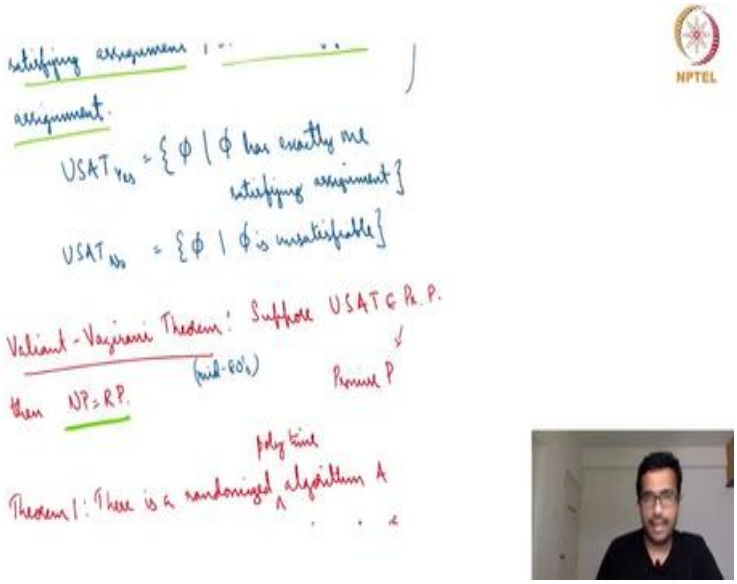
Computational Complexity
Prof. Subrahamanyam Kalyanasundaram
Department of Computer Science and Engineering
Indian Institute of Technology, Hyderabad

Lecture - 48
Valiant Vazirani Theorem Continued

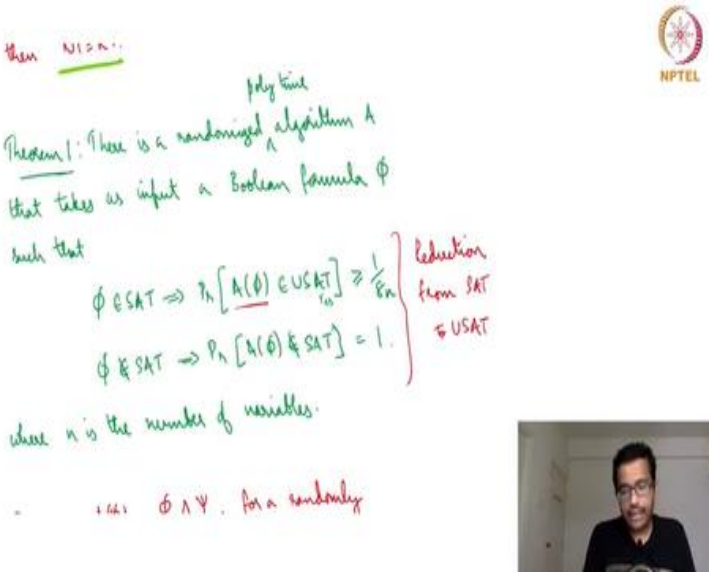
(Refer Slide Time: 00:15)



Hello and welcome to lecture 48 of the course computational complexity. In the previous lecture we saw promise problems and then we saw the unique SAT problem. So, unique SAT is just like SAT but with the promise that the number of satisfying assignments for the given boolean formula is either 1 or 0 and we have to determine which of this is the case. And then we stated the Valiant Vazirani theorem which said that if unique SAT has a polynomial time algorithm then NP = RP.

Or in other words since we believe NP is or since we feel NP is unlikely to be equal to RP it is unlikely that a unique SAT will have a polynomial time algorithm. So, this is the Valiant Vazirani theorem. So, now in this lecture we will see the proof of this fact.

(Refer Slide Time: 01:12)



So, the bulk of the proof can be stated in this following technical statement. Maybe I will just give it a different colour. So, maybe I will give it a green colour. There is a randomized polynomial time algorithm A that takes a boolean formula and reduces it to another boolean formula or constructs another boolean formula such that basically it is a reduction from a satisfiability to unique satisfiability. So, it is actually a reduction from SAT to unique satisfiability.

So, what does it do? It takes phi A boolean formula as input and produces A phi. If phi is satisfiable then A phi will be in the s instance of unique satisfiability. So, this is USAT s and if phi is not satisfiable then A phi will not be satisfiable. So, when I say unique satisfiability s instance it means A phi will have exactly one satisfying assignment and end with some probabilities.

So, if phi is satisfiable then the probability that the reduced or the A phi which is the output of the randomized algorithm which is another boolean formula. The probability that A phi will have a unique satisfying assignment is at least 1 divided by 8 n and if it is not satisfiable then if phi is not satisfiable then A phi is not satisfiable with probability 1. So, the second clause is definite. This is for sure.

If it is a no instance meaning you start with an unsatisfiable boolean formula then we will get an unsatisfiable boolean formula. But if we start with the satisfiable boolean formula we want to get to a boolean formula. That has exactly one satisfying assignment and that happens with some probability. So, unsatisfiable remains unsatisfiable whereas satisfiable we may or may not get into the unique SAT s instance.

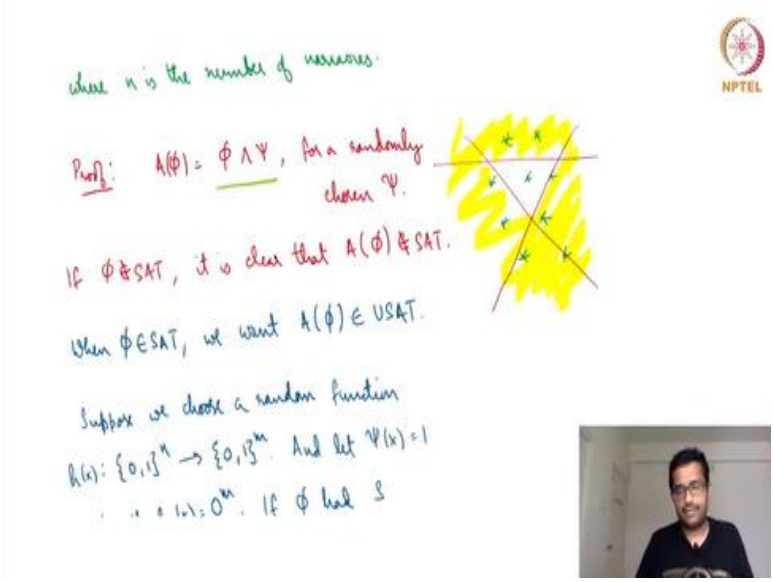
So, this is like a one-sided reduction and also this reduction is a probabilistic reduction. So, it is a randomized reduction from satisfiability to USAT. So, using that we now you may see the proof of how this implies NP = RP. So, if USAT has a polynomial time algorithm we have already shown it or if theorem 1 is true that implies a reduction from satisfiability to USAT. So, if USAT has a polynomial time algorithm this implies that any language in NP can be reduced to USAT in randomized polynomial time.

