18.405J/6.841J Advanced Complexity Theory	April 7, 1998
Lecture 15	
Lecturer: Daniel A. Spielman	*Scribe: David Ratajczak*

1 Proof of Razborov's Theorem (cont.)

We start with a recap of the previous lecture. Razborov's Theorem asserts that any monotone circuit[1] that computes $CLIQUE_{\log n}$ has at least $n^{\Omega(\log n)}$ gates. This theorem (and the more recent Alon-Boppana theorem) establishes one of the most important circuit lower bounds presently known: NP-complete problems cannot be computed by polynomially-sized monotone circuits.

Razborov's approach is as follows: Consider the $CLIQUE_k$ problem[2] for a graph on n nodes. We wish to approximate a circuit that decides instances of $CLIQUE_k$ with a *clean* function. Recall from last time that a *clean* function is an $\vee$ of a set of no more than a fixed number of *clique indicators*, where each clique indicator of size l is just an $\wedge$ of at most $\binom{l}{2}$ inputs.

The inputs to the circuit are certainly clean functions (trivially, they are an $\vee$ of one clique indicator of size 1). We can thus approximate the whole circuit with a clean function by showing that if we are given two clean functions f and g, that $f \wedge g$ and $f \vee g$ can both be approximated by a clean function. Our circuit approximation would be constructed by recursively replacing each gate of the circuit with a clean function. We therefore define,

Approximate-$\vee$ ($\sqcup$) For every $\vee$ in the circuit, we can replace it with $\sqcup$ which is defined as:

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

where *weed* was defined to be: repeatedly *pluck* sunflowers with at least $p = (log(n))^3$ petals until at most $m = (p-1)^k k!$ clique indicators are left.

Approximate-$\wedge$ ($\sqcap$) For every $\wedge$ in the circuit, we replace it with $\sqcap$ defined as:

$$(f \sqcap g) = weed(trim(cover(f \wedge g))),^3$$

1.1 Proof Continued

In the last lecture we saw that every monotone function has a characteristic set of **minterms** and **maxterms**. We found that in the case of the k-clique problem.

minterms (C): graphs on n nodes with exactly one k-clique and no other edges.

[1] Circuits with only $\vee$ and $\wedge$ gates.

[2] The decision problem of determining whether or not a k clique exists in a graph of n nodes.

[3] Refer to the previous lecture for definitions of the *pluck*, *trim* and *cover* operations, as well as the Sunflower Theorem.

maxterms (D): complete $k-1$ partite graph (nodes divided into at most $k-1$ colors, and an edge exists between any two nodes of different colors.)

We are only going to analyze the performance of the Approximate-$\vee$ and Approximate-$\wedge$ operations on these two sets, rather than on all instances of the k-clique problem. We will show that for these two sets, a clean function will be unable to compute k-clique.

We also defined $\#_c(h)$ and $\#_d(h)$ to be the number of minterms and maxterms that are accepted by a clean function h. We found that

$$\begin{aligned}
\#_c(cover(h)) &= \#_c(h)\\
\#_d(cover(h)) &\leq \#_d(h)\\
\#_c(trim(h)) &\geq \#_c(h) - k\\
\#_d(trim(h)) &\leq \#_d(h)\\
\#_c(pluck(h)) &\geq \#_c(h)\\
\#_d(pluck(h)) &\leq \#_d(h) + n^{-\log(n)}|D|
\end{aligned}$$

From the definitions of $\sqcup$ and $\sqcap$ above, it is straightforward to compute:

$$\begin{aligned}
\#_c(f \sqcup g) &\geq \#_c(f \vee g) && (1)\\
\#_d(f \sqcup g) &\leq \#_d(f \vee g) + 2n(n^{-\log(n)})|D| && (2)\\
\#_c(f \sqcap g) &\geq \#_c(f \wedge g) - n^2 && (3)\\
\#_d(f \sqcap g) &\leq \#_d(f \wedge g) + n^2 n^{-\log(n)}|D| && (4)
\end{aligned}$$

Proof

1. Follows trivially from $f \sqcup g = weed(f \vee g)$.
2. During the *weed* operation, This part follows from Lemma 2 and the fact that during the *weed* operation we pluck the sunflower at most $2n$ times.
3. $(f \sqcap g) = weed(trim(cover(f \wedge g)))$. This result follows.
4. We only need to worry about the weed operation here, since it is the only operation that increases $\#_d$. We only need to pluck at most n^2 times after $trim(cover())$ has been applied. The above result follows.

■

The main idea is that the $\sqcup$ and $\sqcap$ approximations neither decrease the probability of accepting a maxterm by too much, nor increase the probability of accepting minterm by too much.

