CS 6810: Theory of Computing **Fall 2023**

Lecture 13-15: Oct 3, 5, and 12, 2023

Lecturer: Eshan Chattopadhyay *Scribe: Andy He, Ishan Bansal, Stephanie Ma*

1 AC^O Lower Bounds via the Switching Lemma

Theorem 1.1. *If an AC^0 circuit C (with size s and depth t) computes PARITY, then $S(n) \geq 2^{\Omega(n^{1/t})}$.*

The proof of this theorem will come later, when we discuss the switching lemma. Let us first introduce the notion of random restrictions on circuits.

Definition 1.2 (Restriction)**.** *A restriction R is described by a set S of inputs that are free (or alive), and an assignment $z \in \{0,1\}^{\bar{S}}$ to the bits in $\overline{S}$ that are fixed.*

Definition 1.3 (Random Restriction)**.** *Let $\mathcal{R}_l$ be the set of all restrictions with exactly l alive bits. $R \sim \mathcal{R}_l$ (chosen uniformly at random) is called a random restriction.*

Definition 1.4 (Restricted Function)**.** *Let f be a computable function and R be a random restriction that fixes all bits in some set $\overline{S}$. Then $f|_R : \{0,1\}^S \to \{0,1\}$ is the function obtained by fixing the bits in $\overline{S}$ according to the restriction R.*

1.1 Decision Trees

Let us define another (non-uniform) model of computation, Decision Trees. A decision tree is a binary tree where each internal node is labeled by a variable, and each leaf is labeled by 0 or 1. The tree is traversed by starting at the root, and at each internal node, we query the variable at that node. If the variable is 0, we go to the left child, otherwise we go to the right child. We output the label of the leaf we reach.

The depth of the decision tree is the length of the longest path from root to leaf, and the size is the number of edges.

Suppose DT T computes PARITY on n bits, then the depth of the tree must be at least n, since we need to query all bits. Further, a tree with depth n can compute any function on n bits, so the depth of the tree is exactly n.

1.2 Switching Lemma

Recall that a disjunctive normal form (DNF) ϕ is a disjunction (OR) of terms, where each term is a conjunction (AND) of literals. ϕ is a width w-DNF if no term contains more than w literals.

Theorem 1.5 (Håstad's Switching Lemma)**.** *Let f be computable by a width w-DNF. For any $p \leq 1/10$, $d \in \mathbb{N}$*

$$\mathbb{P}_{R \sim \mathcal{R}_{pn}}[DT_{depth}(f|_R) \geq d] \leq (5pw)^d$$

where $DT_{depth}(f|_R)$ is the depth of the shallowest decision tree that computes $f|_R$.

Before proving the above theorem, here is an example application.

Example 1.6. *Let $p = \frac{1}{10w}$, $d = \log n$, then*

$$\mathbb{P}_{R\sim\mathcal{R}_{pn}}[DT_{depth}(f|_R) \geq d] \leq (5pw)^d = (5w \cdot \frac{1}{10w})^{\log n} = 1/n.$$

Thus, given a width-w DNF, a random restriction with $p = 1/10w$ will result in the restricted function $f|_R$ to be computable by a decision tree of depth $\log n$ with very high probability.

The proof of the switching lemma we shall see is due to Razborov.

Proof. Let $f = T_1 \vee T_2 \vee \ldots \vee T_\ell$ be a DNF where each T_i is AND of at most w literals. Let $r = pn$ be the number of alive variables, and $\mathcal{R}_r$ be the set of restrictions with r alive variables. Define the subset BAD $\subset \mathcal{R}_r$ to be all restrictions $R \in \mathcal{R}_r$ such that $DT_{\text{depth}}(f|_R) \geq d$. We want to show that $|\text{BAD}|/|\mathcal{R}_r| \leq (20pw)^d$. The proof will be based on a clever encoding argument.

Consider an encoding H from BAD to some set E. Observe that H must be injective for the encoding to be decodable. Thus, $|\text{BAD}| \leq |E|$. We will show that E can be taken as $\mathcal{R}_{r-d} \times \text{Aux}$, where Aux is a set of size $(\beta_2 w)^d$. Observe that $|\mathcal{R}_r| = \binom{n}{r}2^{n-r}$ and $|\mathcal{R}_{r-d}| = \binom{n}{n-r+d}2^{n-r+d}$, so

$$\frac{|\mathcal{R}_{r-d}|}{|\mathcal{R}_r|} = \frac{r(r-1)(r-2)\ldots(r-d+1)}{(n-r+d)(n-r+d-1)\ldots(n-r+1)}2^d \leq \left(\frac{2r}{n-r+d}\right)^d = (\beta_1 p)^d$$

Thus, the size of the image of the encoding is at most $(\beta_1 p)^d \cdot (\beta_2 w)^d = (\beta_1\beta_2 pw)^d$, and we can tune the constants β_1 and β_2 as required.

