CS 810: Introduction to Computational Complexity Theory 03/13/2003

Lecture 16: Decision-Tree Complexity, Switching Lemma

Instructor: Jin-Yi Cai Scribe: Michael DeCoster, Vinod Ganapathy, and Charles Kahn

In this lecture, we will discuss the notions of decision tree complexity and random restrictions. These will be useful in proving the switching lemma and circuit lower bounds for parity function.

1 Decision Trees

Decision trees are used to represent boolean functions.

Definition 1 (Decision trees). *A* decision tree *is a rooted binary tree t with every leaf labeled as 0 or 1. Every internal node is labeled with an x_i. The two edges from an internal node are labeled with* 0 *and* 1 *respectively. We do not disallow variables to repeat along a path from the root to the leaf.*

Definition 2 (Decision tree depth). *The* depth *of a decision tree is defined to be the number of edges in the longest path from the root to a leaf.*

Figure 1 shows an example of a decision tree.

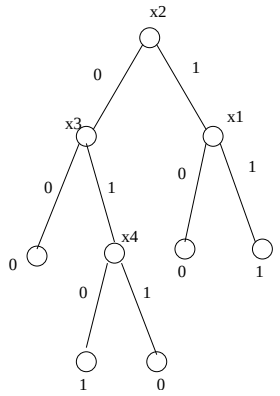

Figure 1: An example of a decision tree

A decision tree T defines a boolean function as follows:

1. if depth$(T) = 0$, then it is a constant function (either 0 or 1)
2. if depth$(T) = d + 1$ $(d \geq 0)$: inductively assume that the left subtree and the right subtree respectively define a boolean function. Call the boolean function defined by the left subtree G and the right subtree as H. Also assume that the root is labeled as x_i and has the left edge labeled 0 and the right edge labeled 1. Then,

$$F(x_1, x_2, \ldots, x_n) = \begin{cases} G(x_1, x_2, \ldots, x_n), \text{ if } x_i = 0 \\ H(x_1, x_2, \ldots, x_n), \text{ if } x_i = 1 \end{cases}$$

Both G and H are also functions in x_i, but if we disallow x_i from repeating along any path from root to leaf in the decision tree, we can leave it out of the functions G and H. Then instead, we can write G as a function of $x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$, which is often written as $G(x_1, x_2, \ldots \hat{x_i}, \ldots, x_n)$

Hence, the function F is in fact:

$$F = (G \wedge \bar{x_i}) \vee (H \wedge x_i)$$

We can instead choose to write G as $F|_{x_i=0}$ and H as $F|_{x_i=1}$ and rewrite the above equation as:

$$F = (F|_{x_i=0} \wedge \bar{x_i}) \vee (F|_{x_i=1} \wedge x_i)$$

This expansion gives us a way to write the formula in DNF (i.e., $\vee$ of $\wedge$'s). If $F|_{x_i=0}$ and $F|_{x_i=1}$ can be written as a DNF where every conjunct has size at most t, then we get a formula F in DNF in which every conjunct is of size at most $t+1$.

Definition 3 (Decision tree complexity). *For boolean function f, the decision tree complexity, $DC(f)$, is the minimum depth of the decision trees that compute f.*

Lemma 1. $D(f) = D\left(\bar{f}\right)$

Proof: The set of boolean decision trees that compute f are in 1-1 correspondence with those that compute $\bar{f}$. (Just flip the bits at the leaves of the tree.) □

As demonstrated in the previous section, a boolean function computed by a decision tree of depth d can be expressed as a boolean formula in disjunctive normal form (DNF), where each conjunct has size at most d. Using that idea, we get the following corollary.

Corollary 1. *A boolean function f and its complement $\bar{f}$ can be expressed in DNF in which every conjunct has size $\leq DC(f)$. So f can be written as a DNF formula with conjuncts of size $\leq DC(f)$, and as a CNF formula with disjuncts of size $\leq DC(f)$.*

The above mentioned special case of DNF where every conjunct is bounded in size will be useful in our discussion. The following definition spells this out.

Definition 4 (t-AND-OR). *Boolean formula G is a t-AND-OR if $G = G_1 \wedge G_2 \wedge \cdots \wedge G_w$, where each G_i is the OR of at most t literals. So $G_i = \tilde{x}_1 \vee \tilde{x}_2 \vee \cdots \vee \tilde{x}_s$, for $s \leq t$, and $\tilde{x}_i \in \{x_i, \bar{x}_i\}$. (Likewise, a t-OR-AND is a formula in disjunctive normal form with at most t terms in each conjunct.)*

2 Random Restrictions

Definition 5 (Restriction). *A restriction ρ is a partial assignment to variables of boolean formula G. Specifically, it is a mapping $\rho : \{1, 2, \cdots, n\} \mapsto \{0, 1, *\}$. The restriction of G by ρ, written $G|_\rho$, is the boolean function obtained by setting x_i to $\rho(i)$ if $\rho(i) \in \{0, 1\}$ and leaving x_i as a variable otherwise.*

