Lecture 3: AC^0, the switching lemma

Topics in Complexity Theory and Pseudorandomness (Spring 2013)
Rutgers University
Swastik Kopparty
Scribes: Meng Li, Abdul Basit

1 Pseudorandom sets

We start by proving the existence of small sets that look pseudorandom to all small circuits. Getting your hands on such a set is enough for derandomizing all your randomized algorithms.

Theorem 1. *For any c and n, there is an $m = O(n^{2c})$ and $x_1, \ldots, x_m \subseteq \{0,1\}^n$ such that $\forall$ circuit C with $|C| \leq n^c$*

$$| Pr_{i \in [m]}[C(x_i) = 1] - Pr_{x \in \{0,1\}^n}[C(x) = 1] | \leq 0.1$$

Proof. Pick $x_1, x_2, ..., x_m$ from $\{0,1\}^n$ independently and uniformly at random. For a fixed C, consider the probability

$$\Pr_{x_1,\ldots,x_m}\left[\left|\Pr_{i\in[m]}[C(x_i)=1] - \Pr_{x\in\{0,1\}^n}[C(x)=1]\right| > 0.1\right]$$

Let

$$Y_i = \begin{cases} 1 & C(x_i) = 1 \\ 0 & C(x_i) = 0 \end{cases}$$

Notice that

$$\begin{aligned} \mathrm{E}[Y_i] &= \Pr_{x\in\{0,1\}^n}[C(x)=1] \\ \mathbb{E}\left[\frac{\sum_{i=1}^m Y_i}{m}\right] &= \Pr_{i\in[m]}[C(x_i)=1] \end{aligned}$$

So by the Chernoff bound:

$$\Pr_{x_1,\ldots,x_m}\left[\left|\Pr_{i\in[m]}[C(x_i)=1] - \Pr_{x\in\{0,1\}^n}[C(x)=1]\right| > 0.1\right] \leq e^{-(0.1)^2 m/2}$$

Now we know that for every fixed circuit C, the random collection $x_1, \ldots, x_m$ fools C with high probability. Taking a union bound over all C s.t. $|C| \leq n^c$ (of which there are $(n^c)^{O(n^c)}$):

$$\Pr_{x_1,\ldots,x_m}\left[\exists C, |C| \leq n^c \text{ s.t. } \left|\Pr_{i\in[m]}[C(x_i)=1] - \Pr_{x\in\{0,1\}^n}[C(x)=1]\right| > 0.1\right] \leq e^{-O(n^{2c})} \cdot e^{O(n^c \log n)} < 1$$

Thus with positive probability (and in fact, with high probability), a random choice of $x_1, \ldots x_m$ has the desired property. In particular, such a choice exists. □

2 AC^0 circuits

AC^0 circuits are circuits with $O(1)$ depth and unbounded fan-in AND, OR and NOT gates. It will be the first nontrivial circuit class for which we can prove exponential lower bounds.

First observe: every polynomial size AC^0 circuit has an equivalent circuit in NC^1. We will show that PARITY and MAJORITY both do not have polynomial size AC^0 circuits. Thus AC^0 is a strict subset of NC^1.

Let us warm up by understanding AC^0 circuits when the depth is really small. Depth 0 circuits are simply literals. Depth 1 circuits are simply ANDs or ORs of literals. Circuits of depth 0 and 1 thus cannot even compute some functions, even if we do not care about their size.

Depth 2 circuits are either DNFs (ORs of ANDs of literals) or CNFs(ANDs of ORs of literals). Every function can be computed by a DNF and also by a CNF. The AND gates of a DNF are called *terms*. The OR gates of a CNF are called *clauses*.

A t-DNF is a DNF where all the gates at the bottom level have fan-in at most t. Note that not every function can be computed by a t-DNF if $t < n$.

Theorem 2. *Depth 2 AC^0 circuits for PARITY have size at least $\Omega(2^n)$.*

Proof. Suppose we have a DNF computing PARITY (the case of CNF is handled similarly). Suppose some term T does not include all variables. Consider an assignment which sets that term T to 1, and thus sets the value of the full DNF to 1. Now if we flip the value of any of the variables not in T, the term is still set to 1, and thus so is the DNF. But PARITY changes its value every time we flip a variable. Hence, all terms include all n variables.

Now a term that includes all n variables can equal 1 on exactly 1 input in $\{0,1\}^n$. Since the DNF has to compute PARITY, which equals 1 on 2^{n-1} inputs, there must be at least 2^{n-1} terms in the DNF. □

Already showing that polynomial size depth 3 circuits cannot compute PARITY is quite nontrivial.

We will show that any AC^0 circuit of constant depth and subexponential size cannot compute PARITY.

