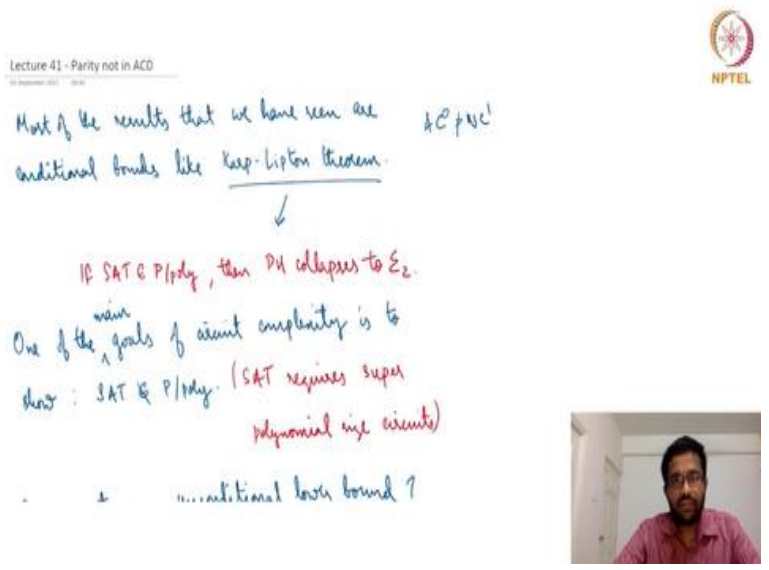
# Computational Complexity

**Prof. Subrahamanyam Kalyanasundaram**
**Department of Computer Science and Engineering**
**Indian Institute of Technology, Hyderabad**

## Lecture - 42
## Parity Not in AC 0: Part 2

**(Refer Slide Time: 00:15)**



Lecture 41 - Parity not in AC0

Most of the results that we have seen are conditional bounds like Karp-Lipton theorem. $AC^0 \neq NC^1$

$\downarrow$

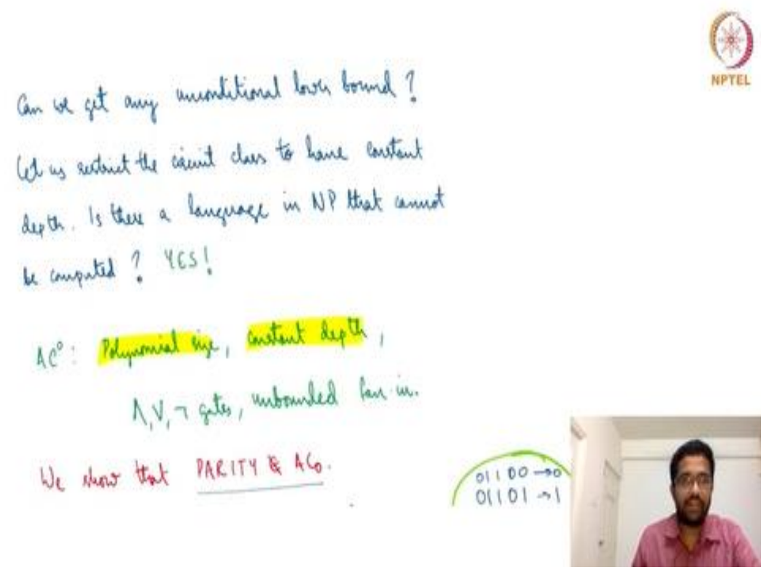
If SAT $\in$ P/poly, then PH collapses to $\Sigma_2$.

One of the main goals of circuit complexity is to show : SAT $\notin$ P/poly. (SAT requires super polynomial size circuits)

...nditional lower bound ?

Hello and welcome to lecture 42 of the course Computational Complexity. In the previous lecture, we started on showing the proof that parity is not in AC 0. We were following the proof technique by Rasbora and Smolensky.

**(Refer Slide Time: 00:38)**



Can we get any unconditional lower bound ?

Let us restrict the circuit class to have constant depth. Is there a language in NP that cannot be computed ? YES!

$AC^0$ : Polynomial size, constant depth,

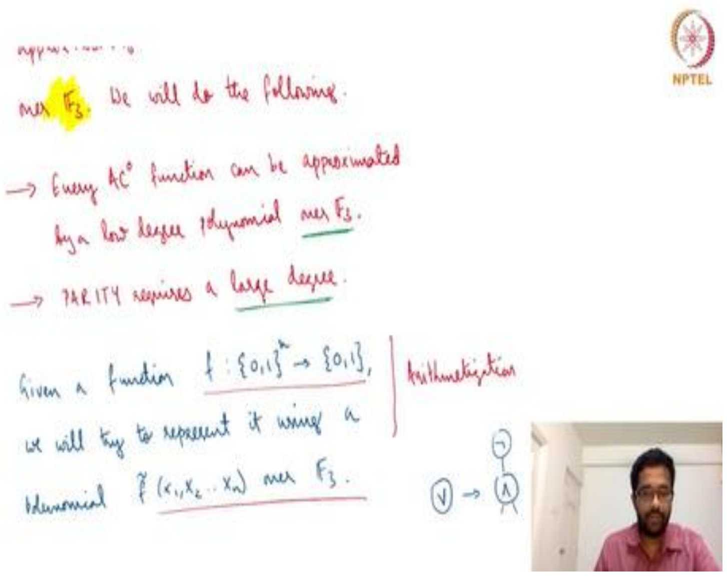
$\wedge, \vee, \neg$ gates, unbounded fan in.

We show that PARITY $\notin AC^0$.

$01100 \to 0$
$01101 \to 1$

The main motivation was to provide concrete meaning unconditional lower bounds on some circuit class.

**(Refer Slide Time: 00:45)**



over $\mathbb{F}_3$. We will do the following.

→ Every $AC^0$ function can be approximated by a low degree polynomial over $\mathbb{F}_3$.

→ PARITY requires a large degree.

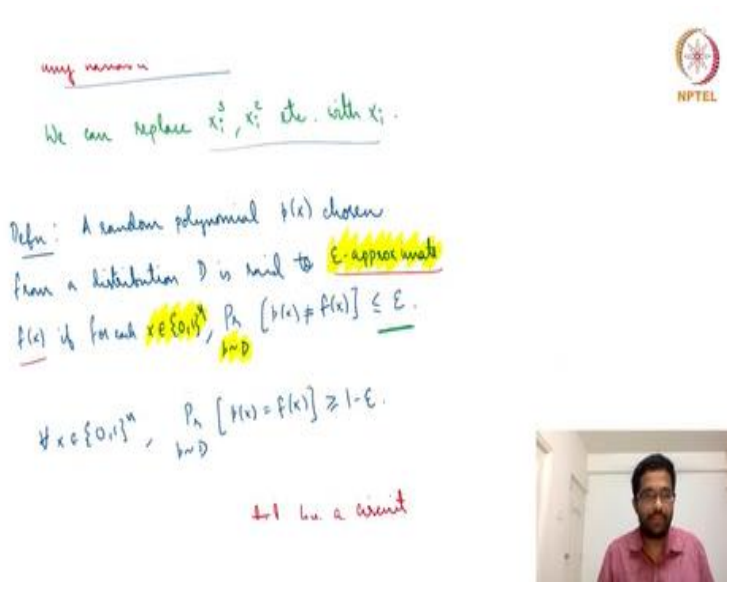
Given a function $f : \{0,1\}^n \to \{0,1\}$, we will try to represent it using a polynomial $\tilde{f}(x_1, x_2 \ldots x_n)$ over $\mathbb{F}_3$. | Arithmetization

$$\begin{tikzcd} & \neg \arrow[d, dash] \\ \vee \arrow[r] & \wedge \end{tikzcd}$$

I just very briefly outline what we have done so far on the proof itself. So, what we have done so far, this is a high-level outline of the proof. Every AC 0 function can be approximated by a low degree polynomial over the F 3 field, F 3 field is just the field which contains 0, 1 and 2 and addition and multiplication happens modulo 3. So, that is F 3, so every AC 0 function can be approximated by a low degree polynomial.

Second parity cannot be parity requires a large degree even to approximate. These are the two main theorems that we will show. In fact, we showed the first statement and a bit more than that.

**(Refer Slide Time: 01:40)**



[illegible]

We can replace $x_i^3, x_i^2$ etc. with $x_i$.

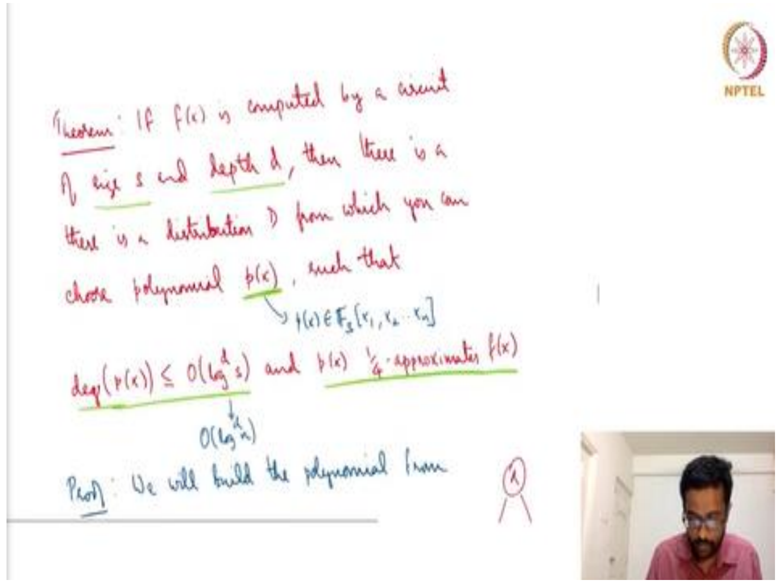
Defn: A random polynomial $p(x)$ chosen from a distribution $D$ is said to $\varepsilon$-approximate $f(x)$ if for each $x \in \{0,1\}^n$, $\Pr_{p \sim D}[p(x) \neq f(x)] \leq \varepsilon$.

