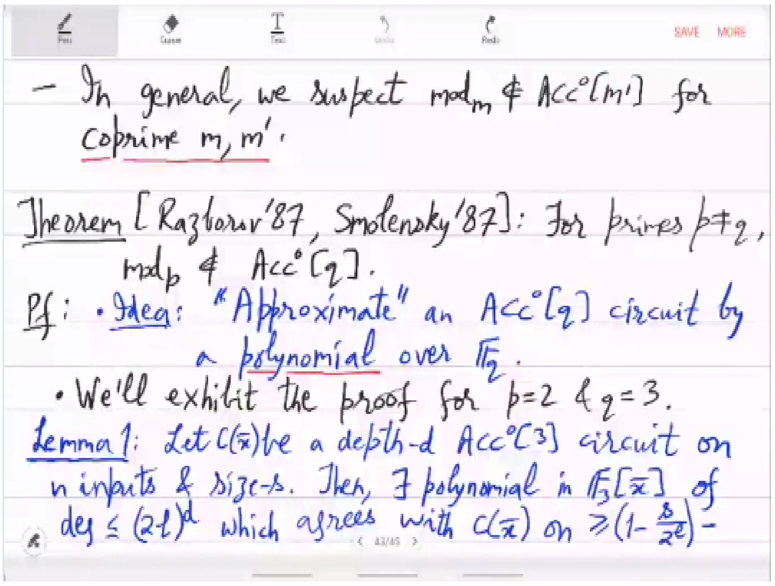
Randomized Methods in Complexity
Prof. Nitin Saxena
Department of Computer Science & Engineering
Indian Institute of Technology - Kanpur

Lecture - 06
ACC0 Lower Bounds Continued…

We are in the middle of proving this theorem due to Razborov and Smolensky that if p q are different primes C 2 and 3.

(Refer Slide Time: 00:28)



– In general, we suspect $\text{mod}_m \notin Acc^0[m']$ for coprime m, m'.

Theorem [Razborov'87, Smolensky'87]: For primes $p \neq q$, $\text{mod}_p \notin Acc^0[q]$.

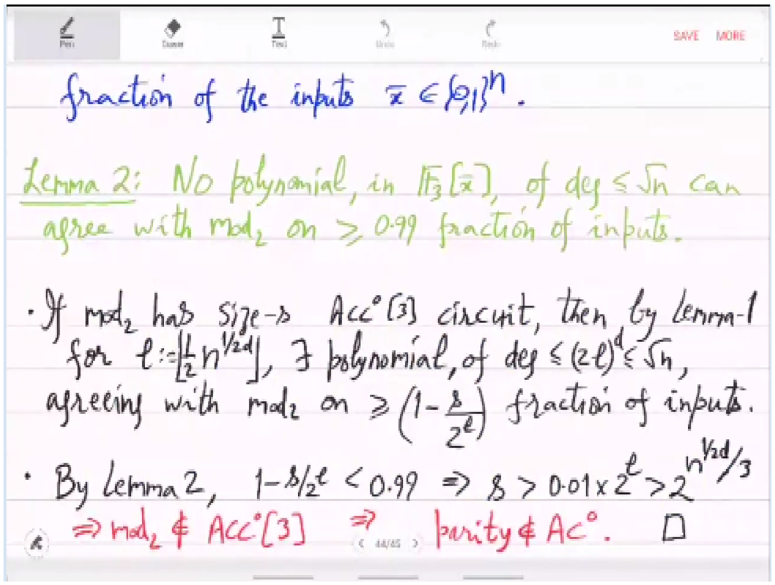
Pf: • Idea: "Approximate" an $Acc^0[q]$ circuit by a polynomial over $\mathbb{F}_q$.

• We'll exhibit the proof for $p=2$ & $q=3$.

Lemma 1: Let $C(\bar{x})$ be a depth-d $Acc^0[3]$ circuit on n inputs & size-s. Then, $\exists$ polynomial in $\mathbb{F}_3[\bar{x}]$ of $\deg \leq (2l)^d$ which agrees with $C(\bar{x})$ on $\geq (1-\frac{s}{2^l})$ -

Then mod 2 cannot be computed using mod 3 in a constant depth Boolean circuit. So, we break up the proof into 2 lemmas. First lemma says that Acc^0 [3] circuit can be captured or reduced to a polynomial of low degree, where degree is (2l)^d, d is the depth of the circuit, which will approximate the circuit. So, it is an approximator with a good number of or fraction of inputs.

(Refer Slide Time: 01:06)


fraction of the inputs $\bar{x} \in \{0,1\}^n$.

