CS 2429 - Propositional Proof Complexity

Lecture #5: 10 October 2002

Lecturer: Toniann Pitassi

Scribe Notes by: Shiva Nejati

In the previous lecture, we introduced a lower bound on the length of Resolution proofs for the pigeonhole principle. We then talked about the width vs. size of tree-like resolution, and resolution proofs based on the width-size relationship.

In this lecture, we first show a lower bound on the length of Resolution proofs for random formulas. Intuitively, these formulas should be hard to prove because they simply have no structure that can be exploited to get a really short proof. We then bring up some open problems about the lower bound of Resolution proofs. Finally, we will talk about automatizability and proof search for Resolution.

1 Random k-CNF Formulas

Define a random distribution $F^k_{n,\Delta}$ on the set of k-CNF formulas in n variables by choosing $m = \Delta n$ clauses independently and uniformly from among the $2^k \binom{n}{k}$ clauses of length k. Δ here is *density* of the formula. For F chosen from this distribution we write $F \sim F^k_{n,\Delta}$. It is conjectured that the typical satisfiability properties of such random k-CNF formulas are determined by a sharp density threshold. Random formulas with density more than the threshold are asymptotically almost surely (a.a.s.) unsatisfiable, whereas those with density below the threshold are a.a.s. satisfiable. For example, for $k = 3$ it can be shown that if there are more than $5n$ clauses then a random 3-CNF is almost certainly unsatisfiable, and less than $2n$ clauses is almost certainly satisfiable, although the exact constant (which seems to be around 4.2 empirically) has not been determined. For random formulas whose density is not too large, we can show that any Resolution proofs of their unsatisfiability are almost surely super-polynomial, as stated more precisely in the following theorem:

Theorem 1 *For $F \sim F^k_{n,\Delta}$, almost certainly for any $\epsilon > 0$,*

1. *Any Davis-Putnam (DLL) proof of F requires size at least $2^{\frac{n}{\Delta^{2/(k-2)+\epsilon}}}$.*
2. *Any* resolution proof *of F requires size at least $2^{\frac{n}{\Delta^{4/(k-2)+\epsilon}}}$.*

This result implies that random k-CNF formulas are provably hard for the most common proof search procedures which are DLL type. In fact, this hardness extends well beyond the threshold. Even at density $\Delta = n^{1/3}$, current algorithms for random 3-CNF have qualitatively the same asymptotic complexity as the best known factoring algorithms, for example.

The proof of this theorem is based on properties of random hypergraphs. We associate each formula to a hypergraph. A k-CNF formula can be associated with hypergraphs in a natural way, where each variable becomes a vertex and each clause becomes a hyperedge. This mapping discards the distinction between a variable and its negation, but is sufficient for proving useful results.
In the previous lecture, we defined *sub-critical* expansion of a set of clauses F as follows:

$$e(F) = \max_{\frac{s(F)}{3} \leq s \leq \frac{2s(F)}{3}} \min \{|\delta G| : G \subseteq F, |G| = s\}$$

We showed that all boundary variables of a set of clauses G appear in a clause C^* such that $G \Rightarrow_P C^*$. We now define the *sub-critical expansion* of a hypergraph. Let F be a hypergraph. Denote by δF the boundary of F, which is the set of degree 1 vertices of F. The density of F is the ratio of the number of hyperedges to the number of vertices. We say that a subset of F has a system of distinct representatives (see Figure 1) iff with each hyperedge in F, we can associate a unique vertex (a representative) belonging to that hyperedge. In Figure 1, it is easy to see that the black nodes or representatives can independently be set to true or false in a way that makes the whole formula true. Let $s_H(F)$ be the size of minimum subset of F that does not have a system of distinct representatives. Define the subcritical expansion $e_H(F)$, where F is a hypergraph, as

$$e_H(F) = \max_{\frac{s_H(F)}{3} \leq s \leq \frac{2s_H(F)}{3}} \min \{|\delta G| : G \subseteq F, |G| = s\}$$

