6.841 Advanced Complexity Theory Apr 29, 2009

Lecture 22

Lecturer: Madhu Sudan *Scribe: Jessica Yuan*

1 Last Time

Last time, we talked about what a propositional proof system is. It's a way of proving that a formula is a tautology (i.e., it's always true). Very quickly, we switched from proving formulas that are always true to formulas that are always false. Without loss of generality, we focused on CNF formulas, since we can transform formulas into CNF form with only a linear blow-up.

A proof system is something that can determine whether a CNF formula F is either

- unsatisfiable: there exists some proof π that is efficiently (poly-time) verifiable deterministically
- satisfiable: there should be no fake proof that can fool us

Why do we care about propositional proof systems?

1) P vs. NP. If there is no proof system that can always provide proofs of polynomial size, this implies P $\neq$ NP.

2) Understand limits of math reasoning

3) SAT-solving. In industrial applications, there are really good algorithms for SAT-solving in linear time, and the best ones are based on a proof system called *resolution*.

We then looked at some examples of proof systems and decided to focus on resolution. This is the proof system where given clauses $C \vee x$ and $D \vee \bar{x}$, we can obtain the clause $C \vee D$. Additionally, we can use a weakening mechanism, where given clause C, we can obtain the weaker clause $C \vee D$. This weakening mechanism is not necessary for proving unsatisfiability, but it is often useful.

We looked at some examples of formula families, such as Graph Tautology, and decided to focus on Pigeonhole Principle. Pigeonhole Principle (PHP_n^m) lets us show that if there are m pigeons and n holes, where $m > n$, there must be two pigeons that share a hole. The unsatisfiable formula consists of two parts:

- Each pigeon must be in a hole: $\bigwedge_{i\in[m]} \bigvee_{j\in[n]} x_i j$
- No two pigeons share a hole: $\bigwedge_{i_1\neq i_2\in[m],j\in[n]} (\overline{x_{i_1 j}} \vee \overline{x_{i_2 j}})$

Combined, this gives us the formula:

$$\bigwedge_{i\in[m]} \bigvee_{j\in[n]} x_i j \wedge \bigwedge_{i_1\neq i_2\in[m],j\in[n]} (\overline{x_{i_1 j}} \vee \overline{x_{i_2 j}})$$

Today, we'll focus on how hard it is to prove that this formula is unsatisfiable.

2 Resolution Proof Size

We're interested in the size of resolution proofs, i.e. the total number of symbols. We can also count the number of clauses in the proof (i.e. the length of the derivation of the contradiction). This tends to be easier to deal with than size, and because length is linearly related to size, we can get a good idea of the size of a proof given its length. The width of the clauses is also a good tool for studying the length of proofs.

The length of a refutation proof π is denoted as $L(\pi)$. The length of the proof showing that a formula F is unsatisfiable is the length of the shortest proof that proves that F is unsatisfiable:

$$L(F \vdash 0) = \min_{\pi : F \vdash 0} \{L(\pi)\}$$

Note that if n is the number of variables in F, $L(F \vdash 0) \lesssim 2^n$.

The width of the proof is similarly denoted as $W(F \vdash 0) = \min_{\pi : F \vdash 0}\{W(\pi)\}$. Note that $W(F \vdash 0) \leq n$.

Any resolution proof can be represented by a directed acyclic graph by drawing edges to clauses from the clauses used to derive them. If this graph is a tree, we call it a *tree-like refutation*, and the length of the shortest tree-like refutation is $L_T(F \vdash 0)$. Generally, there is a big difference between tree-like proof lengths and general proof lengths. We can also hit the formula with an arbitrary restriction (e.g. $x = 1$), and we get the refutation of the restricted formula.

We would like to show that $W(F \vdash 0)$ is large $\rightarrow L(F \vdash 0)$ is large. Then, we will use this fact to prove that

$$L(PHP_n^{n+1} \vdash 0) = \exp(\Omega(n))$$

Resolution proofs are extremely well-studied. Resolution is one of the easiest non-trivial proof systems to deal with. Also, the best SAT-solvers tend to be built on resolution.