We now show that such an encoding exists. We construct a "canonical DT"(CDT) that computes $f|_R$. For any term T_i in the DNF formula of f, if a single bit was set to False, then the term T_i will evaluate to False. Hence the only terms T_i 'alive' after the random restriction R are those where no bit is set to False and at least one bit is not yet set. Let T_{i_1} be the first term alive in $f|_R$, and U_1 be the alive variables in T_{i_1}. Construct a (partial) decision tree that queries (in lexicographical order) each of the alive variables in U_1. There is exactly one path in this decision tree whose assignments satisfy T_{i_1} (since every literal in the formula T_{i_1} must be set to True). Set the corresponding leaf to evaluate to True (since the formula f will now be satisfied). For every other leaf, we add in the restriction of the variables in U_1 obtained by following the path from the root to this leaf and we recurse, i.e., we obtain a new restriction $R \cup R'$ for each leaf respectively and we look at the first alive term in $f|_{R\cup R'}$ and repeat until no undecided leaf exists. We call the resulting decision tree the "canonical DT" that computes $f|_R$.

Let $R \in$ BAD, and construct a CDT T for R by the above procedure. Let us define σ as the leftmost path in T of length at least d (and truncated to length d by triming the last few variables at the end of the path). Such a path must exist since $R \in$ BAD, and thus has a CDT of at least depth d. We want to construct a map $R \to R \circ \sigma$. Observe that $R \in \mathcal{R}_s$ and $R \circ \sigma \in \mathcal{R}_{s-d}$, but we do not have enough information to decode any mapping. Hence, we introduce an auxiliary set Aux that can provide "hints" to decode. Observe that the mapping to $\mathcal{R}_{s-d} \times \sigma$ trivially decodes H, but does not restrict the size of $\mathcal{R}_s$.

Let us introduce some notation that will be useful. Observe that the CDT was constructed recursively in stages $i = 1, 2, \ldots$. Let σ_i denote the path taken in the ith stage of the CDT. Similarly, let π_i be the restriction that corresponds with a satisfying assignment for the first alive term at stage i.

We can be more clever by letting H map R to $R\circ\pi_1$ and then decoding σ_1 from our hint. Given a mapping in $R \circ \pi_1$, we are able to determine which term is T_1, since T_1 must be the first term that is satisfied under the restriction $R \circ \pi_1$. We can thus encode the indices of alive variables in

T_1 in at most $|U_i| \log w$ bits, and the values of the alive variables in $|U_i|$ bits. Thus, the hint to decode σ_1 from $R \circ \pi_1$ is simply $v_1 a_1 v_2 a_2 |$, where v_i denotes the value of alive variable a_i and $|$ is an arbitrary character that delimits this stage of the decoding. Repeat this process of reconstructing $\sigma_1, \sigma_2, ...$ from $\pi_1, \pi_2, ...$ by iteratively solving for σ_i and setting the restriction for the next stage, i.e., with the hint we can decode σ_2 from $R \circ \sigma_1 \circ \pi_2 \cdots$.

In summary, let $\pi = \pi_1 \circ \cdots$, we can construct a function such that $R \mapsto (R \circ \pi, \beta)$, where $R \in R_s$, $R \circ \pi \in R_{s-d}$, and $\beta = (\beta_1, \cdots) \in \text{Aux}$, β_i is the hint for stage i. The function is injective because we can reconstruct R in the abovementioned procedure. Each β_i takes $|U_i|(1 + \log w)$ bits to represent, so β takes $(1 + \log w) \sum_i |U_i| = (1 + \log w)d$ bits to complete in total. Hence, $|\text{Aux}| = 2^{(1+\log w)d}$.

$$\begin{aligned}
&\frac{|\mathcal{R}_{pn-d} \times \text{Aux}|}{|\mathcal{R}_p n|} \\
\leq &\left(\frac{2pn}{pn - r + d}\right)^d \times 2^{(1+\log w)d} \\
= &\left(\frac{4npw}{n - pn + d}\right)^d \\
\leq &(\frac{40}{9} pw)^d \leq (5pw)^d
\end{aligned}$$

