Random \((\log n)\)-CNF are Hard for Cutting Planes (Again)

Dmitry Sokolov

École Polytechnique Fédérale de Lausanne

June 8, 2023

Abstract

The random \(\Delta\)-CNF model is one of the most important distribution over \(\Delta\)-SAT instances. It is closely connected to various areas of computer science, statistical physics, and is a benchmark for satisfiability algorithms. Fleming, Pankratov, Pitassi, and Robere [Fle+22] and independently Hrubeš and Pudlák [HP17] showed that when \(\Delta = \Theta(\log n)\), any Cutting Planes proof for random \(\Delta\)-CNF on \(n\) variables requires size \(2^{\frac{n}{\log^c n}}\) in the regime where the number of clauses guarantees that the formula is unsatisfiable with high probability. In this paper we show tight lower bound \(2^{\Omega(n)}\) on size CP-proofs for random \((\log n)\)-CNF formulas. Moreover, our proof is much simpler and self-contained in contrast with previous results based on Jukna’s lower bound for monotone circuits.

1 Introduction

Proof complexity studies whether there are efficient certificates (or proofs) for the unsatisfiability of boolean formulas. The non-existence of such proofs in any proof system would separate classes \(\text{NP}\) and \(\text{coNP}\). According to Cook’s program, the idea is to prove lower bounds for stronger and stronger proof systems, so eventually, we would be able to do it in a general case.

At the current moment we do not have explicit candidates of hard families of unsatisfiable formulas for all proof systems. And the important problem here is that explicit unsatisfiable formulas are usually accompanied by mathematical reasoning of unsatisfiability, and these reasonings one may translate into formal proofs in some strong enough proof system. However, the situation is different in the case of distribution over formulas that are unsatisfiable with high probability. Candidates of this form are actively studied [CS88, Gri01, Bea+02, AR03, Ale+04, FKO06, Ale11, MT14, Raz13, HP17, Ats+18, Sok20, SS22, Fle+22]. And one of the most popular candidates distribution that generates hard formulas for all proof systems are random \(\Delta\)-CNF formulas.

**Definition 1.1**

Let \(\mathfrak{F}(m, n, \Delta)\) denote the distribution of random \(\Delta\)-CNF on \(n\) variables obtained by sampling \(m\) clauses (out of \(\binom{n}{\Delta}\) possible clauses) uniformly at random with repetitions.

The famous result of Chvátal–Szemerédi says if we pick a formula from with distribution with proper parameters the resulting formula will be unsatisfiable with high probability.

**Theorem 1.2** [Chvátal–Szemerédi, [CS88]]

For any \(\Delta \geq 3\) whp \(\varphi \sim \mathfrak{F}(m, n, \Delta)\) is unsatisfiable if \(m \geq \ln 2 \cdot 2^{\Delta n}\).

Formal conjectures were formulated by Feige [Fei02]: no polynomial time algorithm may prove whp the unsatisfiability of a random \(O(1)\)-CNF formula with arbitrary large constant clause density. Assuming Feige’s conjecture it is known that some problems are hard to approximate: vertex covering, DNF PAC learning, etc.
Related results. The state of art results in proof complexity are presented in figure 1. We know some lower bounds for proof systems in shaded region. The second line indicates proof systems for which we know also lower bounds for random $\Delta$-CNF formulas. In particular, we know lower bounds for random $\Delta$-CNF formulas in:

- Resolution (Chvátal–Szemerédi, [CS88]);
- Polynomial Calculus (Ben-Sasson, Impagliazzo [BI99]);
- Sum-of-Squares (Grigoriev, [Gri01]).

The notable exception is an $AC_0$-Frege proof system for which current techniques require extremely structured formulas and do not allow to deal with random CNFs.

There are two proof systems on the frontier: Cutting Planes and Res$[k]$ ($k$-DNF Resolution). Exponential lower bounds on Res$[k]$-proofs for random CNF formulas was given by Atserias, Bonet and Esteban [ABE02], the result was significantly improved by Segerlind, Buss, Impagliazzo [SBI04] and by Alekhnovich [Ale11].

Recently these results were unified and generalized by Sofronova and Sokolov [SS22]. To show lower bounds for $k \gg \sqrt{\log n}$ is an open problem. Lower bound for Cutting Planes was shown by Hrubeš and Pudlák [HP17] and independently by Fleming, Pankratov, Pitassi, and Robere [Fle+22]. We give an overview of the technique that was used in these papers.

1.1 Technique and Results

All current techniques for proving lower bounds on Cutting Planes proofs can be divided into two classes: interpolation and lifting.

Lifting works for structured formulas [Gar+18, Goo+20] and at the current moment this approach is not useful for random $\Delta$-CNFs and many other classes of formulas. We refer readers to papers [Gar+18] and [Lov+23]. At the same time we notice that proving lifting-like theorem for symmetric gadgets like inner product may also allow advance with random formulas.

