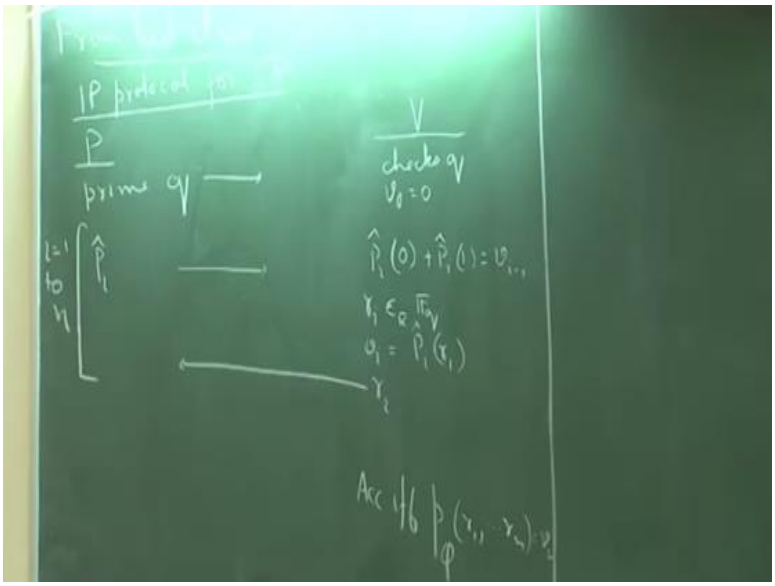
Computational Complexity Theory
Prof. Raghunath Tewari
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

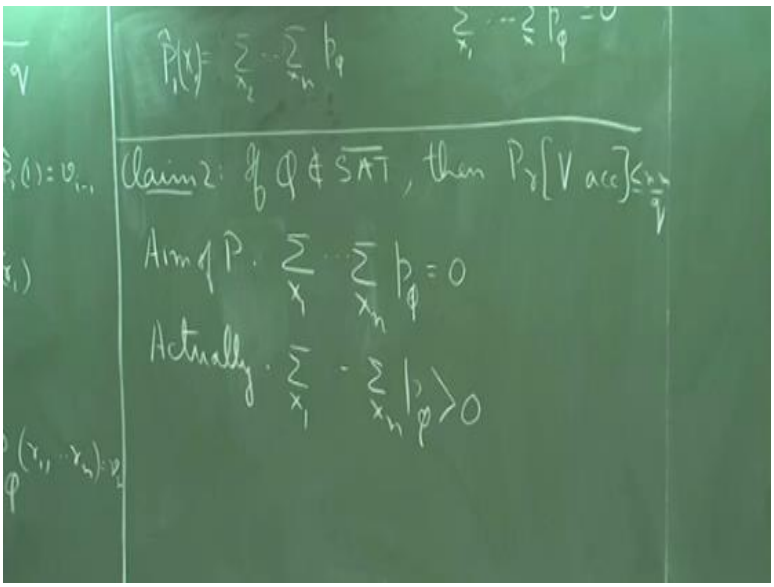
Lecture -35
Introduction

(Refer Slide Time: 00:14)



So, this was the protocol that we had last time so let me wait for one more minute.

(Refer Slide Time: 03:06)



And we also claimed last time that if phi is not satisfiable then V always accepts. And the second claim was that which we will show today is if phi is satisfiable then the probability that V accepts is quite small so its smaller than nm by q. So, some of you had this question last time that why is this final check necessary. I mean in other words as I said that what is the strategy of the prover.

So, suppose phi is not satisfiable what is the strategy of the prover which polynomial will be sent at lets say the first step. So, at the first step basically let me just so I will just it is claim one for the time A claim 2 for the time B. The strategy of the prover is that I mean if phi is not satisfiable is at the first step so he will send the polynomial sigma x 2 to x n of p phi. And this is a polynomial in x 1.

And since this one sums up to 0 so this one together with the sigma operator on x 1 sums to 0 it will the verifier will accept in the first round. And then he will set he will pick some random point in the field and plug that in for x 1 and ask the prover to verify again in the second round that it sums up to V 1. So, the prover will again be able to do it. So, the strategy of the prover in the case when phi is satisfiable his best strategy is to always basically send this polynomial with the required number of operators.