A random p-restriction is a restriction ρ where, for each i, one independently assigns $\rho(i)$ such that:

$$\Pr[\rho(i) = *] = p, \quad \text{and}$$
$$\Pr[\rho(i) = 0] = \Pr[\rho(i) = 1] = \frac{1-p}{2}$$

3 Proving Circuit Lower Bounds for Parity: Overview

Here we will present a high level overview of proving lower bounds relating depth and size of circuits computing parity. A formal proof will be presented in subsequent lectures.
Before proceeding, let us review some conventions. Suppose we have a circuit that uses negation gates. By using de Morgan's law, we can push down the negation gates until they reach the input level. Thus, any boolean circuit can be transformed to an equivalent one such that all the negation gates occur at the bottom level (i.e. any negation gate takes it input only from variables). This transformation can be accomplished increasing the size at most twice the original and without any increase in depth. We assume that the input consists of both the variables and their negations so that size of a circuit refers to the number of AND and OR gates. We do not restrict fan-in or fan-out of the gates. So, without loss of generality, we can assume that the circuit is "leveled": a circuit of depth d is made of d levels so that all gates in a level are of same type and edges are only between adjacent levels. Thus, AND and OR gates will alternate across levels. From now on, we consider only circuits with above properties. By convention, output level (made of just one gate) is numbered 1 and the bottom level (where gates take their input from variables) is numbered d.
Now we are ready to state the the lower bound theorem.

Theorem 1. *For sufficiently large n, Parity_n cannot be computed by a depth-d circuit of size $\leq 2^{cn^{1/d}}$, where $c > 0$ is some constant.*

One can prove the theorem with $c = 0.143781$. Here we will prove a weaker version where $c \approx 0.1$. The main ingredient of the proof is the switching lemma, which is as follows.

Lemma 2. *[Switching Lemma] Let G be a t-And-Or formula. Let ρ be a random p-restriction. Then, for all $\Delta \geq 0$,*

$$\Pr[DC(G|_\rho) > \Delta] \leq (5pt)^{\Delta}. \tag{1}$$

Assuming the switching lemma, we sketch a proof of Theorem 1. Set $p = \gamma_0/t$, where $\gamma_0 > 0$ is a suitable constant fixed later. We will apply the lemma with $\Delta = t$. Then, the lemma says that when we apply a random p-restriction to a t-AND-OR formula G, with probability at least $1 - (5\gamma_0)^t$, the resultant formula $G|_\rho$ has DC $\leq t$. As we noted before, a formula with DC $\leq t$, can be expressed as a t-OR-AND formula. Thus, with high probability, the t-AND-OR formula $G|_\rho$ can be "switched" into an equivalent t-OR-AND formula. Using this ability to "switch", we can prove Theorem 1.
We first consider some restricted type of circuits. A circuit C of depth d is said to be of type $C^d(s,t)$ if it satisfies the two conditions: i) the gates at the bottom-most level (i.e. input level) have fan-in $\leq t$; ii) number of gates at levels above the bottom level $\leq s$. Thus, we have a bottom fan-in condition (bfi), which restricts fan-in of the bottom level gates. The gates above the bottom level (or the "internal" gates) are allowed to have any fan-in. At the end, we will relax these conditions and prove the theorem for any d-depth circuit (this part is easy).

Assume, without loss of generality that, the gates at the bottom level are OR gates (if not consider $\neg C$). Each gate at one level above ($(d-1)^{th}$ level) is an AND gate which gets its input from the OR gates at level d. Think of each such AND gates and along with the OR gates as a small circuit. Any such small circuit sc is computing a t-AND-OR formula, because of the bottom fan in condition. Choose a random p-restriction ρ and consider the circuit $C|_\rho$. By the switching lemma, with probability $\geq 1 - (5\gamma_0)^t$, the formula computed by the small circuit sc (in $C|_\rho$) has DC $\leq t$.

As we discussed, this formula is (also) a t-OR-AND formula. Thus, we can *replace* the AND-OR circuit sc by an OR-AND circuit, where the AND gates have fan-in $\leq t$. Suppose there are s_{d-1} AND gates at level $d-1$. Assume that we were lucky and our random restriction ρ is good so that we are able to replace all these s_{d-1} AND-OR circuits by s_{d-1} OR-AND circuits with bottom fan-in $\leq t$. The resultant circuit is made of OR gates in level $d-1$, OR gates is level $d-2$ and AND gates in level d. We can now *merge* the levels $d-1$ and $d-2$ in $C|_\rho$ to get a circuit that has only $d-1$ levels. There are a few things to note. The new circuit may have more gates at its bottom level than the original circuit. But, number of internal gates in the new circuits is $s - s_{d-1} \leq s$. Thus, the number of internal gates didn't increase. Next, bottom fan-in of the new circuit is $\leq t$. The new circuit has only $d-1$ levels. To summarize, the new circuit is of type $C^{d-1}(s,t)$ and is equivalent to $C|_\rho$. Of course, the new circuit is not computing the same function as the original one. But, this is not an issue when it comes to the parity function. We will elaborate this shortly.

