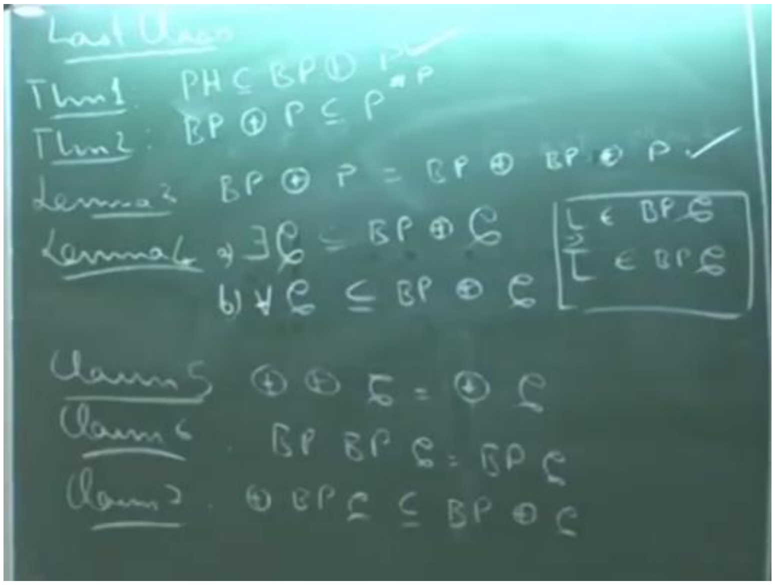
Computational Complexity Theory
Prof. Raghunath Tewari
Department of Computer Science and Engineering, IIT Kanpur
Indian Institute of Technology, Kanpur

Module No # 06
Lecture No # 26
Toda's Theorem - I

(Refer Slide Time: 00:14)



Last Class

Thm1. $PH \subseteq BP \cdot \oplus P$ ✓

Thm2: $BP \oplus P \subseteq P^{\#P}$

Lemma 3 $BP \oplus P = BP \oplus BP \oplus P$ ✓

Lemma 4 a) $\exists \mathcal{C} \subseteq BP \oplus \mathcal{C}$

b) $\forall \mathcal{C} \subseteq BP \oplus \mathcal{C}$

$L \in BP\mathcal{C} \Rightarrow \bar{L} \in BP\mathcal{C}$

Claim 5 $\oplus \oplus \mathcal{C} = \oplus \mathcal{C}$

Claim 6. $BP\, BP\, \mathcal{C} = BP\, \mathcal{C}$

Claim 7. $\oplus BP \mathcal{C} \subseteq BP \oplus \mathcal{C}$

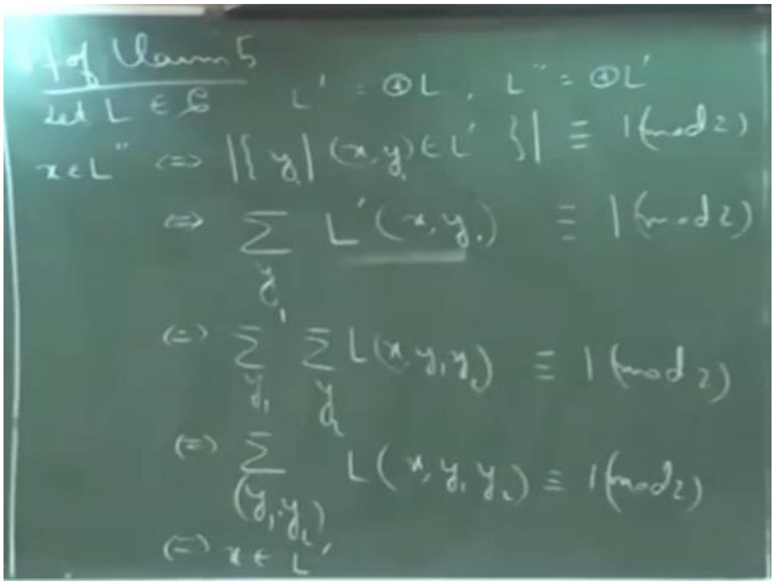
So did I give this theorem 1 or just check? What is the name that I had last time theorem 1? So I guess these are the things that I said last time. So we saw that suppose if we assume that 3 and 4 are true then that would imply theorem 1.So we saw that of course so there was a little bit of a caveat so we need to also show it for pi K. So we only showed that sigma K belong but let me also mention that.