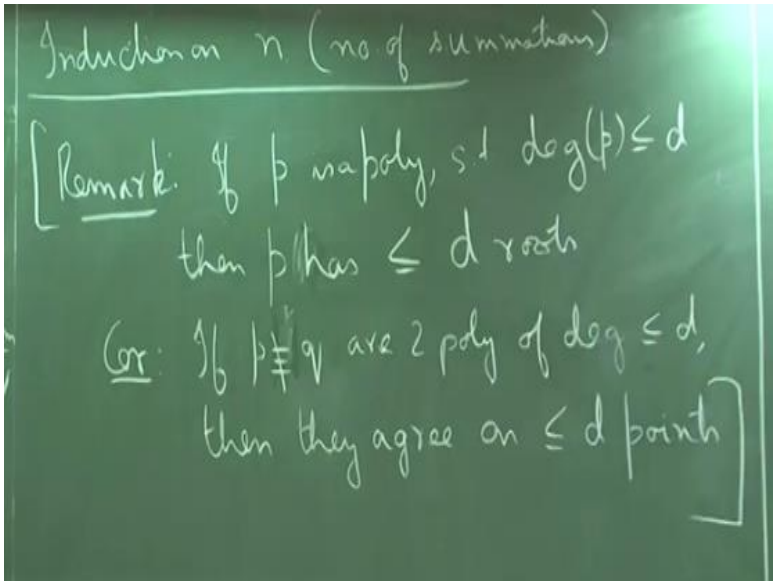
So, in this case this step is kind of redundant I mean basically by the definition I mean by the definition of what the prover is sending, it will happen that finally this will be equal to V n. But the point is that I mean nobody has restricted the prover to send exactly this polynomial at every step. In other words if you look at the other side so suppose if phi satisfiable so then also the prover wants to convince the verifier that phi is unsatisfiable.

So, that is what he wants, in other words he wants to show that the summation sigma x 1 through x n p phi is equal to 0. But now if we just ends up sending this polynomial at each step I mean the verifier will reject in the first round itself. Because what do we have in the first round that sigma so this will be greater than 0 basically. So, the verifier will reject and it will not even go to the second round.

So, the prover will try to cheat I mean he will try to send some polynomial let us say p 1 prime which will make the verifier which will fool the verifier and it will take the thing to the next iteration. So, that is what we have to compute that what is the probability with which the prover will succeed in fooling the verifier for all the n iterations. So, if phi is satisfiable then we claim that the probability that the verifier ends up accepting is at most nm by q.

So, again let me emphasize this I mean it will not harm us so the aim of the prover is to show that sigma x 1 through sigma x n of this polynomial p phi is equal to n. But what we have in the case when phi is satisfiable what we actually have is that this sum is strictly greater than 0. So, somewhere the we expect the prover to falter.

(Refer Slide Time: 09:18)



So, let us look at the proof so the proof will be an inductive argument and will perform our induction on basically this n, so the number of summations that we have. So, the number of summations and before going into the proof let me just make a quick remark which is the in fact the crux of this proof is the very in fact that is the only mathematical tool that we require is, so suppose if we have a polynomial.

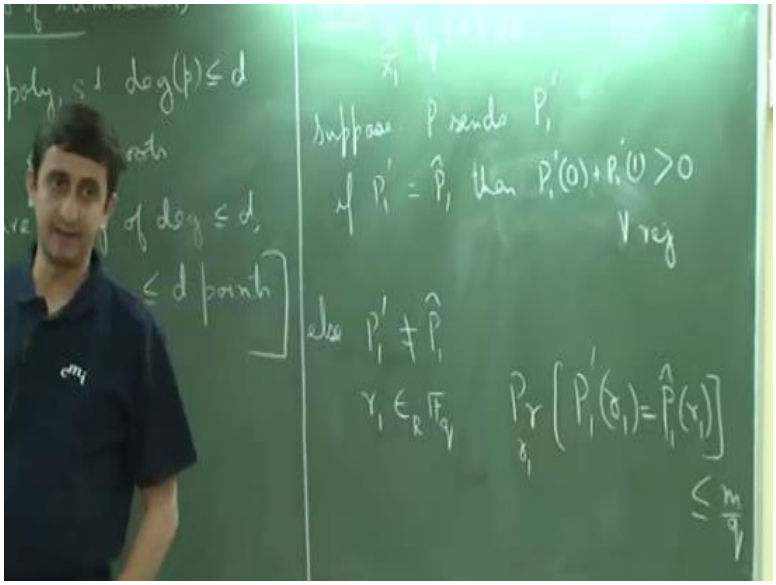
So, if p is a polynomial such that degree of p is less than let us say some quantity d then p has at most d roots. So, this is something which is known to us I mean the mathematical proof is not so obvious. But with little effort it can be shown and so note that we are not working in the field of