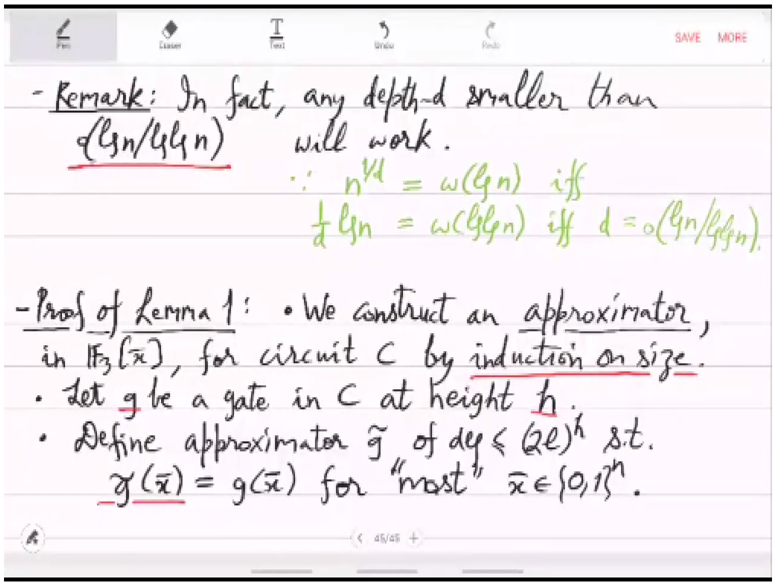
<u>Lemma 2:</u> No polynomial, in $\mathbb{F}_3[\bar{x}]$, of deg $\le \sqrt{n}$ can agree with mod$_2$ on ≥ 0.99 fraction of inputs.

- If mod$_2$ has size-s $Acc^0[3]$ circuit, then by lemma-1 for $\ell := \lfloor \frac{1}{2} n^{1/2d} \rfloor$, $\exists$ polynomial, of deg $\le (2\ell)^d \le \sqrt{n}$, agreeing with mod$_2$ on $\geq \left(1-\frac{s}{2^\ell}\right)$ fraction of inputs.
- By lemma 2, $1-s/2^\ell < 0.99 \Rightarrow s > 0.01 \times 2^\ell > 2^{n^{1/2d}/3}$

$\Rightarrow$ mod$_2 \notin Acc^0[3] \overset{?}{\Rightarrow}$ parity $\notin Ac^0$. $\square$

Once we have shown this, then we will also show that mod 2 does not have this property. Mod 2 does not have good approximators and that will give you the result that parity is not in Ac^0. That is a major result with a very interesting proof technique.

(Refer Slide Time: 01:29)


- <u>Remark:</u> In fact, any depth-d smaller than $(\lg n/\lg\lg n)$ will work.

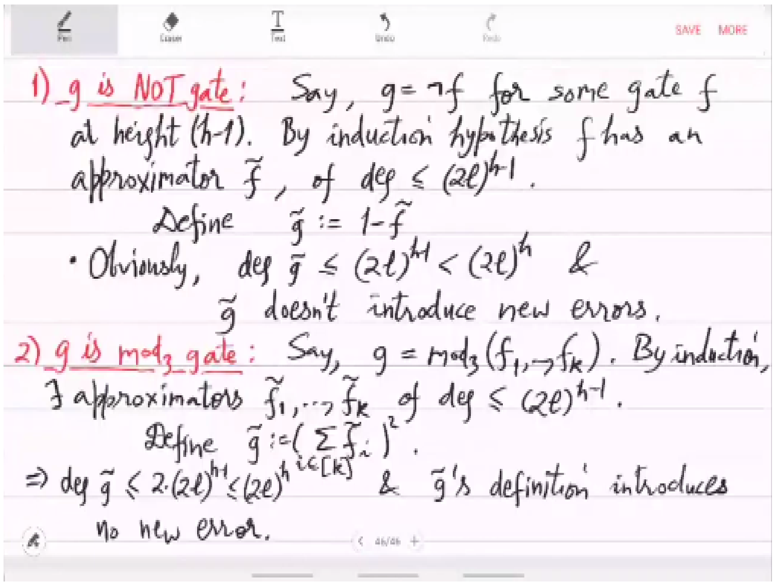
$\because$ $n^{1/d} = \omega(\lg n)$ iff

$\frac{1}{d}\lg n = \omega(\lg\lg n)$ iff $d = o(\lg n/\lg\lg n)$.

- <u>Proof of Lemma 1:</u> • We construct an <u>approximator</u>, in $\mathbb{F}_3[\bar{x}]$, for circuit C by <u>induction on size</u>.
- Let g be a gate in C at height h.
- Define approximator $\tilde{g}$ of deg $\le (2\ell)^h$ s.t. $\tilde{g}(\bar{x}) = g(\bar{x})$ for "most" $\bar{x} \in \{0,1\}^n$.

So, proof of lemma 1, so, approximator will be constructed by induction on the size. So, suppose, you are looking at a gate g at height h and for its inputs you already have approximators. So, for this gate g the approximator that we will build or construct is called g tilde and it we want its degree to be (2l)^h and it should correctly compute g on most of the x's. So, we will do this gate by gate.

(Refer Slide Time: 02:07)



1) g is NOT gate: Say, $g = \neg f$ for some gate f at height $(h-1)$. By induction hypothesis f has an approximator $\tilde{f}$, of deg $\leq (2\ell)^{h-1}$.

Define $\tilde{g} := 1 - \tilde{f}$

• Obviously, deg $\tilde{g} \leq (2\ell)^{h-1} < (2\ell)^{h}$ & $\tilde{g}$ doesn't introduce new errors.

2) g is mod_3 gate: Say, $g = mod_3(f_1, \ldots, f_k)$. By induction, $\exists$ approximators $\tilde{f}_1, \ldots, \tilde{f}_k$ of deg $\leq (2\ell)^{h-1}$.

