PRINCETON UNIVERSITY F'02 COS 597D: A THEORIST'S TOOLKIT

Lecture 5: Circuit Lower Bounds and the Lore and Lure of Expanders

Lecturer: *Sanjeev Arora* Scribe:*Paul Chang*

Today, we first prove another result that is an example of the dimension method: the Razborov-Smolensky method for proving circuit lower bounds. Then we start our study of expander graphs.

1 The Razborov-Smolensky Circuit Lower Bound

We are very far from separating **NP** from **P**; the turing machine model seems too complex to reason about. It is natural to try to prove an exponential lower bound for a restricted model of computation. The Razborov-Smolensky lower bound applies to one such model. The theorem presented in this section was originally proved by Razborov using his *Method of Approximations* (he won the Nevanlinna prize in 1990 for this work). Smolensky later extended this work and clarified this method for the circuit class considered here.

DEFINITION 1 *A circuit C is a directed acyclic graph with n input nodes, labeled x_1 to x_n, and one output node. All other nodes are labeled with a boolean operation. When we label the input nodes with n bits and let the other nodes compute in the obvious manner (computing the boolean operation on their incoming edges and placing the 1-bit result on the outgoing edges) we get a boolean value on the output node, denoted $C(x)$. The size of a circuit is the number of nodes. Note that the circuit nodes may have unbounded degree.*

DEFINITION 2 *A family of circuits is a collection of circuits $\{C_n\}_{n\geq 1}$, such that the nth circuit has n input nodes. We say that a circuit family computes a function $f : \{0,1\}^* \to \{0,1\}$ if for all x, $C_{|x|}(x) = f(x)$.*

REMARK 1 A circuit family can compute undecidable languages - we may hardwire into a circuit any of 2^n functions on n inputs, and may use different circuits for each input length. (For example, there exists a circuit family which computes $\left\{1^{<M,w>} | \text{M halts on w}\right\}$).

We are interested only in a subclass of circuits, the $\mathbf{MOD_3}$ circuits.

DEFINITION 3 *A $\mathbf{MOD_3}$ circuit of depth d on n inputs is one whose depth is bounded by a constant d. Each boolean gate performs any of the four operation AND ($\wedge$), OR ($\vee$), NOT ($\neg$), and sum modulo three (MOD_3). The MOD_3 gate outputs zero if the sum of its inputs is zero modulo three, and one otherwise.*

The inclusion of the MOD_3 gates gives the circuit some power. Nevertheless we show that it cannot compute the MOD_2 function, namely, parity of n bits.

THEOREM 1
The Razborov/Smolensky Circuit Lower Bound. *Computing MOD_2 with MOD_3 circuits of depth d requires a circuit of size* $\exp(\Omega(n^{\frac{1}{2d}}))$.

In the rest of the section we prove this theorem. The proof proceeds in two steps.

Step 1. In the first step, we show (using induction on h) that for any depth h MOD_3 circuit on n inputs and size S, there is a polynomial of degree $(2l)^h$ which agrees with the circuit on $1 - S/2^l$ fraction of the inputs. If our circuit C has depth d then we set $2l = n^{1/2d}$ to obtain a degree $\sqrt{n}$ polynomial that agrees with C on $1 - S/2^{n^{1/2d}/2}$ fraction of inputs.

Step 2 We show that no polynomial of degree $\sqrt{n}$ agrees with MOD_2 on more than $49/50$ fraction of inputs.

Together, the two steps imply that $S > 2^{n^{1/2d}/2}/50$ for any depth d circuit computing MOD_2, thus proving the theorem. Now we give details.

Step 1. Consider a node i in the circuit at a depth h . (The input is assumed to have depth 0.) If $f_i(x_i, \cdots, x_n)$ is the function computed at this node, we desire a polynomial $\tilde{f}_i(x_i, \cdots, x_n)$ over $GF(3)$ with degree $(2l)^h$, such that on any input in $\{0,1\}^n \subseteq GF(3)$, polynomial $\tilde{f}_i$ produces an output in $\{0,1\}$. (In other words, even though $\tilde{f}_i$ is a polynomial over $GF(3)$, for 0/1 inputs it takes values in $\{0,1\} \subseteq GF(3)$.) Furthermore, we desire

$$\mathbf{Pr}_{x \in \{0,1\}^n}[f_i(x) \neq \tilde{f}_i(x)] \leq \left(\frac{\text{circuit size}}{2^l}\right) \tag{1}$$

We construct the approximating polynomial by induction. The case $h = 0$ is trivial. Suppose we have replaced the output of all nodes up to height h with polynomials of appropriately bounded degree and error. We wish to approximate the output of a node g of height $h + 1$ with immediate inputs $f_1, \cdots, f_k$.