If you even if you have a class for all C that; is also contained in BP dot parity dot C. **S**o this is something we left last time. So the reason why this holds true is because the BP operator is closed under compliment. So let me let elaborate on that so suppose if we know that this is true. So suppose there exists C is contained in BP dot parity dot C. So pick a language in for all C. So the compliment of that language is contained in there exists C. So since the compliment is contained in there exists C if I just switch the answers of the BP machine.

So the BP machine by its definition if an instance is in the language it accepts with probability greater than 1 - 1 over 2 to the power q n. And if it is not in the language it rejects with probability that much that is same. So even if I switch the answers at the end it is still a BP machine which is accepting the language L compliment. So if you have any language so for example, I mean the generalization is if you have any complexity class C.

So if L belongs to BPC so this implies that L bar will also belong to BPC and the machine is just flipping the answers. So that basically that part b is true given that part a, is given true. So we saw that this follows from 3 and 4 and we saw that Lemma 3 follows from the claims 5, 6 and 7. So you just switch you just use claim 7 to switch 2 of these operators and then apply claim 5 and 6. So let us look at the proofs of the remaining statements

(Refer Slide Time: 05:53)



Pf of Claim 5

Let $L \in \mathcal{C}$ $\quad L' = \oplus L, \quad L'' = \oplus L'$

$x \in L'' \iff |\{y_1 \mid (x, y_1) \in L'\}| \equiv 1 \pmod 2$

$\iff \sum_{y_1} L'(x, y_1) \equiv 1 \pmod 2$

$\iff \sum_{y_1} \sum_{y_2} L(x, y_1, y_2) \equiv 1 \pmod 2$

$\iff \sum_{(y_1, y_2)} L(x, y_1, y_2) \equiv 1 \pmod 2$

$\iff x \in L'$

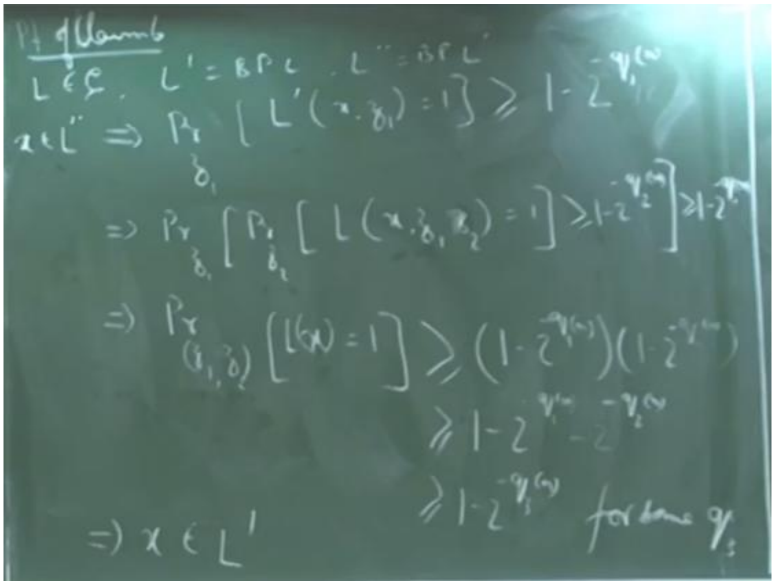
So let L be some language in C and define L prime to be parity L and define L double prime to be parity L prime. So then what we want to show is that L double prime is contained in parity in fact what we will show is that L double prime is the same as L prime. So x belongs to L double prime if and only if the number of certificates y for which the pair x y in L prime is odd congruent to 1 or 2.

This is just an, another way of writing its odd. So I will just use a another notation for this predicate in fact I will just so the way I write this is sigma so let us call this y 1. Sigma over all; strings y 1 of the predicate L prime x y. So let me just see what this means so this means that this predicate. So this predicate that I have so this evaluates to 1 if this argument belongs to the language and otherwise it evaluates to 0.