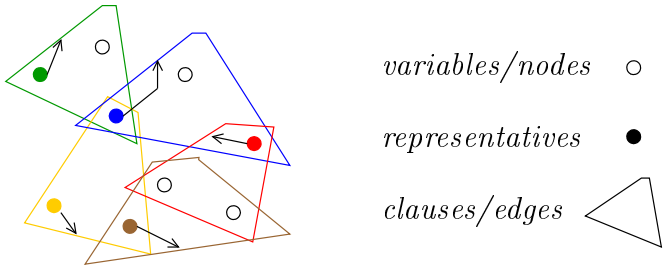


Figure 1: System of distinct representatives

Hall's Theorem allows us to get lower bounds on $s_H(F)$ for random hypergraphs:

Theorem 2 (Hall's Theorem) *A hypergraph F has a system of distinct representatives iff every subgraph of F has density at most 1.*

The expansion of a hypergraph F is also related to the densities of its subgraphs, so we can analyze both $s_H(F)$ and $e_H(F)$ by looking at the density of subgraphs of F.
Having defined $s(F)$ and $s_H(F)$, we can now compare them:

- $s(F)$: the size of the minimum subset of F that is unsatisfiable (F is an unsatisfiable k-CNF formula).
- $s_H(F)$: the size of the minimum subset of F that has no system of distinct representatives (F is the underlying hypergraph of a k-CNF formula).

It is easy to see that if the underlying hypergraph of a k-CNF formula has a system of distinct representatives, then the corresponding k-CNF formula is satisfiable and a satisfying assignment can be obtained by setting each representative to satisfy the clause which it represents. With this translation $s_H(F) \leq s(F)$ and $e_H(F) \leq e(F)$.

Lemma 3 *If $F \sim F^k_{n,\Delta}$, then almost certainly*

1. $s(F) = \Omega\left(\frac{n}{\Delta^{1/(k-2)}}\right)$, *and*
2. $e(F) = \Omega\left(\frac{n}{\Delta^{2/(k-2)+\epsilon}}\right)$ *for any $\epsilon > 0$.*

Proof The proof of this lemma is based on the fact that a k-uniform hypergraph of density bounded below $\frac{2}{k}$, say $\frac{2}{k} - \epsilon$, has average degree bounded below 2. Given a k-uniform hypergraph with density bounded below $\frac{2}{k}$, we first prove that a constant fraction of its nodes are in its boundary.

Assume that δ is the fraction of variables (nodes) that are not in the boundary, V is the number of variables (nodes), and E is the number of hyperedges (clauses). Using the fact that the average degree of this hypergraph is bounded below 2, and its density is bounded below $\frac{2}{k}$, we have:

$$\frac{E}{V} = (\frac{2}{k} - \epsilon)$$

$$2\delta V + (1-\delta)V \leq kE \leq k(\frac{2}{k} - \epsilon)V$$

Therefore,

$$\delta \leq 1 - k\epsilon$$

and,

$$k\epsilon \leq 1 - \delta$$

Thus, a constant fraction of variables (nodes) are in the boundary.

Fix a set S of vertices/variables of size r. The probability p that a single edge/clause lands in S is at most $(r/n)^k$. Therefore the probability that S contains at least q edges is at most

$$Pr[B(\Delta n, p) \geq q] \leq \left(\frac{e\Delta np}{q}\right)^q \leq \left(\frac{e\Delta r^{k-1}}{n^{k-1}}\right)^q$$

To get a bound on $s(F)$, we apply this for $q = r+1$ for all r up to s using union bound:

$$\begin{aligned} Pr[s(F) \leq s] &\leq \sum_{r=k}^{s} \binom{n}{r} \left(\frac{e\Delta r^{k-1}}{n^{k-1}}\right)^{r+1} \\ &\leq \sum_{r=k}^{s} \left(\frac{ne}{r}\right)^r \left(\frac{e\Delta r^{k-1}}{n^{k-1}}\right)^{r+1} \\ &\leq \sum_{r=k}^{s} \left(\frac{e^2\Delta r^{k-2}}{n^{k-2}}\right)^{r+1} \end{aligned}$$

