Math 267a - Propositional Proof Complexity

Lecture #7: 6 February 2002

Lecturer: Sam Buss

Scribe Notes by: Dan Curtis

1 Completeness and Soundness of Resolution Proofs

1.1 Definition of a Resolution Proof

Recall the resolution rule:
$$\frac{C \cup \{x\} \quad D \cup \{\bar{x}\}}{C \cup D}.$$

Definition A set of literals $\{x_1, \ldots, x_n\}$, with x_i in P_k or $\bar{P}_k$, is called a clause.

Definition Resolution refutes a set of clauses if and only all the clauses cannot be simultaneously satisfied.

A clause is a disjunction of literals and a set of clauses is a conjuction of clauses, which can be thought of as a conjuctive normal form formula. We can view resolution as proving disjunctive normal form formulas. For right now, resolution can prove tautologies that are in Disjunctive Normal Form.

Example The Pigeon hole principle (PHP_n^m) can be written as

$$\bigwedge\nolimits_{i=1}^{m} \bigvee\nolimits_{j=1}^{n} p_{ij} \rightarrow \bigvee\nolimits_{i=1}^{m-1} \bigvee\nolimits_{j=i+1}^{m} \bigvee\nolimits_{k=1}^{n} (p_{ik} \wedge p_{jk}).$$

The negation of this ($\neg PHP_n^m$) is

$$\bigwedge\nolimits_{i=1}^{m} \bigvee\nolimits_{j=1}^{n} p_{ij} \wedge \bigwedge\nolimits_{i=1}^{m-1} \bigwedge\nolimits_{j=i+1}^{n} \bigwedge\nolimits_{k=1}^{n} (\bar{p}_{ik} \wedge \bar{p}_{jk}).$$

which is in conjunctive normal form.

Written as a set of clauses:

$$\begin{array}{rcl} \{p_{i,1}, \ldots, p_{i,n}\}, & i = 1, \ldots, m & \leftarrow m \text{ clauses} \\ \{\bar{p}_{ik}, \bar{p}_{jk}\}, & i = 1, \ldots, m-1;\ j = i+1, \ldots, n;\ k = 1, \ldots n & \leftarrow\approx m^2 \text{ clauses} \end{array}$$

A resolution "proof" of PHP means a refutation of this set of clauses.

1.2 Completeness Theorem

Theorem 1 *(Completeness Theorem) If $\mathcal{C}$ is an unsatisfiable set of clauses, then $\mathcal{C}$ has a resolution refutation.*

Proof Using induction on the number of variables in $\mathcal{C}$, assume $\mathcal{C}$ has zero variables. Then either $\mathcal{C} = \{\emptyset\}$, in which case it contains the refutation $\emptyset$, or $\mathcal{C} = \emptyset$ which is satisfiable. Thus the hypothesis holds for any clause with zero variables.

Now, let $\mathcal{C}$ be an unsatisfiable set of clauses and let x be a variable in some clause in $\mathcal{C}$. Define

$$\begin{aligned}
\mathcal{C}_x &= \{\text{the set of clauses in } \mathcal{C} \text{ that contain } x\} \\
\mathcal{C}_{\bar{x}} &= \{\text{the set of clauses in } \mathcal{C} \text{ that contain } \bar{x}\} \\
\mathcal{C}' &= \mathcal{C} - (\mathcal{C}_x \cup \mathcal{C}_{\bar{x}}).
\end{aligned}$$

Then resolve all $\mathcal{C}_x$ clauses with all $\mathcal{C}_{\bar{x}}$ clauses by

$$\frac{D \cup \{x\} \;\; E \cup \{\bar{x}\}}{D \cup E}$$

Let $\mathcal{D} = \mathcal{C}' \cup \{$all resolvents of the form $D \cup E$, where $D \cup \{x\} \in \mathcal{C}_x$ and $E \cup \{\bar{x}\} \in \mathcal{C}_{\bar{x}}\}$

Since $\mathcal{D}$ has fewer variables than $\mathcal{C}$, then by the induction hypothesis, if $\mathcal{D}$ is unsatisfiable, then $\mathcal{D}$ has a refutation. Also, from the construction of $\mathcal{D}$, if $\mathcal{D}$ has a refutation, then $\mathcal{C}$ has a refutation. Thus, if we can show that $\mathcal{D}$ is unsatisfiable, then $\mathcal{C}$ has a refutation.