reals. Here we are working in the field of F q but even in F q this property holds true. And a corollary of this fact is that if p and q are two polynomials of degree at most d.

Then they agree on at most d points because if we take two polynomials p and q and if you consider their difference so if p and q are two degree d polynomials the difference is also a degree d polynomial and that polynomial again by our remark has d roots. And those are exactly the d points on which these polynomials will be equal. So, p and q are two non identical degree d polynomials.

Because if we take two identical polynomials then they agree basically on all points. So, it does not make sense so I can just they are non identical. So, let us come back to the proof of the claim so we induct on n.

(Refer Slide Time: 12:40)



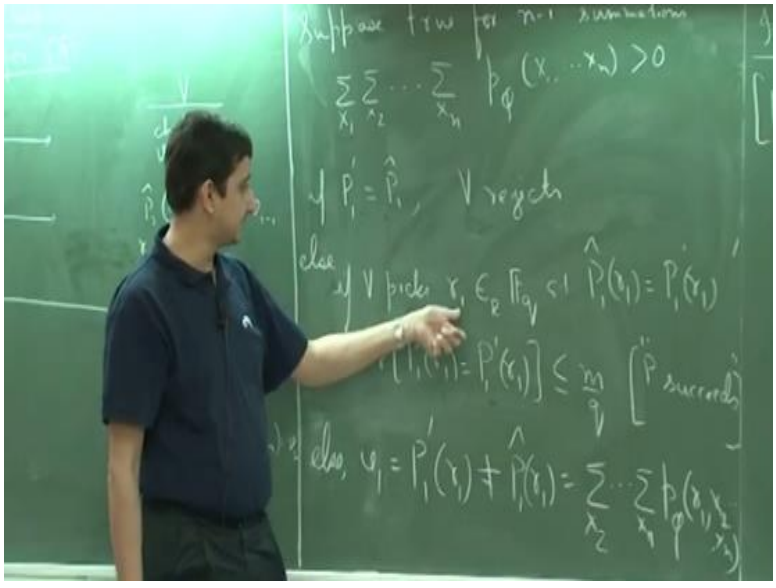
So, suppose if n is equal to 1, so what we have is we have that sigma x 1 so this is greater than 0. So, now the prover sends the polynomial P 1 hat, no so let us write it this way. So, suppose P sends a polynomial P 1 prime and so now there can be two cases. So, if P 1 prime is indeed equal to P 1 hat then what happens? So, the summation is greater than 0. So, then the verifier will immediately reject.

So, if this is the case then P 1 prime 0 + P 1 prime 1 is greater than 0 by this observation and therefore we will reject. But now suppose the prover decides to cheat the verifier and he sends a some other polynomial. So, suppose the polynomial that is sense is not equal to P 1 hat so what happens in this case. If he sends a different polynomial then it may be the case that P 1 hat 0 + p 1 hat 1 is equal to 0.

So, we do not know I mean there are there can be many polynomials which satisfy that inequality. But then what happens so now the verifier he picks a random point r 1 in the field I mean in this case it is r 1 he evaluates P 1 prime at r 1 and he goes on to check whether P 1 hat of r 1 is equal to v 1 or not. So, in other words so now for a random point picked by the verifier in this field what is the probability that these two polynomials are identical on that point.

