CS6840: Advanced Complexity Theory April 10, 2012

Lecture 52 : Monotone Circuit Lower Bounds

Lecturer: Jayalal Sarma M.N. *Scribe: Dinesh K.*

THEME: Circuit Complexity
LECTURE PLAN: Monotone functions and circuits. CLIQUE requires exponential size for any monotone circuit computing it. The overall strategy. Clique Indicators and Approximators. Positive and negative inputs. Sunflower Lemma.

In this lecture, we shall show a super polynomial lower bound for a function in NP computed by monotone circuits.

Recall the fundamental question that we were trying to answer in the scenario of P =?NP.

"Does a language in NP requires super polynomial sized circuits computing them ? "

Suppose we were able to show that there is an $L \in \mathsf{NP}$ that is not in PSIZE then by the results seen earlier,

- $\mathsf{PSIZE} = \mathsf{P/poly}$
- $\mathsf{P} \subseteq \mathsf{P/poly}$

we have $L \notin \mathsf{P/poly}$ and therefore not in P thereby separating P and NP. Hence the holy grail would be to find such a function in NPthat requires very large sized circuits. In this lecture, we show the following result.

Theorem 1. *Any monotone circuit computing clique number of graph requires exponential size.*

Note that if the above theorem can be proved for any general circuit then also it implies $\mathsf{P} \neq \mathsf{NP}$ since clique is NP-complete.

1 Monotone Circuits and Clique function

Recall the definition of monotone functions.

Definition 2. A function $f : \{0,1\}^n \to \{0,1\}$ is said to be monotone if $\forall x, y \in \{0,1\}^n$

$$x \leq y \iff f(x) \leq f(y)$$

where $\leq$ is defined on binary strings as usual.

If we consider the relation $\leq$ on $\{0,1\}^n$, they form a poset with 1^n at the top and 0^n at the bottom and strings in $\{0,1\}^n$ forms chains (The chains can be visualised as paths in an n-dimensional hypercube from corners 0^n to 1^n).

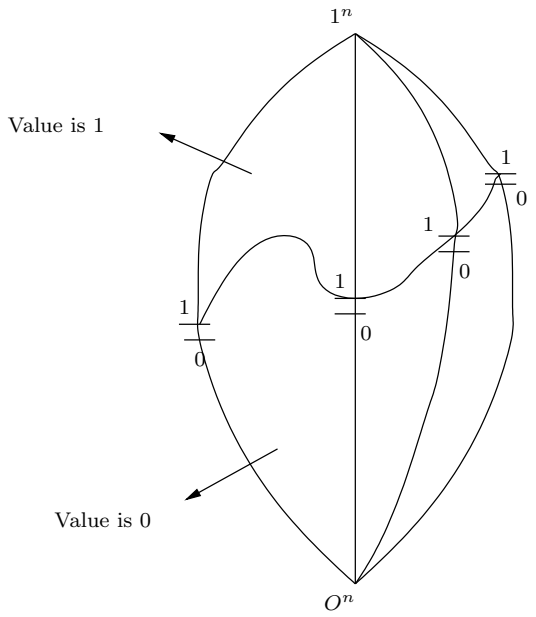

Figure 1: Monotone functions : A visualisation

Now, for any monotone function, there is a point where the function outcome changes from 0 to 1 and from there, as we go up the chain, the value remains the same.

Note that monotone circuits does not require NOT gates. This is because the functions that are of interest for us never change their value from 1 to 0 once they have reached one. Thus any monotone circuit requires only AND and OR gates gates. By Post's characterisation (lect), AND and OR gates can only compute monotone functions. Hence functions is monotone iff they are computed by AND and OR gates.

Also note that unlike our previous circuit characterisations where we had graph theoretic restrictions (size/depth/degree), for monotone circuits we are imposing a functional restriction.

Now consider the n^{th} slice of the $CLIQUE$ language.

Definition 3.