Interpolation technique based on Craig’s Interpolation Theorem. In other words: let $\varphi(x, y, z) := A(x, z) \lor B(z, y)$ be an unsatisfiable CNF formula, then one can define a function $f(z)$ that for given assignment to variables $z$ says which formula $A$ or $B$ is unsatisfiable (we skip the question of what happens if both $A$ and $B$ are unsatisfiable for simplicity). Krajíček [Kra97] showed that if $\varphi(x, y, z)$ has monotone encoding and has an efficient proof in Cutting Planes with bounded coefficients then $f(z)$ can be computed.
by an efficient monotone circuit. And based on known lower bounds for monotone circuits Krajíček showed lower bounds for Cutting Planes with bounded coefficients. The restriction was removed by Pudlák [Pud97] who showed the first lower bound for the full version of Cutting Planes.

Random CNF formula $\varphi$ does not have the structure required by the interpolation technique, however, in both papers [HP17] and [Fle+22] authors suggested an adaptation of this technique. The general plan for proving lower bounds in these papers consists of the following steps.

1. Choose a monotone function $f_\varphi$ associated with the formula $\varphi$.
2. Show that if there is a small CP-proof for $\varphi$ then there is a small real monotone circuit for $f_\varphi$.
3. Use Jukna’s criteria to show that there are no small real monotone circuits for chosen $f_\varphi$.

There are several known ways how to associate a monotone function with a formula in a natural way, see for example [GP18]. In terms of communication complexity these methods are based on the fact that “monotone Karchmer–Wigderson relation is complete” (we refer readers to [RGR22] for more details). Usage of Jukna’s criteria in [HP17, Fle+22] requires a precise description of the $f_\varphi$ that requires some technical job and ideas. In these papers, authors showed that whp smallest CP-proof of random $O(\log n)$-formula has size $2^{\Omega(n)}$.

In this paper we show a much simpler proof of the stronger result.

**Theorem 1.3 [See also Theorem 4.2]**

There is a constant $c > 0$ such that if $\varphi \sim \mathcal{F}(m, n, \Delta)$ where $m = O(n^{2\Delta})$ and $\Delta \geq c \log n$, then whp every semantic CP-proof of $\varphi$ has size $2^{\Omega(n)}$.

To show this Theorem we modify the general plan from papers [HP17] and [Fle+22]. First, we notice, that all extractions of the function $f_\varphi$ utilize dag-like communication protocols as an intermediate constructions either in explicit [Fle+22] or implicit way [HP17, Pud97]. The notion of dag-like protocols formally was introduced by Sokolov in [Sok17] as a simplification of communication PLS games introduced by Razborov [Raz95] and simplified by Pudlák [Pud10] (the restricted version was independently introduced by Hrubeš and Pudlák [HP18]). Instead of extraction of function the $f_\varphi$ we stop at the intermediate step, create the dag-like communication protocol, and give a combinatorial analysis of this protocol. Hence the full plan is the following.

1. We use the lemma from [Sok17] and say that if there is a small CP-proof for $\varphi$ then there is a small real dag-like communication protocol for the Unsatisfied clause search problem (for the sake of completeness we give the proof in the Appendix A.4).
2. We use general idea of the bottleneck counting argument [HC99] (see also [Sok17]) and show that there is no small real dag-like communication protocol for the Unsatisfied clause search problem.

As an intermediate step inside bottleneck counting we introduce a 2-dimensional width measure for the Cutting Planes proofs. We believe that this measure is of independent interest.

**Remarks.** In fact, in Theorem 1.3 it is allowed to have slightly large clause density, namely $m = n^{1+\delta}2^\Delta$ for small enough $\delta$. But due to saving the simplicity of computations we do not try to reach the optimal parameters.

The notion of 2-width is related to the notion of fences from the papers [HC99]. But this connection passed through a reduction between unsatisfied clause search problem and monotone Karchmer–Wigderson relation.

**2 Preliminaries**

Denote by $H(x) := x \log x - \frac{1}{x} \log \frac{1}{x}$ the binary entropy function. We use the symbol $\uplus$ for the disjoint union.
With an unsatisfied CNF formula \( \varphi \) on variables of disjoint union of sets \( V_x \) and \( V_y \) we associate an unsatisfied clause search problem \( \text{Search}_\varphi \subseteq X \times Y \times \emptyset \) where:

- \( X \) is a set of assignments with support \( V_x \);
- \( Y \) is a set of assignments with support \( V_y \);
- \( \emptyset \) is a set of clauses of \( \varphi \);
- \( (x, y, o) \in \text{Search}_\varphi \) iff clause \( o \in \varphi \) is not satisfied by assignments \( x \) and \( y \).

Communication protocols and triangles. Consider a bipartite input domain \( X \times Y \). A triangle \( T \subseteq X \times Y \) is a set that can be written as \( T := \{(x, y) \in X \times Y \mid a_T(x) < b_T(y)\} \) for some labeling \( a_T : X \to \mathbb{R} \) of the set \( X \) and labelling \( b_T : Y \to \mathbb{R} \) of the set \( Y \) by real numbers.

For a triangle \( T \subseteq X \times Y \) and \( x \in X \) let \( T_x := \{(x, y) \in T \mid y \in Y\} \) be a horizontal cut and for \( y \in Y \) let \( T_y := \{(x, y) \in T \mid x \in X\} \) be a vertical cut.