$\forall x \in \{0,1\}^n, \Pr_{p \sim D}[p(x) = f(x)] \geq 1-\varepsilon$.

Let [illegible] be a circuit

So, the first statement was showed that, every AC 0 function can be approximated by a low degree polynomial.

**(Refer Slide Time: 01:48)**



Theorem: If $f(x)$ is computed by a circuit of size $s$ and depth $d$, then there is a there is a distribution $D$ from which you can chose polynomial $p(x)$, such that $\rightarrow p(x) \in \mathbb{F}_3[x_1, x_2 \cdots x_n]$

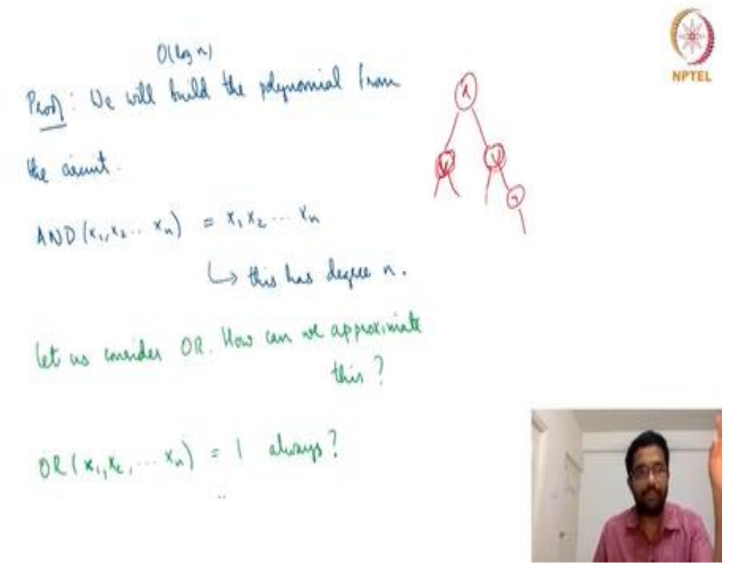
$\deg(p(x)) \leq O(\log^d s)$ and $p(x)$ $\frac{1}{4}$-approximates $f(x)$

$\downarrow O(\log^d n)$

Proof: We will build the polynomial from

So, by low degree what we meant is, if it is a circuit of size s and depth d then we could approximate with a log s to the power d degree polynomial and the error of that is like 1 by 4. So, let say 1 by 4 is some fixed number. So, it could be 1 by 10 also then still some constant factor will change, but we will get roughly this. So, log s is the size of the circuit so it is like log to the power d because s is just n to the power some k.

**(Refer Slide Time: 02:29)**



$O(\log n)$

<u>Proof</u>: We will build the polynomial from the circuit.

$\text{AND}(x_1, x_2 \ldots x_n) = x_1 x_2 \ldots x_n$

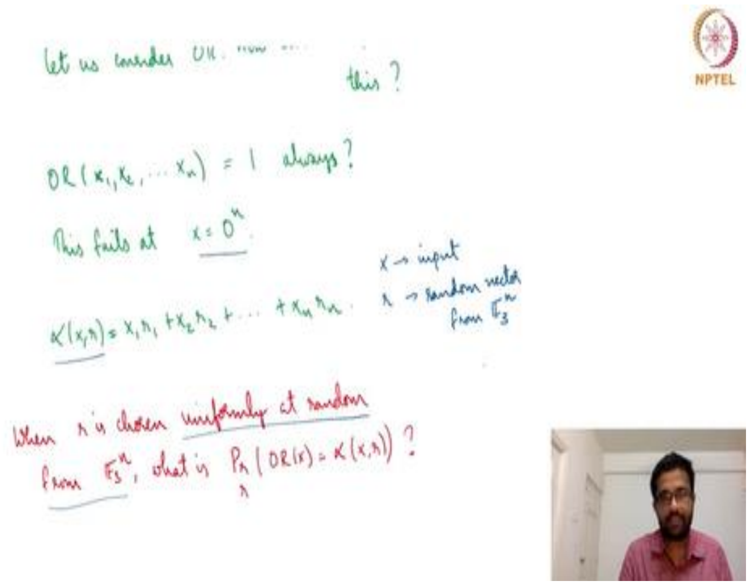
↳ this has degree $n$.

Let us consider OR. How can we approximate this?

$\text{OR}(x_1, x_2, \ldots x_n) = 1$ always?

The main idea was to build a circuit where we replace AND gate, OR gate and NOT gate with the corresponding polynomials. And what we did in this proof is to show that we could approximate OR gates.

**(Refer Slide Time: 02:45)**



Let us consider OR. How … this?

$\text{OR}(x_1, x_2, \ldots x_n) = 1$ always?

This fails at $\underline{x = 0^n}$.

$\underline{\alpha(x, r)} = x_1 r_1 + x_2 r_2 + \ldots + x_n r_n$.

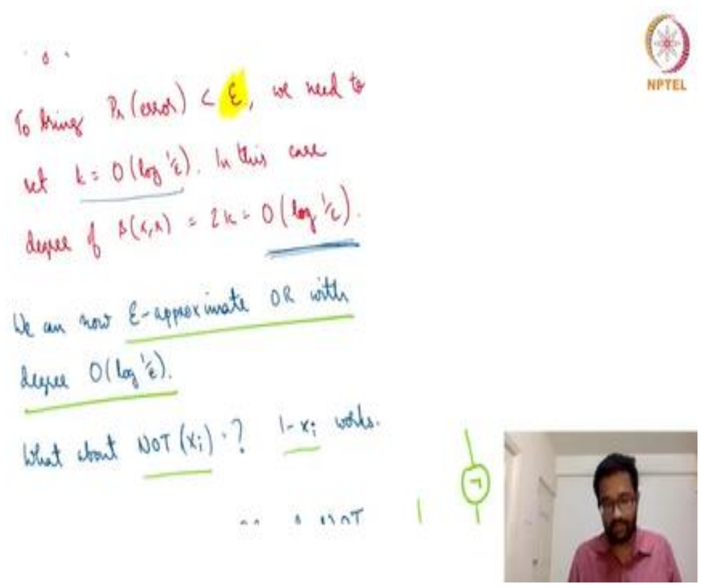
$x \to$ input

$r \to$ random vector from $\mathbb{F}_3^n$

When $r$ is chosen <u>uniformly at random</u> <u>from $\mathbb{F}_3^n$</u>, what is $\Pr_r(\text{OR}(x) = \alpha(x, r))$?

NOT gates are straight forward because it is just corresponding to the polynomial 1- x and AND gate can be derived from or gate using the De Morgan laws without any extra degree.

**(Refer Slide Time: 03:00)**



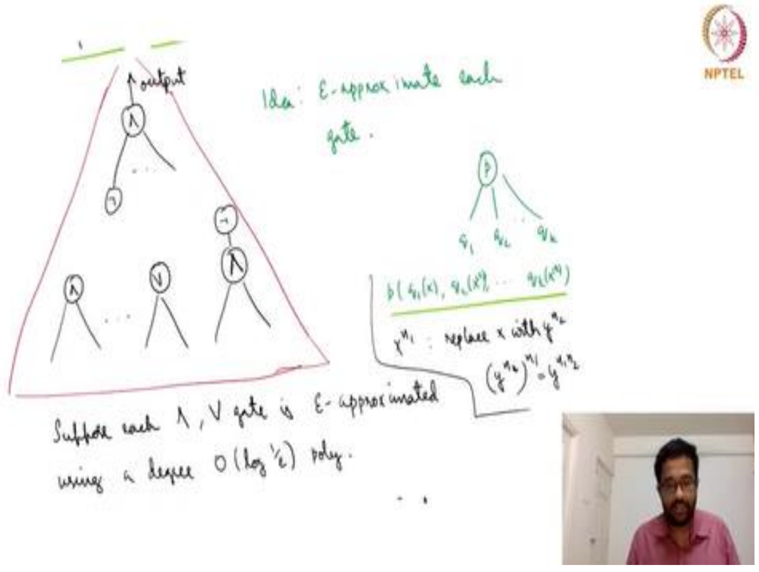
To bring $\Pr(\text{error}) < \varepsilon$, we need to set $k = O(\log 1/\varepsilon)$. In this case degree of $p(r,x) = 2k = O(\log 1/\varepsilon)$.

We can now $\varepsilon$-approximate OR with degree $O(\log 1/\varepsilon)$.

What about $\text{NOT}(x_i)$? $1 - x_i$ works.

So, all we had to do was to build an OR gate which we did and we built an OR gate that had degree order log 1 by Epsilon, where Epsilon was a desired error. Now we just use these OR gates AND gates everything NOT gates to replace to so polynomial corresponding to the OR gates NOT gates AND gates to in the circuit to build a jumbo polynomial that would approximate the original circuit.

**(Refer Slide Time: 03:33)**



Idea: $\varepsilon$-approximate each gate.