Suppose $\mathcal{D}$ is satisfiable and τ is a truth assignment that satisfies $\mathcal{D}$. Define τ^+ to be the same as τ with the addition that $\tau(x) = T$, and define τ^- to be the same as τ with the addition that $\tau(x) = F$.

Suppose τ^+ does not satisfy $\mathcal{C}$. Then there is a $E \cup \{\bar{x}\} \in \mathcal{C}$ such that τ^+ does not satisfy $E \cup \{\bar{x}\}$. But then τ does not satisfy E. Similarly, if τ^- does not satisfy $\mathcal{C}$, then there is a $D \cup \{x\}$ such that τ does not satisfy D. However, since τ satisfies $\mathcal{D}$, τ satisfies $D \cup E$. So, either τ^+ or τ^- satisfies $\mathcal{C}$.

1.3 Size of a Resolution Proof

The size of a resolution proof can be measured in two ways:

a) Total number of literals in all clauses.

b) Number of clauses.

Clearly, (b) $\leq$ (a) $\leq$ (b)$\cdot$(number of distinct variables), so a polynomial size bound on b) implies a polynomial size bound on a).

1.4 Subsumption Rule

Definition The subsumption rule (weakening rule), for any two clauses C and D with $C \subseteq D$, is given by

$$\frac{C}{D}.$$

Theorem 2 *A resolution and subsumption refutation of a set $\mathcal{C}$ of clauses can be converted into a smaller resolution refutation of $\mathcal{C}$.*

In practice, a theorem prover has $C_1, \ldots, C_k$ as input clauses and generates clauses with resolution. At some point, if it has clauses D and E with $E \subseteq D$, then it is alright to discard D without any negative consequences.

Proof Let $\phi_1, \ldots, \phi_k = \emptyset$ be a refutation using resolution and subsumption. A new refutation $\psi_1, \ldots, \psi_k = \emptyset$, built recursively in the following way using only resolution, will have the property that $\psi_i \subseteq \phi_i$ for each $i \leq k$.

For each $i \leq k$, define ψ_i as follows:

1) If $\phi_i \in \mathcal{C}$, then set $\psi_i = \phi_i$. In this case, clearly $\psi_i \subseteq \phi_i$.

2) If ϕ_i is inferred by subsumption $\frac{\phi_l}{\phi_i}$ for some $l \leq i$, with $\phi_l \subseteq \phi_i$, then set $\psi_i = \psi_l$. Here, we have $\psi_i = \psi_l \subseteq \phi_l \subseteq \phi_i$.

3) If ϕ_i is inferred by resolution, for some $j, l \leq i$,

$$\frac{\phi_j \ \phi_l}{\phi_i}$$

resolving on $x \in \phi_j$ and $\bar{x} \in \phi_l$, do the following:

a) If $x \notin \psi_j$, set $\psi_i = \psi_j \subseteq \phi_i$.

b) If $\bar{x} \notin \psi_l$, set $\psi_i = \psi_l \subseteq \phi_i$.

c) Otherwise, set $\psi_i = \mathrm{res}_x(\psi_j, \psi_l)$, where res_x is defined to be the resolvent obtained by the resolution using the literal x. Since $\psi_j \subseteq \phi_j$ and $\psi_l \subseteq \phi_l$, then $\psi_i \subseteq \phi_i$.

Clearly, $\psi_k = \emptyset$, since $\psi_k \subseteq \phi_k = \emptyset$. Finally, erase any duplicate ψ_i's.

1.5 Refutation Proof of the Pigeon Hole Principle

As a point of notation, throughout this proof, we will use $[k]$ to denote the set $\{1, \ldots, k\}$.