1. If g is a NOT gate, we may write $g = \neg f_i$. Then, $\tilde{g} = 1 - \tilde{f}_i$ is an approximating polynomial with the same degree and error as that of f_i.

2. If g is a MOD_3 gate, we use the approximation $\tilde{g} = (\sum_i \tilde{f}_i)^2$. The degree increases by at most a factor of 2, and there is no increase in the error rate.

3. If g is an AND or an OR gate, the required approximation is slightly harder to produce. Suppose $g = \wedge_{i \in I} f_i$. The naive approach approximates an AND node g with a polynomial $\Pi_{i \in I} \tilde{f}_i$. If $g = \vee_{i \in I} f_i$ we use De Morgan's law and similarly obtain the naive approximator $1 - \prod_{i \in I}(1 - \tilde{f}_i)$. Unfortunately, both of these increase the degree of the polynomial by a factor of $|I|$, which could be much larger than the allowed $2l$.

   Let us give the correct solution for OR, leaving the case of AND to yet another application of De Morgan's laws.

   If g is an OR gate, then $g = 1$ if and only if at least one of the $f_i = 1$. We observe that if any of the $f_i = 1$, the sum (over $GF(3)$) of a random subset of $\{f_i\}$ is one with probability at least $1/2$.

   Pick l subsets $S_1, \cdots, S_l$ of $\{1, \cdots, k\}$ randomly. We compute the l polynomials $\sum_{j \in S_i} (\tilde{f}_j)^2$, each of which has degree at most twice that of the largest input polynomial. We then compute the OR of these l terms using the naive approach. The result

is a polynomial with degree at most $2l$ times that of the largest input polynomial. For any x, the probability over the choice of subsets that this polynomial differs from $OR(\tilde{f}_1, \cdots, \tilde{f}_k)$ is at most $\frac{1}{2^l}$. So, by the expectation argument, there exists a choice for the l subsets such that the probability over the choice of x that this polynomial differs from $OR(\tilde{f}_1, \cdots, \tilde{f}_k)$ is at most $\frac{1}{2^l}$.

(There was a question in class about how the errors at different gates affect each other. An approximator may introduce some error, but another approximator higher up may introduce another error which cancels this one. The answer is that we are ignoring this issue, and using just the union bound to *upperbound* the probablity that any of the approximator polynomials anywhere in the circuit miscompute.)

This completes the inductive construction of the proper polynomial.

Step 2. Suppose that a polynomial f agrees with the MOD_2 function for all inputs in a set $G' \subseteq 0, 1^n$. If the degree of f is bounded by $\sqrt{n}$, then we show $|G'| < \left(\frac{49}{50}\right)2^n$.

Consider the change of variables $y_i = 1 + x_i \pmod 3$. (Thus $0 \to 1$ and $1 \to -1$.) Then, G' becomes some subset of $\{-1, 1\}^n$, and f becomes some other polynomial, say $g(y_1, y_2, \ldots, y_n)$, which still has degree $\sqrt{n}$. Moreover,

$$MOD_2(x_1, x_2, \ldots, x_n) = \begin{cases} 1 & \Rightarrow \Pi_{i=1}^n y_i = -1 \\ 0 & \Rightarrow \Pi_{i=1}^n y_i = 1 \end{cases} \tag{2}$$

Thus $g(y_1, y_2, \ldots, y_n)$, a degree $\sqrt{n}$ polynomial, agrees with $\Pi_{i=1}^n y_i$ on G. This is decidedly odd, and we show that any such G must be small. Specifically, let F_G be the set of all functions $S : G \to \{0, 1, -1\}$. Clearly, $|F_G| = 3^{|G|}$, and we will show $|F_G| \leq 3^{\left(\frac{49}{50}\right)2^n}$, whence Step 2 follows.

Lemma 2

For every $S \in F_G$, there exists a polynomial g_S which is a sum of monomials $a_I \prod_{i \in I} y_i$ where $|I| \leq \frac{n}{2} + \sqrt{n}$ such that $g_S(x) = S(x)$ for all $x \in G$.