And then we can use the polynomial time decider of USAT to decide the original NP language. So, we have a randomized polynomial time algorithm for any language in NP which is what the statement says NP = RP. So, that is a statement and this is the main theorem that we will spend most of the time on. Given a boolean formula phi we will show a randomized procedure randomized algorithm by which we will construct another formula A phi.

Such that if phi is satisfiable A phi is likely to have a unique satisfying assignment. And if phi is not satisfiable then A phi is not going to have a unique satisfying assignment or it is not going to have a satisfying assignment at all. So, it will remain unsatisfiable and the idea is very simple. A phi is simply phi and with some other formula psi and where psi is randomly chosen. So, it is immediately clear that if phi is not satisfiable then A phi also will not be satisfiable.

Because it is not possible to satisfy phi itself. So, how can you satisfy phi and something else. So, the second part, the fact that phi is not in SAT implies this part that I am underlying this part is now immediate because of the construction of the formula A phi.

(Refer Slide Time: 06:02)



What we will what remains to be shown is that when phi is satisfiable if A phi has exactly one satisfying assignment with a good probability and the good probability we will shoot for is 1 divided by 8 n. And because it is a one sided error we could easily boost it by just repeating and we do not even need things lecture not bound. So, what is the idea here? I will first tell you a very high level idea. So, suppose what we will do is we will choose a phi.

So, think about a satisfying assignment. There are 2 power n possible assignments. So, let it have n variables. So, n is in fact I missed to say that n is the number of variables. So, this is 1 by 8 n where n is the number of variables. So, there are 2 power n assignments possible satisfying or unsatisfying whatever. And we will choose the randomly chosen psi in such a way that psi will kind of cut many assignments. Psi will make many of the assignments as not satisfying.

So, if there were 100 assignments, psi will cut it by enough numbers. So, what is likely to remain is exactly one satisfying assignment. So, it is basically you can think of psi as cutting every time you can say you chop the space into half. So, now whatever number of let us say phi had 10 satisfying assignments you can think of psi being cutting this space of satisfiable or all assignments many times 2, 3 times so that 10 becomes 5, 5 becomes 2 and so on.

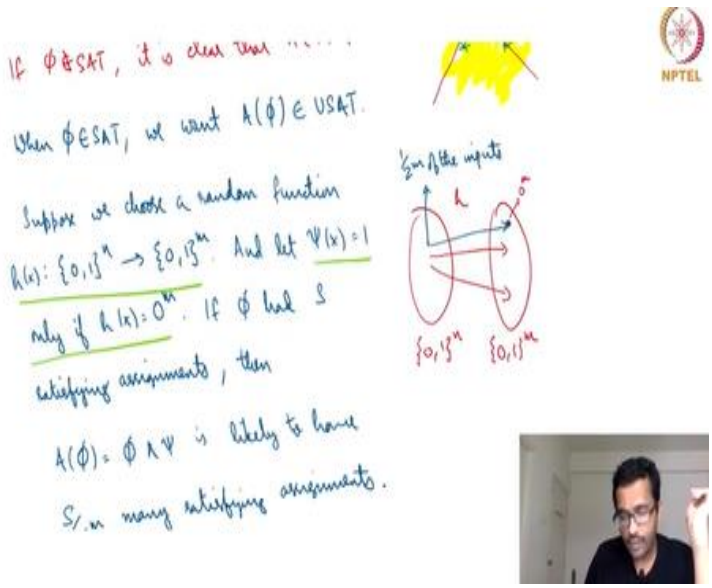
And until it becomes there is a good chance of getting a unique satisfying assignment. So, basically psi just eliminates a portion of the assignments making them unsatisfiable and because

it is randomly chosen we expect it to also eliminate a portion of the satisfying assignments. So, all that will remain is one small fraction of satisfying assignments which we hope will be very close to 1. This is a very very high level idea.

So, maybe just to draw a pictorial representation of course I am drawing it in 2D. But things happen in n dimension. So, maybe psi is like a series of cuts of the half of the plane. So, let us say first the bottom part is eliminated. This yellow part is eliminated and with the second cut this part is eliminated and with the third cut let us say this top part is eliminated and leaving just 1. So, psi will be a series you can think of it as a series of cuts.

And what remains is every time let us say the space becomes half and what remains will be very likely to be one satisfying assignment.

(Refer Slide Time: 09:13)



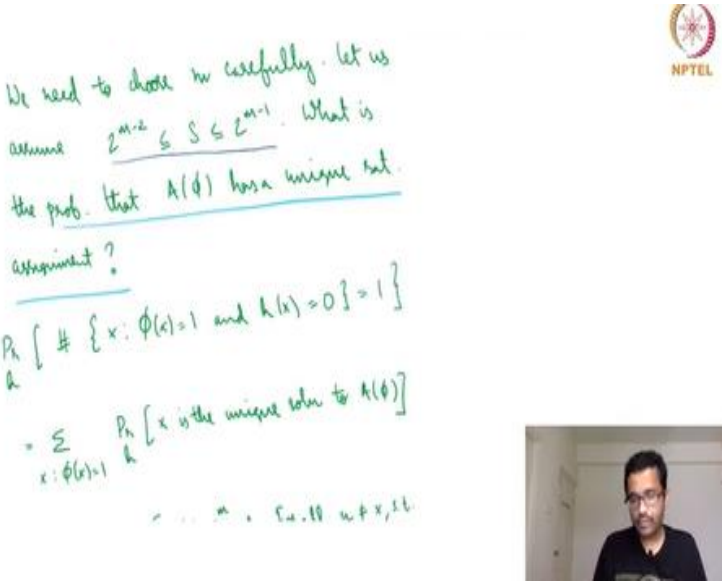
So, now we will see that this is just a high level intuition. Now we will see the more formal parts of the proof. So, what we will set psi to be we want it to be a random function. So, a randomly chosen boolean formula which we will derive from a random function. So, psi will be derived from a random function h. So, think of h as a boolean function that goes from 0, 1 power n to 0, 1 power m. So, n to m and we will say what is m later.

