Theorem 4.4 and deduce from Lemma 4.2 and Lemma 4.5, this theorem follows. \[\Box\]
Bibliography


# Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor’s thesis, Master’s thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

**Title of work** (in block letters):

**Authored by** (in block letters):

*For papers written by groups the names of all authors are required.*

<table>
<thead>
<tr>
<th>Name(s):</th>
<th>First name(s):</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

With my signature I confirm that
- I have committed none of the forms of plagiarism described in the ‘Citation etiquette’ information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

**Place, date**

<table>
<thead>
<tr>
<th>Signature(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

*For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.*