So this should be odd so now again I can expand out this predicate and write this as an x y 1 belongs to L prime if overall strings y 2 x, y 1, y 2 belongs to L congruent to 1 and 2. So now so, what this says is that every certificate y1 I have an odd number of certificates here. So basically what I am doing here is that I am taking an odd number of sums of some odd numbers. So if I have just, combine these certificates and I can just write look at them as just 1 pair. So the number of such pairs is odd.

Because I am summing a certain number of odd numbers so that will give me an odd number again. So this implies that x belongs to L prime because I have an odd number of certificates for that string x any questions? And also the proof of claim 6 is not very different in terms of idea.

(**Refer Slide Time: 10:17)**



Pf of Claim 6

$L \in \mathcal{C}$, $L' = BP L$, $L'' = BP L'$

$x \in L'' \Rightarrow \Pr_{y_1}[L'(x, y_1) = 1] \geq 1 - 2^{-q_1(n)}$

$\Rightarrow \Pr_{y_1}\left[\Pr_{y_2}[L(x, y_1, y_2) = 1] \geq 1 - 2^{-q_2(n)}\right] \geq 1 - 2^{-q_1(n)}$

$\Rightarrow \Pr_{(y_1, y_2)}[L(w) = 1] \geq (1 - 2^{-q_1(n)})(1 - 2^{-q_2(n)})$

$\geq 1 - 2^{-q_1(n)} - 2^{-q_2(n)}$

$\geq 1 - 2^{-q_3(n)}$ for some q_3

$\Rightarrow x \in L'$

So there also you combine certificates but instead of looking at the number of certificates you look at the probability. So what is happening here? So this says that the number of y 1 that is good for my string x is odd. So now instead of L prime if i look at the language L so that means that so now fix a y 1. So suppose you have a fixed y 1. So for a fixed x and a fixed y 1 the number of y 2 that is good for this x, y 1 is odd.

So you have an odd number of y 2 that is good for some y 1 you move on to the next y 1 again you again have an odd number of y 2 is that is good for that y 2. So I have a sum which is basically taken over an odd number of odd numbers. So therefore the resultant is also an odd number. I mean I can just so basically this last summation this just means that I am combining all the certificates together.

So again similarly So let be some language in C. L prime b. BP dot L and let L double prime BP dot L prime. So x belongs to L. So I cannot use maybe in can but I just look at one direction so x belongs to L if the probability over a string z that x belongs to L double prime that is what I want to show the problem no x belongs to l prime because this is acting as a certificate for x. So this pair y 1 y 2 is a certificate for x and the number of such certificates that I have is an odd number.

So this is giving me a parity machine for this language L so I started with parity machine for that and I converted it into; just 1 parity. So L prime of and this at z 1 x, z1 is one occurs with let us say some probability 1 - 2 to the power –q 1 of n. So this implies so this is true if probability over some other string z 2 L of x, z 1, z 2 = 1 happens with some probability greater than 1 - 2 to the power -q 2 of n and this is greater than 1 - 2 to the power –q 1 of n.

So now how do I combine these 2 random strings again by the same thing so think of a string as, a good string**.** So if it gives the correct answer for a given input. Suppose here I fix my x is already fixed suppose I fix a z1 are the fraction of z 2's that are good for x, z 1 is 1- I mean this fraction of z 2's are good and the fraction of z 1 that are good for x is this many. So therefore if I look at the concatenation of these 2 strings the fraction of the concatenated strings that are good for some x is how much?