Therefore, we can conclude

$$\mathop{\mathbb{P}}_{R \sim \mathcal{R}_{pn}} [DT_{\text{depth}}(f|_R) \geq d] \leq (5pw)^d.$$

□

1.3 Application of the Switching Lemma to get AC0 lower bounds

Claim 1.7. *Let f be computable by a depth w DT, then f is computable by a width w DNF/CNF.*

Proof. Consider a decision tree of depth w, for each path that leads to a 1, write a conjunction that corresponds to the path, then take the disjunction of all these conjunctions. This is a width w DNF.

Observe also that $\neg f$ also has a decision tree, we can construct a CNF by taking the negation of the DNF of the decision tree of $\neg f$. □

We can now prove the theorem stated earlier.

Claim 1.8. *If an AC^0 circuit C (with size S and depth t) computes PARITY, then $S(n) \geq 2^{\Omega(n^{1/t})}$.*

Proof. Let C be a circuit (size s, depth t) that decides parity which is alternating and has fan-out 1. Note that any circuit can be made into fan-out 1 by introducing duplicate gates with polynomial overhead on the size. The assumption that the circuit is alternating (has or layers followed by and layers) can be made without loss of generality. Assume that the bottom layer of gates (layer $t-1$) are all AND gates and the layer above that (layer $t-2$) are all OR gates. Thus layer $t-2$ is a series of DNFs.

We can further assume that the bottom layer of gates (layer $t-1$) has fan-in of $\leq \alpha \log S$. If this is not the case, we can apply a random restriction to the circuit to reduce all large fan-in circuits in the bottom layer to constants (with high probability) as follows: For each variable i,

leave it unrestricted with probability q, restrict it to 0 with probability $(1-q)/2$ and restrict it to 1 with probability $(1-q)/2$. Then a bottom gate with fan-in $\geq \alpha \log S$ is unfixed if none of the variables are set to 0. The probability of this is smaller than $(1-\frac{1-q}{2})^{\alpha \log S} = \frac{1}{poly(S)}$ and so we can apply a union bound to fix all large fan-in circuits in the bottom layer (with high probability). Furthermore, the probability that at least nq/k variables survive grows exponentially with k and so with high probability, a constant factor of the variables still survive.

After all these valid assumptions, layer $t-2$ is a series of w-DNFs, where $w = \alpha \log S$. Given such a circuit C, we will apply a restriction $R \sim R_{pn}$ to form circuit $C|_R$. Let $p = 1/40w$, $d = w$. Let ϕ_i be a DNF in layer $t-2$. By the switching lemma, we have that

$$\mathbb{P}_{R \sim \mathcal{R}_{pn}}[DT_{\text{depth}}(\phi_i|_R) \geq d] \leq (20pw)^d = \frac{1}{2^d} = \frac{1}{S^\alpha}$$

From union bound, it follows that the probability a random restriction collapses all circuits Q_i to a DTree of depth less than d is strictly positive. Thus there exists restriction R such that $\forall i, DT_{\text{depth}}(\phi_i|_R) \leq d$. We can thus collapse each of these ϕ_i to decision trees of depth d, and hence to a CNF of width d. But now layers $t-2$ and $t-3$ both comprise of only AND gates and so can be collapsed into one layer, of width $w = d = \alpha \log S$ CNFs. This gives us a new circuit $C|_R$ which has pn alive variables and depth $t-1$ and all bottom gates still have fan-in at most $\alpha \log S$.

We can keep applying this same procedure $t-2$ times to obtain a circuit $C|R'$ with depth 2 and at least $p^{t-2}n$ alive variables. This circuit will be a DNF (or a CNF). Then apply one more random restriction to reduce the final DNF to a decision tree. By the switching lemma, this decision tree must have depth at most $\alpha \log S$. The depth of this decision tree must be at least the number of alive variables $p^{t-1}n$ since it computers PARITY on so many bits. Hence,

$$\begin{aligned}
&\alpha \log S \geq p^{t-1}n \\
&\implies \alpha \log S \geq \left(\frac{1}{40\alpha \log S}\right)^{t-1} n \\
&\implies (\beta \log S)^t \geq n \\
&\implies S \geq 2^{\Omega(n^{1/t})}
\end{aligned}$$

This completes the proof. □