And psi is the formula that corresponds to if h of x is equal to all zeros. So, it is a random function. So, roughly you will expect h of x will distribute. So, it is like this. So, this is a function from 0, 1 power n to 0, 1 power m. So, you expect h to kind of it is a random function. So, h will distribute all the inputs to a random output on the right hand side. So, how many inputs do you expect to go to the all 0 output? You expect roughly equal numbers to go to each output.

So, you expect 1 divided by 2 power m of the inputs to go to the random to the all 0 output. So, you expect 1 divided by 2 power m of the inputs to be mapped to the all 0 output. So, we choose h in such a way that or h will be chosen in such a way that the number of satisfying assignments will be roughly of the order of 2 power m so that 1 by 2 power m of the inputs means it will be roughly of the order of 1.

So, again what I said earlier just written down if phi had s number of satisfying assignments then A phi which is just phi and psi. So, it is just a fraction of the satisfying assignment that goes to all 0s and we have already said that the fraction is likely to be 1 divided by 2 power m. So, it is likely to have s divided by 2 power m many satisfying assignments. So, now all that we need to do is what should be m?

(Refer Slide Time: 11:58)



So, for that we look at what is s where s is the number of satisfying assignments. One may wonder how you choose s. But we will explain that as we go along because given a boolean

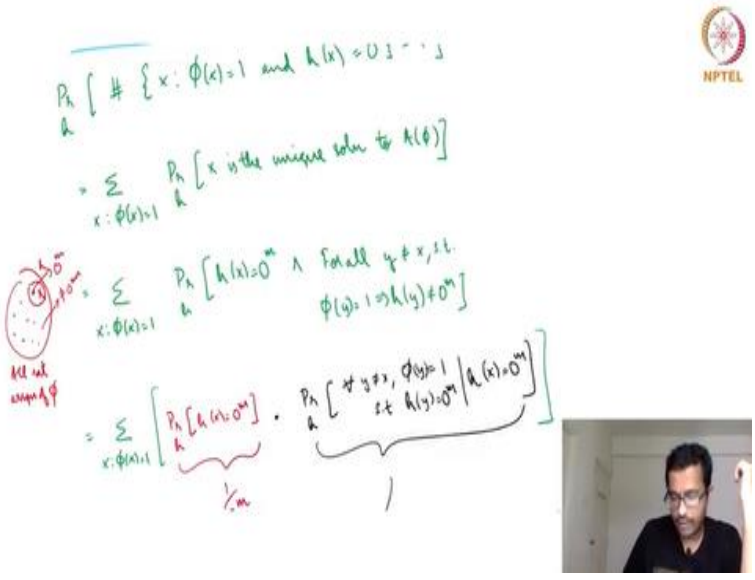
formula we do not know if it is satisfiable or not satisfiable. Forget how many satisfying assignments it has. We do not even know that but our process will be such that even without knowing s we will do something that gives us a good probability.

So, let us assume that the number of satisfying assignments is between 2 power m - 2 and 2 power m - 1 for some m. So, we know that the number of satisfying assignments could be as much as all the assignments which is 2 power n or it could be as low as 0. So, it could be anything between 0 and 2 power n. So, let us assume that it is in the range 2 power m - 2 up to 2 power m - 1. Now what is the probability that A phi has a unique satisfying assignment.

And this is the most complex problem calculation that we will do in this lecture. And I say most complex I mean in this lecture but it is not very difficult. It is just very basic probability and conditional probability. So, now what is the probability that A phi has a unique satisfying assignment? This is the probability that we are going to calculate. So, in other words the probability is taken over h the number of satisfying assignments of phi being such that A phi has a unique satisfying assignment what is the probability that this is equal to 1.

So, this is A phi is nothing but phi and psi where psi corresponds to h of x is equal to all zeros. So, h of x = 0 power n.

(Refer Slide Time: 14:13)

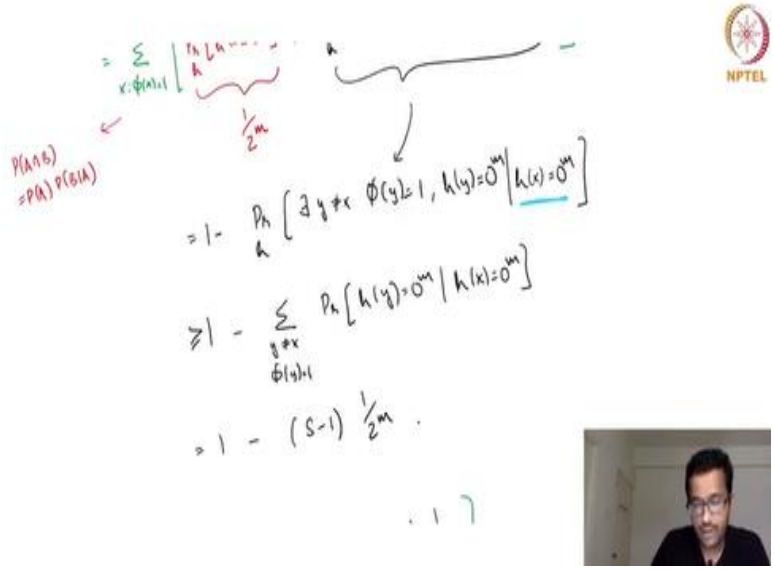


So, in other words we can say one thing it can do is we can consider this to be a summation over each x and where x is from 0, 1 where x is a satisfying assignment. There x is a satisfying assignment of phi. So, what is the probability that this is likely to be the unique satisfying assignment for A phi. Because if it is not a satisfying assignment for phi it cannot be a satisfying assignment for A phi.

So, for each satisfying assignment of phi we are seeing the probability that it can be a unique satisfying assignment for A phi. And these are all mutually exclusive events so you could just take the sum. There is nothing lost here. It is still equality. So, what is it in other words what do we want? So, we are only saying that if you look at all the satisfying assignments let us say this is a set of all satisfying assignments.

We want this x to be alone mapping to all zeros, h should map this to all zeros. Everything else should not map to all zeros. All the remaining things should not map to all zeros which is what I have written here. For all the y that is not x if it is satisfying phi then h of y should not be all zeros.

(Refer Slide Time: 15:59)



So, I can further split it in the following sense in the following way. I am sorry for this bit of technical trouble. I can split it like this. I can take the probability that h of x is all 0s which is the first part and this and can be split like given h of x is all 0s. What is the probability that? All the

remaining y do not get mapped to all 0. So, there is a small type here that does not get mapped to all 0s. For all the remaining y that are not equal to x they do not get mapped to all 0s.