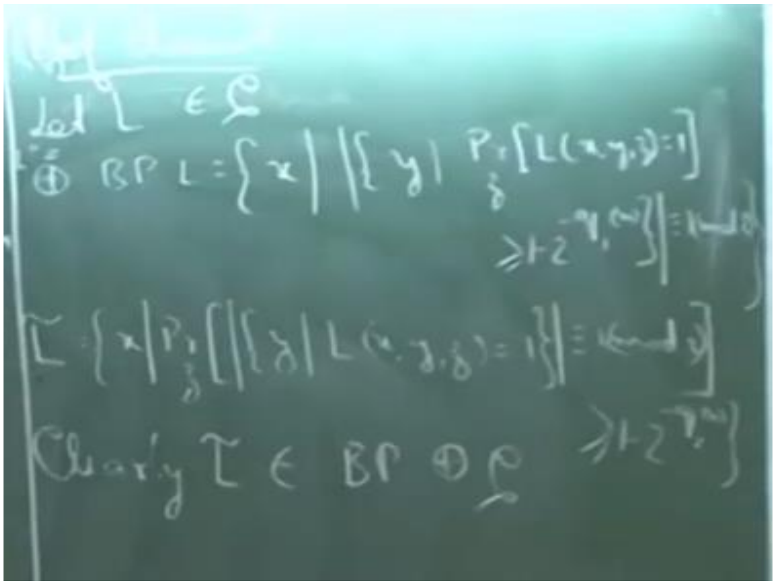
Product; of the 2 probabilities so it is greater than 1 - 2 to the power –q 1 of n. This I can write this as 1 - 2 to the power -q 1 of n - 2 to the power –q 2 of n and just ignore the last term since it is a positive term and this is greater than 1 - 2 to the power -q 3 of n for some polynomial q 3. So the essential idea is same in both these claims so let us so let us look at the last claim 7. So I left the other direction but you can complete it so it will be the same basically I mean the argument is the same.

So if x does not belong to L the fraction of z 1 that are good is less than 2 to the power of -q 1 of n then again you take the product it will be less than something so you can work that out. Which one this one or that one here so what is happening here? So here I am saying that think of this as a random string. So the fraction of random string that are good for x so that acts as a positive evidence for x is at least this much so what is that imply?

So this implies that and this number we know that it is some polynomial q 3 so if I just write down the definition of BP dot a complexity class C x was where so I had picked x from my language L double prime L was my language in C. So therefore this means that x belongs to L prime because it is a definition of BP dot l. It is a set of those strings for which I have a large fraction of random string which are positive evidences same with parity also.

They are independent I mean definitely in this case these 2 are independent events. So what are 7 is parity BPC is containing BP dot parity dot C? So here so let me start the proof.

(Refer Slide Time: 19:12)



Let $L \in \mathcal{C}$

$$\oplus BP\, L = \left\{ x \;\middle|\; \left|\left\{ y \;\middle|\; \Pr_z[L(x,y,z)=1] \geq 1-2^{-q_1(n)} \right\}\right| \equiv 1 \pmod 2 \right\}$$

$$\tilde{L} = \left\{ x \;\middle|\; \Pr_z\left[\left|\left\{ y \;\middle|\; L(x,y,z)=1 \right\}\right| \equiv 1 \pmod 2 \right] \geq 1-2^{-q_2(n)} \right\}$$

Clearly $\tilde{L} \in BP \oplus \mathcal{C}$

So let L be a language in C so then what is parity dot BP dot L is the set of all strings x such that the number of y's such that the probability over a random string z that L x y z = 1 is greater than 1 - 2 to the power minus let us say q 1 of n. So the number of y is which makes this event happen is odd. So let me just call the language l double prime may be I will need that. So what we need to show is that L double prime is contained in BP dot parity dot C and what we will show is something more, stronger actually.

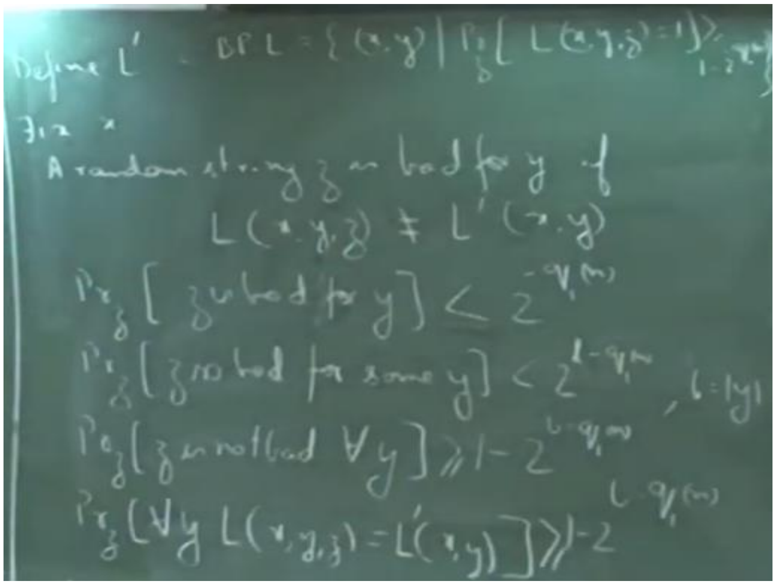
We will show that L double prime is actually equal to this language this l tilde up where L tilde is defined as set of all x is such that probability over z that the number of y is such that L x y z =