Now the idea is to apply the above process of switching $d-2$ times: we pick a random p-restriction ρ_1 and apply it to C, then pick a random p-restriction ρ_2 and apply it to $C|_{\rho_1}$ and so on. Suppose we were lucky in picking all these $d-2$ random restrictions. Then, we will be left with a circuit C' of type $C^2(1,t)$.

Suppose the original circuit C computes parity on n bits. What does the new circuit C' compute? After applying all the $d-2$ random restrictions, some of the n variables would have been assigned 1 or 0 and the rest would remain as variables (i.e assigned *). Let the number of variables remaining be N (which is a random variable). Observe that C' computes the parity on these N variables.

Suppose all the $d-2$ random restrictions were good and suppose $N > t$. Then, C' is a circuit of type $C^2(1,t)$ that computes parity on $N > t$ variables. That is impossible. (C' computes a formula $G_1 \wedge G_2 \wedge \ldots \wedge G_w$, where each G_i is an OR of $\leq t$ literals. In particular, G_1 has some $a \leq t < N$ literals. Set all these literals to 0. Then output of C' is 0. But, since $N > a$, G_1 does not include some variable x. Set all the variables except those in G_1 and x to be 0. Now, set x appropriately, so that the parity is 1. Thus, C' does not compute parity of N variables).

The rest of proof is to show that with non-zero probability two properties are satisfied: i) the $d-2$ random restrictions are good (i.e. they allow us to "switch" t-AND-OR circuits to t-OR-AND circuits in all instances) and ii) $N > t$. We first note that instead of picking $d-2$ random restrictions, we could equivalently pick just one random restriction. Suppose we have a formula F we apply a random p_1-restriction followed by a random p_2-restriction. Observe that, equivalently, we can apply a random $(p_1 \cdot p_2)$-restriction. (Distribution of the boolean functions obtained in the two cases will be the same). In our scenario, instead of applying $d-2$ random p-restriction one by one, we can equivalently apply a single random p^{d-2}-restriction.

We have a circuit C of type $C^d(s,t)$ and a random p^{d-2}-restriction ρ. We need an estimate on the probability that $C|_\rho$ can be transformed (by switching) into a $C^2(1,t)$ circuit. As C has s internal gates, we will need do s many "switches". The probability of failing in any one of these instances is bounded $\leq (5pt)^\Delta = (5\gamma_0)^t$. Thus, probability that we fail to convert $C|_\rho$ into an equivalent $C^2(1,t)$ circuit is $\leq s \cdot (5\gamma_0)^t$. Formally,

$$\Pr[C|_\rho \text{ is not equivalent to a } C^2(1,t) \text{ circuit}] \leq s \cdot (5\gamma_0)^t \tag{2}$$

Now we estimate the random variable N, the number of input variables of $C|_\rho$. $E[N] = n \cdot p^{d-2} =$

$n \cdot (\gamma_0/t)^{d-2}$.
We will fix γ_0 and t to obtain a bound on $E[N]$ and the RHS of (2). Set $\gamma_0 = 1/10$ and $t = \gamma_0 n^{1/(d-1)}$ Then,

$$\Pr[C|_\rho \text{ is not equivalent to a } C^2(1,t) \text{ circuit}] \leq s \cdot (5\gamma_0)^t = s \cdot 2^{-(0.1)n^{\frac{1}{(d-1)}}}.$$

We have $E[N] = n^{1/(d-1)}$. Then, using Chernoff bound, we get

$$\Pr[N < t] = \Pr[N < \gamma_0 \cdot E[N]] < e^{-\frac{(1-\gamma_0)^2}{2} \cdot n^{\frac{1}{d-1}}} < e^{-(0.4)n^{\frac{1}{d-1}}}$$

Let $c < 0.1$ be a constant. For sufficiently large n, if $s \leq 2^{-c \cdot n^{1/(d-1)}}$, then

$$s \cdot 2^{-(0.1)n^{\frac{1}{(d-1)}}} + e^{-(0.4)n^{\frac{1}{d-1}}} < 1.$$

Let us summarize. Let $s < 2^{-c \cdot n^{1/(d-1)}}$ and C be a circuit of type $C^d(s,t)$ that computes parity of n bits. Then, with non-zero probability, $C|_\rho$ can be transoformed into a $C^2(1,t)$ circuit that computes parity on $N > t$ bits. In particular, there exists a random restriction ρ_0 that satisfies both these properties. But, a circuit of type $C^2(1,t)$ cannot compute parity on $> t$ bits. A contradiction. We have proved the following theorem.

Theorem 2. *For $s \leq 2^{-c \cdot n^{1/(d-1)}}$, circuits of type $C^d(s,t)$ cannot compute parity on n bits. Here, c can be any constant < 0.1.*

It is now easy to prove Theorem 1. Observe that a circuit of size s and depth d can be viewed as a circuit of type $C^{d+1}(s,1)$. Then use Theorem 2. Later, we will prove a better version of the switching lemma, do a careful analysis and improve the contant c from 0.1 to 0.143781.