Recall that the negation of the Pigeon Hole Principle can be written as:

$$\bigwedge_{i=1}^{m} \bigvee_{j=1}^{n} p_{ij} \wedge \bigwedge_{i=1}^{m-1} \bigwedge_{j=i+1}^{n} \bigwedge_{k=1}^{n} (p_{ik} \wedge p_{jk}).$$

For this proof, we will prove the special case PHP_n^{n+1} (i.e. $m = n + 1$). Writing this as a set of clauses, we get

$$\mathcal{C} = \{\{P_{i,1}, \ldots, P_{i,n}\}, 1 \leq i \leq n\} \cup \{\{\bar{P}_{i,k}, \bar{P}_j, k\}, 1 \leq i \leq j \leq m; 1 \leq k \leq n\}$$

Proof The refutation will proceed in a series of stages, $s = n, n-1, \ldots, 0$. At stage s, we have the following clauses: For each injective map $\pi : \{1, \ldots, s\} \to \{1, \ldots, n\}$ we have the clause $\{\bar{P}_{1,\pi(1)}, \bar{P}_{2,\pi(2)}, \ldots, \bar{P}_{s,\pi(s)}\}$.

At stage $s = 0$, the only map is $\pi : \emptyset \to [n]$ and the clause is $\emptyset$.

At stage $s = n$, for any injective map $\pi : [n] \to [n]$, start with the initial clause $\{P_{n+1,1}, \ldots, P_{n+1,n}\}$ and resolve with the initial clauses $\{\bar{P}_{i,\pi(i)}, \bar{P}_{n+1,\pi(i)}\}$ for each $1 \leq i \leq n$.

For the induction step, assume we have the stage $s+1$ clauses. Given any injective map $\pi : [s] \to [n]$ we need to derive $\{\bar{P}_{1,\pi(1)}, \bar{P}_{2,\pi(2)}, \ldots, \bar{P}_{s,\pi(s)}\}$. For $j \notin \text{Range}(\pi)$, define π_j to be $\pi \cup \{(s+1) \mapsto j\}$. Since $\pi_j : [s+1] \to [n]$, then from stage $s+1$ we already have

$$(*_j) \qquad \{\bar{P}_{1,\pi(1)}, \bar{P}_{2,\pi(2)}, \ldots, \bar{P}_{s,\pi(s)}, \bar{P}_{s+1,j}\}.$$

To derive the stage s clauses, start with the initial clause $\{P_{s+1,1}, \ldots, P_{s+1,n}\}$ and resolve with the initial clauses $\{\bar{P}_{i,\pi(i)}, \bar{P}_{s+1,\pi(i)}\}$ for each $1 \leq i \leq s$. After resolving with each of the s clauses, we get

$$\{\bar{P}_{1,\pi(1)}, \bar{P}_{2,\pi(2)}, \ldots, \bar{P}_{s,\pi(s)}, P_{s+1,j_1}, \ldots, P_{s+1,j_{n-s}}\}$$

where $[n] - \text{Range}(\pi) = \{j_1, \ldots, j_{n-s}\}$. Finally, resolve with the $(*_j)$ clauses for $j = j_1, \ldots, j_{n-s}$ and we get $\{\bar{P}_{1,\pi(1)}, \bar{P}_{2,\pi(2)}, \ldots, \bar{P}_{s,\pi(s)}\}$ as desired.

1.6 Size of Proof of Pigeon Hole Principle

There are n stages for this proof of PHP_n^{n+1}. At each stage, there are on the order of $O(n^s)$ injective maps $\pi : [s] \to [n]$. Also, there are n steps required to derive each clause. Thus, the size of this proof is on the order of $O(n \cdot n \cdot n^n) = 2^{O(n \log n)}$ total number of clauses.

However, a more honest measure of the size of the proof is in terms of the number of variables $v = \Omega(n^2)$. In terms of v, the size of the proof is on the order $2^{O(\sqrt{v} \log \sqrt{V})} = 2^{O(\sqrt{v} \log V)}$.

1.7 Soundness Theorem

Theorem 3 *(Soundness Theorem) If $\mathcal{C}$ is a set of clauses with a refutation, then $\mathcal{C}$ is unsatisfiable.*

Proof Proof of the soundness theorem is deferred until the next lecture.