A triangle-dag (aka real dag-like communication protocol) for a search problem \( S \subseteq X \times Y \times \emptyset \) is a directed acyclic graph \( H \) of fan-out at most 2 where each node \( h \) is associated with a triangle \( T_h \subseteq X \times Y \) satisfying the following:

- **root**: there is a distinguished root node \( r \) (fan-in 0), and \( T_r = X \times Y \);
- **non-leaves**: for each non-leaf node \( h \) with children \( u, u' \), we have \( T_h \subseteq T_u \cup T_{u'} \);
- **leaves**: each leaf node \( h \) is labeled with an output \( o_h \in \emptyset \) such that \( T_h \subseteq S^{-1}(o_h) \).

Expanders. We use the following notation: \( N_G(S) \) is the set of neighbours of the set of vertices \( S \) in the graph \( G \), i.e. the set \( \{v \in V \mid v \text{ share an edge with some } u \in S\} \) where \( V \) is the set of vertices of \( G \). We omit the index \( G \) if the graph is evident from the context.

A bipartite graph \( G := (L, R, E) \) is an \((r, \Delta, c)\)-expander if all vertices \( u \in L \) have degree at most \( \Delta \) and for all sets \( S \subseteq L, |S| \leq r \), it holds that \( |N(S)| \geq c \cdot |S| \).

Cutting Planes. We consider a semantic version of the Cutting Planes (CP) proof system \([\text{CCT87, Hru13}]\).

A proof in semantic CP for CNF formula \( \varphi \) is a sequence of linear inequalities with real coefficients \( C_1, C_2, \ldots, C_\ell \), such that \( C_\ell \) is the trivially unsatisfiable inequality \( 0 \geq 1 \) and \( C_i \) can be obtained by one of the following rules:

- \( C_i \) is a linear inequality that encodes a clause of formula \( \varphi \);
- \( C_i \) semantically follows on \( \{0, 1\} \) values from \( C_j \land C_k \) where \( j, k < i \).

The size of proof is the number of inequalities \( \ell \).

### 3 Formulas and Partitions

With a boolean formula \( \varphi \) we associate a dependency graph \( G := (U, V, E) \) in a natural way. There are two bijections: between clauses of \( \varphi \) and vertices from \( U \), and between variables of \( \varphi \) and vertices from \( V \). Edge \( (u, v) \in E \) iff variable \( v \) is appear in the clause \( u \).

A well-known fact is that dependency graph of a random CNF formula is an expander.

\[\text{Lemma 3.1}\]

Let \( \Delta := c \log n, m \leq \alpha n 2^\Delta \), for some constants \( \alpha > 0, c > 0 \). For any constant \( \varepsilon > 0 \) there is a constant \( \kappa > 0 \) such that whp for \( r := \kappa \cdot \frac{n}{\Delta} \) a dependency graph of \( \varphi \sim \mathcal{G}(m, n, \Delta) \) is an \((r, \Delta, (1 - \varepsilon)\Delta)\)-expander.

\[\text{Proof. For proof see Appendix A.3}\]

---

4
The following notion is of technical nature, but it will be useful for the main theorem. Let \( \varphi \) be a \( \Delta \)-CNF formula over boolean variables from a set \( Z \). We say that a partition \( Z \equiv V_x \uplus V_y \) is \( \delta \)-good if \( \varphi \) can be represented as \( \psi \land \psi_x \land \psi_y \) such that:

- \( |V_x| - |V_y| \leq 10\sqrt{|Z|} \);
- each clause \( C \in \psi \) contains at least \( \delta \Delta \) variables from \( V_x \) and at least \( \delta \Delta \) variables from \( V_y \);
- \( \Pr[\psi_x|\rho] = 1 \geq 0.9 \), where \( \rho \) is taken uniformly at random over all assignments with support \( V_x \);
- \( \Pr[\psi_y|\rho] = 1 \geq 0.9 \), where \( \rho \) is taken uniformly at random over all assignments with support \( V_y \).

**Lemma 3.2**

For every constants \( \alpha > 0, c > 1 \), if \( \varphi \sim \mathcal{F}(m, n, \Delta) \) where \( m = \alpha n^2 \Delta \) and \( \Delta \geq c \log n \) there exists \( \delta \)-good partition of variables \( V_x \uplus V_y \) of \( \varphi \) for any \( \delta \) such that \( c > (1 - \delta - H(\delta))^{-1} \).

**Proof.** For proof see Appendix A.3.

### 4 Main Theorem

In this section we show the main result. We start with a technical theorem.

**Theorem 4.1**

Let \( \varphi \) be a CNF formula over variables from a set \( Z \) where \( |Z| = n \). If dependency graph of \( \varphi \) is \( (r, \Delta, (1 - \delta/2)\Delta) \)-expander and there is \( \delta \)-good partition \( Z \equiv V_x \uplus V_y \) then any triangle-dag for Search\( \varphi \) has size at least \( 2^{k\Delta \delta/4 - 10\sqrt{n}} \) where \( k = \min(r, 2^{\Delta \delta/8}) \).

Here we may think that dependency graph of a given formula \( \varphi \) defines the expansion parameter and hence it defines \( \delta \). For fixed \( \delta \) we know how good the partition of variables should be. And parameter \( k \) depends on parameter of the partition.

We defer the proof of this Theorem to the Section 4.1. And start with the application to random formulas.