$CLIQUE_{k,n} = \{(x_1, x_2, \ldots, x_{\binom{n}{2}}) |$ Graph G represented by the tuple has a clique of size k $\}$

Here x_i corresponds to the i^{th} edge in the graph for $i \in \{1, 2, \ldots, \binom{n}{2}\}$.

The following claims holds true for $CLIQUE_{k,n}$.

Claim 4. *$CLIQUE_{k,n}$ is monotone.*

Proof. Key observation is that adding an edge does not destroys a k-clique. Also, adding an edge corresponds to a change in input bit from 0 to 1. Thus, if the input bit is flipped from 0 to 1, the outcome will continue to remain 1 if it is already 1 and never gets down to 0. Hence the claim follows. □

Claim 5. *There exists a trivial monotone circuit computing clique*

Proof. Construct a circuit that looks at k possible inputs, AND them and takes an OR over all the $\binom{n}{k}$ possibilities. It can be easily verified that the circuit computes clique function $CLIQUE_{k,n}$ and the size of the circuit will be $\binom{n}{k} \sim n^k$. □

For $k = n$, this will be super polynomial. It however turns out to be that we do require at least super polynomial sized monotone circuits to compute clique functions.

Theorem 6. *Any monotone circuit computing $CLIQUE_{k,n}$ must have a size at $n^{\Omega(\sqrt{k})}$.*

In particular for $k = n - 1$, the size must be at least $n^{\sqrt{n}} = 2^{\Omega(n \log n)}$

Before proceeding to the proof, let us ponder over the following thoughts.

- We now know of a super polynomial lower bound albeit for a special circuit class.
- Now, suppose we have a transformation function that can convert a polynomial sized circuit that uses NOT gates to another polynomial sized circuit without NOT gates computing the same function. What could this imply ?
- Existence of such a transformation directly imply $\mathsf{P} \neq \mathsf{NP}$. This is because if P were equal to NP then $CLIQUE$ has polynomial sized circuits computing it and by the size preservation transformation, we can get a monotone circuit computing $CLIQUE$ function of polynomial size. But this directly contradicts theorem 6.
- Does such a transformation exists ? The question has been asked by Razbarov and Wigderson who proved the following result.

Theorem 7. *There exists a monotone function in* P *that requires exponential size for any monotone circuit computing it.*

This rules out the existence of such a transformation from non-monotone to monotone circuits preserving the size.

- Monotone circuits seems to be some what handicapped essentially due to the absence of negation gates. Say if we allow constant number of negations gates, can we still prove the lower bound ?

  Observe that for any circuit computing an function f on n inputs needs $O(n)$ gates only. This is because all the negation gates can be pushed down to the leaf level by applying de-Morgan's law. Hence providing $O(n)$ negation gates will make the monotone function too powerful. Can we manage with $O(\log n)$ gates ? The answer to this question has been proved in affirmative by Fisher who showed,

  Theorem 8. *Any circuit computing f on n inputs needs only $O(\log n)$ negation gates.*

  Hence, if we are able to prove the lower bound result for monotone circuits allowing $O(\log n)$ gates then P and NP can be separated. (The current best bound is known with $O(\log(\log n))$ negation gates. That is, even with $\log(\log n)$ NOT gates $CLIQUE_{k,n}$ cannot be computed by monotone polynomial sized circuits.)

2 Building tools for proving theorem 6

The proof idea is the following. Let C be a monotone circuit computing $CLIQUE_{k,n}$.

1. Start by "approximating" each gate in the circuit C by a gadget that approximates the output of that gate.
2. Start approximating the leaf nodes and move up replacing each gate by an approximator ending up with an approximator C' for the root of the circuit.
3. We shall use this approximation to argue the size of C. First define two set of inputs called (a) *Positive inputs*, (b) *Negative inputs* and shall see the error that C' makes relative to C.
4. We shall argue that the error introduced by replacing each gate by the approximator is small.
5. We shall then argue that, overall, the circuit C' (that is the approximated output of root) is making too much errors.
6. Hence one can conclude that the number of gate in the circuit C must be too many in number.