We want to prove lower bounds on length, but we'll first focus on width. Let's look at a connection between width and length:

If a refutation is narrow (width $\leq W$), then it must also be short. Length $\leq (2|\text{Variables in } F|)^W$

More interestingly, Ben-Sasson and Wigderson in 1999 in their paper "Short Proofs Are Narrow," they presented the idea that if there is a short proof, there must also exist a reasonably narrow proof. (Sidenote: the terms *proof* and *refutation* will be used interchangeably.)

We can quantify this for tree-like refutations:

Theorem 1 (BW '99) *Define $W(F)$ to be the width of the largest clause in formula F. Then,*

$$W(F \vdash 0) \leq W(F) + log_2 L_T(F \vdash 0)$$

Corollary 2 $L_T(F \vdash 0) \geq 2^{W(F \vdash 0) - W(F)}$

But tree-like resolutions are significantly weaker than general resolutions, so we'll want the general case:

Theorem 3 (BW '99) *Let n be the number of variables in formula F. Then,*

$$W(F \vdash 0) \leq W(F) + \sqrt{8n * ln(L(F \vdash 0))}$$

Corollary 4

$$L_T(F \vdash 0) = \exp\left(\frac{(W(F \vdash 0) - W(F))^2}{8n}\right)$$

We'll use the general theorem, but we'll only be proving the weaker tree-like theorem. The idea behind the proof for the general theorem is the same as for the weaker one, but the calculations are messier.

One way of thinking about this is that the width is upper-bounded by the logarithm of the length in Theorem 1. n is about the logarithm of the worst-case proof. You can think of it as

$$\text{width} \lesssim \sqrt{ln(\text{worst-case}) * ln(\text{actual-case})}$$

You might be thinking, maybe we can do better. Unfortunately, this bound is essentially tight, as proven by Bonet and Galesi in 1999.

To get anything interesting, we need a few things:

- We need $W(F)$ to be small.
- We need $W(F \vdash 0) = \omega(\sqrt{n * ln(n)})$.

Now, we'll prove Theorem 1 and show how to use the general theorem to get the lower bound for the refutation of Pigeonhole Principle. We'll need two lemmas.

Lemma 5 *If* $W(F_{\restriction x} \vdash D) = w$, *then* $W(F \vdash D \vee \bar{x}) = \max(W(F), w+1)$.

Note that $F_{\restriction x}$ is the formula F with the variable x set to true.

Proof Let $F = C_1 \wedge C_2 \wedge ..., C_m$.

Suppose the refutation for $F_{\restriction x} \vdash D$ looks like $\pi = D_1, D_2, ..., D_C$.

The refutation for $F \vdash D \vee \bar{x}$ would involve listing all the clauses in F, then list all the D clauses with $\bar{x}$ appended to them: $\pi' = C_1, C_2, ..., C_m, D_1 \vee \bar{x}, D_2 \vee \bar{x}, ..., D_C \vee \bar{x}$.

This is always allowed by the rules of resolution, and we can prove this by cases:

- Case 1: $D_i \vee \bar{x} \in F$. In this case, $D_i \vee \bar{x}$ is trivially allowed because it was already in F.
- Case 2: $D_i \in F$. In this case, $D_i \vee \bar{x}$ is allowed because we're only weakening the clause D_i.
- Case 3: Otherwise, it must be the case that there exist two clauses D_j and D_k such that D_j and D_k resolve to D_i, or else there is no way for D_i to be in the refutation for $F_{\restriction x} \vdash D$. By induction, we know that $D_j \vee \bar{x}$ and $D_k \vee \bar{x}$ are in the refutation for $F \vdash D \vee \bar{x}$. We can thus use the resolution rule to get $D_i \vee \bar{x}$.

■

Lemma 6 *If* $W(F_{\restriction x} \vdash 0) \leq w-1$ *and* $W(F_{\restriction \bar{x}} \vdash 0) \leq w$, *then* $W(F \vdash 0) \leq \max(w, W(F))$.

Proof Let's apply Lemma 1 to $F_{\restriction x}$.

- 1. From F, derive $\bar{x}$ in width $\leq \max(W(F), w)$.
- 2. Resolve every $C \in F$ containing x with $\bar{x}$. This gives us $F_{\restriction \bar{x}}$.
- 3. Conclude that $W(F_{\restriction \bar{x}} \vdash 0) \leq w$.

■

Now we want to go back and prove Theorem 1 (Ben-Sasson and Wigderson).

Proof We'll prove by induction over b and the number of variables n that if $L_T(F \vdash 0) \leq 2^b$, then $W(F \vdash 0) \leq W(F) + b$.