$p(q_1(x), q_2(x), \ldots q_k(x))$

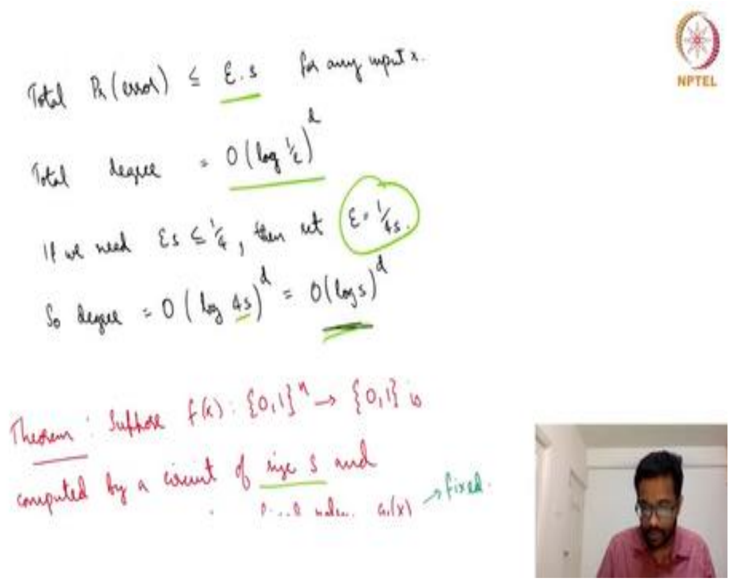
$x^{n_1}$ : replace $x$ with $y^{n_2}$

$(y^{n_2})^{n_1} = y^{n_1 n_2}$

Suppose each $\wedge, \vee$ gate is $\varepsilon$-approximated using a degree $O(\log 1/\varepsilon)$ poly.

So, the jumbo polynomial has like the error is approximated by the union bound. So, we choose epsilon small enough so that epsilon times the size is also less than 1 by 4. The degree of any individual polynomial to the power the depth of the circuit because it should be a Nitridated polynomial of a polynomial of a polynomial.

**(Refer Slide Time: 03:57)**



Total $\Pr(\text{error}) \leq \underline{\varepsilon . s}$ for any input x.

Total degree $= \underline{O(\log 1/\varepsilon)^d}$

If we need $\varepsilon s \leq 1/4$, then set $\boxed{\varepsilon = \frac{1}{4s}}$.

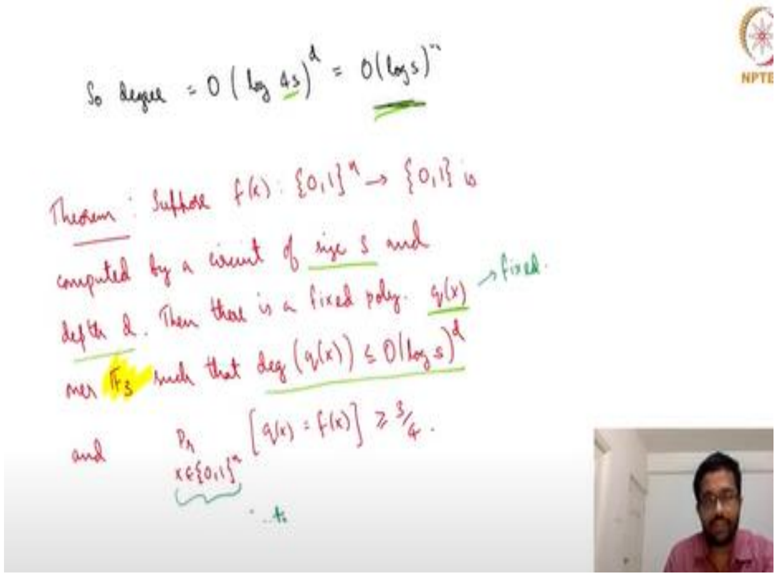
So degree $= O(\log 4s)^d = \underline{O(\log s)^d}$

<u>Theorem</u>: Suppose $f(x) : \{0,1\}^n \to \{0,1\}$ is computed by a circuit of <u>size s</u> and

$q_i(x) \to$ fixed.

So, therefore if you do the math, you will get that epsilon has to be 1 by 4 s and the degree will be order log s whole power d. So, this is not a single polynomial but a family of polynomials rather a polynomial chosen from a distribution at some using some random bits and which will so and when say epsilon, here the epsilon is over the choice of the randomness and for any fixed input. So, even after you fix the input the probability of success is at least three fourth.

**(Refer Slide Time: 04:32)**



So degree $= O(\log 4s)^d = \underline{O(\log s)^d}$

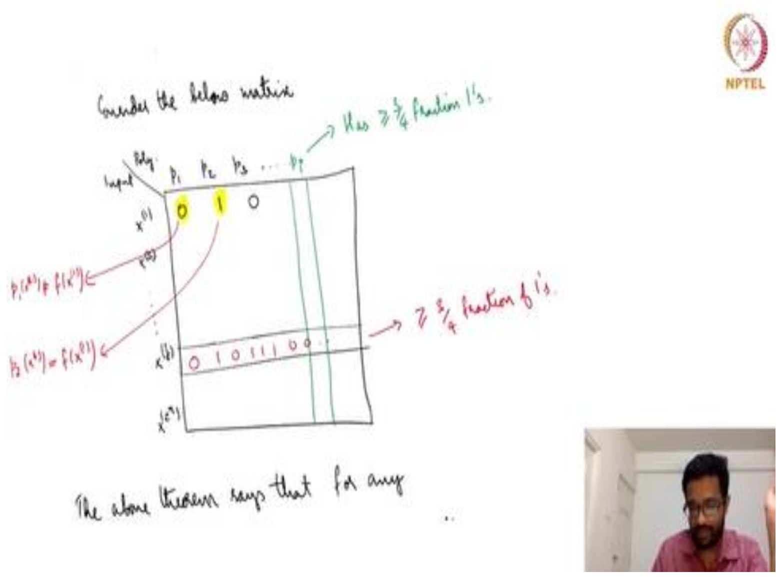
<u>Theorem</u>: Suppose $f(x) : \{0,1\}^n \to \{0,1\}$ is computed by a circuit of <u>size s</u> and <u>depth d</u>. Then there is a fixed poly. $\underline{q(x)}$ $\to$ fixed. over $\mathbb{F}_3$ such that $\underline{\deg(q(x)) \leq O(\log s)^d}$

and $\Pr_{x \in \{0,1\}^n} [q(x) = f(x)] \geq 3/4$.

And then we showed that if we have a family and a distribution that epsilon approximates or 1 by 4 approximates. Then there is a single polynomial, there is one polynomial in that in the distribution. That will actually give us, at most one fourth error if you choose the input at random. So, now from the situation where we want for any input at most one fourth error over where the error is computed or the probability is computed over the charge of the polynomial.

You can move to a situation where polynomial is fixed the error the probability is over the choice of the input.

**(Refer Slide Time: 05:22)**



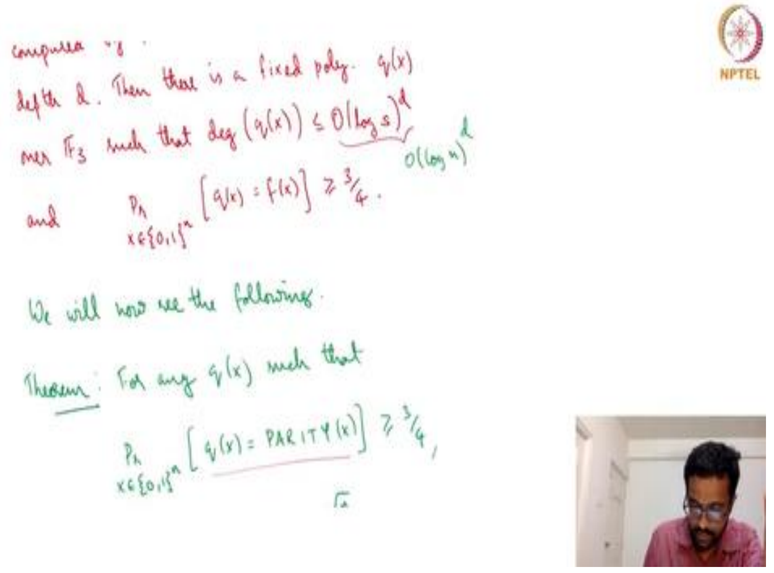
Consider the below matrix

Poly.
Input
$p_1$ $p_2$ $p_3$ ... $p_t$
→ Has $\geq \frac{3}{4}$ fraction 1's.
$x^{(1)}$
$x^{(2)}$
$x^{(i)}$
$x^{(2^n)}$
0 1 0
0 1 0 1 1 1 0 0 ..
$p_1(x^{(1)}) \neq f(x^{(1)})$
$p_2(x^{(1)}) = f(x^{(1)})$
→ $\geq \frac{3}{4}$ fraction of 1's.

The above theorem says that for any

This was a very simple argument, using some counting based argument. So, the main technique for the first part was this arithmetization how to write the OR using some polynomial.

**(Refer Slide Time: 05:38)**



computed by

depth d. Then there is a fixed poly. $q(x)$

over $\mathbb{F}_3$ such that $\deg(q(x)) \leq \underbrace{O(\log s)^d}$ $O(\log n)^d$

and $\Pr_{x \in \{0,1\}^n} [q(x) = f(x)] \geq 3/4$.

We will now see the following.

Theorem: For any $q(x)$ such that

$\Pr_{x \in \{0,1\}^n} [q(x) = \text{PARITY}(x)] \geq 3/4$,

$\sqrt{n}$

So, now we are starting off from here, suppose there is a function that is computed by a size s and depth d circuit then there is a fixed polynomial. We will call it q x so this is fixed over f 3 such that the degree is ordered longest power d and probability is chosen over the inputs. Then what is the probability that the q agrees with the function that we want to compute and this is at least three fourths.