Positive inputs They are graphs on n vertices having a k clique and $n-k$ isolated vertices. There are $\binom{n}{k}$ of them. On these graphs the original circuit C will evaluate to 1.

Negative inputs They are graphs on n vertices which are $(k-1)$ partite complete graphs (They contain $k-1$ cliques). Number of such graphs will be $(k-1)^n$ (Counting is based on the colouring of vertices, i.e $(k-1)$ ways to colour each of the vertices. Though this leads to over counting, we will be using this value only to lower bound errors).

The *clique approximator function* and (m, l) *approximator* is defined as follows.

Definition 9. For an $X \subseteq [n]$, clique indicator function,

$$I_X = \begin{cases} 1 & \text{If graph induced by subset } X \text{ forms a clique} \\ 0 & \text{Otherwise} \end{cases}$$

An (m, l) approximator is defined as,

$$\bigvee_{i=1}^{r} I_{X_i}$$

where $X_1, X_2, \ldots, X_r \subseteq [n]$, $|X_i| \leq l$, $r \leq m$.

We shall be requiring the following lemma proved by Erdos and Rado.

Lemma 10. *(Sunflower lemma) A sunflower of p petals is defined as, $Z_1, Z_2, \ldots, Z_p \subseteq [n]$ such that $\forall\ i, j, i \neq j, Z_i \cap Z_j = Z$. The sets $\{Z_i\}$ are the petals and Z is called as the* core.

Let $F \subseteq 2^{[n]}$, integers l, p, such that for all $S_i \in F$, $|S_i| \leq l$ and $|F| > (p-1)^l \times l!$, then there is a sunflower of p petals.

Proof. (By induction on l) For $l = 1$, we have, $|S_i| \leq 1$ and $|F| > (p-1)$. Hence $|F| \geq p$. The p petals will be the singleton sets, each containing an element of F, with core being empty. Hence the base case holds.

Suppose the result holds for $l > 1$. That is, for $F \subseteq 2^{[n]}$, $\forall\ i, S_i \in F$ has $|S_i| \leq l$ and $|F| > (p-1)^l \times l!$, there is a p-petal sunflower.

Now, consider the $F \subseteq 2^{[n]}$ with $|F| > (p-1)^{l+1} \times (l+1)!$ with every $|S_i| \leq (l+1)$. Let $Z_1, Z_2, \ldots, Z_q$ be the maximal number of disjoint sets in F. If $q \geq p$, then we already have p petals with an empty core and the lemma holds for $(l+1)$.

If $q \leq p-1$, consider the set

$$Z = \bigcup_{i=1}^{p-1} Z_i$$

Also

$$|Z| = \sum_{i=1}^{p-1} |Z_i| \leq (p-1)(l+1)$$

To reduce to the inductive case, we need to show that there exists an element in Z that appears in "many" of the sets in F. To do this, observe that every element in Z will be covered in some set of F which follows from maximality of $Z_1, Z_2, \ldots, Z_q$[1]. Now, the average number of sets in which any element in Z appears is

$$\frac{|F|}{|Z|} > \frac{(p-1)^{l+1}(l+1)!}{(p-1)(l+1)} = (p-1)^l l!$$

By pigeon hole principle, there must be an element $e \in Z$ that appears in at least $(p-1)^l \times l!$ sets. Now look at the sets F' obtained by removing e from all the sets in F containing e. Hence $|F'| > (p-1)^l \times l!$ and $|S'_i|$ will be at most l. By induction hypothesis, there is sunflower of p petals in F'. Now by adding back the element e to the petals, we get a sunflower to the set F. □

[1]If there exists a set Z' in F that does not contain any element in Z, then $\{Z_1, Z_2, \ldots, Z_q, Z'\}$ will form $q+1$ disjoint sets contradicting the maximality of Z_is