Base cases

- $n \leq W(F)$. This case is trivial, because the width can never be larger than the number of variables.
- $b = 0$. The length of refutation is 1, so the refutation must be just 0, which is possible if F contained the 0 clause.

Inductive step

Consider the tree-like resolution $L_T(F \vdash 0)$. By induction, $L_T(F \vdash 0) \leq 2^b$.

The last step of this refutation π must be the derivation of some variable x and another derivation of $\bar{x}$. Let's call these derivations π_x and $\pi_{\bar{x}}$.

$L(\pi) = L(\pi_x) + L(\pi_{\bar{x}}) + 1$. So at least one of these subderivations has length $\leq 2^{b-1}$. Without loss of generality, let's suppose it is π_x. $L(\pi_x) \leq 2^{b-1}$. This means that $L(F_{\restriction x}) \leq 2^{b-1}$ as well.

Apply the inductive hypothesis. We get

$$W(F_{\restriction x} \vdash 0) \leq W(F_{\restriction x}) + b - 1 \leq W(F) + b - 1$$

We also know that $L(\pi_{\bar{x}} \leq 2^b$ and $L(F_{\restriction \bar{x}}) \leq 2^b$. Applying the inductive hypothesis there, we get

$$W(F_{\restriction \bar{x}} \vdash 0) \leq W(F_{\restriction \bar{x}}) + b \leq W(F) + b$$

By Lemma 6, we conclude that $W(F \vdash 0) \leq W(F) + b$. ■

Note that in each step in our induction, the length sort of doubles, so at the end, we have a really narrow proof, but it might be of exponential length. In fact, for tree-like resolution, we know that this length blow-up is necessary. It's an open question as to whether this is necessary for general resolution.

Why doesn't this work in the general case? It fails at $L(\pi) = L(\pi_x) + L(\pi_{\bar{x}}) + 1$, because generally, the two derivations π_x and $\pi_{\bar{x}}$ will share some steps and clauses. But the general idea of hitting the formula with $x = 0$ and $x = 1$ is still the same. However, it's trickier to find a good x, the calculations are messier, and the bounds are not as good.

Now we can finally prove the lower bound for PHP resolution.

As a reminder, PHP_n^m consists of clauses

- $P^i = \bigvee_{j \in [n]} x_i j$ "pigeon i must be in some hole"
- $H_j^{i_1,i_2} = \overline{x_{i_1,j}} \vee \overline{x_{i_2,j}}$ "hole j does not hold both pigeons i_1 and i_2"

When $m > n$, PHP is unsatisfiable. We'll focus on the hardest case to prove, which is when $m = n + 1$.

Theorem 7 (Haken '85) $L(PHP_n^{n+1} \vdash 0) = \exp(\Omega(n))$

We want to use the Ben-Sasson and Wigderson machinery described in Theorem 3, but PHP currently does not satisfy the requirements that $W(PHP)$ is small and that $W(PHP \vdash 0) = \omega(\sqrt{n * ln(n)})$. To get around this, we want to make the formula "sparser."

We can do this by thinking about PHP as a bipartite graph $G = (U \vee V, E)$, where U is the set of the m vertices on the left corresponding to the m pigeons, V is the set of the n vertices on the right corresponding to the n holes, and E is the set of edges connecting a left vertex (pigeon) to a right vertex (hole). Plus, we define $N(u)$ to be the set of neighbors a pigeon vertex $u \in U$ has. Likewise, for a set of pigeon vertices U, $N(U)$ is the set of all vertices that are a neighbor to at least one vertex in U. Note that all vertices in $N(u)$ or $N(U)$ are hole vertices.

Each edge $x_{u,v}$ in the graph connects a pigeon vertex u with a hole vertex v. $x_{u,v}$ is set to true if pigeon u goes into hole v. Then, we can say that the graph is satisfiable if there is an assignment of edges such that $PHP(G)$ is satisfied:

$$PHP(G) = \bigwedge_{u \in U} \bigvee_{v \in N(u)} x_{u,v} \wedge \bigwedge_{v \in V} \bigwedge_{u \neq u' \in N(v)} \overline{x_{u,v}} \vee \overline{x_{u',v}}$$

The simplest way of thinking about PHP is to consider G as a complete bipartite graph. However, to prove a lower bound on the proof complexity of PHP, it is not necessary to use a complete bipartite graph. In fact, if $G' = (U \vee V, E')$ has $E' \subseteq E$, then $L(PHP(G') \vdash 0) \leq L(PHP(G) \vdash 0)$. In other words, if we can find a lower bound for a sparser version of G, the lower bound also applies to G itself.