So, this three fourth also choices somewhat arbitrary I could have made it nine tenth or something. It would still you could get a similar proof. So, this is where we are starting from, maybe I have repeated in this note as well the second notes as well 42 notes. So, if we have a Boolean function computed by depth d and sizes circuit. Then you could have a there is a fixed polynomial that has degree log s power d.

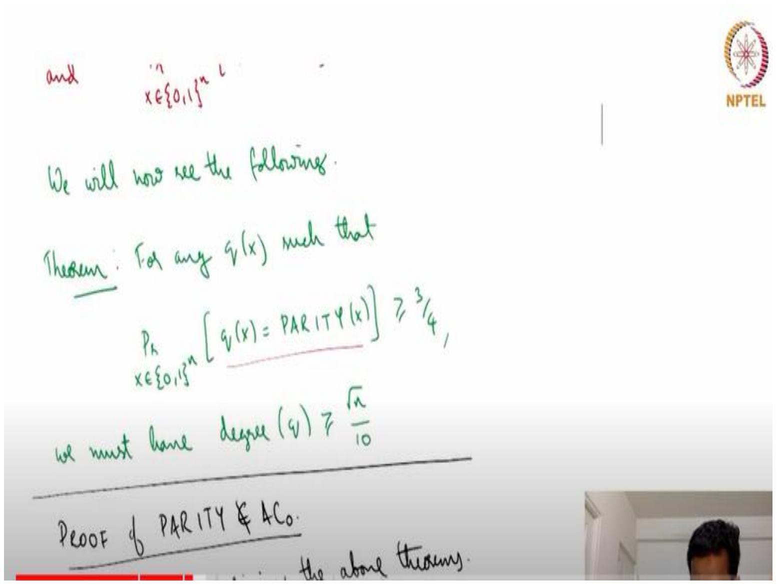
And approximate not even approximated such that the polynomial agrees with the function on 75% of the inputs. So, this is what we will see today and so this is what we already saw. What we will see in this lecture is that for any fixed polynomial that for which agrees with the parity function on 75% of the inputs. This polynomial should have degree at least square root n divided by 10. So, any fixed polynomial that agrees with parity function.

Recall that parity of x being a bit string is one if and only if an odd number of ones in x are 1 or if x contains an odd number of ones, if x contains an even number of one parity is 0. It just the bit twice like or it is the x or computer over the bits. So, what we are saying is that if q x or if parity is computed by a part or computed by a polynomial in this manner like this as the polynomial agrees with parity on 75% of the inputs.

Then the degree of the polynomial should be actually square root and divided by 10 and what we are saying here is that for any function that is computed by a sizes and depth d circuit then if it agrees with the any polynomial that agrees with it there is a polynomial that agrees with it on with degree at most log s power d. So, when it is an AC 0 circuit, it is like this log x power d is order log n power some constant.

Because order log s is just k times log n for some constant depth d is a constant. So, this is so log n power d is much smaller than square root n.

**(Refer Slide Time: 08:47)**



and $x \in \{0,1\}^n$

We will now see the following.

Theorem: For any $q(x)$ such that

$$\Pr_{x \in \{0,1\}^n}\left[q(x) = \text{PARITY}(x)\right] \geq 3/4,$$

we must have $\text{degree}(q) \geq \frac{\sqrt{n}}{10}$

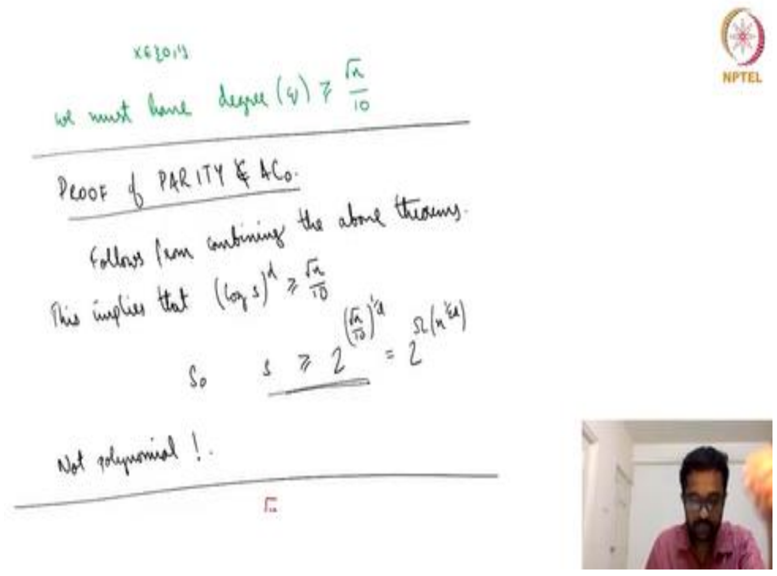
PROOF of PARITY $\notin$ AC$_0$.

the above theorems.

Here we have the second theorem, we are saying that any polynomial that approximates parity in this manner requires square root and degree. Here we are saying log n power constant is enough so that is the contradiction. So, because of that we cannot compute the parity using an AC 0 circuit. This is the contradiction in fact we will see a bit more detail than this. So, we will see, so we will actually compute how much size can the parity circuit have.

If it has constant depth and we will see that the size is more than polynomial. So, this is what we said if you combine the above theorems, what it is saying is that if parity has to be approximated by polynomial in this matter degree has to be at least square root n divided by 10 which means, order log s to the power d has to be square root and divided by 10.

**(Refer Slide Time: 09:49)**



$x \in \{0,1\}^n$

we must have $\text{degree}(q) \geq \frac{\sqrt{n}}{10}$

PROOF of PARITY $\notin$ AC$_0$.

Follows from combining the above theorems.

This implies that $(\log s)^d \geq \frac{\sqrt{n}}{10}$

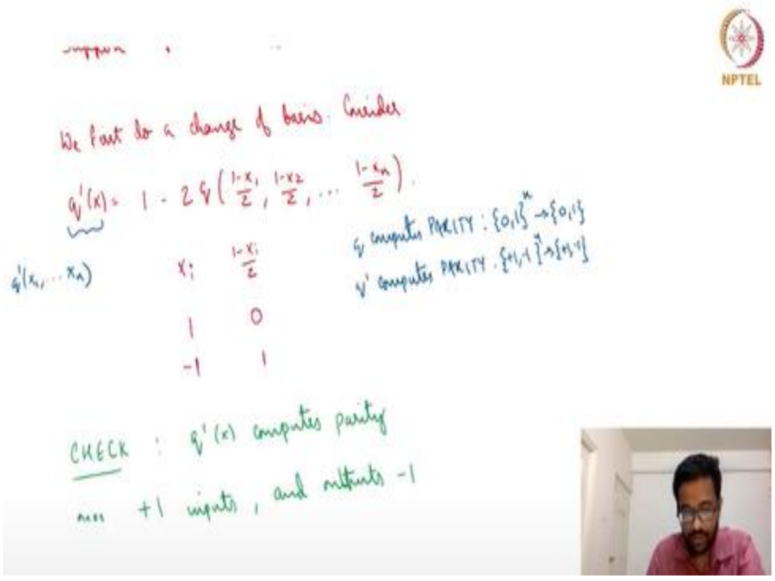
So $s \geq 2^{\left(\frac{\sqrt{n}}{10}\right)^{1/d}} = 2^{\Omega(n^{1/2d})}$

Not polynomial!.

Which means s has to be square root 2 power square root n divided by 10 whole power or power 1 by d which you can just write it as 2 power n power 1 by 2d. So, any circuit that computes parity must have size at least this quantity, so which is this is clearly super polynomial this is like something exponential that is the reason why parity is not in AC 0. So, now what remains is to show the proof of this theorem.

That any q that agrees with parity on 75% of the inputs must have square root and degree, square root n and divided by 10 degree.

**(Refer Slide Time: 11:05)**



We first do a change of basis. Consider

$q'(x) = 1 - 2q\left(\frac{1-x_1}{2}, \frac{1-x_2}{2}, \dots \frac{1-x_n}{2}\right)$.

$q'(x_1, \dots x_n)$

| $x_i$ | $\frac{1-x_i}{2}$ |
|---|---|
| 1 | 0 |
| -1 | 1 |

q computes PARITY : $\{0,1\}^n \to \{0,1\}$

q' computes PARITY . $\{+1,-1\}^n \to \{+1,-1\}$

CHECK : $q'(x)$ computes parity

... +1 inputs, and outputs -1

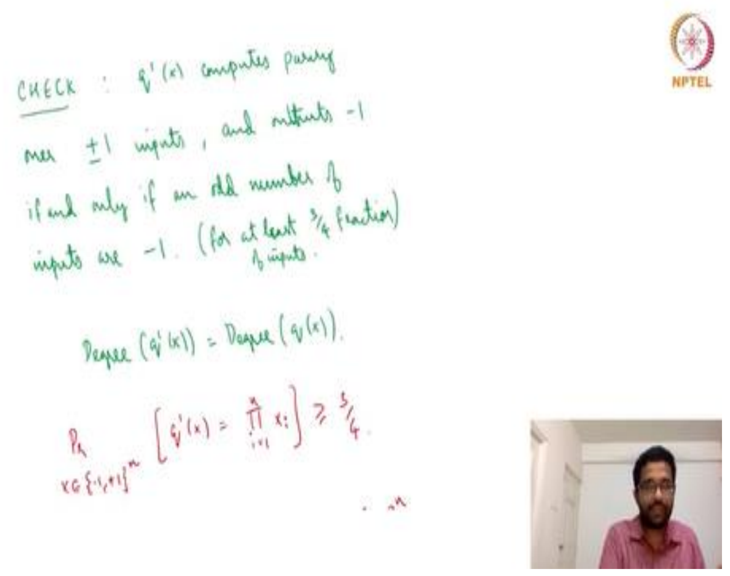
So, we will shoot for a contradiction. Again, this proof is also very simple and uses simple principles but this uses a bit of linear algebra and will be shorter than what we saw in the previous lecture. Suppose there was such a q with degree less than square root n and divided by 10. So, first let us do a change of basis. So, let us define q prime, and what is q prime? It is 1- 2 q of this, so where we are replacing the variables.