So, I will say it again we want for each x that is outside summation we want h of x to be mapped to 0 and for all the remaining solutions of phi satisfying assignments of phi let us say we call them y that are not equal to x. We want h of phi to be not equal to 0, h of y to be not equal to 0. And because we are multiplying these things we need to make this a conditional probability. So, this probability of A intersection B is simply probability of A into probability of B given A which is what we have used here.

Maybe I should just write that. What we are using is probability of A intersection B is probability of A multiplied by probability of B given A. So, A being the event that h x is all 0s and B being the event that for all y that is not equal to x, h y is not 0. So, what is the probability that h x = 0? So, as I said before h x is a random function, what is the probability that it maps an x into all 0s? It is just simply 1 by 2 power m because it is 2 power m possible output.

So, this is simply 1 by 2 power m. Now we will just focus on the black part which is the part that is written over here. Probability that for all remaining y's that are not x, h y is not all 0s. What we can do is to consider the complement event, 1 minus the probability that there is a y for which the h of y maps to all 0s. So, instead of saying for all the y's h of y should map to non zero. We can say 1 minus the complement event, complement event means though there is a y for which h of y maps to all zeros.

Again the condition remains the condition being h of x is all 0s. So, we use a complement event in the condition setting and the probability is taken over all the h. Now we have the second term 1 - probability of h. This probability term now we can take each and every satisfying assignment of phi and check the probability that the h of y is 0. So, instead of saying that x is a y for which the problem h of y is 0 we can just add up the probabilities each y, h of y is 0.

Because all we are doing is we are taking the union bound of all these events the probability that h of y is zero so when we take the union bound it is an upper bound. But we have a negative sign

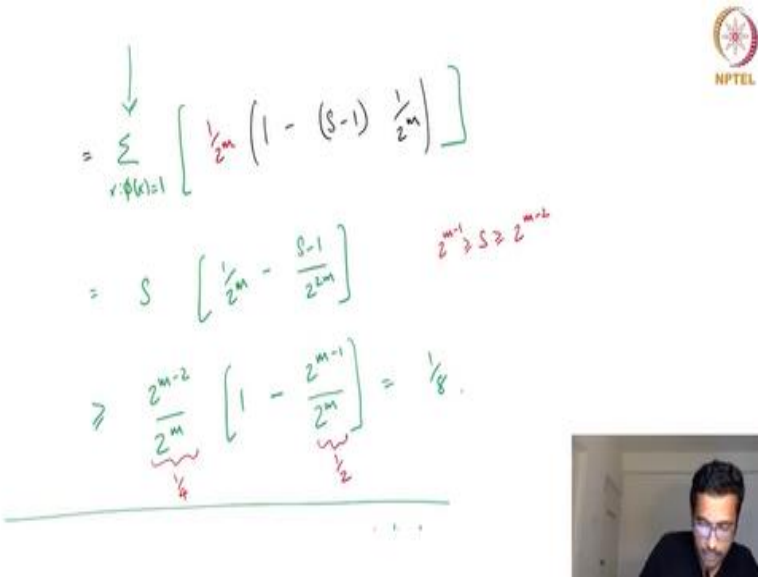
over here so it gives a lower bound. So, we have a union bound which is an upper bound but we have taken the negative sign. So, it is a lower bound. So, what we are doing is we are replacing this probability with the summation over all the y's such that y is a satisfying assignment of phi and y is not equal to x.

What is the probability that h of y is 0 given h of x is 0? It boils down to that. What is the probability? So, h of x being whatever it is, h is a random function. So, what is x mapped to has no bearing on what is y mapped to. So, this is simply the probability that h of y is also mapped to all 0s or h of y is all 0s. So, the probability is 1 divided by 2 power m. And how many y's are there? Let me turn up the light because it is getting dark.

By assumption we said that there are s satisfying assignments for phi so there are s - 1. So, once we keep x aside we have s - 1 satisfying assignment y that is not equal to x. So, this is simply 1 minus s - 1 into 1 by 2 power m. So, the red part was simply 1 divided by 2 power m and the black part is 1 - s -1 into 1 by 2 power m. So, now fitting this back into these big green square brackets what we have is so the outside summation still remains x being.

So, we are just moving back over from here. So, this red part is 1 by 2 power m and the black part is what we calculated. So, we just substitute it back 1 by 2 power m into this.

(Refer Slide Time: 22:38)



And this summation so now the inside part is independent of phi or anything. So, now how many terms are there? How many x's are there? The number of x's is simply s. So, I can just replace the summation by just the quantity s and if you just see what happens here you have 1 by 2 power m in the square bracket - s - 1 divided by 2 power 2 m. Because this 2 power m outside and the 2 power m inside gives rise to 2 power 2 m sorry, and now we want to lower bound this again.

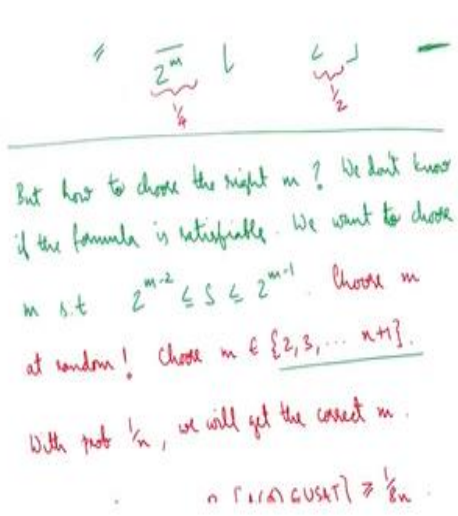
So, we earlier said that s lies between 2 power m - 2 and 2 power m - 1. So, the lower bound of s is 2 power m - 2 and let me use a different color. And this s -1 needs to be upper bound because it is a negative sign. So, that is we take the upper bound because of the sign and we can take this 2 power m outside. So, this 2 power m comes outside and what we get is 1 by this. This reduces to 1 by 4 and this becomes 1 by 2. So, 1 by 4 multiplied by 1 - 1 by 2 which is 1 by 2 again.

So, this probability turns out to be 1 by 8. So, all we did is a simple calculation of a random function. What is the probability that this random function maps a certain value to all zeros. This is what we are doing here. So, the probability that maps x to all 0s is 1 by 2 power m and the rest was just a standard separation of probabilities into different cases and adding them up or subtracting them up.

So, what we get at the end is what is the probability that A x or A phi has a unique satisfying assignment that is 1 by 8 provided phi has a satisfying assignment in the range 2 power m - 1 and 2 power m - 2. If it has in this range the probability that A phi has a unique satisfying assignment is 1 by 8. But you may notice that we choose m the number of bits m in such a way that there is roughly 1 by 2 power m probability of a satisfying assignment being selected.