This quantity is $o(1)$ in n for $s = O(n/\Delta^{1/(k-2)})$. In a similar way, we get a bound on $e(F)$ by summing the probability for $q = 2r/(k+\epsilon')$ for all r between $s/3$ and $2s/3$.

$$\begin{aligned} Pr[e(F) \leq s] &\leq \sum_{r=s/3}^{2s/3} \binom{n}{r} \left(\frac{e\Delta r^{k-1}}{n^{k-1}}\right)^{2r/(k+\epsilon')} \\ &\leq \sum_{r=s/3}^{2s/3} \left(\frac{ne}{r}\right)^{r} \left(\frac{e\Delta r^{k-1}}{n^{k-1}}\right)^{2r/(k+\epsilon')} \\ &\leq \sum_{r=s/3}^{2s/3} \left(\frac{e^{1+(k+\epsilon')/2}\Delta r^{k-1-(k+\epsilon')/2}}{n^{k-1-(k+\epsilon')/2}}\right)^{2r/(k+\epsilon')} \end{aligned}$$

This is $o(1)$ in n for $s = \Theta(n/\Delta^{2/(k-2-\epsilon')})$.

2 Open Problems

In this lecture and the previous lecture, we showed:

- For sufficiently large n, any Resolution refutation of PHP_n^{n+1} requires exponential size (i.e., $2^{n/20}$).
- For sufficiently large n, any Resolution refutation of formula F such that $F \in F_{n,\Delta}^k$ requires exponential size.

A problem which was open for a long time was to prove exponential lower bounds for the pigeonhole principle, PHP_n^m, where the number of pigeons, m, is large (say n^5). More precisely, find an ϵ such that:

$$\forall m \text{ any Resolution Proof of } PHP_n^m \text{ requires size of } 2^{o(n^\epsilon)}$$

When m is significantly larger than n, we have more clauses, and intuitively the formula is more unsatisfiable, therefore we might be able to get shorter Resolution proofs. Ran Raz has recently proven the above theorem for $\epsilon = 1/10$. His result was further improved and simplified by Razborov.

In the other direction, it is known that PHP_n^m has quasi-polynomial size depth 2 Frege proofs for $m \geq (1+\epsilon)n$. For $m = n + n/polylogn$, superpolynomial bounded-depth Frege lower bounds have been proven recently by (Beame, Buresh-Oppenheim, Pitassi, Raz and Sabharwal). But for $m = 2n$ it is it is not known whether polynomial-size AC^0-Frege proofs of PHP_n^m are possible.

3 Automatizability and Proof Search for Resolution

Bounding the size of proof systems is useful in relation to the goal of proving **NP≠coNP**. The proof system definition, however, does not say anything about how costly is to find a short proof in the given proof system. Whereas short proofs might exists, finding them may not be easy.

Here, we want to prove lower bounds on the hardness of finding short Resolution refutations of a given unsatisfiable CNF formula f. The problem can be stated precisely as the following optimization problem for any proof system P:

Definition [Optimization Problem associated with proof system P]
Minimum Length Proof (MLP_P)
Instance: A propositional formula f which is a tautology.
Solution: A P-proof of f.
Objective: Minimize the size of the proof.

Here, we are interested in finding Resolution refutations quickly. Therefore we consider the following problem:

Definition [Optimization Problem associated with Resolution]
Minimum Length Proof (MLP_{Res})
Instance: A propositional formula f which is unsatisfiable.
Solution: A *Res*-proof of f.
Objective: Minimize the size of the refutation.

We shall only discuss algorithms that are polynomial time in the size of the shortest proof of the input. A proof system P is *automatizable* if:

Definition A proof system P is automatizable if there is a polynomial-time algorithm that approximates MLP_P to within a polynomial factor.

In 1995, Samuel Buss proved that for a particular Frege system F_1, MLP_{F_1} is **NP**-hard. In 1997, Iwama proved that it is **NP**-hard to find the shortest Resolution refutation. In other words, Iwama proved that MLP_{Res} is **NP**-hard. In 1998, Alekhnovich, Buss, Moran, and Pitassi proved that:

- If **P** $\neq$**NP**, then there is no polytime approximation scheme for MLP_P, and
- If **NP** $\not\subseteq$ **QP** then there is no polytime algorithm to approximate MLP_P to within a factor of $2^{log^{1-\epsilon}n}$ for any ϵ.