Define $\tilde{g} := \left(\sum_{i \in [k]} \tilde{f}_i\right)^2$.

$\Rightarrow$ deg $\tilde{g} \leq 2 \cdot (2\ell)^{h-1} \leq (2\ell)^{h}$ & $\tilde{g}$'s definition introduces no new error.

So, the first case is, first is g is a NOT gate. So, say$g = \neg f$ for some gate f at height(h -1), So, by induction hypothesis, f has an approximator, polynomial f tilde, whose degree is$(2l)^{h-1}$. So, using this f tilde, can you get g tilde? It is very easy. You just want to make 0, 1; 1, 0. So, you can define $\overline{g} \ := 1 - \overline{f}$. So, when f tilde is 0, this is 1; when it is 1 this is 0.

So, this is the NOT approximator. In fact, this does not make any mistakes. So, obviously the degree also has not changed and g tilde does not introduce any errors. So, when it is a NOT gate, then induction hypothesis carries forward successfully with this induction step. What is the base case of this induction? Base case, you can think of as the leaves which are the variables, obviously, the polynomial will be the variable.

So, base case is fine and when g is a NOT gate, you have this induction step. What happens when g is a mod 3 gate? So, remember there are 4 types of gates, NOT, mod 3, AND, OR. So, let us second induction step is for mod_3 gate. So, say g=mod_3 (f_1 ,....f_k) These are k gates, input gates. So, by induction, there exists an approximator; approximators($\overline{f}_1$,...... $\overline{f}_k$) of deg$\leq$ $(2l)^{h-1}$

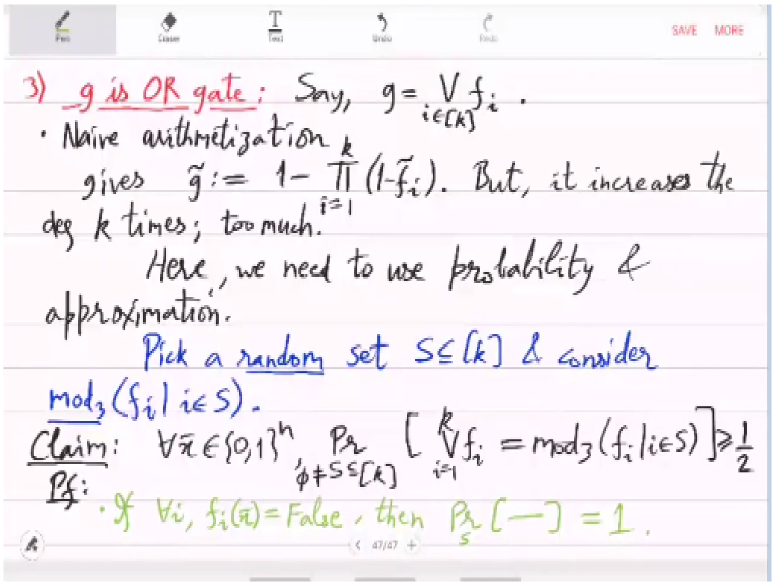
which respectively they approximate f_i , if i tilde approximates f_i well. So, using this, what will be the g tilde? So, g tilde has to be just the sum. So, g tilde is just you can take it to be the sum. So, remember that you wanted a polynomial mod_3 over the field f_3. So, there are 3 field constants

0, 1 and 2. So, the problem is that this sum can also have value 2. So, it is not really a Boolean expression, Boolean value.

So, how do you solve that problem? You just square it and when you square it, then 0, 1, 2 all these values they are mapped to 0 or 1. So, now g tilde is Boolean. It will give you only 0 or 1 values on 0 and 1 input. So, that is the correct choice, correct definition of g tilde. What is the degree? So, degree is whatever it was before doubled. So, it is definitely smaller than $(2l)^h$ and g tilde's definition again introduces no new error because if g was 0, so, mod_3 value of $f_1.,......f_k$

If it was 0, then g tilde will evaluate to 0. If it was 1, then it will evaluate to 1 and if it was 2, then g tilde will evaluate to 1. So, 0 is mapped to 0; 1 and 2 is mapped to 1. So, that is the correct polynomial. So, that is the second kind of induction step. Let us now do OR gate.

(Refer Slide Time: 08:38)



3) g is OR gate: Say, $g = \vee_{i\in[k]} f_i$.

• Naive arithmetization gives $\tilde{g} := 1 - \prod_{i=1}^{k}(1-\tilde{f}_i)$. But, it increases the deg k times; too much.

Here, we need to use probability & approximation.

Pick a random set $S \subseteq [k]$ & consider $mod_3(f_i \mid i \in S)$.

Claim: $\forall \bar{x} \in \{0,1\}^n$, $\Pr_{\phi \neq S \subseteq [k]} \left[\vee_{i=1}^{k} f_i = mod_3(f_i \mid i \in S) \right] \geq \frac{1}{2}$

Pf: • If $\forall i$, $f_i(\bar{x}) = False$, then $\Pr_S[-] = 1$.