So, this is less than or equal to m by q. Because there are both the polynomials have degree at most m so there are at most m points on which they agree upon and the total number of possible choices is q. So, even if the polynomials are not equal the probability by which the verifier will be full is at most m by q. So, in this step the verifier ends up accepting with probability at most m by q. So, is this clear to everybody?

(Refer Slide Time: 16:16)



So, suppose what we are claiming is true for n - 1 summations. So, now let us look at a case when so what we have is sigma x 1 sigma x 2 sigma x n so, we have n summation and the prover

wants to and what we have is P phi of x 1 through x n is greater than 0. So, what happens in the first round so again in the first round if the polynomial sent by the prover let us say P 1 prime. So, if that is equal to P 1 hat so then we immediately rejects.

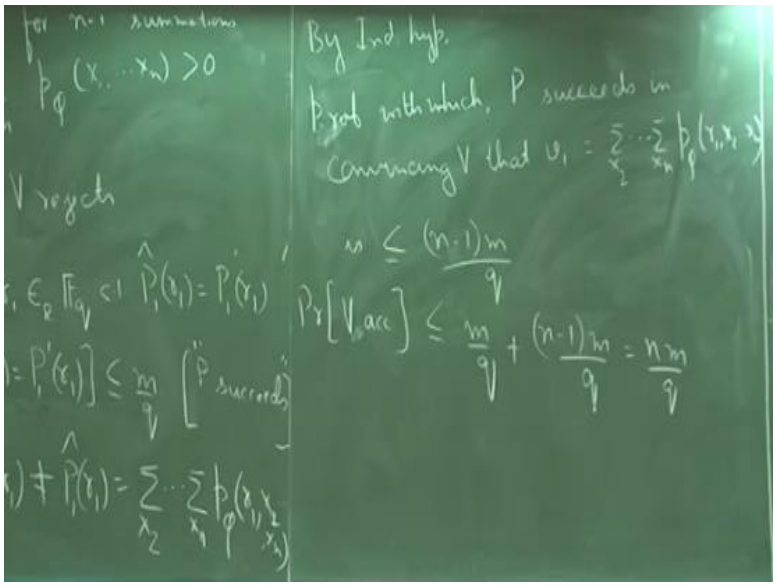
Again by the same thing because we are summing up x 1 and I mean he will find that summation does not equal 0. Else so these polynomials are not the same so V picks r 1 in F q and again the probability that P 1 hat r 1 is equal to P 1 prime r 1 is at most m by q. And so suppose if this happens suppose if V ends up picking an r 1 for which these 2 polynomials are the same we will just say that if that happens then P succeeds.

So, P succeeds in fooling the verifier. On the other hand suppose it picks if the verifier picks an r 1 for which these two polynomials are not the same. So, else this was one case we picks r 1 such that P 1 hat r 1 is equal to P 1 prime r 1, so that probability occurs. So, the probability of that occurring is m by q. Else the V 1 that the verifier computes which is nothing but P 1 prime at r 1. So, this is not the same as P 1 hat r 1 which by definition is equal to x 2 up to x n P phi.

The first point it is evaluated at r 1 and the remaining are the bound variables sorry this will be from x 2. So, let us again go through this let us look at case by case. If the polynomial that the prover picks is equal to P 1 hat then the verifier immediately rejects. If it picks a polynomial that is not equal but the verifier picks a point on which the polynomials agree then we will just claim that if that happens then the prover succeeds and the probability of that happening is so much.

And the third case is that if the verifier picks a point on which the two polynomials do not agree then we have a V 1 that is not equal to this summation.

(Refer Slide Time: 21:33)



So, now we can use our induction hypothesis that now we have an n - 1 summation. So, by our induction hypothesis probability with which the prover succeeds in convincing V that this false claim is true that is V 1 is equal to sigma x 2 up to x n, P phi of r 1 x 2 to x n is less than or equal to n - 1 m by q. Because that is what our induction hypothesis is so we want to show that so I had erased claim 2 but the probability with which V accepts is nm by q.