So, if there were more satisfying assignments our m would have been bigger. If there are less satisfying assignments our m would have been smaller. So, this is what I meant by cutting it to the correct size to whatever is the required size.

(Refer Slide Time: 26:05)

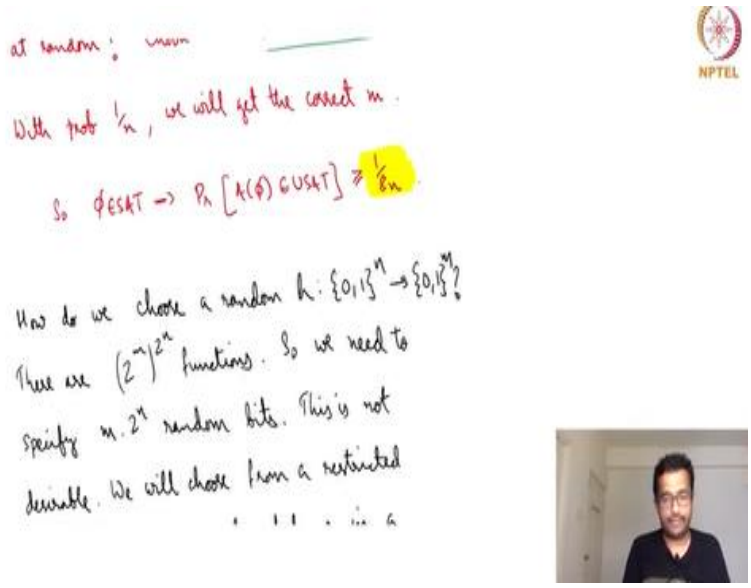


But how do we choose the m itself? How do we choose the m? Because we forget the number. We do not even know if the given formula is satisfiable. Again, what we do is we again go back to randomness and we choose m randomly. So, we know that the number of satisfying assignments could be as high as 2 power m and as low as 1 assuming it is satisfiable. The unsatisfiable case we do not care at all because we are just taking the and with another formula.

So, the unsatisfiable formula remains unsatisfiable. So, this 2 power m - 2 or 2 power m - 1 could be as big as n. So, m should be chosen could be as big as n + 1 and the lower bound could be as small as 1. So, m could be as small as 2 because 2 power m - 2 has to be equal to 1. So, we choose m from this range 2 to 3, 4, 5 up to n + 1 uniformly at random. And out of these choices there are n such values 2, 3 up to n + 1 and any of these possibilities occur with probability 1 by n.

And with probability 1 by n you are in the correct window where the above probability calculation applies. So, if the formula is satisfiable the probability of choosing the correct m is 1 by n multiplied by given that we chose the correct n, what is the probability that the resulting A phi has a unique solution it is 1 by 8.

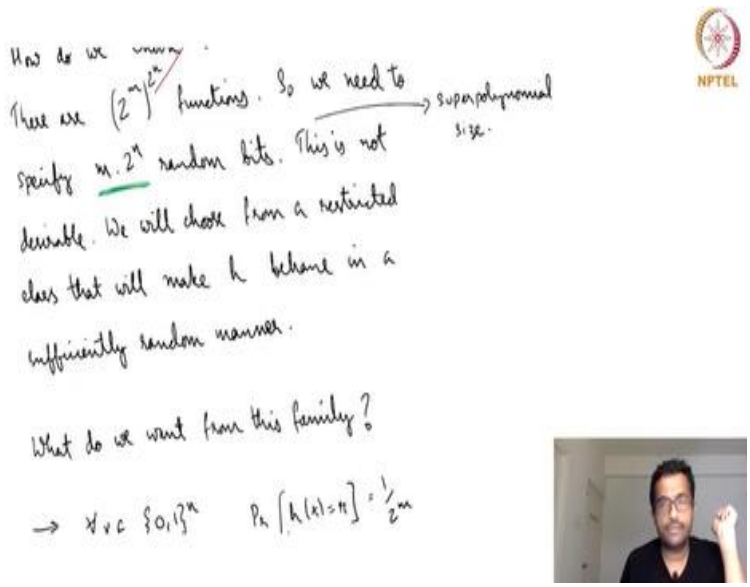
(Refer Slide Time: 27:55)



So, we get it is 1 divided by 8 n which is what we said we will do. So, we do not need to know what s is. That is the nice thing about this. We do not even need to know what s is. If it has some satisfying assignment there is some m for which that s falls in the correct window and that happens with probability at least 1 by n. And once we choose that correct m the probability of A phi having a unique satisfying assignment is 1 by 8.

So, it is 1 by n multiplied by 1 by 8 at least this. So, it is 1 by 8n. So, we have shown that by a random process we could convert phi to A phi in such a way that if phi is satisfiable A phi is satisfiable A phi has a unique satisfying assignment with probability at least 1 by 8 and if phi is unsatisfiable then A phi is also unsatisfiable. So, SAT gets reduced to unique SAT with probability at least 1 by 8n. So, it seems like we are done. But we are not done. We will see why.

(Refer Slide Time: 29:20)



We said that we need to choose a random function from 0,1 power n to 0, 1 power m. How many such functions are there? So, given each input, the input is an n bit vector there are 2 power m possible values to assign. So, 2 power m possible h could be any of the 2 power m possible values. So, there are 2 power m whole power 2 power n such functions. So, what I mean is 2 power m whole power 2 power n such functions.

In other words this is nothing but maybe I will just write it outside. This is nothing but 2 power m multiplied by 2 power n. This is the quantity and to specify a function that out of these many functions we need to describe it with the log that many bits. So, the number of bits required to describe this function is nothing but m times 2 power n random bits are necessary. So, just to choose a function from all the functions require m multiplied by 2 power n random bits.

So, just in the time we take to generate these random bits even if each random bit is generated per one time step even this itself is more than polynomial. This is not polynomial or this is not desirable because it is a super polynomial size. Super polynomial means something that is bigger than a polynomial. So, the number of bits needed to describe this function is super polynomial. And recall our goal was to do a reduction in randomized polynomial time.