1 is odd the probability of this event happening that this number is odd is greater than 1 - 2 to the power -q 2 of n for some polynomial q 2. So we will appropriately fix our q 1 and q 2 later on but essentially we will show that these 2 languages are the same.

So I mean from the definition of L tilde it clearly follows that L tilde is in is in which class? So look at the definition of L tilde so with high probability the number of y's which makes this predicate evaluate to 1 is odd. So it is by definition in BP dot parity dot C. So L was a language in C. So let us define I will prove them equal L double prime and L tilde. So actually it is equal so when i stated that the claim last time I did not realize but actually these 2 classes are the same.

Because you can appropriately fix those polynomials so that if you take any language here it will be in here and vice versa. But so the thing is that I do not think this is true for all complexity classes C. I mean definitely it is true for the kind of classes that we are considering here but I think if you have classes which are very large. Which looks at these kinds of certificates which are exponentially long probably there it will not work. So when we look at the proof we will realize that what kind of classes does this work for.

(Refer Slide Time: 24:22)



Define L' = BP L $= \{(x,y) \mid \Pr_z[L(x,y,z)=1] \geq 1-2^{-q_1(n)}\}$

Fix x

A random string z is bad for y if

$L(x,y,z) \neq L'(x,y)$

$\Pr_z[z \text{ is bad for } y] < 2^{-q_1(n)}$

$\Pr_z[z \text{ is bad for some } y] < 2^{l-q_1(n)}$, $l=|y|$

$\Pr_z[z \text{ is not bad } \forall y] \geq 1-2^{l-q_1(n)}$

$\Pr_z[\forall y\ L(x,y,z)=L'(x,y)] \geq 1-2^{l-q_1(n)}$

So define L prime based on L as the class BP dot L. So BP dot L is set of all I can think of them as tuples x, y such that the probability that L x y z is 1 is greater than 1 - 2 to the power -q 1 of n. So let us look at a fixed x so let us fix some string x. So we see that a random string z is bad for a string y if L x y z is not the same as L prime x y. So we have a fixed x on so look at all the random strings z a we try to see that when these 2 events.

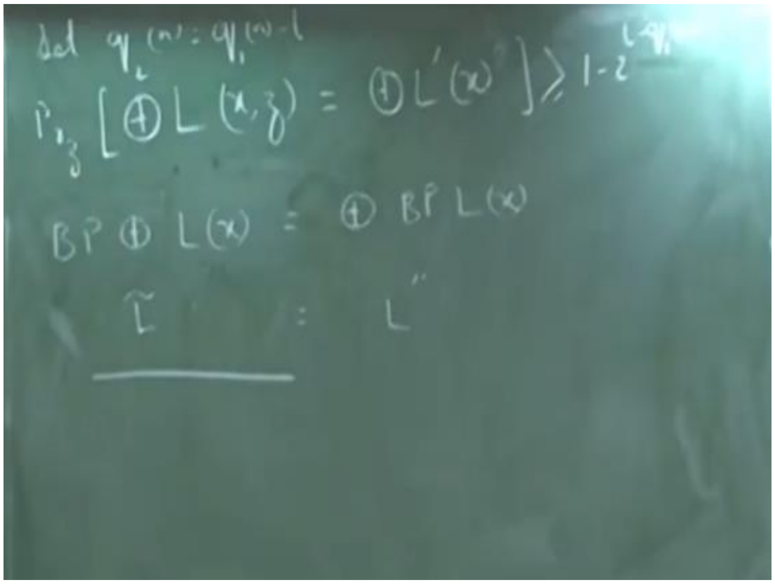
I mean if we think of them as predicates when these 2 predicates do not give the same value. So if for some random string they give different value I call that random string a bad random string. So what we know from a definition is that most random strings are good for most random string it does give me the correct value. I mean it follows from the definition that a huge fraction is good but there can be some bad strings as well.