**Theorem 4.2 [Reformulation of Theorem 1.3]**

For constants \( \alpha > \ln 2 \) and \( c > 1 \), if \( \varphi \sim \mathcal{F}(m, n, \Delta) \) for \( m = \alpha n^2 \Delta \) and \( \Delta \geq c \log n \), then every semantic CP-proof of \( \varphi \) has size \( 2^{n\Omega(1)} \) whp over the choice of \( \varphi \). Moreover, if \( c > 800 \) then every semantic CP-proof of \( \varphi \) has size \( 2^{\Omega(n)} \) whp over the choice of \( \varphi \).

**Proof.** By Lemma A.3 instead of considering CP-proofs we show lower bound for triangle-dags for Search\( \varphi \).

Fix \( \delta > 0 \) such that \( c(1 - \delta - H(\delta)) > 1 \). Whp by Lemma 3.1 the dependency graph of \( \varphi \) is an \((r, \Delta, (1 - \delta/2)\Delta) \)-expander where \( r = \kappa \frac{n}{\Delta} \). By Lemma 3.2 there exists \( \delta \)-good partition of variables, hence the statement follows from Theorem 4.1 since \( \min(r, 2^{\Delta \delta/8}) = n^{\Omega(1)} \).

For the “moreover” part we fix \( \delta := \frac{1}{100} \). Note that \( c(1 - \delta - H(\delta)) > 1 \) and \( \min(r, 2^{\Delta \delta/8}) = r = \kappa \frac{n}{\Delta} \) for some constant \( \kappa > 0 \) by Lemma 3.1. Hence the statement follows from Theorem 4.1.  

Let us informally describe the bottleneck counting argument for proving Theorem 4.1.

1. Fix some triangle-dag \( H \) for Search\( \varphi \). Partition of variables gives a representation of \( \varphi \) as a conjunction \( \psi \land \psi_x \land \psi_y \). On clauses of \( \psi \) the partition is well-behaved, what cannot be said about clauses of \( \psi_x \) and \( \psi_y \). By using properties of good partitions we do a pruning step and get rid of assignments that do not satisfy \( \psi_x \) and \( \psi_y \). Since we deal with the unsatisfied clause search problem then we also can switch from \( \varphi \) to \( \psi \).
2. For each assignement \( z \) either with support \( V_y \) or with support \( V_y \) and each node \( h \in H \). We define a measure \( w(h, z) \) that we call 2-width such that if \( w(h, z) > k \) for some chosen threshold \( k \) then triangle \( T_h \) contain some useful information about \( z \). Informally speaking \( T_h \) contain useful information about \( z \) iff in the formula \( \psi \) restricted by \( z \) is still hard to find an unsatisfied clause even on set of assignments \( T_h^z \).

3. We show that for most of assignements \( z \) there should be some node \( h \) that contains useful information about \( z \).

4. At the same time we consider the bottomest node \( h \) such that \( T_h \) contain useful information about some \( z \). We show that \( T_h \) may contain useful information only about few assignmets \( z' \) (since it is the bottomest for \( z \)).

4.1 Proof of Theorem 4.1

4.1.1 Pruning

Let \( C' \) be a set of clauses appearing in \( \varphi \). Remind that we fix some \( \delta \)-good partition of variables \( V_x \cup V_y \). Let \( X' \) be a set of assignments with support \( V_x \) and \( Y' \) be a set of assignments with support \( V_y \).

Consider a triangle-dag \( H' \) for Search\( _{\varphi} \subseteq X' \times Y' \times C' \). We start by pruning the protocol and erasing all assignments with support \( V_x \) that do not satisfy \( \psi_x \) and all assignments with support \( V_y \) that not satisfy \( \psi_y \). To be more formal, let \( X \subseteq X' \) be a set of assignments with support \( V_x \) that satisfy \( \psi_x \) and \( Y \subseteq Y' \) be a set of assignments with support \( V_y \) that satisfy \( \psi_y \). We define a protocol \( H \) by taking \( H' \) and replacing each triangle \( T_h \) by a new triangle \( T_h \leftarrow T_h \cap (X \times Y) \). Note that \( H \) is a triangle-dag that solves Search\( _{\psi} \) on \( X \times Y \), and moreover all leaves are marked by clauses \( C \subseteq C' \) that correspond to \( \psi \) (since all other clauses are satisfied by any assignment from \( X \times Y \)). Hence \( H \) solves Search\( _{\psi} \) on \( X \times Y \). Note that by definition of a good partition

\[
|X| \geq 0.9 \cdot 2^{n-5\sqrt{n}} \geq 2^{n-6\sqrt{n}} \quad \text{and} \quad |Y| \geq 2^{n-6\sqrt{n}}.
\]

Denote by \( M := \{ R \subseteq X \times Y \mid R \text{ is a rectangle, } \exists C \in C', \forall (x, y) \in R, C(x, y) \text{ is unsat} \} \) the collection of monochromatic rectangles of Search\( _{\psi} \).

4.1.2 Formal Idea

Following the ideas of a bottleneck counting argument we define a partial map \( \mu : X \cup Y \to H \) such that:

- \( |\text{Dom}(\mu)| \geq \frac{1}{4} \min(|X|, |Y|) \geq 2^{n-10\sqrt{n}} \); \\
- for all \( h \in H \) : \( |\mu^{-1}(h)| \leq 2^{n-k\Delta\delta/4} \).

Hence size of the image of \( \mu \) (that is the size of \( H \)) is as desired.