Proof: Let $\hat{S} : GF(3)^n \to GF(3)$ be any function which agrees with S on G. Then $\hat{S}$ can be written as a polynomial in the variables y_i. However, we are only interested in its values on $(y_1, y_2, \ldots, y_n) \in \{-1, 1\}^n$, when $y_i^2 = 1$ and so every monomial $a\Pi_{i \in I} y_i^{r_i}$ wlog has $r_i \leq 1$. Thus wlog, $\hat{S}$ is a polynomial of degree at most n. Now consider any of its monomial terms $\Pi_{i \in I} y_i$ of degree $|I| > n/2$. We can rewrite it as

$$\Pi_{i \in I} y_i = \Pi_{i=1}^n y_i \Pi_{i \in \bar{I}} y_i, \tag{3}$$

which takes the same values as $g(y_1, y_2, \ldots, y_n)\Pi_{i \in \bar{I}} y_i$ over $\{-1, 1\}^n$. Thus wlog every monomial in $\hat{S}$ has degree at most $\frac{n}{2} + \sqrt{n}$. □

To conclude, we bound the number of polynomials whose every monomial with a degree at most $\frac{n}{2} + \sqrt{n}$. Clearly this number is $\#\text{polynomials} \leq 3^{\#\text{monomials}}$, and

$$\#\text{monomials} \leq \left|\{N \subseteq \{1 \cdots n\}\,|\,|N| \leq \frac{n}{2} + \sqrt{n}\right| \tag{4}$$

$$\leq \sum_{i \leq \frac{n}{2} + \sqrt{n}} \binom{n}{i} \tag{5}$$

Using knowledge of the tails of a binomial distribution,

$$\leq \frac{49}{50}2^n \tag{6}$$

2 The Lore and Lure of Expanders

Now we move to the study of Expander graphs. Their numerous applications to a areas such as error correcting codes, routing networks, sorting networks, derandomization, and PCP reductions make them indispensable in any theorist's toolkit.

DEFINITION 4 *An (α, β) expander is a bipartite graph such that for all $\alpha\beta < 1$, each set S of nodes on the left where $|S| \leq \alpha n$ has at least $\beta |S|$ neighbors.*

As shown in an earlier problem set, for every α and β there exists a d such that almost all d-regular bipartite graphs on n nodes are (α, β) expanders. Today we show one example of such graphs for telephone switching networks.

2.1 Lockdown routing/circuit switching.

This example is partially motivated by the study of massively parallel computers and partly by telephone networks. We model a communications network as a graph with N inputs and N outputs. We wish to route calls from the input users to the output users through routers represented by internal nodes. Each call occupies the routers it uses, so separate calls must take vertex disjoint paths. In particular, we would like to be able to route any permutation matching inputs to outputs with decentralized control. In this section we construct a network in which all permutations can be routed. In the next lecture, we will decentralize the control.

REMARK 2 Any network which can route all of the permutations has size $\Omega(NlogN)$, since

$$N! \leq \#states \leq d^{\#nodes} \tag{7}$$

DEFINITION 5 *A* Delta network *on N inputs is defined recursively: it consists of the first level, which contains the N input nodes, and two subnetworks that are Delta networks on $\frac{N}{2}$ inputs, and are called the* up *and* down *networks respectively. Each input node has edges to inputs in both up and down networks. Outputs whose address has the MSB (most significant bit) 1 lie in the up subnetwork and the remaining outputs are in the down subnetwork. Clearly, any call to a destination whose MSB is* 1 *must go from the input node to some node in the up subnetwork. Reasoning similarly for all the levels, we see that the delta network is a layered network in which the up/down decision at level i depends on the ith bit of the destination node. We may route to any of the input nodes of the smaller networks.*

An example of the Delta network is the butterfly*, where each input node $(b_0 b_1 \ldots b_{\log N})$ has exactly two outgoing edges, one each to the up and down subnetwork. The edges go to nodes whose label in the that subnetwork is $(b_1 b_2, \ldots, b_{\log N})$.*