So the probability is what I said right now that if i fix a y z is bad for y is how much? What is the probability that for a fixed y my random string z is bad is less than 2 to the power -q 1 of n. So the probability that z is bad for some y is how much? So let us see we have let us see we know that our y has length L. So the total number of such strings is 2 to the power L. So the number of z is which are bad for some y is 2 to the power L times –q 1 of n where L is the length of my y.

So the probability that z is not bad for all strings y is basically the compliment of this event. So this is greater than 1 - 2 to the power l - q 1 n. So again I can just write this as the probability that for all strings y z is not bad so which means that these 2 x y z = L prime x y. So I have the freedom of choosing what I want my q 2 to be? So I will set q 2 to be what I want is that? This probability should be this entire probability should be greater than 1 -2 to the power –q 2 of n.

So -q 2 of n = l- q 1 of n will this work? Anyway let us just not fix what our q 2 is we will probably fix it later on.

(Refer Slide Time: 30:26)


def $q_2(n) = q_1(n) - l$
$\Pr_z [\oplus L(x,z) = \oplus L'(x)] \geq 1-2^{l-q_1}$
$BP \oplus L(x) = \oplus BP L(x)$
$\tilde{L} = L''$

So I will just leave this blank for the time being but let us see what we want correct? So the probability over a random string z that for all y's these 2 predicates gives the same answer so much. So if it happens for all wise I can write this as probability over all Z parity of L x z is equal to the parity of L prime x. So what we have here is that for all y's these 2 predicates are the same so more particularly the parities also should be the same.

So what do we have here so what do we have here is that for most z that is with probability greater than 1 over 2 to the power L - qn. So I had fixed an x so this predicate is equal to this predicate which means that. So what does this mean? BP dot parity dot L of a string x so I just subsumed the probability in this operator. So this is equal to parity dot L prime but L prime is nothing but I think I erased it is t no.

So l prime is there BP dot l and this is what our l tilde was and this was our l double prime. So we showed that this happens for any fixed x in general i mean this happens for all x. So to put this BP operator what I need is that this should be greater than whatever polynomial I want I mean initially I wanted this to be -q 2 of n so q 2 of n = q 1 of n - l. So that will work because see this predicate does not depend on the random string.

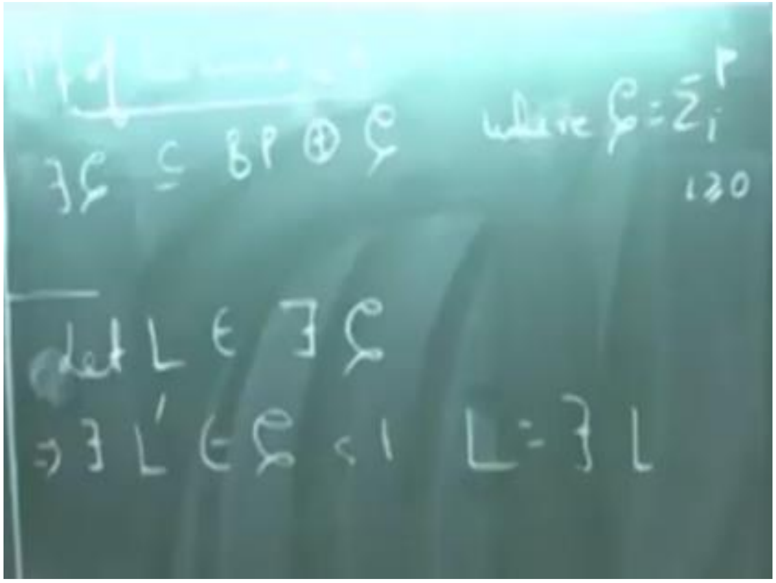
So this predicate is true for all x's I mean I just started off with an x and this predicate is true for that x. So the reason why I said that this might not be true for all classes C is because here I am assuming that my y has length l which is some polynomial for this expression to make sense this should be a polynomial. So if L is too large so let us say if it is some exponentially large number. I mean I do not know whether things will work out fine but as far as we are concerned we are looking at the polynomial hierarchy.