Notice that P can be almost any proof system: Frege, Extended Frege, Sequent Calculs, Cut-free sequent calculus, Resolution, Polynomial calculus, ..., in tree-like or dag-like form.

In 2001, Alekhnovich and Razborov showed that neither general Resolution nor tree-like Resolution is automatizable unless the class **W[P]** from the hierarchy of parameterized problems is fixed-parameter tractable by randomized algorithms with one-side error. A less technical restatement of this result is that if Resolution is automatizable then we can solve the Clique problem (or a problem of the same class as the Clique problem) in time $f(k)n^{O(1)}$ instead of n^k.

We now introduce the Monotone Minimum Satisfying Assignment problem and discuss the relevant prior results about the hardness of approximating **NP**-optimization problems.

Definition
Monotone Minimum Satisfying Assignment (MMSA)
Instance: A monotone formula $\varphi(x_1, \ldots, x_n)$ over the basis $\{\wedge, \vee\}$.
Solution: A truth assignment τ such that $\varphi(\tau) = 1$.
Objective: Minimize the number of 1's in τ.

The Hardness Theorem for Resolution proof system is stated as follows:

Theorem 4 (Hardness Theorem)

a. *If* $\mathbf{P} \neq \mathbf{NP}$*, then there is no polynomial time algorithm which can approximate* MLP_{Res} *to within a constant factor.*

b. *If* $\mathbf{NP} \not\subseteq \mathbf{QP}$*, then there is no polynomial time algorithm which can approximate* MLP_{Res} *to within a factor of* $2^{log^{(1-\epsilon)}n}$ *for any* ϵ*.*

These hardness results apply to both dag-like and tree-like Resolution. To prove Hardness Theorem, we shall reduce the Monotone Minimum Satisfying Assignment problem (MMSA) to Minimum Length Resolution Refutation problems (MLP_{Res}). We give polynomial-time approximation-preserving reductions from MMSA to MLP.

Let $\varphi(x_1, ..., x_k)$ be an instance of MMSA, $|\varphi| = n$. Enumerate the subformulas of φ as $\varphi_1, \ldots, \varphi_l$, where the input variables are first in the enumeration and where each φ_i is listed only after all of its own subformulas are enumerated, and thus φ_l is φ. Obviously the number l of subformulas is less than the number of symbols n in φ. We introduce new propositional variables $y_1, ..., y_l$, and define the set Γ_φ to contain the following clauses:

a. The clause $\{\overline{y_l}\}$ is in Γ_φ.

b. For each $i \leq l$, if φ_i is $(\varphi_j \wedge \varphi_k)$, then the clause $\{\overline{y_j}, \overline{y_k}, y_i\}$ is in Γ_φ.

c. For each $i \leq l$, if φ_i is $(\varphi_j \vee \varphi_k)$, then clauses $\{\overline{y_j}, y_i\}$ and $\{\overline{y_k}, y_i\}$ are in Γ_φ.

The above clauses describe the evaluation of φ; however, note that they say nothing about the truth of the input variables $x_1 \ldots x_p$ of φ. For each variable x_i of φ, we introduce new variables $x_{i,j}$ for $j = 1, 2, \ldots, m$, and further include in Γ_φ the following clauses:

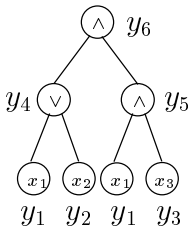
d. For each $i \leq p$, the clauses $\{x_{i,1}\}$ and $\{\overline{x_{i,m}}, y_i\}$ and

$$\{\overline{x_{i,j}}, x_{i,j+1}\} \quad for\ j = 1, \ldots, m-1$$

are included in Γ_φ. These clauses are said to be *associated* with x_i.

That completes the definition of Γ_φ. Informally, Γ_φ asserts that there exists a truth-evaluation to all subformulas of φ such that: the truth evaluation given to the output formula is 0, the truth evaluation given to all input variables is 1, and the truth evaluation is consistent with all intermediate gate values. Clearly, this is an unsatisfiable formula since φ is monotone.