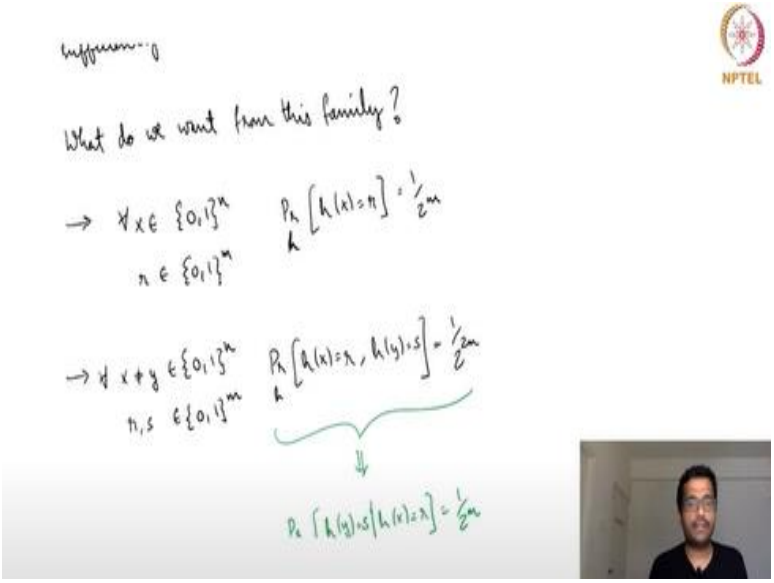
This is a randomized polynomial time is what I said at the beginning. Randomized polynomial time algorithm A such that which takes as input formula phi and outputs A phi. So, we cannot do

that. This is not possible. So, what else can we do? So, what we notice here is where did we use the loss of probability here. So, we use the loss of probability at two places I think or we will use the when I mean loss of probability we use it throughout.

But where did we use the function specific probability? One is over here we said that the probability of a specific x going to 0 power m is 1 divided by 2 power m. This is one place where we use probability and the second is this point. The probability that h y = 0 power m given h x is 0 power m is 1 divided by 2 power m. So, here it is like independence that we are saying regardless of what x suppose we did not know what x was, what is the probability that h y was 0 power n?

It would be 1 divided by 2 power m. But now we are saying that even given h x is 0 power m even then it remains the same. So, it is like independence. So, these are the two places where we use I will erase this, here and here. These are the two places where we use the probabilities of the function and these are the only two places where we use the probability of the function and the rest was this arithmetic in calculation.

(Refer Slide Time: 33:11)



So, all we need is a random value should be mapped to all 0s with probability 1 by 2 power m and even if x is mapped to 0 power m with 1 by 2 power m, y should be mapped independently. So, what we really want is we need not want all the functions. We can do with the restricted class

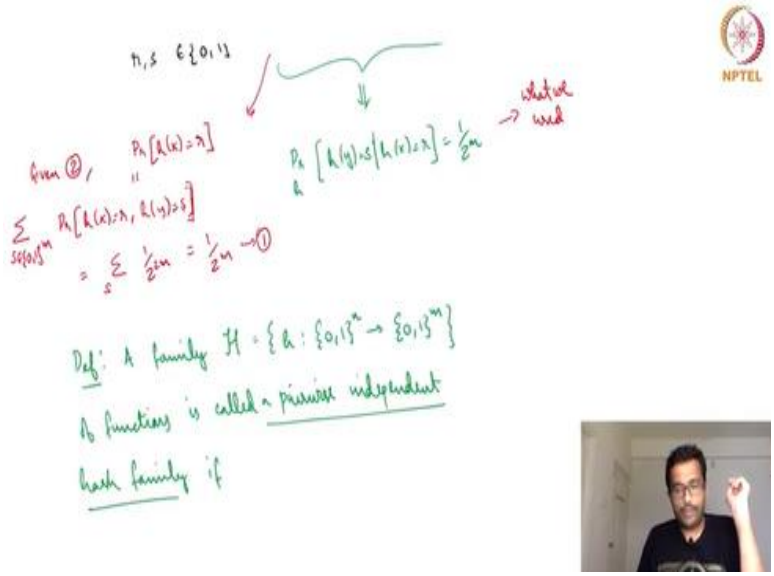
of functions as long as it has enough independence in it. So, what we want is I am summarizing what we want here in these two rules.

One is that given any x in 0, 1 power n and any r in 0, 1 power m, what is the probability that h x is equal to r? This probability should be 1 divided by 2 power n. This is property 1 and second is that given x, y where x is not equal to y and given r, s in 0, 1 power m r and s may or may not be the same it does not matter. What is the probability that h x maps to r and h y maps to s? This should be simply the product of the probabilities 1 by 2 power m whole square or 1 by 2 power 2 m.

So, this is the independence part. If this is true you automatically see that the bottom part also gets true because what is the probability that h y = s given h x = r. So, it is the probability that h y = s and h x = r divided by the probability that h x = r which we already calculated to be 1 by 2 power m. So, this is 1 by 2 power m. So, this is what we used actually. So, we used one and then what we infer from two.

But notice that even one can be derived from two independently without we do not need one stated explicitly. Because given two we could just take the summation over all the possible different y's sorry all the possible values of s we can take the summation over all the possible values of s. And if you take the summation you will get 1 by 2 power 2 m multiplied by 2 power m. So, 2 power m will cancel and what we will have is the probability that h of x = r is 1 by 2 power m.

(Refer Slide Time: 35:54)



So, what I am saying is that given this given two probability of so if you just summation over all s in 0, 1 power m probability that h x = r and h y = s this is equal to it is simply summation over all s times 1 by 2 power 2 m which will be 1 by 2 power m. But this quantity is actually nothing but probability that h x = r. Because we are allowing h y to be, we are just adding all the possibilities that h y can take. So, it is simply probability that h x = r so which is x exactly 1.

So, what we are trying to show or what we really want is the property that we want h need not be a completely random function and it can be using a restricted randomness and a restricted randomness is sufficient for us. And this particular restricted randomness that we require is called pairwise independence. So, basically pairwise independence means h should be chosen in such a way that the images of two x and y two different x and y should be very independent.

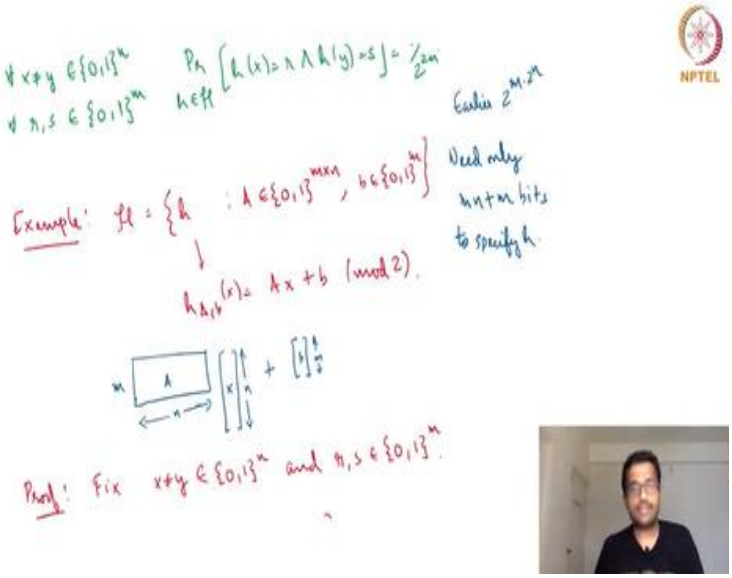
It need not happen that three different x, y and z the images are independent. It is enough that two are independent and that is good enough for us. And what we will see is that the number of random functions is 2 power m multiplied by 2 power n. But we will now see a much smaller class which has pairwise independence and we will see that it is sufficient to work in this class. It is enough to work in this class.