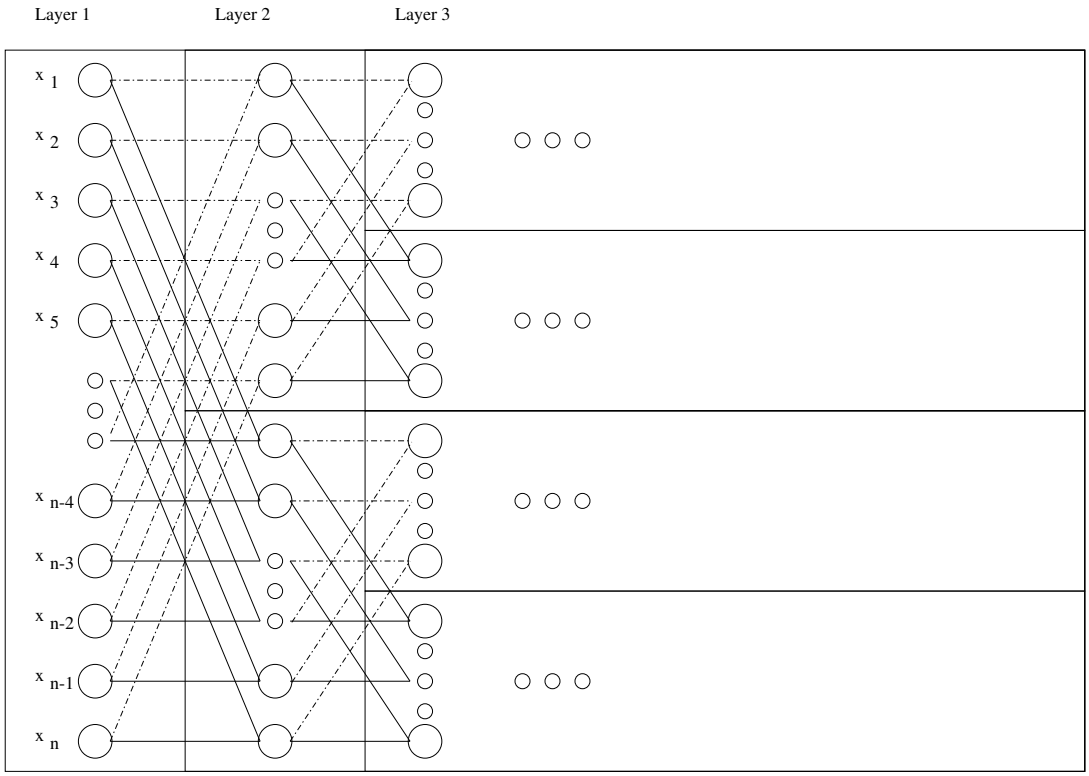

Figure 1: An example delta network. Each layer contains a dotted set of edges up and a solid set of edges down. In the multibutterfly, each of these sets forms an expander graph.

Butterfly networks cannot route all permutations with vertex disjoint paths. For instance, they function poorly on the transpose permutation. Instead, we present the multibutterfly network of Upfal '88. In the butterfly network, each of N nodes in the input layer is connected to one of the $\frac{N}{2}$ top nodes of the next layer. In the multibutterfly, the degree of that node is d and the subgraph of edges from each of N nodes to the $\frac{N}{2}$ top nodes of the next layer is an (α, β) expander for $\alpha\beta < \frac{1}{2}$.

We assume that the number of inputs (also, number of outputs) is $2\alpha N$ and each is connected to $1/2\alpha$ successive inputs (resp., outputs) of the multibutterfly. (Alternatively, we may still use N for the number of inputs and outputs and then connect them using multibutterflies with $N/2\alpha$ inputs.)

We claim that any permutation on these $2\alpha N$ inputs/outputs can be routed using vertex disjoint paths. We do this layer by layer. At most $\alpha N/2$ calls need to be routed to the top (or bottom) subnetwork. Let S be the set of nodes at which these calls originate. We can route these calls in a vertex disjoint manner if there exists a matching between this subset S of input nodes and any subset S' of nodes in the top network. To see that this can be done, we apply Hall's theorem, noting that the expander property guarantees that S has at least $|S|$ neighbors.

THEOREM 3
Hall's Theorem. *If $G = (V_1, V_2, E)$ is a bipartite graph, there exists a perfect matching involving V_1 if and only if for all subsets $S \subseteq V_1$, $|\Gamma(S)| \geq |S|$.*

PROOF: Apply the max-flow min-cut theorem. □

Repeating for all levels we obtain our vertex disjoint paths.