MATH 262A LECTURE 11: LOWER BOUNDS FOR CONSTANT DEPTH CIRCUITS FOR PARITY

SCRIBE: CHRISTIAN WOODS

1. Depth Bounds for Decision Trees

Definition. *A decision tree is a tree that queries variable values, possibly based on previous queries. A decision tree of depth d is a decision tree whose maximum number of queries on any branch is d.*

Theorem. *Let f be an n-ary Boolean function computed by a depth d decision tree T. Then f can be expressed as a d-CNF and as a d-DNF.*

Proof. To form the d-DNF computing f, identify the paths π in T that terminate in the result 1. For each such π, consider the conjunction

$$t_\pi = \bigwedge \{x : \pi \text{ asserts } x \text{ is true}\}.$$

Then

$$f = \bigvee_{\pi:\text{ output is } 1} t_\pi.$$

Since the decision tree for f has depth d, each t_π contains at most d conjuncts, and therefore the above is a d-DNF.

To form the d-CNF for f, consider the decision tree for $\neg f$, obtained by flipping any resulting 0's in T to 1's and any resulting 1's in T to 0's. By the above proof, we can form a d-DNF for $\neg f$. By using DeMorgan's Law, this will result in a d-CNF for f. □

Theorem. *If f is expressible as a k_1-CNF and as a k_2-DNF, then f is expressible as a depth $k_1 k_2$ decision tree.*

Proof. Let C_1 be a k_1-CNF for f, and let C_2 be a k_2-DNF for f. So $C_1 \equiv C_2$.

Claim. *Any conjunct $y_1 \vee y_2 \vee \ldots \vee y_{l_1}$, $l_1 \leq k_1$ in C_1 and any disjunct $z_1 \wedge z_2 \wedge \ldots \wedge z_{l_2}$, $l_2 \leq k_2$ in C_2 share some literal in common.*

Date: November 1, 2013.

Proof. Suppose not. Then we may set each literal in $z_1 \wedge \ldots \wedge z_{l_2}$ to 1 and set each literal in $y_1 \vee \ldots \vee y_{l_1}$ to 0 with the same variable assignment. But this one assignment would make C_2 true and C_1 false, despite the fact that they represent the same Boolean function. This is a contradiction. □

We now proceed with the proof of the theorem by induction on the value of $k_1 + k_2$.

For the only nontrivial base case, suppose $k_1 = k_2 = 1$. Then $C_1 = x_{i_1} \wedge x_{i_2} \wedge \ldots x_{i_k}$ and $C_2 = x_{j_1} \vee x_{j_2} \vee \ldots \vee x_{j_l}$. Since $C_1 \equiv C_2$, it must be the case that C_1 and C_2 are identical formulas with $k = l = 1$ and $x_{i_1} = x_{j_1}$. So the decision tree is simply the querying of this literal, and hence has depth $1 = k_1 k_2$.

Now suppose $k_1 > 1$. Take some clause $y_1 \vee \ldots \vee y_l$ in C_1. Form a decision tree by querying $y_1, y_2, \ldots, y_l$ in order. There is only one branch where $y_1, y_2, \ldots, y_l$ are set to 0 (and hence C_1 is 0, and no other querying is necessary), and the rest have at least one of these y_i set to 1.

On the branches where C_1 is yet unknown, each term (conjunction) in C_2 as had at least one literal set true or false (by the claim). Thus C_2 has been simplified to a $(k_2 - 1)$-DNF. So, for these branches, the result is expressible as a k_1-CNF and as a $(k_2 - 1)$-DNF. By the inductive hypothesis, we may add to each such branch a new decision tree computing this simplified function. This decision tree has depth

$$k_1(k_2 - 1) = k_1 k_2 - k_1.$$

Therefore, the resulting decision tree for f has $(k_1 k_2 - k_1) + k_1$ queries on its longest branch, and hence has depth $k_1 k_2$. □

<u>Exercise:</u> Is this bound $(k_1 k_2)$ tight?

2. Decision Trees for Parity

Let us consider now decision trees for the function $Parity_n$.

Of course we may query each variable in sequence to get a decision tree of depth n.

What about a decision tree T of depth $n - 1$? T will only agree with $Parity_n$ on exactly half of the inputs in $\{0, 1\}^n$.

Similarly, we can write $Parity_n$ as a k-DNF (CNF) for $k = n$, but not for any $k < n$. Notice that this gives us a lower bound of 2^{n-1} on depth 2 circuits for $Parity_n$. We improve this result with the next theorem:

Theorem. *For $d \geq 2$, any depth $(d + 1)$ circuit for $Parity_n$ has size*

$$2^{\Omega\left(n^{1/d}\right)}.$$

Definition. *Let f be an n-ary Boolean function. Define $R(f)$ to be the minimum number of variables which can be set to 0 or 1 to force the value of f to be constant.*

For example, $R(Parity_n) = n$.