So, say $g = \vee f_i$. So, value of g is 0 if and only if all the f_i's are zero, otherwise it is 1 and you have by induction hypothesis polynomials if i tildes. How do you compute the OR polynomial, approximator? So, you have seen before when we did arithmetization of formulas, Boolean formulas. So, if you use that kind of arithmetization, what will you get? So, naive arithmetization gives $\overline{g} := 1 - \prod_{i=1}^{k}(1 - \overline{f_i})$.

So, this has the property that when all the f_i tildes are 0, then this evaluates to 0; otherwise it evaluates to 1. So, this is not introducing any error. It is correct approximator but the problem is, it is degree, has multiplied k times. We do not want degree to grow, so, fast. So, we have to somehow reduce this k but we do not control k; k will depend on the size of the Boolean circuit.

There may be many, many inputs to r. So, we have to do something else. It increases the degree k times which is too much. So, here we need to use probability and approximation. So, till now, in cases 1 and 2, we did not need probabilistic method and we did not need any approximation but now, we will use it because we cannot afford to compute or exactly. So, instead, what we will do?

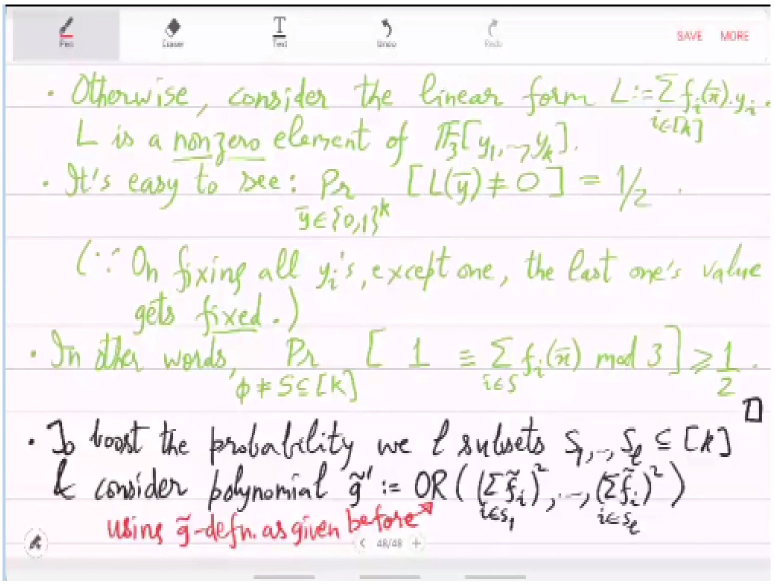
We will just randomly pick f_i's out of $f_1.,......f_k$. We will pick few f_i's and arithmetize. So, that is the idea. So, pick a random set $S \subseteq [k]$ and consider mod_3. So, it is not or it is actually mod_3 that we are looking at. So, pick a random subset and compute mod_3. mod_3 has the randomize that in case 2 you have already designed a polynomial for it, approximator polynomial and the degree of that was as required. It was not too big irrespective of how big S is.

S may even be k of size k but then there will be some errors introduced, because when you are picking a random set S, maybe, you were unlucky and all these values of f_i were 0; $f_i = 1$ was happening somewhere else and you missed that. So, we have to calculate that probability. What is that probability? So, for all inputs, $\Pr_{\phi \neq S \subseteq [k]} [\vee^k_{i=1} = mod_3 (f_i | i \epsilon S)] \geq 1/2$.

So, there is some success probability although it is not too high but what you can do is you can repeat this and then the error probability will reduce you can pick S many times ok. So, this will be a useful step. So, how do you show that r is equal to this mod_3 of a random set of f_i's on half of the inputs. So, in simplest form the idea is that suppose only one of the f_i's was 1; the rest were all 0.

So, what is the chance that you will pick this f_i which is 1 or what is the chance that you will not pick it. So, you have to show that the chance is half of picking it or not picking it. So, let us formalize that. So, if $\forall i, f_i(\overline{x}) = False$, then this happens with probability 1. So, this is an easy case. x says when all the f_i's are false then clearly or matches mod_3 function.

(Refer Slide Time: 15:57)



- Otherwise, consider the linear form $L := \sum_{i \in [k]} f_i(\overline{x}) \cdot y_i$. L is a nonzero element of $\mathbb{F}_3[y_1, \ldots, y_k]$.
- It's easy to see: $\Pr_{\overline{y} \in \{0,1\}^k} [L(\overline{y}) \neq 0] = 1/2$.

  ($\because$ On fixing all y_i's, except one, the last one's value gets fixed.)
- In other words, $\Pr_{\phi \neq S \subseteq [k]} [1 \equiv \sum_{i \in S} f_i(\overline{x}) \bmod 3] \geq \frac{1}{2}$. $\square$
- To boost the probability we ℓ subsets $S_1, \ldots, S_\ell \subseteq [k]$ & consider polynomial $\tilde{g}' := OR((\sum_{i \in S_1} \tilde{f}_i)^2, \ldots, (\sum_{i \in S_\ell} \tilde{f}_i)^2)$

  using $\tilde{g}$-defn. as given before

Otherwise consider the linear form L:= $\sum_{i \in [k]} f_i(\overline{x}) y_i$. So, consider this linear form in the variables y_1 ,...y_k . $f_i(\overline{x})$ are actually values. They are 0 or 1. We are thinking of them as 0 or 1. So, false is equated with 0. This L is a non-zero element of $F_3[y_1$,...y_k] Key thing is that it is a non-zero linear form mod 3. Why? Because one of the f_i is 1.