So all certificates have polynomial length so; life is fine **y**. So what have we seen so we have seen these 3 claims so what remains of theorem 1 is Lemma 4 yes because so let me explain so what does this mean. So I started off with an X. So what this means is that may be this is not a correct notation. I should not use I mean I wanted to use x because I wanted to show this for a fixed string.

But i think together with I should also use the random string that is generated by the BP and the y that is generated by the parity operator. So what this essentially means is that with high probability. So if you look at all random strings for most random strings the number of y that will act as evidence that x belongs to the language is odd. So notationally I mean erase this and rephrase this in English but I mean as long as the idea is clear I think it should be fine.

This one I mean this follows from the definition so l prime was BP dot l. So I am just adding the parity operator to it. So for an odd number of y's that happens what this meant here was that? For an odd number of y's x belong to l prime. Such that x, z belongs to l and x belongs to l prime if there is some z. I mean if there is a large fraction of z which acts as an evidence. So think about it so there is nothing very intricate happening here but think about it.

(Refer Slide Time: 37:27)

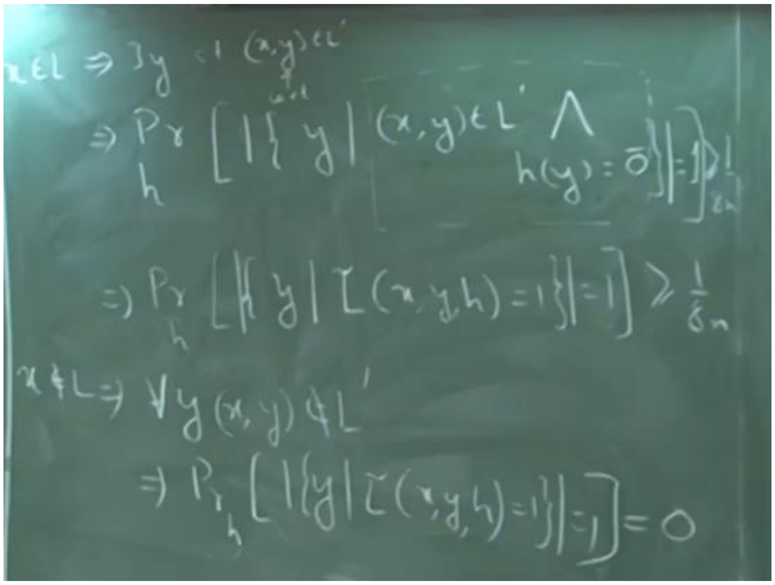


So again here what we will assume is that c is not that any complexity class but some class in the polynomial hierarchy. So what we will show is that? There exists C is contained in BP dot parity dot C where C = sigma I greater than 0. So for any greater than 0 will show that it is true and that will be sufficient. So this proof is quite easy I mean once we assume that we know how Valiant Vazirani works?

So what this means is that so let l be a language in there; exists C. So this implies that there exists some l in C such that there exist. So there is some l prime in c such that l is equal to there exists dot l. L should be a polynomial so that is why I said that I mean we should be fine as long as we look at classes where the certificates are of polynomial length. And since we are dealing within the polynomial hierarchy I mean everything is fine.

But I may be it also works for higher classes I am not quite sure but I doubt it. This proof will not work may be something else will work.