Theorem 3. *If C is an AC^0 circuit of size s, depth d computing PARITY, then $s \geq 2^{\Omega(n^{\frac{1}{d-1}})}$*

In order to prove the theorem, we need to use a powerful theorem describing the behavior of AC^0 circuits under restrictions. As in the case of formula lower bounds, we will say that there is a restriction of any AC^0 circuit which simplifies it drastically. On the other hand, any restriction of the PARITY function is a PARITY function, and will mean that it cannot be computed by a drastically simplified AC^0 circuit (unless the originall AC^0 circuit was very big to begin with).

How can we hope to prove the existence of such a restriction? One idea is to induct on the circuit. Consider all the subcircuits feeding in to the top gate. By induction, we can get a simplifying restriction for each of the subcircuits. But how do these restrictions relate to each other? In fact,

these restrictions may be mutually inconsistent. This dooms any such naive strategy to find such a restriction.

The key idea is to aim higher. We will try to show that *most* restrictions simplify a circuit drastically. This kind of statement lends itself to induction very nicely; if for each subcircuit of a circuit, most restrictions simplify that subcircuit drastically, then by a union bound, most restriction simplify all the subcircuits of a circuit simultaneously. The crucial aspect is to analyze how these simplified subcircuits interact with the top gate of the circuit. This is precisely the content of the **switching lemma**.

The switching lemma is concerned with using random restrictions to simplify a CNF/DNF. To generate a random restriction, we pick a random set of variables, and set them to 0/1 uniformly and independently, and leave the remaining variables unset. We denote the formula f after restriction ρ by $f|_\rho$.

Lemma 4. *Given a t-CNF f, apply a random restriction ρ on it, leaving pn variables alive ($p < 1/2$).*

$$Pr[f|_\rho \textit{can not be represented as a s-DNF}] \leq (16pt)^s$$

Proof. Let $l = pn$ be the number of alive variables, R^k be the set of restrictions leaving k variables alive. We have:

$$|R^k| = \binom{n}{k} 2^{n-k}$$

Let B be the set of "bad" restrictions in R^ℓ for which the restricted function cannot be represented as an s-DNF. Obviously, $B \subseteq R^l$. The probability that f can't be rewritten as a s-DNF is $\frac{|B|}{|R^l|}$. To find a upper bound on $|B|$, we try to find an "encoding" of B. Specifically, we will give an injection from $B \to R^{l-s} \times W$, where W is a set of size at most $(4t)^s$. Therefore we will get:

$$\frac{|B|}{|R^l|} \leq \frac{\binom{n}{l-s} 2^{n-l+s} (4t)^s}{\binom{n}{l} 2^{n-l}} \leq (\frac{l}{n-l})^s (8t)^s \leq (\frac{p}{1-p})^s (8t)^s \leq (16tp)^s$$

What kind of property is "not representable as an s-DNF"? Define a *minterm* of a function to be a minimal set of variables which has the property that there is a setting of those variables which forces the function to equal 1. It is a simple fact that if every minterm of a function is of size $\leq s$, then the function can be represented as an s-DNF. So if a function is not representable as an s-DNF, then it must have a minterm of size $> s$.

We fix an ordering of the clauses of f, and within each clause fix an ordering of the variables. Take a bad restriction ρ. Because ρ is bad, $f|_\rho$ is not identically 0 or 1; thus no clause of f is fixed to 0 under ρ (some clauses are fixed to 1, and the rest of the clauses are left unfixed).

We now use the minterm properties of functions not representables as s-DNFs. Since ρ is bad, there is a restriction π of $> s$ of the remaining variables that fixes $f|_\rho$ to 1, but any setting of a subset of $\leq s$ of the variables of π does not fix the function $f|_\rho$ to 1.

Since $\rho\pi$ fixes f to 1, it must fix each clause of f to 1. Let C_1 be the first clause which is not fixed to 1 by ρ (but is fixed to 1 by $\rho\pi$). Suppose C_1 has d_1 variables from π. Let π_1 be the part of π restricting those variables. Let $\overline{\pi}_1$ be the unique setting to the variables of π_1 which prevents C_1 from being fixed to 1 (this exists since C_1 is OR of literals).

Now suppose we know only the restriction $\rho\overline{\pi}_1$, but not ρ. The key observation is that this is enough to identify the clause C_1! Indeed, we can consider the clauses of f in order, and identify the first one which is not fixed to 1 by $\rho\overline{\pi}_1$. By giving a little bit more information along with $\rho\overline{\pi}_1$, we can also identify $\overline{\pi}_1$ and thus ρ. Thus we found a way of describing a bad ρ by giving an element of $R^{\ell-d_1}$, along with a little bit more information. This is the main trick of the proof.