So, when you pick f_i's randomly and take a sum that is same as setting$y_i$is to 1 and the othery_i is, y_j is to 0. So, y_i is you can think of as being picked randomly 0 or 1. So, just observe that probability of picking $\overline{y}$ randomly in the space{ 0, 1}^{k} but not 0 probability that what is the probability that L($\overline{y}$) is 0 or not zero. When you pick $\overline{y}$ randomly value 0, 1, what is the chance that L ($\overline{y}$)does not vanish.

So, there are some, there is at least one y_i, which is appearing in L. So, actually only look at thosey_i's that appear in L and you are fixing them 0, 1. So, say, it easy $y_1 + y_2$ So, when you randomly fix y_1 and y_2, the chance that it will vanish is only half. Because whatever you will fix $y_1$or whatever you will pick y_1, y_2 has to cancel it. So, y_2's value will get fixed.

So, chance of picking that value ofy_2 is half. So, this vanishing the event of $L(\overline{y})$ vanishing, the probability is half. I should take all the$(\overline{y})$. So, this probability is exactly half. Since, on fixing all y_i's except 1 the last one gets fixed. So, out of 2 possible values, this last one can take only one. So, that is 1 over 2 that is the probability. So, in other words, the probability over s, 1 to k, you are picking a subset.

So, the probability that or matches $\Pr S \subseteq [k][\vee^{k}_{i=1} \equiv \sum_{i \in S} f_i(\overline{x}) \bmod_3]$

So,$L(\overline{y})$you are looking at that version. We are in the case where this is actually 1. So, this OR is already 1 and $L(\overline{y})$ that we are looking at is those y_i's such that i is in s. So, that is$\sum f_i$ for these i's. This sum is equal to 1. This probability we have shown is, is half. This is at least 1/2, which is what we wanted to show that is our claim.

So, for OR being equal to 0, probability is 1; OR being equal to 1, probability is at least 1/2. But this is success probability is only 1/2. We want to boost it. So, to boost the probability, we pick many subsets $S_1 \ldots\ldots S_{l \subseteq [k]}$and for each S_i, we do this experiment. What is the experiment? That you compute the sum of these particularf_i's according to the subset $\bmod_3$. So, you will get l values.

What do you do with them? You take OR. So, basically if even one of them is 1, then u output 1; if all of them are 0, then u output 0 hoping that original OR $f_i$was indeed 0. So, you consider the polynomial $\overline{g'} := OR((\sum_{i \in S_1} \overline{f}_i)^2 \ldots (\sum_{i \in S_l} \overline{f}_i)^2)$ so, this OR is the polynomial that we have defined in the here in the beginning, let us just remember this. So, this OR is using g tilde.

So, use the g tilde type of definition of OR. So, that will multiply the degree of these input approximators by l because you are using l polynomial inputs here or l functions here but l is small. So, this is not a big blow up; previously the blow up was by k that we have reduced to l.

(Refer Slide Time: 24:51)

So, observe that $deg\,\overline{g'} \leq\ l.2.(2l)^{h-1}$ So, we have the bound of $(2l)^h$ as required how well is the approximator approximating. So, $\forall$ x, Pr[$\overline{g'} \neq \vee_{i\epsilon[k]} f_i$]. g tilde prime is defined by this randomly chosen l subsets. So, that is what the probability is over.

Once you have picked $S_1 ..... S_l$, g tilde prime is defined, well defined polynomial and that value at $\overline{x}$ that polynomial at $\overline{x}$evaluates to something different from OR. What is the chance of that? So, that basically means that .S_l, the mod_3 value was wrong; at S_2 it was wrong; at S_1, it was wrong. Each time, this happens with error probability half.

So, this error probability is less than $1/2^l$, which means so, from this, we can deduce that there is actually a good choice of $S_1 ..... S_l$, such that this event happens. So, we are swapping the quantifiers now. We are swapping x bar with the choice $S_1 ..... S_l$. So, when you swap this, what you will get is that there exists subsets $S_1 ..... S_l$, such that the probability now over $\overline{x}$ of this bad event happening. This is equally small.

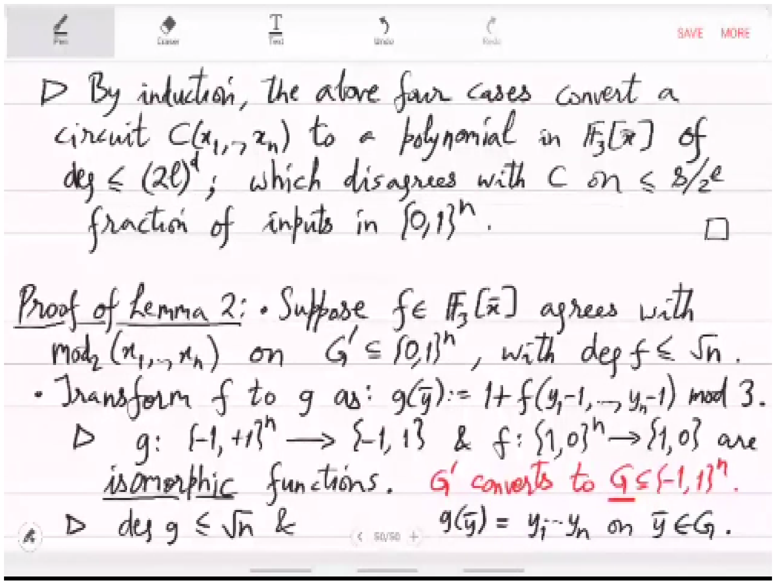
So, you just swap. Intuitively, you can formally prove it but intuitively why this should be true is because if suppose so, this being false means that for every choice of $S_1 ..... S_l$. The probability is bigger than$1/2^l$. for every choice of $S_1 ..... S_l$ but then that will contradict this for all $\overline{x}$ claim above. So, it is basically an averaging argument. So, we have deduced that there exists this good choice.