Lemma. *Let $f(x_1, \dots, x_n)$ be an n-ary Boolean function computed by a depth $(d+1)$ circuit over $\bigwedge, \bigvee, x_i, \overline{x}_i$ of size S. Then*

$$R(f) \leq n - \frac{n}{c_d(\log S)^{d-1}} + 2\log S,$$

where $c_d = 2^{6d-1}$.

Proof of Theorem from Lemma. Let $Parity_n$ be computed by a depth $(d+1)$ circuit of size S. Then by the lemma, since $R(Parity_n) = n$,

$$\begin{aligned} n \leq n - \frac{n}{c_d(\log S)^{d-1}} + 2\log S &\Rightarrow \\ \frac{n}{c_d(\log S)^{d-1}} \leq 2\log S &\Rightarrow \\ 2(\log S)^d \geq \frac{n}{c_d} &\Rightarrow \\ (\log S)^d \geq \frac{n}{2c_d} &\Rightarrow \\ (\log S)^d \geq \frac{n}{2^{6d}} &\Rightarrow \\ \log S \geq \frac{n^{1/d}}{2^6} &\Rightarrow \\ S \geq 2^{\frac{n^{1/d}}{2^6}} &\Rightarrow \\ S = 2^{\Omega\left(n^{1/d}\right)}. \end{aligned}$$

□

Proof of Lemma. We use the Switching Lemma.

Suppose we have a circuit C of size S and depth $(d+1)$ (with a level of $\bigwedge$'s at the bottom).

Add extra 1-input $\bigvee$'s to the bottom so that the circuit now has depth $(d+2)$ and bottom fan-in 1. Now we may apply the switching lemma with $t = 1$, $s = 2\log S$, and $p = \frac{1}{32}$. Each $\bigwedge$ of $\bigvee$'s of size 1 at the bottom of C becomes a $\bigvee$ of $\bigwedge$'s of size at most s with probability at least

$$\begin{aligned} 1 - (16pt)^s &= 1 - \left(\frac{16}{32}\right)^{2\log S} \\ &= 1 - S^{-2}. \end{aligned}$$

Since there are at most S many $\bigwedge$ of $\bigvee$'s at the bottom of C, the probability that at least one fails to switch is at most

$$S\left(\frac{1}{S^2}\right) = \frac{1}{S}.$$

Now we have a depth $(d+1)$ circuit, still with S nodes (not counting the very bottom level) with bottom fan-in at most $2\log S$.

Now continue to switch from depth d' to depth $d'-1$ (for $d' = d+1, d, \ldots, 3$) using $t = 2\log S$, $s = 2\log S$, and $p = \frac{1}{64(\log S)}$. We end up with a depth two sub circuit with probability at least

$$1 - (16pt)^s = 1 - \frac{1}{S^2}$$

and the probability that something at the bottom fails to switch is still at most $\frac{1}{S}$.

At the end of the process we will have a depth two circuit with bottom fan-in $2\log S$, having set all but

$$n\left(\frac{1}{32}\right)\left(\frac{1}{64(\log S)}\right)^{d-1} = \frac{n}{2^{6d-1}(\log S)^{d-1}}$$

variables.

Now by setting only $2\log S$ more variables (the ones at the bottom level), we may force the function f to be constant. Therefore,

$$R(f) \leq n - \frac{n}{c_d(\log S)^{d-1}} + 2\log S.$$

□

If we continue the construction by switching once more, we express the circuit as a $\bigwedge$ of $\bigvee$'s of bottom fan-in $2\log S$ and a $\bigvee$ of $\bigwedge$'s of bottom fan-in $2\log S$. By the earlier theorem, this means that f has a decision tree of depth

$$(2\log S)^2 < S.$$

Thus this decision tree is wrong for $Parity_n$ on exactly half of its inputs.

We've set at most $n - n\left(\frac{1}{2^{6d-1}(\log S)^{d-1}}\right)$ variables at random and get with probability at least $1 - \frac{(d+1)}{S}$ a decision tree that is wrong on $Parity_n$ on exactly half of its inputs.

By an averaging argument, there is some set of L variables such that, when assigned values 0 or 1 at random, with probability at least $1 - \frac{(d+1)}{S}$ the result (as computed by C) is wrong half of the time.

So the original circuit disagreed with $Parity_n$ at least on

$$2^L\left(1 - \frac{(d+1)}{S}\right)2^{n-L-1} = 2^n\left(\frac{1}{2} - \frac{(d+1)}{2S}\right)$$

settings. In other words, C is wrong on about $\frac{1}{2} - \frac{(d+1)}{2S}$ of the inputs. We say that the inapproximability of $Parity_n$ is

$$\frac{1}{2} - \Omega\left(\frac{1}{2^{n^{1/(d-1)}}}\right).$$

In 2012 Beame-Impagliazzo-Srinivasan, and later Hastad, showed better bounds. In particular, Hastad showed an inapproximability of

$$\frac{1}{2} - 2^{-\Omega\left(\frac{n}{(\log S)^{d-1}}\right)}.$$