So, our hypothesis is this is at most n - 1 m by q. But so now what is the total probability that V accepts? So, the probability that V accepts is either he picks a point on which the polynomials agree which is m by q or he picks a point on which they do not agree in which in the successive rounds somewhere the prover will be able to I mean somewhere the verifier will be able to detect this error with probability at most or at least 1 minus that.

So, the probability that V accepts is n - 1 m by q. So, this is n m by q as we want. Because these are the two things that can happen. So, either the V picks a point on which these two polynomials agree upon. If they agree upon that point, so then we say that the prover actually succeeds. So, it will go on to the next level anywhere and the probability of this happening is at most m by q. On the other hand if he picks a point on which they do not agree then the prover is left with an incorrect claim to prove in the successive iterations.

So, then the prover has to prove the following thing that V 1 which is actually not equal to this. So, he has to prove this incorrectly. And that by our induction hypothesis is at most n - 1 m times q. So, it is so nothing I mean very deep happening here it is simple induction. So, what is our induction hypothesis? So our induction hypothesis, so let me start from there. So, our induction hypothesis is that;

Suppose the prover wants to prove an incorrect term, so here we have n summations and we want to and what we have is that this summation is greater than 0. And the goal of the prover is to show that this is actually equal to 0 and we assume that if I mean the prover succeeds with probability at most nm by q. So, that is our induction hypothesis. So, if we have n such summations the prover succeeds in showing that P phi x 1 through x n is equal to 0 with probability at most n m by q if you have n summation.

So, in other words if we just instead of keeping n as our induction variable you can think of that as k also. So, now what is happening here? So, suppose if these two polynomials are the same then it is easy. If these two polynomials are not the same then we know that these two polynomials can agree on at most m points. So, if the verifier picks a point on which they agree so that can happen with probability m by q.

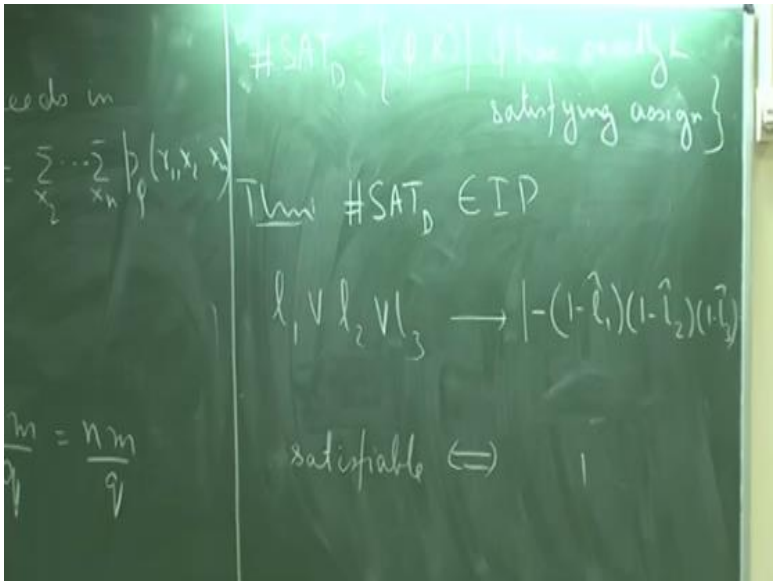
And here we claim that the prover has succeeded at least in this iteration. So, the thing moves on to the next iteration. So, again in that iteration the prover will send some polynomial P 2, some polynomial P 2 prime the verifier will pick a random point r 2 and the prover has to convince the verifier that those two things are the same. So, again that can happen with m by q the next round. But the other case is that if the verifier picks an r 1 on which these two polynomials do not agree;

Then what we have is that again what we claimed in our induction hypothesis that he is left with an incorrect claim to proof in the remaining n - 1 iterations that is he has to now show that v 1 is equal to this quantity but actually that is not the case. I mean actually from the way v 1 is defined and the way P 1 hat is defined these two quantities are not the same. But the verifier cannot check that.

I mean the only thing the verifier checks is whether P 1 prime 0 + P 1 prime 1 is equal to V 0. And then he picks an r 1 he computes V 1 and he sends V 1 to the prover. Now the prover has to come up with a P 2 prime such that P 2 prime 0 + P 2 prime 1 is equal to v 1 and so on. But now the prover himself knows that no matter what he does that is not the case. So, his best bet is to somehow cheat the verifier.