So, we should actually fix $S_1 ..... S_l$ to this and denote the corresponding g tilde prime as the final g tilde. So, we have the property that degree of g tilde is at most $(2l)^h$and introduces errors less than equal to 2^{-l}fractions. So, these many varies x's have been added during this design. So, that completes the description of OR approximator. Finally let us go to AND. g is AND gate say, g=$\wedge\ k_{i=1}\ f_i$.

So, you can apply NOT on this. So,$\neg g\ =\vee\ k_{i=1}(\neg f_i)$ and now what you will do is fromf_i tilde, you will define the approximator of NOT $f_i$and then you will use the OR design or approximated design given above that will give you approximator for g which is g tilde. So, we have reduced the question to cases 1 and 3. So, we have covered all the gates namely NOT mod_3 OR, AND.

The degree at height h of a gate approximator is$(2l)^h$and the errors that we introduce that error fraction is at most $1/2^l$.

(Refer Slide Time: 31:47)



▷ By induction, the above four cases convert a circuit $C(x_1,..,x_n)$ to a polynomial in $\mathbb{F}_3[\overline{x}]$ of $\deg \le (2l)^d$; which disagrees with C on $\le s/2^l$ fraction of inputs in $\{0,1\}^n$. $\square$

<u>Proof of Lemma 2:</u> • Suppose $f \in \mathbb{F}_3[\overline{x}]$ agrees with $mod_2(x_1,..,x_n)$ on $G' \subseteq \{0,1\}^n$, with $\deg f \le \sqrt{n}$.

• Transform f to g as: $g(\overline{y}) := 1 + f(y_1-1,..,y_n-1) \bmod 3$.

▷ $g: \{-1,+1\}^n \longrightarrow \{-1,1\}$ & $f: \{1,0\}^n \rightarrow \{1,0\}$ are <u>isomorphic</u> functions. G' converts to $\underline{G} \subseteq \{-1,1\}^n$.

▷ $\deg g \le \sqrt{n}$ & $g(\overline{y}) = y_1 \cdots y_n$ on $\overline{y} \in G$.

So, let us write that. So, by induction the above 4 cases convert a circuit $C(x_1 \ldots . x_n)$to a polynomialin F_3 [x] of deg$\le(2l)^d$ because the root has depth or height d; d is a constant. So, that is the degree bound on the final polynomial and the error is, so, which disagrees with C on at most $s/2^l$ remember that there are at most s gates. So, every gate can add $s/2^l$ fraction. So, this $s/2^l$ is the total fraction by the union bound.

So, in the space $\{0,1\}^n$, this is the fraction. So, that is exactly the statement of lemma 1. Lemma 1 is proved that ACC 0 circuit whatever the Boolean function, it computes can be approximated by a low degree polynomial on a good fraction of inputs. Let us now do lemma 2. What is the approximate degree for mod 2? So, suppose f which is a polynomial agrees with $mod_2(x_1 ,...x_n)$hich is again a Boolean function parity. It agrees on some inputs say, call it g prime and as degree low degree.

So, mod_2 is happening for n variables but the approximator has degree square root n. Now, what we want to show is that for this low degree polynomial,G' should be small.G'cannot be, I think in the lemma 2 statement we had 99 percent. So, we want to show that G' is less than 99. So, this is a very nice statement independent of circuits. This is actually a fact about just this parity function and it is approximator.

So, the way, we will show it is, we will first change the notation from 0, 1 to a sine plus minus 1 that will be very helpful. So, transform f to a g as: $g(\overline{y}) :=(y_1 -1,.....y_n-1)mod_3$, what did we do here? So, when y_i is or y_1, let us say 0 then y_1 -1 is minus 1 and when it is one then y_1 -1is 0. So, $g(\overline{y})$we will think about as$\pm$ 1. So, for –1, + 1, if you feed this in g, so, you are taking y_i to be let us see, you take y_1 to be minus 1 then what is y_1 -1? It is minus 2, which mod_3 is 1.