(Refer Slide Time: 39:32)



$x \in L \Rightarrow \exists y$ s.t. $(x,y) \in L'$

$$\Rightarrow \Pr_h \left[\left| \{ y \mid (x,y) \in L' \wedge h(y) = \bar{0} \} \right| = 1 \right] \geq \frac{1}{8n}$$

$$\Rightarrow \Pr_h \left[\left| \{ y \mid \tilde{L}(x,y,h) = 1 \} \right| = 1 \right] \geq \frac{1}{8n}$$

$x \notin L \Rightarrow \forall y \ (x,y) \notin L'$

$$\Rightarrow \Pr_h \left[\left| \{ y \mid \tilde{L}(x,y,h) = 1 \} \right| = 1 \right] = 0$$

So x is in l implies that there exists some string y such that x, y is an l bar is the definition. So now if I apply Variant Vazirani to this what i can say is that this is true if I mean if this is true then with high probability if I pick an appropriate hash function. So here my y is certificate so think of it as the satisfying assignment of the formula x. So with high probability if I pick a hash function h then the probability that x, y that the number of certificates y.

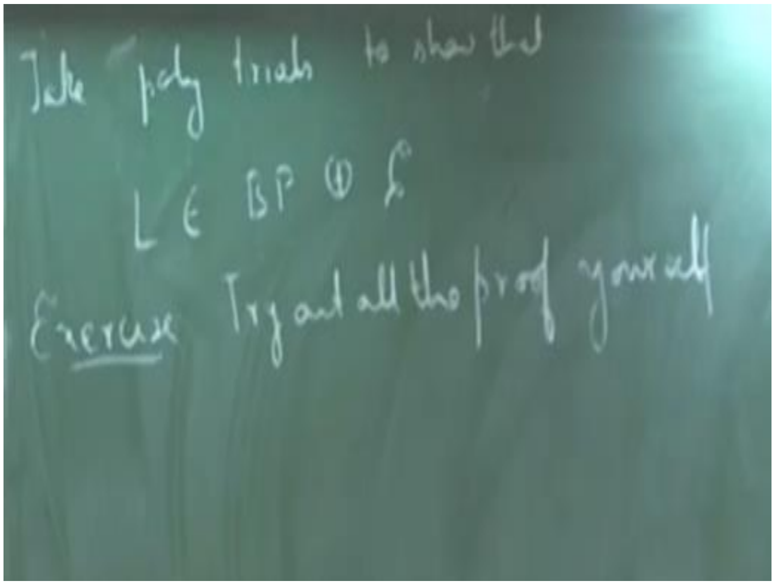
Such that x, y belongs to l prime and h of y gives me the all 0's vector is 1 with probability greater than 1 over 8 n. So again we can think of that set T I mean if you look at the main theorem that we proved for variant vazirani we can always look at the abstract set of certificates. Which has size between 2 to the power k and 2 to the power k + 1 then this holds. So all we need to show is that this entire computation can again be done by an algorithm which is C.

So of course the first part can be done by an algorithm which is in the class c the claim is that the second part also can be done by an algorithm which is in c. Because what this is essentially doing is this is taking a hash function which has some very nice form. It is not some obituary function and it is just evaluating a string on that function and checking if that equals the all 0's. So I can just write thus as the probability over h of some l the number of y such that l tilde of x, y, h evaluates to 1 with probability 1 over 8 n.

So now again so this is if x belongs to l and if x does not belong to l what we know is that? For all y x, y does not belong to l prime. So with if I again take the same machine l tilde this will evaluate to 0 on all such pairs because the moment I am taking this and for all wise this does not belong to L prime so this predicates always evaluates to false. So therefore the probability that the number of y is such that L tilde x y h is 1 is 0.

Now again we can since this is a one sided error machine I can take some polynomially many trails and amplify our success probability, so that this becomes a BP machine.

(Refer Slide Time: 44:36)



Take poly trials to show that

$L \in BP \oplus \mathcal{C}$

Exercise Try out all the proof yourself

So take just this some polynomially trails to show that l is in BP dot parity dot c. So once again i have stated this as just 1 solution but you can interpret this as a an odd number of solution because. Of course if I take many trails this property will not be preserved but what we know will be preserved is the parity of the number of certificates. So if this has an odd number of certificates I can always construct a proper language an appropriate language which will belong to parity dot C.

So again what I advise is that as an exercise for today so you try out all the proofs yourself. So they are not difficult proofs but once you try them out it will make you more confident. So I will stop here today. So what we have is we have Lemma 4 as well. So what remains to be seen now is y BP dot parity dot p is in p to the sharp p. So that will prove Toda’s theorem.