And by our induction hypothesis he can do that with probability at most n - 1 m by q. So, these two events are exclusive events so the probability of them happening is at most this plus this. So, as I said I mean the only mathematical tool or the only deep mathematical observation that is being used in this theorem is that a polynomial with d degree cannot have more than d roots. So, th is what they use. So, now it is very easy to extend this argument to sharp P. So, what we have shown is SAT bar is in IP.

(Refer Slide Time: 29:39)

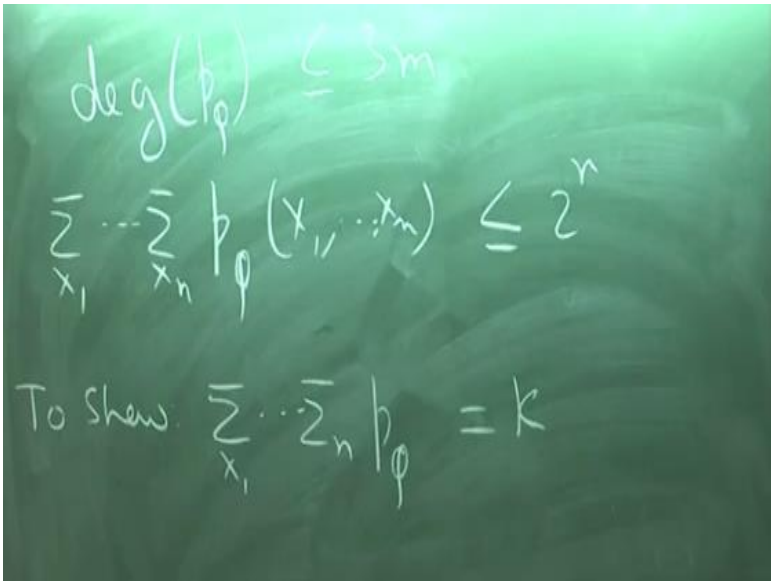


But we can easily extend this and show the following. So, let me define a language before stating the theorem. So, let sharp SAT D, so D stands for decision is the following decision problem. So, we are given a 3 CNF Boolean formula phi and some number k in binary such that phi has exactly k satisfying assignments. So, this is just the decision version of the sharp SAT function. So, what we can show is that this language is in IP again and the proof is exactly the same but with just some trivial modifications.

So, recall how did we construct this polynomial P phi in the case of SAT bar. The polynomial that we constructed was for we replaced every positive literal with variable x i and a negated literal with 1 - x i. If we had an or if we had or of 2 literals we just took their sum and we for and we took the product. So, here what we do is suppose if we have a clause which is an or of 3 literals l 1, l 2, l 3. So, we replace it with the polynomial 1 - 1 - l 1 hat, 1 - l 2 hat 1 - l 3 hat.

And what this gives us is that if this is satisfiable then this evaluates to 1 and if this is not satisfiable then this will evaluate to 0 from the definition. So, what is the so what do we have so there are some small changes now. So, the first change is that every I mean the polynomial corresponding to a clause now has degree three so earlier it had only degree one so the total degree of P phi was just m.

(Refer Slide Time: 32:54)