We start with a definition of a 2-width complexity measure \( w : H \times (X \cup Y) \to \mathbb{N} \) that helps us to define the mapping \( \mu \). We define \( w \) as follows for all \( h \in H \) and all \( z \in X \cup Y \):

\[
w(h, z) := \min(M' \subseteq M \mid M' \text{ is a covering of } T_h^z).
\]

See the Figure 2 for the example.

**Remark 4.3**

Measure \( w \) is semi-additive w.r.t the second argument, i.e., for all \( z \in X \cup Y \), \( w(h, z) \leq w(h', z) + w(h'', z) \) where \( h', h'' \) are children of \( h \).

**Proof.** Note that \( T_h \subseteq T_{h'} \cup T_{h''} \) hence \( T_h^z \subseteq T_{h'}^z \cup T_{h''}^z \). Thus union of monochromatic coverings of \( T_{h'}^z \) and \( T_{h''}^z \) are also covering of \( T_h^z \) and the observation follows. \( \square \)
Now we describe the construction of $\mu$, see Algorithm 1. Informally, for each vertex $h$ in topological order starting from leaves we put all $z \in X \cup Y$ such that $w(h, z) > k$ and erase $z$ from the universe and repeat this process.

Algorithm 1 Definition of $\mu$

1: for $h \in H$ in topological order starting from leaves do
2: for $x \in X$ do
3: if $w(h, x) > k$ then
4: $\mu(x) := h$.
5: Erase the line $\{x\} \times Y$ from triangles $T_h$ for all $h \in H$.
6: for $y \in Y$ do
7: if $w(h, y) > k$ then
8: $\mu(y) := h$.
9: Erase the line $X \times \{y\}$ from triangles $T_h$ for all $h \in H$.

Remark 4.4

1. In the beginning of the iteration of Algorithm 1 in node $h$: for all $z \in X \cup Y$ it holds that $w(h, z) \leq 2k$.
2. In the end of the iteration of Algorithm 1 in node $h$: for all $z \in X \cup Y$ it holds that $w(h, z) \leq k$.

Proof. The second observation follows from the description of Algorithm 1 and the fact that after erasing any point from triangle $T_h$ the measure $w$ may only decrease.

The first observation follows from the Remark 4.3 and the fact that we process the node $h$ when we have already processed all the children of $h$.

To conclude the proof we show the required properties of $\mu$.

4.1.3 Size of Domain

In this section we show that $|\text{Dom}(\mu)| \geq \frac{1}{2} \min(|X|, |Y|)$. For the sake of contradiction assume that $|\text{Dom}(\mu)| < \frac{1}{2} \min(|X|, |Y|)$. In the remainder of the section we analyse the triangle protocol $H$ after the application of Algorithm 1.

Note that after the application of Algorithm 1 in the root $r$ of $H$ we are left with a triangle (that is also a rectangle) $T_r = X_r \times Y_r \subseteq X \times Y$ that consists of pairs $(x, y)$ such that $x \notin \text{Dom}(\mu)$ and $y \notin \text{Dom}(\mu)$. By our assumption $|X_r| > \frac{1}{2}|X|$ and $|Y_r| > \frac{1}{2}|Y|$. For any point $x_0 \in X_r$ it holds that $w(r, x_0) \leq k$ by Remark 4.4 and hence there are at most $k$ monochromatic rectangles that cover the line $T_r^{x_0} = \{x_0\} \times Y_r$. 

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{triangle_protocol.png}
\caption{“Width”-measure. The covering witnesses the fact $w(h, x_0) \leq 3$}
\end{figure}
We recall that monochromatic rectangles consist of points that violate some specific clause of $\psi$. Thus the fact that the line $T_x^r$ can be covered by at most $k$ monochromatic rectangles implies that there is a set $S$ of at most $k$ clauses of the formula $\psi$ such that any point $y \in Y_r$ does not satisfy at least one clause in $S$. At the same time, pick a point $y$ from $Y'$ uniformly at random:

\[
\Pr_{y \sim Y'}[y \in Y_r] \leq \Pr_{y \sim Y'}[y \text{ does not satisfy some clause } C \in S] \leq \sum_{C \in S} \Pr_{y \sim Y'}[y \text{ does not satisfy } C] \leq |S| \cdot \max_{C \in S} \Pr_{y \sim Y'}[y \text{ does not satisfy } C] \leq k \cdot 2^{-\delta\Delta},
\]

where the last inequality holds since each clause of $\psi$ has at least $\delta\Delta$ of $V_y$ variables. Hence

\[
|Y_r| \leq k \cdot 2^{-\delta\Delta} |Y'| \leq \frac{10}{9} 2^{\delta\Delta/8} \cdot 2^{-\delta\Delta} |Y| \leq \frac{10}{9} 2^{\frac{7}{8}\delta\Delta} |Y| \leq \frac{|Y|}{2}.
\]

This is a contradiction with the assumption.

### 4.1.4 Size of Preimage

In this section we show that for all $h \in H$: $|\mu^{-1}(h)| \leq 2^{n-k\Delta\delta/4}$.