So, the variables of q are x 1, x 2 etcetera. So, now we are replacing it with 1 – x, 1 divided by 2, 1 –x, 2 divided by 2 and so on up to 1 – x n divided by 2. So, what have we done here when x i is 0 or x i is 1? q computes parity on Boolean input, this is what it computes 0, 1 power n. Now the claim is that q prime computes parity on - 1, +1 or rather maybe I should see the opposite, just to be consistent +1, -1 power n.

So, suppose x i so suppose the input of q prime is actually so q prime takes as input x 1 up to x n and think of x 1 etcetera coming from plus minus 1 not 0 1. Now what is happening here is that if x 1 is 1 then 1 – x 1 is 0. So, 1-1 divided by 2 is also 0. However, if x 1 is -1 then 1-1 is actually 2 and 2 divided by 2 is actually 1. So, when x i is 1 it is gets map to 0 when x i is minus 1 it is get map to 1 and q computes a parity over the boolean inputs.

It checks with an odd number of ones are there or an even number of ones are there. So, if there is an odd number of ones it outputs 1. If it outputs 1 then we get 1 - 2 in the right-hand side which is 1 - 2 is actually minus 1, if it outputs 0, we get 1.

**(Refer Slide Time: 14:25)**



CHECK : $q'(x)$ computes parity over $\pm 1$ inputs, and outputs $-1$ if and only if an odd number of inputs are $-1$. (for at least $3/4$ fraction of inputs.

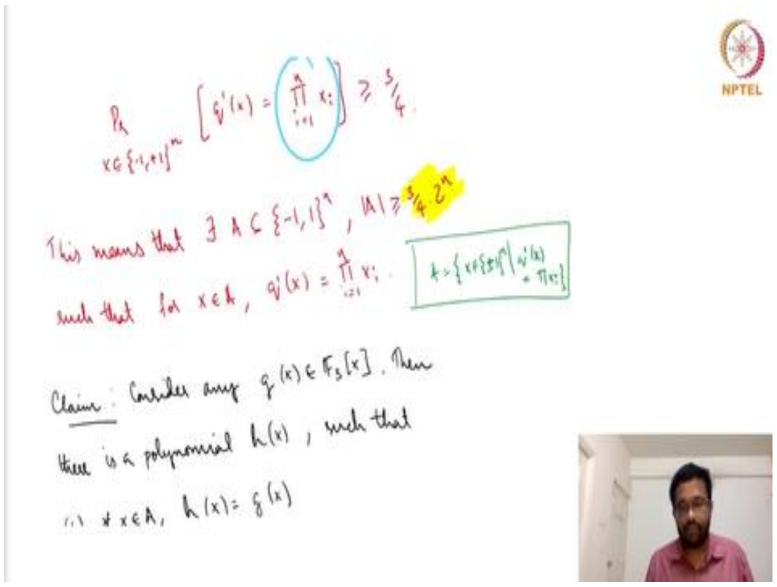
Degree $(q'(x))$ = Degree $(q(x))$.

$$\Pr_{x \in \{-1,+1\}^n} \left[ q'(x) = \prod_{i=1}^{n} x_i \right] \geq 3/4.$$

So, you can check that q prime computes parity over plus minus 1 inputs and outputs -1 if and only if an odd number of my inputs are -1. So, it is exactly like 0, 1 but it is like we replace 0 with +1 and 1 with -1. So, this helps because, now we see why it helps? We can now write a simple polynomial for this and that will help us in some aspects. The fact is that q prime simply does a change of basis from q going from 0 1 to plus minus 1.

It does so without any change of degree it is just linear operations here 1 – 2 q, so degree of q prime is the same as degree of q. Since q computes parity or q is supposed to compute parity on at least 75% of the inputs, q prime also computes parity on the same 75% of the inputs but q prime has this change of basis. So, that where we are now.

**(Refer Slide Time: 15:33)**



$$\Pr_{x \in \{-1,+1\}^n}\left[q'(x) = \prod_{i=1}^{n} x_i\right] \geq 3/4.$$

This means that $\exists A \subseteq \{-1,1\}^n$, $|A| \geq 3/4 \cdot 2^n$ such that for $x \in A$, $q'(x) = \prod_{i=1}^{n} x_i$.

$$\boxed{A = \{x \in \{\pm 1\}^n \mid q'(x) = \prod x_i\}}$$

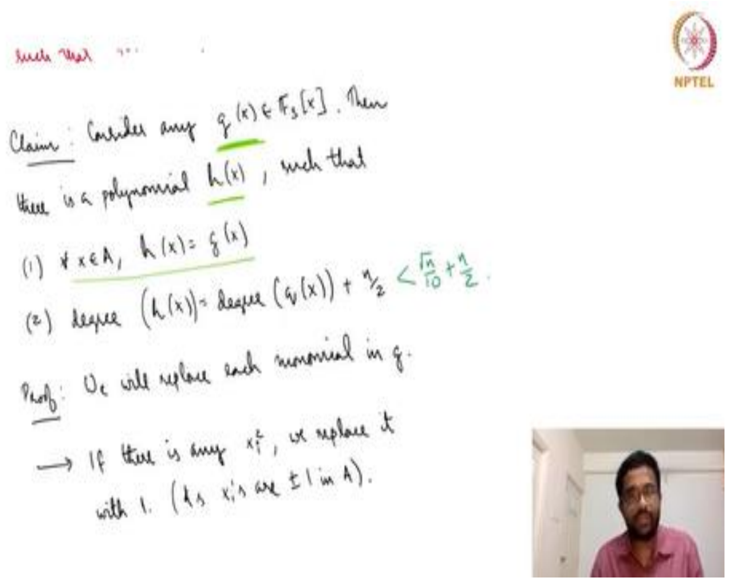
Claim: Consider any $q(x) \in \mathbb{F}_3[x]$. Then there is a polynomial $h(x)$, such that $\forall x \in A$, $h(x) = q(x)$

And in this + - 1 basis what is parity? Parity is just the product. So, what is the product of x size where the x size come from + - 1? So, it will be just how many -1 are there if there is an odd number of -1 the product will be -1. If there is an even number of -1 the product will be +1 which is exactly what parity has been defined to be. So, we can replace parity with this term over here or this term over here, product of all the individual x size.

So, what is the probability that q prime x is equal to this. It is the same as parity, it is at least three fourths. This means that there is a set A, it agrees with the parity on three quarters many inputs, so let those inputs on which q prime agrees with parity we call the set A. So, in other words maybe I can write here in other words, A is the set of all x + - 1 power n such that q prime x is equal to the product of x i.

And we know that A agrees with parity or the q prime agrees with parity on 75% of the inputs, so A is of size at least 75% of the entire domain. So, the domain is + - 1 power n which is of is 2 power n 75% is three fourths of 2 power n. On this many inputs q prime agrees with parity.

**(Refer Slide Time: 17:30)**



such that ...

Claim: Consider any $g(x) \in \mathbb{F}_3[x]$. Then there is a polynomial $h(x)$, such that

(1) $\forall x \in A$, $h(x) = g(x)$

(2) degree $(h(x))$ = degree $(q(x)) + n/2 < \frac{\sqrt{n}}{10} + \frac{n}{2}$.

Proof: We will replace each monomial in $g$.

$\longrightarrow$ If there is any $x_i^2$, we replace it with 1. (As $x_i$'s are $\pm 1$ in A).

Now we will see an interesting claim that if now consider any polynomial h or any function g over F 3 for any g it does not matter what it is now, we can write a polynomial that has the following properties then restricted to A, h = g, so A is this set where q prime is the same as parity. So, under that set this polynomial and this function will be the same under that set the polynomial will equal the function.

And then the degree of h will be degree of q whatever it is, so by assumption degree of q was less than square root n divided by 10 this is what we assumed. Degree of h is degree of q + n divided by 2. So, degree of q in fact we can go on and write that degree of q is actually assumed to be square root n by 10 + n by 2 we have by assumption we already have this. So, whatever be the function now we are saying that if there is such a polynomial.