We now fully describe the encoding. Find the first clause C_2 which is not fixed to 1 by $\rho\pi_1$. Let π_2 be the restriction to C_2 of $\pi \setminus \pi_1$. Let $\overline{\pi}_2$ be the unique setting to the variables of π_2 which prevents C_2 from being fixed to 1. In general, let C_i be the first clause which is not fixed to 1 by $\rho\pi_1\pi_2 \dots \pi_{i-1}$. Let π_i be the restriction to C_i of $\pi \setminus (\pi_1\pi_2 \dots \pi_{i-1})$. Let $\overline{\pi}_i$ be the unique setting to the variables of π_i which prevents C_2 from being fixed to 1. Continuing we get $\pi_1, \pi_2, \dots$ and $\overline{\pi}_1, \overline{\pi}_2, \dots$. We stop this process after the total number of variables restricted by $\pi_1 \dots \pi_k$ and $\overline{\pi}_1 \dots \overline{\pi}_k$ exactly equals s (at the last step, we may restrict a subset of the variables of π in C_i).

Let $\overline{\pi}_*$ be the restriction $\overline{\pi}_1\overline{\pi}_2 \dots \overline{\pi}_k$ truncated to exactly s variables. Define $\rho_* = \rho\overline{\pi}_*$. For each $i \in [k]$, define $a_i \in \{0,1\}^t$ as follows: a_i indicates which of the t variables of clause C_i are set by π_i. Also define a string $b \in \{0,1\}^s$ which contains the values of π for each of the s variables fixed by $\pi_1, \pi_2, \dots, \pi_k$. We will show that $(\rho_*, (a_i)_{i=1}^k, b)$ completely determines ρ. Furthermore, we will show that $((a_i)_{i=1}^k, b)$ lies in a set of size at most $(4t)^s$.

As indicated earlier, using $\rho\overline{\pi}_*$ we can identify C_1: we find the first clause not fixed to 1 by $\rho\overline{\pi}_*$. Then using a_1 and b we can find π_1 and $\overline{\pi}_1$. Then consider the restriction $\rho_1 = \rho\overline{\pi}_* \setminus \overline{\pi}_1 \cup \pi_1$. Restriction ρ_1 lets us identify C_2: it is the first clause which is not fixed to 1 by ρ_1. Using a_2, b we can identify π_2 and $\overline{\pi}_2$. Then consider $\rho_2 = \rho_1 \setminus \overline{\pi}_2 \cup \pi_2$, and proceed. In the end we know $\rho_k = \rho\pi_1\pi_2 \dots \pi_k$, as well as $\pi_1, \pi_2, \dots, \pi_k$, and thus we know ρ. Thus we have an encoding of ρ.

Finally, observe that $(a_i)_{i=1}^k$ comes from the set of all sequences of length at most k, such that each element of the sequence being an element of $\{0,1\}^t$ with at least one 1, and the total number of 1's in all the a_i equals s. It is a simple exercise to show that the number of such sequences is at most $(2t)^s$. □

Theorem 5. *If C is a circuit of size M and depth d that computes the parity of n inputs, then*

$$M \geq 2^{\Omega(n^{1/(d-1)})}$$

Proof Idea: The proof will use the Switching Lemma extensively. Given an AC^0 circuit, a random restriction is very likely to simplify each DNF (assuming it has small width) at the bottom two layers . Specifcally, it's quite likely that the restriction of each small-width DNF will be computable by small-width CNF. So we switch all the small-width DNFs at the bottom two layers with small-width CNFs. This lets us merge two layers of AND gates, and hence shrink the depth by 1. We then repeat, overall making $d-2$ random restrictions. The final restricted function is computable by a small width DNF/CNF. We have that a good fraction of the variables are still unset and that any restriction of Parity is either Parity (or its negation). But Parity on m variables (or its negation) cannot be computed by a DNF/CNF of width $< m$.

Proof. Suppose C is a depth d AC^0 circuit of size M which computes parity. Assume without loss of generality that the bottom layer of C is $\wedge$ gates (a symmetric argument works if the bttom layer is $\vee$ gates). Then the second layer consist of $\vee$ gates computing DNF formulas. We first apply a random restriction α_0 which sets each variable with probability $1/2$; this serves to make every gate at the bottom layer have fan-in bounded by t (with t TBA).

A $\wedge$ gate survives iff none of its variables has been set to 0. For a fixed gate, the probability that a $\wedge$ gate of width $> t$ survives is $\leq (3/4)^t$. Then by the union bound, we have that the probability that some gate of height 1 and width $> t$ survives is $\leq M(3/4)^t$. Also, by the Chernoff bound the probability that fewer than $n/4$ variables are unset is $\exp(-n)$.