Lemma 1 *Let A be a monotone circuit with g gates that computes k-clique, where $k = log(log(n))$. If A' is the circuit obtained by replacing all $\wedge$'s and $\vee$'s with $\sqcap$'s and $\sqcup$'s respectively, then we have $\#_c(A) = |C|$, $\#_d(A) = 0$, and*

$$\#_c(A') \geq |C| - gn^2, \text{ and } \#_d(A') \leq gn^{-log(n)+2}|D|.$$

Proof This follows from applying the inequalities listed above for each of the g gates in the circuit A. ■

Lemma 2 *If h is a clean function, then either*

$$\#_d(h) = |D|, \text{ or } \#_c(h) \leq \frac{|C|}{2}.$$

Proof The first case occurs if h is an $\vee$ of no $\wedge$'s. This circuit accepts all graphs and therefore $\#_d(h) = |D|$. The second case occurs when h is an $\vee$ of at most n $\wedge$'s. For this case we choose a k-clique graph at random and check whether it is accepted by a particular $\wedge$.

The probability that a particular $\wedge$ accepts a particular k-clique (chosen randomly) is certainly less than the probability that a given edge (in the $\wedge$) is in the k-clique. This probability is just $\frac{\binom{n-2}{k-2}}{\binom{n}{k}}$ as this is the number of k-cliques that contain a given edge divided by the total number of k-cliques. We can simplify this and say that

$$\frac{\binom{n-2}{k-2}}{\binom{n}{k}} < \frac{k^2}{n^2}.$$

Since there are at most n $\wedge$ gates, then the probability that a random k-clique is accepted by the circuit is no greater than

$$\frac{k^2}{n^2} n = \frac{(log(log(n)))^2}{n} < \frac{1}{2}.$$

This means the number of maxterms accepted by h is less than half the total or $\#_c(h) \leq \frac{|C|}{2}$. ■

Combining the above two lemmas, we can see that if A is a monotone circuit with g gates that computes the k-clique problem, then

$$g \geq min\left(\frac{|C|}{2n^2}, n^{log(n)-2}\right) = min\left(\frac{\binom{n}{k}}{2n^2}, n^{log(n)-2}\right) = min\left(\frac{\binom{n}{log(log(n))}}{2n^2}, n^{log(n)-2}\right).$$

And this proves the desired lower bound of $n^{\Omega(log(n))}$ for monotone circuits computing $CLIQUE_{log(n)}$.

2 The Parity Function Revisited

We now revisit a familiar result which was established in a previous lecture.

Theorem 3 (Furst, Saxe, Sipser) *The parity function cannot be computed by constant depth, polynomial size, unbounded fan-in circuits.*[4]

We aim to reprove this result, but using techniques that were established in the original publication of this result by Furst, Saxe, and Sipser. In this lecture, we will outline the proof and produce some necessary intermediate results.

[4] Any NOT gates should only be at the input.

Proof Idea

In this proof we will use the idea of a *restriction*, where we fix some of the inputs of a circuit while leaving others variable.

We will prove the result by contradiction. We start by letting d be the smallest depth admitting polynomial size parity circuits. Now suppose we have a depth d circuit with alternating $\vee$'s and $\wedge$'s at each level, and the NOT gates moved to the inputs. It is not difficult to show that we can convert any constant depth circuit into this form with only a factor of 2 increase in its size. We will assume that the bottom level gates are $\wedge$ gates.

We first apply a random restriction (chosen from a suitable distribution) so that the gates on the bottom level have constant fan-in. Then we apply a second set of restrictions to this circuit to obtain constant fan-in in all gates on the bottom two levels. Finally, we will interchange the bottom two levels and collapse the circuit to a constant $d-1$ depth circuit. We will show that with high probability, this final circuit will also compute parity.

By repeating this method and applying new restrictions, we can collapse the circuit down to a depth 2 circuit where the bottom level has constant fan-in. Lupanov originally showed that no polynomially sized depth 2 circuits can compute parity, so this completes the proof.

Proof

Let us begin with depth d circuits C_1, C_2,..., where C_n computes the parity of its n inputs. We will consider a random restriction $\rho : x_1, x_2, \ldots, x_n \longrightarrow \{0, 1, *\}$ where for each i,

$$\begin{aligned} Pr[\rho(x_i) = *] &= \frac{1}{\sqrt{n}} \\ Pr[\rho(x_i) = 1] &= Pr[\rho(x_i) = 0] = \frac{1 - 1/\sqrt{n}}{2} \end{aligned}$$

We wish to show that there is a non-zero probability that the restriction ρ leaves more than $\sqrt{n}/2$ inputs variable but which bounds the fan-in of the bottom level $\wedge$ gates.

The probability that there are less than $\sqrt{n}/2$ inputs left variable is $O(n^{-1/2})$.[5]

Lemma 4 *Let G be a bottom level $\wedge$ gate, then for all $l > 0$ there is a constant c such that the probability that G has fan-in greater than c after the restriction ρ is less than n^{-l}.*

This result will imply that we can choose c large enough that the probability that a random restriction does *not* satisfy the above requirements approaches 0 as n increases. We will prove this lemma and complete the rest of the proof in the next lecture.

References

[1] Furst, Saxe, Sipser, *Parity, Circuits, and the Polynomial-Time Hierarchy.* Mathematical Systems Theory, vol. 17, no. 1, p. 13-27. April 1984

[5]Actually, you can say it is $O(2^{-c\sqrt{n}})$ for some constant c, but it is not necessary.