Lecture 14

Lecturer: Daniel A. Spielman *Scribe: Xiangwei Liu*

In today's lecture, we will discuss the circuit lower bounds problem and prove the first part of razborov theorem.

1 Circuit Lower Bounds

The *circuit lower bounds problem* is to study the difficulty of deciding a class of functions(maybe defined by other means, such as FP or $\sharp$P) by circuits, on the hope of a better understanding of the mysterious complexity hierachy.

Obviously, for general circuits, i.e., a circuits with all three different gates, it maybe too powerful for us to get a nice result. Actually, the best known lower bounds for deciding NP funtions under general circuits is only 4n. (Note that Linear lower bounds for circuits, or more generally, *all kinds of linear stuffs* are not so useful in complexity theory). So to make our task of proving a lower bound easier, we will restrict the power of our circuits. Specifically, we will study the circuit lower bounds for deciding **CLIQUE functions** under **monotone circuits**.

First, let's define the CLIQUE functions and monotone circuits.

Definition 1 *A function f is* **monotone** *if $f(\bar{x}) = 1 \Rightarrow f(\bar{y}) = 1$ for all $\bar{y}$'s such that all bits that are 1 in $\bar{x}$ are also 1 in $\bar{y}$. (So we can see that $f(\bar{y}) = 0 \Rightarrow f(\bar{x}) = 0$ for all $\bar{x}$'s such that all bits that are 0 in $\bar{y}$ are also 0 in $\bar{x}$)*

Definition 2 *A circuit C is* **monotone** *if it only has $\vee$ and $\wedge$ gates, with no $\neg$ gates.*

We can tell by simple observation that monotone circuits only computes monotone functions, and vice versa.

Proposition 3 *A monoton circuit computes monotone functions and every monotone function can be computed by a monotone circuit.*

Definition 4 *let CLIQUE function $CLIQUE_{k,n}$ to be the function on n-nodes graphs that outputs 1 if and only if the input graph has a k-clique.*

From the defintion, we conclude that CLIQUE funtions are monotone functions. (if CLIQUE(k, G)=1, we know G has a k-clique, so $f(G')$ will be 1 for all $G's$ that has all the edges G have.)

Now, we will present the main theorem of today's lecture, the *Razborob Theorem.* Let's make one definition first:

Definition 5 *We define the* **Size** *function to be the function that on input of a $\{0,1\}^* \rightarrow \{0,1\}$ function h outputs the circuit complexity needed to compute h. (Or, minimum circuit size needed to compute h.)*

Theorem 6 (Razborov) $Size(CLIQUE_{\log n, n}) \geq n^{-2(\log n)}$

A similar case is studied by Alon and Boppana:

Theorem 7 (Alon-Boppana) $Size(CLIQUE_{n^{\frac{2}{3}}, n}) > 2^{\frac{n^{\frac{1}{3}}}{\log n}}$

The rest of today's lecture will be devoted to the first part of the proof of *Razborov theorem.*

Variables k, l, and m will appear in the proof. While we will not decide on the best values for these until the end of the proof, we will give away an approximation of the values we will choose so that we can have something in mind as we go through the proof.

$$k \sim n^{\frac{1}{4}}, l \sim \sqrt{k}, m \sim l^l$$

First, let's see how can we start. It seems that to consider all cases simultaneously is hard, so we have to simplifize the problem a little bit somehow(But without lose of generality.) To simplifize the proplem, We will only look at the performance of CLIQUE functions on restricted test graphs, more specifically, we will only study the performance of CLIQUE on minimal positive instances and maximal negative instances, and then argue that this way didn't make us lose any generality.

The minimal positive graphs are those graphs that contain a clique on some set of k vertices, and no other edges. The maximal negative graphs are the maximal $(k-1)$-colorable graphs. These are obtained by coloring each vertex in the graph with one of $k-1$ colors, and placing edges between each pair of differently-colored vertices. The should be no edges between vertices of the same color.

Definition 8 *A* **Clique indicator** *is a function that is 1 if input graph contains some particular clique, and 0 otherwise. The* size *of a clique indicator is the number of vertices in the clique that it looks for.*

Implement: Take l vertices, and compute the $\wedge$ for all the edges between them to see if for any 2 vertices in this group, there is a edge between them.

Definition 9 *A* **Clean function** *is an OR of at most m clique indicators of size less than l.*

Now comes the proof :
Idea: Every small monotone circuit can be approximated by a clean function.

We will prove this by an induction on the circuit, beginning with the inputs. Observe that the inputs to the circuit are clique indicators of size 2. We will show that the AND and OR of two functions that are approximated by clean functions can also be approximated by a clean functions.

But there is one problem, even if f and g are both clean functions, neither $f \wedge g$ nor $f \vee g$ need to be clean. So how do we solve this dilemma??Here is one natuaral approach.
Fix: we are going to define $\sqcup$ and $\sqcap$ such that $f \sqcup g$ and $f \sqcap g$ are clean if f and g are clean, and $f \sqcup g$ and $f \sqcap g$ approximate $f \vee g$ and $f \wedge g$ on the test graphs.

Note that if f and g are clean, then $f \vee g$ is the OR of at most 2m clique indicators. So to make $f \vee g$, we must somehow reduce the number of indicators. To reduce the number of indicator, we will need a result of Combinatoric Theory:*The Sunflower Theorem.*

Definition 10 *A* **sunflower** *is a collection of sets $S_1, S_2, ..., S_m$ such that*

$$\forall i < j, S_i \cap S_j = \bigcap_i S_i$$

We call $\bigcap_i S_i$ the "core" of the sunflower, and each S_j is called a "petal".

Theorem 11 (Sunflower theorem) *Let C be a collection of sets of size $\leq l$, and $|C| > (p-1)^l l!$, then C contains at least one sunflower with p petals.*

To define our $\sqcap$ and $\sqcup$, we first define some auxilary functions:

pluck(f) : let f be an OR of at least m clique indicators, pluck(f) is obtained by finding a sunflower among the clique indicators, discard those indicators, and adding their core as a new indicator.

weed(f) : pluck f until no more pluck possible.

cover(f,g) : Replace $indicator(S_i) \wedge indicator(T_j)$ with indicator($S_i \cup T_j$).

Trim(Cover($f \wedge g$)) :Cover($f \wedge g$) excluding indicators of size$\geq l$.

Now we can define the $\sqcap$ and $\sqcup$ operations:

Definition 12 *We define*

$$f \sqcup g = weed(f \vee g)$$

If $f = \underset{i}{OR}(indicator(S_i))$ and $g = \underset{i}{OR}(indicator(T_j))$, then

$$f \wedge g = \underset{i,j}{OR}(indicator(S_i) \wedge indicator(T_j))$$

.

So, we define

$$f \sqcap g = weed(trim(\underset{i,j}{OR}\ cover(indicator(S_i) \wedge indicator(T_j))))$$

In next lecture, we will proof the latter part of razborov theorem.