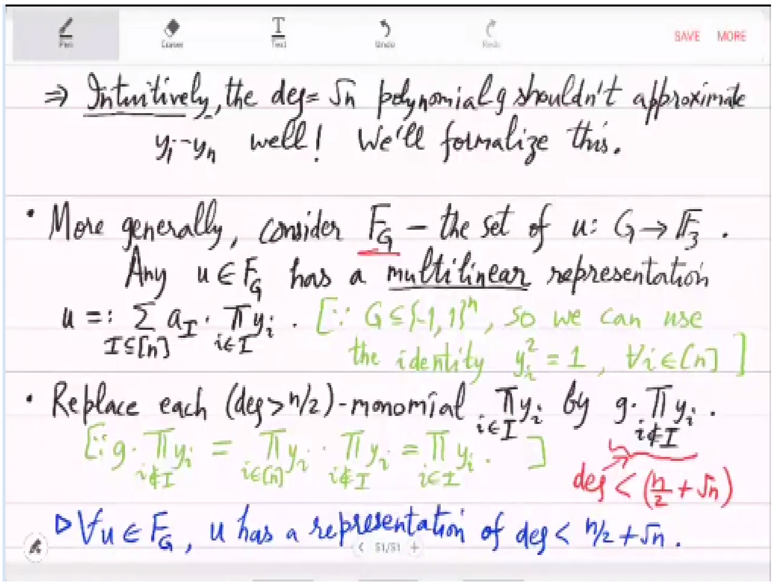
When you are taking y_1 to be 1, then y_1 -1 is 0. So, actually we are converting 1 to - 1, 0 to 1 and then this you will compute f, which will come out to be 0 or 1. So, we are adding +1 to make it 1 or 2 and 2 is -1. So, that is what we have done. So, g is a polynomial function that on values plus minus 1 evaluates to plus minus 1. So, g has become$\pm$ 1, f was 0, 1.

So, 1 goes to minus 1, 0 goes to + 1 and we have converted f to g. The whole problem statement is now converted from 0, 1 to plus minus 1. They are isomorphic functions. And degree has not changed. So, degree of g $\leq \sqrt{n}$ and on this, So, let us also define G' becomes G. So, $G'$was subset of $\{0,1\}^n$, which now, will convert to some other subset G in the new domain $\{-1,+1\}^n$.

So, on G, what is the property of on G, the subset G? What is the property of g? It is equal to the product. Because g is supposed to be mod_2, the parity in the 0, 1 setting. Now, when you move to +1, -1 setting, then the parity is just product. Because previously, if the number of once was even, now that would mean number of minus ones is even. So, sine will come out to be 1, which is the previous 0.

So, let me write this clearly that this equals on $\overline{y}$ in g. So, that is the advantage of this transformation that you get a product parity becomes just product.

(Refer Slide Time: 41:43)



$\Rightarrow$ Intuitively, the deg $=\sqrt{n}$ polynomial g shouldn't approximate $y_1 \cdots y_n$ well! We'll formalize this.

• More generally, consider F_G – the set of $u: G \to \mathbb{F}_3$.
Any $u \in F_G$ has a multilinear representation
$u =: \sum_{I \subseteq [n]} a_I \cdot \prod_{i \in I} y_i$. [$\because G \subseteq \{-1,1\}^n$, so we can use the identity $y_i^2 = 1$, $\forall i \in [n]$]

• Replace each (deg $> n/2$)-monomial $\prod_{i \in I} y_i$ by $g \cdot \underbrace{\prod_{i \notin I} y_i}_{\deg < (\frac{n}{2} + \sqrt{n})}$.
[$\because g \cdot \prod_{i \notin I} y_i = \prod_{i \in [n]} y_i \cdot \prod_{i \notin I} y_i = \prod_{i \in I} y_i$.]

$\triangleright$ $\forall u \in F_G$, u has a representation of deg $< n/2 + \sqrt{n}$.

So, intuitively, a degree $\sqrt{n}$ polynomial should not approximate this g polynomial. So,$\sqrt{n}$ polynomials degree $\sqrt{n}$ polynomial should not be able to approximate this product of $y_1, \ldots\ldots y_n$ well. That is the intuition. So, this is what we will formalize. So, how should we proceed with this formalization? So, what we will do is we will use this fact that there is an approximator for g of low degree.

There is an approximator of product $y_1, \ldots\ldots y_n$. A degree root n polynomial, the degree root n polynomial g should not approximate $y_1, \ldots\ldots y_n$ well. So, the way, we will formalize will get to this is, we will use this low degree approximator g to convert more general polynomials and reduce their degree. So, more generally consider F_G the set of functions from this domain G $\to 3F$. And we will now prove something about F_G using small g.

So, consider this set of functions from this domain big G taking values in the field$3F$ so, 0, 1 2 value. Now, the g will give you some property on these functions, all these functions. So, what you can do is any u here has a multi-linear representation, u as monomials y_i where big I is a subset of the variables, how many variables, n variables. So, there is a every u has a multi-linear representation. Why is it multi-linear?

Basically, we are saying that these monomials y_i's there, there is no square dividing it. So, the individual degree of $y_i$is either 1 or 0. It cannot be 2. So, why can we claim that? Because G is a subset of$\{ -1, +1\}^n$. So, we can use the identity by a square equal to 1; for all i. So, the point is that your domain does not have 0. Your domain is± value. So, in particular any variable y_i if you square it, you get 1.