After the restriction α_0, we have that the gates at the second layer compute t-DNF's. We now apply a restriction α_1 with $p = 1/100t$. From the Switching Lemma, we have that the probability that a particular restricted DNF can not be represented by a t-CNF is $(10pt)^t = (1/10)^t$. Then by the union bound, the probability that some gate fails to become a t-CNF is $\leq M(1/10)^t$. If we now "plug in" these CNFs to the circuit, we can collapse the bottom two laters and get a new circuit of depth $d-1$, with the gates at bottom level having fan-in at most t.

We can fix such a restriction, and repeat. We apply restrictions $\alpha_2, \ldots, \alpha_{d-2}$, each with $p = 1/100t$, collapsing the circuit to depth 2. After the last restriction, the number m of variables remaining is $\geq \frac{n}{4(100t)^{d-2}}$ with probability at least $\exp(-n)$. Also, with probability at most $O(dM \cdot (1/10)^t)$, the resulting depth 2 circuit is a t-DNF/t-CNF.

Now if $t < m$, this is a contradiction, since a t-DNF/t-CNF cannot compute parity or its negation on $> t$ bits. Thus if

$$O(dM \cdot (1/10)^t) + \exp(-n) < 1,$$
$$t < \frac{n}{4(100t)^{d-2}}$$

then we have a contradiction.

Taking $t = O(n^{\frac{1}{d-1}})$ the second condition holds, and so to invalidate the first condition we must have $M \geq 2^{\Omega(n^{\frac{1}{d-1}})}$, as desired.

□

3 Lower Bounds for Majority

Theorem 6. *If C is a circuit of size M and depth d that computes the majority of n inputs, then*

$$M \geq 2^{\Omega(n^{1/(d-1)})}$$

Theorem 6 can be proved using the Switching Lemma. It is a good exercise to try and do so. We prove the following weaker version of the theorem:

Theorem 7. *If C is a circuit of size M and depth d that computes the majority of n inputs, then*

$$M \geq 2^{\Omega(n^{1/d})}$$

The proof of the theorem follows from the following Lemma:

Lemma 8. *$MAJ \in \mathsf{AC}^0 \Rightarrow PARITY \in \mathsf{AC}^0$.*

Proof. Say that Majority $\in \mathsf{AC}^0$. Then given $x \in \{0,1\}$, we can test if $|x| \leq k$ or $|x| \geq k$, by feeding fixed 0/1 bits along with x into the Majority circuit. It follows that we can test $|x| = k$. Let C_k be a circuit that tests if $|x| = k$. Then we can realize Parity by taking $\bigvee_{k \text{ odd}} C_k$. □

4 Randomized AC^0

Theorem 9. *For every $C(x,r) \in \mathsf{AC}^0$, with $|C| \leq n^c$ and function $f : \{0,1\}^n \to \{0,1\}$ such that*

$$Pr_{r \in \{0,1\}^n}[C(x,r) \neq f(x)] \leq \epsilon \quad \forall x \in \{0,1\}^n$$

then $\exists\ C'(x) \in \mathsf{AC}^0$ s.t. $C'(x) = f(x)$, $\forall x \in \{0,1\}^n$ with $|C'| \leq n^{2c}$.

In the last lecture, we showed that any randomized circuit C computing f could be converted into a deterministic one. The proof involved amplifying the success probability by running the circuit several times with independent random seeds and then taking the majority. Once we have a circuit C' computing f with error probability at most 2^{-n}, then there exists a random string r that gives the correct answer for every $x \in \{0,1\}^n$. Fixing r into C' gives us a deterministic circuit.

Since we proved that Majority $\notin \mathsf{AC}^0$, the same argument will not work for converting randomized AC^0 circuits to deterministic ones. A beautiful observation of Ajtai and Ben-Or is that we don't really require the full power of Majority for that argument to work. Instead one can work with a weaker notion, Approximate Majority. It turns out that AC^0 circuits can compute Approximate Majority. With this new tool, we will be able to prove Theorem 9.

Definition 10 (Approximate Majority)**.** *An Approximate Majority is a boolean function $f : \{0,1\}^n \to \{0,1\}$ such that:*

- *$f(x) = 0$ for every x of Hamming weight at most $n/4$.*
- *$f(x) = 1$ for every x of Hamming weight at least $3n/4$.*

Theorem 11 (Ajtai, Ben-Or)**.** *There exists an Approximate Majority computable with polynomial size AC^0 circuits.*

Thus every function computable by a polynomial size randomized AC^0 can also be computed by a polynomial size deterministic AC^0 circuit.

We will prove this by coming up with a distribution of circuits C such that

- if $|x| \leq n/4$ then $Pr_C[C(x) \neq 0] << 2^{-n^2}$.
- if $|x| \leq 3n/4$ then $Pr_C[C(x) \neq 1] << 2^{-n^2}$.

If we can come up with such a distrubution, then there exists a circuit in AC^0 that computes Approximate Majority.