So, let me define what a pairwise independent hash family is. So, these hash functions are called hash functions. So, family script h is called a pairwise independent hash family. So, where h is

all the family is functions from 0, 1 power n to 0, 1 power m. If it is the property 2 that I have written here x and y are not the same from 0, 1 power n, r and s from 0, 1 power m. If you randomly choose a small h from the family what is the probability that h x is r and h y is m?

This probability should be 1 divided by 2 power 2 m. Now all that remains to show for our earlier proof is that we can specify the random h in using polynomially many bits, m multiplied by 2 power n was not so polynomially many bits.

(Refer Slide Time: 39:08)



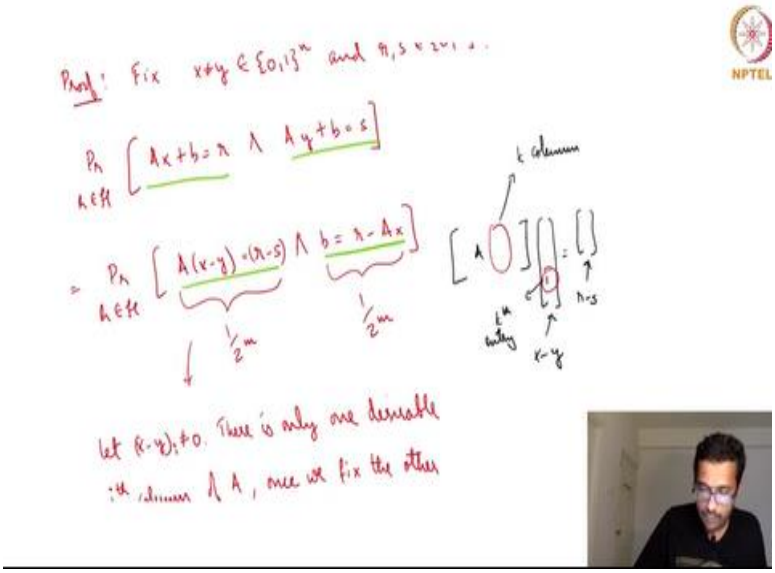
So, we will see one example again. There are different ways of constructing pairwise independent hash functions. We will see one example which is fairly simple and which will convince you that we can do this and that would be enough for us to see the proof. So, given x and y two different x and y we want h of x to be mapped to r and h of y to be mapped to x. So, what we do is we have a matrix A and a vector m or vector b.

So, a is m rows and n columns and b has sorry I think I am writing it wrong b should be like this b has a single column. It has m rows it is like this. But x is 0, 1 power n. So, x is 1 power n. So, it is like this. So, this Ax + b is the function. So, we choose A randomly, b randomly. A is a 0, 1 matrix, m by n matrix b is a 0, 1 vector m of length m. So, here now how many bits does it choose to specify h?

Once we fix A and b the function h is fixed and we require only m n bits to specify A and m bits to specify b. So, we need only m n + m bits to specify h. Earlier the number was something like 2 power m multiplied by 2 power n or something like that. So, this was very high 2 power m multiplied by 2 power n and now we are seeing that it can be much smaller. So, all that remains to show is this and everything happens modulo 2 x is a 0 1 vector, A x + b will happen in modulo 2.

So, now all that remains to show is that this probability satisfies this condition that I have circled over here. When x and y are randomly chosen or x and y are chosen such that x is not equal to y and r and s are chosen then the probability of this happening is 1 by 2 power 2 n.

(Refer Slide Time: 42:20)



So, suppose we choose x and y and r and s in this manner. What is the probability that h x = r and h y = s? So, h is simply A x + b and h of y is simply A y + b. So, these two have to be satisfied. Now I can rearrange this. If these two are true then you can get that A multiplied by x - y = r - s and if you take the difference you will get that once you solve for A then you can get b to be r - A x or s - A y.

But those two are equivalent like you can use any one of them. Now this is asking what is the probability that A multiplied by x - y is equal to r - s. So, what is the probability that A multiplied by a certain vector is r - s? So, I will just explain. So, this is A and multiplied by a

certain vector. So, let us say this vector is x - y this is equal to r - s. What we know now is that x and y are not equal. So, that means there is some entry here.

Let us say the kth entry that is not 0 so that is 1 kth entry is 1. Now look at the kth column here. Suppose everything outside the kth column was fixed. All the A's except the kth column were fixed and you multiply x - y with the rest of A and now we know that the kth entry in x - y is 1 and we are trying to get it to sum to a specific r - s. Now whatever be the remaining entry sum there is only one value for each of the entries in the kth column here.

That will make it happen to be equal to r - s. So, each of these has to be fixed to some particular entry. Since there are m rows there are m entries in the kth column that have to be chosen in a way that is desirable for us. So, each of these m entries has to be 0 or 1 depending on how the others are chosen and whatever it is we can determine the probability. The kth column happens to be the way we want it and that probability is 1 divided by 2 power m.

So, each of these m entries fall in the correct way. And now whatever is once A is fixed r – A x is just another vector and b we want to b = r – A x. So, again b is just a vector of m entries what is the probability that b is equal to that vector? Again b was randomly chosen. Capital A was randomly chosen, capital B was randomly chosen. And each entry of b = r – A x. Again there are m entries each one of them should be chosen in a similar way in the favourable manner.

So, it is 1 by 2 power n for that as well. So, this is 1 by 2 power m. The choice of A and the choice of b is also 1 by 2 power n. So, again I have written here if x minus the ith column so here I have set ith column maybe I will just change it to kth column. Since x is not equal to y we know that it is not equal to there is some entry of x - y that will not be 0. So, both of these are set to 1 by 2 power m and the choice of A is independent from the choice of b.

