

**Notes for Lecture 18**  
**Malka Rosenthal**

## 1 Review of Resolution

**Definition 1** A formula in CNF - Conjunctive Normal Form is expressed as a set of clauses  $\{C_1, \dots, C_k\}$ , where each  $C_i$  is a set of literals  $\{l_1, \dots, l_j\}$  interpreted as the disjunction of those literals,  $l_1 \vee \dots \vee l_j$  and the formula is interpreted as the conjunction of the clauses.

**Definition 2** A resolution derivation from  $C_1, \dots, C_k$  is a sequence  $D_1, \dots, D_l$  such that each  $D_i$  is either

1. one of the  $C_i$ 's or
2. of the form  $D_i = C_{j_1} \cup C_{j_2} \setminus \{x_r, \overline{x_r}\}$  where  $C_{j_1}$  and  $C_{j_2}$  have  $x_r, \overline{x_r}$  as their only pair of complementary literals.

**Definition 3** A refutation is a derivation whose last element is  $\{\}$  (denoted  $\square$ ).

This empty clause has the meaning of False; the only way it can be obtained in a derivation is for  $C_{j_1} = \{x\}, C_{j_2} = \{\overline{x}\}$  for some literal  $x$  and clearly  $x \wedge \overline{x}$  cannot be True.

If  $\varphi$  is a formula in DNF, then we may abuse notation slightly and speak of  $\neg\varphi$  as being in CNF (whereas it would be more precise to consider the formula in CNF that results by applying DeMorgan's laws to  $\neg\varphi$ ).

**Theorem 4** Let  $\varphi$  be in DNF. Then  $\models \varphi$  (i.e.  $\varphi$  is a tautology) iff there is a resolution refutation of  $\neg\varphi$ .

Although this form of proof is limited, it is in fact used and is therefore of interest.

## 2 Connection between Circuit Lower Bounds and Propositional Logic

We will build a set of clauses which encode the clique and coloring problems. Consider the following clauses:

$$\begin{aligned} \{q_{i,1}, q_{i,2}, \dots, q_{i,n}\} & \text{ for } 1 \leq i \leq k \\ \{\neg q_{i,m}, \neg q_{j,m}\} & \text{ for } 1 \leq m \leq n \text{ and } 1 \leq i < j \leq k \\ \{\neg q_{i,m}, \neg q_{j,l}, p_{m,l}\} & \text{ for } 1 \leq m < l \leq n \text{ and } 1 \leq i, j \leq k \end{aligned}$$

The above clauses encode a graph which contains a  $k$ -clique as follows:

- The  $q$ 's encode a one-to-one function from  $\{1, \dots, k\} \rightarrow \{1, \dots, n\}$  if we set  $q_{i,j} = 1 \Leftrightarrow q(i) = j$  so that the clause  $\{q_{i,1}, \dots, q_{i,n}\}$  which means  $[(q(i) = 1 \vee \dots \vee (q(i) = n)]$  is defining a function and the clause  $\{\neg q_{i,m}, \neg q_{j,m}\}$  which is equivalent to  $[q(i) = m \Rightarrow q(j) \neq m]$  ensures that the function is one-to-one.
- The  $p$ 's encode a graph if we take  $p_{m,l} = 1$  to mean that there's an edge between  $m$  and  $l$ . With this intuition, the last set of clauses  $\{\neg q_{i,m}, \neg q_{j,l}, p_{m,l}\}$  is equivalent to  $q(i) = m, q(j) = l \Rightarrow m \leftrightarrow l$ .

Thus,  $\{p_{m,l} : 1 \leq m < l \leq n\}$  encodes a graph which contains a  $k$ -clique iff there exist assignments to the  $q$  variables making these clauses true.

Next, consider the sets of clauses that encode the property of being an  $l$ -partite graph:

$$\begin{array}{ll} \{r_{i,1}, \dots, r_{i,l}\} & \text{for } 1 \leq i \leq n \\ \{\neg r_{i,a}, \neg r_{i,b}\} & \text{for } 1 \leq i \leq n, 1 \leq a < b \leq l \\ \{\neg r_{i,a}, \neg r_{j,a}, \neg p_{i,j}\} & \text{for } 1 \leq a \leq l \text{ and } 1 \leq i < j \leq n \end{array}$$

Here the explanation is as follows:

- The first set of clauses encode the coloring function:  $r_{i,c} = 1$  means vertex  $i$  has color  $c$ , ( $r(i) = c$ ) and the second set of clauses means that each vertex has at most one color.
- Finally, we make this a proper coloring with the last set of clauses: If  $r(i) = a$  and  $r(j) = a$  then  $p_{i,j} = 0$ , i.e. there is no edge between vertices  $i$  and  $j$ .

**Claim:** If  $k = l + 1$  then these clauses (i.e. both sets together) are unsatisfiable.

**Proof:** Every assignment to the  $p$ 's gives a graph. If all clauses are satisfiable, there is some assignment to the  $q$ 's which encodes a  $k$ -clique in the graph and at the same time an assignment to the  $r$ 's which gives a proper  $k \Leftrightarrow 1$ -coloring of the graph. This is of course impossible.

By the following theorem, our circuit lower bound for cliques gives an exponential lower bound on the length of proof needed to refute these clauses.