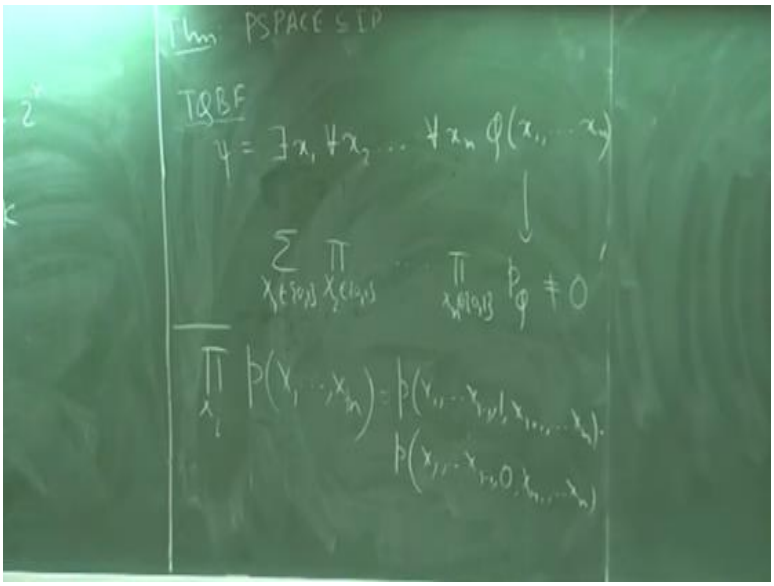


But now degree of P phi can be as large as 3 m. But this summation x 1 through x n how large can the value of this summation be now. So, earlier we had claimed that it would be as large as 2 to the power n times 3 to the power m. So, that will be exactly the number of satisfying assignments and more specifically that will be bounded by 2 to the power n or more generally I should say.

So, now basically we can choose our prime to be some number that is greater than 2 to the power n. So, now what we have to show is that this summation sigma x 1 through x n of P phi is equal

to k. So, this is what we have to show and we and the thing proceeds exactly in the same manner. So, now let us look at our main objective, so the main objective with which we started this entire program was to show that the class PSPACE is contained in IP.

(Refer Slide Time: 34:38)



So, what we have shown so far is that a class which is as large as sharp P is contained in IP because of this observation. But PSPACE is even bigger than sharp P, so how do we show this. So, again you can argue that so let us look at PSPACE complete problem so we can look at TQBF. So, this language TQBF is basically quantified Boolean formula, let us call it psi where every variable is quantified.

So, let us say that the variable x 1 is quantified with a there exist quantifier x 2 with a for all and finally x n is quantified with a for all quantifier and we have some unquantified 3 CNF formula at the end. So, I am just assuming n to be even, does not matter. So, this problem is complete for PSPACE so this we had at least claimed at one of our earlier lectures. So, how do we show that there is some IP protocol for this problem.

So, TQBF stands for true quantified Boolean formulas, so it is set of all Boolean formulas phi such that for some assignment to the variable x 1 and for all possible assignments to x 2 and so on up till x n this will evaluate to true. So, it is basically the same as SAT, but in SAT every

variable is unquantified. So, basically for some assignment to x 1 and some assignment to x 2 and so on the formula should evaluate to true.

But here certain variables are also quantified with a for all quantifier that is no matter whether you choose 0 or 1, it should evaluate to true. So, what is the I mean, what would the strategy be in this case? So, here again we can construct our polynomial as follows. I mean what we can do is we can construct so from phi we construct let us say our polynomial P phi. And now for the for all quantifiers I consider the product with I mean over x 1 coming from 0, 1.

And for there exist quantifiers I can again look at look at it as x 1 being summed over 0, 1 and then our goal is to show that this is basically not equal to 0. Because we want all true formulas. So, this is the algebraic version of the problem. So, earlier what we had earlier in the case of SAT bar what we had was that all the variables were getting quantified with a summation operator. But here because of the for all certain variables are getting quantified with the product operator.

So, till this point there is no problem I mean we can do this. But now there comes a problem so what happens if we do this? So, where will the degree blow up? So, how exactly so what was the protocol for let us say SAT bar. So, basically the prover sent a polynomial P 1 let us say p 1 prime and the verifier needed to check whether p 1 prime 0 + p 1 prime 1 was equal to some quantity so, in that case is 0.

And then send back a random point on which again it has it had to convince. So, in this case what the verifier can do is it can basically compute the product I mean it can compute P 1 prime 0 times P 1 prime 1. So, that can be done but so as you said the degree will blow up. But the degree blows up but the degree of which polynomial so what blows up in this case. So, precisely so basically the polynomial that the verify I mean the polynomial that the prover is sending.