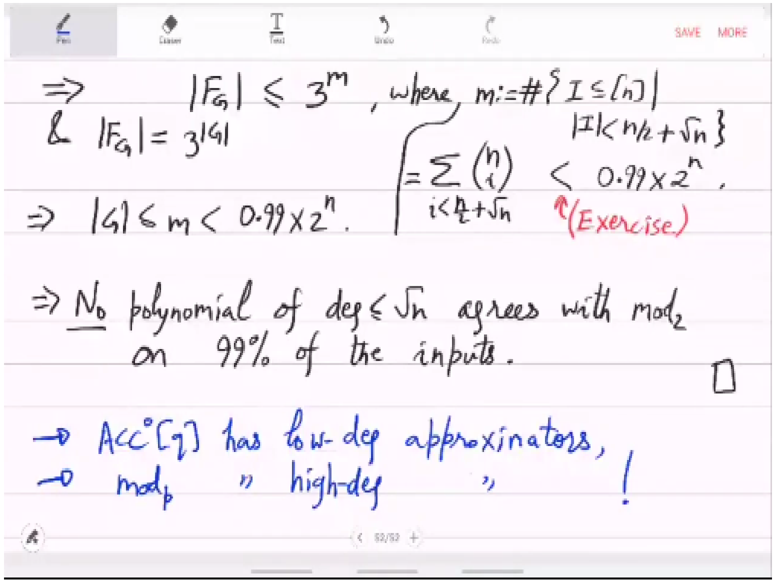
So, hence if your polynomial u if there is any monomial with y_i^2 appearing, you can replace it by 1. If there is a y_i^3, you can replace it by y_i and so on. So, individual degree of y_i in u can be made 0 or 1. So, hence it is a multi-linear polynomial representation over the field F_3. So, a_I's are 0, 1 or - 1. So, this will now help us together with small g. So, what we can do is this small trick?

So, replace each degree greater than n/ 2 monomial y_i by g times y_i, see, you can do this because g times product y_i, i not in I is the same as product of y_i, for all i and this multiplier $g. \prod_{i \notin I} y_i = \prod_{i \in [n]} \cdot \prod_{i \notin I} y_i$. Now, if you look at the same y_i, then you will get in the product y_i^2which will be 1. So, what you are left with is only those i's which are in I. So, we can write these monomials using little g. In this way, what is the advantage?

The advantage is previously that you had n /2 or more than n / 2 y_i now, previously you had something far bigger than n /2. Now, you will have almost n/ 2 because the degree of this is, this has deg $<(n/2+\sqrt{n})$because the set big I had more than n / 2 elements. So, the complement has less than n/ 2 plus the degree of g, which is at most $\sqrt{n}$.

So, any monomial whose degree is more than n/2 + $\sqrt{n}$, the degree falls after this transformation. In fact, what we have proved is this nice property that for every u, every function u in big F big G, u has a representation of degree less than n / 2+$\sqrt{n}$. So, the small g will actually have an impact on all these functions, all the functions on this domain big G. So, this is a strong property. What does this tell you?

(Refer Slide Time: 51:41)



$\Rightarrow \quad |F_G| \leqslant 3^m$, where $m := \#\{I \subseteq [n] \mid |I| < n/2 + \sqrt{n}\}$

& $|F_G| = 3^{|G|}$ $\qquad = \sum_{i < \frac{n}{2}+\sqrt{n}} \binom{n}{i} < 0.99 \times 2^n$. ↑(Exercise)

$\Rightarrow \quad |G| \leqslant m < 0.99 \times 2^n$.

$\Rightarrow$ No polynomial of deg $\leqslant \sqrt{n}$ agrees with mod_2 on 99% of the inputs. $\square$

$\rightarrow$ $AC^0[q]$ has low-deg approximators,

$\rightarrow$ mod_p " high-deg " !

So, this means that look at the size of $|F_G|$ How many functions can there be? So, the number of functions is 3^m, where m is the number of these low degree monomials. So, basically the function u in the representation, you have these a I's. They take 3 values potentially and how many I's are there, big I's are there? So, the number of big I is m, which is this how many n/2 + $\sqrt{n}$ and degree monomials are there.

And each a I takes 3 values. So, 3 into m, this is the maximum number of u's, you can have. That is one thing. So, let us estimate this m. So, this $\frac{n}{i}$, for all these i's. So, how many subsets are there of size i that is $\frac{n}{i}$ and instead of going all the way to n, we are stopping around n/2. So, if you stop in the middle, then you get half of 2^n but since you are, you might go up 2 n/ 2 +$\sqrt{n}$, there will be more than half of 2^n.

It will be, you can show that it is 0.99 times 2^n. Do this as an exercise. So, that is what m is. m is a fraction of 2^n. On the other hand, F_G in terms of the domain G, you get $3^{|G|}$. So, we are counting it in 2 ways. One is simply by the domain size of G and so, that gives you $3^{|G|}$ many functions. The other is more complicated which is by the representation of the function as a polynomial of low degree that gives you 3^m.

So, which implies that size of g is less than equal to m, which is less than fraction of 2^n. So, the domain cannot be large. That is what we wanted to show. So, no polynomial of degree$\leq \sqrt{n}$ agrees with |2| on 99% of inputs. Let us just see that. So, this finishes, this tells you that mod parity has high degree approximators.

And the first lemma told you that ACC 0 has low degree approximators. So, that is a contradiction That is what these are the 2 facts, we have shown very interesting techniques and this tells you that mod p cannot be in ACC 0 q for p and q different.