**Theorem 5** *Given any resolution refutation of the above clauses, we can build a monotone circuit of size  $\approx$  the length of the refutation, that evaluates to 1 on all  $k$ -cliques and to 0 on all  $k \Leftrightarrow 1$ -partite graphs.*

**Corollary 6** Any resolution refutation of these clauses requires length  $2^{\Omega(\sqrt{k})}$ , if  $k \leq \sqrt[3]{n}$ .

We will use the following theorem from propositional logic:

**Theorem 7 (Craig’s Interpolation Theorem)** Let  $A(\vec{x}, \vec{y})$  and  $B(\vec{x}, \vec{z})$  be propositional formulae using only the variables in  $\vec{x}, \vec{y}, \vec{z}$  where  $\vec{x}, \vec{y}, \vec{z}$  are disjoint lists of variables:  $\{x_1, \dots, x_{i_1}, y_1, \dots, y_{i_2}, z_1, \dots, z_{i_3}\}$ . Suppose that the following statement is a tautology:

$$A(\vec{x}, \vec{y}) \Rightarrow B(\vec{x}, \vec{z})$$

Then there is a propositional formula  $C(\vec{x})$  such that  $A(\vec{x}, \vec{y}) \Rightarrow C(\vec{x})$  and  $C(\vec{x}) \Rightarrow B(\vec{x}, \vec{z})$ .

**Proof.** Note that if we have made assignments to  $\vec{x}$  such that  $A(\vec{x}, \vec{y})$  is satisfiable, then there is no way to assign truth values to  $\vec{z}$  to make  $\neg B(\vec{x}, \vec{z})$  true. Let  $\vec{x}_1, \dots, \vec{x}_r$  be all such assignments to  $\vec{x}$  and let

$$C(\vec{x}) = \bigvee_{i=1}^r (\vec{x} = \vec{x}_i).$$

Then  $A(\vec{x}, \vec{y}) \Rightarrow C(\vec{x})$  and also  $C(\vec{x}) \Rightarrow B(\vec{x}, \vec{z})$ . ■

This theorem doesn’t seem particularly difficult; however, there is an interesting question which can be asked: If  $A$  and  $B$  are “small” formulae, is there such a  $C$  having a small formula (or circuit)? Consider the following example:

**Example 1** Consider the set of pairs  $\{(x, y) : x \text{ is the binary representation of a number and there is a prime } p, p|x \text{ such that the binary representation of } p \text{ has } y \text{ as a prefix}\}$

**Claim** This set, which we will call “FACT” is in both  $\mathcal{NP}$  and  $\mathcal{CO} \Leftrightarrow \mathcal{NP}$ .

**Proof**

1.  $\text{FACT} \in \mathcal{NP}$ : Nondeterministically, guess  $p_1, p_2, \dots, p_r$  and check that  $p_1 \cdot \dots \cdot p_r = x$  and guess proofs that each  $p_i$  is prime. Accept iff  $y$  is a prefix of some  $p_i$
2.  $\text{FACT} \in \mathcal{CO} \Leftrightarrow \mathcal{NP}$  : Do the same as above and accept iff  $\neg \exists i, y$  is a prefix of  $p_i$ .

**Claim** If FACT has polynomial size circuits then the problem of Factorization does too. This is an open question and an affirmative answer is not expected (so we do not expect FACT to have poly-size circuits).

**Proof** Consider the following routine:(This routine runs a binary search)

Input =  $x$

If  $(x, 1) \notin FACT$  output  $x$   
 Else Set  $y = 1$   
 Repeat[There is no need to end this loop as a factor will be found]:  
 If  $(x, y0) \in FACT$  set  $y = y0$   
 Else set  $y = y1$   
 If  $y|x$  output  $y$ (as  $y$  is a factor)

By Cook's reduction formula which shows that SAT is  $\mathcal{NP}$ -Complete, we know that there are formulae  $\varphi_1, \varphi_2$  such that:

$$(x, y) \in FACT \Leftrightarrow \varphi_1(x, y, z_1) \in SAT$$

$$(x, y) \notin FACT \Leftrightarrow \varphi_2(x, y, z_2) \in SAT.$$

Note that  $(x, y) \in FACT \Leftrightarrow \neg[(x, y) \notin FACT]$

Thus ,

$$\varphi_1(x, y, z_1) \in SAT \Leftrightarrow \varphi_2(x, y, z_2) \notin SAT \Leftrightarrow \models \neg\varphi_2(x, y, z_2)$$

and similarly,

$$\varphi_2(x, y, z_2) \in SAT \Leftrightarrow \models \neg\varphi_1(x, y, z_1)$$

Thus,

$$\models \varphi_1(x, y, z_1) \rightarrow \neg\varphi_2(x, y, z_2)$$

These formulae ( $\varphi_1, \varphi_2$ ) are small (i.e. polynomial size as FACT is polynomial-time reducible to SAT) and therefore an interpolating formula which is small (i.e. has small circuits) would compute FACT in polynomial size circuits.

In conclusion, we assume that not all small formulae,  $A$  and  $B$ , have small interpolating formulae.

On the other hand, it is known that if there is a short resolution derivation for  $\models A(\vec{x}, \vec{y}) \Leftrightarrow B(\vec{x}, \vec{z})$  then we can build a short  $C$ . In the case of the clique\bipartition problem we will show that a short resolution derivation will actually give a small monotone circuit  $C$ .