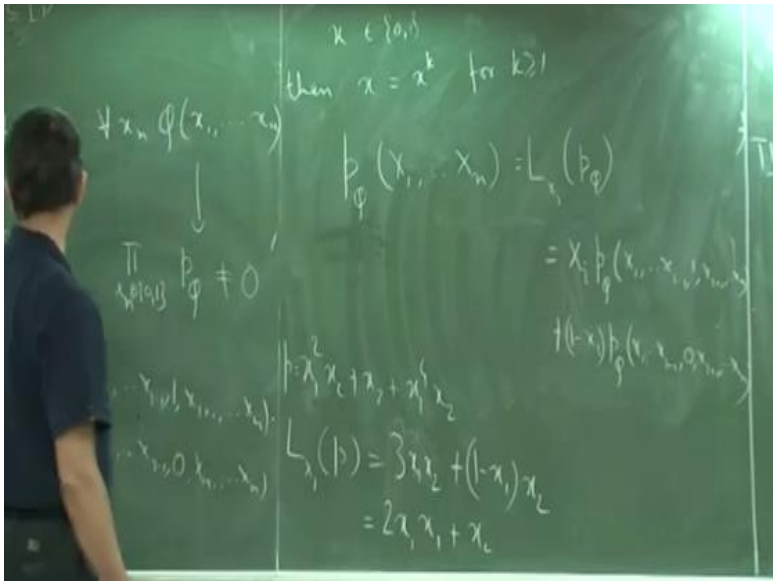
So, that because see what is happening here. So, here when we are summing up over a variable then we have no problem. But when we are taking the product over this variable we are basically doing something like this. So, suppose if we do a product of some x i of a polynomial P on x 1

up to x 1 through x n. So, what does this mean? So, this is equal to P of x 1 through x i – 1, 1, x i + 1 to x n times P of x 1 through x i – 1, 0, x i + 1 through x n.

So, if P had let us say degree d the product polynomial will now can have degree at most 2 d. So, in other words if we take this product too many times if we take this product for example order n times the thing can go up to 2 to the power n. So, the polynomial that the prover needs to send on which the verifier will get convinced can have degree as large as 2 to the power n. So, this is why and now since the verifier is just a machine which has a polynomial time restriction.

It cannot evaluate anything that is that large. So, what is the idea? So, the idea again is very simple. So, as I said that all these work kind of happened in a very short span of time and this result showing that P space is contained in i P was done by Shamir.

(Refer Slide Time: 42:51)



So, what he observed is that suppose if we have a variable x which belongs to 0, 1 then x is always equal to x to the power k for any k. So, he made this simple observation that if x is 0 then x to the power k is equal to 0 and if x is 1 then x to the power k is again equal to 1. Of course this does not happen for other values, but it is sufficient for us. Because we are only dealing with inputs from 0 and 1.

So, then what is said that so with this observation what we can do is although the polynomials degree keeps on growing at each stage, we can using this observation, we can reduce it and that will not have any effect on the value of the polynomial. So, in other words so suppose if we have a polynomial P phi over x 1 through x n then the value of this polynomial will be equal to let me call it L x i of P phi which is defined as x i P phi of x 1 up to x i – 1, 1, x i + 1 up to x n + 1 - x i up to x n.

So, suppose if we have some polynomial on which the variable x i has degree that is greater than 1. Suppose it has degree 2, so let us take a small example. So, suppose we take a polynomial x 1 square x 2 + x 2 let us say x 1 to the power 4 x 2 and we apply this operator on this polynomial. So, what would L x 1 of P, so first we substitute 1 for x 1. So, if we substitute 1 for x 1 we have x 2 + x 2.

So, we have 3 x 2 and now if we substitute 0 and multiply it with 1 - x 1 we have 1 - x 1 times x 2 which is equal to 2 x 1, no this was actually 3 x 1 x 2. So, it is x 1 times this polynomial so it was 3 x 1 x 2 so we have 2 x 1 x 2 + x 2. In other words what we are doing is that we are reducing the degree of x 1 to 1 and then we are just collecting the terms. So, this operator has that property so what Shamir did was so he said that together with there exist I mean together with the sum and the product operator.

If I intersperse them with operators with of this kind which will reduce the degree at each step also and then come up with an appropriate protocol. So, that will be able to take care of the class PSPACE or this language TQBF as well so that was. So, we will talk a little bit more about this next time. So, let us stop here.