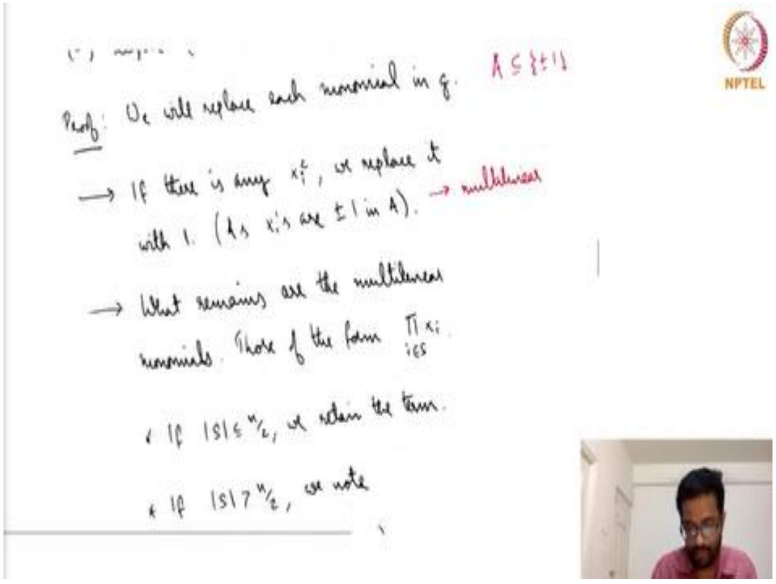
Whatever be the function just because parity can be approximated that implies that any function can be approximated by a polynomial of degree very close to n by 2 and that seems unbelievable and that is what will give us a contradiction. So, it is saying that any function that you take as long as you are only looking at this the set A, so we call it a is 75% of the inputs. If you are restricting yourself to the set a then you can approximate it with a polynomial of degree roughly n by 2+ something like square root n.

So, square root n is much smaller than n so, the point is that there are too many functions g, g is could be any function. And if we restrict ourselves to low degree polynomials or the polynomials of degree n by 2, they are not enough polynomials. So, the number of functions and number of polynomials will be there is a huge difference and that will give us a contradiction.

So, now first let us prove the claim why this is true. So, this claim is true as assuming whatever we have so far, so we assume that q is of degree the parity approximation polynomial q of degree at least at most square root n and divided by 10 and that gave us this particular q prime and that will give us this claim. So, we have a function g and we will see how to get a polynomial h that approximates g. So, first of all g has a polynomial, so g is a function.

So, g you can write down the entire polynomial in f. First thing enter polynomial over f 3, so there will be huge polynomial. First thing to note is that g is over so we are only bothered about the elements in A, so A is a subset of + - 1 whole power n. So, inside this if there is any x i's squared or x size cube x i's squared, we can replace it with 1. Because all x is are + - 1.

**(Refer Slide Time: 21:38)**



$A \subseteq \{\pm 1\}$

Proof: We will replace each monomial in g.

→ If there is any $x_i^2$, we replace it with 1. (As $x_i$'s are ±1 in A). → multilinear

→ What remains are the multilinear monomials. Those of the form $\prod_{i \in S} x_i$.

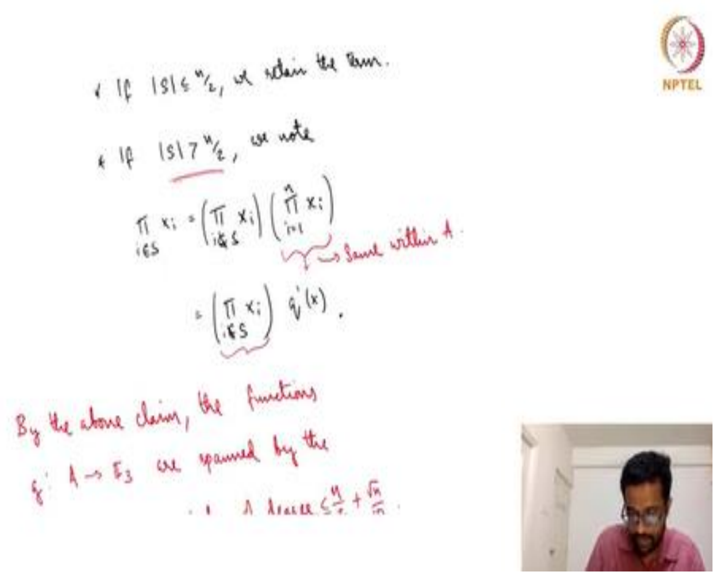
* If $|S| \le n/2$, we retain the term.

* If $|S| > n/2$, we note

So, x i's squared is always 1. So, any term like if you see x i's power 10 you can make it 1 x power 11 x i power 10 is 1 so x i power 11 is just x i. So, all that will remain is all the terms will be either degree 1 or 0 so it will be multi-linear polynomial. So, this will give us multi-linear polynomial. So, you start with the polynomial for g now we are simplifying it so multi-linear polynomial it is still fine as long as we are restricting ourselves to A.

And that is all that we care about. Now you look at each monomial. So, each monomial will be of the form x i, a product of x i where i comes from some set s where s is some subset of 1 to n. Now if this set is of size smaller than n by 2 then we just retain the term because again the target degree that we have is something like n by 2 + some small thing.

**(Refer Slide Time: 22:53)**



* If $|S| \le n/2$, we retain the term.

* If $\underline{|S| > n/2}$, we note

$$\prod_{i\in S} x_i = \left(\prod_{i\notin S} x_i\right)\underbrace{\left(\prod_{i=1}^{n} x_i\right)}_{} \rightsquigarrow \text{Same within } A.$$

$$= \underbrace{\left(\prod_{i\notin S} x_i\right)}_{} q'(x).$$

By the above claim, the functions $g: A \to \mathbb{F}_3$ are spanned by the

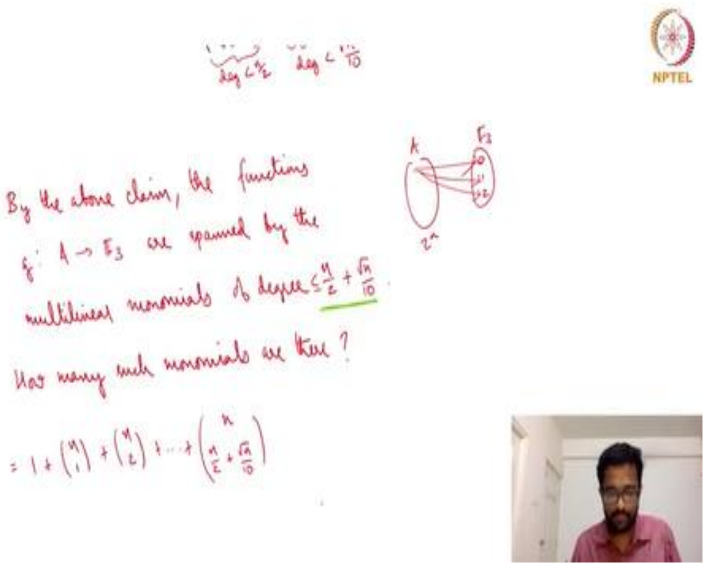
$\therefore$ A degree $\le \frac{n}{2} + \frac{\sqrt{n}}{10}$.

So, if this s is smaller than n by 2, we are fine, the degree will be smaller than n by 2. If it is bigger than n by 2 then we have to do something. So, all that we need to notice is that if s is bigger than n by 2 you could write it as the product of all the i multiplied by the product of i that are not in s, so product of all the i which is the second term here multiplied by the product of i that are not in s.

Because the i; that is not in s, appear twice and x i squared as I said already is equal to 1, so all that will remain is all the i that are in s. But what is the product of all the x i? It is simply inside A it is simply q prime x. This is the assumption, within A q prime and this are same. So, this is the same within A and we are only carrying we only care about what is there within A so we get this.

Since s is in this part, s is assumed to be of size bigger than n by 2 the complement of s will be always less than n by 2 or at most n by 2. So, this is degree less than n by 2 and this is degree less than square root n by 10 and gets as a polynomial. So, we got a polynomial of degree n by 2 + square root n by 10.

**(Refer Slide Time: 24:37)**



$\underbrace{\cdots}_{\deg < n/2}$ $\underbrace{\cdots}_{\deg < \frac{\cdots}{10}}$

By the above claim, the functions $f: A \to \mathbb{F}_3$ are spanned by the multilinear monomials of degree $\leq \underline{\frac{n}{2} + \frac{\sqrt{n}}{10}}$.

How many such monomials are there?

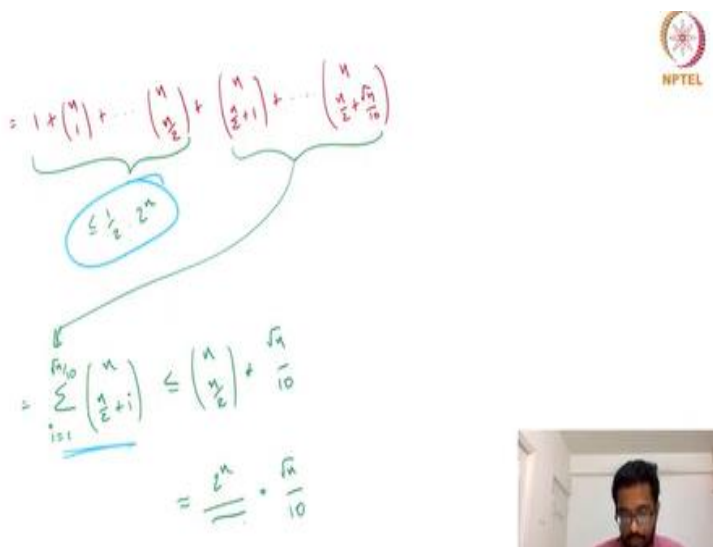
$= 1 + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{\frac{n}{2} + \frac{\sqrt{n}}{10}}$

Now where is the contradiction? The contradiction is so again the claim is now proved or this particular claim is now proved now we will get a contradiction. As I already said how many functions are there consider all the functions that go from A to F 3, so the number of functions is actually 3 power A, these because each entry in A could be 0, 1 or 2. Each entry in a could be mapped to 0, 1 or 2.

So, there is a set a which has two power n elements and then F 3 which has 0,1 or 2, so each one of them could be mapped to one of these each one in the right left-hand side could be mapped to 0 or 1 or 2. So, it is 3 power A or 3 power the size of A. And now another thing that we know is that all these functions can are approximated or captured by polynomials not even approximated.