φ :



Γ_φ :

$$
\begin{array}{ll}
 & (\bar{y}_6) \\
 & (\bar{y}_4\ \bar{y}_5\ y_6) \\
\rightarrow & (\bar{y}_1\ y_4)\quad (\bar{y}_2\ y_4) \\
\rightarrow & (\bar{y}_1\ \bar{y}_3\ y_5) \\
 & (x_{11})\ \ (\bar{x}_{11}\ x_{12})\ \ \ldots\ \ (\bar{x}_{1m}\ y_1) \\
 & (x_{21})\ \ (\bar{x}_{21}\ x_{22})\ \ \ldots\ \ (\bar{x}_{2m}\ y_2) \\
 & (x_{31})\ \ (\bar{x}_{31}\ x_{32})\ \ \ldots\ \ (\bar{x}_{3m}\ y_3)
\end{array}
$$

3-Phase Refutation:

a. $(\bar{y}_4\ \bar{y}_5),\ (\bar{y}_1\ \bar{y}_5),\ (\bar{y}_1\ \bar{y}_3)$
b. $(y_1),\ (y_2),\ (y_3)$
c. Λ

Figure 2: An example of MMSA problem

Example An example is given in Figure 2. Γ_φ is the formula associated with φ, and is polynomial in the size of φ. To refute φ, if we start the proof from bottom to the top then the size of the Resolution would be $nm+n$ (the worst case). To make the proof shorter, we can do the Resolution proof from top to bottom. In this way, we can improve the size of Resolution refutation to $km+n$.

Lemma 5 *Let φ be an instance of Monotone Minimum Satisfying Assignment and let ρ equal the cardinality of the minimum satisfying assignment for φ. Γ_φ has a tree-like Resolution refutation with $O(\rho m + n)$ clauses.*

Proof Let $I \subseteq \{x_1, \ldots, x_p\}$ specify a satisfying assignment for φ of cardinality ρ. We use a top-down procedure to generate the refutation. The first phase of the refutation starts with the clause $\{\overline{y_l}\}$ and derives successively clauses of the form $\{\overline{y_{k_1}}, \overline{y_{k_2}}, \ldots, \overline{y_{k_r}}\}$ with $k_1 > k_2 > \ldots > k_r$. Such a clause is resolved with one of the (at most two) clauses that contain y_{k_1} positively. This continues until we have a clause which contains only literals $\overline{y_i}$ corresponding to input x_i of φ. It is possible to do this so that the remaining clause is just $\{\overline{y_i} : x_i \in I\}$. For the second phase of the refutation, derive the clauses $\{y_i\}$, for $x_i \in I$, with ρm steps, and for the third phases, use ρ resolutions to derive the empty clause.

There are obviously $O(n)$ steps in the first and third phases of the derivation, so the whole refutation has $O(\rho m + n)$ steps.

Lemma 6 *Let φ and ρ be as above. Then any resolution refutation must have at least ρm clauses.*

Proof Let R be a Resolution refutation. An input variable x_i is defined to be R-*analyzed* if every one of the $(m+1)$-clauses associated with x_i is used in the refutation R. Obviously it will suffice to prove that at least ρ input variables are R-analyzed. In fact, if I is defined to equal the set of R-analyzed variables, then I implies a satisfying assignment for φ.

This last fact is almost immediate. To prove it formally, we define a truth assignment τ as follows: (1) τ assigns truth values to variables y_i according to the value I assigns to φ_i (2) τ assigns True to $x_{i,k}$ iff each clause $\{x_{i,1}\}$ and $\{\overline{x_{i,j}}, x_{i,j+1}\}$ for $1 \leq j < k$ is used in R. If I doesn't satisfy φ, then τ would satisfy all the clauses used in the refutation R, which is impossible. Therefore, I is a satisfying assignment for φ.

Using lemma 5 and lemma 6, we can prove that MMSA cannot be ϵ-approximated, unless **P=NP**. This immediately implies Theorem 4 Part a.