Pick some vertex $h \in H$. We consider a situation in the beginning of the iteration of Algorithm\[ in node $h$ and fix this time moment for the rest of the section. By Remark 4.4 we know that for all $z \in X \cup Y$, $w(h, z) \leq 2k$. By using this fact we estimate the number of $x \in X$ that can be mapped into the vertex $h$ by $\mu$, or in other words, the number of $x \in X$ such that $w(h, x) > k$. In order to realize this we build a collection $\mathcal{P}$ of potential monochromatic coverings of size $k$ and count the number of $x \in X$ such that $T_h^x$ is not covered by any covering in our collection $\mathcal{P}$. By analogy the same counting holds for the set $Y$.

Let $X_0 \subseteq X$ be a set of points $x$ such that $T_h^x$ is not empty. The set $X \setminus X_0$ is not interesting for us since $w(h, x') = 0$ for all $x' \in X \setminus X_0$. Let $T_h = \{(x, y) \subseteq X \times Y \mid a_T(x) < b_T(y)\}$. We sort $x \in X$ wrt increasing order of $a_T(x)$ and we sort $y \in Y$ wrt to decreasing order of $b_T(x)$, see the resulting triangle in Figure\[. Now we are ready to build the potential monochromatic coverings. During this process we create an auxiliary tree $L$ whose vertices will be marked by subsets (subtriangles) of $T_h$ and all edges are marked by monochromatic rectangles. The collection $\mathcal{P}$ will correspond to the set of root-leaf paths in this tree. We start with the formal construction and give a description after.
Algorithm 2 Construction of $\mathcal{P}$

1: $L$ consists of a single node $r$ that is labelled by $T_h$.
2: $A := \{r\}$ set of active leaves of $L$.
3: while $A$ is not empty do
4:   Pick $a \in A$. Erase $a$ from $A$. Let $T \subseteq T_h$ be the label of $a$.
5:   Pick a first $y$ such that $T^y \neq \emptyset$.
6:   Let $M_1, M_2, M_3, \ldots, M_\ell \in M$ be the smallest covering of $T^y$.
7:   for $i \in [\ell]$ do
8:       Let $M_i = X_i \times Y_i$.
9:       Add $a_i$ in $L$ as a child of $a$. Mark the edge $(a, a_i)$ by $M_i$.
10:      Mark $a_i$ by $T_{a_i} := T \cap (X_i \times (Y \setminus Y_i))$.
11:     if the height of $a_i$ in $T$ is less than $k$ and $T_{a_i} \neq \emptyset$ then
12:         Add $a_i$ into $A$.

In this algorithm we start with a triangle $T_h$ and first (in our order) $y \in Y$. We start building our collection $\mathcal{P}$ by considering the smallest monochromatic covering $M_1, M_2, \ldots, M_\ell$ of the line $T_h^y$. For any $x \in X_0$ the line $T_h^x$ intersects with at least one rectangle $M_i$, or in other words $M_i$ covers part of $T_h^x$. We divide $X_0$ into $\ell$ parts (intersections are allowed) wrt which monochromatic rectangles intersects with $T_h^x$ and deal with each part independently. This division induces division of $T_h$ into $\ell$ subtriangles $T_{a_1}, \ldots, T_{a_\ell}$ such that $T_{a_i} := T_h \cap (X_i \times (Y \setminus Y_i))$, here we also erase parts that are already covered by $M_i$. Note that $\ell \leq 2k$ since $w(h, y) \leq 2k$. See an example of the first iteration of the Algorithm 2 in Figure 4. We repeat this process for each $T_{a_i}$ independently.

For each $x \in X_0$ we trace a path $P_x \subseteq L$ starting from the root in the natural way: suppose we reach node $a$ with a label $T$,

- if $T^x = \emptyset$ then stop;

Figure 4: First iteration of Algorithm 2
• if \( a \) is a leaf then stop;

• pick an edge \((a, a')\) marked by \( M = X_M \times Y_M \) such that \( x \in X_M \) (if there are more than one edge pick any). Note that such an edge is always guaranteed to exists since the collection \( M \), as defined in Algorithm 2, covers \( T^y \) where \( y \) is the smallest element of \( Y \) according to our order. Hence point \((x, y)\) is in \( T \) and it is also covered.

Note two properties of these constructed paths.

1. If an edge \((a, a')\) is marked by \( M \) is in \( P_x \) then the monochromatic rectangle \( M \) intersects with \( T^x_{h} \), or in other words \( x \) does not satisfy a clause of \( \psi \) that corresponds to \( M \).

2. If the length of \( P_x \) is less than \( k \) then \( T^x_{h} \) is covered by the monochromatic rectangles from edges of \( P_x \). Indeed, consider a node \( a \in P_x \) with label \( T \) and it is child \( a' \in P_x \) with label \( T' \). By construction \( T^x_{h} \setminus (T')^x \) is covered by monochromatic rectangle on the edge \((a, a')\), or say otherwise, while tracing \( P_x \) we step by step cut parts of \( T^x_{h} \) that are covered by monochromatic rectangles on edges of \( P_x \). But note that \((T')^x = \emptyset \) where \( T' \) is the label of the last vertex of \( P_x \) since the length of \( P_x \) is less than \( k \) and the desired property follows.