Actually, captured by polynomials of degree at most square root n by 2 + square root n by 10. Now we will just count how many such polynomials are there and we further know that these are all multi-linear as well. So, how many such monomials are there? So, how many monomials can be form of this degree and then you could add them and all that. So, we will see how that works.

**(Refer Slide Time: 26:50)**



$$= \underbrace{1 + \binom{n}{1} + \cdots \binom{n}{n/2}}_{\boxed{\le \frac{1}{2} \cdot 2^n}} + \underbrace{\binom{n}{\frac{n}{2}+1} + \cdots \binom{n}{\frac{n}{2}+\frac{\sqrt{n}}{10}}}$$

$$= \underline{\sum_{i=1}^{\sqrt{n}/10} \binom{n}{\frac{n}{2}+i}} \le \binom{n}{n/2} \cdot \frac{\sqrt{n}}{10}$$

$$= \underline{\underline{2^n}} \cdot \frac{\sqrt{n}}{10}$$

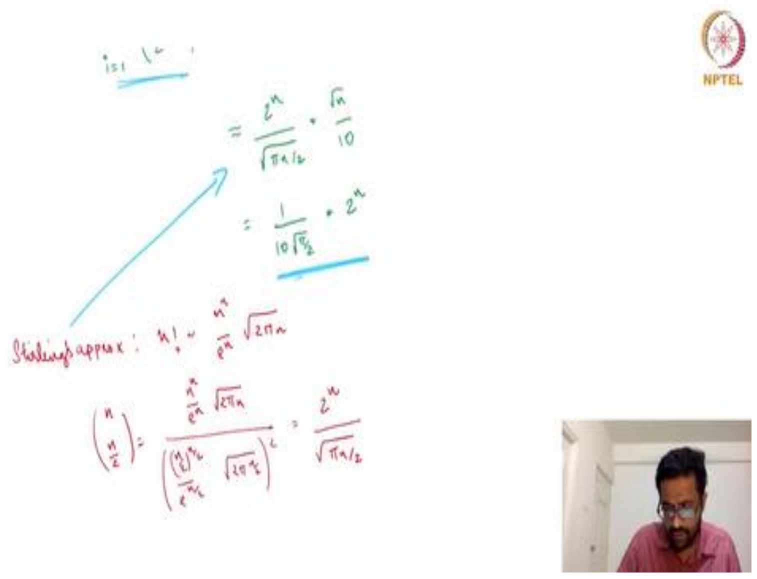
How many such monomials are there? So, the degree is at most n by 2 + square root n by 10. So, it is just the number of ways to choose, so there is one is a constant term and I am not counting the coefficients now. So, the constant term is this constant term n choose 1 is the single degree terms and choose 2 where you choose 2 x i and x j and so on up to n choose n by 2 + square root n by 10.

So, up to n choose n by 2, up to the halfway mark we know that one to halfway mark; this is the middle binomial coefficient. So, this is like at most half times 2 power n so we know that all the binomial coefficients 1 2 1 and choose 1 and choose 2 up to and choose n will give us 2 power n as a sum. So, up to the midway point it will be at most it will be equal to half times 2 power n up to the midway point.

Then what remains is n choose n by 2 +1 and n choose n by 2 + 2 and so on up to n choose n by 2+ square root n by 10. So, this first part is done this is simply half times 2 power n, second part one has to notice that there are how many terms are there are square root n by 2, square root n by 10 terms. This is the summation over here and we know that the middle binomial coefficient is the largest and what is the middle binomial coefficient, this n choose n by 2.

So, each one of them; can be upper bounded by the middle binomial coefficient times the number of terms here which is square root n by 10.

**(Refer Slide Time: 28:49)**



$i=1$

$$= \frac{2^n}{\sqrt{\pi n/2}} \cdot \frac{\sqrt{n}}{10}$$

$$= \frac{1}{10\sqrt{\pi/2}} \cdot 2^n$$

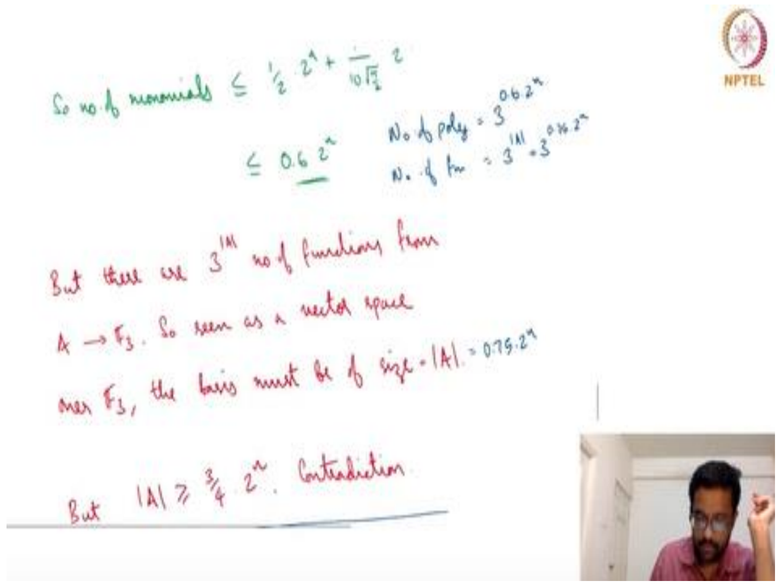
Stirling's approx: $n! \sim \frac{n^n}{e^n}\sqrt{2\pi n}$

$$\binom{n}{n/2} = \frac{\frac{n^n}{e^n}\sqrt{2\pi n}}{\left(\frac{(n/2)^{n/2}}{e^{n/2}}\sqrt{2\pi \frac{n}{2}}\right)^2} = \frac{2^n}{\sqrt{\pi n/2}}$$

And middle binomial coefficient you can actually use Stirling's approximation here so this is what we have used here this is Stirling's approximation which states that n factorial is I have written below in the red and factorial is n power n divided by e power n into square root 2 pi n. So, n choose n by 2 if you if you write n and n by 2 factor factorials in this form you will get this 2 power n divided by pi n divided by 2.

And this is what I have replaced here this is n choose n by 2. So, you can cancel the square root ends and what is left is 1 divided by 10 square root of pi by 2 multiplied by 2 power this is what we have. Now what is 1 divided by 10 square root of pi by 2? Anyway, so it is smaller than 1 by 10.

**(Refer Slide Time: 29:52)**



So no. of monomials $\leq \frac{1}{2} \cdot 2^n + \frac{1}{10\sqrt{\pi/2}} 2^n$

$\leq \underline{0.6 \; 2^n}$

No. of poly $= 3^{0.6 \cdot 2^n}$

No. of fn $= 3^{|A|} = 3^{0.75 \cdot 2^n}$

But there are $3^{|A|}$ no. of functions from $A \to \mathbb{F}_3$. So seen as a vector space over $\mathbb{F}_3$, the basis must be of size $= |A| = 0.75 \cdot 2^n$

But $|A| \geq \frac{3}{4} \cdot 2^n$. Contradiction

So, whatever square root of pi by 2 is pi by 2 itself is 1.5 1.57 and square root of that may be some something between 1 and 2. Let us say. So, this is like less than 1 by 10, so this is smaller than 1 by 10 and the first term so again I am starting from this point here the first n by two terms were at least half times to power n and then the remaining terms were 1 divided by 10 square root of pi by 2 to power n.

So, I said this entire thing is less than 0.1, the second term, so overall we get that this is at most 0.6 times to power n. This is the number of different monomials and how many ways can you how many polynomials can we generate. So, each monomial can get coefficient 0 1 or 2 so it is so it is just 3 power. So, number of polynomials is 3 power 0.6 times 2 power n. But I have already said that the number of functions from A to F 3 is actually 3 power the size of A which is 3 power 2 power n.

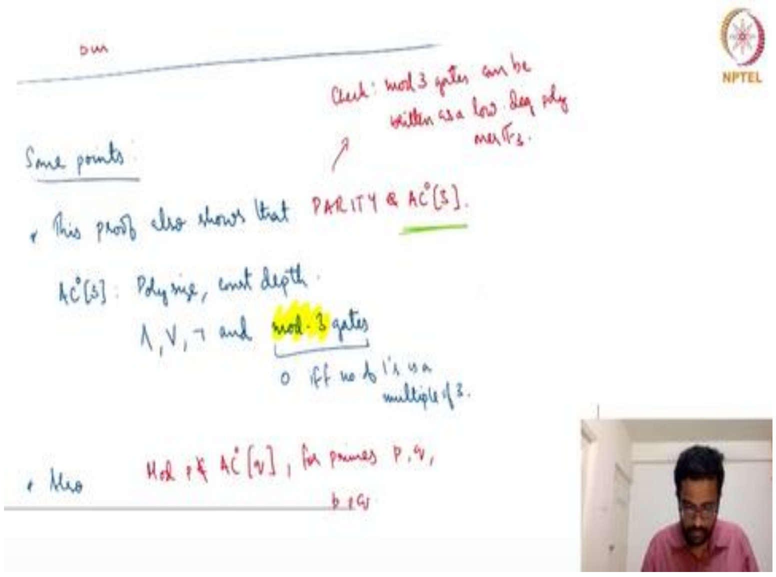
So, there you see the mismatch. Number of functions is actually 3 power the size of A which is 3 power 2 power n that gives the mismatch by whatever we have assumed so, far we got that the number of each function can be written by a polynomial, meaning each function as long as we are restricting to ourselves to A we can represent represented by a polynomial. But then the number of polynomials is smaller than the number of functions.