So, this probability is 1 divided by 2 power 2 n. So, it is just multiplication that you can see which is what we were seeking as well. So, we wanted the probability to be 1 by 2 power 2 m and we have shown that it is 1 by 2 power 2 m.

(Refer Slide Time: 47:11)

And as I said earlier it requires this description of A and b requires m n + m bits and you contrast this with what we had earlier which was 2 power m whole power of 2 power n which was much more than that. Earlier we required m times 2 power n bits. So, which is a polynomial number, this is a polynomial number. Again, just to summarize now we have shown the existence of these pairwise independent hash functions.

And if we can just use these hash functions instead of the random h this will be a much smaller class of functions. But if you choose these functions randomly from this family this will have the properties that we need. This will not have all the independence. But we do not need all the independence. This is the extent of independence that we need.

(Refer Slide Time: 48:23)

So, given a formula phi we first have to choose m uniformly at random from this range from 2, 3 up to n + 1. Once we choose m, we choose a function h small h from the pairwise independent hash family of functions. So, we need to choose capital a and small b. So, this requires m n + m random bits. And then you output the formula phi which is the same as what was input and a boolean formula that checks whether A x + b = 0.

So, the boolean formula will have the same n variables x 1 to x n and it will check whether A x + b = 0. So, we can write that in a boolean formula and this formula we know that if phi is unsatisfiable this entire thing will be unsatisfiable. And if phi is satisfiable we saw that the probability of this formula having a unique satisfying assignment is at least 1 by 8 n and which is what we claimed. This is the proof of theorem 1.

So, I will just summarize the proof of theorem 1 again. So, given a formula phi randomly we construct boolean formula A phi such that if phi is satisfiable A phi has a unique satisfying assignment with probability 1 by 8 n and if phi is unsatisfiable then A phi is also unsatisfied. A phi is just A and of phi with another formula which will make sure that if it is unsatisfiable it remains unsatisfiable. And if it is satisfiable we want to cut down on the solution space.

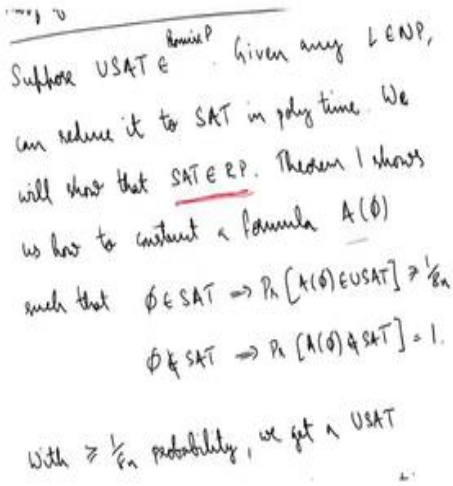
So, to cut down we need to know or we would like to know how many solutions it has to begin with and then we cut down the space accordingly. So, we do not need to know the exact number,

we would like to know the order of the number of solutions. And we choose h in such a way that the likelihood of the solution space being cut to some number that is close to 1 is high. But we do not even know the number of solutions. So, we randomly choose that number as well.

And finally we saw that the number of random functions we cannot choose from the space of all the functions at random. Because the number of functions is high. Even to specify a function at random requires exponentially many bits. What we will do is that? We will choose from a smaller family of functions which has all the required properties that we want. This family is called pairwise independent hash family which has the property that we want.

That is for two values x and y where x is not equal to y probability that h x = r and h y = s is 1 divided by 2 power 2 n. And we finally saw how to construct this family and one example of one such construction using just simple matrices and mod 2 arithmetic. And this family requires only polynomially many bits much smaller than the exponentially many bits if you choose the entire function space.

(Refer Slide Time: 52:06)

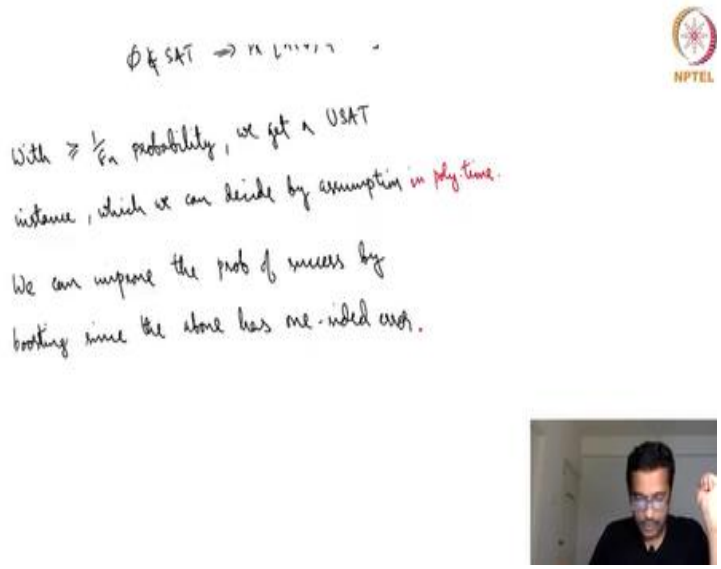


So, again just to summarize the Valiant Vazirani theorem which said that if USAT is in promise P, if USAT has a polynomial time algorithm then maybe I will just write promise P. Then the claim was L NP = RP. So, given any language L in NP we can reduce it to SAT in polynomial

time. That is because that is NP complete. Now we will show that SAT is an RP. Why? Because given a boolean formula we want to solve SAT satisfiability.

We have already seen by theorem one we have seen how to construct a formula A phi such that it has a unique satisfying assignment with high probability A phi is satisfiable. Otherwise it does not have any satisfying assignment. Basically, we reduce SAT in a randomized manner to USAT and then the assumption is that USAT is in promise P.

(Refer Slide Time: 53:16)



So, we can decide USAT in polynomial time. So, we started with the random arbitrary language L in NP, we reduce it to SAT and then we reduce it to USAT using a randomized reduction. So, that reduces any language in NP to any language L in NP in randomized polynomial time which is RP. So, we get a one-sided error, that is why it is RP. And to improve the probability of success it is not very difficult because it is a one-sided error.

So, we can just repeat if any of the ones output it is satisfiable we just output that it is satisfiable. Because whenever it is unsatisfiable it is for sure it will always output unsatisfiable. So, we can also improve the probability of success. So, again improving the probability of success will not only require polynomially many iterations. So, that concludes the proof of the Valiant Vazirani theorem.

So, this is interesting that even when you have this promise that the given formula has either 0 or 1 satisfying assignments even then we do not expect that it has a polynomial time algorithm. So, that is the main take away from this lecture. So, I have already summarized the proofs and with that I think I will conclude. Thank you.