Let \( \mathcal{P} \) be the set of paths in \( L \) from root of size \( k \). A straightforward corollary from the second property is that \( w(x, h) \geq k \) implies \( P_x \in \mathcal{P} \). Hence the number of \( x \in X \) such that \( w(x, h) \geq k \) is

\[
\sum_{P \in \mathcal{P}} |\{x \in X \mid P \text{ is } P_x\}| \leq |\mathcal{P}| \cdot \max_{P \in \mathcal{P}} |\{x \in X \mid P \text{ is } P_x\}|
\]

By Remark 4.4 for any \( y \in Y \) there is a monochromatic covering of \( T^y_{h} \) of size at most \( 2k \). Hence the degree of the tree \( L \) is at most \( 2k \) and there are at most \( (2k)^k \) different paths in \( \mathcal{P} \). Fix some path \( P \in \mathcal{P} \). By the first property there is a set \( S \) of size \( k \) of clauses of the formula \( \psi \) such that if \( P = P_x \) for some \( x \) then \( x \) does not satisfy any clause from \( S \). Note that there is at most one assignment with support \( N(S) \cap V_x \) that does not satisfy all clauses from \( S \) hence there are at most \( 2^{n-|N(S) \cap V_x|} \) different points \( x \in X \) that do not satisfy any clause in \( S \).

\[
2^{n-|N(S) \cap V_x|} = 2^{n-|N(S)|} \leq 2^{n-(|N(S)|-(1-\delta)\Delta|S|)} \leq 2^{n-(1-\delta/2)\Delta|S|-(1-\delta)\Delta|S|} \leq 2^{n-|S|\Delta/2},
\]

We can use expansion property since \( |S| = k \) that is at most \( r \). Hence

\[
\max_{P \in \mathcal{P}} |\{x \in X \mid P \text{ is } P_x\}| \leq 2^{n-k\Delta/2}.
\]

Altogether, the number of \( x \in X \) such that \( w(x, h) \geq k \) is at most \( 2^{n-k\Delta/2+\log(k+1)} = 2^{n-k\Delta/2-(\log k+1)} \). An analogous counting argument shows that the number of \( y \in Y \) such that \( w(y, h) \geq k \) is bounded by \( 2^{n-k\Delta/2-(\log k+1)} \). Hence \( |\mu^{-1}(h)| \leq 2^{n-k(\Delta/2-(\log k+1))} \leq 2^{n-k(\Delta/2-2\log k)} \leq 2^{n-k\Delta/4} \) as needed.

Acknowledgments

I would like to thank Kilian Risse, Paul Beame, Nikola Galesi and others who forced me to write down this paper. Thank Kilian Risse for error correction. Part of this work was done while the author was visiting the Simons Institute for the Theory of Computing. This work was supported by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number MB22.00026.
References


A Properties of Random Formulas

A.1 Probabilistic Tools

**Theorem A.1 [Chernoff bound, [MU05]]**

Suppose $Z_1, \ldots, Z_n$ are independent random variables taking values in $\{0, 1\}$. Let $X$ denote their sum and let $\mu = E(X)$ denote the sum’s expected value. Then for any $0 < \delta \leq 1$ we have:

$$\Pr[|X - \mu| \geq \delta \mu] \leq \exp \left(-\frac{\delta^2 \mu}{3}\right).$$

A.2 Expansion of Random Formulas

For $m, n, \Delta \in \mathbb{N}$, we denote by $\Theta(m, n, \Delta)$ the distribution over bipartite graphs with disjoint vertex sets $U := \{u_1, \ldots, u_m\}$ and $V := \{v_1, \ldots, v_n\}$ where the neighbourhood of a vertex $u \in U$ is chosen by sampling.
a subset of size $\Delta$ uniformly at random from $V$.

**Theorem A.2**

Let $\Delta := c \log n, m \leq \alpha n 2^\Delta$, for some constants $\alpha > 0, c > 0$. For any constant $\varepsilon > 0$ there is a constant $\kappa > 0$ such that whp for $r := \frac{n}{\Delta}$ a randomly sampled graph $G \sim \mathcal{G}(m, n, \Delta)$ is an $(r, \Delta, (1-\varepsilon)\Delta)$-expander.

**Proof.** Let $\varepsilon < 1/2$. We estimate the probability that $G$ is not an $(r, \Delta, (1-\varepsilon)\Delta)$-expander for some parameter $r$. Let $G := (U, V, E)$. We first estimate the probability that a set $S \subseteq U$ of size at most $r$ violates the expansion. For brevity, let us write $s = |S|$ and $d = (1-\varepsilon)\Delta$. The probability that $S$ violates the expansion can be bounded by:

$$
\Pr[|N(S)| < ds] \leq n \cdot \left(\frac{ds}{\Delta}\right)^{s} \\
\leq n \cdot \left(\frac{ds}{n}\right)^{d} \\
\leq \left[\left(\frac{en}{ds}\right)^{d} \cdot \left(\frac{ds}{n}\right)^{\Delta}\right]^{s}
$$

Hence, the probability that $G$ is not an expander can be bounded by

$$
\Pr[G \text{ is not an expander}] \leq \sum_{s \in [r]} \left(\frac{m}{s}\right) s \cdot \left[\left(\frac{en}{ds}\right)^{d} \cdot \left(\frac{ds}{n}\right)^{\Delta}\right]^{s} \\
\leq \sum_{s \in [r]} \left(\frac{ms}{s}\right) s \cdot \left[\left(\frac{en}{ds}\right)^{d} \cdot \left(\frac{ds}{n}\right)^{\Delta}\right]^{s} \\
\leq \sum_{s \in [r]} \left[\left(\frac{en}{ds}\right)^{d} \cdot \left(\frac{ds}{n}\right)^{\Delta}\right]^{s} \\
\leq \sum_{s \in [r]} \left[e^{1+2\alpha n 2^\Delta (s \varepsilon^2)}\right]^{s} \\
\leq \sum_{s \in [r]} \left[\alpha (\frac{2c+1}{c}) \log n\right]^{s} \\
\leq \sum_{s \in [r]} \left[\alpha (2^{2c+1} \kappa^{c+1}) \log n\right]^{s}.
$$

And if $\kappa < 2^{-(3c+1)/\log c}$ this sum is $o(1)$.