So, that leads to the contradiction. Another way to see it is that all these functions form a vector space and the dimension of this vector space must be of size A which is actually 2 power n but not 2 power n I am sorry even here I have to like 0.75 times 2 power n over here also 0.75 times here also I said it wrong. So, size of A is this but here we are saying the number of monomials will be 0.6 times 2 power so that is a contradiction.

So, that is it so we assume that parity could be written as a small degree polynomial or small as in n by roughly n by 2 degree polynomial and that gave us that there is a large set in which any function can be approximated by a roughly n by 2 degree polynomial and I have to repeat we assume that parity could be approximated or parity could be approximated by a square root n degree polynomial.

And that gave us that there is a roughly large set of cycle, three quarters time 2 power n. All the functions from that set can be approximated by a polynomial of degree roughly n by 2 but then that is add up because number of functions is way more than the number of polynomials.

**(Refer Slide Time: 33:48)**



Some points:

- This proof also shows that PARITY $\notin$ $AC^0[3]$.

  $AC^0[3]$: Poly size, const depth.
  $\wedge, \vee, \neg$ and $\underbrace{\text{mod-3 gates}}_{0 \text{ iff no of 1's is a multiple of 3.}}$

Check: mod 3 gates can be written as a low deg poly over $\mathbb{F}_3$.

- Also Mod $p \notin AC^0[q]$, for primes $p, q$, $p \neq q$

That gives us a contradiction and that is about it. So, we cannot have a degree square root n divided by 10 polynomial approximating parity; approximating or even computing parity. In fact, in the last lecture, we saw that epsilon approximation or one fourth approximation leads to a concrete polynomial that agrees with uh parity on 75% of the inputs and that is not possible and it completes the proof.

I think I have gone over the details carefully so I will not really go over again for a summary, but I just like to add some points some 2-3 points. In fact, this proof is actually more general. We can actually use it to show that parity is not in AC 03. What is AC 03? AC 03 is the class of, it is just like AC 0 but in addition we also have mod 3 gates. So, it is polynomial size constant depth and OR and NOT gates of unbounded fan in.

But we are also allowed to use mod 3 gates. So, what is mod 3 gates? So, what did I define parity, as parity is the; if there are odd number of ones in the input it outputs one if there are even number of ones in the input it outputs 0. Just like that if there is A if the number of ones in the input is a multiple of 3 this will output 0 if the number of ones in the input is a multiple of 3 it will output 0.

So, it is like parity what parity was for two mod 3 is for 3 and you can say mod 5 or any number. So, in this case AC 03 is what I am saying, this proof also can be used to show that parity is not in AC 03, so even if you allow the circuit to have mod 3 still parity is not computable. Why is this? So, the first half of the proof will work even in this case. So, second half of the proof does not need any changing.

The fact that parity requires a large degree does not need any changing what has to be fixed or what has to be modified slightly is that computed by a circuit that we said here in the proof that we already saw even if even if the circuit has a mod 3 gate also even then we have to approximate polynomial so we saw how to approximate AND, OR and NOT. But even mod 3 has to be approximated.

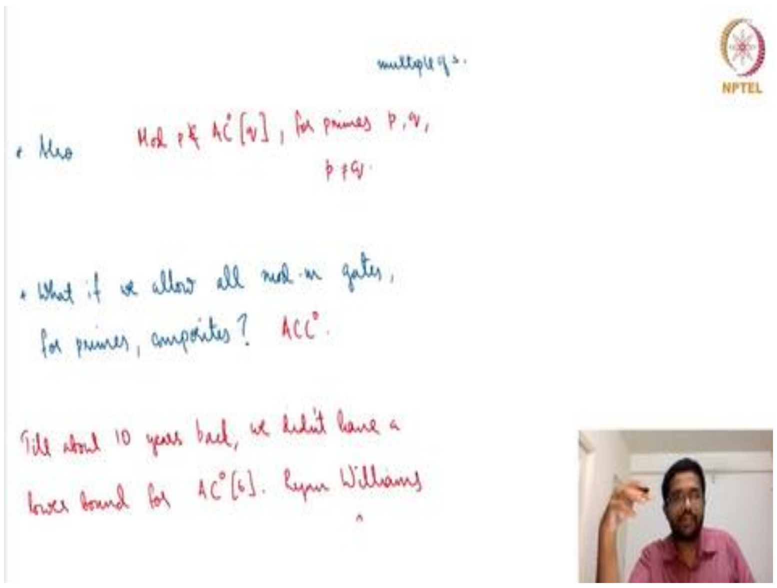
And you may it may not be that hard to convince yourself that you can also approximate mod 3 because we are already computing over F 3 so we are already the computations over F 3 so how do you compute mod 3? So, F 3 things happen already in mod 3 but now how do you compute mod 3 over F 3 that you can think. So, once that happens then you can you can generalize this or you can modify this theorem that we saw in the previous lecture.

To say that everything when we computed everything that is computed by a size s depth d a circuit of which has and now AND OR NOT and mod 3 gates can be approximated by a low degree polynomial and the second part is the same. So, this is one point, the proof is actually stronger than what we already saw so maybe I will just write here. So, check mod 3 gates can be written as a small part as a low degree polynomial over F 3.

F 3 is once again is the field 0 1 2 that contains 0 1 2 and everything happens in modulo 3. In fact, it actually shows the same technique exactly the same proof works. So, parity is like mod 2 and AC 03 is mod like 3 contains mod 3. So, now instead of mod 2 and mod 3 I could say the same for mod p and mod q. So, any mod p is not contained in AC 0 mod AC 0 q where p and q are two different primes.

So, maybe I will just say it and the thing that you can check is if we need p and q to be two different primes otherwise this to work. So, for instance if you can always write mod 3 using mod 9 gates for instance. Just think about it how you can how you will do that.

**(Refer Slide Time: 38:59)**


multiple q's.

• Also Mod p $\notin AC^0[q]$, for primes p, q, $p \neq q$.

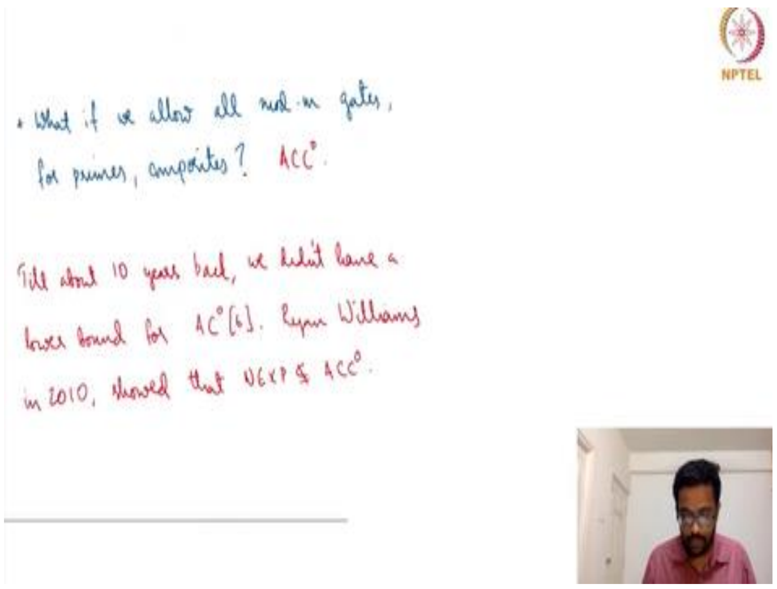
• What if we allow all mod-m gates, for primes, composites? $ACC^0$.

Till about 10 years back, we didn't have a lower bound for $AC^0[6]$. Ryan Williams

So, if they are not primes sometimes these kinds of things happen. Now this is AC 0 q. So, we have a language parity that is not in AC 0. Do we know a language that is not in AC 0 q? Yes, I just said mod p is a language that is not an ac or a function that is not in AC 0 q. What if when q is a prime? But we did not know any language or we did not even know if there is a language or a function that is not in AC 06.

So, 6 does not get captured in the previous statement over here because this the statement mod p is not in AC 0 q only held for p and q primes. So, we did not know of a language that is not there in is or even if was it a language that is not there in AC 06. So, parity and mod p are like small languages and till about 10 years back we did not know this.

**(Refer Slide Time: 40:20)**


• What if we allow all mod-m gates, for primes, composites? $ACC^0$.

Till about 10 years back, we didn't have a lower bound for $AC^0[6]$. Ryan Williams in 2010, showed that $NEXP \notin ACC^0$.

And Ryan Williams showed that there are languages so in fact it is not NP, NP exponential time nothing. So, he had to go till next to find or to show that there is a language in next that is non-deterministic exponential time that is a class next. There is a language in next that is not computable by AC 06, AC 06 means AC 0 circuit along with mod 6 gates. In fact, he proved slightly stronger statement he said that there is a language there are languages in x that is not in ACC 0.

So, what is ACC 0? It is like AC 0 polynomial size log of constant depth but we allow all modem gates not only primes composites everything, so that is ACC 0. So, till about 10 years back we did not have any such thing but now at least we have this we have to go all the way up to non-deterministic exponential time. So, NP is not non-deterministic polynomial time so we have to do exponential time and non-determinism to find a language that is there but not in ACC 0.

So, as I said in the previous lecture the progress of lower bounds in circuit complexity has been slow and I do not think there are there have been significant breakthroughs after these results. So, I think I will conclude here. I think I have already said a good enough overview of this proof of what we did in the previous lecture and what we did in this lecture so I will skip doing another summary. Thank you.