Suppose G has constant left degree $d \geq 2$. Then $PHP(G)$ is a d-CNF formula with $d * m$ variables, meaning it has a small width and small number of variables. This would satisfy the requirements for using the Ben-Sasson and Wigderson machinery.

The problem is, we want to make the graph sparser, but maintain the difficulty of the Pigeonhole Problem in order to obtain a high lower bound. The reason why the Pigeonhole Problem is so difficult is that it's a global problem. All the pigeon variables have to be observed. If there are n pigeons and n holes, the problem is satisfiable, but once there are $n + 1$ pigeons, it is not.

So what we want is a sparse graph with good connectivity, which hopefully makes the graph harder to satisfy and gives us a better lower bound. This type of graph is called an *expander*.

$G = (U \vee V, E)$ is a (d, s, e)-expander if

- the left degree of the graph $\leq d$ (d is the outdegree bound of the pigeon vertices)
- $\forall U' \subseteq U \ |U'| \leq s \rightarrow |N(U')| \geq e|U'|$

Basically, what this definition says is that for all subsets U' of U, the set of pigeon vertices, up to a limiting size $s \geq |U'|$, the number of hole vertex neighbors that this subset U' has is greater than some multiplicative expansion factor e of the size of the subset $|U'|$. In other words, if you look at a small number of pigeons, there's e times that number of holes that they can go into.

Actually, we'll need something stronger called a *unique neighbor expander*.

G is a (d, s, e)-unique neighbor expander if

- the left degree of the graph $\leq d$
- $\forall U' \subseteq U \ |U'| \leq s \rightarrow |\delta(U')| \geq e|U'|$ where $v \in \delta(U')$ if $|N(v) \cap U'| = 1$.

This definition is slightly different from the definition for the general expander. e is no longer the expansion factor of the neighbor vertices, but rather the *boundary* vertices. These are the vertices in $N(U')$ that only have one neighbor in U', hence the name "unique neighbor." In other words, these are the holes that only allow one certain pigeon from U', i.e. no two pigeons are fighting over any hole in this set of boundary vertices.

Proposition: Any (d, s, k)-expander is a $(d, s, 2k - d)$-unique neighbor expander.

Now, we'll introduce and prove the following two lemmas and then use them in conjuction with Ben-Sasson and Wigderson to prove a lower bound for PHP.

Lemma 8 (A) *For a (d, s, e)-unique neighbor expander where $e \geq 1$, $W(PHP(G) \vdash 0) \geq (s * e)/2$.*

Lemma 9 (B) *There is a $c > 1$ such that $\forall$ sufficiently large n, there are $(5, \frac{n}{c}, 1)$-unique neighbor expanders.*

Lemma B can be proven probabilistically. Just take five neighbors of each node on the left, and with high probability, you'll end up with a unique neighbor expander.

To prove Lemma A, we start by defining a "progress measure" μ: {clauses} $\rightarrow \mathbb{N}$ with the following properties.

- μ(axioms) ≤ 1
- μ(final empty clause 0) $\geq$ large
- μ only increases gradually.
- For a medium progress clause D_i, its width $W(D_i) \geq$ large

As before, let $\mathbb{H}$ denote all the "hole" axioms and P denote all the "pigeon" axioms. P^u refers to the clause that describes which holes pigeon u can go into (e.g. $P^4 : x_{4,1} \vee x_{4,3}$ if there are only edges to holes 1 and 3). Then we can define such a progress measure μ:

$$\mu(D) = \min\{|U'| : \bigwedge_{u \in U'} P^u \wedge H \vDash D\}$$

To confirm that this measure satisfies the properties:

- μ(axioms) ≤ 1. This is satisfied because for any hole axiom H^v, $\mu(H^v) = 0$. For any pigeon axiom P^u, $\mu(P^u) = 1$.
- μ(final empty clause 0) $\geq$ large. It turns out that $\mu(0) > s$, and the s we'll be using is the s in the (5, $\frac{n}{c}$, 1)-unique neighbor expander guaranteed to us by Lemma B. By the definition of unique neighbor expanders, if we have s or fewer pigeons, we are guaranteed to have at least that many unique neighbors. So, we must be considering more than s pigeons if we want to achieve a proof of unsatisfiability.
- μ only increases gradually. Quantitatively, if D_j and D_j resolve to D_i, then $\mu(D_i) \leq \mu(D_j) + \mu(D_k)$. This ensures that there is some D_i with a medium-sized $\mu(D_i)$
- For a medium progress clause D_i, its width $W(D_i) \geq$ large. By the previous requirement, we know that there must be a medium progress clause D such that $\frac{s}{2} \leq \mu(D_i) \leq s$, since each derivation cannot increase the progress measure by more than doubling. We then consider the width of this clause D.

For this D, fix U' of size $\mu(D)$ such that $\bigwedge_{u \in U'} P^u \wedge H \vDash D$.

We then make the claim that $\forall v \in \delta(U')$ $\exists$ variable $x_{u,v}$ in D. We can prove this by contradiction.

Proof Suppose $\exists v* \in \delta(U')$ such that there is no variable of the form $x_{u,v*}$ in D. Because $v* \in \delta(U')$, we know it has a unique neighbor $u* \in U'$. We also know that by the definition of the progress measure μ, if we remove $u*$ from U', we should no longer be able to derive D; otherwise, U' would not be the minimal subset.

Thus, it must be the case that there is some truth-value assignment α such that

- $\alpha(\bigwedge_{i \in U' \setminus \{u*\}} P^i) = 1$

- $\alpha(\mathbb{H}) = 1$
- $\alpha(D) = 0$

In other words, there should be some truth-value instance where the axioms (minus $u*$'s pigeon axiom) are satisfied but the derived clause D is not. Now, we can manipulate the values $x_{u,v}$ a little. (Recall that $x_{u,v}$ is true iff pigeon u goes into hole v.)

First, we can set $x_{u,v*}$ to false, for all u. This won't affect $\alpha(D)$, because by our assumption, there is no variable of the form $x_{u,v*}$ in D. This won't affect the pigeon axioms, since $v*$ only has one neighbor $u*$, who is not included in the axioms. And this can only serve to strengthen the hole axioms, since they are negations.

Then, we can set $x_{u*,v*}$ to true. Again, this won't affect $\alpha(D)$ or the pigeon axioms. This also isn't affecting any of the hole axioms, because we set all other $x_{u,v*}$ to false, so any hole axiom involving $v*$ is true, regardless of the value of $x_{u*,v*}$. (Recall that each hole axiom is of the form $\overline{x_{i_1 j}} \vee \overline{x_{i_2 j}}$.)

Now we have $\alpha(P^{u*}) = 1$, because pigeon $u*$ was able to find a hole. We can combine that with $\alpha(\bigwedge_{i \in U' \setminus \{u*\}} P^i) = 1$ to get $\alpha(\bigwedge_{i \in U'} P^i) = 1$. However, now we have

- $\alpha(\bigwedge_{i \in U'} P^i) = 1$
- $\alpha(\mathbb{H}) = 1$
- $\alpha(D) = 0$

This is a contradiction, since we should be able to derive D from the pigeon and hole axioms. ■

Now we know that the width of D is at least as great as $|\delta(U')|$. But $|\delta(U')| \geq e|U'| \geq (s * e)/2$. Hence, $W(D) \geq (s * e)/2$ and Lemma A follows.

To finish up the proof for the lower bound of PHP, we plug this information into the Ben-Sasson and Wigderson equation. Recall:

$$L_T(F \vdash 0) = \exp\left(\frac{(W(F \vdash 0) - W(F))^2}{8x}\right)$$

Note that in this formula, x is the number of variables. If n is the number of holes and $n+1$ is the number of pigeons, then the number of variables is on the order of $5n$, given that we are using the $(5, \frac{n}{c}, 1)$-unique neighbor expander. $W(F)$ is $O(1)$, because each hole axiom has width 2 and each pigeon axiom has width 5. $W(F \vdash 0)$ is at least $(s * e)/2$, or $\Omega(n)$. Plug it all in, and we get

$$L_T(F \vdash 0) = \exp\left(\frac{(\Omega(n))^2}{n}\right) = \exp(\Omega(n))$$