**A.3 Proof of Lemma 3.2**

For every constants $\alpha > 0, c > 1$, if $\varphi \sim \mathcal{G}(m, n, \Delta)$ where $m = \alpha n 2^\Delta$ and $\Delta \geq c \log n$, then there exists $\delta$-good partition of variables $V_x \uplus V_y$ of $\varphi$ for any $\delta$ such that $c > (1 - \delta - H(\delta))^{-1}$. 

14
Proof. We add variables into $V_x$ with probability $\frac{1}{2}$ uniformly at random, and put all other variables into $V_y$ respectively. We show that with constant probability the partition is good.

Note that by Chernoff bound:

$$\Pr \left[ \left| |V_x| - \frac{n}{2} \right| \geq 5\sqrt{n} \right] \leq \exp \left( -\frac{(10/\sqrt{n})^2 \cdot (n/2)}{3} \right) \leq e^{-10}.$$  

Hence $|V_x| - |V_y| \leq 10\sqrt{n}$ with probability $1 - e^{-10}$.

Let $\psi_x \subseteq \varphi$ consists of all clauses that contain more than $(1 - \delta)\Delta$ variables from $V_x$, and $\psi_y \subseteq \varphi$ is defined by analogy. To show the remaining properties we show that size of $\psi_x$ and $\psi_y$ are not so big, and random assignment satisfy so small formulas with high probability. We analyze $\psi_x$ and by analogy the same holds for $\psi_y$.

Consider some clause $C \in \varphi$:

$$\Pr [C \text{ contains at most } (1 - \delta)\Delta \text{ variables from } V_x] = 2^{-H(\delta)}\Delta \sum_{i=0}^{\Delta} \left( \frac{\Delta}{i} \right) \leq 2^{-H(\delta)\Delta}.$$  

Hence by Markov inequality

$$\Pr \left[ |\psi_x| \geq 3m2^{(H(\delta)-1)\Delta} \right] = \Pr \left[ |\psi_x| \geq 3\alpha n2^{H(\delta)\Delta} \right] \leq \frac{1}{3}.$$

Now assume that $|\psi_x| \leq 3m2^{(H(\delta)-1)\Delta}$. Random assignment to variables $V_x$ does not satisfy some clause $C \in \psi_x$ with probability at most $2^{-2(1-\delta)\Delta}$ since it contains at least $(1 - \delta)\Delta$ variables from $V_x$. Hence

$$\Pr_{a \in \{0,1\}^{V_x}} [\exists C \in \psi_x : C(a) \text{ is not satisfied}] \leq 3\alpha n \frac{2^{H(\delta)\Delta}}{2^{2(1-\delta)\Delta}} = 3\alpha \frac{n}{2^{2(1-\delta-H(\delta))\Delta}} = 3\alpha \frac{n}{n^{1-\delta-H(\delta)c}} = o(1).$$

by the choice of $\delta$

Hence with probability $\frac{2}{3} - e^{-10} - o(1)$ random assignment is $\delta$-good.

A.4 From Cutting Planes to Communication Protocols

**Lemma A.3 [Sokolov, Sok17]**

Let $\varphi$ be an unsatisfiable CNF formula on variables $V_x \uplus V_y$, $X$ be a set of assignments to variables $V_x$ and $Y$ is a set of assignment to variables $V_y$. If there is a semantic CP-proof for $\varphi$ of size $S$ then there is a triangle-dag of size $S$ for $\text{Search}_\varphi$.

**Proof.** Let graph $H$ of the triangle dag be the graph of the semantic CP proof of the formula $\varphi$ with inverted edges. Consider a vertex $h \in H$, there is a proof line $f(V_x) + g(V_y) \geq c$ that corresponds to $h$. We associate with the node $h$ a triangle $T_h$ defined by labelling functions: $a_{T_h}(x) = f(u) - c$ and $b_{T_h}(y) = -g(y)$. Note that $(x, y) \in T_h$ iff $a_{T_h}(x) < b_{T_h}(y)$, hence $f(x) + g(y) < c$, i.e. the inequality is falsified by the assignment $(x, y)$.

The root $r$ of $H$ corresponds to the trivially false inequality $0 \geq 1$, hence the triangle $T_r = X \times Y$. If an assignment satisfies all inequalities in the children $h', h'' \in H$ of some vertex $h \in H$ then this assignment also satisfies the inequality in $h$. Thus, $T_h \subseteq T_{h'} \cup T_{h''}$.

By contruction in a leaf $h$ we have a triangle that consists of points that violate some clause $C \in \varphi$, we mark the leaf $h$ by $C$. 
